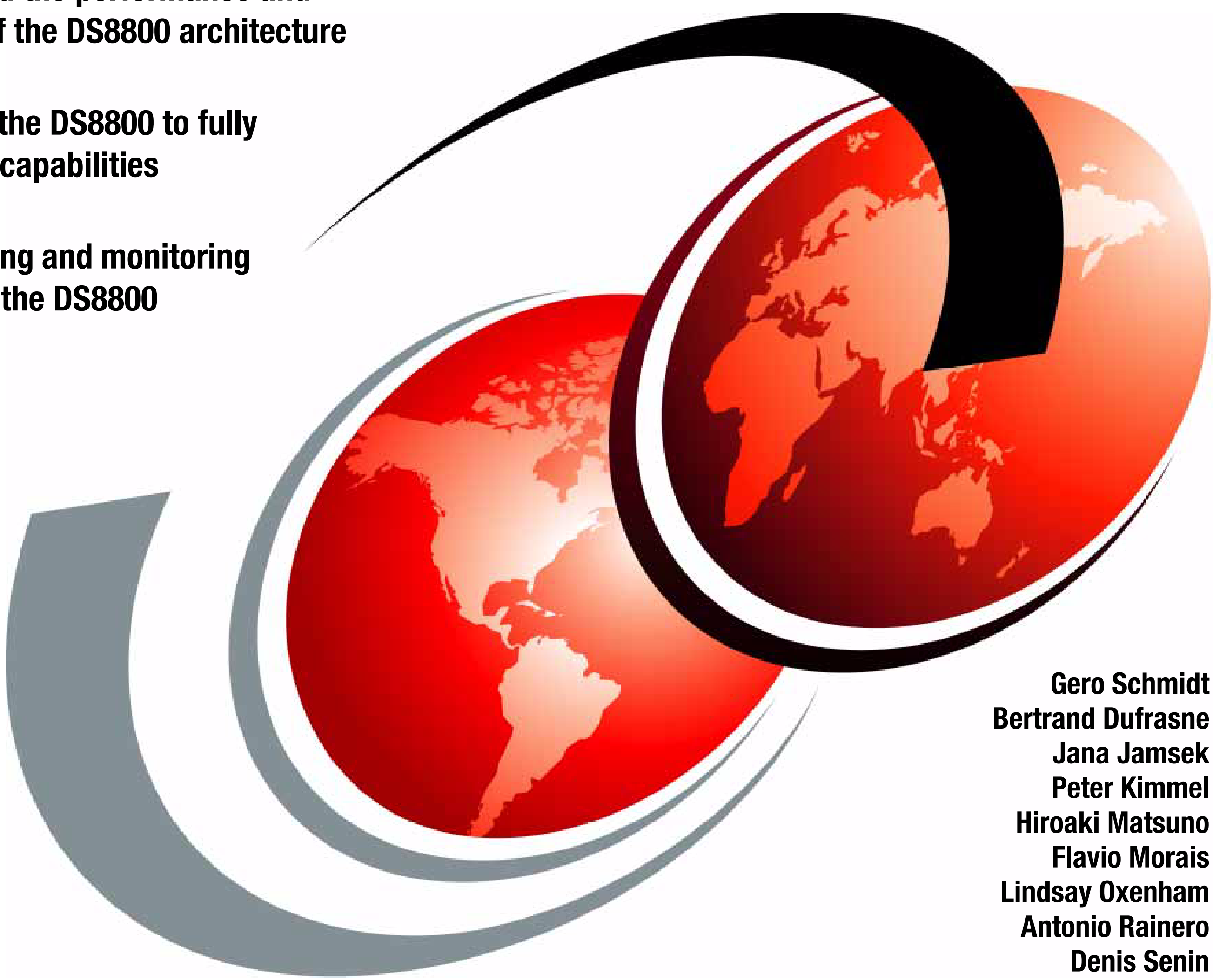
IBM

DS8800 Performance Monitoring and Tuning

Understand the performance and features of the DS8800 architecture

Configure the DS8800 to fully exploit its capabilities

Use planning and monitoring tools with the DS8800



Gero Schmidt
Bertrand Dufrasne
Jana Jamsek
Peter Kimmel
Hiroaki Matsuno
Flavio Morais
Lindsay Oxenham
Antonio Rainero
Denis Senin

Redbooks

ibm.com/redbooks

International Technical Support Organization

DS8800 Performance Monitoring and Tuning

July 2012

SG24-8013-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xv.

First Edition (July 2012)

This edition applies to the IBM System Storage DS8700 with DS8000 Licensed Machine Code (LMC) level 6.6.2x.xxx and the IBM System Storage DS8800 with DS8000 Licensed Machine Code (LMC) level 7.6.2x.xxx.

Contents

Notices xv
Trademarks xvi

Preface xvii
The team who wrote this book xvii
Now you can become a published author, too! xix
Comments welcome. xix
Stay connected to IBM Redbooks xx

Part 1. DS8000 performance considerations 1

Chapter 1. DS8000 characteristics 3
1.1 The storage server challenge 4
1.1.1 Performance numbers 4
1.1.2 Recommendations and rules 4
1.1.3 Modeling your workload 5
1.1.4 Allocating hardware components to workloads. 5
1.2 Meeting the challenge: DS8000 5
1.2.1 DS8000 models and characteristics 6
1.3 DS8000 performance characteristics overview 7
1.3.1 Advanced caching techniques 8
1.3.2 IBM System Storage multipath Subsystem Device Driver (SDD) 9
1.3.3 Performance characteristics for System z. 9
1.3.4 Easy Tier 10
1.3.5 I/O Priority Manager 17

Chapter 2. Hardware configuration 25
2.1 Storage system 26
2.2 Processor memory and cache 27
2.2.1 Cache and I/O operations 27
2.2.2 Determining the right amount of cache storage 33
2.3 I/O enclosures, PCIe infrastructure, and RIO-G interconnect 34
2.3.1 I/O enclosures. 34
2.3.2 PCIe infrastructure 35
2.3.3 RIO-G interconnect for server-to-server communication 35
2.4 Disk subsystem 35
2.4.1 Device adapters 35
2.4.2 DS8800 Fibre Channel switched interconnection at the back end 36
2.4.3 Disk enclosures. 38
2.4.4 DDMs 39
2.4.5 SAS 2.5 inch drives compared to FC 3.5 inch drives 39
2.4.6 Enterprise drives compared to nearline drives 40
2.4.7 Solid-state drives: SLC and E-MLC 41
2.4.8 Installation order 42
2.4.9 The Business Class model for DS8800 only: Feature Code 1250 44
2.4.10 Performance Accelerator feature for DS8700 only: FC1980. 45
2.4.11 Device adapter performance. 45
2.5 Host adapters 46
2.5.1 Fibre Channel and FICON host adapters 46

2.5.2 Multiple paths to Open Systems servers. 48
2.5.3 Multiple paths to System z servers 48
2.5.4 Spreading host attachments 48
2.6 Tools to aid in hardware planning 49
2.6.1 White papers 49
2.6.2 Disk Magic 50
2.6.3 Capacity Magic 50

Chapter 3. Logical configuration concepts and terminology 51
3.1 RAID levels and spares. 52
3.1.1 RAID 5 overview 52
3.1.2 RAID 6 overview 53
3.1.3 RAID 10 overview 54
3.1.4 Array across loops 55
3.1.5 Spare creation. 56
3.2 The abstraction layers for logical configuration. 57
3.2.1 Array sites 58
3.2.2 Arrays 58
3.2.3 Ranks 59
3.2.4 Extent pools 61
3.2.5 Logical volumes 64
3.2.6 Space-efficient volumes 66
3.2.7 Extent allocation methods (EAMs) 68
3.2.8 Allocating, deleting, and modifying LUNs and CKD volumes 70
3.2.9 Logical subsystem (LSS) 72
3.2.10 Address groups. 73
3.2.11 Volume access 74
3.2.12 Summary of the DS8000 logical configuration hierarchy 75
3.3 Data placement on ranks and extent pools. 76
3.3.1 Rank and array considerations 77
3.3.2 Extent pool considerations 78
3.3.3 Easy Tier considerations. 84

Chapter 4. Logical configuration performance considerations 87
4.1 Review the tiered storage concepts and Easy Tier. 88
4.1.1 Tiered storage concept 88
4.1.2 IBM System Storage Easy Tier. 89
4.2 Configuration principles for optimal performance 90
4.2.1 Workload isolation. 91
4.2.2 Workload resource-sharing. 93
4.2.3 Workload spreading 94
4.2.4 Using workload isolation, resource-sharing, and spreading 94
4.3 Analyzing application workload characteristics. 95
4.3.1 Determining skew and storage requirements for Easy Tier 95
4.3.2 Determining isolation requirements. 96
4.3.3 Reviewing remaining workloads for feasibility of resource-sharing. 98
4.4 Planning allocation of disk and host connection capacity 98
4.4.1 Planning DS8000 hardware resources for isolated workloads 98
4.4.2 Planning DS8000 hardware resources for resource-sharing workloads. 98
4.5 Planning volume and host connection spreading 99
4.5.1 Spreading volumes for isolated and resource-sharing workloads. 99
4.5.2 Spreading host connections for isolated and resource-sharing workloads 100
4.6 Planning array sites. 101

4.6.1 DS8800 configuration example: Array site planning considerations 102
4.7 Planning RAID arrays and ranks 103
4.7.1 RAID-level performance considerations 103
4.7.2 RAID array considerations 109
4.7.3 Rank considerations 112
4.8 Planning extent pools 115
4.8.1 Overview of DS8000 extent pool configurations with Easy Tier 116
4.8.2 Single-tier extent pools 118
4.8.3 Multi-tier extent pools 122
4.8.4 Additional implementation considerations for multi-tier extent pools 124
4.8.5 Extent allocation in homogeneous multi-rank extent pools 125
4.8.6 Balancing workload across available resources 130
4.8.7 Extent pool configuration examples 131
4.8.8 Assigning workloads to extent pools 138
4.9 Planning address groups, LSSs, volume IDs, and CKD PAVs 140
4.9.1 Volume configuration scheme using application-related LSS/LCU IDs 142
4.10 I/O port IDs, host attachments, and volume groups 145
4.10.1 I/O port planning considerations 147
4.11 Implement and document DS8000 logical configuration 151

Part 2. DS8000 performance management 155

Chapter 5. Understanding your workload 157
5.1 General workload types 158
5.1.1 Typical online transaction processing (OLTP) workload 158
5.1.2 Microsoft Exchange Server workload 159
5.1.3 Sequential workload 159
5.1.4 Batch jobs workload 159
5.1.5 Sort jobs workload 160
5.1.6 Read-intensive cache friendly and unfriendly workloads 160
5.2 Database workload 161
5.2.1 Database query workload 162
5.2.2 Database logging workload 162
5.2.3 Database transaction environment workload 162
5.2.4 Database utilities workload 163
5.3 Application workload 163
5.3.1 General file serving 164
5.3.2 Online transaction processing 164
5.3.3 Data mining 165
5.3.4 Video on demand 165
5.3.5 Data warehousing 165
5.3.6 Engineering and scientific applications 165
5.3.7 Digital video editing 166
5.4 Profiling workloads in the design phase 166
5.5 Understanding your workload type 168
5.5.1 Monitoring the DS8000 workload 168
5.5.2 Monitoring the host workload 169
5.5.3 Modeling the workload and sizing the system. 171

Chapter 6. Performance planning tools 175
6.1 Disk Magic. 176
6.1.1 The need for performance planning and modeling tools. 176
6.1.2 Overview and characteristics 177
6.1.3 Input data needed for Disk Magic study 177

6.1.4 Output information 179
6.1.5 Workload growth projection. 179
6.1.6 Disk Magic modeling. 179
6.2 Disk Magic for System z 181
6.2.1 Process the DMC file 183
6.2.2 zSeries model to merge the two ESS-800s to a DS8300 188
6.2.3 Disk Magic performance projection for the zSeries model 195
6.2.4 Workload growth projection for zSeries model 197
6.3 Disk Magic for Open Systems. 198
6.3.1 Process the Tivoli Storage Productivity Center CSV output file 199
6.3.2 Open Systems model to merge the two ESS-800s to a DS8300 206
6.3.3 Disk Magic performance projection for an Open Systems model. 212
6.3.4 Workload growth projection for an Open Systems model. 213
6.4 Disk Magic SSD modeling. 215
6.4.1 SSD advisor example 215
6.5 Disk Magic Easy Tier modeling. 221
6.5.1 Easy Tier prediction 222
6.6 General configuration planning guidelines 228
6.7 Storage Tier Advisor Tool 231
6.7.1 STAT output samples 232

Chapter 7. Practical performance management. 235
7.1 Introduction to practical performance management 236
7.2 Performance management tools 236
7.2.1 Tivoli Storage Productivity Center overview 237
7.2.2 Tivoli Storage Productivity Center data collection. 237
7.2.3 Tivoli Storage Productivity Center measurement of DS8000 components 240
7.2.4 General Tivoli Storage Productivity Center measurement considerations 249
7.3 Tivoli Storage Productivity Center data collection. 250
7.3.1 Time stamps 250
7.3.2 Duration 252
7.3.3 Intervals 253
7.4 Key performance metrics 254
7.4.1 DS8000 key performance indicator thresholds 257
7.5 Tivoli Storage Productivity Center reporting options 258
7.5.1 Alerts and constraints 260
7.5.2 Predefined performance reports in Tivoli Storage Productivity Center 265
7.5.3 Ad hoc reports. 265
7.5.4 Batch reports. 269
7.5.5 TPCTOOL 272
7.5.6 SAN Planner 275
7.5.7 Storage Optimizer. 276
7.5.8 TPC Reporter for Disk. 279
7.6 Monitoring performance of a SAN switch or director. 285
7.6.1 SAN configuration examples. 285
7.6.2 Tivoli Storage Productivity Center Fabric Manager alerts. 289
7.6.3 Tivoli Storage Productivity Center Fabric Manager reporting 290
7.6.4 Tivoli Storage Productivity Center Fabric Manager metrics 291
7.7 End-to-end analysis of I/O performance problems 292
7.7.1 Performance analysis examples. 300
7.8 Tivoli Storage Productivity Center for Disk in mixed environments. 306

Part 3. Performance considerations for host systems 309

Chapter 8. Host attachment . . . 311
8.1 DS8000 host attachment . . . 312
8.2 Attaching Open Systems hosts . . . 312
8.2.1 Fibre Channel . . . 312
8.2.2 Storage area network implementations . . . 313
8.2.3 Multipathing . . . 315
8.3 Attaching IBM System z hosts . . . 320
8.3.1 FICON . . . 320
8.3.2 FICON configuration and sizing considerations . . . 322
8.3.3 z/VM, z/VSE, and Linux on System z attachment . . . 324

Chapter 9. Performance considerations for UNIX servers . . . 327
9.1 Planning and preparing UNIX servers for performance . . . 328
9.1.1 UNIX disk I/O architecture . . . 329
9.1.2 Modern approach to the Logical Volume Management in UNIX Systems . . . 331
9.1.3 Queue depth parameter (qdepth) . . . 332
9.2 AIX disk I/O components . . . 337
9.2.1 AIX Journaled File System (JFS) and Journaled File System 2 (JFS2) . . . 338
9.2.2 Symantec Veritas File System (VxFS) for AIX . . . 341
9.2.3 General Parallel File System (GPFS) . . . 342
9.2.4 IBM Logical Volume Manager . . . 343
9.2.5 Symantec Veritas Volume Manager (VxVM) . . . 347
9.2.6 IBM Subsystem Device Driver (SDD) for AIX . . . 347
9.2.7 MPIO with SDDPCM . . . 347
9.2.8 Symantec Veritas Dynamic MultiPathing (DMP) for AIX . . . 348
9.2.9 FC adapters . . . 348
9.2.10 Virtual I/O Server (VIOS) . . . 350
9.3 AIX performance monitoring tools . . . 353
9.3.1 AIX vmstat . . . 353
9.3.2 pstat . . . 355
9.3.3 AIX iostat . . . 356
9.3.4 lvmstat . . . 360
9.3.5 topas . . . 361
9.3.6 nmon . . . 362
9.3.7 fcstat . . . 365
9.3.8 filemon . . . 366
9.4 Oracle Solaris disk I/O components . . . 369
9.4.1 UFS . . . 369
9.4.2 Symantec Veritas File System (VxFS) for Solaris . . . 370
9.4.3 Oracle Solaris ZFS . . . 371
9.4.4 Solaris Volume Manager (formerly Solstice DiskSuite) . . . 373
9.4.5 Veritas Volume Manager (VxVM) for Solaris . . . 373
9.4.6 IBM Subsystem Device Driver for Oracle Solaris . . . 375
9.4.7 MPxIO . . . 375
9.4.8 Symantec Veritas Dynamic MultiPathing (DMP) for Oracle Solaris . . . 375
9.4.9 Array Support Library (ASL) . . . 375
9.4.10 FC adapter . . . 376
9.5 Oracle Solaris performance monitoring tools . . . 376
9.5.1 fcachestat and directiostat . . . 376
9.5.2 Solaris vmstat . . . 378
9.5.3 Solaris iostat . . . 378
9.5.4 vxstat . . . 380
9.5.5 dtrace . . . 380

9.6 HP-UX disk I/O architecture 382
9.6.1 HP-UX High Performance File System (HFS). 382
9.6.2 HP-UX Journaled File System (JFS). 383
9.6.3 HP Logical Volume Manager (LVM) 384
9.6.4 Veritas Volume Manager (VxVM) for HP-UX 388
9.6.5 PV Links 388
9.6.6 Native multipathing in HP-UX 388
9.6.7 Subsystem Device Driver (SDD) for HP-UX 389
9.6.8 Veritas Dynamic MultiPathing (DMP) for HP-UX 389
9.6.9 Array Support Library (ASL) for HP-UX 389
9.6.10 FC adapter 389
9.7 HP-UX performance monitoring tools 389
9.7.1 HP-UX System Activity Report (SAR) 389
9.7.2 vxstat. 392
9.7.3 HP Perfview/Measureware 392
9.8 Verifying your system 392

Chapter 10. Performance considerations for Microsoft Windows servers 397
10.1 General Windows performance tuning 398
10.2 I/O architecture overview 399
10.3 Windows Server 2008 I/O Manager enhancements 399
10.4 Filesystem. 400
10.4.1 Windows filesystem overview 400
10.4.2 NTFS guidelines 400
10.4.3 Paging file 401
10.5 Volume management 402
10.5.1 Microsoft Logical Disk Manager (LDM). 402
10.5.2 Veritas Volume Manager (VxVM) 402
10.5.3 Determining volume layout 403
10.6 Multipathing and the port layer 405
10.6.1 SCSIport scalability issues 406
10.6.2 Storport scalability features. 406
10.6.3 Subsystem Device Driver 407
10.6.4 Subsystem Device Driver Device Specific Module 407
10.6.5 Veritas Dynamic MultiPathing (DMP) for Windows. 407
10.7 Host bus adapter (HBA) settings. 408
10.8 I/O performance measurement 408
10.8.1 Key I/O performance metrics 409
10.8.2 Windows Performance console (perfmon) 411
10.8.3 Performance log configuration and data export 413
10.8.4 Collecting configuration data. 413
10.8.5 Correlating performance and configuration data. 419
10.8.6 Analyzing performance data 419

Chapter 11. Performance considerations for VMware. 425
11.1 Disk I/O architecture overview 426
11.2 Host type for the DS8000 429
11.3 Multipathing considerations. 430
11.4 Performance monitoring tools 432
11.4.1 Virtual Center performance statistics 432
11.4.2 Performance monitoring with esxtop. 433
11.4.3 Guest-based performance monitoring. 435
11.4.4 Workload spreading 435

11.4.5 Virtual machines sharing the LUN. 436
11.4.6 ESX/ESXi filesystem considerations. 437
11.4.7 Aligning partitions 439
11.5 Tuning virtual machines 443

Chapter 12. Performance considerations for Linux 445
12.1 Supported platforms and distributions. 446
12.2 Linux disk I/O architecture. 446
12.2.1 I/O subsystem architecture 447
12.2.2 Cache and locality of reference 448
12.2.3 Block layer 449
12.2.4 I/O device driver 450
12.3 Specific configuration for storage performance. 450
12.3.1 Host bus adapter for Linux 451
12.3.2 Multipathing in Linux 452
12.3.3 Logical Volume Management 454
12.3.4 Tuning the disk I/O scheduler 456
12.3.5 Filesystems 458
12.4 Linux performance monitoring tools 462
12.4.1 Gathering configuration data. 462
12.4.2 Disk I/O performance indicators 469
12.4.3 Identifying disk bottlenecks 470

Chapter 13. Performance considerations for IBM i 475
13.1 IBM i storage architecture 476
13.1.1 Single-level storage 476
13.1.2 Object-based architecture. 476
13.1.3 Storage management 476
13.1.4 Disk pools in IBM i 478
13.2 Fibre Channel adapters and Multipath 478
13.2.1 FC adapters for native connection 479
13.2.2 FC adapters in VIOS. 479
13.2.3 Multipath 480
13.3 Performance guidelines for DS8000 with IBM i. 480
13.3.1 RAID level 480
13.3.2 Number of ranks 481
13.3.3 Number and size of LUNs. 482
13.3.4 DS8000 extent pools for IBM i workloads 483
13.3.5 Disk Magic modeling for IBM i. 484
13.4 Analyzing performance data 484
13.4.1 IBM i performance tools 484
13.4.2 DS8000 performance tools 489
13.4.3 Periods and intervals of collecting data. 489
13.5 Easy Tier with IBM i 489
13.5.1 Hot data in an IBM i workload 489
13.5.2 IBM i methods for hot-spot management 490
13.5.3 Skew level of an IBM i workload 492
13.5.4 Using Easy Tier with IBM i 495
13.6 I/O Priority Manager with IBM i 496

Chapter 14. Performance considerations for System z servers 499
14.1 Overview 500
14.2 Parallel Access Volumes. 500
14.2.1 Static PAV, Dynamic PAV, and HyperPAV 501

14.2.2 HyperPAV compared to dynamic PAV test 501
14.2.3 PAV and large volumes 503
14.3 Multiple Allegiance 504
14.4 How PAV and Multiple Allegiance work 504
14.4.1 Concurrent read operation 505
14.4.2 Concurrent write operation 506
14.5 I/O Priority Queuing. 507
14.6 I/O Priority Manager 507
14.7 Logical volume sizes 508
14.7.1 Selecting the volume size 508
14.7.2 Larger volume compared to smaller volume performance 508
14.7.3 Planning the volume sizes of your configuration. 510
14.8 FICON 511
14.8.1 Extended Distance FICON 511
14.8.2 High Performance FICON. 512
14.8.3 MIDAW 518
14.9 z/OS planning and configuration guidelines 519
14.9.1 Channel configuration 519
14.9.2 Considerations for mixed workloads 520
14.10 DS8000 performance monitoring tools 521
14.11 RMF 522
14.11.1 I/O response time 522
14.11.2 I/O response time components 523
14.11.3 IOP/SAP 525
14.11.4 FICON host channel 526
14.11.5 FICON director 527
14.11.6 Processor complex 528
14.11.7 Cache and NVS 528
14.11.8 DS8000 FICON/Fibre port and host adapter. 530
14.11.9 DS8000 extent pool and rank 532

Chapter 15. IBM System Storage SAN Volume Controller attachment 533
15.1 IBM System Storage SAN Volume Controller. 534
15.1.1 SAN Volume Controller concepts 534
15.1.2 SAN Volume Controller multipathing. 537
15.1.3 SAN Volume Controller Advanced Copy Services 538
15.2 SAN Volume Controller performance considerations 539
15.3 DS8000 performance considerations with SAN Volume Controller 541
15.3.1 DS8000 array 541
15.3.2 DS8000 rank format 541
15.3.3 DS8000 extent pool implications. 542
15.3.4 DS8000 volume considerations with SAN Volume Controller. 543
15.3.5 Volume assignment to SAN Volume Controller 544
15.4 Performance monitoring 544
15.4.1 Monitoring the SAN Volume Controller with TPC for Disk 545
15.4.2 TPC Reporter for Disk. 546
15.4.3 The TPC Storage Tiering Reports and Storage Performance Optimizer 546
15.5 Sharing the DS8000 between a server and the SAN Volume Controller 547
15.5.1 Sharing the DS8000 between Open Systems servers and the IBM SAN Volume Controller 547
15.5.2 Sharing the DS8000 between System z servers and the IBM SAN Volume Controller 548
15.6 Advanced functions for the DS8000 548

15.6.1 Cache-disabled VDisks . . . 548
15.7 Configuration guidelines for optimizing performance . . . 549
15.8 Where to place the SSDs . . . 550
15.9 Where to place Easy Tier . . . 551

Chapter 16. IBM ProtecTIER deduplication . . . 553
16.1 IBM TS7600 ProtecTIER deduplication . . . 554
16.2 DS8000 attachment considerations . . . 555

Part 4. Advanced topics . . . 557

Chapter 17. Database performance . . . 559
17.1 DB2 in a z/OS environment . . . 560
17.1.1 Understanding your database workload . . . 560
17.1.2 DB2 overview . . . 561
17.1.3 DB2 storage objects . . . 561
17.1.4 DB2 dataset types . . . 562
17.2 DS8000 considerations for DB2 . . . 563
17.3 DB2 with DS8000 performance recommendations . . . 563
17.3.1 Know where your data resides . . . 563
17.3.2 Balance workload across DS8000 resources . . . 564
17.3.3 Solid-state drives . . . 564
17.3.4 Take advantage of VSAM data striping . . . 565
17.3.5 Large volumes . . . 566
17.3.6 Modified Indirect Data Address Words (MIDAWs) . . . 566
17.3.7 Adaptive Multi-stream Prefetching (AMP) . . . 566
17.3.8 DB2 burst write . . . 566
17.3.9 DB2 Adaptive List Prefetch . . . 567
17.3.10 Monitoring DS8000 performance . . . 567
17.4 DS8000 DB2 UDB in an Open Systems environment . . . 567
17.4.1 DB2 UDB storage concepts . . . 568
17.5 DB2 UDB with DS8000 performance recommendations . . . 572
17.5.1 Know where your data resides . . . 573
17.5.2 Balance workload across DS8000 resources . . . 573
17.5.3 Use DB2 to stripe across containers . . . 574
17.5.4 Selecting DB2 logical sizes . . . 574
17.5.5 Selecting the DS8000 logical disk sizes . . . 576
17.5.6 Multipathing . . . 577
17.6 IMS in a z/OS environment . . . 578
17.6.1 IMS overview . . . 578
17.6.2 IMS logging . . . 578
17.7 DS8000 considerations for IMS . . . 579
17.8 IMS with DS8000 performance recommendations . . . 580
17.8.1 Know where your data resides . . . 580
17.8.2 Balance workload across DS8000 resources . . . 580
17.8.3 Large volumes . . . 580
17.8.4 Monitoring DS8000 performance . . . 581
17.9 Oracle with DS8000 performance considerations . . . 581
17.9.1 Architecture overview . . . 581
17.9.2 DS8000 performance consideration . . . 582
17.9.3 Oracle for AIX . . . 583

Chapter 18. Copy Services performance . . . 585
18.1 Copy Services introduction . . . 586

18.2 FlashCopy . . . 587
18.2.1 FlashCopy objectives . . . 588
18.2.2 FlashCopy performance considerations . . . 589
18.2.3 Performance planning for IBM FlashCopy SE . . . 594
18.3 Metro Mirror . . . 596
18.3.1 Metro Mirror configuration considerations . . . 597
18.3.2 Metro Mirror performance considerations . . . 601
18.3.3 Scalability . . . 602
18.4 Global Copy . . . 602
18.4.1 Global Copy configuration considerations . . . 603
18.4.2 Global Copy performance considerations . . . 605
18.4.3 Scalability . . . 606
18.5 Global Mirror . . . 606
18.5.1 Global Mirror performance considerations . . . 608
18.5.2 Global Mirror session parameters . . . 611
18.5.3 Avoid unbalanced configurations . . . 614
18.5.4 Growth within Global Mirror configurations . . . 617
18.6 z/OS Global Mirror . . . 619
18.6.1 z/OS Global Mirror control dataset placement . . . 621
18.6.2 z/OS Global Mirror tuning parameters . . . 621
18.6.3 z/OS Global Mirror enhanced multiple reader . . . 622
18.6.4 zGM enhanced multiple reader performance improvement . . . 622
18.6.5 XRC Performance Monitor . . . 624
18.7 Metro/Global Mirror . . . 624
18.7.1 Metro/Global Mirror performance . . . 625
18.7.2 z/OS Metro/Global Mirror . . . 625
18.7.3 z/OS Metro/Global Mirror performance . . . 626
18.8 Considerations for Easy Tier and remote replication . . . 626

Part 5. Appendixes . . . 629

Appendix A. Performance management process . . . 631
Introduction . . . 632
Purpose . . . 633
Operational performance subprocess . . . 634
Inputs . . . 635
Outputs . . . 637
Tasks, actors, and roles . . . 637
Performance troubleshooting . . . 638
Tactical performance subprocess . . . 639
Inputs . . . 640
Outputs . . . 640
Tasks, actors, and roles . . . 640
Strategic performance subprocess . . . 641
Inputs . . . 642
Outputs . . . 642
Tasks, actors, and roles . . . 643

Appendix B. Benchmarking . . . 645
Goals of benchmarking . . . 646
Requirements for a benchmark . . . 647
Defining the benchmark architecture . . . 647
Defining the benchmark workload . . . 647
Monitoring the performance . . . 648

Defining the benchmark time frame . . . 648
Use caution when using benchmark results to design production . . . 649

Appendix C. Planning and documenting your logical configuration . . . 651
Considering hardware resources for throughput . . . 652
Performance tools . . . 656
DS8000 Query and Reporting tool (DSQTOOL) . . . 656

Appendix D. Microsoft Windows server performance log collection . . . 659
Windows Server 2003 log file configuration . . . 660
Configuring logging of disk metrics in Windows Server 2003 . . . 661
Saving counter log settings . . . 664
Importing counter log properties . . . 665
Analyzing disk performance from collected data . . . 665
Retrieving data from a counter log file . . . 665
Exporting logged data on Windows Server 2003 . . . 669
Windows Server 2008 log file configuration . . . 671
Windows Server 2008 Export . . . 675

Appendix E. Post-processing scripts . . . 677
Introduction . . . 678
Dependencies . . . 678
Sample scripts . . . 678
perfmon-essmap.pl . . . 678
iostat_aix53_essmap.pl . . . 682
iostat_sun-mpio.pl . . . 686

Index . . . 689

Related publications . . . 697
IBM Redbooks publications . . . 697
Other publications . . . 698
Online resources . . . 699
How to get IBM Redbooks publications . . . 699
Help from IBM . . . 700

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX 5L™
AIX®
AS/400®
CICS®
Cognos®
DB2 Universal Database™
DB2®
developerWorks®
DS4000®
DS6000™
DS8000®
Easy Tier®
ECKD™
Enterprise Storage Server®
ESCON®
FICON®
FlashCopy®
GDPS®
Geographically Dispersed Parallel Sysplex™
GPFS™
HACMP™
i5/OS®
IBM®
IMS™
iSeries®
Lotus®
MVS™
Netcool®
OMEGAMON®
OS/390®
Parallel Sysplex®
Power Systems™
POWER6+™
POWER6®
POWER7®
PowerHA®
PowerPC®
PowerVM®
POWER®
ProtecTIER®
Rational®
Redbooks®
Redpaper™
Redbooks (logo) ®
Resource Measurement Facility™
RMF™
Storwize®
Symphony™
Sysplex Timer®
System i®
System p®
System Storage®
System z9®
System z®
Tivoli Enterprise Console®
Tivoli®
XIV®
z/Architecture®
z/OS®
z/VM®
z/VSE®
z10™
z9®
zSeries®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication provides guidance about how to configure, monitor, and manage your IBM System Storage® DS8800 and DS8700 storage systems to achieve optimum performance. It describes the DS8800 and DS8700 performance features and characteristics, including IBM System Storage Easy Tier® and DS8000® I/O Priority Manager. It also describes how they can be used with the various server platforms that attach to the storage system. Then, in separate chapters, we detail specific performance recommendations and discussions that apply for each server environment, as well as for database and DS8000 Copy Services environments.

We also outline the various tools available for monitoring and measuring I/O performance for different server environments, as well as describe how to monitor the performance of the entire DS8000 storage system.

This book is intended for individuals who want to maximize the performance of their DS8800 and DS8700 storage systems and investigate the planning and monitoring tools that are available.

The IBM System Storage DS8800 and DS8700 storage system features, as described in this book, are available for the DS8700 with Licensed Machine Code (LMC) level 6.6.2*x.xxx* or higher and the DS8800 with Licensed Machine Code (LMC) level 7.6.2*x.xxx* or higher.

For information about optimizing performance with the previous DS8000 models, DS8100 and DS8300, see the following IBM Redbooks publication: *DS8000 Performance Monitoring and Tuning*, SG24-7146.

Performance: Any sample performance measurement data provided in this book is for comparative purposes only. Remember that the data was collected in controlled laboratory environments at a specific point in time by using the configurations, hardware, and firmware levels available at that time. Current performance in real-world environments can vary. Actual throughput or performance that any user experiences also varies depending on considerations, such as the I/O access methods in the user's job, the I/O configuration, the storage configuration, and the workload processed. The data is intended only to help illustrate how different hardware technologies behave in relation to each other. Contact your IBM representative or IBM Business Partner if you have questions about the expected performance capability of IBM products in your environment.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Gero Schmidt is an IT Specialist in the IBM Advanced Technical Support (ATS) technical sales support organization in Germany. He joined IBM in 2001 working at the European Storage Competence Center (ESCC) in Mainz, providing technical support for a broad range of IBM storage systems (ESS, DS4000®, DS5000, DS6000™, DS8000, storage area networks (SAN) Volume Controller (SVC), and XIV®) in Open Systems environments. His primary focus is on IBM Enterprise drive storage solutions, storage system performance, and IBM Power Systems™ with AIX® including PowerVM® and PowerHA®. He participated in the product rollout and major release beta test programs of the IBM System Storage

DS6000/DS8000 series. He has been a speaker on several international IBM technical conferences and co-author of several IBM Redbooks publications. He holds a degree in Physics (Dipl.-Phys.) from the Technical University of Braunschweig, Germany.

Bertrand Dufrasne is an IBM Certified Consulting I/T Specialist and Project Leader for System Storage disk products at the International Technical Support Organization, San Jose Center. He has worked at IBM in various I/T areas. He has authored many IBM Redbooks publications and has also developed and taught technical workshops. Before joining the ITSO, he worked for IBM Global Services as an Application Architect. He holds a Masters degree in Electrical Engineering.

Jana Jamsek is an IT Specialist for IBM Slovenia. She works in Storage Advanced Technical Support for Europe as a specialist for IBM Storage Systems and the IBM i (i5/OS®) operating system. Jana has eight years of experience in working with the IBM System i® platform and its predecessor models, as well as eight years of experience in working with storage. She has a Masters degree in Computer Science and a degree in Mathematics from the University of Ljubljana in Slovenia.

Peter Kimmel is an IT Specialist and ATS team lead of the Enterprise Disk Solutions team at the European Storage Competence Center (ESCC) in Mainz, Germany. He joined IBM Storage in 1999 and since then worked with all the various Enterprise Storage Server® (ESS) and System Storage DS8000 generations, with a focus on architecture and performance. He was involved in the Early Shipment Programs (ESPs) of these early installs, and co-authored several DS8000 IBM Redbooks publications. Peter holds a Diploma (MSc) degree in Physics from the University of Kaiserslautern.

Hiroaki Matsuno is an IT Specialist in IBM Japan. He has three years of experience in IBM storage system solutions working in the IBM ATS System Storage organization in Japan. His areas of expertise include DS8000 Copy Services, SAN, and Real-time Compression Appliance in Open Systems environments. He holds a Masters of Engineering from the University of Tokyo, Japan.

Flavio Morais is a GTS Storage Specialist in Brazil. He has six years of experience in the SAN/storage field. He holds a degree in Computer Engineering from Instituto de Ensino Superior de Brasilia. His areas of expertise include DS8000 Planning, Copy Services, TPC and Performance Troubleshooting. He has extensive experience solving performance problems with Open Systems.

Lindsay Oxenham is a Mainframe Storage Specialist working in Melbourne, Australia. He has over thirty years of experience in the Mainframe environment working as an application programmer and in performance and tuning areas. He joined IBM in 1998 and has worked in the storage area since 2005. He has a Bachelors degree in Applied Science (Computing). He has presented papers at SAS user meetings and Computer Management Group of Australia (CMGA) conferences.

Antonio Rainero is a Certified IT Specialist working for the Integrated Technology Services organization in IBM Italy. He joined IBM in 1998 and has more than 10 years of experience in the delivery of storage services both for z/OS® and Open Systems clients. His areas of expertise include storage subsystems implementation, performance analysis, SANs, storage virtualization, disaster recovery, and high availability solutions. Antonio holds a degree in Computer Science from University of Udine, Italy.

Denis Senin is an IT Specialist in IBM Russia. He has 10 years of experience in IT industry and has worked at IBM for six years. Denis holds a Masters degree as a design-engineer of computer systems from The Moscow State Institute of Radiotechnics, Electronics and Automatics and has a real background of systems design and development. His current areas of expertise include Open Systems high-performing and disaster recovery storage solutions.

Figure 1 The team: Hiroaki, Antonio, Flavio, Peter, Bertrand, Denis, Jana, Gero, and Lindsay

Thanks to the following people for their contributions to this project:

Andreas Bär, Werner Bauer, Lawrence Chiu, Nick Clayton, Thomas Edgerton, Lee La Frese, Mark Funk, Kirill Gudkov, Hans-Paul Drumm, Rob Jackard, Chip Jarvis, Frank Krüger, Yang Liu (Loren), Michael Lopez, Joshua Martin, Dennis Ng, Markus Oscheka, Brian Sherman, Rick Ripberger, Louise Schillig, Günter Schmitt, Uwe Schweikhard, Jim Sedgwick, Cheng-Chung Song, Markus Standau, Paulus Usong, Alexander Warmuth, Sonny Williams, and Yan Xu

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Part 1

DS8000 performance considerations

This part gives an overview of the IBM System Storage DS8000 characteristics and logical configuration concepts to achieve optimum performance.

This part includes the following topics:

- DS8000 characteristics
- Hardware configuration
- Logical configuration concepts and terminology
- Logical configuration performance considerations

1

DS8000 characteristics

This chapter contains a high-level discussion and introduction to the storage server performance challenge. Then, we provide an overview of the DS8000 model characteristics that allow the IBM System Storage DS8000 series to meet this performance challenge. Whenever we use DS8000, we include both the DS8800 models and also the DS8700 models, which share a common microcode base with DS8800.

1.1 The storage server challenge

One of the primary criteria in judging a storage server is performance: how fast it responds to a read or write request from an application server. How well a storage server accomplishes this task depends on the design of its hardware and firmware.

Data continually moves from one component to another component within a storage server. The objective of server design is hardware of sufficient throughput to keep data flowing smoothly without waiting because a component is busy. When data stops flowing because a component is busy, a bottleneck forms. Obviously, it is desirable to minimize the frequency and severity of bottlenecks.

The ideal storage server is one in which all components are used and bottlenecks are few. This scenario is the case if the following conditions are met:

- The machine is designed well, with all hardware components in balance. To provide this balance over a range of workloads, a storage server must allow a range of hardware component options.
- The machine is sized well for the client workload. That is, where options exist, the right quantities of each option are chosen.
- The machine is set up well. That is, where options exist in hardware installation and logical configuration, these options are chosen correctly.

Automatic rebalancing and tiering options can help achieve optimum performance even in an environment of ever changing workload patterns. But they cannot replace a correct sizing of the machine.

1.1.1 Performance numbers

Raw performance numbers provide evidence that a particular storage server is better than the previous generation model, or better than the competitive product. But isolated performance numbers are often out of line with a production environment. It is important to understand how raw performance numbers relate to the performance of the storage server in processing a particular production workload.

Throughput numbers are achieved in controlled tests that push as much data as possible through the storage server as a whole, or perhaps through a single component. At the point of maximum throughput, the system is so overloaded that response times are greatly extended. Trying to achieve such throughput numbers in a normal business environment brings protests from the users of the system, because response times are poor.

To assure yourself that the DS8800 offers the latest and fastest technology, look at the performance numbers for the individual disks, adapters, and other components of the DS8800, as well as for the total device. The DS8800 uses the most current technology available. But, use a more rigorous approach when planning the DS8800 hardware configuration to meet the requirements of a specific environment.

1.1.2 Recommendations and rules

Hardware selections are sometimes based on general recommendations and *rules*. A general rule is a simple guideline for making a selection based on limited information. You can make a quick decision, with little effort, that provides a solution that works most of the time. The disadvantage is that it does not work all the time; sometimes, the solution is not at all what the client needs. You can increase the chances that the solution works by making it more conservative. However, a conservative solution generally involves more hardware, which means a more expensive solution. In this chapter, we provide recommendations and general rules for different hardware components. Remember, use general rules when no information is available to make a more informed decision. And in general, when options are available for automated tiering and rebalancing, in the most typical cases, use these options.

1.1.3 Modeling your workload

A much better way to determine the hardware requirements for your workload is to run a Disk Magic model. Disk Magic is a modeling tool, which shows the throughput and response time of a storage server based on workload characteristics and the hardware resources of the storage server. By converting the results of performance runs into mathematical formulas, Disk Magic allows the results to be applied to a wide range of workloads. Disk Magic allows many variables of hardware to be brought together. This way, the effect of each variable is integrated, producing a result that shows the overall performance of the storage server.

For additional information about this tool, see 6.1, “Disk Magic” on page 176.

1.1.4 Allocating hardware components to workloads

Two contrasting methods are available to allocate the use of hardware components to workloads.

The first method is *spreading* the workloads across components, which means that you try to share the use of hardware components across all, or at least many, workloads. The more hardware components are shared among multiple workloads, the more effectively the hardware components are used, which reduces total cost of ownership (TCO). For example, to attach multiple hosts, you can use the same host adapters for all hosts instead of acquiring a separate set of host adapters for each host. However, the more that components are shared, the more potential exists that one workload dominates the use of the component.

The second method is *isolating* workloads to specific hardware components, which means that specific hardware components are used for one workload, and other hardware components are used for different workloads. The downside of isolating workloads is that certain components are unused when their workload is not demanding service. On the upside, when that workload demands service, the component is available immediately. The workload does not contend with other workloads for that resource.

Spreading the workload maximizes the usage and performance of the storage server as a whole. Isolating a workload is a way to maximize the workload performance, making the workload run as fast as possible. *Automatic I/O prioritization* can help avoid a situation in which less-important workloads dominate the mission-critical workloads in shared environments, and allow more shared environments.

For a detailed discussion, see 4.2, “Configuration principles for optimal performance” on page 90.

1.2 Meeting the challenge: DS8000

The DS8800 and its predecessor models are members of the DS product family. They offer disk storage servers with a wide range of hardware component options to fit many workload requirements, in both type and size. The DS8800 can scale well to the highest disk storage capacities. The scalability is supported by design functions that allow the installation of additional components without disruption.

The IBM System Storage DS8800 performs so that multiple workloads can be easily consolidated into a single storage subsystem.

1.2.1 DS8000 models and characteristics

The System Storage DS8800, which is the fourth generation of the DS8000, currently offers three models. The models are a 4-way model, 2-way model, and 2-way model with “Business Class” cabling, which allows more disks with a few device adapter (DA) pairs. The predecessor system DS8700 is available in a 4-way version and a 2-way version.

Before these models, the first DS8000 generation offered the DS8100 (2-way) and DS8300 (4-way). The second generation offered the DS8100 Turbo and DS8300 Turbo models with faster processors. The DS8300 and DS8100, however, do not share the current code level options that exist for DS8800 and therefore are not covered in this book.

The DS8800 base frame houses the processor complexes, including system memory, up to eight host adapters, and up to 240 disk modules. The first expansion frame houses up to eight additional host adapters (for a total of 16) and up to 336 additional disk modules (for a total of 576). A second or third DS8800 expansion frame houses up to 480 disk modules (for a total of 1056 or 1536). There are no additional host adapters installed for the second or third expansion frames. The maximum disk numbers for DS8800 are valid when using the 2.5 inch small form factor (SFF) disks with 15K or 10K rpm. When using 3.5 inch nearline (7,200 rpm) disks, the maximum numbers for these disks are half, due to their dimensions.

Table 1-1 provides an overview of the DS8000 models, including processor, memory, host adapter, and disk specifications for each model. Each of the models comes with one Storage Facility Image (SFI). All DS8800 Standard models can be upgraded nondisruptively from the smallest 2-way model up to the largest multi-frame 4-way model.

Table 1-1 DS8000 processor models overview

	DS8800 4-way Standard	**DS8800 2-way Standard**	**DS8800 2-way Business Class**	**DS8700 4-way**	**DS8700 2-way**
Base frame model	951	951	951	941	941
Expansion frame model	95E			94E	
Number of processor complexes	2	2	2	2	2
Number of processors per complex	4	2	2	4	2
Processor speed	5.0 GHz	5.0 GHz	5.0 GHz	4.7 GHz	4.7 GHz
Processor memory options (cache)	32 - 384 GB	32 - 128 GB	16 - 128 GB	32 - 384 GB	32 - 128 GB
Host adapters, minimum/maximum	2 - 16	2 - 4	2 - 4	2 - 32 (4 Gbps) 2 - 16 (8 Gbps)	2 - 16 (4 Gbps) 2 - 8 (8 Gbps)
Ports per Fibre Channel Protocol (FCP)/ Fibre Channel connection (FICON®) host adapter	4/8	4/8	4/8	4	4
Disk drive modules (DDMs), maximum	1536	144	240	1024	128

	DS8800 4-way Standard	DS8800 2-way Standard	DS8800 2-way Business Class	DS8700 4-way	DS8700 2-way
Disk drive interface technology	SAS 6 Gbps	SAS 6 Gbps	SAS 6 Gbps	FC 4 Gbps	FC 4 Gbps

Next, we provide a short description of the main hardware components.

POWER6+ processor technology

The DS8000 series uses the IBM POWER6+™ technology, which is the foundation of the storage system logical partitions (LPARs). The DS8800 (951) 2-way and 4-way models use the dual 2-way and 4-way processor complexes of the 64-bit microprocessors. Within the POWER6+ servers, the DS8000 series offers up to 384 GB of cache.

Internal communication

The DS8000 comes with a high-bandwidth, fault-tolerant internal interconnection, which is also used in the IBM Power Systems servers. It is called RIO-G (Remote I/O) and used for the cross-cluster communication.

Switched PCIe disk connection

The disk interconnection changed between DS8800/DS8700 in comparison to the first DS8000 generations, such as the DS8300. The DS8300 used a switched Fibre Channel Arbitrated Loop (FC-AL) implementation based on PCI Extended (PCI-X) connections. The DS8800 architecture connects the I/O bays through PCI Express (PCIe).

Disk drives

The DS8800 offers a selection of industry standard Serial Attached SCSI (SAS 2.0) disk drives. Most drive types (15,000 and 10,000 rpm) are 0.76 m (2.5 inch) small form factor (SFF) sizes, with drive capacities of 146 GB - 900 GB. The solid-state drives (SSDs) are 2.5 inch SAS 2.0. The nearline drives (7,200 rpm) are 3.5 inch size drives with an SAS interface. With the current maximum number and type of drives, the storage system can scale to over 2 petabytes (PB) of total gross capacity. All drive types of the previous DS8000 generations, such as DS8700, use 3.5 inch Fibre Channel (FC) disk drives.

Host adapters

The DS8800 offers enhanced connectivity with the availability of either eight-port or four-port FC/FICON host adapters (DS8700: four ports per host bus adapter (HBA) only). The 8 Gb/s Fibre Channel/FICON host adapters, which are offered in longwave and shortwave versions, can also auto-negotiate to 4 Gb/s or 2 Gb/s link speeds. With this flexibility, you can benefit from the higher performance, 8 Gb/s SAN-based solutions and also maintain compatibility with existing infrastructures. In addition, you can configure the ports on the adapter with an intermix of FCP and FICON, which can help protect your investment in Fibre adapters and increase your ability to migrate to new servers. A DS8800 can support up to a maximum of 16 host adapters, which provide up to 128 FC/FICON ports.

1.3 DS8000 performance characteristics overview

The IBM System Storage DS8000 series offers optimally balanced performance. For instance, the DS8800 offers up to three times the maximum sequential throughput of a DS8300. A DS8800 4-port, 8 Gbps HBA offers a sequential HBA throughput that is up to more than five times higher than a DS8300 4-port, 4 Gbps PCI-X HBA. These throughputs are possible, because the DS8800 incorporates many performance enhancements:

- Dual-clustered POWER6+ servers
- PCIe 8 Gbps FC/FICON host adapters
- New SAS disk drives
- High-bandwidth, fault-tolerant internal interconnections with an 8-Gbps switched PCIe architecture throughout the machine

With all these new components, the DS8800 is positioned at the top of the high-performance category. The following hardware components contribute to the high performance of the DS8000:

- Redundant Array of Independent Disks (RAID)
- *Array across loops* (AAL)
- POWER6® symmetrical multiprocessor system (SMP) processor architecture
- Switched PCIe 8-Gbps architecture.

In addition to these hardware-based enhancements, we describe even more contributions to performance in the following sections.

1.3.1 Advanced caching techniques

The DS8000 benefits from advanced caching techniques, most of which are unique to IBM storage systems.

Sequential Prefetching in Adaptive Replacement Cache (SARC)

Another performance enhancer is the use of the self-learning cache algorithms. The DS8000 series caching technology improves cache efficiency and enhances cache-hit ratios, especially in environments that change dynamically. One of the patent-pending algorithms that is used in the DS8000 series is called *Sequential Prefetching in Adaptive Replacement Cache (SARC)*.

SARC provides these advantages:

- Sophisticated, patented algorithms to determine what data to store in cache based on the recent access and frequency needs of the hosts
- *Prefetching*, which anticipates data before a host request and loads it into cache
- Self-learning algorithms to adapt and dynamically learn what data to store in cache based on the frequency needs of the hosts

Adaptive Multi-Stream Prefetching (AMP)

AMP introduces an autonomic, workload-responsive, self-optimizing prefetching technology. This technology adapts both the amount and the timing of prefetch on a per-application basis to maximize the performance of the system.

AMP provides a provable, optimal sequential read performance and maximizes the sequential read throughputs of all RAID arrays where it is used, and therefore of the system.

Intelligent Write Caching (IWC)

Another additional cache algorithm, *Intelligent Write Caching* (IWC), is implemented in the DS8000 series. IWC improves performance through better write-cache management and a better destaging order of writes.

By carefully selecting the data that we destage and, at the same time, reordering the destaged data, we minimize the disk head movements that are involved in the destage processes. Therefore, we achieve a large performance boost on random-write workloads.

1.3.2 IBM System Storage multipath Subsystem Device Driver (SDD)

SDD is a pseudo device driver on the host system that is designed to support the multipath configuration environments in IBM products. It provides load balancing and enhanced data availability capability. By distributing the I/O workload over multiple active paths, SDD provides dynamic load balancing and eliminates data flow bottlenecks. SDD also helps eliminate a potential single point of failure by automatically rerouting I/O operations when a path failure occurs.

SDD is provided with the DS8000 series at no additional charge. Fibre Channel (SCSI-FCP) attachment configurations are supported in the AIX, Hewlett-Packard UNIX (HP-UX), Linux, Microsoft Windows, and Oracle Solaris environments.

In addition, the DS8000 series supports the built-in multipath options of many distributed operating systems.

1.3.3 Performance characteristics for System z

The DS8000 series supports the following IBM performance innovations for System z® environments:

- *FICON* extends the ability of the DS8000 series system to deliver high-bandwidth potential to the logical volumes that need it, when they need it. Working together with other DS8000 series functions, FICON provides a high-speed pipe that supports a multiplexed operation.
- *High Performance FICON for z (zHPF)* takes advantage of the hardware available today, with enhancements that are designed to reduce the overhead associated with supported commands. These enhancements can improve FICON I/O throughput on a single DS8000 port by 100%. Enhancements to the z/Architecture® and the FICON interface architecture deliver improvements for online transaction processing (OLTP) workloads. When used by the FICON channel, the z/OS operating system, and the control unit, zHPF is designed to help reduce overhead and improve performance.
- *Parallel Access Volume (PAV)* enables a single System z server to simultaneously process multiple I/O operations to the same logical volume, which can help to reduce device queue delays. This function is achieved by defining multiple addresses per volume. With Dynamic PAV, the assignment of addresses to volumes can be managed automatically to help the workload meet its performance objectives and reduce overall queuing. PAV is an optional feature on the DS8000 series.
- *HyperPAV* allows an alias address to be used to access any base on the same control unit image per I/O base. Multiple HyperPAV hosts can use one alias to access separate bases. This capability reduces the number of alias addresses required to support a set of bases in a System z environment with no latency in targeting an alias to a base. This functionality is also designed to enable applications to achieve equal or better performance than possible with the original PAV feature alone while also using the same or fewer z/OS resources.
- *Multiple Allegiance* expands the simultaneous logical volume access capability across multiple System z servers. This function, along with PAV, enables the DS8000 series to process more I/Os in parallel, helping to improve performance and enabling greater use of large volumes.

- *Application and host integration*. With the conversion of DB2® I/Os to zHPF, starting with R6.2 (Licensed Machine Code (LMC) 7.6.20), massive performance gains for certain database tasks can be achieved. For instance, the DB2 application can give cache prefetching hints down into the storage system.
- *I/O priority queuing* allows the DS8000 series to use I/O priority information provided by the z/OS Workload Manager to manage the processing sequence of I/O operations.
- *I/O Priority Manager integration with zWLM.* In 1.3.5, "I/O Priority Manager" on page 17, we explain how the DS8000 I/O Priority Manager feature offers additional add-ons for System z. With these add-ons, the z/OS host can manage and reassign a count key data (CKD) volume automatically to another DS8000 performance class. This capability is useful when the priorities of the CKD volume do not match the priorities preset in the System z Workload Manager.

1.3.4 Easy Tier

IBM System Storage Easy Tier is an optimizing and balancing feature at no charge that is available for the System Storage DS8800 and DS8700 models. It offers enhanced capabilities through the following functionalities:

- Volume rebalancing
- Auto-rebalancing
- Hot spot management
- Rank depopulation
- Manual volume migration
- Thin provisioning support

Easy Tier determines the appropriate tier of storage based on data access requirements. It then automatically and nondisruptively moves data, at the subvolume or sublogical unit number (LUN) level, to the appropriate tier on the DS8000.

IBM announced Easy Tier with R5.1 (LMC 6.5.1) for the DS8700. IBM enhanced Easy Tier with the R6.1 (LMC 7.6.10/6.6.10) and R6.2 (LMC 7.6.20/6.6.20) microcode versions that are currently available for DS8800 and DS8700. Starting with DS8000 R6.2 LMC, Easy Tier Automatic Mode provides automatic inter-tier or cross-tier storage performance and storage economics management for up to three tiers. It also provides automatic intra-tier performance management (auto-rebalance) in multi-tier (hybrid) or single-tier (homogeneous) extent pools.

IBM System Storage Easy Tier is designed to balance system resources to address application performance objectives. It automates data placement across Enterprise-class (Tier 1), nearline (Tier 2), and SSD (Tier 0) tiers, as well as among the ranks of the same tier (auto-rebalance). The system can automatically and non-disruptively relocates data (at the extent level) across any two or three tiers of storage and manually relocates full volumes. The potential benefit is to align the performance of the system with the appropriate application workloads. This enhancement can help clients to improve storage usage and address performance requirements for multi-tier systems that do not yet deploy SSDs. For those clients that already use some percentage of SSDs, Easy Tier analyzes the system and migrates only those parts of the volumes, whose workload patterns benefit most from the valuable SSD space, to the higher tier.

Use the capabilities of Easy Tier to support these functions:

- Three tiers: Use three tiers and enhanced algorithms to improve system performance and cost-effectiveness.
- Cold demotion: Cold data (or extents) stored on a higher-performance tier is demoted to a more appropriate tier. Expanded cold demotion demotes part of the sequential workload to the Nearline tier to better use the bandwidth in the Nearline tier.

- Warm demotion: To avoid higher-performance tiers from becoming overloaded in hybrid extent pools, the Easy Tier automatic mode monitors the performance of the ranks. It triggers the movement of selected active extents from the higher-performance tier to the lower-performance tier. This warm demotion helps to ensure that the higher-performance tier does not suffer from saturation or overload conditions that might affect the overall performance in the extent pool.
- Manual volume rebalance: Volume rebalancing relocates the smallest number of extents of a volume and restripes those extents on all available ranks of the extent pool.
- Auto-rebalance: This capability automatically balances the workload across ranks of the same storage tier within homogeneous and hybrid pools based on usage to improve system performance and resource utilization. It enables the enhanced auto-rebalancing functions of Easy Tier to manage a combination of homogeneous and hybrid pools, including relocating hot spots on ranks. Even with homogeneous pools, systems with only one tier can use Easy Tier technology to optimize their RAID array utilization.
- Rank depopulation: This capability allows ranks with extents (data) allocated to them to be unassigned from an extent pool. It uses extent migration to move extents from the specified ranks to other ranks within the pool.

Easy Tier also provides a performance monitoring capability, regardless of whether the Easy Tier license feature is activated. Easy Tier uses the monitoring process to determine what data to move and when to move it when using automatic mode. The usage of thin provisioning (extent space-efficient (ESE) volumes for Open Systems) is also possible with Easy Tier.

You can enable monitoring independently (with or without the Easy Tier license feature activated) for information about the behavior and benefits that can be expected if automatic mode is enabled. Data from the monitoring process is included in a summary report that you can download to your Microsoft Windows system. Use the IBM System Storage DS8000 Storage Tier Advisor Tool (STAT) application to view the data when you point your browser to that file.

For downloading the monitoring tool STAT, follow the web link:

http://www.ibm.com/support/docview.wss?uid=ssg1S4000982

IBM Easy Tier is described in detail in *IBM System Storage DS8000 Easy Tier*, REDP-4667.

Prerequisites

The following conditions must be met to enable Easy Tier:

- The Easy Tier license feature is enabled (required for both manual and automatic mode, except when monitoring is set to All Volumes).
- For automatic mode to be active, the following conditions must be met:
  - Easy Tier automatic monitoring is set to either All or Auto Mode.
  - For Easy Tier to manage pools, the Easy Tier Auto Mode setting must be set to either *Tiered Pools* or *All Pools* (in the Storage Image Properties panel in the DS GUI).

Easy Tier: Automatic mode

Use of the automatic mode of Easy Tier requires the no-charge Easy Tier license feature.

In Easy Tier, both I/O per second (IOPS) and bandwidth algorithms determine when to migrate your data. This process can help you improve performance.

Use the automatic mode of Easy Tier to relocate your extents to their most appropriate storage tier in a hybrid pool, based on usage. Because workloads typically concentrate I/O operations (data access) on only a subset of the extents within a volume or LUN, automatic mode identifies the subset of your frequently accessed extents. It then relocates them to the higher-performance storage tier.

Subvolume or sub-LUN data movement is an important option to consider in volume movement because not all data at the volume or LUN level becomes hot data. For any workload, there is a distribution of data considered either hot or cold. Significant overhead can be associated with moving entire volumes between tiers. For example, if a volume is 1 TB, you do not want to move the entire 1 TB volume if the heat map generated indicates that only 10 GB is considered hot. This capability uses your higher performance tiers while reducing the number of drives that you need to optimize performance.

Using automatic mode, you can use high-performance storage tiers with a smaller cost. Therefore, you invest a small portion of storage in the high-performance storage tier. You can use automatic mode for relocation and tuning without intervention. Automatic mode can help generate cost-savings while optimizing your storage performance.

Three-tier automatic mode is supported with the following Easy Tier functions:

- Support for ESE volumes with the thin provisioning of your FB volumes
- Support for a matrix of device or disk drive modules (DDMs) and adapter types
- Enhanced monitoring of both bandwidth and IOPS limitations
- Enhanced data demotion between tiers
- Automatic mode auto-performance rebalancing (auto-rebalance), which applies to the following situations:
  - Redistribution within a tier after a new rank is added into a managed pool (adding new capacity)
  - Redistribution within a tier after extent pools are merged
  - Redistribution within a tier after a rank is removed from a managed pool
  - Redistribution when the workload is imbalanced on the ranks within a tier of a managed pool (natural skew)

To help manage and improve performance, Easy Tier is designed to identify hot data at the subvolume or sub-LUN (extent) level, based on ongoing performance monitoring. Then, it automatically relocates that data to an appropriate storage device in an extent pool that is managed by Easy Tier. Easy Tier uses an algorithm to assign heat values to each extent in a storage device. These heat values determine which tier is best for the data, and migration takes place automatically. Data movement is dynamic and transparent to the host server and to applications that use the data.

By default, automatic mode is enabled (through the DSCLI and DS Storage Manager) for heterogeneous pools, when the Easy Tier license feature is activated. You can temporarily disable automatic mode.

Functions and features of Easy Tier: Automatic mode

This section describes the functions and features of Easy Tier in automatic mode.

Auto-rebalance

Auto-rebalance is a function of Easy Tier automatic mode to balance the utilization within a tier by relocating extents across ranks based on usage. Auto-rebalance is now enhanced to support single-tier managed pools, as well as multi-tier hybrid pools. You can use the Storage

Facility Image (SFI) control to enable or disable the auto-rebalance function on all pools of an SFI. When you enable auto-rebalance, every standard and ESE volume is placed under Easy Tier control and auto-rebalancing management. Using auto-rebalance gives you the advantage of these automatic functions:

- Easy Tier auto-rebalance operates within a tier, inside a managed single-tier or multi-tier storage pool, relocating extents across ranks of the same tier based on utilization and workload.
- Easy Tier auto-rebalance automatically detects performance skew and rebalances utilization across ranks within the same tier.

In any tier, placing highly active (hot) data on the same physical rank can cause the hot rank or the associated DA to become a performance bottleneck. Likewise, over time, skew can appear within a single tier that cannot be addressed by migrating data to a faster tier alone. It requires some degree of workload rebalancing within the same tier. Auto-rebalance addresses these issues within a tier in both hybrid (multi-tier) and homogeneous (single-tier) pools. It also helps the system to respond in a more timely and appropriate manner to overload situations, skew, and any under-utilization. These conditions can occur for the following reasons:

- Addition or removal of hardware
- Migration of extents between tiers
- Merger of extent pools
- Changes in the underlying volume configuration
- Variations in the workload

Auto-rebalance adjusts the system to continuously provide optimal performance by balancing the load on the ranks and on DA pairs.

The latest version of Easy Tier provides support for auto-rebalancing even within homogeneous, single-tier extent pools. If you set the Easy Tier Auto Mode control to manage *All Pools*, Easy Tier also manages homogeneous extent pools with only a single tier and performs intra-tier performance rebalancing. If Easy Tier is turned off, no volumes are managed. If Easy Tier is turned on, it manages all supported volumes, either standard or ESE volumes. Track space-efficient (TSE) volumes are not supported with Easy Tier and auto-rebalancing.

Warm demotion

To avoid overloading higher-performance tiers in hybrid extent pools, and thus potentially degrading overall pool performance, Easy Tier Automatic Mode monitors performance of the ranks. It can trigger the move of selected extents from the higher-performance tier to the lower-performance tier based on predefined either bandwidth or IOPS overload thresholds. The Nearline tier drives perform almost as well as SSDs and Enterprise hard disk drives (HDDs) for sequential (high bandwidth) operations.

This automatic operation is rank-based, and the target rank is randomly selected from the lower tier. Warm demote is the highest priority to quickly relieve overloaded ranks. So, Easy Tier continuously ensures that the higher-performance tier does not suffer from saturation or overload conditions that might affect the overall performance in the extent pool. Auto-rebalancing movement takes place within the same tier. Warm demotion takes place across more than one tier. Auto-rebalance can be initiated when the rank configuration changes the workload that is not balanced across ranks of the same tier. Warm demotion is initiated when an overloaded rank is detected.

Cold demotion

Cold demotion recognizes and demotes cold or semi-cold extents to the lower tier. Cold demotion occurs between HDD tiers, that is, between Enterprise and nearline drives. Cold extents are demoted in a storage pool to a lower tier, as long as that storage pool is not idle.

Cold demotion occurs when Easy Tier detects any of the following scenarios:

- Segments in a storage pool become inactive over time, while other data remains active. This scenario is the most typical use for cold demotion, where inactive data is demoted to the Nearline tier. This action frees up segments on the Enterprise tier before the segments on the Nearline tier become hot, which helps the system to be more responsive to new, hot data.
- In addition to cold demote, which uses the capacity in the lowest tier, segments with moderate bandwidth but low random IOPS requirements are selected for demotion to the lower tier in an active storage pool. This demotion better utilizes the bandwidth in the Nearline tier (expanded cold demote).

If all the segments in a storage pool become inactive simultaneously due to either a planned or an unplanned outage, cold demotion is disabled. Disabling cold demotion assists the user in scheduling extended outages or when experiencing outages without changing the data placement.

Figure 1-1 illustrates all of the migration types supported by the latest Easy Tier enhancements in a three-tier configuration. The auto-rebalance might also include additional swap operations.

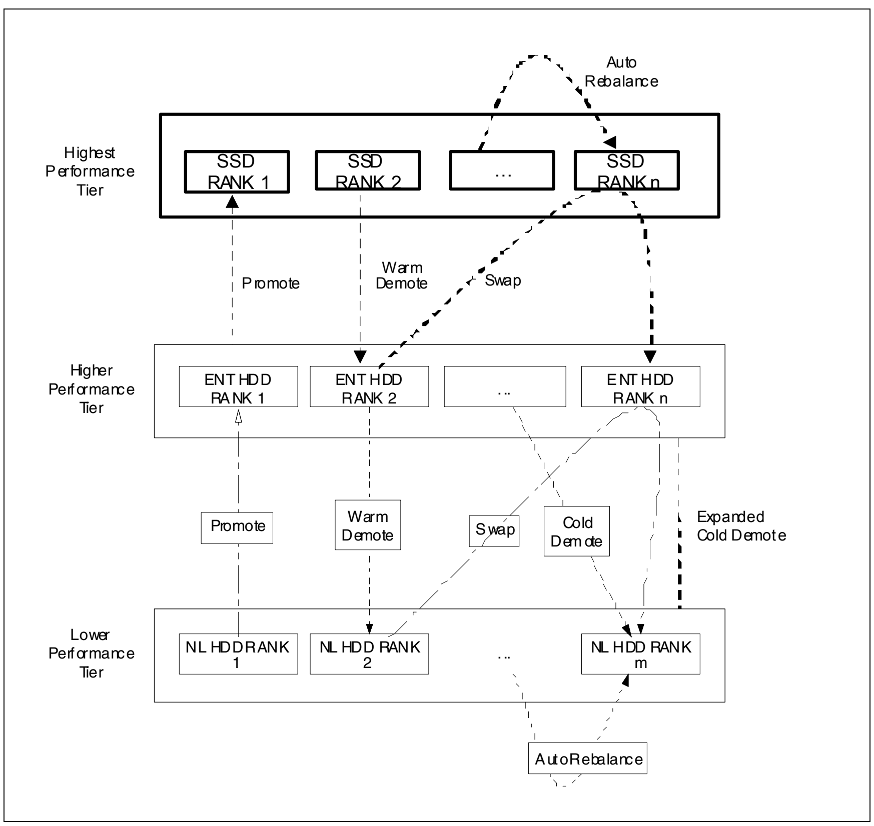


Figure 1-1 Features of Easy Tier automatic mode

Easy Tier: Manual mode

Easy Tier in manual mode provides the capability to migrate volumes and merge extent pools, under the same DS8000 system, concurrently with I/O operations.

In Easy Tier manual mode, you can dynamically relocate a logical volume between extent pools or within an extent pool to change the extent allocation method of the volume or to redistribute the volume across new ranks. This capability is referred to as *dynamic volume relocation*. You can also merge two existing pools into one pool without affecting the data on the logical volumes associated with the extent pools. In an older installation with many pools, you can introduce the automatic mode of Easy Tier with automatic inter-tier and intra-tier storage performance and storage economics management in multi-rank extent pools with one or more tiers. Easy Tier manual mode also provides a rank depopulation option to remove a rank from an extent pool with all the allocated extents on this rank automatically moved to the other ranks in the pool.

The enhanced functions of Easy Tier manual mode provide additional capabilities. You can use manual mode to relocate entire volumes from one pool to another pool. Upgrading to a new disk drive technology, rearranging the storage space, or changing storage distribution for a workload are typical operations that you can perform with volume relocations.

You can more easily manage configurations that deploy separate extent pools with different storage tiers or performance characteristics. The storage administrator can easily and dynamically move volumes to the appropriate extent pool. Therefore, the storage administrator can meet storage performance or storage economics requirements for these volumes transparently to the host and the application. Use manual mode to achieve these operations and increase the options to manage your storage.

Functions and features of Easy Tier: Manual mode

This section describes the functions and features of Easy Tier manual mode.

Volume migration

You can select which logical volumes to migrate, based on performance considerations or storage management concerns:

- Migrate volumes from one extent pool to another. You might want to migrate volumes to a different extent pool with more suitable performance characteristics, such as different disk drives or RAID ranks. For example, a volume that was configured to stripe data across a single RAID can be changed to stripe data across multiple arrays for better performance. Also, as different RAID configurations or drive technologies become available, you might want to move a logical volume to a different extent pool with different characteristics. You might also want to redistribute the available disk capacity between extent pools.
- Change the extent allocation method that is assigned to a volume. You can relocate a volume within the same extent pool but with a different extent allocation method (EAM). For example, you might want to change the extent allocation method to help spread I/O activity more evenly across ranks. If you configured logical volumes in an extent pool with fewer ranks than now exist in the extent pool, you can use Easy Tier volume migration to manually redistribute the volumes across new ranks using *manual volume rebalance*. If you specify a different extent allocation method for a volume, the new extent allocation method is effective immediately.

The overhead that is associated with volume migration is comparable to a FlashCopy® operation that runs in a background copy.

Manual volume rebalance

Manual volume rebalance is designed to redistribute the extents of volumes within a non-managed, single-tier (homogeneous) pool so that workload skew and hot spots are less likely to occur on the ranks. Manual volume rebalance is an enhancement of the dynamic volume relocation feature introduced with the DS8000 R6.1 LMC. With this feature, a volume is migrated back into the same extent pool, and the source extent pool becomes the target extent pool. The extent allocation method (EAM) is the rotate extents method, which uses storage pool striping. In this case, the algorithm attempts to evenly spread the volume extents across all ranks in the extent pool. This feature is useful for redistributing extents after adding new ranks to an existing non-managed extent pool or after merging homogeneous extent pools. You can use this feature to balance the capacity and thus the workload of the volumes across all available ranks in a pool.

Manual volume rebalance provides manual performance optimization by rebalancing the capacity of a volume within a non-managed homogeneous extent pool. It can also be referred to as *capacity rebalance.* It balances the extents of the volume across the ranks within an extent pool without actually taking any workload statistics or device utilizations into account (as otherwise done by the auto-rebalance feature). A balanced distribution of the extents of a volume across all available ranks in a pool is supposed to provide a balanced workload distribution and minimize skew and hot spots. It is also referred to as *volume restriping* and supported for standard and ESE thin-provisioned volumes.

During extent relocation with manual volume rebalance, only one extent at a time is allocated rather than to pre-allocate the full volume. Only a minimum amount of free capacity is required in the extent pool, and only the required number of extents is relocated.

This option is preferred wherever the auto-rebalance option can be enabled for automatic rebalancing of the workload across ranks of the same storage tier It relocates the extents based on their actual workload pattern and provides an ongoing performance optimization even when workload patterns change over time, or additional ranks are added. Manual volume rebalance redistributes the volume capacity without considering any performance characteristics or natural performance skew.

The manual volume rebalance is revoked in any managed or hybrid extent pools. It does not matter whether the hybrid pool is currently managed or non-managed. Hybrid pools are always assumed to be created for Easy Tier automatic mode management. Manual volume rebalance is only available in non-managed homogeneous (single-tier) extent pools.

Volume rebalance can be achieved by initiating a manual volume migration into the same pool by using rotate extents as the EAM. Use volume migration to manually rebalance the volume capacity when a rank is added to a pool, when homogeneous extent pools are merged, or when large volumes with an EAM of *rotate volumes* are deleted. Manual volume rebalance is also referred to as capacity rebalance, because it balances the distribution of extents without factoring in the actual extent usage or workload.

Use volume rebalance to relocate the smallest number of extents of a volume and restripe the extents of that volume on all available ranks of the pool where it is located.

Dynamic extent pool merge

The dynamic extent pool merge capability allows a user to initiate a merging process of one extent pool (source extent pool) into another extent pool (target extent pool). During this process, all the volumes in both source and target extent pools remain accessible to the hosts. For example, you can manually combine two existing extent pools with a homogeneous or a hybrid disk configuration into a single extent pool with SSDs to use Easy Tier automatic mode. When the merged hybrid pools contain SSD-class, Enterprise-class, and Nearline-class tiers, they all can be managed by Easy Tier automatic mode as a three-tier storage hierarchy.

Easy Tier monitoring

The IBM Storage Tier Advisory Tool (STAT) collects and reports volume performance monitoring data. It provides performance monitoring data even if the license feature is not activated. The monitoring capability of the DS8000 enables it to monitor the usage of storage at the volume extent level. Monitoring statistics are gathered and analyzed every 24 hours. In an Easy Tier managed extent pool, the analysis is used to form an extent relocation plan for the extent pool. This plan provides a recommendation, based on your current plan, for relocating extents on a volume to the most appropriate storage device. The results of this data are summarized in a report that is available for download.

Figure 1-2 describes the Easy Tier monitor settings in conjunction with the installation of the Easy Tier LIC feature.

	Easy Tier license feature	
Monitor Setting	**Not installed**	**Installed**
All Volumes	All volumes are monitored	All volumes are monitored
Auto-Mode Volumes	No volumes are monitored	Volumes in extent pools managed by Easy Tier are monitored
No Volumes	No volumes are monitored	No volumes are monitored

Figure 1-2 DSCLI and DS Storage Manager settings for Easy Tier monitoring

IBM System Storage DS8000 Storage Tier Advisor Tool

The DS8000 offers a reporting tool called IBM System Storage DS8000 Storage Tier Advisor Tool (STAT). The Storage Tier Advisor Tool is a Microsoft Windows application that provides a graphical representation of performance data that is collected by Easy Tier over a 24-hour operational cycle. This application generates a report that can be displayed with any Internet browser. The STAT provides configuration recommendations about solid-state drive (SSD)-class, Enterprise-class, and Nearline-class tiers, as well as information about rank utilization and volume heat distribution.

For an introduction of the STAT features, see 6.7, “Storage Tier Advisor Tool” on page 231.

1.3.5 I/O Priority Manager

I/O Priority Manager is an optional feature for the IBM System Storage DS8800 and DS8700 models. It enables more effective storage performance management combined with the ability to align quality of service (QoS) levels to separate workloads in the system. The DS8000 prioritizes access to system resources to achieve the desired qualities of service for the volume based on defined performance goals of high, standard, or low. I/O Priority Manager constantly monitors and balances system resources to help applications meet their performance targets automatically, without operator intervention.

It is increasingly common to use one storage system to serve many categories of workloads with different characteristics and requirements. The widespread use of virtualization and the advent of cloud computing that facilitates consolidating applications into a shared storage infrastructure are common practices. However, business-critical applications can suffer performance degradation because of resource contention with less important applications. Workloads are forced to compete for resources, such as disk storage capacity, bandwidth, DAs, and ranks.

Clients more often reserve the desired QoS for selected applications, even when the contentions occur. This environment is challenging because of the dynamic and diverse characteristics of storage workloads, different types of storage system components, and real-time requirements.

I/O Priority Manager at work

A *performance group* (PG) attribute associates the logical volume with a performance group object. Each performance group is associated with a performance policy that determines how the I/O Priority Manager processes I/O operations for the logical volume.

The I/O Priority Manager maintains statistics for the set of logical volumes in each performance group that can be queried. If management is performed for the performance policy, the I/O Priority Manager controls the I/O operations of all managed performance groups to achieve the goals of the associated performance policies. The performance group of a volume defaults to PG0, if none is specified. Table 1-2 lists the performance groups that are predefined with their associated performance policies.

Table 1-2 DS8000 I/O Priority Manager: Performance group to performance policy mapping

Performance group	Performance policy	Priority	QoS target	Ceiling (max. delay) factor [%]	Performance policy description
0	1	0	0 (N/A)	0 (N/A)	Fixed block, No Management CKD, No management
1 - 5	2	1	70	0 (N/A)	Fixed block high priority
6 - 10	3	5	40	500	Fixed block medium priority
11 - 15	4	15	0 (N/A)	2000	Fixed block low priority
16 - 18	16 - 18	0	0 (N/A)	0 (N/A)	CKD, No management
19	19	1	80	2000	CKD high priority 1
20	20	2	80	2000	CKD high priority 2
21	21	3	70	2000	CKD high priority 3
22	22	4	45	2000	CKD medium priority 1
23	23	4	5	2000	CKD medium priority 2
24	24	5	45	2000	CKD medium priority 3
25	25	6	5	2000	CKD medium priority 4
26	26	7	5	2000	CKD low priority 1
27	27	8	5	2000	CKD low priority 2
28	28	9	5	2000	CKD low priority 3
29	29	10	5	2000	CKD low priority 4
30	30	11	5	2000	CKD low priority 5
31	31	12	5	2000	CKD low priority 6

Each performance group comes with a predefined priority and QoS target. Because mainframe volumes and Open Systems volumes are on separate extent pools with different rank sets, they do not interfere with each other, unless in rare cases of overloaded DAs.

The user must associate the DS8000 logical volume with the performance policy that characterizes the expected type of workload for that volume. This policy assignment allows the I/O Priority Manager to determine the relative priority of I/O operations.

Open Systems use several performance groups (PG1 - PG5) and share the priority and QoS characteristics. You can put applications into different groups for monitoring purposes, without assigning different QoS priorities.

If the I/O Priority Manager detects resource overload conditions, such as resource contention that leads to insufficient response times for higher-priority volumes, it throttles the I/O for those volumes. The volumes are in lower-priority performance groups. This method allows the higher-performance group applications to run faster and again meet their QoS targets.

Important: Lower-priority I/O operations are delayed by I/O Priority Manager only if contention exists on a resource that causes a deviation from normal I/O operation performance. The I/O operations that are delayed are limited to operations that involve the RAID arrays or DAs that experience contention.

Performance groups are assigned to a volume at the time of the volume creation, as shown in Figure 1-3.

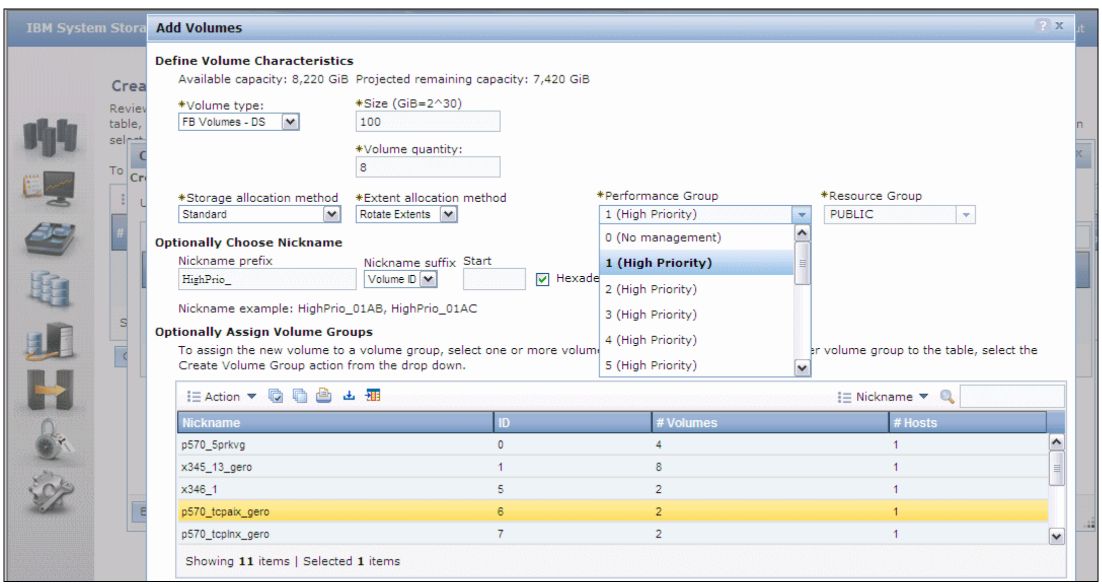


Figure 1-3 Creating new volumes with an assigned performance group and priority

You can also assign performance groups to existing volumes by using the DSCLI **`chfbvol`** and **`chckdvol`** commands. With the DS Storage Manager GUI, at any time, you can reassign volumes online to other performance groups with a lower or higher QoS priority if performance targets are either not met or constantly exceeded and inhibiting other applications.

Modes of operation

I/O Priority Manager can operate in the following modes:

- Disabled: I/O Priority Manager does not monitor any resources and does not alter any I/O response times.

- Monitor: I/O Priority Manager monitors resources and updates statistics that are available in performance data. This performance data can be accessed from the DS8000 command-line interface (DSCLI) or the GUI. No I/O response times are altered.
- Manage: I/O Priority Manager monitors resources and updates statistics that are available in performance data. This performance data can be accessed from the DSCLI or the GUI. I/O response times are altered on volumes that are in lower-QoS performance groups if resource contention occurs.

In both monitor and manage modes, I/O Priority Manager can send Simple Network Management Protocol (SNMP) traps to alert the user when certain resources detect a saturation event.

Monitoring and performance reports

I/O Priority Manager generates performance statistics every 60 seconds for DAs, ranks, and performance groups. These performance statistics samples are kept for a specified period. I/O Priority Manager maintains statistics for the last immediate set of values:

- 60 one-minute intervals
- 60 five-minute intervals
- 60 15-minute intervals
- 60 one-hour intervals
- 60 four-hour intervals
- 60 one-day intervals

An interesting option of I/O Priority Manager is to be able to monitor the performance of the entire storage system. For instance on a storage system, where all the volumes are still in their default performance group PG0 (which is not managed by I/O Priority Manager), a regular performance report of the machine can be obtained. See Example 1-1.

Example 1-1 Monitoring default performance group PG0 for one entire month, in one-day intervals

```
dscli> lsperfgrprpt -start 32d -stop 1d -interval 1d pg0
time                grp                  resrc            avIO  avMB     avresp pri avQ tgtQ %hlpT %dlyT %impt
==============================================================================================================
2011-10-01/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  8617 489.376  0.554    0 43  0    0     0     0
2011-10-02/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 11204 564.409  2.627    0 37  0    0     0     0
2011-10-03/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 21737 871.813  5.562    0 27  0    0     0     0
2011-10-04/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 21469 865.803  5.633    0 32  0    0     0     0
2011-10-05/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 23189 1027.818 5.413    0 54  0    0     0     0
2011-10-06/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 20915 915.315  5.799    0 52  0    0     0     0
2011-10-07/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 18481 788.450  6.690    0 41  0    0     0     0
2011-10-08/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 19185 799.205  6.310    0 43  0    0     0     0
2011-10-09/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 19943 817.699  6.069    0 41  0    0     0     0
2011-10-10/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 20752 793.538  5.822    0 49  0    0     0     0
2011-10-11/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 23634 654.019  4.934    0 97  0    0     0     0
2011-10-12/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 23136 545.550  4.961    0 145 0    0     0     0
2011-10-13/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 19981 505.037  5.772    0 92  0    0     0     0
2011-10-14/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  5962 176.957  5.302    0 93  0    0     0     0
2011-10-15/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  2286 131.120  0.169    0 135 0    0     0     0
2011-10-16/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  2287 131.130  0.169    0 135 0    0     0     0
2011-10-17/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  2287 131.137  0.169    0 135 0    0     0     0
2011-10-18/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 10219 585.908  0.265    0 207 0    0     0     0
2011-10-19/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 22347 1281.260 0.162    0 490 0    0     0     0
2011-10-20/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 13327 764.097  0.146    0 507 0    0     0     0
2011-10-21/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  9353 536.250  0.151    0 458 0    0     0     0
2011-10-22/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  9944 570.158  0.127    0 495 0    0     0     0
2011-10-23/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 11753 673.871  0.147    0 421 0    0     0     0
2011-10-24/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 11525 660.757  0.140    0 363 0    0     0     0
2011-10-25/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  5022 288.004  0.213    0 136 0    0     0     0
2011-10-26/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  5550 318.230  0.092    0 155 0    0     0     0
2011-10-27/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  8732 461.987  0.313    0 148 0    0     0     0
2011-10-28/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 13404 613.771  1.434    0 64  0    0     0     0
2011-10-29/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 25797 689.529  0.926    0 51  0    0     0     0
2011-10-30/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 25560 725.174  1.039    0 49  0    0     0     0
2011-10-31/17:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 25786 725.305  1.013    0 49  0    0     0     0
```

A client can create regular performance reports and store them easily, for instance on a weekly or monthly basis, such this report on the first day of a month, for the previous month. This report shows a long-term overview of the amount of IOPS, MB/s, and their average response times.

If all volumes are in their default performance group, which is PG0, Example 1-1 on page 20 is possible. However, as soon as we want to start the I/O Priority Manager QoS management and throttling of the lower-priority volumes, we need to put the volumes into non-default performance groups. We use PG1 - PG15 for Open Systems and PG19 - PG31 for CKD volumes.

Monitoring is then possible on the performance-group level, as shown in Example 1-2, the RAID-rank level, as shown in Example 1-3, or the DA pair level, as shown in Example 1-4 on page 22. Figure 1-4 shows how to obtain reports by using the DS Storage Manager GUI.

Example 1-2 Showing reports for a certain performance group PG28 for a certain time frame

```
dscli> lsperfgrprpt -start 3h -stop 2h pg28
time                grp                   resrc            avIO avMB   avresp pri avQ tgtQ %hlpT %dlyT %impt
========================================================================================================
2011-11-01/14:10:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  204 11.719 1.375    9 17  5    0     0     0
2011-11-01/14:15:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  401 23.020 1.273    9 18  5    0     0     0
2011-11-01/14:20:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181 1287 73.847 1.156    9 28  5    0     0     0
2011-11-01/14:25:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  315 18.102 1.200    9 19  5    0     0     0
2011-11-01/14:30:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  261 15.013 1.241    9 22  5    0     0     0
2011-11-01/14:35:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  330 18.863 1.195    9 24  5    0     0     0
2011-11-01/14:40:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  535 30.670 1.210    9 19  5    0     0     0
2011-11-01/14:45:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  514 29.492 1.209    9 20  5    0     0     0
2011-11-01/14:50:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181  172 9.895  1.191    9 21  5    0     0     0
2011-11-01/14:55:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181   91 5.262  0.138    9 299 5    0     0     0
2011-11-01/15:00:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181    0 0.000  0.000    9 0   5    0     0     0
2011-11-01/15:05:00 IBM.2107-75TV181/PG28 IBM.2107-75TV181    0 0.000  0.000    9 0   5    0     0     0
```

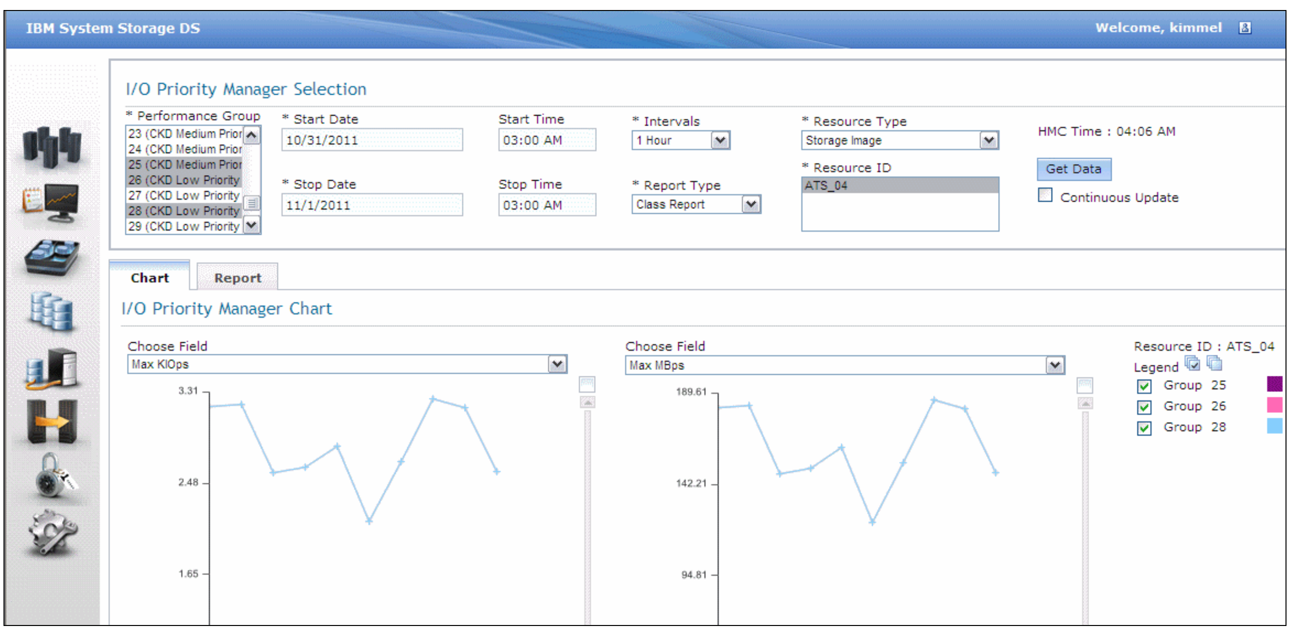


Figure 1-4 Getting I/O Priority Manager class reports and resource reports by using the GUI

Example 1-3 Monitoring on the rank level for rank R19

```
dscli> lsperfrescrpt R19
time                resrc avIO avMB  avresp %Hutl %hlpT %dlyT %impt
===================================================================
```

```
2011-11-25/12:30:00 R19   1552 6.357 7.828  100   24    7     202
2011-11-25/12:35:00 R19   1611 6.601 8.55   100   4     6     213
2011-11-25/12:40:00 R19   1610 6.596 7.888  50    7     4     198
2011-11-25/12:45:00 R19   1595 6.535 6.401  100   12    7     167
2011-11-25/12:50:00 R19   1586 6.496 8.769  50    5     3     219
```

The **lsperfrescrpt** command displays performance statistics for individual ranks, as shown in Example 1-3 on page 21:

- The first three columns show the average number of IOPS, throughput, and average response time, in milliseconds, for all I/Os on that rank.
- The %Hutl column shows the percentage of time that the rank utilization was high enough (over 33%) to warrant workload control.
- The %hlpT column shows the average percentage of time that I/Os were helped on this rank for all performance groups. This column shows the percentage of time where lower priority I/Os were delayed to help higher priority I/Os.
- The %dlyT column specifies the average percentage of time that I/Os were delayed for all performance groups on this rank.
- The %impt column specifies, on average, the length of the delay.

Example 1-4 Monitoring on DA pair level for DP1

```
dscli> lsperfgrprpt -dapair dp1 pg0
time                grp                   resrc avIO  avMB     avresp pri  avQ  tgtQ %hlpT %dlyT %impt
=====================================================================================================
2011-11-01/16:55:00 IBM.2107-75TV181/PG0 DP1   55234 1196.234 0.586    0 110 0    0     0     0
2011-11-01/17:00:00 IBM.2107-75TV181/PG0 DP1   51656 1190.322 0.627    0 110 0    0     0     0
2011-11-01/17:05:00 IBM.2107-75TV181/PG0 DP1   52990 1181.204 0.609    0 108 0    0     0     0
2011-11-01/17:10:00 IBM.2107-75TV181/PG0 DP1   55108 1198.362 0.589    0 109 0    0     0     0
2011-11-01/17:15:00 IBM.2107-75TV181/PG0 DP1   54946 1197.284 0.589    0 110 0    0     0     0
2011-11-01/17:20:00 IBM.2107-75TV181/PG0 DP1   49080 1174.828 0.657    0 113 0    0     0     0
2011-11-01/17:25:00 IBM.2107-75TV181/PG0 DP1   53034 1198.398 0.607    0 110 0    0     0     0
2011-11-01/17:30:00 IBM.2107-75TV181/PG0 DP1   55278 1193.400 0.583    0 109 0    0     0     0
2011-11-01/17:35:00 IBM.2107-75TV181/PG0 DP1   55112 1198.100 0.584    0 110 0    0     0     0
2011-11-01/17:40:00 IBM.2107-75TV181/PG0 DP1   53204 1197.236 0.610    0 112 0    0     0     0
2011-11-01/17:45:00 IBM.2107-75TV181/PG0 DP1   54316 1200.102 0.594    0 110 0    0     0     0
2011-11-01/17:50:00 IBM.2107-75TV181/PG0 DP1   55222 1190.582 0.583    0 110 0    0     0     0
```

I/O Priority Manager with System z

The DS8000 now also features I/O Priority Manager for CKD volumes. Together with z/OS Workload Manager (zWLM), this capability is designed to enable more effective storage consolidation and performance management combined with the ability to align QoS levels to separate workloads in the system. This capability is exclusive to the DS8000 and System z systems.

The DS8000 prioritizes access to system resources to achieve the desired QoS of the volume based on the defined performance goals (high, medium, or low) of any volume. I/O Priority Manager constantly monitors and balances system resources to help applications meet their performance targets automatically without operator intervention based on input from the zWLM software.

The zWLM integration for the I/O Priority Manager is available with z/OS V1.11 and higher. You must apply the following APARs: OA32298, OA34063, and OA34662.

You can run the I/O Priority Manager for z/OS the same way for Open Systems. A user can control the performance results of the CKD volumes and assign them online into higher-priority or lower-priority performance groups, if needed. The zWLM integration offers an additional level of automation. The zWLM can control and fully manage the performance behavior of all volumes.

With z/OS and zWLM software support, the user assigns application priorities through the Workload Manager. z/OS then assigns an “importance” value to each I/O, based on the zWLM inputs. Then, based on the prior history of I/O response times for I/Os with the same importance value, and based on the zWLM expectations for this response time, z/OS assigns an “achievement” value to each I/O.

The importance and achievement values for each I/O are compared. The I/O becomes associated with a performance policy, independently of the performance group or policy of the volume. When saturation or resource contention occurs, I/O is then managed according to the preassigned zWLM performance policy.

I/O Priority Manager and Easy Tier

I/O Priority Manager and Easy Tier work independently. They can coexist in the same pool and, if both enabled, provide independent benefits:

- I/O Priority Manager attempts to ensure that the most important I/O operations are serviced in resource contention situations when a rank is overloaded by the workload on the system by delaying less important I/Os. In contrast to Easy Tier, it does not relocate any data or extents. I/O Priority Manager is, in that sense, a reactive algorithm that works on a short-term time scale only when resource contention is present.
- Easy Tier (automatic mode) attempts to relocate extents across ranks and storage tiers to identify the most appropriate storage tier for the subvolume data based on the workload and access pattern. Easy Tier provides automatic cross-tier and intra-tier performance management through automated data relocation. Easy Tier helps maximize storage performance and storage economics throughout the system and maintain an optimal resource utilization.

  Easy Tier, although it is based on past performance monitoring, moves the extents proactively (except with warm-demote). Its goal is to minimize any potentially upcoming bottleneck conditions so that the system performs at its optimal point. Also, it works on a much longer time scale than I/O Priority Manager.

Together these features can help to consolidate various applications on a single DS8000 system. These features optimize overall performance through automated tiering in a simple and cost-effective manner while sharing resources. The DS8000 can help address storage consolidation requirements, which, in turn, helps to manage increasing amounts of data with less effort and lower infrastructure costs.

DS8000 I/O Priority Manager, REDP-4760, describes the I/O Priority Manager in detail. Also, the following IBM patent provides insight:

http://depatisnet.dpma.de/DepatisNet/depatisnet?action=pdf&docid=US000007228354B2&switchToLang=en

2

Hardware configuration

In this chapter, we describe the System Storage DS8800 hardware configuration and its relationship to the performance of the device.

Understanding the hardware components, the functions that are performed by each component, and the technology can help you select the correct components to order and the quantities of each component. However, do not focus too much on any one hardware component. Instead, ensure that you balance components to work together effectively. The ultimate criteria for storage server performance is the total throughput.

We describe the major DS8000 hardware components:

- Storage unit, processor complex, and storage facility image
- Cache
- Disk enclosures, PCIe infrastructure, and RIO-G interconnect
- Disk subsystem and device adapters
- Host adapters

2.1 Storage system

It is important to understand the naming conventions that are used to describe the DS8000 components and constructs.

Storage unit

A storage unit consists of a single DS8000 system (including expansion frames). A storage unit can consist of several frames: one base frame and up to three expansion frames for a DS8800 system and up to four expansion frames for a DS8700 system. The storage unit ID is the DS8000 base frame serial number, ending in 0 (for example, `75-06570`).

Processor complex

A DS8800 processor complex is one POWER6+ p570 copper-based symmetric multiprocessor (SMP) system unit. Each processor complex runs on 5.0 GHz, either in a 2-way or 4-way model.

On all DS8000 models, there are two processor complexes (servers), which are housed in the base frame. These processor complexes form a redundant pair so that if either processor complex fails, the surviving processor complex continues to run the workload.

Storage Facility Image (SFI)

In a DS8000, an SFI is a union of two logical partitions (processor LPARs), one from each processor complex. Each LPAR hosts one server. The SFI controls one or more device adapter (DA) pairs and two or more disk enclosures. Sometimes, an SFI is called a *storage image* or a *storage LPAR*.

In a DS8000, a server is effectively the software that uses a processor logical partition (a processor LPAR) and that has access to the memory and processor resources available on a processor complex. The DS8800 model 951, as well as the DS8700 model 941, are single-SFI models, and so, have one storage LPAR using 100% of the resources.

A Storage Facility Image has the following DS8000 resources dedicated to its use:

- Processors
- Cache and persistent memory
- I/O enclosures
- Disk enclosures

For the internal naming, we work with server 0 and server 1.

Figure 2-1 on page 27 shows an overview of the architecture of the DS8800 storage system.

The architecture of the previous system, DS8700, is similar to Figure 2-1 on page 27, with the following exceptions:

- Instead of POWER6+, the DS8700 runs POWER6 (with a slightly slower cycle speed).
- The internal cabling is Fibre Channel (FC)/copper instead of optical cabling.
- The DS8700 protocol to the disks is 2 Gbps. The DS8800 uses 8 Gbps up to the point of the FC-to-SAS bridge.
- The 4 Gbps host adapters of the DS8700 contained a PCI-X-to-PCIe interface, as explained in 2.5, “Host adapters” on page 46.

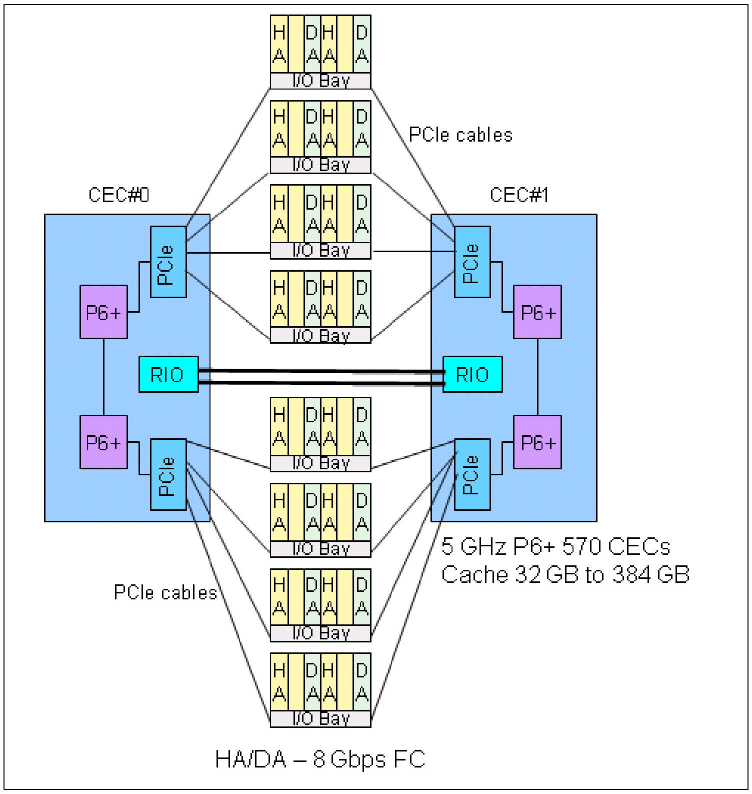


Figure 2-1 DS8800 general architecture

2.2 Processor memory and cache

The DS8800 and DS8700 models offer processor memory options of 32 GB, 64 GB, 128 GB, 256 GB, or 384 GB. The DS8800 Business-Class optionally starts at 16 GB, 32 GB, 64 GB, or 128 GB.

On all DS8000 models, each processor complex has its own system memory. Within each processor complex, the system memory is divided into these parts:

- Memory used for the DS8000 control program
- Cache
- Persistent cache

The amount allocated as persistent memory scales according to the processor memory that is selected.

2.2.1 Cache and I/O operations

Caching is a fundamental technique for hiding I/O latency. Cache is used to keep both the data read and written by the host servers. The host does not need to wait for the hard disk drive to either obtain or store the data that is needed, because cache can be used as an intermediate repository. Prefetching data from the disk for read operations, as well as operations of writing to the disk, are done by the DS8000 asynchronously from the host I/O processing.

Cache processing improves the performance of the I/O operations done by the host systems that attach to the DS8000. Cache size, the efficient internal structure, and algorithms of the

DS8000 are factors that improve I/O performance. The significance of this benefit is determined by the type of workload that is run.

Read operations

These operations occur when a host sends a read request to the DS8000:

- A cache hit occurs if the requested data resides in the cache. In this case, the I/O operation does not disconnect from the channel/bus until the read is complete. A read hit provides the highest performance.
- A cache miss occurs if the data is not in the cache. The I/O operation logically disconnects from the host. Other I/O operations occur over the same interface. A stage operation from the disk subsystem occurs.

Write operations: Fast writes

A fast write hit occurs when the write I/O operation completes as soon as the data received from the host is transferred to the cache and a copy is made in the persistent memory. Data written to a DS8000 is almost 100% fast-write hits. The host is notified that the I/O operation is complete as soon as the data is stored in the two locations.

The data remains in the cache and persistent memory until it is destaged, at which point it is flushed from cache. Destage operations of sequential write operations to RAID 5 arrays are done in parallel mode, writing a stripe to all disks in the RAID set as a single operation. An entire stripe of data is written across all the disks in the RAID array. The parity is generated one time for all the data simultaneously and written to the parity disk. This approach reduces the parity generation penalty associated with write operations to RAID 5 arrays. For RAID 6, data is striped on a block level across a set of drives, similar to RAID 5 configurations. A second set of parity is calculated and written across all the drives. This technique does not apply for the RAID 10 arrays, because there is no parity generation required. Therefore, no penalty is involved other than a double write when writing to RAID 10 arrays.

It is possible that the DS8000 cannot copy write data to the persistent cache because it is full, which can occur if all data in the persistent cache waits for destage to disk. In this case, instead of a fast write hit, the DS8000 sends a command to the host to retry the write operation. Having full persistent cache is not a good situation, because it delays all write operations. On the DS8000, the amount of persistent cache is sized according to the total amount of system memory. The amount of persistent cache is designed so that the probability is low of full persistent cache occurring in normal processing.

Cache management

The DS8000 system offers superior caching algorithms:

- Sequential Prefetching in Adaptive Replacement Cache (SARC)
- Adaptive Multi-stream Prefetching (AMP)
- Intelligent Write Caching (IWC)

IBM Storage Development in partnership with IBM Research developed these algorithms.

Sequential Prefetching in Adaptive Replacement Cache (SARC)

SARC is a self-tuning, self-optimizing solution for a wide range of workloads with a varying mix of sequential and random I/O streams. This cache algorithm attempts to determine four things:

- When data is copied into the cache
- Which data is copied into the cache
- Which data is evicted when the cache becomes full
- How the algorithm dynamically adapts to different workloads

The DS8000 cache is organized in 4 KB pages called *cache pages* or *slots*. This unit of allocation ensures that small I/Os do not waste cache memory.

The decision to copy an amount of data into the DS8000 cache can be triggered from two policies: demand paging and prefetching.

Demand paging means that eight disk blocks (a 4K cache page) are brought in only on a cache miss. Demand paging is always active for all volumes and ensures that I/O patterns with locality find at least some recently used data in the cache.

Prefetching means that data is copied into the cache even before it is requested. To prefetch, a prediction of likely future data access is required. Because effective, sophisticated prediction schemes need extensive history of the page accesses, the algorithm uses prefetching only for sequential workloads. Sequential access patterns are commonly found in video-on-demand, database scans, copy, backup, and recovery. The goal of sequential prefetching is to detect sequential access and effectively preload the cache with data to minimize cache misses.

For prefetching, the cache management uses tracks. A track is a set of 128 disk blocks (16 cache pages). To detect a sequential access pattern, counters are maintained with every track to record if a track is accessed together with its predecessor. Sequential prefetching becomes active only when these counters suggest a sequential access pattern. In this manner, the DS8000 monitors application read patterns and dynamically determines whether it is optimal to stage into cache:

- Only the page requested
- The page requested, plus remaining data on the disk track
- An entire disk track or multiple disk tracks not yet requested

The decision of when and what to prefetch is made on a per-application basis (rather than a system-wide basis) to be responsive to the data reference patterns of various applications that can run concurrently.

With the System z integration of newer DS8000 codes, a host application, such as DB2, can send cache hints to the storage system and manage the DS8000 prefetching, reducing the number of I/O requests.

To decide which pages are flushed when the cache is full, sequential data and random (non-sequential) data are separated into different lists, as illustrated in Figure 2-2 on page 30.

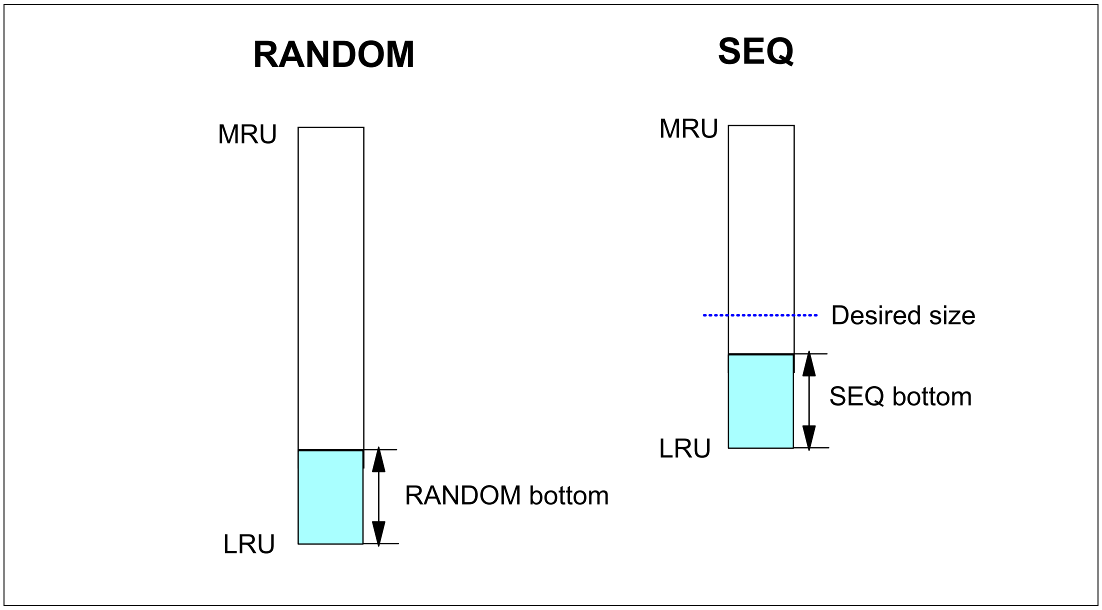


Figure 2-2 Sequential Prefetching in Adaptive Replacement Cache (SARC)

In Figure 2-2, a page that is brought into the cache by simple demand paging is added to the Most Recently Used (MRU) head of the RANDOM list. With no further references to that page, it moves down to the Least Recently Used (LRU) bottom of the list. A page that is brought into the cache by a sequential access or by sequential prefetching is added to the MRU head of the sequential (SEQ) list. It moves down that list as more sequential reads are done. Additional rules control the management of pages between the lists so that the same pages are not kept in memory twice.

To follow workload changes, the algorithm trades cache space between the RANDOM and SEQ lists dynamically. Trading cache space allows the algorithm to prevent one-time sequential requests from filling the entire cache with blocks of data with a low probability of being read again. The algorithm maintains a desired size parameter for the SEQ list. The desired size is continually adapted in response to the workload. Specifically, if the bottom portion of the SEQ list is more valuable than the bottom portion of the RANDOM list, the desired size of the SEQ list is increased. Otherwise, the desired size is decreased. The constant adaptation strives to make optimal use of limited cache space and delivers greater throughput and faster response times for a specific cache size.

SARC performance

IBM simulated a comparison of cache management with and without the SARC algorithm. The new algorithm, with no change in hardware, provided these results:

- Effective cache space: 33% greater
- Cache miss rate: 11% reduced
- Peak throughput: 12.5% increased
- Response time: 50% reduced

Figure 2-3 on page 31 shows the improvement in response time due to SARC.

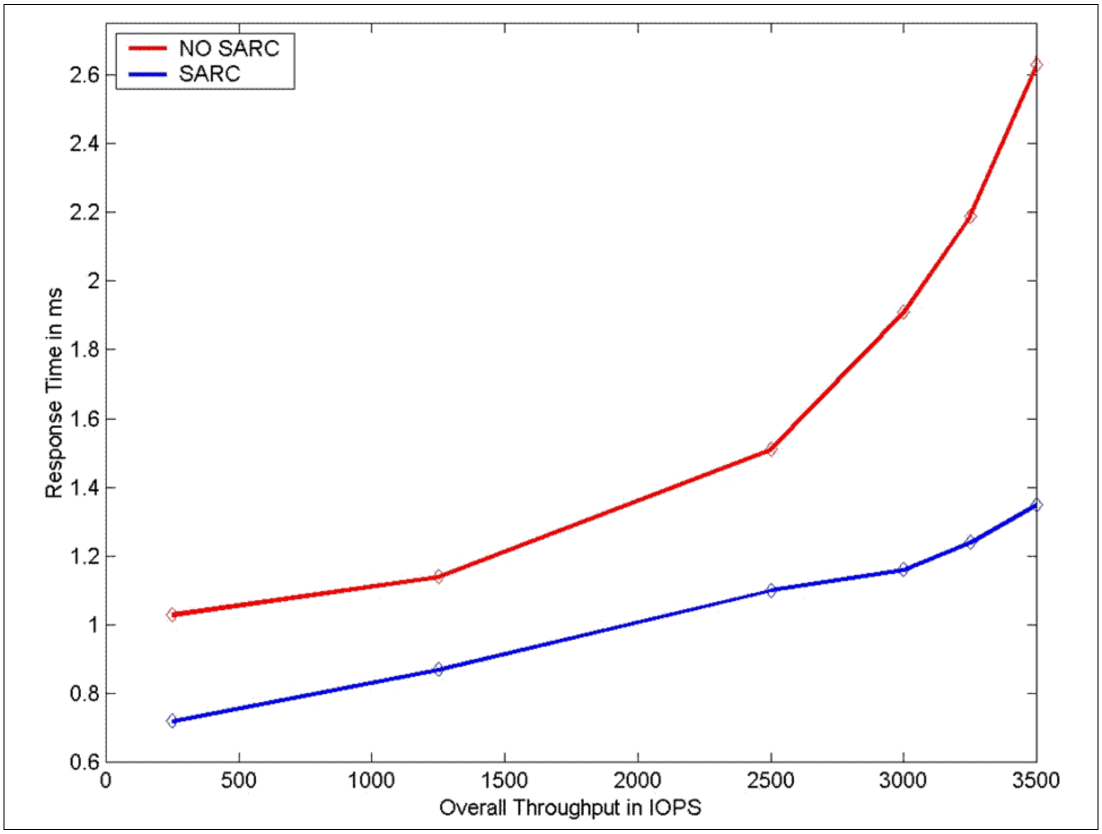


Figure 2-3 Response time improvement with SARC

Adaptive Multi-stream Prefetching (AMP)

SARC dynamically divides the cache between the RANDOM and SEQ lists. The SEQ list maintains pages that are brought into the cache by sequential access or sequential prefetching.

The SEQ list is managed by the Adaptive Multi-stream Prefetching (AMP) technology, which was developed by IBM research. AMP introduces an autonomic, workload-responsive, self-optimizing prefetching technology that adapts both the amount of prefetch and the timing of prefetch on a per-application basis in order to maximize the performance of the system. The AMP algorithm solves two problems that plague most other prefetching algorithms:

- Prefetch wastage occurs when prefetched data is evicted from the cache before it can be used.
- Cache pollution occurs when less useful data is prefetched instead of more useful data.

By choosing the prefetching parameters, AMP provides optimal sequential read performance and maximizes the aggregate sequential read throughput of the system. The amount prefetched for each stream is dynamically adapted according to the needs of the application and the space available in the SEQ list. The timing of the prefetches is also continuously adapted for each stream to avoid misses and, at the same time, to avoid any cache pollution.

AMP dramatically improves performance for common sequential and batch processing workloads. It also provides performance synergy with DB2 by preventing table scans from being I/O bound. It improves the performance of index scans and DB2 utilities, such as Copy and Recover. Furthermore, AMP reduces the potential for array hot spots that result from extreme sequential workload demands.

SARC and AMP play complementary roles. SARC carefully divides the cache between the RANDOM and the SEQ lists to maximize the overall hit ratio. AMP manages the contents of the SEQ list to maximize the throughput obtained for the sequential workloads. SARC affects cases that involve both random and sequential workloads. AMP helps any workload that has a sequential read component, including pure sequential read workloads.

Intelligent Write Caching (IWC)

Another additional cache algorithm, the *Intelligent Write Caching* (IWC), is implemented in the DS8000 series. IWC improves performance through better write-cache management and a better destaging order of writes. This algorithm is a combination of CLOCK, a predominantly read-cache algorithm, and CSCAN, an efficient write-cache algorithm. Out of this combination, IBM produced a powerful and widely applicable write-cache algorithm.

The CLOCK algorithm uses *temporal* ordering. It keeps a circular list of pages in memory, with the "hand" pointing to the oldest page in the list. When a page needs to be inserted in the cache, then an *R* (recency) bit is inspected at the "hand" location. If *R* is zero, the new page is put in place of the page to which the "hand" points and *R* is set to 1. Otherwise, the *R* bit is cleared and set to zero. Then, the clock hand moves one step clockwise forward and the process is repeated until a page is replaced.

The CSCAN algorithm uses *spatial* ordering. The CSCAN algorithm is the circular variation of the SCAN algorithm. The SCAN algorithm tries to minimize the disk head movement when servicing read and write requests. It maintains a sorted list of pending requests along with the position on the drive of the request. Requests are processed in the current direction of the disk head, until it reaches the edge of the disk. At that point, the direction changes. In the CSCAN algorithm, the requests are always served in the same direction. When the head arrives at the outer edge of the disk, it returns to the beginning of the disk and services the new requests in this direction only. This algorithm results in more equal performance for all head positions.

The idea of IWC is to maintain a sorted list of write groups, as in the CSCAN algorithm. The smallest and the highest write groups are joined, forming a circular queue. The addition is to maintain a recency bit for each write group, as in the CLOCK algorithm. A write group is always inserted in its correct sorted position, and the recency bit is set to 0 at the beginning. When a write hit occurs, the recency bit is set to 1. The destage operation proceeds. The destage pointer is maintained that scans the circular list looking for destage victims. Now, this algorithm only allows destaging of write groups whose recency bit is 0. The write groups with a recency bit of 1 are skipped and the recent bit is then turned off and reset to 0. This method gives an "extra life" to those write groups that are hit since the last time the destage pointer visited them.

Figure 2-4 on page 33 shows how this mechanism works.

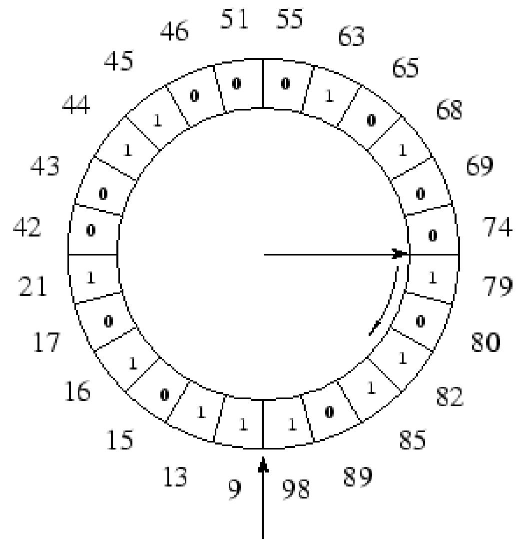


Figure 2-4 Intelligent Write Caching

In the DS8000 implementation, an IWC list is maintained for each rank. The dynamically adapted size of each IWC list is based on the workload intensity on each rank. The rate of destage is proportional to the portion of nonvolatile storage (NVS) occupied by an IWC list. The NVS is shared across all ranks in a cluster. Furthermore, destages are smoothed out so that write bursts are not translated into destage bursts.

IWC has better or comparable peak throughput to the best of CSCAN and CLOCK across a wide gamut of write-cache sizes and workload configurations. In addition, even at lower throughputs, IWC has lower average response times than CSCAN and CLOCK. The random-write parts of workload profiles benefit from the IWC algorithm. The costs for the destages are minimized, and the number of possible write-miss IOPS greatly improves compared to a system not utilizing IWC.

The IWC algorithm can be applied to storage systems, servers, and their operating systems. The DS8000 implementation is the first for a storage system. Because IBM is getting patents for this algorithm and the other advanced cache algorithms, it is unlikely that a competitive system uses them.

2.2.2 Determining the right amount of cache storage

A common question is "How much cache do I need in my DS8000?" Unfortunately, there is no quick and easy answer.

There are a number of factors that influence cache requirements:

- Is the workload sequential or random?
- Are the attached host servers System z or Open Systems?
- What is the mix of reads to writes?
- What is the probability that data will be needed again after its initial access?
- Is the workload cache friendly (a cache-friendly workload performs better with relatively large amounts of cache)?

It is a common approach to base the amount of cache on the amount of disk capacity, as shown in these common general rules:

- For Open Systems, each TB of capacity needs 2 GB - 4 GB of cache.
- For System z, each TB of disk capacity needs 2 GB - 5 GB of cache.

For SAN Volume Controller (SVC) attachments, consider the SVC node cache in this calculation, which might lead to a slightly smaller DS8000 cache. However, most installations come with a minimum of 128 GB of DS8000 cache. Using solid-state drives (SSDs) in the DS8000 does not change the prior values typically. SSDs are beneficial with cache-unfriendly workload profiles, because SSDs generally reduce the cost of cache misses.

Most storage servers support a mix of workloads. These general rules can work well, but many times, they do not. Use a general rule only if you have no other information on which to base your selection.

When coming from an existing disk storage server environment and you intend to consolidate this environment into DS8000s, follow these recommendations:

- Choose a cache size for the DS8000 series that has a similar ratio between cache size and disk storage to that of the configuration that you currently use.
- When you consolidate multiple disk storage servers, configure the sum of all cache from the source disk storage servers for the target DS8000 processor memory or cache size.

For example, consider replacing four DS8300s, each with 21 TB and 64 GB cache, with a single DS8800. The ratio between cache size and disk storage for each DS8300 is 0.3% (64 GB/21 TB). The new DS8800 is configured with 110 TB to consolidate the four 21 TB DS8300s, plus provide capacity for growth. This DS8800 requires 330 GB of cache to keep the original cache-to-disk storage ratio. Round up to the next available memory size, which is 384 GB for this DS8800 configuration. When using an SVC in front, round down to 256 GB of DS8800 cache.

The cache size is not an isolated factor when estimating the overall DS8000 performance. Consider it with the DS8000 model, the capacity and speed of the disk drives, and the number and type of host adapters. Larger cache sizes mean that more reads are satisfied from the cache, which reduces the load on device adapters (DAs) and disk drive modules (DDMs) associated with reading data from disk. To see the effects of different amounts of cache on the performance of the DS8000, run a Disk Magic model. See 6.1, “Disk Magic” on page 176.

2.3 I/O enclosures, PCIe infrastructure, and RIO-G interconnect

We describe I/O enclosures, PCIe infrastructure, and RIO-G interconnect in the following sections.

2.3.1 I/O enclosures

The DS8800 base frame and first expansion frame (if installed) both contain I/O enclosures. I/O enclosures are installed in pairs. There can be one or two I/O enclosure pairs installed in the base frame (depending on configuration: 2-way or 4-way). Two I/O enclosures are installed in the first expansion frame. Each I/O enclosure has six slots. DAs and host adapters (HAs) are installed in the I/O enclosures. The I/O enclosures provide connectivity between the processor complexes and the HAs or DAs. The DS8800 can have up to two DAs and two HAs installed in an I/O enclosure (DS8700 can have up to four HAs per enclosure, when using the 4 Gbps).

Depending on the number of installed disks and the number of required host connections, several I/O enclosures might not contain any adapters.

DAs are called *DA pairs*, because DAs are always installed in quantities of two (one DA is attached to each processor complex). The members of a DA pair are split across two I/O enclosures for redundancy. The number of installed disk devices determines the number of required DAs. In any I/O enclosure, the number of individual DAs installed can be zero, one, or two.

2.3.2 PCIe infrastructure

The DS8800 processor complex and the DS8700 processor complex use a PCI Express (PCIe) infrastructure to access the I/O enclosures. This infrastructure greatly improves performance over previous DS8000 models. Each CEC has an onboard GX+ bus to a P5IOC2 adapter. This P5IOC2 adapter drives four single-port PCIe adapters, which connect to four I/O enclosures. There is also a second GX+ bus, which is driven by the second CPU module, if installed (4-way feature). The second GX+ bus drives a 4-port PCIe adapter, which connects to the other four I/O enclosures.

2.3.3 RIO-G interconnect for server-to-server communication

In a DS8800, the remote I/O (RIO)-G ports are used for inter-processor communication that is isolated from I/O traffic. The RIO-G evolved from earlier versions of the RIO interconnect. Each RIO-G port can operate at 1 GHz in bidirectional mode and can pass data in each direction on each cycle of the port. It is designed as a high performance, self-healing interconnect.

The first DS8000 models, DS8100 and DS8300, used the RIO-G connection with the I/O bays on them in loops. The approximate 2 GB/s throughput of the RIO-G port limited the maximum sequential throughput of the older models. The newer architecture (Figure 2-1 on page 27) is not limited in this way, because all I/O data traffic occurs directly between each I/O bay and both processor complexes.

2.4 Disk subsystem

The DS8800 models use a selection of Serial-Attached SCSI (SAS) 2.0 interface disk drives. Previous models, such as the DS8700, used Fibre Channel (FC) disk drives. The disk subsystem consists of three components:

- The DAs are in the I/O enclosures. These DAs are RAID controllers that are used by the storage images to access the RAID arrays.
- The DAs connect to switched controller cards in the disk enclosures, and they create a switched FC disk network.
- The disks are called *disk drive modules* (DDMs).

2.4.1 Device adapters

Each DS8000 device adapter (DA), which is installed in the I/O enclosure, provides four FC ports. These ports are used to connect the processor complexes to the disk enclosures. The DA is responsible for managing, monitoring, and rebuilding the disk RAID arrays. The PCIe RAID DA is built on PowerPC® technology with four FC ports and high-function, high-performance application-specific integrated circuits (ASICs). To ensure maximum data integrity, the RAID DA supports metadata creation and checking. The RAID DA is based on PCIe Gen. 2 and operates at 8 Gbps.

DAs are installed in pairs, because each processor complex requires its own DA to connect to each disk enclosure for redundancy. DAs in a pair are installed in separate I/O enclosures to eliminate the I/O enclosure as a single point of failure.

Each DA performs the RAID logic and frees up the processors from this task. The actual throughput and performance of a DA is not only determined by the port speed and hardware that are used, but also by the firmware efficiency. Figure 2-5 shows the detailed cabling between the DAs and the 24-drive Gigapacks.

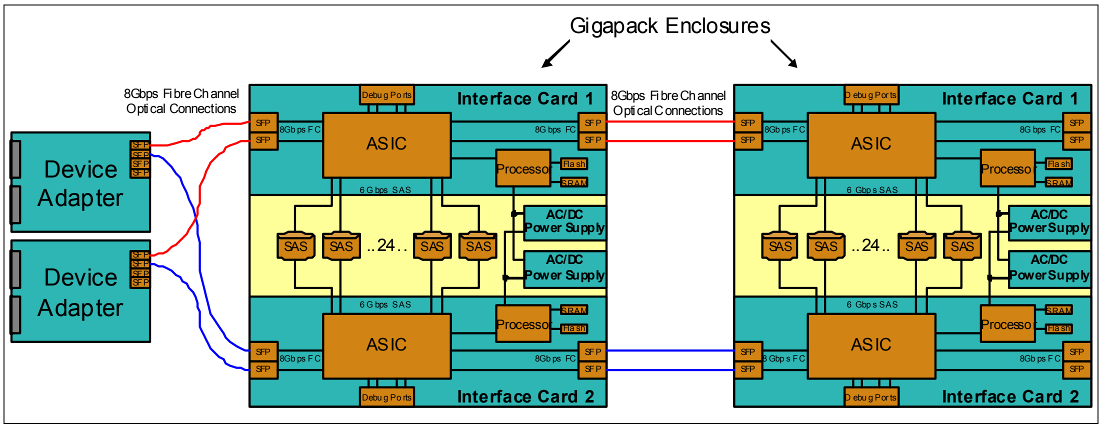


Figure 2-5 Detailed DA disk back-end diagram

The ASICs provide the FC-to-SAS bridging function from the external SFP connectors to each of the ports on the SAS disk drives. The processor is the controlling element in the system.

2.4.2 DS8800 Fibre Channel switched interconnection at the back end

DS8800 works with SAS disks. Until shortly before the FC-to-SAS conversion is made, FC switching is used in the DS8800 back end.

The FC technology is commonly used to connect a group of disks in a daisy-chained fashion in a Fibre Channel Arbitrated Loop (FC-AL). For commands and data to get to a particular disk, they must traverse all disks ahead of it in the loop. Conventional FC-AL has these shortcomings:

- In arbitration, each disk within an FC-AL loop competes with the other disks to get on the loop, because the loop supports only one operation at a time.
- If a failure of a disk or connection in the loop occurs, it can be difficult to identify the failing component, which leads to lengthy problem determination. Problem determination is more difficult when the problem is intermittent.
- A third issue with conventional FC-AL is the increasing amount of time that it takes to complete a loop operation as the number of devices increases in the loop.

Switched disk architecture

To overcome the arbitration issue within FC-AL, the DS8800 architecture is enhanced by adding a switch-based approach and creating FC-AL switched loops. It is called a *Fibre Channel switched disk system.*

These switches use the FC-AL protocol and attach to the SAS drives (bridging to SAS protocol) through a point-to-point connection. The arbitration message of a drive is captured in the switch, processed, and propagated back to the drive, without routing it through all the other drives in the loop.

Performance is enhanced because both DAs connect to the switched FC subsystem back end, as shown in Figure 2-6. Each DA port can concurrently send and receive data.

The two switched point-to-point connections to each drive, which also connect both DAs to each switch, have these characteristics:

- There is no arbitration competition and interference between one drive and all the other drives, because there is no hardware in common for all the drives in the FC-AL loop. This approach leads to an increased bandwidth with the full 8 Gbps FC speed to the back end where the FC-to-SAS conversion is made. This approach uses the full SAS 2.0 speed for each drive.
- This architecture doubles the bandwidth over conventional FC-AL implementations due to two simultaneous operations from each DA to allow for two concurrent read operations and two concurrent write operations at the same time.
- In addition to superior performance, this setup offers improved reliability, availability, and serviceability (RAS) over conventional FC-AL. The failure of a drive is detected and reported by the switch. The switch ports distinguish between intermittent failures and permanent failures. The ports understand intermittent failures, which are recoverable, and collect data for predictive failure statistics. If one of the switches fails, a disk enclosure service processor detects the failing switch and reports the failure using the other loop. All drives can still connect through the remaining switch.

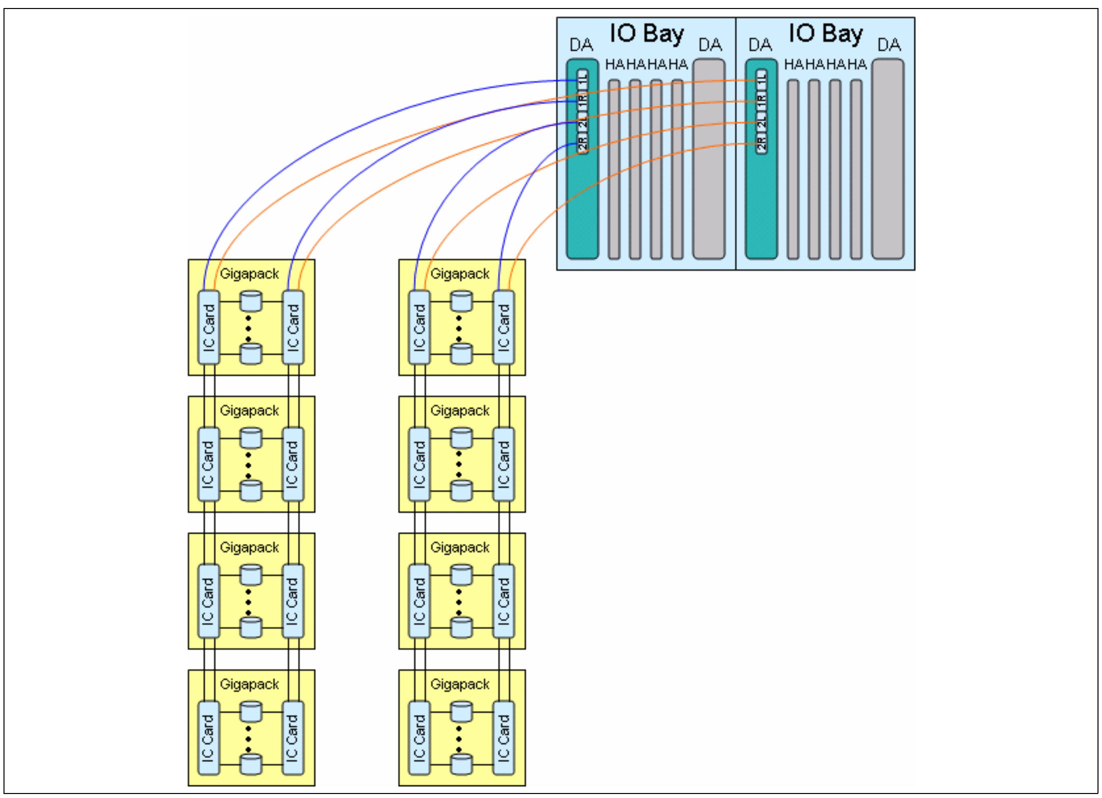


Figure 2-6 High availability and increased bandwidth connect both DAs to two logical loops

2.4.3 Disk enclosures

DS8800 disks are mounted in a disk enclosure, or *Gigapack*. Each enclosure holds 24 disks, when using the small form factor (SFF) 2.5 inch disks. Disk enclosures are called *disk enclosure pairs* or *expansion enclosure pairs*, because you order and install them in groups of two.

The DS8800 now supports two types of high-density storage enclosure: the 2.5 inch SFF enclosure and the new 3.5 inch large form factor (LFF) enclosure. The new high-density and lower-cost LFF storage enclosure accepts 3.5 inch drives, offering 12 drive slots. The previously introduced SFF enclosure offers twenty-four 2.5 inch drive slots. The front of the LFF enclosure differs from the front of the SFF enclosure, with its 12 drives slotting horizontally rather than vertically.

DDMs are added in increments of 16 (except for SSDs or nearline disks, which can also be ordered with a granularity of eight). For each group of 16 disks, eight are installed in the first enclosure and eight are installed in the next adjacent enclosure. These 16 disks form two array sites of eight DDMs each, from which RAID arrays are built during the logical configuration process. For each array site, four disks are from the first (or third, fifth, and so on), and four disks are from the second (or forth, sixth, and so on) enclosure.

All disks within a disk enclosure pair must be the same capacity and rotation speed. A disk enclosure pair that contains less than 48 DDMs must also contain dummy carriers called *fillers*. These fillers are used to maintain airflow.

SSDs can be ordered in increments of eight drives (= one array-site/rank = half drive set). However, we suggest that you order them in increments of 16 drives (= two array-sites/ranks = one drive set). Then, you can create a balanced configuration across the two processor complexes, especially with the high-performance capabilities of SSDs. In general, an uneven number of ranks of similar type drives (especially SSDs) can cause an imbalance in resources, such as cache or processor usage. Use a balanced configuration with an even number of ranks and extent pools, for instance, one even and one odd extent pool, and each with one SSD rank. This balanced configuration enables Easy Tier automatic SSD/hard disk drive (HDD) cross-tier performance optimization on both processor complexes. It distributes the overall workload evenly across all resources. With only one SSD rank and multiple HDD ranks of the same drive type, you can only assign the SDD rank to *one* extent pool. And, you can run Easy Tier automated SSD/HDD cross-tier performance optimization on *one* processor complex only.

For nearline HDDs, however, especially with only three array-sites per 3.5 inch disk enclosure pair on a DS8800 system (in contrast to six array-sites in a 2.5 inch disk enclosure pair), it is not so critical as it is with SSDs to go with an uneven number of ranks. nearline HDDs show generally lower performance characteristics.

Arrays across loops

Each array site consists of eight DDMs. Four DDMs are taken from one enclosure, and four DDMs are taken from a neighboring enclosure, of an enclosure pair. When a RAID array is created on the array site, half of the array is in each enclosure. Because the first enclosure is on one switched loop, and the adjacent enclosure is on another, the array is split across two loops, which is called *array across loops (AAL)*.

By putting half of each array on one loop and half of each array on another loop, there are more data paths into each array. This design provides a performance benefit, particularly in situations where a large amount of I/O goes to one array, such as sequential processing and array rebuilds.

2.4.4 DDMs

At the time of writing this book, the DS8800 provides a choice of the following DDM types:

- 146 GB, 15K rpm SAS disk, 2.5 inch SFF
- 300 GB, 15K rpm SAS disk, 2.5 inch SFF
- 450 GB, 10K rpm SAS disk, 2.5 inch SFF
- 600 GB, 10K rpm SAS disk, 2.5 inch SFF
- 900 GB, 10K rpm SAS disk, 2.5 inch SFF
- 3 TB, 7,200 rpm Nearline-SAS disk, 3.5 inch LFF
- 300 GB e-MLC SAS SSDs (enterprise-grade Multi-Level Cell Solid-State Drives), 2.5 inch SFF

The 15K and 10K rpm disks are also available as encrypted drives or Full Disk Encryption (FDE). FDE drives have essentially the same performance as the non-encrypted drives. Additional drive types are constantly in evaluation and added to this list when available.

The DS8700 provides a choice of the following DDM types:

- 300 GB, 15K rpm FC disk, 3.5 inch
- 450 GB, 15K rpm FC disk, 3.5 inch
- 600 GB, 15K rpm FC disk, 3.5 inch
- 2 TB, 7,200 rpm SATA disk, 3.5 inch
- 600 GB e-MLC FC SSDs (enterprise-grade Multi-Level Cell Solid-State Drives), 3.5 inch

These disks provide a range of options to meet the capacity and performance requirements of various workloads, and to introduce automated multi-tiering.

2.4.5 SAS 2.5 inch drives compared to FC 3.5 inch drives

When comparing the SAS 2.0 SFF drives that we now mainly use in the DS8800 with the former FC 3.5 inch drives in the DS8700, it is difficult to compare them directly. The faster internal architecture of the DS8800 influences the results. For a comparable drive (rpm) speed, the SFF SAS drives are faster.

Figure 2-7 shows a comparison between the DS8700 15K rpm drives (blue) to the current 15K and 10K rpm drives in the DS8800 for one RAID-5 array, for both random and sequential workloads.

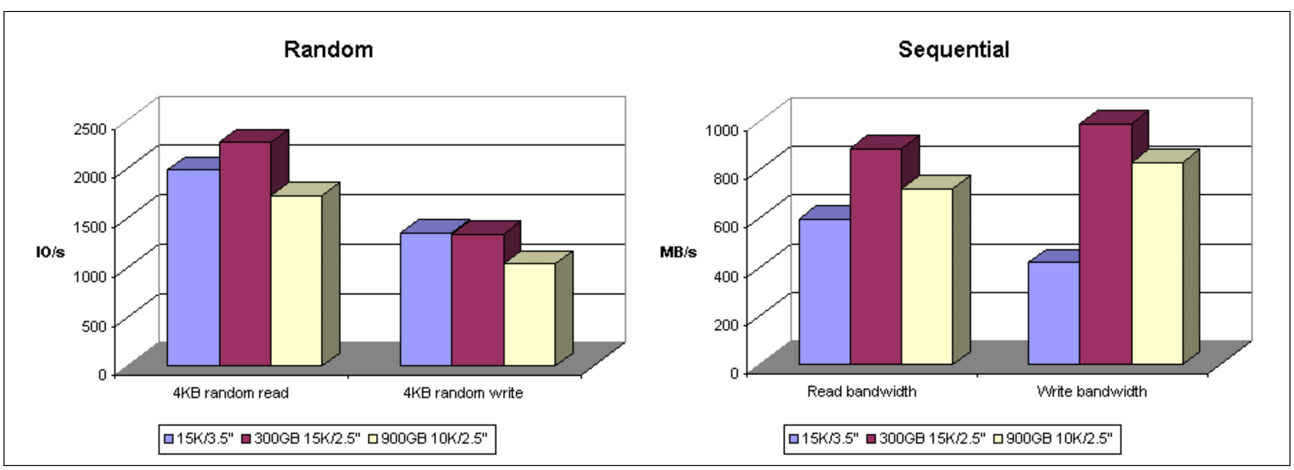


Figure 2-7 Comparison 2.5-inch SAS (DS8800) versus 3.5-inch FC drives (DS8700) for one RAID-5 array

Sample data: The sample performance measurement data shown in Figure 2-7 is for comparative purposes only. We collected this data in a controlled laboratory environment at a specific point in time by using the configurations, hardware, and firmware levels available at that time. Current performance in real-world environments varies. The data is intended only to help illustrate how different hardware technologies behave in relation to each other. Contact your IBM representative or IBM Business Partner if you have questions about the expected performance capability of IBM products in your environment.

For random reads, which are the most prevalent type of reads in a workload profile, consider these comparisons:

- When comparing 4K random reads, the 2.5 inch 10K SAS drives deliver only 10% fewer IOPS than the 3.5 inch 15K FC drives.
- The 2.5 inch 15K SAS drives deliver around 10% more IOPS than the 3.5 inch 15K FC drives.

For sequential loads, the DS8800 2.5 inch 10K SAS drives perform equivalently or better than the DS8700 3.5 inch 15K FC drives. The 2.5 inch SFF drives consume half as much power as the 3.5 inch HDDs and save on floor space.

The 10K rpm Enterprise-SAS HDDs with larger capacities are often combined with SSDs to replace installations with 15K rpm FC small-capacity drives. This combination decreases the footprint. Ensure that you analyze the workload and size the system.

2.4.6 Enterprise drives compared to nearline drives

The Enterprise-SAS drives for the System Storage DS8800 run at 10,000 rpm - 15,000 rpm. The nearline drives run at 7,200 rpm. The rotational speed of the disk drives is an important aspect of performance. For the best performance with random workloads, such as online transaction processing (OLTP) applications, choose Enterprise drives or an Enterprise HDD and SSD combination. For pure sequential streaming loads, nearline drives deliver almost the same throughput as Enterprise-class drives or SSD-class drives at a lower cost. For random workloads, the IOPS performance between nearline class drives and Enterprise class drives differs considerably and directly translates into longer access times for nearline drives. However, nearline drives offer large capacities and a good price-per-TB ratio. Due to their slower random performance, we suggest nearline drives only for small I/O access densities. *Access density* is the amount of I/Os per second per gigabyte of usable storage (IOPS/GB).

Tip: Use Enterprise drives to project or size a new DS8800 system as a core tier. Then, add an SSD percentage to these pools to increase performance. Introduce nearline drives later when you are in production. The IBM System Storage DS8000 Storage Tier Advisor Tool (STAT) application can show the cold data that can be demoted to less expensive drives, when further capacity upgrades are pending. See “Drive Selection with Easy Tier” in *IBM System Storage DS8800 and DS8700 Performance with Easy Tier 3rd Generation*, WP102024, for additional information.

Another difference between these drive types is the RAID rebuild time after a drive failure. This rebuild time grows with larger capacity drives. Therefore, RAID 6 is used for the large-capacity nearline drives to prevent a second disk failure during rebuild from causing a loss of data. See 3.1.2, “RAID 6 overview” on page 53.

2.4.7 Solid-state drives: SLC and E-MLC

SSDs have a high IOPS capability, especially for small blocks, when compared to HDDs (spinning drives). SSDs can respond with short response times (lower than 1 ms) for a 4K cache-miss. SSDs are suitable for workloads that are small-block random and cache-unfriendly workload profiles. For large blocks, or for workloads that are generally cache friendly, SSDs currently do not offer a comparable price-per-performance ratio as HDDs for this part of the workload profile.

Easy Tier in automatic mode considers these differences and measures for each extent the accumulated response times. It also differentiates the workload profile between random/sequential, reads/writes. It considers the blocksize to determine whether to move an extent upward to SSDs. Loads with large-block workload profiles are not moved up to SSDs, because HDDs can respond to these I/O requests.

Figure 2-8 shows that for small-block (4K) 100% random-read loads, one SSD rank can deliver over 40,000 IOPS. Writes, or a read+write mixture, show smaller results, but the results are still good when compared to one HDD rank (as shown in Figure 2-7 on page 39). We also see that the DS8800 performs better when used with SSDs than the DS8700 due to the enhanced DAs.

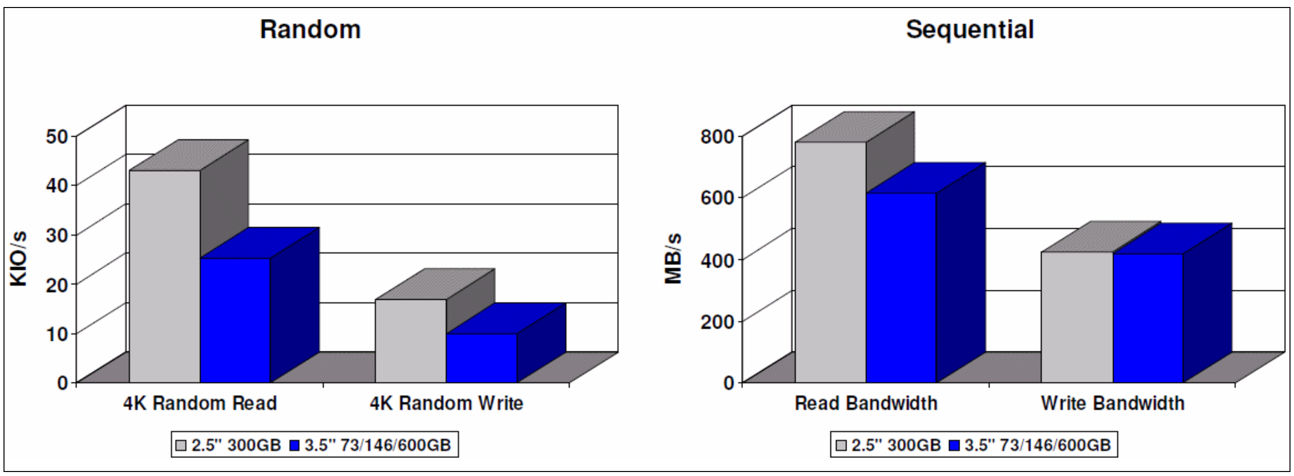


Figure 2-8 SSD comparison DS8800 (2.5 inch) versus DS8700 (3.5 inch) for random and sequential with one array RAID 5

> **Sample data:** The sample performance measurement data shown in Figure 2-8 is for comparative purposes only. We collected the data in a controlled laboratory environment at a specific point in time by using the configurations, hardware, and firmware levels available at that time. Current performance in real-world environments varies. The data is intended to help illustrate how different hardware technologies behave in relation to each other. Contact your IBM representative or IBM Business Partner if you have questions about the expected performance capability of IBM products in your environment.

Sequential loads, as shown in Figure 2-8, are slightly faster than the HDD sequential throughputs that are shown in Figure 2-7 on page 39. However, the difference is not large enough to justify using SSDs for large-block sequential loads.

SSDs are available as single-level cell (SLC) and multi-level cell (MLC), which refers to whether we use two or four voltage states per flash cell (NAND transistor). MLC SSDs allow higher capacities, by storing 2 bits per cell, but fewer writes per flash cell. SSD technology is evolving at a fast pace. In the past, we used SLC SSDs for enterprise-class applications.

Now, a generation of e-MLC (enterprise-grade or enhanced multi-level cell) SSDs offer enhanced endurance and a longer lifetime than typical HDDs under the expected heavy-duty cycles when using SSDs in the highest tier.

Over-provisioning techniques are used for failing cells, and data in worn-out cells is copied proactively. There are several algorithms for wear-leveling across the cells. The algorithms include allocating a rarely used block for the next block to use or moving data internally to less-used cells to enhance lifetime. Error detection and correction mechanisms are used, and bad blocks are flagged.

SSDs are mature and can be used in critical production environments. Their high performance for small-block/random workloads makes them financially viable for part of a hybrid HDD/SSD pool mix.

2.4.8 Installation order

Use the following installation order.

DS8800

A disk enclosure (Gigapack) pair of the DS8800 holds 24 drives when using the 2.5 inch SFF drives, or 12 drives for the LFF HDDs. Each disk enclosure is installed in a specific physical location within a DS8000 frame and in a specific sequence. Disk enclosure pairs are installed in most cases bottom-up. The fastest drives of a configuration (usually a solid-state drive (SSD) drive set) are installed first. Then, the Enterprise HDDs are installed. Then, the nearline HDDs are installed. Figure 2-9 shows an exception with the 2-way models.

Figure 2-9 shows how each disk enclosure is associated with a certain DA pair and position in the DA loop. When ordering more drives, you can order more DA pairs for the DS8800 up to a maximum of eight DA pairs.

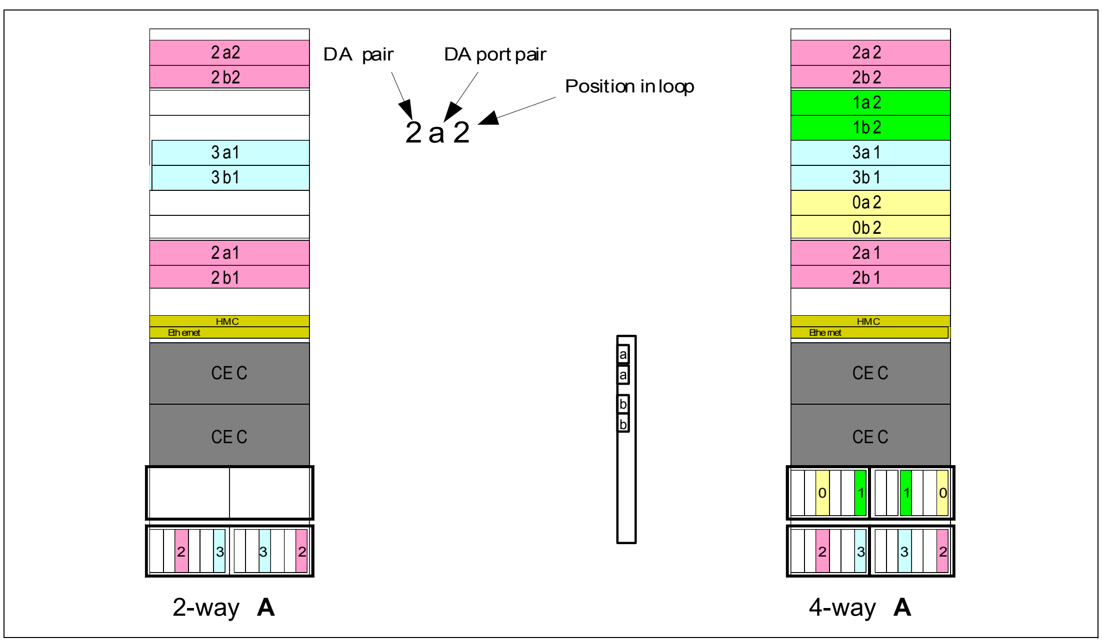


Figure 2-9 DS8800 A-Frame DA pair installation order for 2-way and 4-way Standard models

The 4-way Standard model can hold up to 1536 SFF drives with the B, C1, and C2 expansion frames. Figure 2-10 shows a DS8800 4-way Standard model that is fully equipped with 1536 SFF SAS drives.

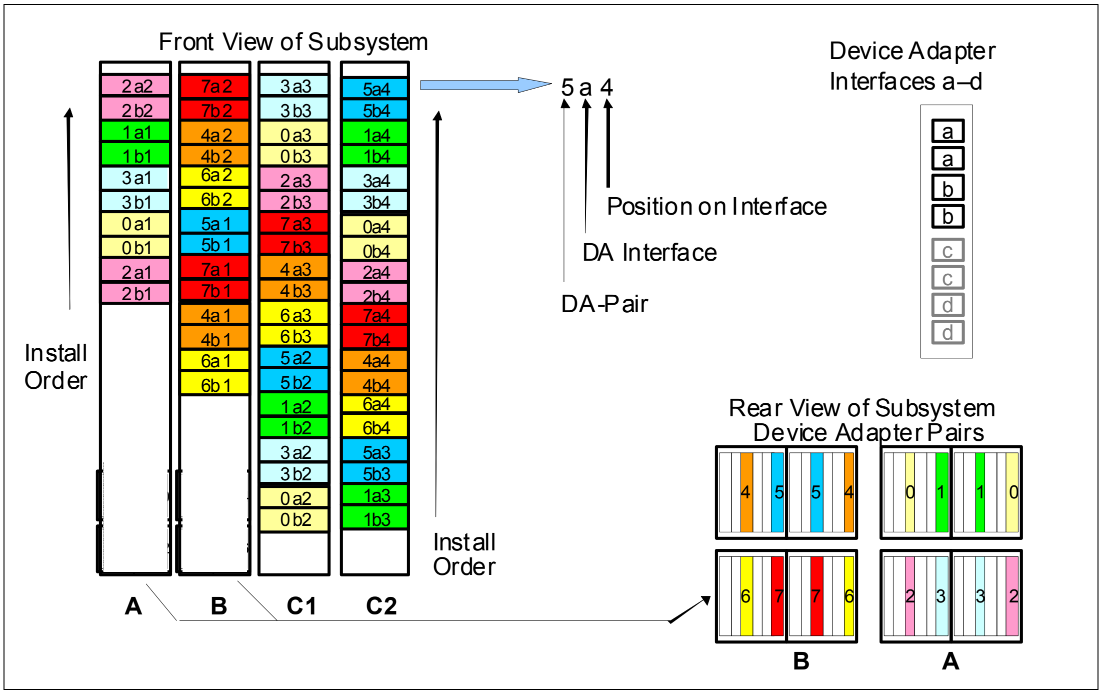


Figure 2-10 DS8800 4-frame DA pair installation order

The C and D interfaces of the DAs can potentially be used to attach more disk enclosures that are not used yet in the current DS8800 model.

DS8700

The DS8700 models can connect up to four expansion frames. The base frame and the fourth expansion frame hold 128 DDMs. The first, second, and third expansion frames hold 256 DDMs. In total, 1024 (3.5 inch) drives can be installed in a DS8700.

Figure 2-11 on page 44 shows which disk enclosures are associated with which DA pair. With more than 512 disks, each DA pair handles more than 64 DDMs (eight ranks). For usual workload profiles, this number is not a problem. It is rare to overload DAs. The performance of many storage systems is determined by the number of disk drives, as long as there are no HA bottlenecks.

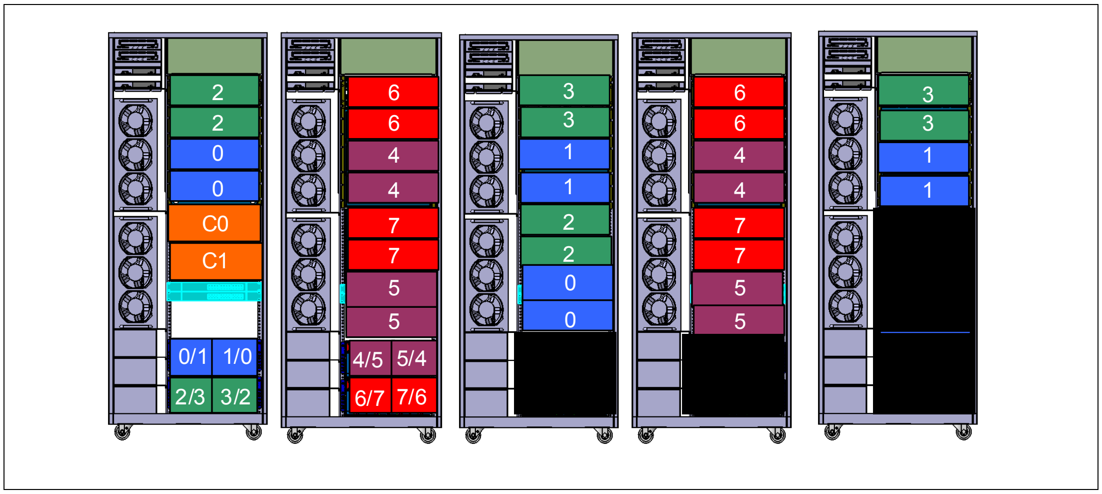


Figure 2-11   DS8700 DA pair installation order

2.4.9 The Business Class model for DS8800 only: Feature Code 1250

There are two models of the DS8800 2-way. For typical environments, the Enterprise Class or Standard model scales up to 144 disks in the A frame. For simpler environments without high performance requirements, the Business Class cabling model (FC1250) scales up to 240 drives with only two DA pairs. Standard cabling needs four DA pairs and the 4-frame model to fill all 240 drive slots of the A frame.

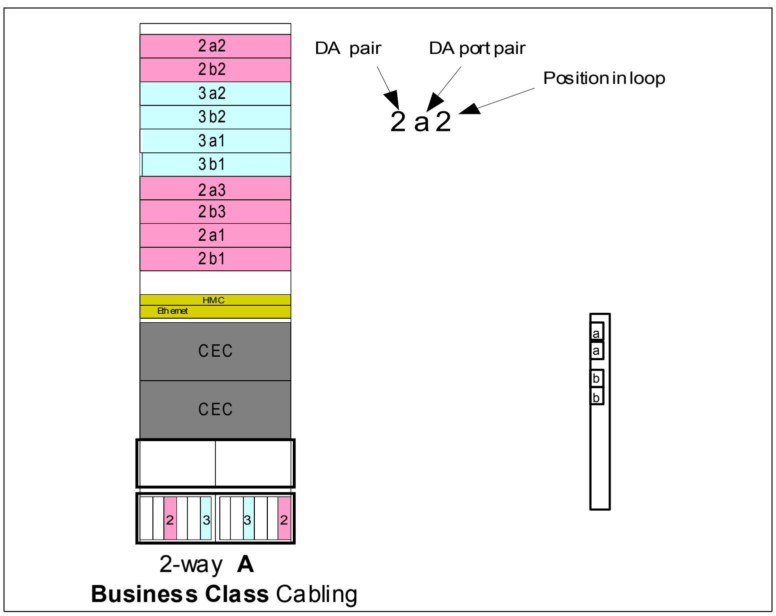


Figure 2-12   DS8800 2-way Business Class

The 2-way Business Class feature is designed for clients with lower performance requirements. For example, a client might replace a DS6800 that was attached to a System z host, but still need a DS8800 for attachment reasons. These clients can save on the number of processors, HAs, and DAs, and still have 240 possible disk drives available. However, the Business Class model is practically limited to the A frame only.

When you plan to potentially expand the DS8800 to more than 240 drives, want to maintain maximum flexibility, or have higher performance requirements, order the Standard models (Enterprise cabling). Only the Standard models can be expanded up to the maximum number of frames, drives, processors, or HAs.

2.4.10 Performance Accelerator feature for DS8700 only: FC1980

By default, the DS8700 comes with a new pair of DAs for each 64 DDMs. So, each DA pair serves a minimum of eight ranks. If you order a system with 128 drives, you get two DA pairs. When ordering 512 disk drives, you get the maximum of eight DA pairs. Certain sequential workloads require many DA pairs to achieve a higher throughput level. For example, a data warehouse installation requires a throughput of 1 GB/s or more.

In your environment, perhaps your sequential throughput requirements are high, but your capacity requirements are low. For instance, you might have capacity requirements for 256 disks only, but you still want the full sequential throughput potential of all DAs. For these situations, IBM offers the Performance Accelerator feature (FC1980). This feature provides one new DA pair for each 32 DDMs. This feature is offered for DS8700 941 models that have one base frame and one expansion frame. For example, this feature provides six DA pairs with 192 disk drives. By using 256 drives, you can use the maximum of eight DA pairs.

Few clients needed this feature due to the high performance of the DAs without it.

2.4.11 Device adapter performance

Figure 2-13 shows the DA and DA pair sequential performance for the DS8800 DAs compared to the previous models. Figure 2-14 on page 46 shows the DA performance comparison for 4K random workloads.

You can see the high numbers that we achieved; it is difficult to overload one DA now. The internal architecture is improved between the DS8800 and earlier DS8000 models.

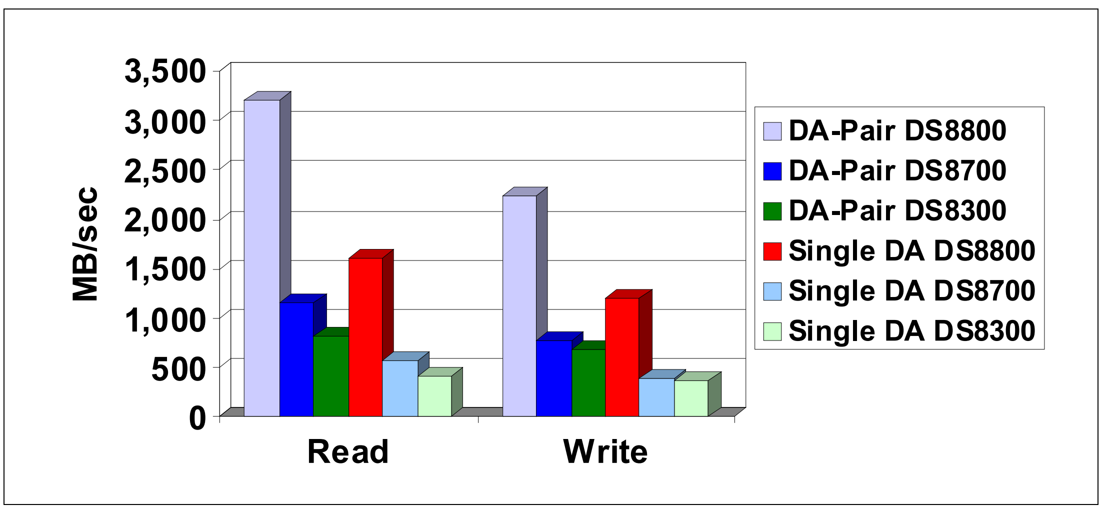


Figure 2-13 DA performance for large-block sequential load

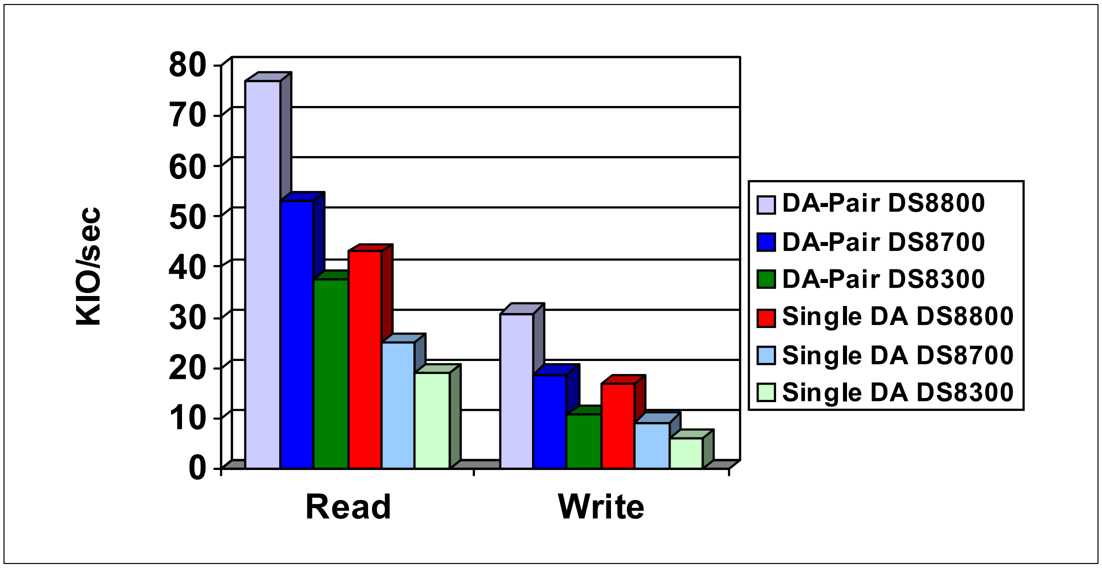


Figure 2-14 DA performances for 4K random workloads

Sample data: The sample performance measurement data shown in Figure 2-13 and Figure 2-14 is for comparative purposes only. We collected the data in a controlled laboratory environment at a specific point in time by using the configurations, hardware, and firmware levels available at that time. Current performance in real-world environments varies. The data is intended to help illustrate how different hardware technologies behave in relation to each other. Contact your IBM representative or IBM Business Partner if you have questions about the expected performance capability of the IBM products in your environment.

2.5 Host adapters

The DS8800 supports two types of host adapters (HAs): Fibre Channel/FICON 4-port and 8-port HA cards with a nominal port speed of 8 Gbps. HAs are installed in I/O enclosures. There is no affinity between the HA and the processor complex. Either processor complex can access any HA.

2.5.1 Fibre Channel and FICON host adapters

Fibre Channel (FC) is a technology standard that allows data to be transferred from one node to another node at high speeds and great distances. Using the long-wave fibre port HAs, connections are possible up to a 10 km (6.2 miles) distance. That distance is not user-configurable; it depends on the type of ordered HA and switch. All ports on each HA are the same type, either long wave or short wave. The DS8000 uses Fibre Channel Protocol (FCP) to transmit SCSI traffic inside FC frames. It also uses FC to transmit Fibre Channel connection (FICON) traffic, which uses FC frames to carry System z I/Os.

Each DS8800 FC adapter offers 4 Gbps or 8 Gbps FC ports. The cable connector that is required to attach to this adapter is an LC type. Each of the ports on a DS8800 HA can also independently be either FCP or FICON. The type of the port can be changed through the DS Storage Manager GUI or by using the DS8000 command-line interface (DSCLI) commands. A port cannot be both FICON and FCP simultaneously, but it can be changed as required.

The card is PCIe Gen 2. The card is driven by a new high-function, high-performance application-specific integrated circuit (ASIC). To ensure maximum data integrity, it supports metadata creation and checking. Each FC port supports a maximum of 509 host login IDs and 1280 paths so that you can create large storage area networks (SANs).

HAs and DAs essentially have the same architecture.

The front end with the 8 Gbps ports scales up to 128 ports for a DS8800, using the 8-port HAs, which results in a theoretical aggregated host I/O bandwidth of 128 x 8 Gbps.

The HA architecture includes the following improvements, which result in higher HA throughputs for the DS8800 compared to the DS8700 4-Gbps HAs:

- The architecture is fully on 8 Gbps.
- Due to the x8 Gen2 PCIe interface, no PCI-X-to-PCIe bridge carrier is needed.
- Adapter memory is increased fourfold.
- The single-core 1 GHz PowerPC processor (750 GX) is replaced by a dual-core 1.5 GHz (Freescale MPC8572).

The 8 Gbps adapter ports can each negotiate to 8, 4, or 2 Gbps speeds (not 1 Gbps). For attachments to 1 Gbps hosts, use a switch in between.

The 8-port HAs offer essentially the same total maximum throughput when taking loads of all its ports together as the 4-port HAs of the DS8800. Therefore, the 8-port HAs are meant for more attachment options, but not for more performance.

Compared to previous HA generations in DS8700 and DS8300, the DS8800 HAs have a much higher possible total aggregated throughput over all ports, apart from the differences in nominal port speed. More detailed HA measurements are in the *IBM System Storage DS8800 Performance Whitepaper*, WP102025.

Random small-block performance is usually no issue when considering HA performance, because this type of port can deliver up to 100K IOPS for a 4K (small-block) cache-hit workload. With several HAs, potential IOPS figures from the HAs are far higher than the number of disk drives that are currently supported in a DS8800 can deliver.

Automatic port queues

For I/O between a server and a DS8800 FC port, both the server HA and the DS8800 HBA support queuing I/Os. The length of this queue is the *queue depth.* Because several servers can communicate with few DS8800 posts, the queue depth of a storage HBA is larger than the queue depth on the server side. This difference is also true for the DS8800, which supports 2048 FC commands queued on a port. However, sometimes the port queue in the DS8800 HA can be flooded. When the number of commands sent to the DS8000 port exceeds the maximum number of commands that the port can queue, the port must discard these additional commands. This operation is a *normal* error recovery operation in the FCP to prevent more damage. The normal recovery is a 30-second timeout for the server, after which time the command is resent. The server has a *command retry* count before it fails the command. Command Timeout entries show in the server logs.

Automatic Port Queues is a mechanism the DS8800 uses to self-adjust the queue based on the workload. The Automatic Port Queues mechanism allows higher port queue oversubscription while maintaining a fair share for the servers and the accessed LUNs. The port that the queue fills goes into SCSI Queue Fill mode, where it accepts no additional commands to slow down the I/Os. By avoiding error recovery and the 30-second blocking SCSI Queue Full recovery interval, the overall performance is better with Automatic Port Queues.

2.5.2 Multiple paths to Open Systems servers

For high availability, each host system must use a multipathing device driver, such as Subsystem Device Driver (SDD). And, each host system must have a minimum of two host connections to HA cards in different I/O enclosures on the DS8800.

After you determine your throughput workload requirements, you must choose the appropriate number of connections to put between your open system hosts and the DS8000 to sustain this throughput. Use an appropriate number of HA cards to satisfy high throughput demands. The number of host connections per host system is primarily determined by the required bandwidth.

Host connections frequently go through various external connections between the server and the DS8000. Therefore, you need enough host connections for each server so that if half of the connections fail, processing can continue at the level before the failure. This availability-oriented approach requires that each connection carry only half the data traffic that it otherwise might carry. These multiple lightly loaded connections also help to minimize the instances when spikes in activity might cause bottlenecks at the HA or port. A multiple-path environment requires at least two connections. Four connections are typical, and eight connections are not unusual.

SAN switches and directors

Because many hosts, each using multiple paths, can be connected to the DS8000, it might not be feasible to directly connect each path to the DS8000. The solution is to use SAN switches or directors to switch logical connections from multiple hosts. Each adapter on the host server attaches to the SAN device, and then, the SAN device attaches to the DS8000. When using switches to connect multiple paths between servers and a DS8000, we suggest using two separate switches to avoid a single point of failure.

2.5.3 Multiple paths to System z servers

In the System z environment, the normal practice is to provide multiple paths from each host to a disk subsystem. Typically, four paths are installed. The channels in each host that can access each Logical Control Unit (LCU) in the DS8000 are defined in the hardware configuration definition (HCD) or I/O configuration dataset (IOCDS) for that host. Dynamic Path Selection (DPS) allows the channel subsystem to select any available (non-busy) path to initiate an operation to the disk subsystem. Dynamic Path Reconnect (DPR) allows the DS8000 to select any available path to a host to reconnect and resume a disconnected operation, for example, to transfer data after disconnection due to a cache miss. These functions are part of the System z architecture and are managed by the channel subsystem on the host and the DS8000.

In a System z environment, you need to select a SAN switch or director that also supports FICON. An availability-oriented approach applies to the System z environments similar to the Open Systems approach. Plan enough host connections for each server so that if half of the connections fail, processing can continue at the level before the failure.

2.5.4 Spreading host attachments

The DS8000 8 Gbps Fibre Channel/FICON HA cards can provide four or eight ports. A common question is how to distribute the server connections. Take the example of four host connections from each of four servers, all running a similar type workload, and four HAs. We suggest that you spread the host connections with each host attached to one port on each of four adapters. Now, consider the scenario in which the workloads differ. You probably want to isolate mission-critical workloads (for example, customer order processing) from workloads that are lower priority but I/O intensive (for example, data mining). Prevent the I/O-intensive workload from dominating the host adapter. If one of the four servers is running an I/O-intensive workload, we suggest that you acquire two additional host adapters and attach the four connections of the I/O intensive server to these adapters. Connect two host connections on each adapter. The other three servers remain attached to the original four adapters, one host connection per adapter.

We also offer the following general guidelines:

- For FC and FICON paths with a high utilization, do not use more than *four* ports on a DS8800 8 Gbps HA, which might increase the number of HAs required. Avoid using all eight ports of the 8-port HA cards for demanding workloads, because the performance of the HA card does not scale with more than four ports.
- Spread the paths from all host systems across the available I/O ports, HA cards, and I/O enclosures to optimize workload distribution across the available resources, depending on your workload sharing and isolation considerations.
- Spread the host paths that access the same set of volumes as evenly as possible across the available HA cards and I/O enclosures. This approach balances workload across hardware resources and helps to ensure that a hardware failure does not result in a loss of access.
- Ensure that each host system uses a multipathing device driver, such as SDD and a minimum of two host connections to different HA cards in different I/O enclosures on the DS8800. Preferably, evenly distribute them between left side (even-numbered) I/O enclosures and right side (odd-numbered) I/O enclosures for the highest availability and a balanced workload across I/O enclosures and HA cards.
- Do not use the same DS8000 I/O port for host attachment and Copy Services remote replication (such as Metro Mirror or Global Mirror).
- Consider using separate HA cards for FICON protocol and FCP. Even though I/O ports on the same HA can be configured independently for the FCP and the FICON protocol, it might be preferable to isolate your z/OS environment (FICON) from your Open Systems environment (FCP).

See 4.10.1, “I/O port planning considerations” on page 147 for additional guidelines.

2.6 Tools to aid in hardware planning

In this chapter, we discussed the hardware components of the DS8000. Each component is designed to provide high performance in its specific function and to mesh well with the other hardware components to provide a well-balanced, high performance storage system. There are a number of tools that can assist you in planning your specific hardware configuration.

2.6.1 White papers

IBM regularly publishes white papers that document the performance of specific DS8000 configurations. Typically, workloads are run on multiple configurations, and performance results are compiled so that you can compare the configurations. For example, workloads can be run using different DS8000 models, different numbers of host adapters, or different types of DDMs. By reviewing these white papers, you can make inferences about the relative performance benefits of different components to help you to choose the type and quantities of components to best fit your particular workload requirements. Your IBM representative or IBM Business Partner has access to these white papers and can provide them to you.

2.6.2 Disk Magic

A knowledge of the DS8000 hardware components helps you understand the device and its potential performance. However, we suggest Disk Magic to model your planned DS8000 hardware configuration to ensure that it can handle the required workload. Your IBM representative or IBM Business Partner has access to this tool and can run a Disk Magic study to configure a DS8000 based on your specific workloads. For additional information about the capabilities of this tool, see 6.1, "Disk Magic" on page 176. The tool can also be acquired by clients directly from IntelliMagic B.V. at this website:

http://www.intellimagic.net/intellimagic/products/disk-magic-for-ibm

2.6.3 Capacity Magic

Determining the usable capacity of a disk configuration is a complex task, dependent on DDM types, the RAID technique, and the logical volume types that are created. We suggest that you use the Capacity Magic tool to determine effective utilization. Your IBM representative or IBM Business Partner has access to this tool and can use it to validate that the planned physical disk configuration can provide enough effective capacity to meet your storage requirements.

3

Logical configuration concepts and terminology

This chapter summarizes the important concepts that need to be understood in preparation of the DS8000 logical configuration and about performance tuning. You can obtain more information about the DS8000 logical configuration concepts in the following IBM Redbooks publication: *IBM System Storage DS8000 Architecture and Implementation*, SG24-8886.

In this chapter, we review these topics:

- RAID levels and spares
- The abstraction layers for logical configuration
- Data placement on ranks and extent pools

3.1 RAID levels and spares

The arrays on the DS8000 are configured in either a RAID 5, RAID 6, or RAID 10 configuration. The following RAID configurations are supported:

- 6+P RAID 5 configuration: The data and parity information in this array is spread across seven drives. From a capacity perspective, this configuration represents the logical capacity of six data drives and one parity drive. The remaining drive on the array site is used as a spare.
- 7+P RAID 5 configuration: The data and parity information in this array is spread across eight drives. From a capacity perspective, this configuration represents the logical capacity of seven data drives and one parity drive.
- 5+P+Q RAID 6 configuration: The data and parity information in this array is spread across seven drives. From a capacity perspective, this configuration represents the logical capacity of five data drives and two parity drives. The remaining drive on the array site is used as a spare.
- 6+P+Q RAID 6 configuration: The data and parity information in this array is spread across eight drives. From a capacity perspective, this configuration represents the logical capacity of six data drives and two parity drives.
- 3+3 RAID 10 configuration: The array consists of three data drives that are mirrored to three copy drives. The two remaining drives on the array site are used as spares.
- 4+4 RAID 10 configuration: The array consists of four data drives that are mirrored to four copy drives.

For a list of drive combinations and RAID configurations, see 8.5.2, “Disk capacity,” in *IBM System Storage DS8000 Architecture and Implementation*, SG24-8886.

3.1.1 RAID 5 overview

The DS8000 series supports RAID 5 arrays. RAID 5 is a method of spreading volume data plus parity data across multiple disk drives. RAID 5 provides faster performance by striping data across a defined set of disk drive modules (DDMs). Data protection is provided by the generation of parity information for every stripe of data. If an array member fails, its contents can be regenerated by using the parity data.

RAID 5 implementation in the DS8000

In a DS8000, a RAID 5 array built on one array site contains either seven or eight disks, depending on whether the array site is supplying a spare. A seven-disk array effectively uses one disk for parity, so it is called a 6+P array (where P stands for parity). The reason only seven disks are available to a 6+P array is that the eighth disk in the array site used to build the array is used as a spare. This array is called a 6+P+S array site (where S stands for spare). An 8-disk array also effectively uses one disk for parity, so it is called a 7+P array.

Drive failure with RAID 5

When a DDM fails in a RAID 5 array, the device adapter (DA) starts an operation to reconstruct the data that was on the failed drive onto one of the spare drives. The spare that is used is chosen based on a smart algorithm that looks at the location of the spares and the size, speed, and location of the failed DDM. The DA rebuilds the data by reading the corresponding data and parity in each stripe from the remaining drives in the array, performing an exclusive-OR operation to re-create the data, and then writing this data to the spare drive.

While this data reconstruction occurs, the DA can still service read and write requests to the array from the hosts. There might be performance degradation while the sparing operation is in progress, because the DA and switched network resources are used to reconstruct the data. Due to the switch-based architecture, this effect is minimal. Additionally, any read requests for data on the failed drive require data to be read from the other drives in the array. Then, the DA reconstructs the data.

Performance of the RAID 5 array returns to normal when the data reconstruction onto the spare device completes. The time taken for sparing can vary, depending on the size of the failed DDM and the workload on the array, the switched network, and the DA. The use of arrays across loops (AAL) both speeds up rebuild time and decreases the impact of a rebuild.

Smart Rebuild, introduced with the DS8000 Licensed Machine Code (LMC) R6.2, further reduces the risk of a second drive failure for RAID 5 ranks during rebuild by detecting a failing drive early and copying the drive data to the spare drive in advance. If a RAID 5 array is predicted to fail, a "rebuild" is initiated by copying off that failing drive to the spare drive before it fails, decreasing the overall rebuild time. If the drive fails during the copy operation, the rebuild continues from the parity information like a regular rebuild.

3.1.2 RAID 6 overview

The DS8000 supports RAID 6 protection. RAID 6 presents an efficient method of data protection in double disk errors, such as two drive failures, two coincident medium errors, or a drive failure and a medium error. RAID 6 protection provides more fault tolerance than RAID 5 in disk failures and uses less raw disk capacity than RAID 10. RAID 6 allows for additional fault tolerance by using a second independent distributed parity scheme (*dual parity*). Data is striped on a block level across a set of drives, similar to RAID 5 configurations, and a second set of parity is calculated and written across all the drives.

RAID 6 is best used in combination with large capacity disk drives, for example, such as 2 TB and 3 TB nearline drives, because of longer rebuild times and the increased risk of an additional medium error in addition to the failed drive during the rebuild. In many environments today, RAID 6 is also considered already for 600 GB (and above) Enterprise drives in cases where reliability is favored over performance and the trade-off in performance versus reliability can be accepted. However, with the *Smart Rebuild* capability that was introduced with the DS8000 LMC R6.2, the risk of a second drive failure for RAID 5 ranks is also further reduced.

Comparing RAID 6 to RAID 5 performance provides about the same results on reads. For random writes, the throughput of a RAID 6 array is only two thirds of a RAID 5 array because of the additional parity handling. Workload planning is important before implementing RAID 6, specifically for write-intensive applications, including Copy Services targets and FlashCopy SE repositories where they are not generally recommended to be used. Yet, when properly sized for the I/O demand, RAID 6 is a considerable reliability enhancement.

RAID 6 implementation in the DS8000

A RAID 6 array from one array site of a DS8000 can be built on either seven or eight disks:

- In a seven disk array, two disks are always used for parity, while the eighth disk of the array site is needed as a spare. This RAID 6 array is called as a *5+P+Q+S array*, where P and Q stand for parity and S stands for spare.
- A RAID 6 array, consisting of eight disks, is built when all necessary spare drives are available. An eight disk RAID 6 array also always uses two disks for parity, so it is called a *6+P+Q array*.

Drive failure with RAID 6

When a DDM fails in a RAID 6 array, the DA starts to reconstruct the data of the failing drive onto one of the available spare drives. A smart algorithm determines the location of the spare drive to be used, depending on the size, speed, and location of the failed DDM. After the spare drive replaces a failed one in a redundant array, the entire contents of the new drive are recalculated by reading the corresponding data and parity in each stripe from the remaining drives in the array and then writing this data to the spare drive.

During the rebuild of the data on the new drive, the DA can still handle I/O requests of the connected hosts to the affected array. A performance degradation can occur during the reconstruction, because DAs and back-end resources are involved in the rebuild. Additionally, any read requests for data on the failed drive require data to be read from the other drives in the array, and then the DA reconstructs the data. Any later failure during the reconstruction within the same array (second drive failure, second coincident medium errors, or a drive failure and a medium error) can be recovered without loss of data.

Performance of the RAID 6 array returns to normal when the data reconstruction, on the spare device, completes. The rebuild time varies, depending on the size of the failed DDM and the workload on the array and the DA. The completion time is comparable to a RAID 5 rebuild, but slower than rebuilding a RAID 10 array in a single drive failure.

3.1.3 RAID 10 overview

RAID 10 provides high availability by combining features of RAID 0 and RAID 1. RAID 0 optimizes performance by striping volume data across multiple disk drives at a time. RAID 1 provides disk mirroring, which duplicates data between two disk drives. By combining the features of RAID 0 and RAID 1, RAID 10 provides a second optimization for fault tolerance. Data is striped across half of the disk drives in the RAID 1 array. The same data is also striped across the other half of the array, creating a mirror. Access to data is preserved if one disk in each mirrored pair remains available. RAID 10 offers faster random write operations than RAID 5, because it does not need to manage parity. However, with half of the DDMs in the group used for data and the other half to mirror that data, RAID 10 disk groups have less capacity than RAID 5 disk groups.

RAID 10 is not as commonly used as RAID 5, mainly because more raw disk capacity is required for every gigabyte of effective capacity. Typically, RAID 10 is used for workloads with a high random write ratio.

RAID 10 implementation in the DS8000

In the DS8000, the RAID 10 implementation is achieved by using either six or eight DDMs. If spares exist on the array site, six DDMs are used to make a three-disk RAID 0 array, which is then mirrored. If spares do not exist on the array site, eight DDMs are used to make a four-disk RAID 0 array, which is then mirrored.

Drive failure with RAID 10

When a DDM fails in a RAID 10 array, the DA starts an operation to reconstruct the data from the failed drive onto one of the hot spare drives. The spare that is used is chosen based on a smart algorithm that looks at the location of the spares and the size and location of the failed DDM. A RAID 10 array is effectively a RAID 0 array that is mirrored. Thus, when a drive fails in one of the RAID 0 arrays, we can rebuild the failed drive by reading the data from the equivalent drive in the other RAID 0 array.

While this data reconstruction occurs, the DA can still service read and write requests to the array from the hosts. There might be degradation in performance while the sparing operation is in progress, because DA and switched network resources are used to reconstruct the data.

Due to the switch-based architecture of the DS8000, this effect is minimal. Read requests for data on the failed drive typically are not affected, because they can all be directed to the good RAID 1 array.

Write operations are not affected. Performance of the RAID 10 array returns to normal when the data reconstruction, onto the spare device, completes. The time taken for sparing can vary, depending on the size of the failed DDM and the workload on the array and the DA.

In relation to RAID 5, RAID 10 sparing completion time is a little faster. Rebuilding a RAID 5 6+P configuration requires six reads plus one parity operation for each write. A RAID 10 3+3 configuration requires one read and one write (a direct copy).

3.1.4 Array across loops

One DA pair creates two switched loops:

- For DS8700, the front enclosure populates one loop, and the rear enclosure populates the other loop, in a disk enclosure pair. Each enclosure can hold up to 16 DDMs.
- For DS8800, the upper enclosure populates one loop, and the lower enclosure populates the other loop, in a disk enclosure pair. Each enclosure can hold up to 24 DDMs.

Each enclosure places two Fibre Channel (FC) switches onto each loop. FC/SAS DDMs are purchased in groups of 16 (drive set). Half of the new DDMs go into one disk enclosure, and half of the new DDMs go into the other disk enclosure of the pair. The same setup applies to SSD and NL SAS drives where we also have a half drive set purchase option with only eight DDMs.

An array site consists of eight DDMs. Four DDMs are taken from one enclosure in the disk enclosure pair, and four are taken from the other enclosure in the pair. Therefore, when a RAID array is created on the array site, half of the array is on each disk enclosure.

One disk enclosure of the pair is on one FC switched loop, and the other disk enclosure of the pair is on a second switched loop. The array, or AAL, is split across two loops.

AAL is used to increase performance. When the DA writes a stripe of data to a RAID 5 array, it sends half of the write to each switched loop. By splitting the workload in this manner, each loop is worked evenly. This setup aggregates the bandwidth of the two loops and improves performance. If RAID 10 is used, two RAID 0 arrays are created. Each loop hosts one RAID 0 array. When servicing read I/O, half of the reads can be sent to each loop, again improving performance by balancing workload across loops.

Array across loops and RAID 10

The DS8000 implements the concept of arrays across loops (AAL). With AAL, an array site is split into two halves. Half of the site is on the first disk loop of a DA pair, and the other half is on the second disk loop of that DA pair. AAL is implemented primarily to maximize performance, and it is used for all the RAID types in the DS8000. However, in RAID 10, we can take advantage of AAL to provide a higher level of redundancy. The DS8000 RAS code ensures that one RAID 0 array is maintained on each of the two loops created by a DA pair. Therefore, in the unlikely event of a complete loop outage, the DS8000 does not lose access to the RAID 10 array. While one RAID 0 array is offline, the other RAID 0 array remains available to service disk I/O.

3.1.5 Spare creation

When the arrays are created on a DS8000, the microcode determines which array sites contain spares. The first array sites on each DA pair that are assigned to arrays contribute one or two spares (depending on the RAID option), until the DA pair has access to at least four spares, with two spares placed on each loop.

A minimum of one spare is created for each array site assigned to an array until the following conditions are met:

- Minimum of four spares per DA pair
- Minimum of four spares for the largest capacity array site on the DA pair
- Minimum of two spares of capacity and rpm greater than or equal to the fastest array site of any capacity on the DA pair

Floating spares

The DS8000 implements a smart floating technique for spare DDMs. A floating spare is defined this way. When a DDM fails and the data it contained is rebuilt onto a spare, then when the disk is replaced, the replacement disk becomes the spare. The data is not migrated to another DDM, such as the DDM in the original position that the failed DDM occupied.

The DS8000 microcode takes this idea one step further. It might choose to allow the hot spare to remain where it is moved, but it can instead choose to migrate the spare to a more optimum position. This migration can better balance the spares across the DA pairs, the loops, and the disk enclosures. It might be preferable that a DDM that is currently in use as an array member is converted to a spare. In this case, the data on that DDM is migrated in the background onto an existing spare. This process does not fail the disk that is being migrated, although it reduces the number of available spares in the DS8000 until the migration process is complete.

The DS8000 uses this smart floating technique so that the larger or higher rpm DDMs are allocated as spares. Allocating the larger or higher rpm DDMs as spares ensures that a spare can provide at least the same capacity and performance as the replaced drive. If we rebuild the contents of a 450 GB DDM onto a 600 GB DDM, one-fourth of the 600 GB DDM is wasted, because that space is not needed. When the failed 450 GB DDM is replaced with a new 450 GB DDM, the DS8000 microcode most likely migrates the data back onto the recently replaced 450 GB DDM. When this process completes, the 450 GB DDM rejoins the array and the 600 GB DDM becomes the spare again.

Another example is if we fail a 146 GB 15K rpm DDM onto a 600 GB 10K rpm DDM. The data is now moved to a slower DDM and wastes significant space. The array has a mix of RPMs, which is not desirable. When the failed disk is replaced, the replacement is the same type as the failed 15K rpm disk. Again, a smart migration of the data is performed after suitable spares are available.

Hot pluggable DDMs

Replacement of a failed drive does not affect the operation of the DS8000, because the drives are fully hot pluggable. Each disk plugs into a switch, so there is no loop break associated with the removal or replacement of a disk. In addition, there is no potentially disruptive loop initialization process.

Overconfiguration of spares

The DDM sparing policies support the overconfiguration of spares. This possibility might be of interest to certain installations, because it allows the deferral of the repair of certain DDM failures until a later repair action is required.

Important: In general, when a drive fails either in RAID 10, 5, or 6, it might cause a minimal degradation of performance on DA and switched network resources during the sparing process. DS8800 and DS8700 architecture and features minimize the effect of this behavior.

By using the DSCLI Tool, it is possible to check for any failed drives when issuing the DSCLI **`lsddm -state not_normal`** command. See Example 3-1.

Example 3-1 Output of DSCLI lsddm command

```
dscli> lssi
Name   ID               Storage Unit     Model WWNN             State  ESSNet
==============================================================================
ATS_04 IBM.2107-75TV181 IBM.2107-75TV180 951   500507630AFFC29F Online Enabled

dscli> lsddm -state not_normal IBM.2107-75TV181
CMUC00234I lsddm: No DDM FRU found.
```

3.2 The abstraction layers for logical configuration

We describe the terminology and necessary steps to configure the logical volumes that can be accessed from attached hosts.

The definition of *virtualization* is the abstraction process from the physical disk drives to a logical volume that is presented to hosts and servers in a way that they see it as though it were a physical disk.

When talking about virtualization, we mean the process of preparing physical disk drives (DDMs) to become an entity that can be used by an operating system, which means we are talking about the creation of logical unit numbers (LUNs).

The DDMs are mounted in disk enclosures and connected in a switched FC topology, by using a Fibre Channel Arbitrated Loop (FC-AL) protocol. The DDM physical installation differs between DS8800 and DS8700:

- For the DS8700, DDMs are mounted in 16 DDM enclosures. You can order disk drives in groups of 8 or 16 drives of the same capacity and revolutions per minute (rpm). The options for 8-drive sets apply for the 600 GB solid-state drives (SSDs) and 2 TB nearline drives.
- The DS8800 small form factor disks are mounted in 24 DDM enclosures. Disk drives can be ordered in groups of eight or 16 drives of the same capacity and rpm. The option for 8-drive sets apply for the 300 GB SSDs and 3 TB nearline drives. The DS8800 now also supports a new high-density and lower-cost large form factor (LFF) storage enclosure. This enclosure accepts 3.5 inch drives, offering 12 drive slots. The previously introduced small form factor (SFF) enclosure offers twenty-four 2.5 inch drive slots. The appearance of the front of the LFF enclosure differs from the appearance of the front of the SFF enclosure, with its 12 drives that slot horizontally rather than vertically.

The disk drives can be accessed by a pair of DAs. Each DA has four paths to the disk drives. One device interface from each DA connects to a set of FC-AL devices so that either DA can access any disk drive through two independent switched fabrics (the DAs and switches are redundant).

Each DA has four ports, and because DAs operate in pairs, there are eight ports or paths to the disk drives. All eight paths can operate concurrently and can access all disk drives on the attached fabric. In normal operation, however, disk drives are typically accessed by one DA. Which DA owns the disk is defined during the logical configuration process to avoid any contention between the two DAs for access to the disks.

Because of the switching design, each drive is in close reach of the DA, and certain drives require a few more hops through the FC switch. Therefore, it is not a loop but a switched FC-AL loop with the FC-AL addressing schema, that is, Arbitrated Loop Physical Addressing (AL-PA).

3.2.1 Array sites

An *array site* is a group of eight DDMs. The DDMs that make up an array site are predetermined by the DS8000, but no predetermined processor complex affinity exists for array sites. The DDMs that are selected for an array site are chosen from two disk enclosures on different loops; see Figure 3-1. The DDMs in the array site are of the same DDM type, which means that they are the same capacity and speed (rpm).

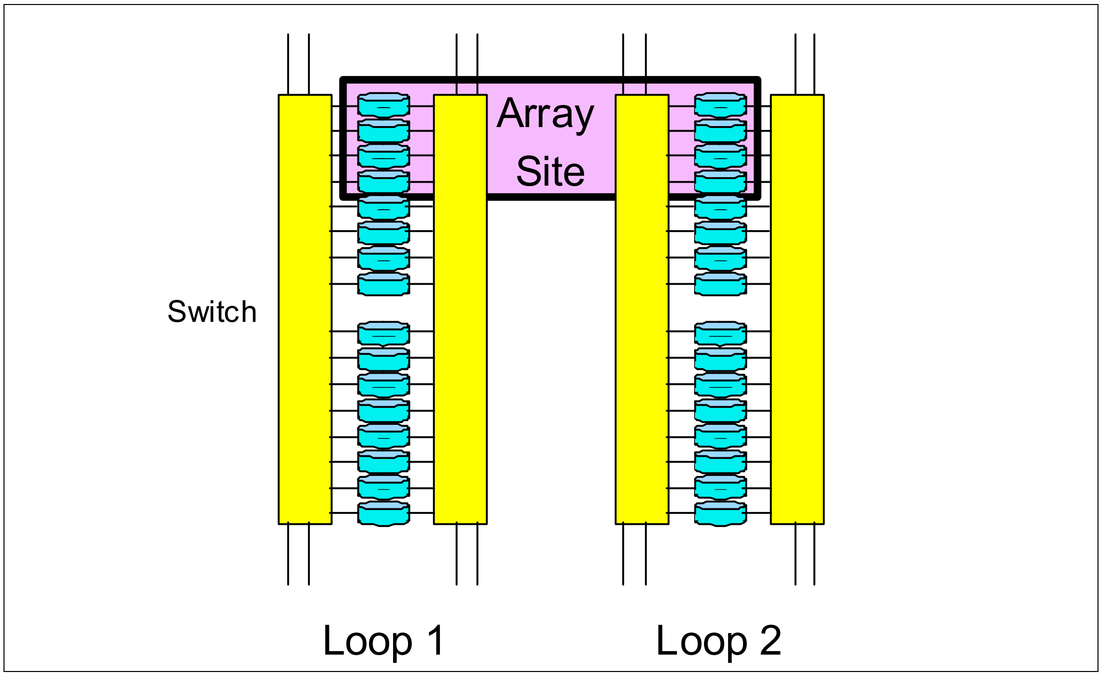


Figure 3-1 Array site

As you can see in Figure 3-1, array sites span loops. Four DDMs are taken from loop 1 and another four DDMs from loop 2. Array sites are the building blocks that are used to define arrays.

3.2.2 Arrays

An *array* is created from one array site. Forming an array means defining it as a specific RAID type. The supported RAID types are RAID 5, RAID 6, and RAID 10 (see 3.1, “RAID levels and spares” on page 52). For each array site, you can select a RAID type. The process of selecting the RAID type for an array is also called *defining* an array.

Important: In the DS8000 implementation, one array is defined by using one array site.

Figure 3-2 shows the creation of a RAID 5 array with one spare, which is also called a 6+P+S array (capacity of six DDMs for data, capacity of one DDM for parity, and a spare drive). According to the RAID 5 rules, parity is distributed across all seven drives in this example.

On the right side in Figure 3-2, the terms, D1, D2, D3, and so on, represent the set of data contained on one disk within a stripe on the array. If, for example, 1 GB of data is written, it is distributed across all disks of the array.

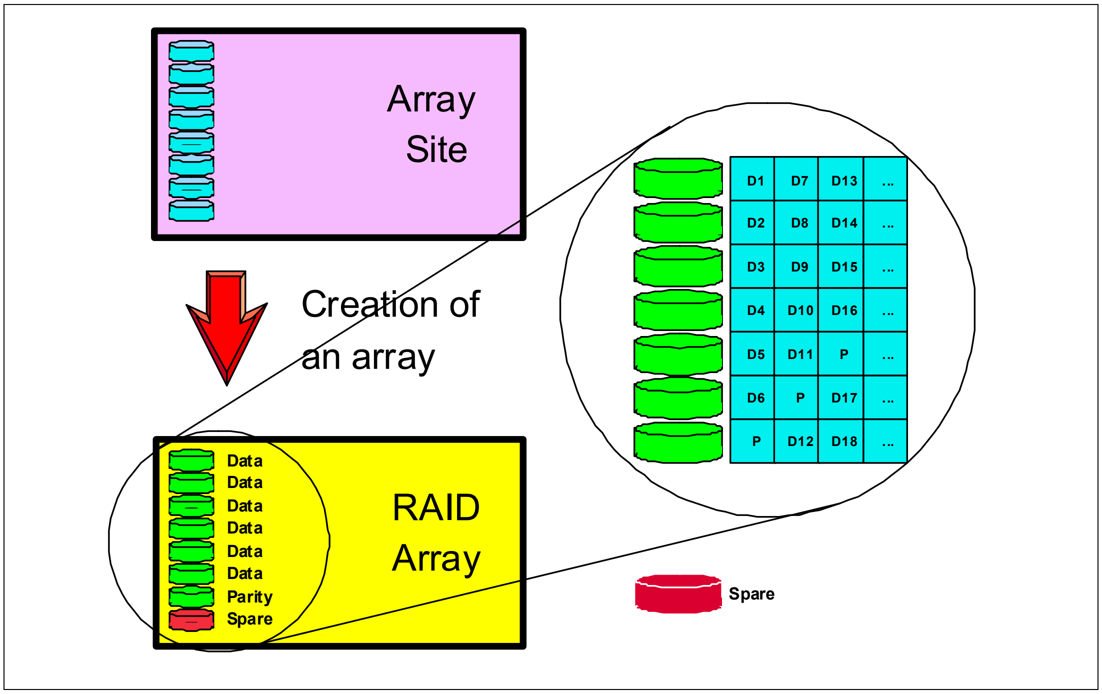


Figure 3-2 Creation of an array

So, an array is formed by using one array site, and while the array can be accessed by each adapter of the DA pair, it is managed by one DA. Later in the configuration process, you define the adapter and the server that manage this array.

3.2.3 Ranks

In the DS8000 virtualization hierarchy, there is another logical construct, a *rank*. When defining a new rank, its name is chosen by the DS Storage Manager, for example, R1, R2, and R3. You must add an array to a rank.

Important: In the DS8000 implementation, a rank is built by using only one array.

The available space on each rank is divided into extents. The *extents* are the building blocks of the logical volumes. An extent is striped across all disks of an array, as shown in Figure 3-3 on page 60, and indicated by the small squares in Figure 3-4 on page 62.

The process of forming a rank includes two jobs:

- The array is formatted for either fixed block (FB) type data (Open Systems) or count key data (CKD) (System z). This formatting determines the size of the set of data contained on one disk within a stripe on the array.
- The capacity of the array is subdivided into equal-sized partitions, which are called *extents*. The extent size depends on the *extent type*: FB or CKD.

An FB rank has an extent size of 1 GiB (where 1 GiB equals 2^{30} bytes).

IBM System z users or administrators typically do not deal with gigabytes or gibibytes, and instead they think of storage in the original 3390 volume sizes. A 3390 Model 3 is three times the size of a Model 1. A Model 1 has 1113 cylinders (about 0.94 GB). The extent size of a CKD rank is one 3390 Model 1 or 1113 cylinders.

Figure 3-3 shows an example of an array that is formatted for FB data with 1 GiB extents (the squares in the rank indicate that the extent is composed of several blocks from DDMs).

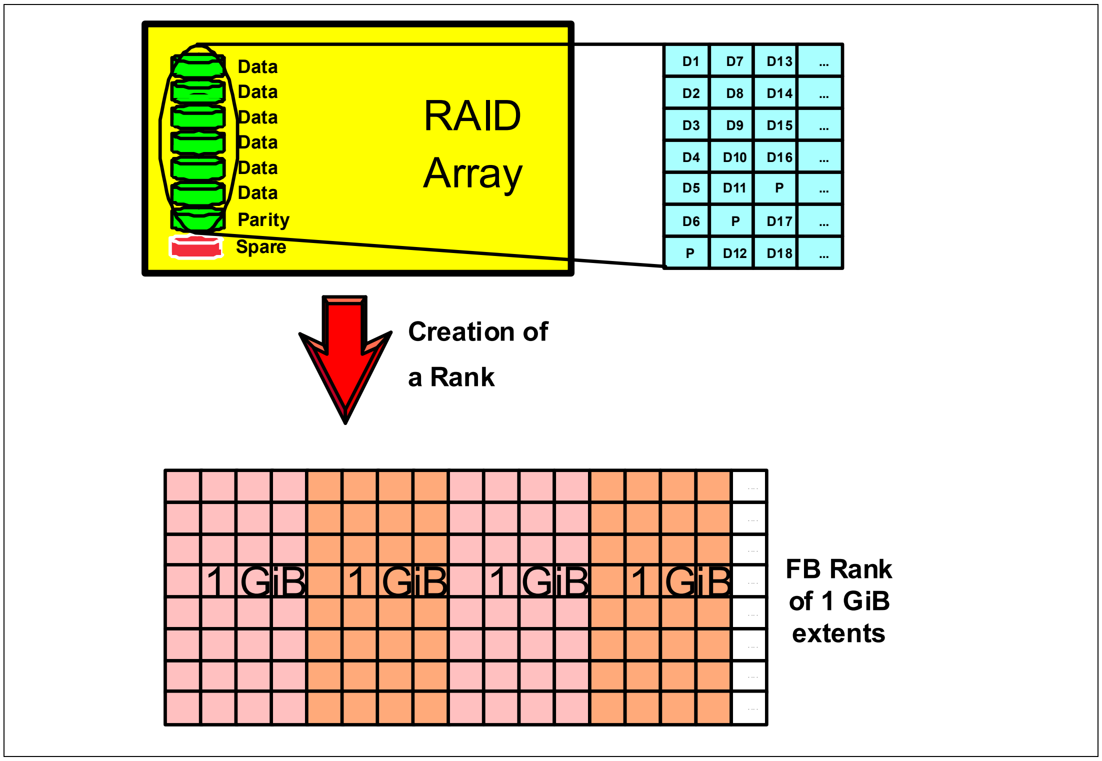


Figure 3-3 Forming an FB rank with 1 GiB extents

It is still possible to define a CKD volume with a capacity that is an integral multiple of one cylinder or a fixed block LUN with a capacity that is an integral multiple of 128 logical blocks (64 KB). However, if the defined capacity is not an integral multiple of the capacity of one extent, the unused capacity in the last extent is wasted. For example, you can define a one cylinder CKD volume, but 1113 cylinders (one extent) are allocated and 1112 cylinders are wasted.

Encryption group

A DS8000 series can be ordered with encryption-capable disk drives. If you plan to use encryption before creating a rank, you must define an *encryption group*. For more information, see *IBM System Storage DS8700 Disk Encryption Implementation and Usage Guidelines*, REDP-4500. Currently, the DS8000 series supports only one encryption group. All ranks must be in this encryption group. The encryption group is an attribute of a rank. So, your choice is to encrypt everything or nothing. You can switch on (create an encryption group) encryption later, but then all ranks must be deleted and re-created, which means that your data is also deleted.

3.2.4 Extent pools

An *extent pool* is a logical construct to aggregate the extents from a set of ranks to form a domain for extent allocation to a logical volume. Typically, the ranks in the extent pool have the same RAID type and the same disk rpm characteristics. This way, the extents in the extent pool have homogeneous characteristics if you configure extent pools that are not supposed to be automatically managed by IBM System Storage Easy Tier. Easy Tier was first introduced with LMC release R5.1 on DS8700 and R6.1 on DS8800.

With Easy Tier, it is possible to mix ranks with different characteristics and features in managed extent pools to achieve the best performance results. With R6.2, you can mix up to three storage classes or storage tiers within the same extent pool that feature SSD, Enterprise, and nearline-class disks.

> **Important:** In general, do not mix ranks with different RAID levels or disk types (size and RPMs) in the same extent pool if you are not implementing Easy Tier automatic management of these pools. Easy Tier has algorithms for automatically managing performance and data relocation across storage tiers and even rebalancing data within a storage tier across ranks in multi-tier or single-tier extent pools, providing automatic storage performance and storage economics management with best price, performance, and energy savings costs.

There is no predefined affinity of ranks or arrays to a storage server. The affinity of the rank (and its associated array) to a certain server is determined at the point that the rank is assigned to an extent pool.

One or more ranks *with the same extent type (FB or CKD)* can be assigned to an extent pool. One rank can be assigned to only one extent pool. There can be as many extent pools as there are ranks.

With *storage pool striping* (the extent allocation method (EAM) *rotate extents*), you can create logical volumes striped across multiple ranks. This approach typically enhances performance. To benefit from storage pool striping (see “Rotate extents (storage pool striping) extent allocation method” on page 68), multiple ranks in an extent pool are required.

Storage pool striping enhances performance, but it also increases the failure boundary. When one rank is lost, for example, in the unlikely event that a whole RAID array failed due to multiple failures at the same time, the data of this single rank is lost. Also, all volumes in the pool that are allocated with the *rotate extents* option are exposed to data loss. To avoid exposure to data loss for this event, consider mirroring your data to a remote DS8000.

When an extent pool is defined, it must be assigned with the following attributes:

- Server affinity (or rank group)
- Storage type (either FB or CKD)
- Encryption group

Just like ranks, extent pools also belong to an encryption group. When defining an extent pool, you must specify an encryption group. Encryption group 0 means no encryption. Encryption group 1 means encryption. Currently, the DS8000 series supports only one encryption group, and encryption is on for all extent pools or off for all extent pools.

The minimum reasonable number of extent pools on a DS8000 is *two*. One extent pool is assigned to storage server 0 (rank group 0), and the other extent pool is assigned to storage server 1 (rank group 1) so that both DS8000 storage servers are active. In an environment where FB storage and CKD storage share the DS8000 storage system, four extent pools are required to assign each pair of FB pools and CKD pools to both storage servers, balancing capacity and workload across both DS8000 processor complexes. Figure 3-4 is an example of a mixed environment with CKD and FB extent pools. Additional extent pools might be desirable to segregate workloads.

Extent pools are expanded by adding more ranks to the pool. Ranks are organized in two *rank groups*: Rank group 0 is controlled by storage server 0 (processor complex 0), and rank group 1 is controlled by storage server 1 (processor complex 1).

Important: For the best performance, balance capacity between the two servers and create at least two extent pools, with one extent pool per DS8000 storage server.

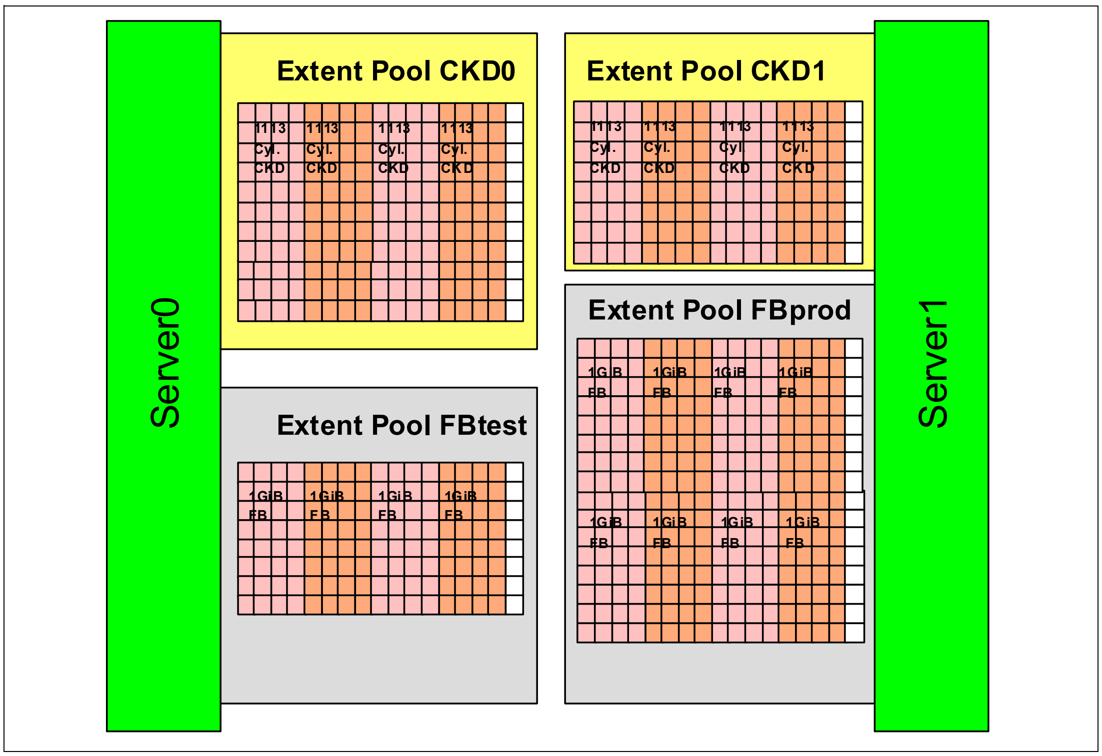


Figure 3-4 Extent pools

Dynamic extent pool merge

Dynamic extent pool merge is a capability that is provided by the Easy Tier manual mode feature.

Dynamic extent pool merge allows one extent pool to be merged into another extent pool while the logical volumes in both extent pools remain accessible to the host servers. Dynamic extent pool merge can be used for the following scenarios:

- Use dynamic extent pool merge for the consolidation of smaller extent pools of the same storage type into a larger homogeneous extent pool that uses *storage pool striping*. Creating a larger extent pool allows logical volumes to be distributed evenly over a larger number of ranks, which improves overall performance by minimizing skew and reducing the risk of a single rank that becomes a hot spot. In this case, a *manual volume rebalance* needs to be initiated to restripe all existing volumes evenly across all available ranks in the new pool. Newly created volumes in the merged extent pool allocate capacity automatically across all available ranks by using the *rotate extents* EAM (storage pool striping), which is the default.

- Use dynamic extent pool merge for consolidating extent pools with different storage tiers to create a merged multi-tier extent pool with a mix of storage classes (SSD, Enterprise disk, and nearline disk) for automated management by Easy Tier automatic mode.
- Or, under certain circumstances, use dynamic extent pool merge for consolidating extent pools of the same storage tier but different drive types or RAID levels that can eventually benefit from storage pool striping and Easy Tier automatic mode intra-tier management (auto-rebalance), using the Easy Tier micro-tiering capabilities.

Important: You can only apply dynamic extent pool merge among extent pools associated with the same DS8000 storage server affinity (storage server 0 or storage server 1) or rank group. All even-numbered extent pools (P0, P2, P4, and so on) belong to rank group 0 and are serviced by storage server 0. All odd-numbered extent pools (P1, P3, P5, and so on) belong to rank group 1 and are serviced by storage server 1 (unless one DS8000 storage server failed or is quiesced with a failover to the alternate storage server).

Additionally, the dynamic extent pool merge is not supported in these situations:

- If source and target pools have different storage types (FB and CKD)
- If both extent pools contain track space-efficient (TSE) volumes
- If there are TSE volumes on the SSD ranks
- If you select an extent pool that contains volumes that are being migrated
- If the combined extent pools have 2 PB or more of extent space-efficient (ESE) volume logical capacity

For more information about dynamic extent pool merge, see *IBM System Storage DS8000 Easy Tier,* REDP-4667.

General considerations when adding capacity to an extent pool

If you must add capacity to an extent pool that is almost fully allocated on a DS8000 system without the Easy Tier feature, we suggest that you add several ranks to this pool at the same time, not just one. With this approach, the new volumes can be striped across the newly added ranks and reduce skew and hot spots.

With the Easy Tier feature, you can easily add capacity and even single ranks to existing extent pools without concern about performance.

Manual volume rebalance

When adding new ranks to an existing non-managed, single-tier extent pool, all existing volumes on the previous ranks leave the new capacity unused. Restriping the volumes evenly across all ranks in the pool improves performance and reduces workload skew and rank hot spots. If a single-tier pool is not managed by Easy Tier automatic mode, manual volume rebalance provides a way to restripe the volumes evenly across all ranks in the non-managed extent pool.

Manual volume rebalance is a feature of the Easy Tier manual mode and is only available in non-managed, single-tier (homogeneous) extent pools. It allows a balanced redistribution of the extents of a volume across all ranks in the pool. This feature is not available in managed or hybrid pools where Easy Tier is supposed to manage the placement of the extents automatically on the ranks in the pool based on their actual workload pattern, available storage tiers, and rank utilizations.

Manual volume rebalance is designed to redistribute the extents of volumes within a non-managed, single-tier (homogeneous) pool so that workload skew and hot spots are less likely to occur on the ranks. Manual volume rebalance is an enhancement of the dynamic volume relocation feature introduced with the DS8000 R6.1 LMC. It is initiated if a volume is migrated back into the same extent pool with the source extent pool as the target extent pool and the EAM as *rotate extents* (which uses storage pool striping). In this case, the algorithm tries to evenly spread the extents of the volume across all ranks in the extent pool. During extent relocation (manual volume rebalance), only one extent at a time is allocated rather than pre-allocating the full volume. Only a minimum amount of free capacity is required in the extent pool, and only the required number of extents is relocated.

This feature is useful for redistributing extents after adding new ranks to a non-managed extent pool. Also, this feature is useful after merging homogeneous extent pools to balance the capacity and the workload of the volumes across all available ranks in a pool.

Manual volume rebalance provides manual performance optimization by rebalancing the capacity of a volume within a non-managed homogeneous extent pool. It is also called *capacity rebalance*, because it balances the extents of the volume across the ranks in an extent pool without considering any workload statistics or device usage. A balanced distribution of the extents of a volume across all available ranks in a pool is supposed to provide a balanced workload distribution and minimize skew and hot spots. It is also called *volume restripin*g, and it is supported for standard and ESE thin-provisioned volumes.

For further information about manual volume rebalance, see *IBM System Storage DS8000 Easy Tier*, REDP-4667.

Auto-rebalance

With Easy Tier automatic mode enabled for single-tier or multi-tier extent pools, for the DS8000 R6.2 LMC or higher, you can benefit from Easy Tier automated intra-tier performance management (*auto-rebalance*), which relocates extents based on rank utilization, and reduces skew and avoids rank hot spots. Easy Tier relocates subvolume data on extent level based on actual workload pattern and rank utilization (*workload rebalance*) rather than balance the capacity of a volume across all ranks in the pool (*capacity rebalance* as achieved with manual volume rebalance).

When adding capacity to *managed* pools, Easy Tier automatic mode performance management, *auto-rebalance*, takes advantage of the new ranks and automatically populates the new ranks that are added to the pool when rebalancing the workload within a storage tier and relocating subvolume data. Starting with the DS8000 R6.2 LMC, auto-rebalance can be enabled for hybrid and homogeneous extent pools.

For more information about auto-rebalance, see *IBM System Storage DS8000 Easy Tier*, REDP-4667.

3.2.5 Logical volumes

A *logical volume* is composed of a set of extents from one extent pool. On a DS8000, up to 65,280 (we use the abbreviation 64 K in this discussion, even though it is 65,536 – 256, which is not quite 64 K in binary) volumes can be created (either 64 K CKD, or 64 K FB volumes, or a mixture of both types with a maximum of 64 K volumes in total).

Fixed block LUNs

A logical volume composed of fixed block extents is called a *logical unit number (LUN)*. A fixed block LUN is composed of one or more 1 GiB (2^{30} bytes) extents from one FB extent pool. A LUN cannot span multiple extent pools, but a LUN can have extents from different ranks within the same extent pool. You can construct LUNs up to a size of 16 TiB (16×2^{40} bytes, or 2^{44} bytes).

> **Important:** There is currently no Copy Services support for logical volumes larger than 2 TiB (2×2^{40} bytes). Do not create LUNs larger than 2 TiB if you want to use the DS8000 Copy Services for those LUNs, unless you want to use it as managed disks in an IBM SAN Volume Controller (SVC) with release 6.2 (or higher), and use SVC Copy Services instead.

LUNs can be allocated in binary GiB (2^{30} bytes), decimal GB (10^9 bytes), or 512 or 520 byte blocks. However, the physical capacity that is allocated for a LUN always is a multiple of 1 GiB, so it is a good idea to have LUN sizes that are a multiple of a gibibyte. If you define a LUN with a LUN size that is not a multiple of 1 GiB, for example, 25.5 GiB, the LUN size is 25.5 GiB, but a capacity of 26 GiB is physically allocated, wasting 0.5 GiB of physical storage capacity.

CKD volumes

A System z CKD volume is composed of one or more extents from one CKD extent pool. CKD extents are the size of 3390 Model 1, which has 1113 cylinders. However, when you define a System z CKD volume, you do not specify the number of 3390 Model 1 extents but the number of cylinders that you want for the volume.

The maximum size for a CKD volume was originally 65,520 cylinders. Later, the LMC 5.4.0.*xx.xx* introduced the *Extended Address Volume (EAV)* that can grow up to 262668 cylinders (about 223 GB). Starting with microcode R6.2, this limit is raised to 1182006 cylinders (EAV II). Currently, it is possible to create CKD volumes up to 935 GB. For more information about EAV volumes, see *DS8000 Series: Architecture and Implementation*, SG24-8886.

> **Important:** EAV volumes are supported by z/OS Version 1.10 or later. EAV II volumes are supported by z/OS Version 1.12 or later.

If the number of specified cylinders is not an exact multiple of 1113 cylinders, part of the space in the last allocated extent is wasted. For example, if you define 1114 or 3340 cylinders, 1112 cylinders are wasted. For maximum storage efficiency, consider allocating volumes that are exact multiples of 1113 cylinders. In fact, consider multiples of 3339 cylinders for future compatibility.

A CKD volume cannot span multiple extent pools, but a volume can have extents from different ranks in the same extent pool. You can stripe a volume across all ranks in an extent pool by using the *rotate extents* EAM to distribute the capacity of the volume. This EAM distributes the workload evenly across all ranks in the pool, minimizes skew, and reduces the risk of single extent pools that become a hot spot. For more information, see “Rotate extents (storage pool striping) extent allocation method” on page 68.

IBM i LUNs

IBM i LUNs are also composed of fixed block 1 GiB extents. There are, however, special aspects with System i LUNs. LUNs created on a DS8000 are always RAID-protected based on RAID 5, RAID 6, or RAID 10 arrays. However, you might want to deceive IBM i and tell it that the LUN is *not* RAID-protected. Then, the IBM i performs its own mirroring. System i LUNs can have the attribute *unprotected*; in which case, the DS8000 reports that the LUN is not RAID-protected.

IBM i supports certain fixed volume sizes, for example, model sizes of 8.5 GB, 17.5 GB, and 35.1 GB. These sizes are not multiples of 1 GiB, and depending on the model chosen, more or less physical storage capacity is wasted. IBM i LUNs expose a 520-byte block to the host. The operating system uses eight of these bytes, so the usable space is still 512 bytes like other Small Computer System Interface (SCSI) LUNs. The capacities quoted for the IBM i LUNs are the 512-byte block capacity and are expressed in GB (10^9). These capacities must be converted to GiB (2^{30}) when considering effective usage of extents that are 1 GiB (2^{30}). For more information, see *IBM System Storage DS8000: Host Attachment and Interoperability,* SG24-8887.

3.2.6 Space-efficient volumes

When a regular (non-thin-provisioned) FB LUN or CKD volume is created on the DS8000 system, it occupies as much physical capacity as is specified by its capacity.

For the DS8800 with LMC 7.6.1.*xx* or the DS8700 with LMC 6.5.1.*xx*, thin-provisioned volumes are supported. Two types of space-efficient volumes are available: extent space-efficient (ESE) volumes and track space-efficient (TSE) volumes. These concepts are described in detail in *DS8000 Thin Provisioning*, REDP-4554.

A space-efficient volume does not occupy all of its physical capacity at one time when it is created. The space gets allocated when data is written to the volume. The sum of all the virtual capacity of all space-efficient volumes can be larger than the available physical capacity, which is known as over-provisioning or thin provisioning.

TSE volumes on the DS8000 require the IBM FlashCopy SE feature (licensing is required). TSE volumes were introduced in the DS8000 family with the DS8000 LMC Release 3.0 (LMC 5.3*x.xx*). TSE volumes were designed to support the IBM FlashCopy SE function. The allocation of real capacity for TSE logical volumes is from a repository volume, which is created per extent pool. To create a repository volume, the amount of real capacity that is required must be supplied. In addition, the repository also requires a virtual capacity to provide a pool for the TSE logical volumes. The ratio of the virtual capacity to the real capacity represents the storage over-commitment.

Extent space-efficient (ESE) volumes on the DS8000 require the IBM System Storage DS8000 Thin Provisioning feature (licensing is required). The ESE volumes are thin-provisioned volumes designated for standard host access. The virtual capacity pool for ESE volumes is created per extent pool. There is no dedicated repository with physical capacity required for ESE volumes as for TSE volumes. The available physical capacity pool for allocation is the unused physical capacity in the extent pool. When an ESE logical volume is created, the volume is not allocated physical data capacity. However, the DS8000 allocates capacity for metadata that it uses to manage space allocation. Additional physical capacity is allocated when writes to an unallocated address occur. Before the write, a new extent is dynamically allocated, initialized, and eventually assigned to the volume (QuickInit).

The idea behind space-efficient volumes is to allocate physical storage at the time that it is needed to satisfy temporary peak storage needs.

Important: No Copy Services support exists for logical volumes larger than 2 TiB (2×2^{40} bytes). Thin-provisioned volumes (ESE) as released with the DS8000 LMC R6.2 are fully supported by FlashCopy but are not yet supported with CKD volumes and IBM i volumes. Thin-provisioned volumes are not fully supported with all DS8000 Copy Services or advanced functions yet. These restrictions might change with future DS8000 LMC releases, so check the related documentation for your DS8000 LMC release.

Repository for track space-efficient volumes (TSE)

The definition of track space-efficient (TSE) volumes begins at the extent pool level. TSE volumes are defined from virtual space. The size of the TSE volume does not initially use physical storage. However, any data written to a TSE volume must have enough physical storage to contain this written data. This physical storage is provided by the repository.

> **Tip:** Serial Advanced Technology Attachment (SATA) and nearline SAS drives are not generally recommended to be used for space-efficient FlashCopy SE repository data because of performance considerations.

The *repository* is an object within an extent pool. It is similar to a volume within the extent pool. The repository has a physical size and a virtual size. The physical size of the repository is the amount of space that is allocated in the extent pool. It is the physical space that is available for all TSE logical volumes, in total, in this extent pool. The repository is striped across all ranks within the extent pool. There is one repository only per extent pool.

> **Important:** The size of the repository and the virtual space that it uses are part of the extent pool definition. Each extent pool can have a TSE volume repository, but this physical space cannot be shared across extent pools. Virtual space in an extent pool is used for both TSE and ESE volumes. The repository is only used for TSE volumes for FlashCopy SE. ESE volumes use available extents in the extent pool in a similar fashion as standard, fully provisioned volumes as far as their specified extent allocation method, but extents are only allocated as needed when data is written to the ESE volume.

The logical size of the repository is limited by the available virtual capacity for space-efficient volumes. For example, a repository of 100 GB of reserved physical storage exists, and you define a virtual capacity of 200 GB. In this case, you can define 10 TSE LUNs of 20 GB each. So, the logical capacity can be larger than the physical capacity. You cannot fill all the volumes with data, because the total physical capacity is limited by the repository size, which is 100 GB in this example.

> **Important:** In the current implementation of TSE volumes, it is not possible to expand the physical size of the repository. Therefore, careful planning for the size of the repository is required before it is used. If a repository needs to be expanded, all TSE volumes within this extent pool must be deleted, and the repository deleted and re-created with the required size.

Space allocation

Space for a space-efficient volume is allocated when a write occurs. More precisely, it is allocated when a destage from the cache occurs and a new track or extent needs to be allocated. The TSE allocation unit is a track (64 KB for Open Systems LUNs or 57 KB for CKD volumes).

Because space is allocated in extents or tracks, the system needs to maintain tables that indicate their mapping to the logical volumes, so there is a performance impact involved with space-efficient volumes. The smaller the allocation unit (a track for TSE volumes), the larger the tables and possible effect on performance.

Virtual space is created as part of the extent pool definition. This virtual space is mapped to ESE volumes in the extent pool and TSE volumes in the repository, as needed. The virtual capacity equals the total logical capacity of all ESE and TSE volumes. No physical storage capacity (other than for the metadata) is allocated until write activity occurs to the ESE or TSE volumes.

The lifetime of data on TSE volumes is expected to be limited, because TSE volumes are intended to be used as temporary FlashCopy targets only. Physical storage is allocated when data is written to the TSE volumes, and we need a mechanism to free up physical space in the repository when the data is no longer needed. The FlashCopy commands have options to release the space of TSE volumes when the FlashCopy relationship is established or removed. The CLI commands **initfbvol** and **initckdvol** can also release the space for space-efficient volumes (ESE and TSE).

Use of extent space-efficient and track space-efficient volumes

Like standard volumes (which are fully provisioned), ESE volumes can be mapped to hosts. Since the DS8000 LMC R6.2, they are fully supported with FlashCopy. However, TSE volumes are supported as FlashCopy target volumes by using the FlashCopy SE feature. For detailed information about ESE and TSE volumes, see *DS8000 Thin Provisioning*, REDP-4554.

> **Tip:** In the DS8000 LMC R6.2, ESE volumes are fully supported by Easy Tier. TSE volumes are not supported by Easy Tier.

3.2.7 Extent allocation methods (EAMs)

There are two EAMs available on the DS8000 for non-managed extent pools: *rotate volumes* and *rotate extents* (also called *storage pool striping*).

When you create a volume in a managed extent pool, that is, an extent pool that is managed by Easy Tier automatic mode, the EAM of the volume always becomes *managed*.

> **Tip:** *Rotate extents* and *rotate volume* EAMs determine the distribution of a volume capacity and, thus, the volume workload distribution across the ranks in an extent pool. *Rotate extents* (the default EAM) evenly distributes the capacity of a volume at a granular 1 GiB extent level across the ranks in a homogeneous extent pool. It is the preferred method to reduce skew and minimize hot spots, improving overall performance.

Rotate volumes extent allocation method

Extents can be allocated sequentially. In this case, all extents are taken from the same rank until we have enough extents for the requested volume size or the rank is full, in which case the allocation continues with the next rank in the extent pool. If multiple volumes are created in one operation, the allocation for each volume starts in another rank. When allocating several volumes, we *rotate* with the start of each volume through the ranks.

You might want to consider this allocation method when you manage performance manually. The workload of one volume is going to one rank. This method helps you identify performance bottlenecks more easily. However, by placing all volume data and workload on one rank, you increase skew and the likelihood of limiting overall system performance with single ranks that become a bottleneck.

Rotate extents (storage pool striping) extent allocation method

The preferred storage allocation method is *rotate extents* (storage pool striping). It is the default EAM when a volume is created in a non-managed extent pool. The extents of a volume can be striped across several ranks at a granular 1 GiB extent level. It is the preferred method to reduce skew and minimize hot spots, improving overall performance in a homogeneous multi-rank extent pool. An extent pool with multiple ranks is needed to benefit from this storage allocation method.

While all volumes in the extent pool that use *rotate extents* are evenly distributed across all ranks in the pool, the initial start of each volume is additionally rotated throughout the ranks to improve a balanced rank utilization and workload distribution. If the first volume is created starting on rank R(*n*), the allocation for the later volume starts on the later rank R(*n*+1) in the pool. The DS8000 maintains a sequence of ranks. While the extents of each volume are rotated across all available ranks in the pool, the DS8000 additionally tracks the rank in which the last volume allocation started. The allocation of the first extent for the next volume then starts on the next rank in that sequence.

Extent allocation in hybrid and managed extent pools

When you create a volume in a managed extent pool, that is, an extent pool that is managed by Easy Tier automatic mode, the EAM of the volume always becomes managed. This situation is true no matter which extent allocation method is specified at volume creation. The volume is under control of Easy Tier. Easy Tier moves extents to the most appropriate storage tier and rank in the pool based on performance aspects. Any specified EAM, such as *rotate extents* or *rotate volumes*, is ignored. In managed extent pools, an initial EAM similar to *rotate extents* for new volumes is used.

In hybrid or multi-tier extent pools (whether currently managed or non-managed by Easy Tier), initial volume creation always starts on the ranks of the Enterprise tier first. The Enterprise tier is also called the *home tier*. The extents of a new volume are distributed in a *rotate extents* or storage pool striping fashion across all available ranks in this home tier in the extent pool as long as sufficient capacity is available. Only when all capacity on the home tier in an extent pool is consumed does volume creation continue on the ranks of the Nearline tier. When all capacity on the Enterprise tier and Nearline tier is exhausted, then volume creation continues allocating extents on the SSD tier. The initial extent allocation in non-managed hybrid pools differs from the extent allocation in single-tier extent pools with *rotate extents* (the extents of a volume are not evenly distributed across all ranks in the pool because of the different treatment of the different storage tiers). However, the attribute for the EAM of the volume is shown as *rotate extents* if the pool is not under Easy Tier automatic mode control. After the pool is managed by Easy Tier automatic mode, the EAM becomes managed.

In managed homogeneous extent pools with only a single storage tier, the initial extent allocation for a new volume is the same as with *rotate extents* or storage pool striping. For a volume, the appropriate DSCLI command, **`showfbvol`** or **`showckdvol`**, used with the **`-rank`** option, allows the user to list the number of allocated extents of a volume on each associated rank in the extent pool.

Starting with the DS8000 R6.1 LMC, the EAM attribute of any volume created or already in a managed extent pool is changed to *managed* after Easy Tier automatic mode is enabled for the pool. When enabling Easy Tier automatic mode for all extent pools, that is, hybrid and homogeneous extent pools, all volumes immediately become managed by Easy Tier. And, the EAM attribute of all volumes on a DS8000 system is changed to *managed*. When set to managed, the EAM attribute setting for the volume is permanent. All previous volume EAM attribute information, such as *rotate extents* or *rotate volumes*, is lost.

Additional considerations for volumes and the extent allocation method

By using striped volumes, you distribute the I/O load of a LUN/CKD volume to multiple sets of eight disk drives. The ability to distribute a workload to many physical drives can greatly enhance performance for a logical volume. In particular, operating systems that do not have a volume manager that can stripe can benefit most from this allocation method.

However, if you use, for example, Physical Partition striping in AIX already, double striping probably does not improve performance any further.

Tip: Double-striping a volume, for example, by using *rotate extents* in storage and striping a volume on the AIX Logical Volume Manager (LVM) level can lead to unexpected performance results. Consider striping on the storage system level or on the operating system level only. We explain this striping further in 3.3.2, “Extent pool considerations” on page 78.

If you decide to use storage pool striping, it is probably better to use this allocation method for all volumes in the extent pool to keep the ranks equally allocated and utilized.

Tip: When configuring a new DS8000, do not generally mix volumes that use the *rotate extents* EAM (storage pool striping) and volumes that use the *rotate volumes* EAM in the same extent pool.

Striping a volume across multiple ranks also increases the failure boundary. If you have extent pools with many ranks and all volumes striped across these ranks, you lose the data of all volumes in the pool if one rank fails. For example, two disk drives in the same RAID 5 rank fail at the same time.

If multiple EAM types are used in same extent pool, use Easy Tier manual mode to change the EAM from *rotate volumes* to *rotate extents* and vice versa. Use volume migration (dynamic volume relocation) in non-managed, homogeneous extent pools.

However, before switching any EAM of a volume, consider that you might need to change other volumes on same extent pool before distributing your volume across ranks. For example, you create various volumes with the *rotate volumes* EAM and only a few with *rotate extents*. Now, you want to switch only one volume to *rotate extents*. The ranks might not have enough free extents available for Easy Tier to properly balance all the extents evenly over all ranks. In this case, you might have to apply multiple steps and switch every volume to the new EAM type before changing only one volume. Depending on your case, you might also consider moving volumes to another extent pool prior to reorganizing all volumes and extents in the extent pool.

Certain cases, where you merge other extent pools, you also must plan to reorganize the extents on the new extent pool by using, for example, *manual volume rebalance*, so that you can properly redistribute the extents of the volumes across the ranks in the pool.

For more information about Easy Tier features and modes, see *IBM System Storage DS8000 Easy Tier*, REDP-4667.

3.2.8 Allocating, deleting, and modifying LUNs and CKD volumes

All extents of the ranks assigned to an extent pool are independently available for allocation to logical volumes. The extents for a LUN/volume are logically ordered, but they do not have to come from one rank and the extents do not have to be contiguous on a rank.

This construction method of using fixed extents to form a logical volume in the DS8000 series allows flexibility in the management of the logical volumes. We can delete LUNs/CKD volumes, resize LUNs/volumes, and reuse the extents of those LUNs to create other LUNs/volumes, maybe of different sizes. One logical volume can be removed without affecting the other logical volumes defined on the same extent pool.

Because the extents are *cleaned* after you delete a LUN or CKD volume, it can take some time until these extents are available for reallocation. The reformatting of the extents is a background process. Easy Tier mechanisms improve these effects and also help improve performance quickly. For more information, see 3.3.2, “Extent pool considerations” on page 78.

Logical volume configuration states

Each logical volume has a configuration state attribute. When a logical volume creation request is received, a logical volume object is created and the logical volume configuration state attribute is placed in the configuring configuration state.

After the logical volume is created and available for host access, it is placed in the normal configuration state. If a volume deletion request is received, the logical volume is placed in the deconfiguring configuration state until all capacity associated with the logical volume is deallocated and the logical volume object is deleted.

The reconfiguring configuration state is associated with a volume expansion request. The transposing configuration state is associated with an extent pool merge. The migrating, migration paused, migration error, and migration canceled configuration states are associated with a volume relocation request.

Dynamic volume expansion

The size of a LUN or CKD volume can be expanded without losing the data. On the DS8000, you add extents to the volume. The operating system must support this resizing capability. If a logical volume is dynamically expanded, the extent allocation follows the volume EAM. Dynamically reducing the size of a logical volume is not supported, because most operating systems do not support this feature.

> **Important:** Before you can expand a volume, you must remove any Copy Services relationships that involve that volume.

Dynamic volume relocation

Dynamic volume relocation (DVR) or volume migration is a capability provided as part of the Easy Tier manual mode feature. It allows a logical volume to be migrated from its current extent pool to another extent pool or even back into the same extent pool (to relocate the extents of a volume) while the logical volume and its data remain accessible to the attached hosts. The user can request dynamic volume relocation by using the migrate volume function that is available through the DS8000 Storage Manager GUI or DSCLI (**`managefbvol`** or **`manageckdvol`** command). Dynamic volume relocation allows the user to specify a target extent pool and an EAM. The target extent pool can be a different extent pool than the extent pool where the volume currently is located, or even the same extent pool.

> **Important:** Dynamic volume relocation can be applied among extent pools associated with the same DS8000 storage server affinity (storage server 0 or storage server 1) or rank group only. All volumes in even-numbered extent pools (P0, P2, P4, and so on) belong to rank group 0 and are serviced by storage server 0. All volumes in odd-numbered extent pools (P1, P3, P5, and so on) belong to rank group 1 and are serviced by storage server 1 (unless one DS8000 storage server failed or was quiesced with a failover to the alternate storage server).
>
> Additionally, the dynamic volume relocation is not supported in these situations:
>
> - If source and target pools are different storage types (FB and CKD)
> - If the volume to be migrated is a TSE volume

If the same extent pool is specified and *rotate extents* is used as the EAM, the volume migration is carried out as *manual volume rebalance,* as described in "Manual volume rebalance" on page 63. *Manual volume rebalance* is designed to redistribute the extents of volumes within a non-managed, single-tier (homogeneous) pool so that workload skew and hot spots are less likely to occur on the ranks. During extent relocation, only one extent at a time is allocated rather than preallocating the full volume and only a minimum amount of free capacity is required in the extent pool.

> **Important:** A volume migration with *dynamic volume relocation* back into the same extent pool (for example, *manual volume rebalance* for restriping purposes) is not supported in managed of hybrid extent pools. Hybrid pools are always supposed to be prepared for Easy Tier automatic management. In pools under control of Easy Tier automatic mode, the volume placement is managed automatically by Easy Tier. It relocates extents across ranks and storage tiers to optimize storage performance and storage efficiency. However, it is always possible to migrate volumes across extent pools, no matter if those pools are managed, non-managed, or hybrid pools.

For additional information about this topic, see *IBM System Storage DS8000 Easy Tier*, REDP-4667.

3.2.9 Logical subsystem (LSS)

A *logical subsystem* (LSS) is another logical construct. It groups logical volumes and LUNs in groups of up to 256 logical volumes.

On the DS8000 series, there is no fixed binding between a rank and a logical subsystem. The capacity of one or more ranks can be aggregated into an extent pool. The logical volumes configured in that extent pool are not necessarily bound to a specific rank. Different logical volumes on the same logical subsystem can even be configured in separate extent pools. The available capacity of the storage facility can be flexibly allocated across logical subsystems and logical volumes. You can define up to 255 LSSs on a DS8000 system.

For each LUN or CKD volume, you must select an LSS when creating the volume. The LSS is part of the volume ID '`abcd`' and reflected in the first two digits '`ab`'. The volume ID is in hexadecimal notation and needs to be specified upon volume creation.

> **Hexadecimal notation:** To emphasize the hexadecimal notation of the DS8000 volume IDs, LSS IDs, and address groups, an 'X' is used in front of the ID in part of the book.

You can have up to 256 volumes in one LSS. There is, however, one restriction. Volumes are created from extents of an extent pool. An extent pool, however, is associated with one DS8000 storage server (also called a CEC or processor complex): server 0 or server 1. The LSS number also reflects this affinity to one of these DS8000 storage servers. All even-numbered LSSs (X'00', X'02', X'04', up to X'FE') are serviced by storage server 0 (rank group 0). All odd-numbered LSSs (X'01', X'03', X'05', up to X'FD') are serviced by storage server 1 (rank group 1). LSS X'FF' is reserved.

All logical volumes in an LSS must be either CKD or FB. LSSs are even grouped into *address groups* of 16 LSSs. All LSSs within one address group must be of the same storage type, either CKD or FB. The first digit of the LSS ID or volume ID specifies the address group. For more information, see 3.2.10, "Address groups" on page 73.

System z users are familiar with a *logical control unit* (LCU). System z operating systems configure LCUs to create device addresses. There is a one-to-one relationship between an

LCU and a CKD LSS (LSS X'ab' maps to LCU X'ab'). Logical volumes have a logical volume number X'abcd' in hexadecimal notation where X'ab' identifies the LSS and X'cd' is one of the 256 logical volumes on the LSS. This logical volume number is assigned to a logical volume when a logical volume is created and determines with which LSS the logical volume is associated. The 256 possible logical volumes associated with an LSS are mapped to the 256 possible device addresses on an LCU (logical volume X'abcd' maps to device address X'cd' on LCU X'ab'). When creating CKD logical volumes and assigning their logical volume numbers, consider whether Parallel Access Volumes (PAVs) are required on the LCU and reserve addresses on the LCU for alias addresses.

For Open Systems, LSSs do not play an important role other than associating a volume with a specific rank group and server affinity (storage server 0 or storage server 1) or grouping hosts and applications together under selected LSSs for the DS8000 Copy Services relationships and management.

Tip: Certain management actions in Metro Mirror, Global Mirror, or Global Copy operate at the LSS level. For example, the freezing of pairs to preserve data consistency across all pairs is at the LSS level. The option to put all or a set of volumes of a certain application in one LSS can make the management of remote copy operations easier under certain circumstances.

Important: LSSs for FB volumes are created automatically when the first FB logical volume on the LSS is created, and deleted automatically when the last FB logical volume on the LSS is deleted. CKD LSSs require user parameters to be specified and must be created before the first CKD logical volume can be created on the LSS. They must be deleted manually after the last CKD logical volume on the LSS is deleted.

3.2.10 Address groups

Address groups are created automatically when the first LSS associated with the address group is created, and deleted automatically when the last LSS in the address group is deleted. The DSCLI **lsaddressgrp** command displays a list of address groups in use for the storage image and the storage type associated with it, either FB or CKD.

All devices in an address group must be either CKD or FB. LSSs are grouped into address groups of 16 LSSs. LSSs are numbered X'ab', where a is the address group. So all LSSs within one address group have to be of the same type, CKD or FB. The first LSS defined in an address group sets the type of that address group. For example, LSS X'10' to LSS X'1F' are all in the same address group and therefore can all be used only for the same storage type, either FB or CKD.

Figure 3-5 on page 74 shows the concept of volume IDs, LSSs, and address groups.

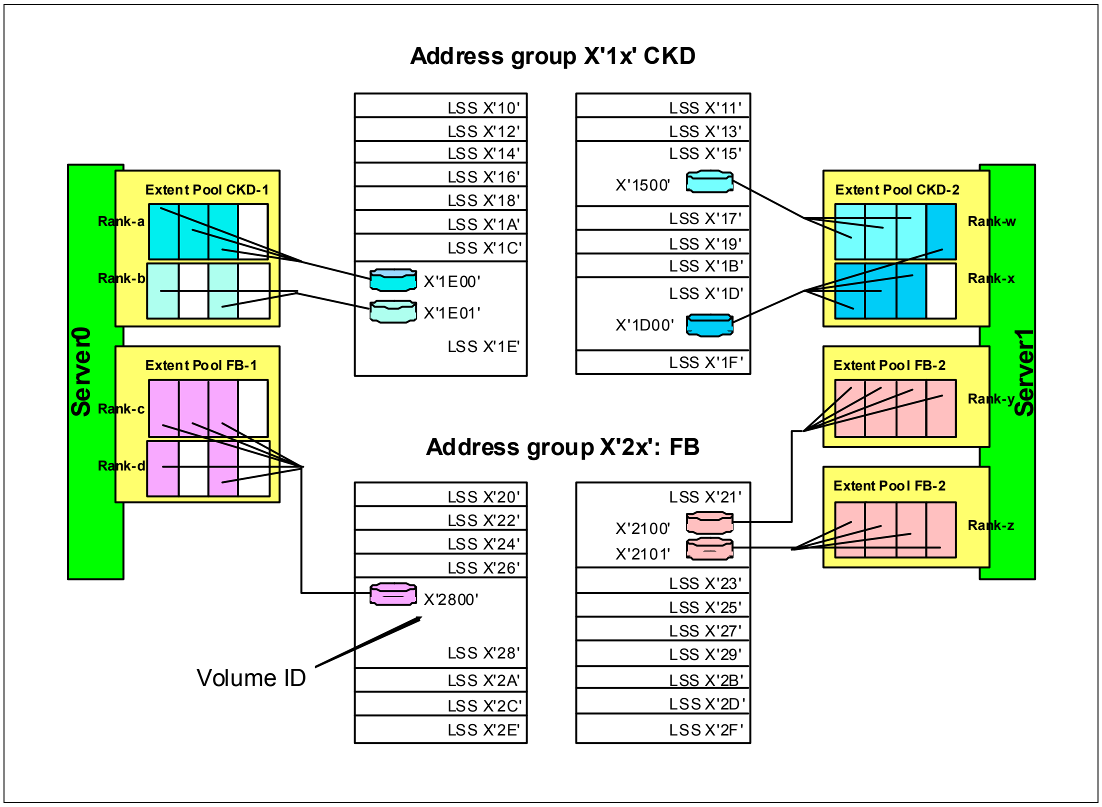


Figure 3-5 Volume IDs, logical subsystems, and address groups on the DS8000 storage systems

The volume ID X'*gabb*' in hexadecimal notation is composed of the address group X'*g*', and the LSS ID X'g*a*', and the volume number X'*bb*' within the LSS. For example, LUN X'2101' denotes the second (X'01') LUN in LSS X'21' of address group 2.

3.2.11 Volume access

A DS8000 provides mechanisms to control host access to LUNs. In most cases, a server or host has two or more host bus adapters (HBAs) and needs access to a group of LUNs. For easy management of server access to logical volumes, the DS8000 introduced the concept of host attachments and volume groups.

Host attachment

Host bus adapters (HBAs) are identified to the DS8000 in a host attachment or host connection construct that specifies the HBA worldwide port names (WWPNs). A set of host ports (host connections) can be associated through a port group attribute in the DSCLI that allows a set of HBAs to be managed collectively. This group is called a *host attachment* within the GUI.

Each host attachment can be associated with a volume group to define which LUNs that HBA is allowed to access. Multiple host attachments can share the volume group. The host attachment can also specify a port mask that controls which DS8000 I/O ports the HBA is allowed to log in to. Whichever ports the HBA logs in to, it sees the same volume group that is defined in the host attachment that is associated with this HBA.

The maximum number of host attachments on a DS8000 is 8,192.

Volume group

A *volume group* is a named construct that defines a set of logical volumes. When used in conjunction with CKD hosts, there is a default volume group that contains all CKD volumes. Any CKD host that logs in to a FICON I/O port has access to the volumes in this volume group. CKD logical volumes are automatically added to this volume group when they are created and automatically removed from this volume group when they are deleted.

When used in conjunction with Open Systems hosts, a host attachment object that identifies the HBA is linked to a specific volume group. You must define the volume group by indicating which FB logical volumes are to be placed in the volume group. Logical volumes can be added to or removed from any volume group dynamically.

One host connection can be assigned to one volume group only. However, the same volume group can be assigned to multiple host connections. An FB logical volume can be assigned to one or more volume groups. Assigning a logical volume to different volume groups allows a LUN to be shared by hosts, each configured with its own dedicated volume group and set of volumes (in case a set of volumes that is not identical is shared between the hosts).

The maximum number of volume groups is 8,320 for the DS8000.

3.2.12 Summary of the DS8000 logical configuration hierarchy

Describing the virtualization hierarchy, we started with multiple disks that were grouped in array sites. An array site is transformed into an array, eventually with spare disks. The array was further transformed into a rank with extents formatted for FB data or CKD. Next, the extents were added to an extent pool that determined which storage server served the ranks and aggregated the extents of all ranks in the extent pool for later allocation to one or more logical volumes. Within the extent pool, we can reserve storage for space-efficient volumes.

Next, we created logical volumes within the extent pools (optionally striping the volumes), assigning them a logical volume number that determined to which logical subsystem they are associated and indicated which server manages them. Space-efficient volumes can be created within the repository of the extent pool. Then, the LUNs can be assigned to one or more volume groups. Finally, the HBAs were configured into a host attachment that is associated with a volume group.

This virtualization concept provides for greater flexibility. Logical volumes can dynamically be created, deleted, migrated, and resized. They can be grouped logically to simplify storage management. Large LUNs and CKD volumes reduce the required total number of volumes, which also contributes to a reduction of management efforts.

Figure 3-6 on page 76 summarizes the virtualization hierarchy.

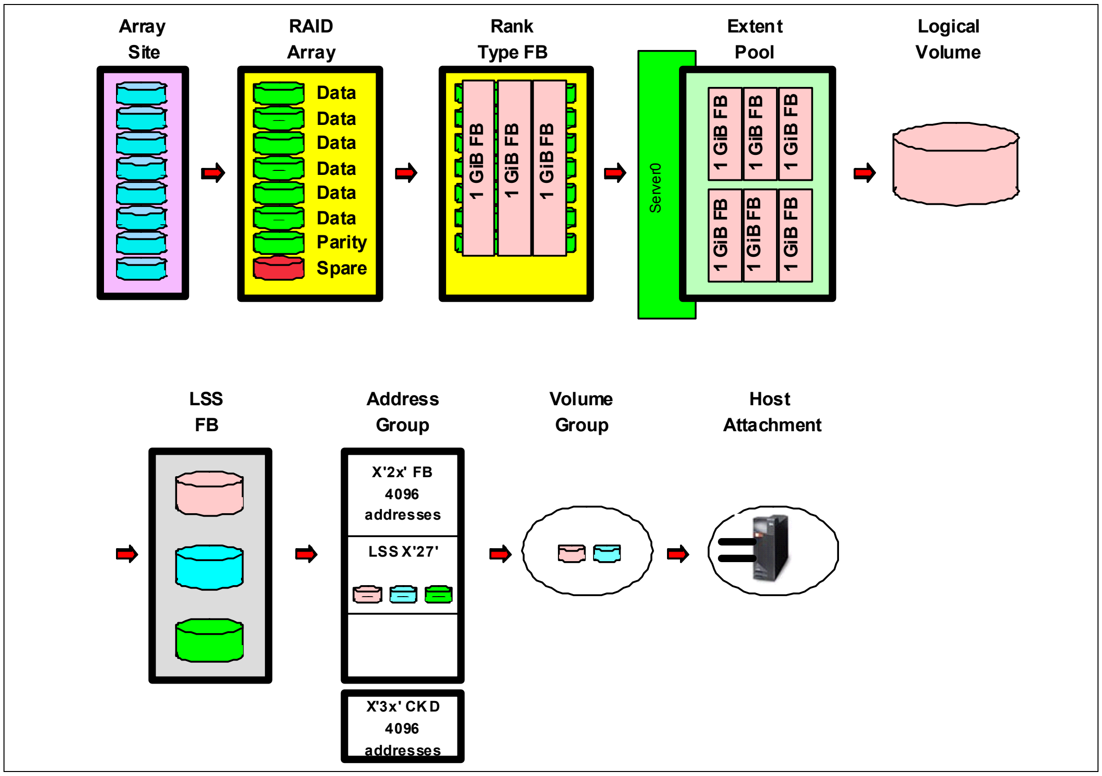


Figure 3-6 DS8000 virtualization hierarchy

3.3 Data placement on ranks and extent pools

It is important to understand how the volume data is placed on ranks and extent pools. This understanding helps you to decide how to create extent pools and choose the required number of ranks within an extent pool. It also helps you to understand and detect performance problems or optimally tweak overall system performance.

It can also help you fine-tune system performance from an extent pool perspective, for example, sharing the resources of an extent pool evenly between application workloads or isolating application workloads to dedicated extent pools. Data placement can help you when planning for dedicated extent pools with different performance characteristics and storage tiers without using Easy Tier automatic management. Plan your configuration carefully to meet your performance goals by minimizing potential performance limitations that might be introduced by single resources that become a bottleneck due to workload skew. For example, use *rotate extents* as the default EAM to help reduce the risk of single ranks that become a hot spot and limit the overall system performance due to workload skew.

If workload isolation is required in your environment, you can isolate workloads and I/O on the rank and DA levels on the DS8000 systems, if required.

Important: It is important to understand that the greatest amount of isolation that you can achieve within the DS8000 is at the array/rank level, not the physical drive level.

You can manually manage storage tiers related to different homogeneous extent pools of the same storage class and plan for appropriate extent pools for your specific performance needs. Remember that DS8800 and DS8700 systems offer the IBM System Storage Easy

Tier feature at no cost, which provides full automatic subvolume data relocation capabilities within extent pools to automatically optimize storage performance and storage efficiency in single-tier and multi-tier extent pools. It provides optimum performance and a balanced resource utilization at a minimum configuration and management effort.

3.3.1 Rank and array considerations

For fine-tuning performance-intensive workloads that use homogeneous pool configurations, a 7+P RAID 5 array performs better than a 6+P array, and a 4+4 RAID 10 performs better than a 3+3. For random I/O, you might see an up to 15% greater throughput on a 7+P and a 4+4 array than a 6+P and a 3+3. For sequential applications, the differences are minimal. Generally, try to balance workload activity evenly across RAID arrays, regardless of the size. It is typically not worth the additional management effort to do it otherwise. However, if you use multiple pools, evaluate whether it makes sense to ensure that each pool contains a similar proportion of 6+P and 7+P (or 3+3/4+4 or 5+P+Q/6+P+Q) ranks.

In a RAID 5 6+P or 7+P array, the amount of capacity equal to one disk is used for parity information. However, the parity information is not bound to a single disk. Instead, it is striped across all the disks in the array, so all disks of the array are involved to service I/O requests equally.

In a RAID 6 5+P+Q or 6+P+Q array, the amount of capacity equal to two disks is used for parity information. As with RAID 5, the parity information is not bound to single disks, but instead is striped across all the disks in the array. So, all disks of the array service I/O requests equally. However, a RAID 6 array might have one less drive available than a RAID 5 array configuration. Nearline drives, for example, allow RAID 6 configurations only.

In a RAID 10 3+3 array, the available usable space is the capacity of only three disks. Two disks of the array site are used as spare disks. When a LUN is created from the extents of this array, the data is always mirrored across two disks in the array. Each write to the array must be performed twice to two disks. There is no additional parity information in a RAID 10 array configuration.

Important: The spares in the mirrored RAID 10 configuration act independently; they are not mirrored spares.

In a RAID 10 4+4 array, the available usable space is the capacity of four disks. No disks of the array site are used as spare disks. When a LUN is created from the extents of this array, the data is always mirrored across two disks in the array. Each write to the array must be performed twice to two disks.

Important: The stripe width for the RAID arrays differs in size with the number of active disks that hold the data. Due to the different stripe widths that make up the extent from each type of RAID array, it is generally not a preferred practice to intermix RAID array types within the same extent pool, especially in homogeneous extent pool configurations that do not use Easy Tier automatic management. With Easy Tier enabled, the benefit of automatic storage performance and storage efficiency management combined with Easy Tier micro-tiering capabilities typically outperforms the disadvantage of different RAID arrays within the same pool.

3.3.2 Extent pool considerations

The data placement and thus the workload distribution of your volumes on the ranks in a non-managed homogeneous extent pool is determined by the selected EAM as described in 3.2.7, "Extent allocation methods (EAMs)" on page 68 and the number of ranks in the pool.

Even in single-tier homogeneous extent pools, you can already benefit from Easy Tier automatic mode (DSCLI **`chsi ETautomode=all`**). It manages the subvolume data placement within the managed pool based on rank utilization and thus reduces workload skew and hot spots (auto-rebalance).

In multi-tier hybrid extent pools, you can fully benefit from Easy Tier automatic mode (DSCLI **`chsi ETautomode=all|tiered`**). It provides full automatic storage performance and storage economics management by optimizing subvolume data placement in a managed extent pool across different storage tiers and even across ranks within each storage tier (auto-rebalance). Easy Tier automatic mode and hybrid extent pool configurations offer the most efficient way to use different storage tiers. It optimizes storage performance and storage economics across three drive tiers to manage more applications effectively and efficiently with a single DS8000 system at an optimum price versus the performance and footprint ratio.

The data placement and extent distribution of a volume across the ranks in an extent pool can always be displayed by using the DSCLI **`showfbvol -rank`** or **`showckdvol -rank`** command, as shown in Example 3-2 on page 85.

Before configuring extent pools and volumes, be aware of the basic configuration principles about workload sharing, isolation, and spreading, as described in 4.2, "Configuration principles for optimal performance" on page 90.

Single-tier extent pools

A single-tier or homogeneous extent pool is formed by using disks of the same storage tier with the same drive characteristics, which means not combining Enterprise class disks with nearline-class disks or SSDs in the same extent pool.

The first example shown in Figure 3-7 on page 79 illustrates an extent pool with only one rank, which is also referred to a *single-rank extent pool*. This approach is common if you plan to use the SAN Volume Controller (SVC), for example, or if you plan a configuration that uses the maximum isolation that you can achieve on the rank/extpool level. In this type of a single-rank extent pool configuration, all volumes created are bound to a single rank. This type of configuration requires careful logical configuration and performance planning, because single ranks are likely to become a hot spot and might limit overall system performance. It also requires the highest administration and management effort, because workload skew typically varies over time. You might constantly monitor your system performance and need to react to hot spots. It also considerably limits the benefits that a DS8000 system can offer regarding its virtualization and Easy Tier automatic management capabilities.

In these configurations, we highly advise that you use host-based striping methods to achieve a balanced distribution of the data and the I/O workload across the ranks and back-end disk resources. For example, you can use IBM AIX LVM or SVC to stripe the volume data across ranks.

With Easy Tier manual mode features, such as dynamic extent pool merge and dynamic volume relocation (volume migration), you can use these functions:

- Easily reconsider your logical configuration needs.
- Dynamically merge extent pools, restripe logical volumes (manual volume rebalance), and migrate volumes from one extent pool to another.
- Gradually move to a multi-rank, multi-tier configuration to fully benefit from the DS8000 enhanced virtualization capabilities, such as storage pool striping and automatic performance management as provided by Easy Tier.

In Figure 3-7, we show the data placement of two volumes created in an extent pool with a single rank. Volumes created in this extent pool always use extents from rank R6 and, thus, are limited to the capacity and performance capability of this single rank. Without the use of any host-based data and workload striping methods across multiple volumes from different extent pools and ranks, this rank is likely to experience rank hot spots and performance bottlenecks.

Also, in this example, one host can easily degrade the whole rank, depending on its I/O workload, and affect multiple hosts that share volumes on the same rank if you have more than one LUN allocated in this extent pool.

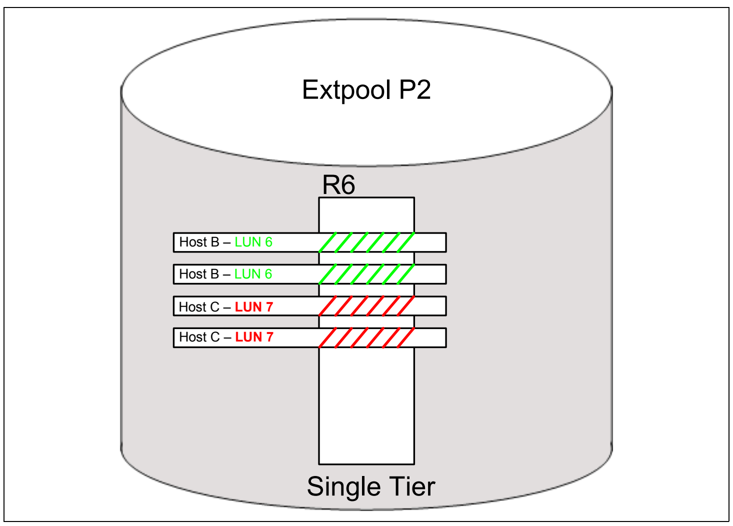


Figure 3-7 Extent pool with a single rank

The second example, which is shown in Figure 3-8 on page 81, illustrates an extent pool with multiple ranks of the same storage class or storage tier, which we refer to as a homogeneous or single-tier extent pool. In general, an extent pool with multiple ranks also is called a *multi-rank extent pool*.

Although in principle, both EAMs (*rotate extents* and *rotate volumes)* are available for non-managed homogeneous extent pools, it is preferable to use the default allocation method of *rotate extents* (*storage pool striping*). Use this EAM to distribute the data and thus the workload evenly across all ranks in the extent pool and minimize the risk of workload skew and a single rank that becomes a hot spot.

The use of the EAM *rotate volumes* still can isolate volumes to separate ranks, even in multi-rank extent pools, wherever such configurations are required. This EAM minimizes the configuration effort, because a set of volumes distributed across different ranks can be created with a single command. Plan your configuration and performance needs to implement host-level-based methods to balance the workload evenly across all volumes and ranks.

However, we discourage the use of both EAMs in the same extent pool without an efficient host-level-based striping method for the non-striped volumes. This approach easily forfeits the benefits of storage pool striping and likely leads to imbalanced workloads across ranks and potential single-rank performance bottlenecks.

Figure 3-8 on page 81 is an example of storage pool striping for LUNs 1 - 4. It shows more than one host and more than one LUN distributed across the ranks. In contrast to our suggestion, we also show an example of LUN 5 being created with the *rotate volumes* EAM in the same pool. The storage system tries to allocate the continuous space available for this volume on a single rank (R1) until there is insufficient capacity left on this rank and then it spills over to the next available rank (R2). All workload on this LUN is limited to these two ranks. This approach considerably increases the workload skew across all ranks in the pool and the likelihood that these two ranks might become a bottleneck for all volumes in the pool, which reduces overall pool performance.

Enabling Easy Tier automatic mode for homogeneous, single-tier extent pools always is an additional option to let the DS8000 system manage system performance in the pools based on rank utilization (*auto-rebalance*). The EAM of all volumes in the pool becomes *managed* in this case. With Easy Tier and its advanced *micro-tiering* capabilities that take different RAID levels and drive characteristics into account for determining the rank utilization in managed pools, even a mix of different drive characteristics and RAID levels of the same storage tier might be an option for certain environments.

Multiple hosts with multiple LUNs, as shown in Figure 3-8 on page 81, share the resources (*resource sharing*) in the extent pool, that is, ranks, DAs, and physical spindles. If one host or LUN has a high workload, I/O contention can result and easily affect the other application workloads in the pool, especially if all applications have their workload peaks at the same time. Alternatively, applications can benefit from a much larger amount of disk spindles and thus larger performance capabilities in a shared environment in contrast to *workload isolation* and only dedicated resources. With resource sharing, we generally expect that not all applications peak at the same time, so that each application typically benefits from the larger amount of disk resources that it can use. The resource sharing and storage pool striping in non-managed extent pools method is a good approach for most cases if no other requirements, such as workload isolation or a specific quality of service requirements, dictate another approach.

With Easy Tier and I/O Priority Manager, the DS8000 systems offer advanced features when taking advantage of resource sharing to minimize administration efforts and reduce workload skew and hot spots while benefitting from automatic storage performance, storage economics, and workload priority management. The use of these features in the DS8000 environments is highly encouraged. These features generally help to provide excellent overall system performance while ensuring quality of service (QoS) levels by prioritizing workloads in shared environments at a minimum of administration effort and at an optimum price-performance ratio.

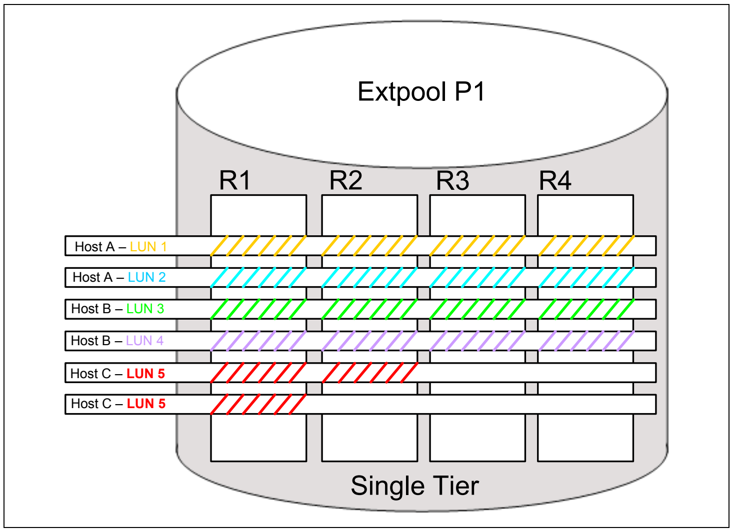


Figure 3-8 Single-tier extent pool with multiple ranks

Multi-tier extent pools

A *multi-tier* or *hybrid extent pool* is formed by combining ranks of different storage classes or different storage tiers within the same extent pool. Hybrid pools are always supposed to be managed by Easy Tier automatic mode, which provides automated storage performance and storage economics management. It provides full automated cross-tier and intra-tier management by dynamically relocating subvolume data (at the extent level) to the appropriate rank and storage tier within an extent pool based on its access pattern. Easy Tier supports automatic management of up to three storage tiers that use Enterprise, nearline, and SSD class drives. The use of Easy Tier to let the system manage data placement and performance is highly encouraged with DS8800 and DS8700 storage systems.

When you create a volume in a *managed* extent pool, that is, an extent pool that is managed by Easy Tier automatic mode, the EAM of the volume always becomes *managed*. This situation is true no matter which EAM is specified at volume creation. The volume is under control of Easy Tier. Easy Tier moves extents to the most appropriate storage tier and rank in the pool based on performance aspects. Any specified EAM, such as *rotate extents* or *rotate volumes,* is ignored.

In managed or hybrid extent pools, an initial EAM that is similar to *rotate extents* for new volumes is used. The same situation applies if an existing volume is manually moved to a managed or hybrid extent pool by using volume migration or dynamic volume relocation. In hybrid or multi-tier extent pools (whether currently managed or non-managed by Easy Tier), initial volume creation always starts on the ranks of the Enterprise tier first. The Enterprise tier is also called the home tier. The extents of a new volume are distributed in a *rotate extents* or storage pool striping fashion across all available ranks in this home tier in the extent pool, as long as sufficient capacity is available. Only when all capacity on the home tier in an extent pool is consumed does volume creation continue on the ranks of the Nearline tier. When all capacity on the Enterprise tier and Nearline tier is exhausted, then volume creation continues allocating extents on the SSD tier. The initial extent allocation in non-managed hybrid pools differs from the extent allocation in single-tier extent pools with *rotate extents* (the extents of a volume are not evenly distributed across all ranks in the pool because of the different treatment of the different storage tiers). However, the attribute for the EAM of the volume is shown as *rotate extents*, as long as the pool is not under Easy Tier automatic mode control. After Easy Tier automatic mode is enabled for a hybrid pool, the EAM of all volumes in that pool becomes *managed*.

Mixing different storage tiers combined with Easy Tier automatic performance and economics management on a subvolume level can considerably increase the performance versus price ratio, increase energy savings, and reduce the overall footprint. The use of Easy Tier automated subvolume data relocation and the addition of an SSD storage tier are generally good for mixed environments with applications that demand both IOPs and bandwidth at the same time. For example, database systems might have different I/O demands according to their architecture. Costs might be too high to allocate a whole database on SSDs. Mixing different drive technologies, for example, SSD with Enterprise or nearline disks, and efficiently allocating the data capacity on the subvolume level across the tiers with Easy Tier can highly optimize price, performance, the footprint, and the energy usage. Only the hot part of the data allocates SSD capacity instead of provisioning SSD capacity for full volumes. Therefore, you can achieve considerable system performance at a reduced cost and footprint with only a few SSDs.

The ratio of SSD capacity to hard disk drive (HDD) disk capacity in a hybrid pool depends on the workload characteristics and skew. Ideally, there must be enough SSD capacity to hold the active (hot) extents in the pool, but not more, to not waste expensive SSD capacity. For new DS8000 orders, 3 - 5% of SSD capacity might be a reasonable percentage to plan with hybrid pools in typical environments. This configuration can result in the movement of 50% of the small and random I/O workload from Enterprise drives to SSDs. This configuration provides a reasonable initial estimate if measurement data is not available to support configuration planning. See “Drive Selection with Easy Tier” in *IBM System Storage DS8800 and DS8700 Performance with Easy Tier 3rd Generation*, WP102024, for additional information.

The *Storage Tier Advisor Tool* (STAT) also provides guidance for SSD capacity planning based on the existing workloads on a DS8000 with Easy Tier monitoring capabilities. For additional information about the STAT, see 6.7, “Storage Tier Advisor Tool” on page 231.

Figure 3-9 shows a configuration of a managed 2-tier extent pool with an SSD and Enterprise storage tier. All LUNs are managed by Easy Tier. Easy Tier automatically and dynamically relocates subvolume data to the appropriate storage tier and rank based on their workload patterns. Figure 3-9 shows multiple LUNs from different hosts allocated in the 2-tier pool with hot data already migrated to SSDs. Initial volume creation in this pool always allocates extents on the Enterprise tier first, as long as capacity is available, before Easy Tier automatically starts promoting extents to the SSD tier.

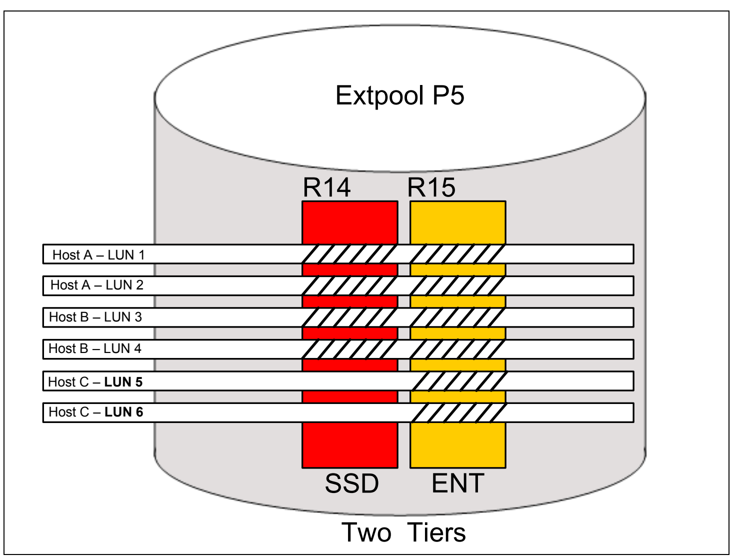


Figure 3-9 Multi-tier extent pool with SSD and Enterprise ranks

Figure 3-10 shows a configuration of a managed 2-tier extent pool with an Enterprise and Nearline storage tier. All LUNs are managed by Easy Tier. Easy Tier automatically and dynamically relocates subvolume data to the appropriate storage tier and rank based on their workload patterns. With more than one rank in the Enterprise storage tier, Easy Tier also balances the workload and resource usage across the ranks within this storage tier (auto-rebalance). Figure 3-10 shows multiple LUNs from different hosts allocated in the 2-tier pool with cold data already demoted to the Nearline tier. Initial volume creation in this pool always allocates extents on the Enterprise tier first, as long as capacity is available, before Easy Tier automatically starts demoting extents to the Nearline tier.

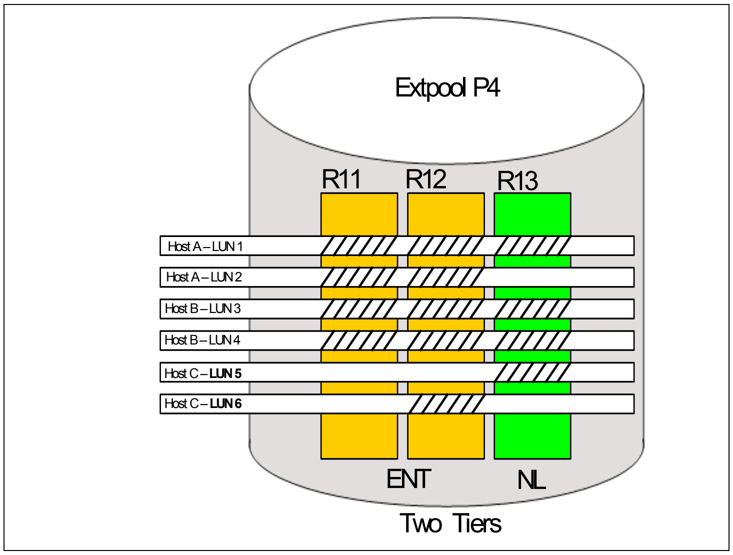


Figure 3-10 Multi-tier extent pool with Enterprise and nearline ranks

Figure 3-11 on page 84 shows a configuration of a managed 3-tier extent pool with an SSD, Enterprise, and Nearline storage tier. All LUNs are managed by Easy Tier. Easy Tier automatically and dynamically relocates subvolume data to the appropriate storage tier and rank based on their workload patterns. With more than one rank in the Enterprise storage tier, Easy Tier also balances the workload and resource usage across the ranks within this storage tier (auto-rebalance). Figure 3-11 on page 84 shows multiple LUNs from different hosts allocated in the 3-tier pool with hot data promoted to the SSD tier and cold data demoted to the Nearline tier. Initial volume creation in this pool always allocates extents on the Enterprise tier first, as long as capacity is available, before Easy Tier automatically starts promoting extents to the SSD tier or demoting extents to the Nearline tier.

We highly encourage the use of hybrid extent pool configurations under automated Easy Tier management. It provides ease of use with minimum administration and performance management efforts while optimizing the system performance, price, footprint, and energy costs.

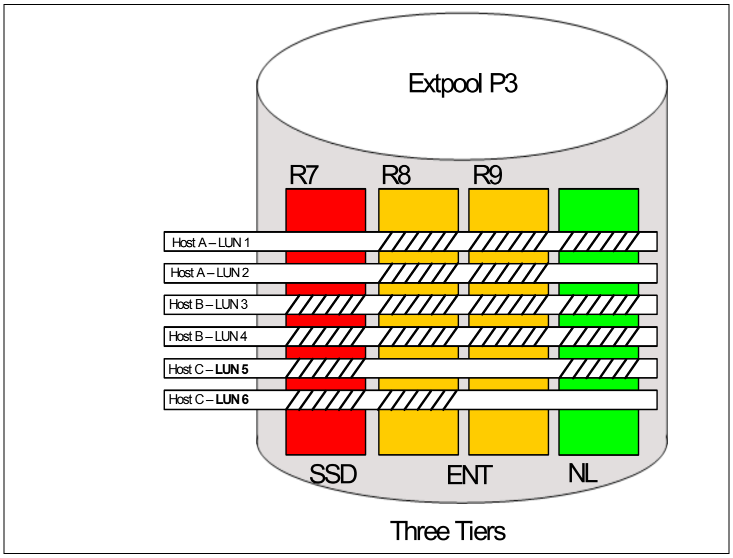


Figure 3-11 Multi-tier extent pool with SSD, Enterprise, and nearline ranks

3.3.3 Easy Tier considerations

As soon as an extent pool is under control of Easy Tier automatic mode, the EAM of all volumes in this pool becomes *managed*. Easy Tier automatically manages subvolume data placement on the extent level and relocates the extents based on their access pattern to the appropriate storage tier in the pool (cross-tier or inter-tier management). Easy Tier also rebalances the extents across the ranks of the same storage tier based on resource usage (intra-tier management or *auto-rebalance*) to minimize skew and avoid hot spots in the extent pool. With the DS8000 R6.2 LMC, Easy Tier can be enabled for multi-tier hybrid extent pools with up to three storage tiers or single-tier homogeneous extent pools and automatically manage the data placement of the volumes in these pools.

The initial allocation of extents for a volume in a managed single-tier pool is similar to the *rotate extents* EAM or *storage pool striping*. So, the extents are evenly distributed across all ranks in the pool right after the volume creation. The initial allocation of volumes in *hybrid* extent pools differs slightly. The extent allocation always begins in a *rotate extents*-like fashion on the ranks of the Enterprise tier first, and then continues on the nearline ranks, and eventually on the SSD ranks, if the capacity on the Enterprise and nearline ranks is not sufficient.

After the initial extent allocation of a volume in the pool, the extents and their placement on the different storage tiers and ranks are managed by Easy Tier. Easy Tier collects workload statistics for each extent in the pool and creates migration plans to relocate the extents to the appropriate storage tiers and ranks. The extents are promoted to higher tiers or demoted to lower tiers based on their actual workload patterns. The data placement of a volume in a managed pool is no longer static or determined by its initial extent allocation. The data placement of the volume across the ranks in a managed extent pool is subject to change over time to constantly optimize storage performance and storage economics in the pool. This process is ongoing and always adapting to changing workload conditions. After Easy Tier data collection and automatic mode are enabled, it might require up to 24 hours before the first migration plan is created and being applied. For Easy Tier migration plan creation and timings, see *IBM System Storage DS8000 Easy Tier*, REDP-4667.

The DSCLI **`showfbvol -rank`** or **`showckdvol -rank`** command can help to show the current extent distribution of a volume across the ranks, as shown in Example 3-2 on page 85. In this example, volume 8601 is managed by Easy Tier and distributed across ranks R0 (SSD), R18 (ENT), and R22 (NL). The **lsarray -l -rank Rxy** command can additionally be used to show the storage class and DA pair of a specific rank Rxy.

Example 3-2 Use the showfbvol -rank to show the volume to rank relationship in a multi-tier pool

```
dscli> showfbvol -rank 8601
Name            -
ID              8601
accstate        Online
datastate       Normal
configstate     Normal
deviceMTM       2107-900
datatype        FB 512
addrgrp         8
extpool         P6
exts            100
captype         DS
cap (2^30B)     100.0
cap (10^9B)     -
cap (blocks)    209715200
volgrp          V8
ranks           3
dbexts          0
sam             Standard
repcapalloc     -
eam             managed
reqcap (blocks) 209715200
realextents     100
virtualextents  0
migrating       0
perfgrp         PG0
migratingfrom   -
resgrp          RG0
==============Rank extents==============
rank extents
============
R0        16
R18       24
R22       60

dscli> lsarray -l -rank r0
Array State    Data   RAIDtype  arsite Rank DA Pair DDMcap (10^9B) diskclass encrypt
=========================================================================================
A0    Assigned Normal 5 (6+P+S) S1     R0   0                300.0 SSD       unsupported

dscli> lsarray -l -rank r18
Array State    Data   RAIDtype  arsite Rank DA Pair DDMcap (10^9B) diskclass encrypt
=========================================================================================
A3    Assigned Normal 5 (6+P+S) S4     R18  2                600.0 ENT       unsupported

dscli> lsarray -l -rank r22
Array State    Data   RAIDtype    arsite Rank DA Pair DDMcap (10^9B) diskclass encrypt
===========================================================================================
A16   Assigned Normal 6 (5+P+Q+S) S17    R22  2               3000.0 NL        unsupported
```

The *volume heat distribution* (volume heat map), provided by the STAT, helps you to identify hot, warm, and cold extents for each volume and its current distribution across the storage tiers in the pool. For more information about the STAT, see 6.7, “Storage Tier Advisor Tool” on page 231.

Figure 3-12 gives an example of a three-tier hybrid pool managed by Easy Tier. It shows the change of the volume data placement across the ranks over time when Easy Tier relocates extents based on their workload pattern and adapts changing workload conditions. At T1, we see that all volumes are initially allocated on the Enterprise tier, evenly balanced across ranks R8 and R9. After becoming cold, some extents from LUN 1 are demoted to nearline drives at T3. LUN 2 and LUN 3, with almost no activity, are demoted to nearline drives at T2. After increased activity, some data of LUN 2 is promoted to Enterprise drives again at T3. The hot extent of LUN 4 is promoted to SSD, and the cold extent is demoted to nearline drives at T2. After changing workload conditions on LUN 4 and the increased activity on the demoted extents, these extents are promoted again to Enterprise drives at T3. Cold extents from LUN 5 are demoted to nearline drives with a constant low access pattern over time. LUN 6 shows some extents promoted to SSD at T2 and further extent allocation changes across the ranks of the Enterprise tier at T3. The changes are due to potential extent relocations from the solid-state drive (SSD) tier (warm demote or swap operations) or across the ranks within the Enterprise tiers (auto-rebalance), balancing the workload based on workload patterns and rank utilization.

Easy Tier constantly adapts to changing workload conditions and relocates extents, so the extent locations of a volume are subject to a constant change over time that depends on its workload pattern. However, the number of relocations decreases and eventually becomes marginal, as long as the workload pattern about the decision windows of Easy Tier remains steady.

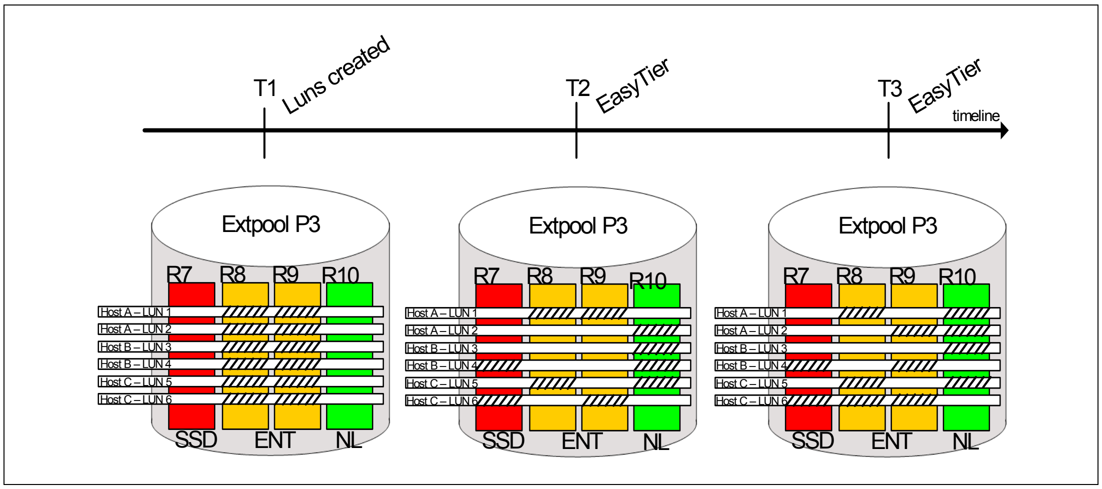


Figure 3-12 Data placement in a managed 3-tier extent pool with Easy Tier over time

4

Logical configuration performance considerations

> **Important:** Before reading this chapter, familiarize yourself with the material covered in Chapter 3, “Logical configuration concepts and terminology” on page 51.

This chapter introduces a step-by-step approach to configuring the DS8000 workload and performance considerations:

- Review the tiered storage concepts and Easy Tier
- Understand the configuration principles for optimal performance:
  - Workload isolation
  - Workload resource-sharing
  - Workload spreading
- Analyze workload characteristics to determine isolation or resource-sharing
- Plan allocation of the DS8000 disk and host connection capacity to identified workloads
- Plan spreading volumes and host connections for the identified workloads
- Plan array sites
- Plan RAID arrays and ranks with RAID-level performance considerations
- Plan extent pools with single-tier and multi-tier extent pool considerations
- Plan address groups, logical subsystems (LSSs), volume IDs, and count key data (CKD) Parallel Access Volumes (PAVs)
- Plan I/O port IDs, host attachments, and volume groups
- Implement and document the DS8000 logical configuration

4.1 Review the tiered storage concepts and Easy Tier

Many storage environments support a diversity of needs and use disparate technologies that cause storage sprawl. In a large-scale storage infrastructure, this environment yields a suboptimal storage design that can be improved only with a focus on data access characteristics analysis and management to provide optimum performance.

4.1.1 Tiered storage concept

Tiered storage is an approach of using different types of storage throughout the storage infrastructure. It is a mix of higher performing/higher-cost storage with lower performing/lower-cost storage and placing data based on specific characteristics, such as the performance needs, age, and importance of data availability. Correctly balancing these tiers leads to the minimal cost and best performance solution. The concept and a cost versus performance relationship of a tiered storage environment is shown in Figure 4-1.

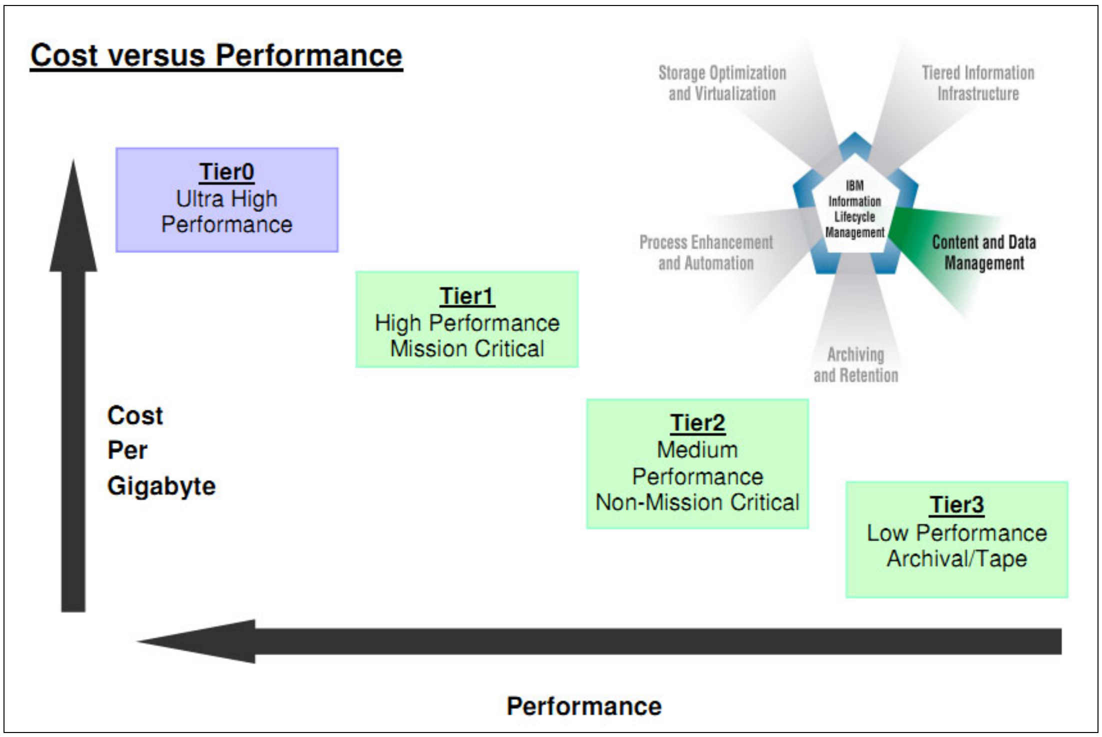


Figure 4-1 Cost versus performance in a tiered storage environment

Typically, an optimal design keeps the active operational data in Tier 0 and Tier 1 and uses Tiers 2 and 3 for less active data, as shown in Figure 4-2 on page 89.

The benefits associated with a tiered storage approach mostly relate to cost. By introducing SSD storage as tier 0, you might more efficiently address the highest performance needs while reducing the Enterprise class storage, system footprint, and energy costs. A tiered storage approach can provide the performance you need and save significant costs associated with storage, because lower-tier storage is less expensive. Environmental savings, such as energy, footprint, and cooling reductions, are possible. However, the overall management effort increases when managing storage capacity and storage performance needs across multiple storage classes.

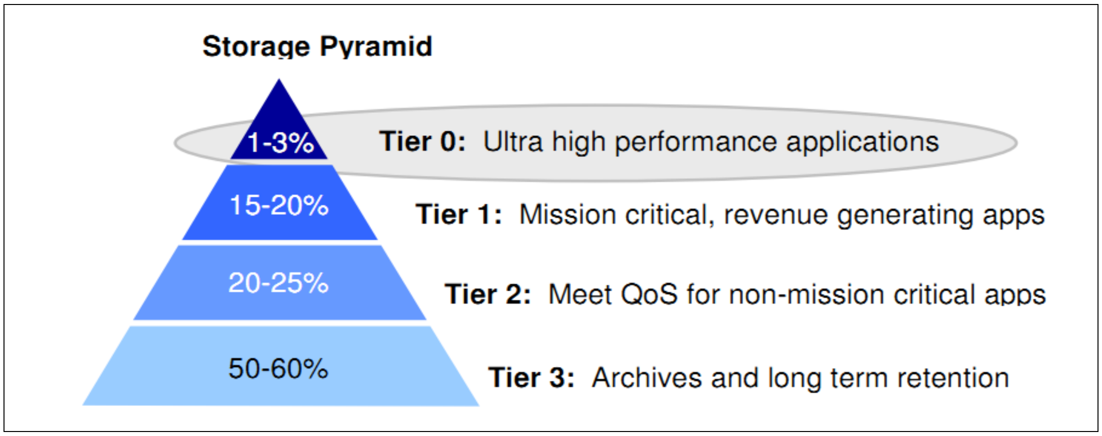


Figure 4-2 Storage pyramid with average relative amount of data in each tier

4.1.2 IBM System Storage Easy Tier

With solid-state drives (SSDs) emerging as an attractive alternative to hard disk drives (HDDs) in the enterprise storage market, tiered storage concepts that use SSD, Enterprise (Ent), and nearline (NL) storage tiers are a promising approach to satisfy storage performance and storage economics needs in mixed environments at an optimum price-performance ratio.

With dramatically high I/O rates, low response times, and IOPS-energy-efficient characteristics, SSDs address the highest performance needs and also potentially can achieve significant savings in operational costs. Although, the current acquisition cost per GB is higher than HDDs. To satisfy most workload characteristics, SSDs need to be used efficiently in conjunction with HDDs in a well-balanced tiered storage architecture. It is critical to choose the right mix of storage tiers and the right data placement to achieve optimal storage performance and economics across all tiers at a low cost.

With the DS8000, you can easily implement tiered storage environments that use SSD, Enterprise, and Nearline class storage tiers. Different storage tiers can be isolated to separate extent pools and volume placement can be managed manually across extent pools where required. Or, better and highly encouraged, volume placement can be managed automatically on a subvolume level in hybrid extent pools by IBM System Storage Easy Tier automatic mode with minimum management effort for the storage administrator. Easy Tier is a no-cost feature on DS8800 and DS8700 systems (however, a license must be ordered). For more information about Easy Tier, see 1.3.4, “Easy Tier” on page 10.

Although Easy Tier manual mode also helps you manage storage tiers across extent pools on a volume level by providing essential features such as dynamic volume relocation/volume migration, dynamic extent pool merge, and rank depopulation, consider Easy Tier automatic mode and hybrid extent pools for managing tiered storage on the DS8000. The overall management and performance monitoring effort increases considerably when manually managing storage capacity and storage performance needs across multiple storage classes and does not achieve the efficiency as provided with Easy Tier automatic mode data relocation on subvolume level (extent level). With Easy Tier, client configurations show less potential to waste SSD capacity than with volume-based tiering methods. And the use of Easy Tier is easy. Configure hybrid extent pools (mixed SSD/HDD storage pools) and turn Easy Tier on. It then provides automated data relocation across the storage tiers and ranks in the extent pool in order to optimize *storage performance* and *storage economics*. It also rebalances the workload across the ranks within each storage tier (*auto-rebalance*) based on rank utilization to minimize skew and hot spots. Furthermore, it constantly adapts to changing workload conditions. There is no need anymore to bother with tiering policies that must be manually applied to accommodate changing workload dynamics.

In environments with homogeneous system configurations or isolated storage tiers that are bound to different homogeneous extent pools, you can benefit from Easy Tier automatic mode. With R6.2 of the DS8000 LMC, Easy Tier provides automatic intra-tier performance management by rebalancing the workload across ranks (auto-rebalance) in homogeneous single-tier pools based on rank utilization. Easy Tier automatically minimizes skew and rank hot spots and helps to reduce the overall management effort for the storage administrator.

Depending on the particular storage requirements in your environment, with the DS8000 architecture, you can address a vast range of storage needs combined with ease of management. On a single DS8000 system, you can perform these tasks:

- Isolate workloads to selected extent pools (or down to selected ranks and DAs)
- Share resources of other extent pools with different workloads
- Use Easy Tier to automatically manage multi-tier extent pools with different storage tiers (or homogeneous extent pools)
- Adapt your logical configuration easily and dynamically at any time to changing performance or capacity needs by migrating volumes across extent pools, merging extent pools, or removing ranks from one extent pool (rank depopulation) and moving them to another pool

Easy Tier helps you to consolidate more workloads onto a single DS8000 system by automating storage performance and storage economics management across up to three drive tiers. In addition, I/O Priority Manager, as described in 1.3.5, “I/O Priority Manager” on page 17, can help you align workloads to quality of service (QoS) levels to prioritize separate system workloads that compete for the same shared and possibly constrained storage resources to meet their performance goals.

For many initial installations, an approach with two extent pools (with or without different storage tiers) and enabled Easy Tier automatic management might be the simplest way to start if you have FB or CKD storage only; otherwise, four extent pools are required. You can plan for more extent pools based on your specific environment and storage needs, for example, workload isolation for some pools, different resource sharing pools for different departments or clients, or specific Copy Services considerations.

Considerations, as described in 4.2, “Configuration principles for optimal performance” on page 90, apply for planning advanced logical configurations in complex environments, depending on your specific requirements. Also, take advantage of new features, such as Easy Tier and I/O Priority Manager.

4.2 Configuration principles for optimal performance

There are three major principles for achieving a logical configuration on a DS8000 subsystem for optimal performance when planning extent pools:

- Workload isolation
- Workload resource-sharing
- Workload spreading

And, you can take advantage of the new DS8000 features offered by Easy Tier and I/O Priority Manager. Both features pursue different goals and can be combined.

Easy Tier provides a significant benefit for mixed workloads and therefore needs to be considered for resource-sharing workloads, and isolated workloads dedicated to a specific set of resources. Furthermore, Easy Tier automatically supports the goal of workload spreading by spreading the workload in an optimum way across all the dedicated resources in an extent pool. It provides automated storage performance and storage economics optimization through dynamic data relocation on extent level across multiple storage tiers and ranks based on their access patterns. And with *auto-rebalance,* it rebalances the workload across the ranks within a storage tier based on utilization to reduce skew and avoid hot spots. Auto-rebalance applies to managed multi-tier pools and single-tier pools and helps to rebalance the workloads evenly across ranks to provide an overall balanced rank utilization within a storage tier or managed single-tier extent pool. Figure 4-3 shows the effect of auto-rebalance in a single-tier extent pool that starts with a highly imbalanced workload across the ranks at T1. Auto-rebalance rebalances the workload and optimizes the rank utilization over time.

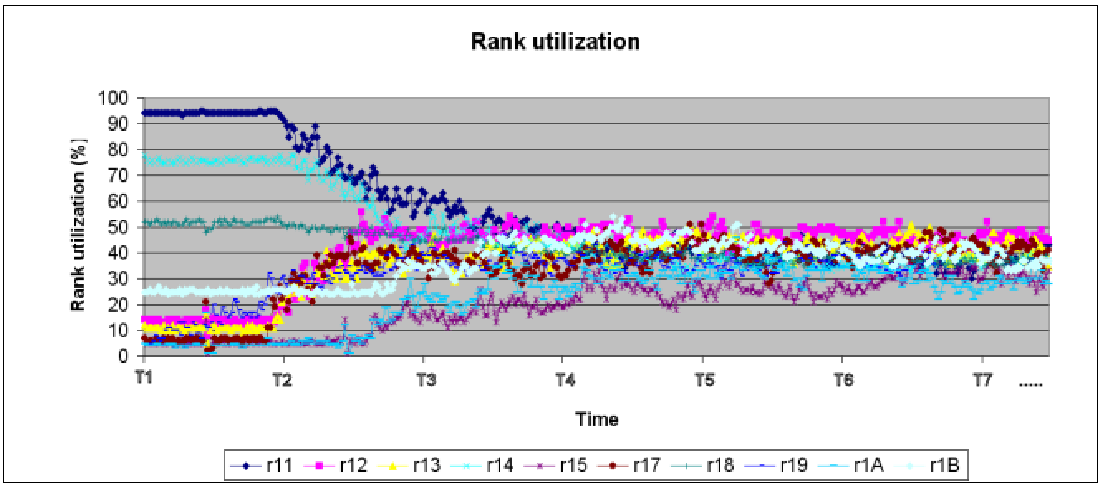


Figure 4-3 Effect of auto-rebalance on individual rank utilization in the system

The *DS8000 I/O Priority Manager* provides a significant benefit for resource-sharing workloads. It aligns QoS levels to separate workloads that compete for the same shared and possibly constrained storage resources. I/O Priority Manager can prioritize access to these system resources to achieve the desired QoS for the volume based on predefined performance goals (high, medium, or low). I/O Priority Manager constantly monitors and balances system resources to help applications meet their performance targets automatically, without operator intervention. I/O Priority Manager only acts if resource contention is detected.

4.2.1 Workload isolation

Workload isolation provides a high priority workload with dedicated DS8000 hardware resources to reduce the impact of less important workloads. Workload isolation can also mean limiting a lower priority workload to a subset of the DS8000 hardware resources so that it does not affect more important workloads by fully utilizing all hardware resources.

Isolation provides guaranteed availability of the hardware resources that are dedicated to the isolated workload. It removes contention with other applications for those resources.

However, isolation limits the isolated workload to a subset of the total DS8000 hardware so that its maximum potential performance might be reduced. Unless an application has an entire DS8000 Storage Image dedicated to its use, there is potential for contention with other applications for any hardware (such as cache and processor resources) that is not dedicated.

However, typically, isolation is implemented to improve the performance of all workloads by separating different workload types.

One preferred practice to isolation is to identify lower priority workloads with heavy I/O demands and to separate them from all of the more important workloads. You might be able to isolate multiple lower priority workloads with heavy I/O demands to a single set of hardware resources and still meet their lower service-level requirements, particularly if their peak I/O demands are at different times. In addition, I/O Priority Manager can help to prioritize different workloads, if required.

Important: For convenience, this chapter sometimes describes isolation as a single isolated workload in contrast to multiple resource-sharing workloads, but the approach also applies to multiple isolated workloads.

DS8000 disk capacity isolation

The level of disk capacity isolation required for a workload depends on the scale of its I/O demands as compared to the DS8000 array and DA capabilities, as well as organizational considerations, such as the importance of the workload and application administrator requests for workload isolation.

You can partition the DS8000 disk capacity for isolation at several levels:

- **Rank level**: Certain ranks are dedicated to a workload. That is, volumes for one workload are allocated on these ranks. The ranks can be a different disk type (capacity or speed), a different RAID array type (RAID 5, RAID 6, or RAID 10, arrays with spares or arrays without spares), or a different storage type (CKD or FB) than the disk types, RAID array types, or storage types that are used by other workloads. Workloads that require different storage types dictate rank, extent pool, and address group isolation. You might consider workloads with heavy random activity for rank isolation, for example.
- **Extent pool level**: Extent pools are logical constructs that represent a group of ranks serviced by storage server 0 or storage server 1. You can isolate different workloads to different extent pools, but you always need to be aware of the rank and DA pair associations. While physical isolation on rank and DA level involves building appropriate extent pools with a selected set of ranks or ranks from a specific DA pair, different extent pools with a subset of ranks from different DA pairs typically share DAs. As long as the workloads are purely disk bound and limited by the capability of the disk spindles rather than the DA pair, it can be considered as an isolation level for the set of ranks and disk spindles set aside in the pool. In combination with Easy Tier on a system with multiple extent pools of different storage tier combinations and likely sharing DA adapters across different extent pools, you might also consider different extent pools also as isolation levels for workloads for different tier combinations (1-tier, 2-tier, and 3-tier pools) and Easy Tier management modes (managed/non-managed). Be aware that isolated workloads to different extent pools might share a DA adapter as a physical resource, which can be a potential limiting physical resource under certain extreme conditions. For example, a condition might be high IOPS rates in combination with SSD ranks or high-bandwidth utilizations in combination with sequential workloads.
- **DA level**: All ranks on one or more DA pairs are dedicated to a workload. That is, only volumes for this workload are allocated on the ranks that are associated with one or more DAs. These ranks can be a different disk type (capacity or speed), RAID array type (RAID 5, RAID 6, or RAID 10, arrays with spares or arrays without spares), or storage type (CKD or FB) than the disk types, RAID types, or storage types that are used by other workloads. You must consider workloads with heavy, large blocksize, and sequential activity for DA-level isolation, because these workloads tend to consume all of the available DA resources.

- **Processor complex level**: All ranks assigned to extent pools managed by processor complex 0 or all ranks assigned to extent pools managed by processor complex 1 are dedicated to a workload. We typically do not suggest this approach, because it can reduce the processor and cache resources and the back-end bandwidth available to the workload by 50%.
- **Storage unit level**: All ranks in a physical DS8000 are dedicated to a workload. That is, the physical DS8000 runs one workload.

DS8000 host connection isolation

The level of host connection isolation required for a workload depends on the scale of its I/O demands as compared to the DS8000 I/O port and host adapter (HA) capabilities. It also depends on organizational considerations, such as the importance of the workload and administrator requests for workload isolation.

The DS8000 host connection subsetting for isolation can also be done at several levels:

- **I/O port level**: Certain DS8000 I/O ports are dedicated to a workload. This subsetting is common. Workloads that require Fibre Channel connection (FICON) and Fibre Channel Protocol (FCP) must be isolated at the I/O port level, because each I/O port on the 4-port or 8-port FCP/FICON-capable HA card can be configured to support only one of these protocols. Although Open Systems host servers and remote mirroring links use the same protocol (FCP), they are typically isolated to different I/O ports. You must also consider workloads with heavy large block sequential activity for HA isolation, because they tend to consume all of the I/O port resources that are available to them.
- **HA level**: Certain HAs are dedicated to a workload. FICON and FCP workloads do not necessarily require HA isolation, because separate I/O ports on the same 4-port/8-port FCP/FICON-capable HA card can be configured to support each protocol (FICON or FCP). However, it can be considered a preferred practice to separate FCP and FICON to different HBAs if available. Furthermore, host connection requirements might dictate a unique type of HA card (Long wave (LW) or Short wave (SW)) for a workload. Workloads with heavy large block sequential activity must be considered for HA isolation, because they tend to consume all of the I/O port resources that are available to them.
- **I/O enclosure level**: Certain I/O enclosures are dedicated to a workload. This approach is not generally necessary.

4.2.2 Workload resource-sharing

Workload resource-sharing means multiple workloads use a common set of the DS8000 hardware resources:

- Ranks
- DAs
- I/O ports
- HAs

Multiple resource-sharing workloads can have logical volumes on the same ranks and can access the same DS8000 HAs or I/O ports. Resource-sharing allows a workload to access more DS8000 hardware than can be dedicated to the workload, providing greater potential performance, but this hardware sharing can result in resource contention between applications that impacts overall performance at times. It is important to allow resource-sharing only for workloads that do not consume all of the DS8000 hardware resources that are available to them. Or, use I/O Priority Manager to align QoS levels to the volumes that are sharing resources and prioritize different workloads, if required.

Easy Tier extent pools typically are shared by multiple workloads, because Easy Tier with its automatic data relocation and performance optimization across multiple storage tiers provides the most benefit for mixed workloads.

To better understand the resource-sharing principle for workloads on disk arrays, see 3.3.2, “Extent pool considerations” on page 78.

4.2.3 Workload spreading

Workload spreading means balancing and distributing overall workload evenly across all of the DS8000 hardware resources available:

- Processor complex 0 and processor complex 1
- DAs
- Ranks
- I/O enclosures
- HAs

Spreading applies to both isolated workloads and resource-sharing workloads.

You must allocate the DS8000 hardware resources to either an isolated workload or multiple resource-sharing workloads in a balanced manner. That is, you must allocate either an isolated workload or resource-sharing workloads to the DS8000 ranks that are assigned to DAs and both processor complexes in a balanced manner. You must allocate either type of workload to I/O ports that are spread across HAs and I/O enclosures in a balanced manner.

You must distribute volumes and host connections for either an isolated workload or a resource-sharing workload in a balanced manner across all DS8000 hardware resources that are allocated to that workload.

You must create volumes as evenly as possible across all ranks and DAs allocated to those workloads.

One exception to the recommendation of spreading volumes might be when specific files or datasets are never accessed simultaneously, such as multiple log files for the same application where only one log file is in use at a time. In that case, you can place the volumes required by these datasets or files on the same resources.

You must also configure host connections as evenly as possible across the I/O ports, HAs, and I/O enclosures that are available to either an isolated or a resource-sharing workload. Then, you can use host server multipathing software to optimize performance over multiple host connections. For more information about multipathing software, see Chapter 8, “Host attachment” on page 311.

4.2.4 Using workload isolation, resource-sharing, and spreading

A typical approach to optimizing performance on the DS8000 is to begin by identifying any workload that has the potential to negatively impact the performance of other workloads by fully utilizing all of the DS8000 I/O ports and the DS8000 ranks available to it.

Additionally, you must identify any workload that is so critical that its performance can never be allowed to be negatively impacted by other workloads.

Then, identify the remaining workloads that are considered appropriate for resource-sharing.

Next, define a balanced set of hardware resources that can be dedicated to any isolated workloads, if required. Then, allocate the remaining DS8000 hardware for sharing among the resource-sharing workloads. Carefully consider the appropriate resources and storage tiers for Easy Tier and multi-tier extent pools in a balanced manner. Also, plan ahead for appropriate I/O Priority Manager alignments of QoS levels to resource-sharing workloads where needed.

The next step is planning extent pools and assigning volumes and host connections to all workloads in a way that is balanced and spread - either across all dedicated resources (for any isolated workload) or across all shared resources (for the multiple resource-sharing workloads). For spreading workloads evenly across a set of ranks, consider multi-rank extent pools that use storage pool striping, which we introduce in “Rotate extents (storage pool striping) extent allocation method” on page 68 and discuss also in 4.8, “Planning extent pools” on page 115.

Without the explicit need for workload isolation or any other requirements for multiple extent pools, starting with two extent pools (with or without different storage tiers) and a balanced distribution of the ranks and DAs might be the simplest configuration to start with using resource-sharing throughout the whole DS8000 system and Easy Tier automatic management - provided that you have FB or CKD storage. Otherwise, four extent pools are required for a reasonable minimum configuration, two for FB storage and two for CDK storage, and each pair distributed across both DS8000 storage servers. In addition, you can plan to align your workloads to expected QoS levels with I/O Priority Manager.

The final step is the implementation of host-level striping (when appropriate) and multipathing software, if desired. If you planned for Easy Tier, do not consider host-level striping, because it dilutes the workload skew and is counterproductive to the Easy Tier optimization.

4.3 Analyzing application workload characteristics

The first and most important step in creating a successful logical configuration for the DS8000 is analyzing the workload characteristics for the applications that access the DS8000. The DS8000 hardware resources, such as RAID arrays and I/O ports, must be correctly allocated to workloads for isolation and resource-sharing considerations. Or, if planning for shared multi-tier configurations and Easy Tier, it is important to determine the skew of the workload and plan for the amount of required storage capacity on the appropriate storage tiers. You need to perform this workload analysis during the DS8000 capacity planning process, and you need to complete it before ordering the DS8000 hardware.

4.3.1 Determining skew and storage requirements for Easy Tier

For Easy Tier configurations, it is important to determine the *skew* of the workload and plan for the amount of required storage capacity on each of the storage tiers. Plan the optimum initial hardware configuration for managed multi-tier environments so that you determine the overall distribution of the I/O workload against the amount of data (data heat distribution) to understand how much of the data is doing how much (or most) of the I/O workload. The workload pattern, small block random or large block sequential read/write operations, also is important. A good understanding of the workload heat distribution and skew helps to evaluate the benefit of an Easy Tier configuration.

For example, the ratio of SSD capacity to HDD capacity in a hybrid pool depends on the workload characteristics and skew. Ideally, there must be enough SSD capacity to hold the active (hot) extents in the pool, but not more, to not waste expensive SSD capacity. For new DS8000 orders, 3 - 5% of SSD capacity might be a reasonable percentage to plan for with hybrid pools in typical environments. This configuration can result in the movement of 50% of the small and random I/O workload from Enterprise drives to SSDs. It provides a reasonable initial estimate if measurement data is not available to support configuration planning. See “Drive Selection with Easy Tier” of *IBM System Storage DS8800 and DS8700 Performance with Easy Tier 3rd Generation*, WP102024, for additional information.

The Storage Tier Advisor Tool (STAT) also can provide guidance for capacity planning of the available storage tiers based on the existing workloads on a DS8000 system with Easy Tier monitoring enabled. For additional information, see 6.7, “Storage Tier Advisor Tool” on page 231.

4.3.2 Determining isolation requirements

The objective of this analysis is to identify workloads that require isolated (dedicated) DS8000 hardware resources, because this determination ultimately affects the total amount of disk capacity required and the total number of disk drive types required, as well as the number and type of HAs required. The result of this first analysis indicates which workloads require isolation and the level of isolation that is required.

You must also consider organizational and business considerations in determining which workloads to isolate. Workload priority (the importance of a workload to the business) is a key consideration. Application administrators typically request dedicated resources for high priority workloads. For example, certain database online transaction processing (OLTP) workloads might require dedicated resources to guarantee service levels.

The most important consideration is preventing lower priority workloads with heavy I/O requirements from impacting higher priority workloads. Lower priority workloads with heavy random activity need to be evaluated for rank isolation. Lower priority workloads with heavy, large blocksize, sequential activity must be evaluated for DA and I/O port isolation.

Workloads that require different disk drive types (capacity and speed), different RAID types (RAID 5, RAID 6, or RAID 10), or different storage types (CKD or FB) dictate isolation to different DS8000 arrays, ranks, and extpools. For more information about the performance implications of various RAID types, see 4.7.1, “RAID-level performance considerations” on page 103.

Workloads that use different I/O protocols (FCP or FICON) dictate isolation to different I/O ports. However, workloads that use the same disk drive types, RAID type, storage type, and I/O protocol need to be evaluated for separation or isolation requirements.

Workloads with heavy, continuous I/O access patterns must be considered for isolation to prevent them from consuming all available DS8000 hardware resources and impacting the performance of other types of workloads. Workloads with large blocksize and sequential activity must be considered for separation from those workloads with small blocksize and random activity.

Isolation of only a few workloads that are known to have high I/O demands can allow all the remaining workloads (including the high priority workloads) to share hardware resources and achieve acceptable levels of performance. More than one workload with high I/O demands might be able to share the isolated DS8000 resources, depending on the service level requirements and the times of peak activity.

The following examples are I/O workloads, files, or datasets that might have heavy and continuous I/O access patterns:

- Sequential workloads (especially those workloads with large blocksize transfers)
- Log files or datasets
- Sort or work datasets or files
- Business Intelligence and Data Mining
- Disk copies (including Point-in-Time Copy background copies, remote mirroring target volumes, and tape simulation on disk)
- Video and imaging applications
- Engineering and scientific applications
- Certain batch workloads

You must consider workloads for all applications for which DS8000 storage is allocated, including current workloads to be migrated from other installed storage subsystems and new workloads that are planned for the DS8000. Also, consider projected growth for both current and new workloads.

For existing applications, consider historical experience first. For example, is there an application where certain datasets or files are known to have heavy, continuous I/O access patterns? Is there a combination of multiple workloads that might result in unacceptable performance if their peak I/O times occur simultaneously? Consider workload importance (workloads of critical importance and workloads of lesser importance).

For existing applications, you can also use performance monitoring tools that are available for the existing storage subsystems and server platforms to understand current application workload characteristics:

- Read/write ratio
- Random/sequential ratio
- Average transfer size (blocksize)
- Peak workload (I/Os per second for random access and MB per second for sequential access)
- Peak workload periods (time of day and time of month)
- Copy Services requirements (Point-in-Time Copy and Remote Mirroring)
- Host connection utilization and throughput (FCP host connections and FICON)
- Remote mirroring link utilization and throughput

Estimate the requirements for new application workloads and for current application workload growth. You can obtain information about general workload characteristics in Chapter 5, “Understanding your workload” on page 157.

As new applications are rolled out and current applications grow, you must monitor performance and adjust projections and allocations. You can obtain more information in Appendix A, “Performance management process” on page 631 and in Chapter 7, “Practical performance management” on page 235.

You can use the Disk Magic modeling tool to model the current or projected workload and estimate the required DS8000 hardware resources. We introduce Disk Magic in 6.1, “Disk Magic” on page 176.

The Storage Tier Advisor Tool can also provide workload information and capacity planning recommendations associated with a specific workload to reconsider the need for isolation and evaluate the potential benefit when using a multi-tier configuration and Easy Tier. For more information, see 6.7, “Storage Tier Advisor Tool” on page 231.

4.3.3 Reviewing remaining workloads for feasibility of resource-sharing

After workloads with the highest priority or the highest I/O demands are identified for isolation, the I/O characteristics of the remaining workloads must be reviewed to determine whether a single group of resource-sharing workloads is appropriate, or whether it makes sense to split the remaining applications into multiple resource-sharing groups. The result of this step is the addition of one or more groups of resource-sharing workloads to the DS8000 configuration plan.

4.4 Planning allocation of disk and host connection capacity

You need to plan the allocation of specific DS8000 hardware first for any isolated workload, and then, for the resource-sharing workloads, including Easy Tier hybrid pools. Use the workload analysis from 4.3.2, “Determining isolation requirements” on page 96 again, this time to define the disk capacity and host connection capacity required for the workloads. For any workload, the required disk capacity is determined by both the amount of space needed for data and the number of arrays (of a specific speed) needed to provide the desired level of performance. The result of this step is a plan that indicates the number of ranks (including disk drive type) and associated DAs and the number of I/O adapters and associated I/O enclosures required for any isolated workload and for any group of resource-sharing workloads.

4.4.1 Planning DS8000 hardware resources for isolated workloads

For the DS8000 disk allocation, isolation requirements might dictate the allocation of certain individual ranks or all of the ranks on certain DAs to one workload. For the DS8000 I/O port allocation, isolation requirements might dictate the allocation of certain I/O ports or all of the I/O ports on certain HAs to one workload.

Choose the DS8000 resources to dedicate in a balanced manner. If ranks are planned for workloads in multiples of two, half of the ranks can later be assigned to extent pools managed by processor complex 0, and the other ranks can be assigned to extent pools managed by processor complex 1. You must also note the DAs to be used. If I/O ports are allocated in multiples of four, they can later be spread evenly across all I/O enclosures in a DS8000 frame if four or more HA cards are installed. If I/O ports are allocated in multiples of two, they can later be spread evenly across left and right I/O enclosures.

4.4.2 Planning DS8000 hardware resources for resource-sharing workloads

There might be one or more groups of resource-sharing workloads or different hybrid pool configurations planned for Easy Tier. By now, the shared set of DS8000 hardware resources that these groups of workloads use must be identified and assigned.

Review the DS8000 resources to share for balance. If ranks are planned for resource-sharing workloads in multiples of two, half of the ranks can later be assigned to processor complex 0 extent pools, and the other ranks can be assigned to processor complex 1 extent pools. You must also identify the DAs to use. If I/O ports are allocated for resource-sharing workloads in multiples of four, they can later be spread evenly across all I/O enclosures in a DS8000 frame if four or more HA cards are installed. If I/O ports are allocated in multiples of two, they can later be spread evenly across left and right I/O enclosures.

Easy Tier later provides automatic intra-tier management in single-tier and multi-tier pools (auto-rebalance) and cross-tier management in multi-tier pools for the resource-sharing workloads. In addition, different QoS levels can be aligned to different workloads to meet performance goals.

4.5 Planning volume and host connection spreading

After hardware resources are allocated for both isolated and resource-sharing workloads, plan the volume and host connection spreading for all of the workloads.

Host connection: In this chapter, we use *host connection* in a general sense to represent a connection between a host server (either z/OS or Open Systems) and the DS8000.

The result of this step is a plan that includes this information:

- The specific number and size of volumes for each isolated workload or group of resource-sharing workloads and how they are allocated to ranks and DAs
- The specific number of I/O ports for each workload or group of resource-sharing workloads and how they are allocated to HAs and I/O enclosures

After the spreading plan is complete, use the DS8000 hardware resources that are identified in the plan as input to order the DS8000 hardware.

4.5.1 Spreading volumes for isolated and resource-sharing workloads

At this point, consider the requirements of each workload for the number and size of logical volumes. For a specific amount of required disk capacity from the perspective of the DS8000, there are typically no significant DS8000 performance implications of using more small volumes as compared to fewer large volumes. However, using one or a few standard volume sizes can simplify management.

However, there are host server performance considerations related to the number and size of volumes. For example, for System z servers, the number of Parallel Access Volumes (PAVs) that are needed can vary with volume size. For more information about PAVs, see 14.2, “Parallel Access Volumes” on page 500. For System i servers, we suggest that you use a volume size that is half the size of the disk drives used. There also can be Open Systems host server or multipathing software considerations related to the number or the size of volumes, so you must consider these factors in addition to workload requirements.

There are significant performance implications with the assignment of logical volumes to ranks and DAs. The goal of the entire logical configuration planning process is to ensure that volumes for each workload are on ranks and DAs that allow all workloads to meet performance objectives.

Follow these steps for spreading volumes across allocated hardware for each isolated workload, and then for each workload in a group of resource-sharing workloads:

1. Review the required number and the size of the logical volumes that are identified during the workload analysis.
2. Review the number of ranks allocated to the workload (or group of resource-sharing workloads) and the associated DA pairs.
3. Evaluate the use of multi-rank or multi-tier extent pools. Use *rotate extents* (storage pool striping) as the EAM in homogeneous extent pools to spread workloads evenly across the

ranks. Evaluate the use of Easy Tier in automatic mode for single-tier and multi-tier extent pools to automatically manage data placement and performance. The default EAM is *rotate extents* if not specified otherwise. See “Extent allocation methods (EAMs)” on page 68 for more details about the EAMs.

4. Start assigning each required logical volume of a specific workload to a different rank or a different set of aggregated ranks (which means an extent pool with multiple ranks that use storage pool striping or are managed by Easy Tier) of the assigned resources wherever possible:

   – If the number of volumes required is less than the number of ranks (or sets of aggregated ranks), assign the volumes evenly to ranks or extent pools that are owned by processor complex 0 (storage server 0 or rank group 0) and ranks or extent pools that are owned by processor complex 1 (storage server 1 or rank group 1) on as many DA pairs as possible.

   – If the number of volumes required is greater than the number of ranks (or sets of aggregated ranks/extent pools), assign additional volumes to the ranks and DAs in a balanced manner. Ideally, the workload has the same number of logical volumes on each of its ranks, on each DA available to it.

   – Use pairs of Easy Tier managed single-tier or multi-tier pools. Each pair must be distributed across both DS8000 storage servers, and the volumes of your Easy Tier-managed workloads must be evenly balanced across the two managed extent pools.

5. If you used striped volumes that use rotate extents or made them automatically managed by Easy Tier, you can skip this step. Otherwise, you can use host-level striping, such as Open Systems Logical Volume Manager striping or z/OS storage groups.

4.5.2 Spreading host connections for isolated and resource-sharing workloads

Next, consider the requirements of each workload for the number and type of host connections. In addition to workload requirements, you also might need to consider the host server or multipathing software in relation to the number of host connections. For more information about multipathing software, see Chapter 8, “Host attachment” on page 311.

There are significant performance implications from the assignment of host connections to I/O ports, HAs, and I/O enclosures. The goal of the entire logical configuration planning process is to ensure that host connections for each workload access I/O ports and HAs that allow all workloads to meet the performance objectives.

Follow these steps for spreading host connections across allocated hardware for each isolated workload, and then for each workload in a group of resource-sharing workloads:

1. Review the required number and type (SW, LW, FCP, or FICON) of host connections that is identified in the workload analysis. You must use a minimum of *two* host connections to different DS8000 HA cards to ensure availability. Some Open Systems hosts might impose limits on the amount of paths and volumes, for example, VMware ESX. In such cases, you might consider not exceeding four paths per volume, which in general is a good approach for performance and availability and also for other operating systems. The DS8800 front-end host ports are 8 Gbps capable and if the expected workload is not explicitly saturating the adapter and port bandwidth with high sequential loads, you might share ports with many hosts.

2. Review the HAs that are allocated to the workload (or group of resource-sharing workloads) and the associated I/O enclosures.

3. Review requirements that need I/O port isolation, for example, remote replication Copy Services, ProtecTIER®, and SAN Volume Controller. If possible try to split them as you split hosts among hardware resources. Do not mix them with other Open Systems, because they can have different workload characteristics.
4. Assign each required host connection to a different HA in a different I/O enclosure if possible, balancing them across the left and right I/O enclosures:
   - If the required number of host connections is less than the available number of I/O enclosures (which can be typical for certain Open Systems servers), an equal number of host connections must be assigned to the left I/O enclosure (0, 2, 4, and 6) and the right I/O enclosure (1, 3, 5, and 7) if possible.
   - Within an I/O enclosure, assign each required host connection to the HA of the required type (SW FCP/FICON-capable or LW FCP/FICON-capable) with the greatest number of unused ports. When HAs have an equal number of unused ports, assign the host connection to the adapter that has the fewest connections for this workload.
   - If the number of required host connections is greater than the number of I/O enclosures, assign the additional connections to different HAs with the most unused ports within the I/O enclosures. When HAs have an equal number of unused ports, assign the host connection to the adapter that has the fewest connections for this workload.

4.6 Planning array sites

This step takes place after the DS8000 hardware is installed. The result of this step is a mapping of specific array site numbers (with associated disk drive type and DA pair number) to planned ranks.

During the DS8000 installation, array sites are dynamically created and assigned to DA pairs. Array site IDs (S*x*) do not have any fixed or predetermined relationship to disk drive physical locations or to the disk enclosure installation order. The relationship between array site IDs and physical disk locations or DA assignments can differ between the DS8000s, even on the DS8000s with the same number and type of disk drives.

After the DS8000 hardware is installed, you can use the output of the DS8000 command-line interface (DSCLI) `lsarraysite` command to display and document array site information, including disk drive type and DA pair. You must check the disk drive type and DA pair for each array site to ensure that arrays, ranks, and ultimately volumes created from the array site are created on the DS8000 hardware resources required for the isolated or resource-sharing workloads.

The result of this step is the addition of specific array site IDs to the plan of workload assignment to ranks.

Now, we look at an example of a DS8800 configuration to review the available disk drive types, DA pairs, and array sites and to describe several isolation and spreading considerations.

4.6.1 DS8800 configuration example: Array site planning considerations

In this configuration example, we present a DS8800 model with a fully populated base frame and three expansion frames. All disk enclosures are populated with disks and also SSDs (Figure 4-4 shows the reserved positions for SSDs in a DS8800 system). It also contains all 16 DA adapters and the third expansion frame. Figure 4-4 shows a schematic of this DS8000.

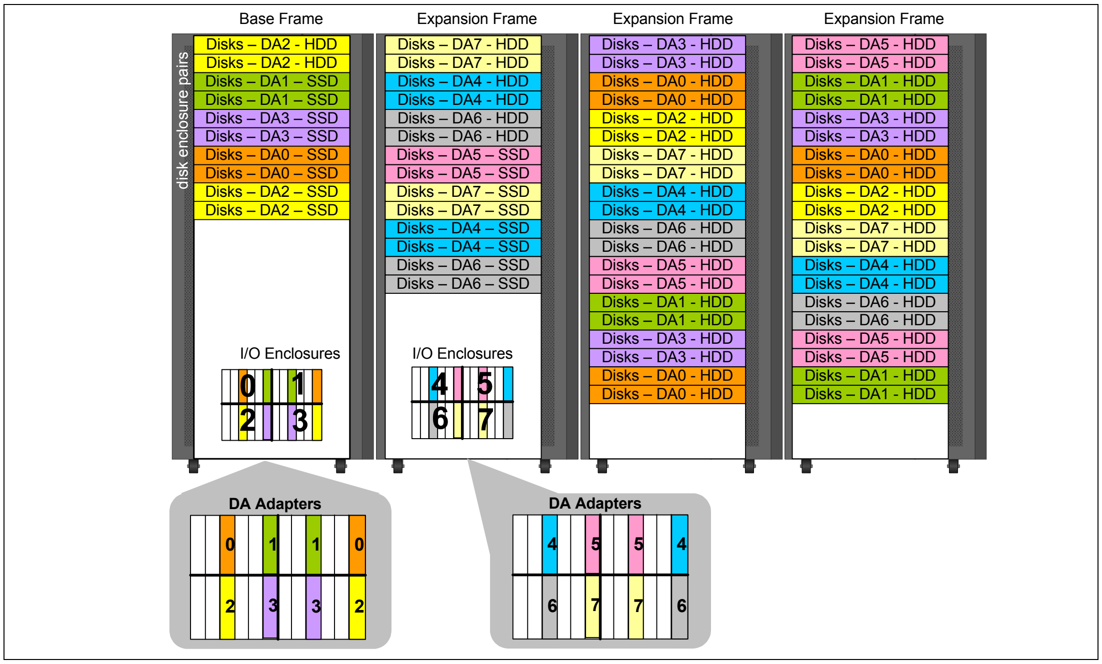


Figure 4-4 DS8800 configuration example with disk and I/O enclosures

Drives: In the schematic, the first two racks on the left (base frame and first expansion frame) contain the I/O enclosures. Each I/O enclosure can contain up to two DAs, a total of eight DAs per rack and 16 total in a full configuration and all expansions possible. The disk enclosure fillers for SSDs are also allocated on the first two racks on the left, but they are spread on all DAs. All other positions are for HDDs only.

Workload isolation considerations

In this example, all DA pairs are available for DS8800. A dedicated workload can be isolated at the DA level if needed.

If workload isolation is required, consider these options:

- One or more ranks can be dedicated to a workload.
- One or more DA pairs can be dedicated to a workload.
- When isolating workloads, you can still evaluate the use of Easy Tier and combine resources, such SSD, ENT, and NL disks.

The individual hardware configuration always needs to match your configuration and isolation plans first. In this example, you probably have SSD (if supplied) and HDD drives on each DA pair. When isolating an important workload to a dedicated DA pair with SSD and HDD ranks, you might use Easy Tier and multi-tier pools for the isolated workload (which can also be a set of individual workloads that share these resources and are isolated from the other workloads on the system). If a subset of ranks from a DA pair is set aside in this example for an isolated workload, understand that the other ranks on the DA pairs are assigned to other workloads. We must achieve an isolation for these workloads at the rank level. Remember that the DA pairs are shared between the workloads.

Workload spreading considerations

On a large and fully configured DS8000 system, as shown in Figure 4-4 on page 102, you can create various configurations, depending on your specific requirements in your environment related to workload, availability (for example, failure boundaries), and Copy Services considerations. Your considerations can include multiple managed or non-managed single-tier or multi-tier extent pools, extent pools for different storage types (FB or CKD), and different RAID types dedicated to isolated or resource-sharing workloads. The use of Easy Tier in this configuration is highly encouraged to provide optimum performance and minimize overall management efforts. Easy Tier supports optimum workload spreading across ranks within managed extent pools. The use of storage pool striping in non-managed extent pools is suggested. The resources for the particular workloads must be allocated in a way to ensure a balanced distribution of the overall system workload across all available system resources, as outlined in 4.2.3, “Workload spreading” on page 94. For additional considerations, see 4.8, “Planning extent pools” on page 115.

4.7 Planning RAID arrays and ranks

The next step is planning the RAID arrays and ranks. Take the specific array sites planned for isolated or resource-sharing workloads and define their assignments to RAID arrays and CKD or FB ranks, including planning array IDs and rank IDs. Because there is a one-to-one correspondence between an array and a rank on the DS8000, you can plan arrays and ranks in a single step. However, array creation and rank creation require separate steps.

Remember that the sequence of steps when creating the arrays and ranks finally determines the numbering scheme of array IDs and rank IDs, because these IDs are chosen automatically by the system during creation. The logical configuration does not depend on a specific ID numbering scheme, but a specific ID numbering scheme might help you plan the configuration and manage performance more easily.

Storage servers: Array sites, arrays, and ranks do not have a fixed or predetermined relationship to any DS8000 processor complex (storage server) before they are finally assigned to an extent pool and thus a rank group (rank group 0/1 is managed by processor complex 0/1).

4.7.1 RAID-level performance considerations

When configuring arrays from array sites, you need to specify the RAID level, either RAID 5, RAID 6, or RAID 10. These RAID levels meet different requirements for performance, usable storage capacity, and data protection. However, you must determine the correct RAID types and the physical disk drives (speed and capacity) related to initial workload performance objectives, capacity requirements, and availability considerations before you order the DS8000 hardware. For more information about implementing the various RAID levels on the DS8000, see 3.1, “RAID levels and spares” on page 52.

The following DS8800/8700 disk RAID types are available:

- DS8700 RAID Types:
  - SSDs are RAID 5 capable and RAID 10 only with a client request for price quotation (RPQ).

  - Enterprise disks are RAID 5, RAID 6, and RAID 10 capable.
  - Nearline disks are RAID 6 and RAID 10 capable.
- DS8800 RAID Types:
  - SSDs are RAID 5 capable and RAID 10 only with a client RPQ.
  - Enterprise disks are RAID 5, RAID 6, and RAID 10 capable.
  - Nearline disks are RAID 6 capable.

RAID 5 is one of the most commonly used levels of RAID protection, because it optimizes cost-effective performance while emphasizing usable capacity through data striping. It provides fault tolerance if one disk drive fails by using XOR parity for redundancy. Hot spots within an array are avoided by distributing data and parity information across all of the drives in the array. The capacity of one drive in the RAID array is lost for holding the parity information. RAID 5 provides a good balance of performance and usable storage capacity.

RAID 6 provides a higher level of fault tolerance than RAID 5 in disk failures but also provides less usable capacity than RAID 5, because the capacity of two drives in the array is set aside to hold the parity information. As with RAID 5, hot spots within an array are avoided by distributing data and parity information across all of the drives in the array. Still, RAID 6 offers more usable capacity than RAID 10 by providing an efficient method of data protection in double disk errors, such as two drive failures, two coincident medium errors, or a drive failure and a medium error during a rebuild. Because the likelihood of media errors increases with the capacity of the physical disk drives, consider the use of RAID 6 in conjunction with large capacity disk drives and higher data availability requirements. For example, consider RAID 6 where rebuilding the array in a drive failure takes a long time. RAID 6 can also be used with smaller serial-attached Small Computer System Interface (SCSI) (SAS) drives, when the primary concern is a higher level of data protection than is provided by RAID 5.

RAID 10 optimizes high performance while maintaining fault tolerance for disk drive failures. The data is striped across several disks, and the first set of disk drives is mirrored to an identical set. RAID 10 can tolerate at least one, and in most cases, multiple disk failures as long as the primary copy and the secondary copy of a mirrored disk pair do not fail at the same time.

In addition to the considerations for data protection and capacity requirements, the question typically arises about which RAID level performs better, RAID 5, RAID 6, or RAID 10. As with most complex issues, the answer is that it depends. There are a number of workload attributes that influence the relative performance of RAID 5, RAID 6, or RAID 10, including the use of cache, the relative mix of read as compared to write operations, and whether data is referenced randomly or sequentially.

Regarding read I/O operations, either random or sequential, there is generally no difference between RAID 5, RAID 6, and RAID 10. When a DS8000 subsystem receives a read request from a host system, it first checks whether the requested data is already in cache. If the data is in cache (that is, a read cache hit), there is no need to read the data from disk, and the RAID level on the arrays does not matter. For reads that must be satisfied from disk (that is, the array or the back end), the performance of RAID 5, RAID 6, and RAID 10 is roughly equal, because the requests are spread evenly across all disks in the array. In RAID 5 and RAID 6 arrays, data is striped across all disks, so I/Os are spread across all disks. In RAID 10, data is striped and mirrored across two sets of disks, so half of the reads are processed by one set of disks, and half of the reads are processed by the other set, reducing the utilization of individual disks.

Regarding random write I/O operations, the different RAID levels vary considerably in their performance characteristics. With RAID 10, each write operation at the disk back end initiates two disk operations to the rank. With RAID 5, an individual random small block write operation to the disk back end typically causes a RAID 5 write penalty, which initiates four I/O operations to the rank by reading the old data and the old parity block before finally writing the new data and the new parity block. For RAID 6 with two parity blocks, the write penalty increases to six required I/O operations at the back end for a single random small block write operation. This assumption is a worst-case scenario that is helpful for understanding the back-end impact of random workloads with a certain read:write ratio for the various RAID levels. It permits a rough estimate of the expected back-end I/O workload and helps you to plan for the correct number of arrays. On a heavily loaded system, it might take fewer I/O operations than expected on average for RAID 5 and RAID 6 arrays. The optimization of the queue of write I/Os waiting in cache for the next destage operation can lead to a high number of partial or full stripe writes to the arrays with fewer required back-end disk operations for the parity calculation.

On modern disk systems, such as the DS8000, write operations are generally cached by the storage subsystem and thus handled asynchronously with short write response times for the attached host systems. So, any RAID 5 or RAID 6 write penalties are generally shielded from the attached host systems in disk response time. Typically, a write request that is sent to the DS8000 subsystem is written into storage server cache and persistent cache, and the I/O operation is then acknowledged immediately to the host system as completed. As long as there is room in these cache areas, the response time seen by the application is only the time to get data into the cache, and it does not matter whether RAID 5, RAID 6, or RAID 10 is used. However, if the host systems send data to the cache areas faster than the storage server can destage the data to the arrays (that is, move it from cache to the physical disks), the cache can occasionally fill up with no space for the next write request, and therefore, the storage server signals the host system to retry the I/O write operation. In the time that it takes the host system to retry the I/O write operation, the storage server likely can destage part of the data, which provides free space in the cache and allows the I/O operation to complete on the retry attempt.

When random small block write data is destaged from cache to disk, RAID 5 and RAID 6 arrays can experience a severe write penalty with four or six required back-end disk operations. RAID 10 always requires only two disk operations per small block write request. Because RAID 10 performs only half the disk operations of RAID 5, for random writes, a RAID 10 destage completes faster and reduces the busy time of the disk subsystem. So, with steady and heavy random write workloads, the back-end write operations to the ranks (the physical disk drives) can become a limiting factor, so that only a RAID 10 configuration (instead of additional RAID 5 or RAID 6 arrays) provides enough back-end disk performance at the rank level to meet the workload performance requirements.

While RAID 10 clearly outperforms RAID 5 and RAID 6 in small-block random write operations, RAID 5 and also RAID 6 show excellent performance in sequential write I/O operations. With sequential write requests, all of the blocks required for the RAID 5 parity calculation can be accumulated in cache, and thus the destage operation with parity calculation can be dynamic as a full stripe write without the need for additional disk operations to the array. So, with only one additional parity block for a full stripe write (for example, seven data blocks plus one parity block for a 7+P RAID 5 array), RAID 5 requires less disk operation at the back end than a RAID 10, which always requires twice the write operations due to data mirroring. RAID 6 also benefits from sequential write patterns with most of the data blocks, which are required for the double parity calculation, staying in cache and thus reducing the amount of additional disk operations to the back end considerably. For sequential writes, a RAID 5 destage completes faster and reduces the busy time of the disk subsystem.

Comparing RAID 5 to RAID 6, the performance of small block random read and the performance of a sequential read are roughly equal. Due to the higher write penalty, the RAID 6 small block random write performance is explicitly less than with RAID 5. Also, the maximum sequential write throughput is slightly less with RAID 6 than with RAID 5 due to the additional second parity calculation. However, RAID 6 rebuild times are close to RAID 5 rebuild times (for the same size disk drive modules (DDMs)), because rebuild times are primarily limited by the achievable write throughput to the spare disk during data reconstruction. So, RAID 6 mainly is a significant reliability enhancement with a trade-off in random write performance. It is most effective for large capacity disks that hold mission-critical data and that are correctly sized for the expected write I/O demand. Workload planning is especially important before implementing RAID 6 for write-intensive applications, including Copy Services targets and FlashCopy SE repositories.

RAID 10 is not as commonly used as RAID 5 for two key reasons. First, RAID 10 requires more raw disk capacity for every GB of effective capacity. Second, when you consider a standard workload with a typically high number of read operations and only a few write operations, RAID 5 generally offers the best trade-off between overall performance and usable capacity. In many cases, RAID 5 write performance is adequate, because disk systems tend to operate at I/O rates below their maximum throughputs, and differences between RAID 5 and RAID 10 are primarily observed at maximum throughput levels. Consider RAID 10 for critical workloads with a high percentage of steady random write requests, which can easily become rank limited. RAID 10 provides almost twice the throughput as RAID 5 (because of the “write penalty”). The trade-off for better performance with RAID 10 is about 40% less usable disk capacity. Larger drives can be used with RAID 10 to get the random write performance benefit while maintaining about the same usable capacity as a RAID 5 array with the same number of disks.

We summarize the individual performance characteristics of the RAID arrays:

- For read operations from disk, either random or sequential, there is no significant difference in RAID 5, RAID 6, or RAID 10 performance.
- For random writes to disk, RAID 10 outperforms RAID 5 and RAID 6.
- For random writes to disk, RAID 5 performs better than RAID 6.
- For sequential writes to disk, RAID 5 tends to perform better.

Table 4-1 shows a short overview of the advantages and disadvantages for the RAID level reliability, space efficiency, and random write performance.

Table 4-1 RAID-level comparison of reliability, space efficiency, and write penalty

RAID level	Reliability (number of erasures)	Space efficiency[a]	Performance write penalty (number of disk operations)
RAID 5 (7+P)	1	87.5%	4
RAID 6 (6+P+Q)	2	75%	6
RAID 10 (4x2)	At least 1	50%	2

a. The space efficiency in this table is based on the number of disks that remain available for data storage. The actual usable, decimal capacities are up to 5% less.

In general, workloads that effectively use storage subsystem cache for reads and writes see little difference between RAID 5 and RAID 10 configurations. For workloads that perform better with RAID 5, the difference in RAID 5 performance over RAID 10 is typically small. However, for workloads that perform better with RAID 10, the difference in RAID 10 performance over RAID 5 performance or RAID 6 performance can be significant.

Because RAID 5, RAID 6, and RAID 10 perform equally well for both random and sequential *read* operations, RAID 5 or RAID 6 might be a good choice for space efficiency and performance for standard workloads with many read requests. RAID 6 offers a higher level of data protection than RAID 5, especially for large capacity drives, but the random write performance of RAID 6 is less due to the second parity calculation. Therefore, we highly suggest that you size for performance, especially for RAID 6.

RAID 5 tends to have a slight performance advantage for sequential writes. RAID 10 performs better for random writes. RAID 10 is generally considered to be the RAID type of choice for business-critical workloads with many random write requests (typically more than 35% writes) and low response time requirements.

For array rebuilds, RAID 5, RAID 6, and RAID 10 require approximately the same elapsed time, although RAID 5 and RAID 6 require more disk operations and therefore are more likely to affect other disk activity on the same disk array.

You can select RAID types for each array site. So, you can select the RAID type based on the specific performance requirements of the data for that site. The best way to compare the performance of a specific workload that uses RAID 5, RAID 6, or RAID 10 is to run a Disk Magic model. For additional information about the capabilities of this tool, see 6.1, "Disk Magic" on page 176.

For workload planning purposes, it might be convenient to have a general idea of the I/O performance that a single RAID array can provide. Figure 4-5 on page 108 and Figure 4-6 on page 109 show measurement results[1] for a single RAID array built from eight 3.5 inch large form factor (LFF) 146 GB 15k FC disk drives when configured as RAID 5, RAID 6, or RAID 10.

[1] The measurements were done with IOmeter (http://www.iometer.org) on Microsoft Windows Server 2003 utilizing the entire available capacity on the array for I/O requests. The performance data that is contained here was obtained in a controlled, isolated environment. Actual results that might be obtained in other operating environments can vary. There is no guarantee that the same or similar results can be obtained elsewhere.

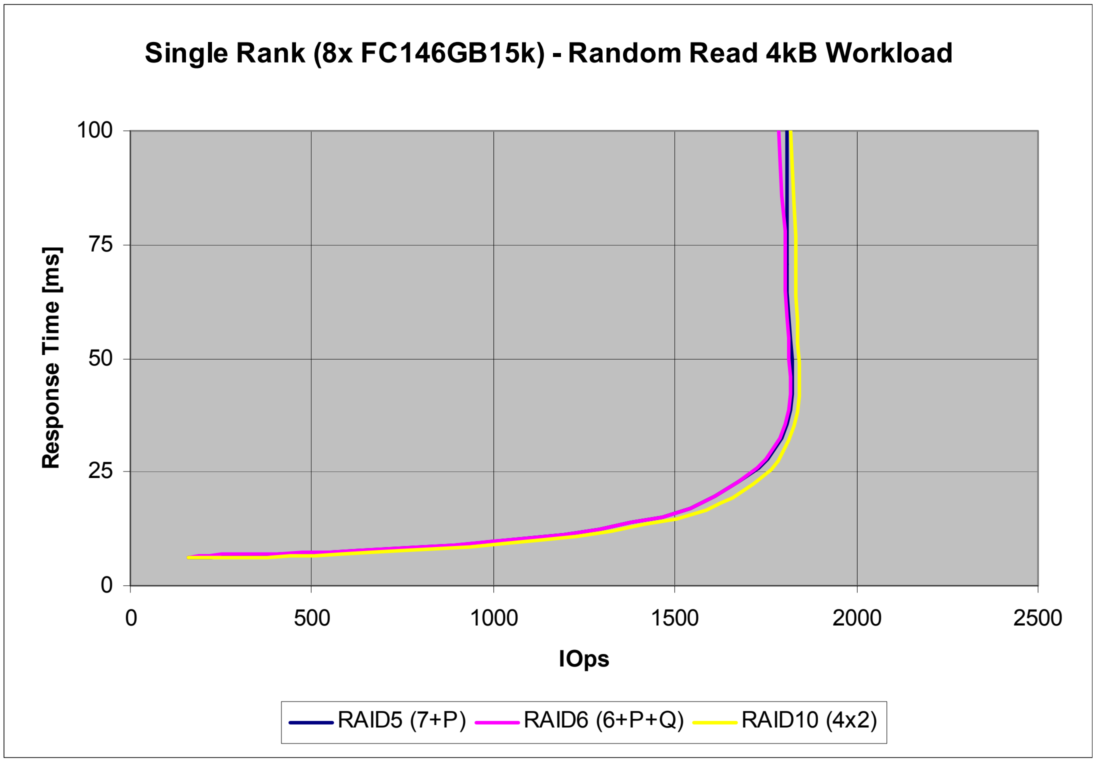


Figure 4-5 Single rank RAID-level comparison for a 4 KB random workload with 100% reads

These numbers are not DS8000 specific, because they represent the limits that you can expect from a simple set of eight physical disks that form a RAID array. As outlined in 2.4.5, "SAS 2.5 inch drives compared to FC 3.5 inch drives" on page 39, you can expect slightly higher maximum I/O rates with 2.5 inch small form factor (SFF) 146 GB 15k SAS drives as used in DS8800 systems.

For small block random read workloads, there is no significant performance difference between RAID 5, RAID 6, and RAID 10, as seen in Figure 4-5. Without considering read cache hits, 1800 read IOPS with high back-end response times above 30 ms mark the upper limit of the capabilities of a single array for random read access by using all of the available capacity of that array.

Small-block random writes, however, make the difference between the various RAID levels and performance. For a typical 70:30 random small block workload with 70% reads (no read cache hits) and 30% writes, as shown in Figure 4-6 on page 109, the different performance characteristics between the RAID levels become evident. With an increasing number of random writes, RAID 10 clearly outperforms RAID 5 and RAID 6. For a standard random small block 70:30 workload, 1500 IOPS mark the upper limit of a RAID 10 array, 1100 IOPS for a RAID 5 array, and 900 IOPS for a RAID 6 array.

In both Figure 4-5 and Figure 4-6 on page 109, no read cache hits are considered. Furthermore, the I/O requests are spread across the entire available capacity of each RAID array. So, depending on the read cache hit ratio of a workload and the capacity used on the array (using less capacity on an array means reducing disk arm movements and reducing average access times), you can expect typically lower overall response times and higher I/O rates. Also, the read:write ratio, as well as the access pattern of a particular workload, either random or sequential, determine the achievable performance of a rank. Figure 4-5 and Figure 4-6 on page 109 (examples of small block random I/O requests) help show the performance capabilities of a single rank for different RAID levels.

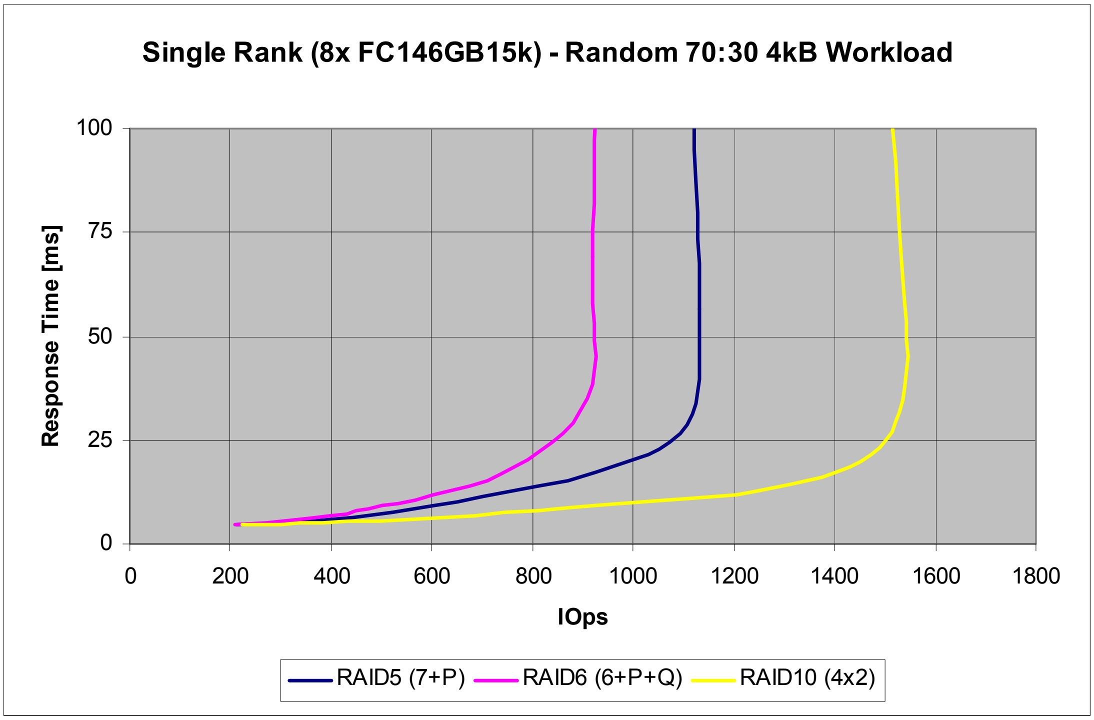


Figure 4-6 Single rank RAID-level comparison for a 4 KB random workload with 70% reads 30% writes

Despite the different RAID levels and the actual workload pattern (read:write ratio, sequential access, or random access), the limits of the maximum I/O rate per rank also depend on the type of disk drives used. As a mechanical device, each disk drive can process a limited number of random I/O operations per second, depending on the drive characteristics. So the number of disk drives used for a specific amount of storage capacity finally determines the achievable random IOPS performance. The 15k drives offer approximately 30% more random IOPS performance than 10k drives. Generally, for random IOPS planning calculations, you can use 160 IOPS per 15k FC drive and 120 IOPS per 10k FC drive. At these levels of disk utilization, you might see already elevated response times. So for excellent response time expectations, consider lower IOPS limits. Low spinning, large-capacity nearline disk drives offer a considerably lower maximum random access I/O rate per drive (approximately half of a 15k drive). Therefore, they are only intended for environments with fixed content, data archival, reference data, or for near-line applications that require large amounts of data at low cost.

4.7.2 RAID array considerations

A DS8000 RAID array supports only a single RAID type (either RAID 5, RAID 6, or RAID 10). If different RAID types are required for certain workloads, the workloads must be isolated in different RAID arrays.

On the DS8000, *four* spare drives (the minimum) are required in a fully populated DA pair, so certain arrays contain spares:

- RAID 5: 6+P+S
- RAID 6: 5+P+Q+S
- RAID 10: 3x2+2S

And, other arrays do not contain any spares:

- RAID 5: 7+P
- RAID 6: 6+P+Q
- RAID 10: 4x2

This requirement leads to arrays with different storage capacities and performance characteristics although they were created from array sites with identical disk types and RAID levels. The spares are assigned during array creation. Typically, the first arrays created from an unconfigured set of ranks on a certain DA pair contain spare drives until the minimum requirements, as outlined in 3.1.5, “Spare creation” on page 56, are met.

Regarding the distribution of the spare drives, you might need to plan the sequence of array creation carefully if a mixture of RAID 5, RAID 6, and RAID 10 arrays are required on the same DA pair. Otherwise, you might not meet your initial capacity requirements and end up with more spare drives on the system than required, wasting storage capacity. For example, if you plan for two RAID 10 (3x2+2S) arrays on a certain DA pair with homogeneous array sites, create these arrays first, because these arrays already reserve two spare drives per array, so that the final RAID 5 or RAID 6 arrays do not contain spares. Or, if you prefer to obtain RAID 10 (4x2) arrays without spare drives, you can instead create four RAID 5 or RAID 6 arrays, which then contain the required number of spare drives before creating the RAID 10 arrays.

To spread the available storage capacity and thus the overall workload evenly across both DS8000 processor complexes, you must assign an equal number of arrays that contain spares to processor complex 0 (rank group 0) and processor complex 1 (rank group 1).

Performance can differ between RAID arrays that contain spare drives and RAID arrays without spare drives, because the arrays without spare drives offer more storage capacity and also provide more active disk spindles for processing I/O operations.

> **Important**: You must confirm spare allocation after array creation and adjust the logical configuration plan *before* creating ranks and assigning them to extent pools.

When creating arrays from array sites, it might help to order the array IDs by DA pair, array size (that is, arrays with or without spares), RAID level, or disk type, depending on the available hardware resources and workload planning considerations.

You can take the mapping of the array sites to particular DA pairs from the output of the DSCLI **`lsarraysite`** command. Array sites are numbered starting with S1, S2, and S3 by the DS8000 microcode. Arrays are numbered with system-generated IDs starting with A0, A1, and A2 in the sequence that they are created.

The following examples provide basic understanding about how the sequence of creation of RAID arrays from array sites can determine the sequence of array IDs. A certain sequence of array IDs can be convenient for different logical configuration approaches. Two approaches are described, one for sorting the arrays by DA pair and one for sorting the arrays by capacity and cycling through the available DA pairs in a round-robin fashion on a homogeneously equipped DS8000 system. However, the sequence of array IDs and rank IDs is not important for the logical configuration and typically depends on your hardware configuration. Especially with different disk drives and RAID levels on the system and the use of Easy Tier and multi-tier extent pools, the sequence is less important and can be automatic with the DS GUI or any way that is convenient for the administrator. The following examples are related to a partially configured DS8700 system with typically a multiple of eight ranks per DA pair. A DS8800 system typically shows a multiple of six ranks per DA pair in partially configured systems.

Array ID sequence sorted by DA pair

One approach to arrange the array sequence is to sort the arrays by DA pair, which gives a quick overview of the array to DA pair association and can help to make the logical configuration easier. This approach was used with early DS8000 models when using single-rank extent pools and a maximum control of volume placement and manual performance management. In this case, create the arrays (A0, A1, and A2, for example) ordered by DA pair, which in most cases is following the sequence of array sites (S1, S2, and S3, for example), as shown in Example 4-1. The sequence of array sites is initially determined by the system and might not always strictly follow the DA pair order. For more information about this configuration strategy with single-rank extent pools, a hardware-related volume, and the logical subsystem (LSS, for example)/logical control unit (LCU) ID configuration concept, see 4.9, “Planning address groups, LSSs, volume IDs, and CKD PAVs” on page 140.

Example 4-1 Array ID sequence sorted by DA pair

```
dscli> lsarray -l
Array State      Data   RAIDtype    arsite rank DA Pair DDMcap (10^9B) diskclass encrypt
=========================================================================================
A0    Unassigned Normal 6 (5+P+Q+S) S1     -    2                146.0 ENT unsupported
A1    Unassigned Normal 6 (5+P+Q+S) S2     -    2                146.0 ENT unsupported
A2    Unassigned Normal 6 (5+P+Q+S) S3     -    2                146.0 ENT unsupported
A3    Unassigned Normal 6 (5+P+Q+S) S4     -    2                146.0 ENT unsupported
A4    Unassigned Normal 6 (6+P+Q)   S5     -    2                146.0 ENT unsupported
A5    Unassigned Normal 6 (6+P+Q)   S6     -    2                146.0 ENT unsupported
A6    Unassigned Normal 6 (6+P+Q)   S7     -    2                146.0 ENT unsupported
A7    Unassigned Normal 6 (6+P+Q)   S8     -    2                146.0 ENT unsupported
A8    Unassigned Normal 6 (5+P+Q+S) S9     -    6                146.0 ENT unsupported
A9    Unassigned Normal 6 (5+P+Q+S) S10    -    6                146.0 ENT unsupported
A10   Unassigned Normal 6 (5+P+Q+S) S11    -    6                146.0 ENT unsupported
A11   Unassigned Normal 6 (5+P+Q+S) S12    -    6                146.0 ENT unsupported
A12   Unassigned Normal 6 (6+P+Q)   S13    -    6                146.0 ENT unsupported
A13   Unassigned Normal 6 (6+P+Q)   S14    -    6                146.0 ENT unsupported
A14   Unassigned Normal 6 (6+P+Q)   S15    -    6                146.0 ENT unsupported
A15   Unassigned Normal 6 (6+P+Q)   S16    -    6                146.0 ENT unsupported
A16   Unassigned Normal 6 (5+P+Q+S) S17    -    7                146.0 ENT unsupported
A17   Unassigned Normal 6 (5+P+Q+S) S18    -    7                146.0 ENT unsupported
A18   Unassigned Normal 6 (5+P+Q+S) S19    -    7                146.0 ENT unsupported
A19   Unassigned Normal 6 (5+P+Q+S) S20    -    7                146.0 ENT unsupported
A20   Unassigned Normal 6 (6+P+Q)   S21    -    7                146.0 ENT unsupported
A21   Unassigned Normal 6 (6+P+Q)   S22    -    7                146.0 ENT unsupported
A22   Unassigned Normal 6 (6+P+Q)   S23    -    7                146.0 ENT unsupported
A23   Unassigned Normal 6 (6+P+Q)   S24    -    7                146.0 ENT unsupported
```

Array ID sequence sorted by array size and cycling through DA pairs

For *initial* configurations that use homogeneous multi-rank extent pools on a storage unit, especially with a homogeneous hardware base (for example, a single type of DDMs that use only one RAID level) and resource-sharing workloads, consider configuring the arrays in a round-robin fashion across all available DA pairs. Create the first array from the first array site on the first DA pair, then the second array from the first array site on the second DA pair, and so on. This sequence also sorts the arrays by array size (that is, arrays *with* or *without* spares), creating the smaller capacity arrays with spare drives first, as shown in Example 4-2 on page 112. If the ranks are finally created in the same ascending ID sequence from the arrays, the rank ID sequence also cycles through all DA pairs in a round-robin fashion, as seen in Example 4-3 on page 112, which might enhance the distribution of volumes across ranks from different DA pairs within multi-rank extent pools. The creation of successive volumes (using the *rotate volumes* EAM) or extents (using the *rotate extents* EAM) within a multi-rank extent pool also follows the ascending numerical sequence of rank IDs.

Example 4-2 shows the array ID sequence sorted by array size (with and without spares) and cycling through all available DA pairs.

Example 4-2 Array ID sequence sorted by array size and cycling through all available DA pairs

```
dscli> lsarray -l
Array State      Data   RAIDtype    arsite rank DA Pair DDMcap (10^9B) diskclass encrypt
=========================================================================================
A0    Unassigned Normal 6 (5+P+Q+S) S1     -    2                146.0 ENT unsupported
A1    Unassigned Normal 6 (5+P+Q+S) S9     -    6                146.0 ENT unsupported
A2    Unassigned Normal 6 (5+P+Q+S) S17    -    7                146.0 ENT unsupported
A3    Unassigned Normal 6 (5+P+Q+S) S2     -    2                146.0 ENT unsupported
A4    Unassigned Normal 6 (5+P+Q+S) S10    -    6                146.0 ENT unsupported
A5    Unassigned Normal 6 (5+P+Q+S) S18    -    7                146.0 ENT unsupported
A6    Unassigned Normal 6 (5+P+Q+S) S3     -    2                146.0 ENT unsupported
A7    Unassigned Normal 6 (5+P+Q+S) S11    -    6                146.0 ENT unsupported
A8    Unassigned Normal 6 (5+P+Q+S) S19    -    7                146.0 ENT unsupported
A9    Unassigned Normal 6 (5+P+Q+S) S4     -    2                146.0 ENT unsupported
A10   Unassigned Normal 6 (5+P+Q+S) S12    -    6                146.0 ENT unsupported
A11   Unassigned Normal 6 (5+P+Q+S) S20    -    7                146.0 ENT unsupported
A12   Unassigned Normal 6 (6+P+Q)   S5     -    2                146.0 ENT unsupported
A13   Unassigned Normal 6 (6+P+Q)   S13    -    6                146.0 ENT unsupported
A14   Unassigned Normal 6 (6+P+Q)   S21    -    7                146.0 ENT unsupported
A15   Unassigned Normal 6 (6+P+Q)   S6     -    2                146.0 ENT unsupported
A16   Unassigned Normal 6 (6+P+Q)   S14    -    6                146.0 ENT unsupported
A17   Unassigned Normal 6 (6+P+Q)   S22    -    7                146.0 ENT unsupported
A18   Unassigned Normal 6 (6+P+Q)   S7     -    2                146.0 ENT unsupported
A19   Unassigned Normal 6 (6+P+Q)   S15    -    6                146.0 ENT unsupported
A20   Unassigned Normal 6 (6+P+Q)   S23    -    7                146.0 ENT unsupported
A21   Unassigned Normal 6 (6+P+Q)   S8     -    2                146.0 ENT unsupported
A22   Unassigned Normal 6 (6+P+Q)   S16    -    6                146.0 ENT unsupported
A23   Unassigned Normal 6 (6+P+Q)   S24    -    7                146.0 ENT unsupported
```

Depending on the installed hardware resources in the DS8000 storage subsystem, you might have different numbers of DA pairs and different numbers of arrays per DA pair. Also, you might not be able to strictly follow your initial array ID numbering scheme anymore when upgrading storage capacity by adding array sites to the storage unit later.

4.7.3 Rank considerations

A DS8000 rank supports only a single storage type (either CKD or FB). If workloads require different storage types (CKD and FB), the workloads must be isolated to different DS8000 ranks. Ranks are numbered with system-generated IDs starting with R0, R1, and R2 in the sequence in which they are created. For ease of management and performance analysis, it might be preferable to create the ranks by following the order of the ascending array ID sequence, so that, for example, rank R27 can be associated with array A27.

The association between ranks, arrays, array sites, and DA pairs can be taken from the output of the DSCLI command **lsarray -l**, as shown in Example 4-3, for a DS8700 system with a homogeneous disk configuration.

Example 4-3 Rank, array, array site, and DA pair association as shown by the lsarray -l command on DS8700

```
dscli> lsarray -l
Array State    Data   RAIDtype    arsite rank DA Pair DDMcap (10^9B) diskclass encrypt
=========================================================================================
A0    Assigned Normal 6 (5+P+Q+S) S1     R0    2                146.0 ENT       unsupported
A1    Assigned Normal 6 (5+P+Q+S) S9     R1    6                146.0 ENT       unsupported
A2    Assigned Normal 6 (5+P+Q+S) S17    R2    7                146.0 ENT       unsupported
```

```
A3    Assigned Normal 6 (5+P+Q+S) S2     R3    2                146.0 ENT       unsupported
A4    Assigned Normal 6 (5+P+Q+S) S10    R4    6                146.0 ENT       unsupported
A5    Assigned Normal 6 (5+P+Q+S) S18    R5    7                146.0 ENT       unsupported
A6    Assigned Normal 6 (5+P+Q+S) S3     R6    2                146.0 ENT       unsupported
A7    Assigned Normal 6 (5+P+Q+S) S11    R7    6                146.0 ENT       unsupported
A8    Assigned Normal 6 (5+P+Q+S) S19    R8    7                146.0 ENT       unsupported
A9    Assigned Normal 6 (5+P+Q+S) S4     R9    2                146.0 ENT       unsupported
A10   Assigned Normal 6 (5+P+Q+S) S12    R10   6                146.0 ENT       unsupported
A11   Assigned Normal 6 (5+P+Q+S) S20    R11   7                146.0 ENT       unsupported
A12   Assigned Normal 6 (6+P+Q)   S5     R12   2                146.0 ENT       unsupported
A13   Assigned Normal 6 (6+P+Q)   S13    R13   6                146.0 ENT       unsupported
A14   Assigned Normal 6 (6+P+Q)   S21    R14   7                146.0 ENT       unsupported
A15   Assigned Normal 6 (6+P+Q)   S6     R15   2                146.0 ENT       unsupported
A16   Assigned Normal 6 (6+P+Q)   S14    R16   6                146.0 ENT       unsupported
A17   Assigned Normal 6 (6+P+Q)   S22    R17   7                146.0 ENT       unsupported
A18   Assigned Normal 6 (6+P+Q)   S7     R18   2                146.0 ENT       unsupported
A19   Assigned Normal 6 (6+P+Q)   S15    R19   6                146.0 ENT       unsupported
A20   Assigned Normal 6 (6+P+Q)   S23    R20   7                146.0 ENT       unsupported
A21   Assigned Normal 6 (6+P+Q)   S8     R21   2                146.0 ENT       unsupported
A22   Assigned Normal 6 (6+P+Q)   S16    R22   6                146.0 ENT       unsupported
A23   Assigned Normal 6 (6+P+Q)   S24    R23   7                146.0 ENT       unsupported
```

Example 4-4 is another example for the DSCLI command **lsarray -l** on a DS8800 system with different disk types, RAID levels, and storage classes (SSD, ENT, and NL). You can see that the rank ID sequence does not necessarily follow the array ID sequence. There are different numbers of arrays per DA pair, for example, six SFF HDD arrays on DA3 and DA1, six SFF and three LFF HDD arrays on DA2, and two SSD arrays on DA0.

Example 4-4 Rank, array, array site, and DA pair association as shown by the lsarray -l command on DS8800

```
dscli> lsarray -l
Array State    Data   RAIDtype    arsite Rank DA Pair DDMcap (10^9B) diskclass encrypt
=========================================================================================
A0    Assigned Normal 5 (6+P+S)   S1     R0   0                300.0 SSD       unsupported
A1    Assigned Normal 5 (6+P+S)   S2     R1   0                300.0 SSD       unsupported
A2    Assigned Normal 5 (6+P+S)   S3     R17  2                600.0 ENT       unsupported
A3    Assigned Normal 5 (6+P+S)   S4     R18  2                600.0 ENT       unsupported
A4    Assigned Normal 5 (6+P+S)   S5     R26  2                600.0 ENT       unsupported
A5    Assigned Normal 5 (6+P+S)   S6     R20  2                600.0 ENT       unsupported
A6    Assigned Normal 5 (7+P)     S7     R2   1                146.0 ENT       unsupported
A7    Assigned Normal 5 (6+P+S)   S8     R3   1                146.0 ENT       unsupported
A8    Assigned Normal 5 (6+P+S)   S9     R6   1                146.0 ENT       unsupported
A9    Assigned Normal 5 (6+P+S)   S10    R7   1                146.0 ENT       unsupported
A10   Assigned Normal 5 (7+P)     S11    R8   1                146.0 ENT       unsupported
A11   Assigned Normal 5 (6+P+S)   S12    R9   1                146.0 ENT       unsupported
A12   Assigned Normal 5 (7+P)     S13    R5   2                600.0 ENT       unsupported
A13   Assigned Normal 5 (7+P)     S14    R16  2                600.0 ENT       unsupported
A14   Assigned Normal 6 (5+P+Q+S) S15    R21  2               3000.0 NL        unsupported
A15   Assigned Normal 6 (5+P+Q+S) S16    R23  2               3000.0 NL        unsupported
A16   Assigned Normal 6 (5+P+Q+S) S17    R22  2               3000.0 NL        unsupported
A18   Assigned Normal 5 (7+P)     S19    R10  3                146.0 ENT       unsupported
A19   Assigned Normal 5 (6+P+S)   S20    R11  3                146.0 ENT       unsupported
A20   Assigned Normal 5 (6+P+S)   S21    R12  3                146.0 ENT       unsupported
A21   Assigned Normal 5 (7+P)     S22    R13  3                146.0 ENT       unsupported
A22   Assigned Normal 5 (6+P+S)   S23    R14  3                146.0 ENT       unsupported
A23   Assigned Normal 5 (6+P+S)   S24    R15  3                146.0 ENT       unsupported
```

The DSCLI command **lsrank -l**, as illustrated in Example 4-5, shows the capacity of the ranks, and after their assignment to extent pools, the association to extent pools and rank groups. This information is important for later configuring the extent pools for the planned workload and capacity requirements.

Unassigned ranks do not have a fixed or predetermined relationship to any DS8000 processor complex. Each rank can be assigned to any extent pool or any rank group. Only when assigning a rank to an extent pool and thus rank group 0 or rank group 1 does the rank become associated with processor complex 0 or processor complex 1. Ranks from rank group 0 (even-numbered extent pools: P0, P2, and P4, for example) are managed by processor complex 0, and ranks from rank group 1 (odd-numbered extent pools: P1, P3, and P5, for example) are managed by processor complex 1.

For a balanced distribution of the overall workload across both processor complexes, half of the ranks must be assigned to rank group 0 and half of the ranks must be assigned to rank group 1. Also, the ranks with and without spares must be spread evenly across both rank groups. Furthermore, it is important that the ranks (from each DA pair) are distributed evenly across both processor complexes; otherwise, you might seriously limit the available back-end bandwidth and thus the overall throughput of the system. If, for example, all ranks of a DA pair are assigned to only one processor complex, only one DA card of the DA pair is used to access the set of ranks, and thus, only half of the available DA pair bandwidth is available.

This practice also is especially important with *solid-state drives* and small block random I/O workloads. To be able to use the full back-end random I/O performance of two (or more) SSD ranks within a certain DA pair, the SSD I/O workload must be balanced across both DAs of the DA pair. This balance can be achieved by assigning half of the SSD ranks of each DA pair to even extent pools (P0, P2, and P4, managed by storage server 0 or rank group 0) and half of the SSD ranks to odd extent pools (P1, P3, and P5, managed by storage server 1 or rank group 1). If, for example, all SSD ranks of the same DA pair are assigned to the same rank group (for example, rank group 0 with extent pool P0 and P2), only one DA card of the DA pair is used to access this set of SSD ranks, providing only half of the available I/O processing capability of the DA pair and severely limiting the overall SSD performance.

Example 4-5 Rank, array, and capacity information as shown by the lsrank -l command on DS8700

```
dscli> lsrank -l
ID  Group State      datastate array RAIDtype extpoolID extpoolnam    stgtype exts usedexts encryptgrp
======================================================================================================
R0      - Unassigned Normal    A0           6 -         -             fb      634        -          -
R1      - Unassigned Normal    A1           6 -         -             fb      634        -          -
R2      - Unassigned Normal    A2           6 -         -             fb      634        -          -
R3      - Unassigned Normal    A3           6 -         -             fb      634        -          -
R4      - Unassigned Normal    A4           6 -         -             fb      634        -          -
R5      - Unassigned Normal    A5           6 -         -             fb      634        -          -
R6      - Unassigned Normal    A6           6 -         -             fb      634        -          -
R7      - Unassigned Normal    A7           6 -         -             fb      634        -          -
R8      - Unassigned Normal    A8           6 -         -             fb      634        -          -
R9      - Unassigned Normal    A9           6 -         -             fb      634        -          -
R10     - Unassigned Normal    A10          6 -         -             fb      634        -          -
R11     - Unassigned Normal    A11          6 -         -             fb      634        -          -
R12     - Unassigned Normal    A12          6 -         -             fb      763        -          -
R13     - Unassigned Normal    A13          6 -         -             fb      763        -          -
R14     - Unassigned Normal    A14          6 -         -             fb      763        -          -
R15     - Unassigned Normal    A15          6 -         -             fb      763        -          -
R16     - Unassigned Normal    A16          6 -         -             fb      763        -          -
R17     - Unassigned Normal    A17          6 -         -             fb      763        -          -
R18     - Unassigned Normal    A18          6 -         -             fb      763        -          -
R19     - Unassigned Normal    A19          6 -         -             fb      763        -          -
```

```
R20     - Unassigned Normal    A20           6 -         -             fb       763        -         -
R21     - Unassigned Normal    A21           6 -         -             fb       763        -         -
R22     - Unassigned Normal    A22           6 -         -             fb       763        -         -
R23     - Unassigned Normal    A23           6 -         -             fb       763        -         -
```

4.8 Planning extent pools

After planning the arrays and the ranks, the next step is to plan the extent pools, which means taking the planned ranks and defining their assignment to extent pools and rank groups, including planning the extent pool IDs.

Extent pools are automatically numbered with system-generated IDs starting with P0, P1, and P2 in the sequence in which they are created. Extent pools that are created for *rank group 0* are managed by processor complex 0 and have even-numbered IDs (P0, P2, and P4, for example). Extent pools that are created for *rank group 1* are managed by processor complex 1 and have odd-numbered IDs (P1, P3, and P5, for example). Only in a failure condition or during a concurrent code load is the ownership of a certain rank group temporarily moved to the alternate processor complex.

To achieve a uniform storage system I/O performance and avoid single resources that become bottlenecks (called "*hot spots*"), it is preferable to distribute volumes and workloads evenly across all of the ranks (disk spindles) and DA pairs that are dedicated to a workload by creating appropriate extent pool configurations.

The assignment of the ranks to extent pools together with an appropriate concept for the logical configuration and volume layout is the most essential step to optimize overall storage system performance. A rank can be assigned to any extent pool or rank group. Each rank provides a particular number of storage extents of a certain storage type (either FB or CKD) to an extent pool. An extent pool finally aggregates the extents from the assigned ranks and provides the logical storage capacity for the creation of logical volumes for the attached host systems. For more information about effective capacity and extents, see 8.5.2, "Disk capacity" in *IBM System Storage DS8000: Architecture and Implementation*, SG24-8886.

When an appropriate DS8000 hardware base is selected for the planned workloads (that is, isolated and resource-sharing workloads), the next goal is to provide a logical configuration concept. This concept widely guarantees a balanced workload distribution across all available hardware resources within the storage subsystem at any time - from the beginning, when only part of the available storage capacity is used, up to the end, when almost all of the capacity of the storage system is allocated.

On the DS8000, we can configure homogeneous single-tier extent pools, with ranks of the same storage class, and hybrid multi-tier extent pools with ranks from different storage classes. The EAMs, such as *rotate extents* or *storage pool striping,* provide easy to use capacity-based methods of spreading the workload data across the ranks in an extent pool. Furthermore, the use of Easy Tier automatic mode to automatically manage and maintain an optimal workload distribution across these resources over time provides excellent workload spreading with the best performance at a minimum administrative effort.

In addition, the various extent pool configurations (homogeneous or hybrid pools, managed or not managed by Easy Tier) can further be combined with the DS8000 I/O Priority Manager to prioritize workloads that are sharing resources to meet QoS goals in cases when resource contention might occur.

In the following sections, we present concepts for the configuration of single-tier and multi-tier extent pools to spread the workloads evenly across the available hardware resources. Also, the benefits of Easy Tier with different extent pool configurations are outlined. Unless otherwise noted, we assume that enabling Easy Tier automatic mode refers to enabling the automatic management capabilities of Easy Tier and Easy Tier monitoring.

4.8.1 Overview of DS8000 extent pool configurations with Easy Tier

Figure 4-7 on page 117 provides a brief overview of different extent pool configurations on DS8800 and DS8700 systems by comparing the management effort and the benefits of Easy Tier. We present four generic extent pool configurations based on the same hardware base in this example. We take step-by-step advantage of automatic storage performance and storage economics management by using Easy Tier.

Starting on the left, we show a DS8000 configuration (red color) with multiple homogeneous extent pools of different storage tiers with the Easy Tier automatic mode not enabled (or without the Easy Tier feature). With dedicated storage tiers bound to individual extent pools, the suitable extent pool for each volume must be chosen manually based on workload requirements. Furthermore, the performance and workload in each extent pool must be closely monitored and managed down to the rank level to adapt to workload changes over time, This monitoring increases overall management effort considerably. Depending on workload changes or application needs, workloads and volumes must be migrated from one highly utilized extent pool to another less utilized extent pool or from a lower storage tier to a higher storage tier. Easy Tier manual mode can help to easily and dynamically migrate a volume from one extent pool to another. However, data placement across tiers must be managed manually and occurs on the volume level only, which might, for example, waste costly SSD capacity. Typically, only a part of the capacity of a specific volume is hot and best suited for SSD placement. The workload can become imbalanced across the ranks within an extent pool and limit the overall performance even with storage pool striping due to natural workload skew. Workload spreading within a pool is only based on spreading the volume capacity evenly across the ranks, not taking any data access patterns or performance statistics into account. After adding new capacity to an existing extent pool, you must restripe the volume data within an extent pool manually, for example, by using *manual volume rebalance*, to maintain a balanced workload distribution across all ranks in a specific pool.

As shown in the second configuration (orange color), we can easily optimize the first configuration and already reduce management efforts considerably by enabling Easy Tier automatic mode. Thus, we automate *intra-tier* performance management (*auto-rebalance*) in these homogeneous single-tier extent pools. Easy Tier controls the workload spreading within each extent pool and automatically relocates data across the ranks based on rank utilization to minimize skew and avoid rank hot spots. Performance management is shifted from rank to extent pool level with correct data placement across tiers and extent pools at the volume level. Furthermore, when adding new capacity to an existing extent pool, Easy Tier automatic mode automatically takes advantage of the capacity and performance capabilities of the new ranks without the need for manual interaction.

We can further reduce management effort by merging extent pools and building different combinations of 2-tier hybrid extent pools, as shown in the third configuration (blue color). We introduce an automatically managed tiered storage architecture but still isolate, for example, our high performance production workload from our development/test environment. We introduce an ENT/SSD pool for our high performance and high priority production workload, efficiently boosting ENT performance with SSDs and automate storage tiering from enterprise-class drives to SSDs by using Easy Tier automated cross-tier data relocation and *storage performance* optimization at the subvolume level. And, we create an ENT/NL pool for our development/test environment or other enterprise-class applications to maintain enterprise-class performance while shrinking the footprint and reducing costs by combining enterprise-class drives with large-capacity nearline drives that use Easy Tier automated cross-tier data relocation and *storage economics* management. In addition, we also benefit from a balanced workload distribution across all ranks within each drive tier due to the Easy Tier intra-tier optimization and automatically take advantage of the capacity and performance capabilities of new ranks when new capacity is added to an existing pool without the need for manual interaction.

The minimum management effort combined with the highest amount of automated storage optimization can be achieved by creating 3-tier hybrid extent pools and by using Easy Tier automatic mode across all three tiers, as shown in the fourth configuration (green color). We use the most efficient way of automated data relocation to the appropriate storage tier with automatic *storage performance* and *storage economics* optimization on the subvolume level at a minimum administrative effort. We use automated cross-tier management across three storage tiers and automated intra-tier management within each storage tier in each extent pool.

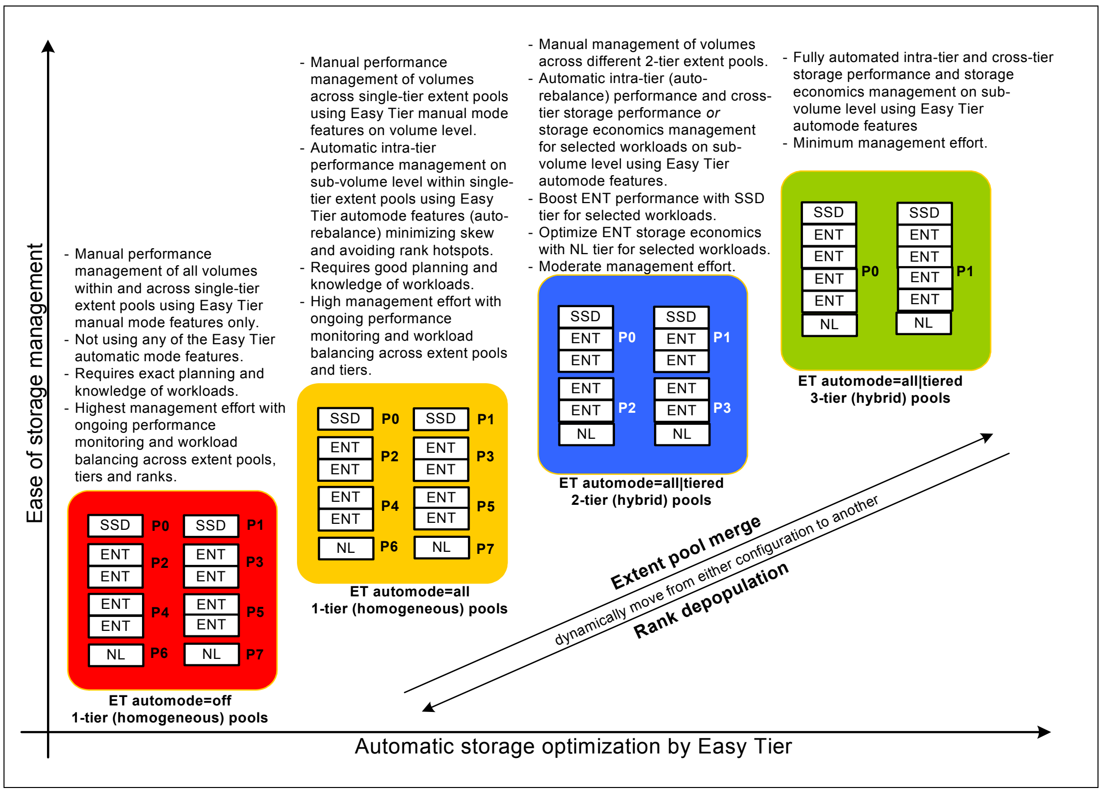


Figure 4-7 Ease of storage management versus automatic storage optimization by Easy Tier

All of these configurations can be combined with *I/O Priority Manager* to prioritize workloads when sharing the same resources and provide QoS levels in case resource contention occurs.

With the Easy Tier feature enabled, you can also take full advantage of the Easy Tier manual mode features, such as dynamic volume relocation (volume migration), dynamic extent pool merge, and rank depopulation to dynamically modify your logical configuration. When merging extent pools with different storage tiers, you can gradually introduce more automatic storage management with Easy Tier at any time. Or with rank depopulation, you can reduce multi-tier pools and automated cross-tier management according to your needs.

Our examples are generic. A single DS8000 system with its tremendous scalability can manage many applications effectively and efficiently, so typically multiple extent pool configurations exist on a large system for various needs for isolated and resource sharing workloads, Copy Services considerations, or other specific requirements. The logical configuration with the DS8000 and Easy Tier is flexible and less burdensome in a bad initial design or when gradually switching from one approach to another, for example, when introducing SSD or NL drives in a future expansion of your storage system. The use of Easy Tier automatic mode management with its automated data relocation and optimization within and across tiers is highly encouraged to provide optimum performance at a minimum administrative effort.

Easy Tier and I/O Priority Manager eventually simplify management with single-tier and multi-tier extent pools and help to spread workloads easily across shared hardware resources under optimum conditions and best performance, automatically adapting to changing workload conditions. You can choose from various extent pool configurations for your resource isolation and resource sharing workloads, combined with Easy Tier and I/O Priority Manager.

4.8.2 Single-tier extent pools

Single-tier or homogeneous extent pools are pools that only contain ranks from one storage class or storage tier:

- SSD (solid-state drive)
- Enterprise disk
- Nearline disk

Single-tier extent pools consist of one or more ranks that can be referred to as single-rank or multi-rank extent pools.

Single-rank extent pools

Single-rank extent pool configurations are used with initial DS8000 models to provide maximum control of volume placement and manual performance management on the rank level. In standard cases, the manual allocation of ranks and the use of single-rank extent pools are obsolete today. The same result, if required, can also be achieved with multi-rank extent pools using the *rotate volumes* EAM at a reduced configuration effort and with fewer limitations for taking advantage of the advanced DS8000 virtualization features.

One reason for single-rank extent pools in the past was the simple one-to-one mapping between ranks and extent pools. Because a volume is always created from a single extent pool, with single-rank extent pools, you can precisely control the volume placement across selected ranks and thus manually manage the I/O performance of the different workloads at the rank level. Furthermore, you can quickly obtain the relationship of a volume to its extent pool using the output of the DSCLI **lsfbvol** or **lsckdvol** command. Thus, with single-rank extent pools, there is a direct relationship between volumes and ranks based on the extent pool of the volume. This relationship helps you manage and analyze performance down to rank level more easily, especially with host-based tools, such as Resource Measurement Facility™ (RMF™) on System z in combination with a hardware-related assignment of LSS/LCU IDs. However, the administrative effort increases considerably, because you must create the volumes for a specific workload in multiple steps from each extent pool separately when distributing the workload across multiple ranks.

With single-rank extent pools, you choose a configuration design that limits the capabilities of a created volume to the capabilities of a single rank for capacity and performance. A single volume cannot exceed the capacity or the I/O performance provided by a single rank. So, for demanding workloads, you need to create multiple volumes from enough ranks from different extent pools and use host-level-based striping techniques, such as volume manager striping, to spread the workload evenly across the ranks dedicated to a specific workload. You are also likely to waste storage capacity easily if extents remain left on ranks in different extent pools, because a single volume can only be created from extents within a single extent pool, not across extent pools.

Furthermore, you benefit less from the advanced DS8000 virtualization features, such as dynamic volume expansion (DVE), FlashCopy SE, storage pool striping, Easy Tier automatic performance management, and workload spreading, which use the capabilities of multiple ranks within a single extent pool.

Single-rank extent pools are selected for environments where isolation or management of volumes on the rank level is desirable, such as in z/OS environments. Single-rank extent pools are selected for configurations using storage appliances, such as the SAN Volume Controller, where the selected RAID arrays are provided to the appliance as simple back-end storage capacity and where the advanced virtualization features on the DS8000 are not required or not wanted in order to avoid multiple layers of data striping. However, the use of homogeneous multi-rank extent pools and storage pool striping to minimize the storage administrative effort by shifting the performance management from the rank to the extent pool level and letting the DS8000 maintain a balanced data distribution across the ranks within a specific pool is popular. It provides excellent performance in relation to the reduced management effort.

Also, you do not need to strictly use *only* single-rank extent pools or *only* multi-rank extent pools on a storage system. You can base your decision on individual considerations for each workload group that is assigned to a set of ranks and thus extent pools. The decision to use single-rank and multi-rank extent pools depends on the logical configuration concept that is chosen for the distribution of the identified workloads or workload groups for isolation and resource-sharing.

In general, single-rank extent pools might not be good in the current complex and mixed environments unless you know that this level of isolation and micro-performance management is required for your specific environment. If not managed correctly, workload skew and rank hot spots that limit overall system performance are likely to occur.

Multi-rank homogeneous extent pools (with only one storage tier)

If the logical configuration concept aims to balance certain workloads or workload groups (especially large resource-sharing workload groups) across multiple ranks with the allocation of volumes or extents on successive ranks, use multi-rank extent pools for these workloads.

With a homogeneous multi-rank extent pool, you take advantage of the advanced DS8000 virtualization features to spread the workload evenly across the ranks in an extent pool to achieve a well-balanced data distribution with considerably less management effort. Performance management is shifted from the rank level to the extent pool level. An extent pool represents a set of merged ranks (a larger set of disk spindles) with a uniform workload distribution. So, the level of complexity for standard performance and configuration management is reduced from managing many individual ranks (*micro*-performance management) to a few multi-rank extent pools (*macro*-performance management).

The DS8000 EAMs, such as *rotate volumes* (**`-eam rotatevols`**) and *rotate extents* (**`-eam rotateexts`**), take care of spreading the volumes and thus the individual workloads evenly across the ranks within homogeneous multi-rank extent pools. *Rotate extents* is the default and preferred EAM to distribute the extents of each volume successively across all ranks in a pool to achieve a well-balanced capacity-based distribution of the workload. *Rotate volumes* is an exception today, but it can help to implement a strict volume to rank relationship. It reduces the configuration effort compared to single-rank extent pools by easily distributing a set of volumes to different ranks in a specific extent pool for workloads where the use of host-based striping methods is still preferred. The size of the volumes must fit to the available capacity on each rank. The number of volumes created for this workload in a specific extent pool must match the number of ranks (or be at least a multiple of this number). Otherwise, the result is an imbalanced volume and workload distribution across the ranks and likely rank bottlenecks. However, efficient host-based striping must be guaranteed in this case to spread the workload evenly across all ranks, eventually from two or more extent pools. See 3.2.7, "Extent allocation methods (EAMs)" on page 68 for more details about the EAMs and how the volume data is spread across the ranks in an extent pool.

Even multi-rank extent pools provide some level of control of volume placement across the ranks in cases where it is necessary to manually enforce a special volume allocation scheme. You can use the DSCLI command **chrank -reserve** to reserve all of the extents from a rank in an extent pool from being used for the next creation of volumes. Alternatively, you can use the DSCLI command **chrank -release** to release a rank and make the extents available again.

Multi-rank extent pools that use storage pool striping are the general configuration approach today on modern DS8800 and DS8700 systems to spread the data evenly across the ranks in a homogeneous multi-rank extent pool and thus reduce skew and the likelihood of single rank hot spots. Without Easy Tier automatic mode management, such non-managed, homogeneous multi-tier extent pools consist only of ranks of the same drive type and RAID level. Although not required (and probably not realizable for smaller or heterogeneous configurations), you can take the effective rank capacity into account, grouping ranks with and without spares into different extent pools when using storage pool striping to ensure a strict balanced workload distribution across all ranks up to the last extent. Otherwise, take additional considerations for the volumes created from the last extents in a mixed homogeneous extent pool that contains ranks with and without spares, because these volumes are probably allocated only on part of the ranks with the larger capacity and without spares.

In combination with Easy Tier, a more efficient and automated way of spreading the workloads evenly across all ranks in homogeneous multi-rank extent pool is available. The automated intra-tier performance management (*auto-rebalance*) of Easy Tier on DS8800 and DS8700 storage systems efficiently spreads the workload evenly across all ranks. It automatically relocates the data across the ranks of the same storage class in an extent pool based on rank utilization to achieve and maintain a balanced distribution of the workload, minimizing skew and avoiding rank hot spots. You can enable *auto-rebalance* for homogeneous extent pool by setting the Easy Tier management scope to *all* extent pools (ETautomode=all).

In addition, Easy Tier automatic mode can also handle storage device variations within a tier that uses a *micro-tiering* capability. An example of storage device variations within a tier is an intermix of ranks with different disk rpm or RAID levels within the same storage class of an extent pool. A typical micro-tiering scenario is, for example, when, after a hardware upgrade, new 15K rpm Enterprise disk drives intermix with existing 10K rpm Enterprise disk drives. In these configurations, the Easy Tier automatic mode micro-tiering capability takes into account the different performance profiles of each micro-tier and performs intra-tier (auto-rebalance) optimizations. Easy Tier does not handle a micro-tier like an additional tier; it is still part of a specific tier. For this reason, the hotness of an extent does not trigger any promotion or demotion across micro-tiers of the same tier, because the extent relocation across micro-tiers can only occur as part of the auto-rebalance feature and is based on rank utilization.

Important: With the DS8000 LMC R6.2, Easy Tier does not differentiate 10K and 15K Enterprise disk drives as separate storage tiers in a managed extent pool. Both disk drives are considered as the *same* storage tier and no automated cross-tier promotion or demotion algorithms are applied between these two storage classes. Easy Tier automated data relocation across tiers to optimize performance and storage economics based on the hotness of the particular extent only takes place between *different* storage tiers. If these drives are mixed in the same managed extent pool, the Easy Tier *auto-rebalance* algorithm only balances the workload across all ranks of this Enterprise-class tier based on overall rank utilization, taking the performance capabilities of each rank (micro-tiering capability) into account.

Figure 4-8 provides two configuration examples for using dedicated homogeneous extent pools with storage classes in combination with and without Easy Tier automatic mode management.

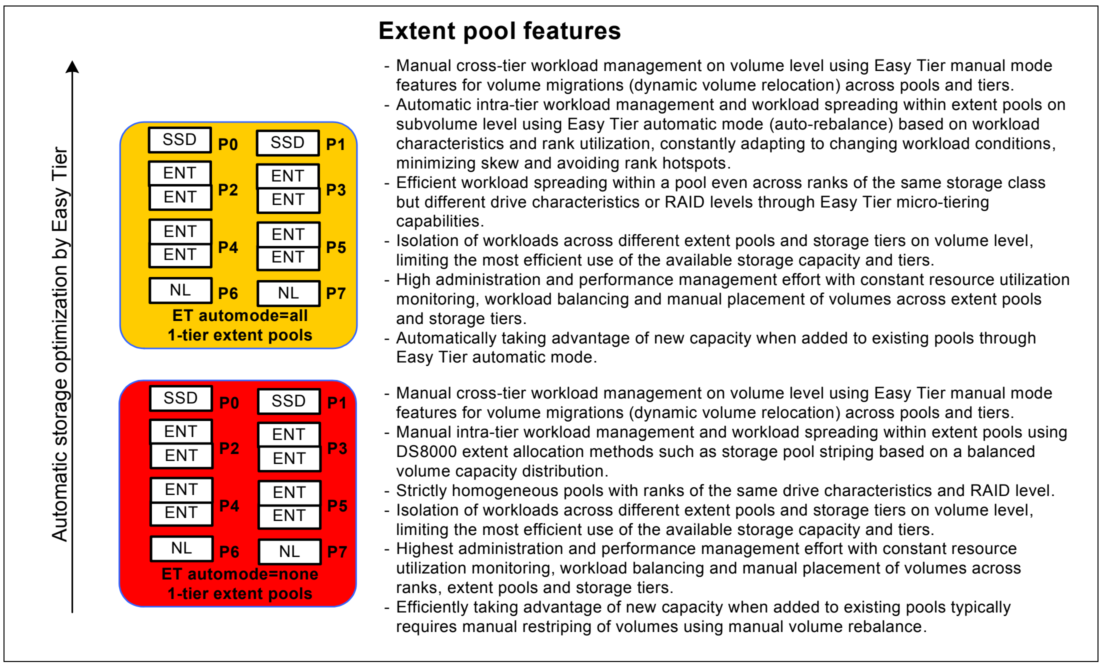


Figure 4-8 Single-tier extent pool configuration examples and Easy Tier benefits

With *multi-rank extent pools*, you can fully use the features of the DS8000 virtualization architecture and Easy Tier that provide ease of use when you manage more applications effectively and efficiently with a single DS8000 system. Consider multi-rank extent pools and the use of Easy Tier automatic management especially for mixed workloads that are to be spread across multiple ranks. Multi-rank extent pools help to simplify management and volume creation. They also allow the creation of single volumes that can span multiple ranks and thus exceed the capacity and performance limits of a single rank.

However, for performance management and analysis reasons in balanced multi-rank extent pools, you sometimes might want to relate *volumes* associated with a specific I/O workload to ranks that provide the physical disk spindles for servicing the workload I/O requests and determine the I/O processing capabilities. The DSCLI **`showfbvol -rank`** or **`showckdvol -rank`** command can be used to show this relationship by displaying the extent distribution of a volume across the ranks, as shown in Example 3-2 on page 85.

Easy Tier manual mode features, such as dynamic extent pool merge, dynamic volume relocation (volume migration), and rank depopulation also help to easily manage complex configurations with different extent pools. You can migrate volumes from one highly utilized extent pool to another less utilized one, or from an extent pool with a lower storage class to another one associated with a higher storage class, and merge smaller extent pools to larger ones. You can also redistribute the data of a volume within a pool using the manual volume rebalance feature, for example, after new capacity is added to a pool or two pools are merged, to manually optimize the data distribution and workload spreading within a pool. However, manual extent pool optimization and performance management, such as *manual volume rebalance,* is not required (and not supported) if the pools are managed by Easy Tier automatic mode. Easy Tier automatically places the data in these pools even if the pools are merged or new capacity is added to a pool.

For more information about data placement in extent pool configurations, see 3.3.2, “Extent pool considerations” on page 78.

> **Important:** Multi-rank extent pools offer numerous advantages with respect to ease of use, space efficiency, and the DS8000 virtualization features. Multi-rank extent pools, in combination with Easy Tier automatic mode, provide both ease of use and excellent performance for standard environments with workload groups that share a set of homogeneous resources.

4.8.3 Multi-tier extent pools

Multi-rank or hybrid extent pools consist of ranks from different storage classes or storage tiers (referred to as *multi-tier* or *hybrid extent pools*). Data placement within and across these tiers can automatically be managed by Easy Tier providing automated storage performance and storage economics optimization.

A multi-tier extent pool can consist of one of the following storage class combinations with up to three storage tiers:

- SSD + Enterprise disk
- SSD + nearline disk
- Enterprise disk + nearline disk
- SSD + Enterprise disk + nearline disk

Multi-tier extent pools are especially suited for mixed, resource sharing workloads. Tiered storage, as described in 4.1, “Review the tiered storage concepts and Easy Tier” on page 88, is an approach of utilizing types of storage throughout the storage infrastructure. It is a mix of higher performing/higher cost storage with lower performing/lower cost storage and placing data based on its specific I/O access characteristics. While SSDs can help to efficiently boost enterprise-class performance, you can additionally shrink the footprint and reduce costs by adding large capacity nearline drives while maintaining enterprise class performance. Correctly balancing all the tiers eventually leads to the lowest cost and best performance solution.

Always create hybrid extent pools for Easy Tier automatic mode management. The extent allocation for volumes in hybrid extent pools differs from the extent allocation in homogeneous pools. Any specified EAM, such as *rotate extents* or *rotate volumes,* is ignored when a new volume is created in or migrated into a hybrid pool. The EAM is changed to *managed*, as soon as the Easy Tier automatic mode is enabled for the pool, and the volume is under the control of Easy Tier. Easy Tier then automatically moves extents to the most appropriate storage tier and rank in the pool based on performance aspects.

In hybrid extent pools (even if currently not managed by Easy Tier), an initial EAM, similar to rotate extents, is used for new volumes. The initial volume creation always starts on the ranks of the Enterprise tier first in a rotate extents-like fashion. It continues on the ranks of the Nearline tier and eventually on the ranks of the SSD tier if insufficient capacity is available on the previous storage tier in that sequence. See “Extent allocation in hybrid and managed extent pools” on page 69 for more details.

Workload spreading across the resources (ranks and DAs) in a *managed* hybrid pool is automatic by Easy Tier by using intra-tier (auto-rebalance) and cross-tier data relocation to optimize storage performance and storage economics based on data performance characteristics. Easy Tier automatic mode adapts to changing workload conditions and automatically promotes hot extents from the lower tier to the upper tier (Enterprise to SSD and nearline to Enterprise). Or, it demotes colder extents from the higher tier to the lower tier (swap extents from SSD with hotter extents from Enterprise tier, or demote cold extents from Enterprise tier to Nearline tier). Easy Tier automatic mode optimizes the Nearline tier by demoting some of the sequential workload to the Nearline tier to better balance sequential workloads. Auto-rebalance rebalances extents across the ranks of the same tier based on rank utilization to minimize skew and avoid hot spots. Auto-rebalance takes different device characteristics into account when different devices or RAID levels are mixed within the same storage tier (micro-tiering).

Regarding the requirements of your workloads, you can create a pair or multiple pairs of extent pools with different 2-tier or 3-tier combinations that depend on your needs and available hardware resources. You can, for example, create separate 2-tier SSD/ENT and ENT/NL extent pools to isolate your production environment from your development environment. You can boost the performance of your production application with SSDs and optimize storage economics for your development applications with NL drives.

Or, you can create 3-tier extent pools for mixed, large resource-sharing workload groups and benefit from fully automated storage performance and economics management at a minimum management effort. You can boost the performance of your high-demand workloads with SSDs and reduce the footprint and costs with NL drives for the lower demand data.

With Easy Tier, managing the data location of your volumes on the extent level across all ranks in a managed hybrid extent pool, you can use the DSCLI **showfbvol -rank** or **showckdvol -rank** command to display the current extent distribution of a volume across the ranks, as shown in Example 3-2 on page 85. Additionally, the *volume heat distribution* (volume heat map), as provided by the STAT, can help to identify the amount of hot, warm, and cold extents for each volume and its current distribution across the storage tiers in the pool. For more information about the STAT, see 6.7, “Storage Tier Advisor Tool” on page 231.

The ratio of SSD, ENT, and NL disk capacity in a hybrid pool depends on the workload characteristics and skew and must be planned when ordering the drive hardware for the identified workloads. See “Multi-tier extent pools” on page 81 and “Drive Selection with Easy Tier” in *IBM System Storage DS8800 and DS8700 Performance with Easy Tier 3rd Generation*, WP102024, for additional information.

With the Easy Tier manual mode features, such as dynamic extent pool merge, dynamic volume relocation, and rank depopulation, you can modify existing configurations easily, depending on your needs. You can grow from a manually managed single-tier configuration into a partially or fully automatically managed tiered storage configuration. You add tiers or merge appropriate extent pools and enable Easy Tier at any time. For more information about Easy Tier, see *IBM System Storage DS8000 Easy Tier,* REDP-4667.

Important: Multi-tier extent pools and Easy Tier help you implement a tiered storage architecture on a single DS8000 system with all its benefits at a minimum management effort and ease of use. Easy Tier and its automatic data placement within and across tiers spread the workload efficiently across the available resources in an extent pool. Easy Tier constantly optimizes storage performance and storage economics and adapts to changing workload conditions. Easy Tier can reduce overall performance management efforts and help to consolidate more workloads efficiently and effectively on a single DS8000 system. It optimizes performance and reduces energy costs and the footprint.

Figure 4-9 provides two configuration examples of using hybrid extent pools with Easy Tier. One example shares all resources for all workloads with automatic management across three tiers. The other example isolates workload groups with different optimization goals to 2-tier extent pool combinations.

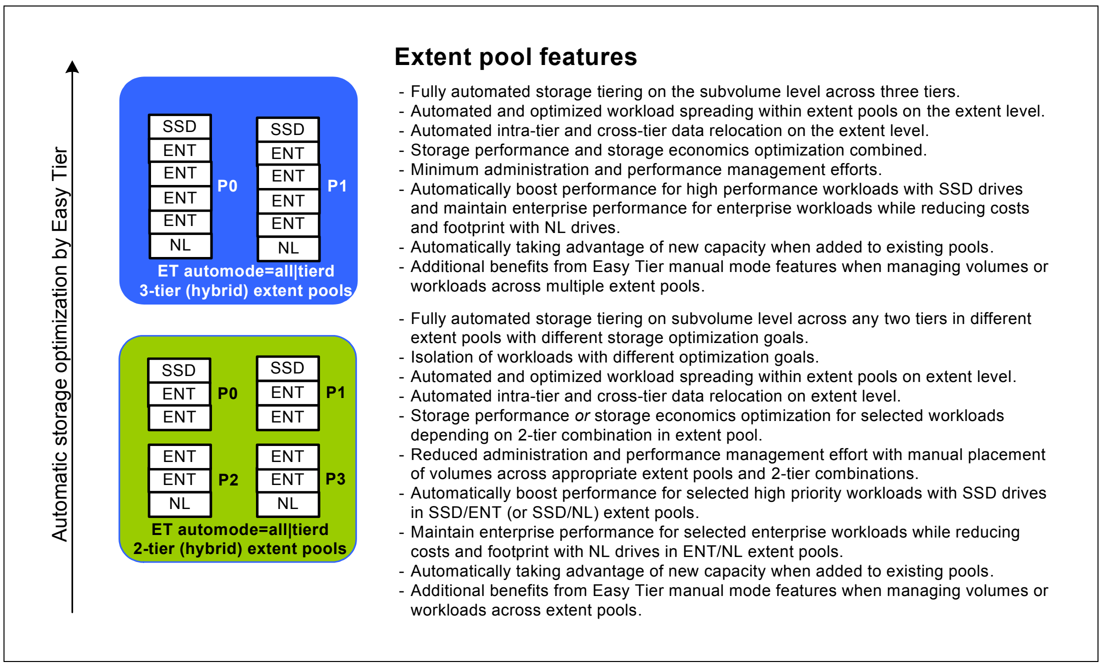


Figure 4-9 Multi-tier extent pool configuration examples and Easy Tier benefits

4.8.4 Additional implementation considerations for multi-tier extent pools

For multi-tier extent pools managed by Easy Tier automatic mode, consider how to allocate the initial volume capacity and how to gradually introduce new workloads to managed pools.

Using thin provisioning with Easy Tier in multi-tier extent pools

In environments that use FlashCopy Copy Services only, which already support the usage of thin-provisioned volumes (ESE) with the DS8000 R6.2 LMC, you might start using thin-provisioned volumes in managed hybrid extent pools, especially with three tiers. You use this configuration to start the initial volume allocation for as many volumes as possible on the Enterprise tier (home tier). Avoid the initial creation of fully provisioned volumes on the nearline drive tier when the storage capacity on the Enterprise tier is already exhausted. This exhaustion is due to the creation of fully provisioned volumes with unused allocated capacities.

In this case, only used capacity is allocated in the pool and Easy Tier does not move unused extents around or move hot extents on a large scale up from the Nearline tier to the Enterprise tier and to the SSD tier. However, thin-provisioned volumes are not fully supported with all DS8000 Copy Services or advanced functions yet, so it might not be a valid approach for all environments at this time. For more information about the initial volume allocation in hybrid extent pools, see “Extent allocation in hybrid and managed extent pools” on page 69.

Staged Implementation approach for multi-tier extent pools

In a new three-tier environment, volumes are first created on the Enterprise tier, and Easy Tier cannot “learn” and optimize before production starts. If you are migrating all the servers at once, some server volumes, for reasons of space, might be placed completely on nearline drives first, although these servers also might have higher performance requirements. For more information about the initial volume allocation in hybrid extent pools, see “Extent allocation in hybrid and managed extent pools” on page 69.

When bringing a new DS8800 system into production to replace an older one, with the older storage system often not using Easy Tier, consider the timeline of the implementation stages, by which you migrate all servers from the older to the new storage system.

Consider a staged approach when migrating servers to a new multi-tier DS8800 system:

- Assign the resources for the high-performing and response time sensitive workloads first, then add the less performing workloads. The other way might lead to situations where all initial resources, such as the Enterprise tier in hybrid extent pools, are allocated already by the secondary workloads. This situation does not leave enough space on the Enterprise tier for the primary workloads, which then must be initially on the Nearline tier.
- Split your servers into several subgroups, where you migrate each subgroup one by one, and not all at once. Then, allow the Easy Tier several days to learn and optimize. Some extents are moved to SSDs and some extents are moved to nearline. You regain space on the Enterprise HDDs. After a server subgroup learns and reaches a steady state, the next server subgroup can be migrated. You gradually allocate the capacity in the hybrid extent pool by optimizing the extent distribution of each application one by one while regaining space in the Enterprise tier (home tier) for the next applications.

4.8.5 Extent allocation in homogeneous multi-rank extent pools

When creating a volume on a DS8800/8700 with microcode R6.1(or later) and using the **mkfbvol** or **mkckdvol** DSCLI command, you can specify the EAM manually. Use either the option **-eam rotatevols** (*rotate volumes*) or **-eam rotateexts** (*rotate extents,* the default) for each volume created in a homogeneous extent pool not managed by Easy Tier. In managed extent pools, Easy Tier places the data across the ranks and the EAM automatically is changed to *managed*. It stays *managed* once Easy Tier is disabled again to reflect that the previous allocation method might not be maintained and that the allocation is under control of Easy Tier. The initial extent allocation differs in hybrid extent pools, as described in “Extent allocation in hybrid and managed extent pools” on page 69.

The EAM determines *how* a volume is created within a multi-rank extent pool for the allocation of the extents on the available ranks. The selection of the appropriate EAM is key when manually spreading the workload evenly across all ranks in non-managed homogeneous extent pools without Easy Tier automatic mode. The EAM can be selected at the volume level and is not an attribute of an extent pool, except if the pool is managed by Easy Tier and the EAM is automatically changed to managed. You can have volumes that are created with the *rotate extents* and *rotate volume* methods in the same extent pool. To lose the benefits of a rotate extents algorithm with a uniform workload distribution across the ranks, do not carelessly use both extent allocation algorithms together within the same extent pool. For example, consider using *rotate volumes* as the EAM for specific workloads where host-level striping or application-based striping is preferred. Use the *rotate extents* algorithm for other workloads in the same multi-rank extent pool where host-level striping is not an option. However, ensure that you do not compromise the benefit of storage pool striping using rotate extents in an extent pool by using rotate volumes for other workloads in the same pool with only a poor implementation of host-level or application-based striping.

The *rotate extents* algorithm spreads the extents of a single volume and the I/O activity of each volume across all the ranks in an extent pool. Furthermore, it ensures that the allocation of the first extent of successive volumes starts on different ranks rotating through all available ranks in the extent pool to optimize workload spreading across all ranks.

With the default rotate extents (**`rotateexts`**) algorithm, the extents (1 GiB for FB volumes and 1113 cylinders or approximately 0.94 GiB for CKD volumes) of each single volume are spread across all ranks within an extent pool (provided the size of the volume in extents is at least equal to or larger than the number of ranks in the extent pool) and thus across more disks. This approach reduces the occurrences of I/O hot spots at the rank level within the storage system. Storage pool striping helps to balance the overall workload evenly across the back-end resources. It reduces the risk of single ranks that become performance bottlenecks while providing ease of use with less administrative effort.

When using the optional *rotate volumes* (**`rotatevols`**) EAM, each volume, one volume after another volume, is placed on a single rank with a successive distribution across all ranks in a round-robin fashion.

Rotate volumes: For environments where dedicated volume-to-rank allocations are preferred or required, you can use either single-rank or multiple-rank extent pools if the workloads need to be spread across multiple ranks with successive volume IDs. However, the use of multi-rank extent pools using the *rotate volumes* EAM together with a carefully planned volume layout is encouraged. It can achieve the same volume distribution across ranks with less administrative effort than can be achieved with single-rank extent pools.

The *rotate extents* and *rotate volumes* EAMs determine the initial data distribution of a volume and thus the spreading of workloads in non-managed, single-tier extent pools. With *Easy Tier* automatic mode enabled for single-tier (homogeneous) or multi-tier (hybrid) extent pools, this selection becomes unimportant. The data placement and thus the workload spreading is managed by Easy Tier. The use of Easy Tier automatic mode for single-tier extent pools is highly encouraged for an optimal spreading of the workloads across the resources. In single-tier extent pools, you can benefit from the Easy Tier automatic mode feature *auto-rebalance*. Auto-rebalance constantly and automatically balances the workload across ranks of the same storage tier based on rank utilization, minimizing skew and avoiding the occurrence of single rank hot spots. It can also take device variations (different drive types) or different RAID levels within a storage tier into account (Easy Tier *micro-tiering* capability).

Additional considerations for specific applications and environments

This section provides considerations when manually spreading the workload in single-tier extent pools and selecting the correct EAM without using Easy Tier. The use of Easy Tier automatic mode also for single-tier extent pools is highly encouraged in all environments where it is appropriate. Shared environments with mixed workloads that benefit from storage pool striping and multi-tier configurations benefit from Easy Tier automatic management. However, Easy Tier automatic management might not be an option in highly optimized environments where a fixed volume-to-rank relationship and well-planned volume configuration is required.

Certain application environments might benefit from the use of storage pool striping (rotate extents):

- Operating systems that do not directly support host-level striping
- VMware datastores
- Microsoft Exchange 2010 or Exchange 2003 database
- Windows clustering environments
- Older Solaris environments
- Environments that need to suballocate storage from a large pool
- Applications with multiple volumes and volume access patterns that differ from day to day
- Resource sharing workload groups that are dedicated to many ranks with host operating systems that do not all use or support host-level striping techniques or application-level striping techniques

However, there might also be valid reasons for *not* using storage pool striping. You might use it to avoid unnecessary layers of striping and reorganizing I/O requests, which might increase latency and not help achieve a more evenly balanced workload distribution. Multiple independent striping layers might be counterproductive. For example, creating a number of volumes from a single multi-rank extent pool that uses storage pool striping and then, additionally, use host-level striping or application-based striping on the same set of volumes might compromise performance. In this case, two layers of striping are combined with no overall performance benefit. In contrast, creating four volumes from four different extent pools from both rank groups that use storage pool striping and then use host-based striping or application-based striping on these four volumes to aggregate the performance of the ranks in all four extent pools and both processor complexes is reasonable.

Consider the following products for specific environments:

- SAN Volume Controller: SAN Volume Controller is a storage appliance with its own methods of striping and its own implementation of IBM Easy Tier. So, you can select to use striping or Easy Tier on the SAN Volume Controller or the DS8000. For more information, see Chapter 15, "IBM System Storage SAN Volume Controller attachment" on page 533.

- ProtecTIER: ProtecTIER is a backup solution and storage appliance. It resembles SAN Volume Controller, so that certain SAN Volume Controller guidelines might apply. For more information, see Chapter 16, "IBM ProtecTIER deduplication" on page 553.

- System i: System i has also its own methods of spreading workloads and datasets across volumes. We describe System i storage configuration suggestions and Easy Tier benefits in detail in Chapter 13, "Performance considerations for IBM i" on page 475.

- System z: z/OS typically implements data striping with storage management subsystem (SMS) facilities. To make the SMS striping effective, storage administrators must plan to ensure a correct data distribution across the physical resources. For this reason, single-rank extent pool configuration was preferred in the past. Using storage pool striping and multi-rank extent pools can considerably reduce storage configuration and management efforts, because a uniform striping within an extent pool can be provided by the DS8000 system. Multi-rank extent pools, storage pool striping, and Easy Tier are reasonable options today for System z configurations. For more information, see Chapter 14, "Performance considerations for System z servers" on page 499.

- Database volumes: If these volumes are used by databases or applications that explicitly manage the workload distribution by themselves, these applications might achieve maximum performance by using their native techniques for spreading their workload across independent LUNs from different ranks. Especially with IBM DB2 or Oracle, where the vendor suggests specific volume configurations, for example, DB2 balanced configuration units (BCUs) or Oracle Automatic Storage Management (ASM), it is preferable to follow those suggestions. For more information about this topic, see Chapter 17, “Database performance” on page 559.

- Applications that evolved particular storage strategies over a long time, with proven benefits, and where it is unclear whether they can additionally benefit from using storage pool striping. When in doubt, follow the vendor recommendations.

The DS8000 storage pool striping is based on spreading extents across different ranks. So, with extents of 1 GiB (FB) or 0.94 GiB (1113 cylinders/CKD), the size of a data chunk is rather large. For distributing random I/O requests, which are evenly spread across the capacity of each volume, this chunk size generally is appropriate. However, depending on the individual access pattern of a specific application and the distribution of the I/O activity across the volume capacity, certain applications perform better. Use more granular stripe sizes for optimizing the distribution of the application I/O requests across different RAID arrays by using host-level striping techniques or have the application manage the workload distribution across independent volumes from different ranks.

Consider the following points for selected applications or environments to use storage pool striping in homogeneous configurations:

- DB2: Excellent opportunity to simplify storage management using storage pool striping. You might prefer to use DB2 traditional recommendations for DB2 striping for performance-sensitive environments.
- DB2 and similar data warehouse applications, where the database manages storage and parallel access to data. Consider generally independent volumes on individual ranks with a careful volume layout strategy that does *not* use storage pool striping. Containers or database partitions are configured according to suggestions from the database vendor.
- Oracle: Excellent opportunity to simplify storage management for Oracle. You might prefer to use Oracle traditional suggestions that involve Automatic Storage Management (ASM) and Oracle striping capabilities for performance-sensitive environments.
- Small, highly active logs or files: Small highly active files or storage areas smaller than 1 GiB with a high access density might require spreading across multiple ranks for performance reasons. However, storage pool striping only offers a striping granularity on extent levels around 1 GiB, which is too large in this case. Continue to use host-level striping techniques or application-level striping techniques that support smaller stripe sizes. For example, assume a 0.8 GiB log file exists with extreme write content, and you want to spread this log file across several RAID arrays. Assume that you intend to spread its activity across four ranks. At least four 1 GiB extents must be allocated, one extent on each rank (which is the smallest possible allocation). Creating four separate volumes, each with a 1 GiB extent from each rank, and then using Logical Volume Manager (LVM) striping with a relatively small stripe size (for example, 16 MiB) effectively distributes the workload across all four ranks. Creating a single LUN of four extents, which is also distributed across the four ranks using the DS8000 storage pool striping, cannot effectively spread the file workload evenly across all four ranks due to the large stripe size of one extent, which is larger than the actual size of the file.
- Tivoli® Storage Manager storage pools: Tivoli Storage Manager storage pools work well in striped pools. But Tivoli Storage Manager suggests that the Tivoli Storage Manager databases need to be allocated in a separate pool or pools.
- AIX volume groups (VGs): LVM and physical partition (PP) striping continue to be powerful tools for managing performance. In combination with storage pool striping, now considerably fewer stripes are required for common environments. Instead of striping across a large set of volumes from many ranks (for example, 32 volumes from 32 ranks), striping is only required across a few volumes from a small set of different multi-rank extent pools from both DS8000 rank groups that use storage pool striping. For example, use four volumes from four extent pools, each with eight ranks. For specific workloads that

use the advanced AIX LVM striping capabilities with a smaller granularity on the KB or MB level, instead of storage pool striping with 1 GB extents (FB), might be preferable to achieve the highest performance.

- Microsoft Windows volumes: Typically, only a few large LUNs per host system are preferred, and host-level striping is not commonly used. So, storage pool striping is an ideal option for Windows environments. You can create single, large capacity volumes that offer the performance capabilities of multiple ranks. A volume is not limited by the performance limits of a single rank, and the DS8000 spreads the I/O load across multiple ranks.
- Microsoft Exchange: Storage pool striping makes it easier for the DS8000 to conform to Microsoft sizing suggestions for Microsoft Exchange databases and logs.
- Microsoft SQL Server: Storage pool striping makes it easier for the DS8000 to conform to Microsoft sizing suggestions for Microsoft SQL Server databases and logs.
- VMware Datastore for Virtual Machine Storage Technologies (VMware ESX Server Filesystem (VMFS) or virtual raw device mapping (RDM) access: Because datastores concatenate LUNs rather than striping them, allocate the LUNs inside a striped storage pool. Estimating the number of disks (or ranks) to support any specific I/O load is straightforward based on the specific requirements.

In general, storage pool striping helps to improve overall performance and reduce the effort of performance management by evenly distributing data and thus workloads across a larger set of ranks, reducing skew and hot spots. Certain application workloads can also benefit from the higher number of disk spindles behind one volume. But, there are cases where host-level striping or application-level striping might achieve a higher performance, at the cost of higher overall administrative effort. Storage pool striping might deliver good performance in these cases with less management effort, but manual striping with careful configuration planning is required to achieve the best possible performance. So for overall performance and ease of use, storage pool striping might offer an excellent compromise for many environments, especially for larger workload groups where host-level striping techniques or application-level striping techniques are not widely used or available.

Note: Business and performance critical applications always require careful configuration planning and individual decisions on a case by case basis. Determine whether to use storage pool striping or LUNs from dedicated ranks together with host-level striping techniques or application-level striping techniques for the best performance.

Storage pool striping is best suited for new extent pools. Adding ranks to an existing extent pool does not restripe volumes (LUNs) that are already allocated in an existing pool, unless you manually restripe LUNs with the **managefbvol** command. You can use Easy Tier manual mode features, such as *manual volume rebalance.* For more information, see *IBM System Storage DS8000 Easy Tier,* REDP-4667.

Always consider that Easy Tier offers more advanced options than storage pool striping to efficiently spread the workload across the resources in an extent pool. Storage pool striping only achieves a balanced capacity distribution of the volumes across the ranks. Easy Tier automatically relocates the data across the ranks based on the actual workload pattern to achieve a balanced resource utilization within a storage tier (or in a homogeneous extent pool). Or with multi-tier extent pool configurations, Easy Tier optimizes storage performance and economics across different storage tiers.

4.8.6 Balancing workload across available resources

To achieve a balanced utilization of all available resources of the DS8000 storage system, you need to distribute the I/O workloads evenly across the available back-end resources:

- Ranks (disk drive modules)
- DA pairs

And, you need to distribute the I/O workloads evenly across the available front-end resources:

- I/O ports
- HA cards
- I/O enclosures

You need to distribute the I/O workloads evenly across both DS8000 processor complexes (also called storage server `0/CEC#0` and storage server `1/CEC#1`), as well.

Configuring the extent pools determines the balance of the workloads across the available back-end resources, ranks, DA pairs, and both processor complexes.

Each extent pool is associated with an extent pool ID (P0, P1, and P2, for example). Each rank has a relationship to a specific DA pair and can be assigned to only one extent pool. You can have as many (non-empty) extent pools as you have ranks. Extent pools can be expanded by adding more ranks to the pool. However, when assigning a rank to a specific extent pool, the affinity of this rank to a specific DS8000 processor complex is determined. No predefined affinity of ranks to a processor complex by hardware exists. All ranks that are assigned to even-numbered extent pools (P0, P2, and P4, for example) form *rank group 0* and are serviced by DS8000 processor complex 0. All ranks that are assigned to odd-numbered extent pools (P1, P3, and P5, for example) form *rank group 1* and are serviced by DS8000 processor complex 1.

In order to spread the overall workload across both DS8000 processor complexes, a minimum of two extent pools is required: one assigned to processor complex 0 (for example, `P0`) and one assigned to processor complex 1 (for example, `P1`).

For a balanced distribution of the overall workload across both processor complexes and both DA cards of each DA pair, apply the following rules. For each type of rank and its RAID level, storage type (FB or CKD), and disk drive characteristics (disk type, rpm speed, and capacity), apply these rules:

- Assign half of the ranks to even-numbered extent pools (rank group 0) and assign half of them to odd-numbered extent pools (rank group 1).
- Spread ranks with and without spares evenly across both rank groups.
- Distribute ranks from each DA pair evenly across both rank groups.

It is important to understand that you might seriously limit the available back-end bandwidth and thus the system overall throughput if, for example, all ranks of a DA pair are assigned to only one rank group and thus a single processor complex. In this case, only one DA card of the DA pair is used to service all the ranks of this DA pair and thus only half of the available DA pair bandwidth is available.

This practice now also becomes important with SSDs and small block random I/O workloads. To be able to use the full back-end random I/O performance of two (or more) SSD ranks within a specific DA pair, the SSD I/O workload also must be balanced across both DAs of the DA pair. This balance can be achieved by assigning half of the SSD ranks of each DA pair to even extent pools (P0, P2, and P4, for example, managed by storage server 0/rank group 0) and half of the SSD ranks to odd extent pools (P1, P3, and P5, for example, managed by storage server 1/rank group 1). If, for example, all SSD ranks of the same DA pair are assigned to the same rank group (for example, rank group 0 with extent pool P0 and P2), only one DA card of the DA pair is used to access this set of SSD ranks and only half of the available I/O processing capability of the DA pair is available, which severely limits the overall SSD performance.

4.8.7 Extent pool configuration examples

The next sections provide examples to demonstrate how to balance the arrays across extent pools and processor complexes to provide optimum overall DS8000 system performance. The configuration on your DS8000 system typically differs considerably from these examples, depending on your hardware configuration number of ranks, DA pairs, storage classes, storage types (FB or CKD), RAID levels, and spares.

DS8700 extent pool configuration examples

This example represents a homogeneously configured DS8700 with four DA pairs and 32 ranks of the same drive type all configured to RAID 5. Based on the specific DS8700 hardware and rank configuration, the scheme typically becomes more complex for the number of DA pairs, ranks, different RAID levels, disk drive classes, spare distribution, and storage types. Each DA pair is typically populated with up to eight ranks. Expect additional ranks on DA pairs DA0 and DA2 on a fully populated system with all expansion frames.

A simple two-extent pool example, evenly distributing all ranks from each DA pair across both processor complexes for a homogeneously configured DS8700 system with one workload group sharing all resources and using storage pool striping, can be seen in Figure 4-10. The volumes created from the last extents in the pool are only distributed across the large 7+P ranks, because the capacity on the 6+P+S ranks is exhausted at this time. Ensure that you use this remaining capacity only for workloads with lower performance requirements in manually managed environments. Or, consider using Easy Tier automatic mode management (auto-rebalance) instead.

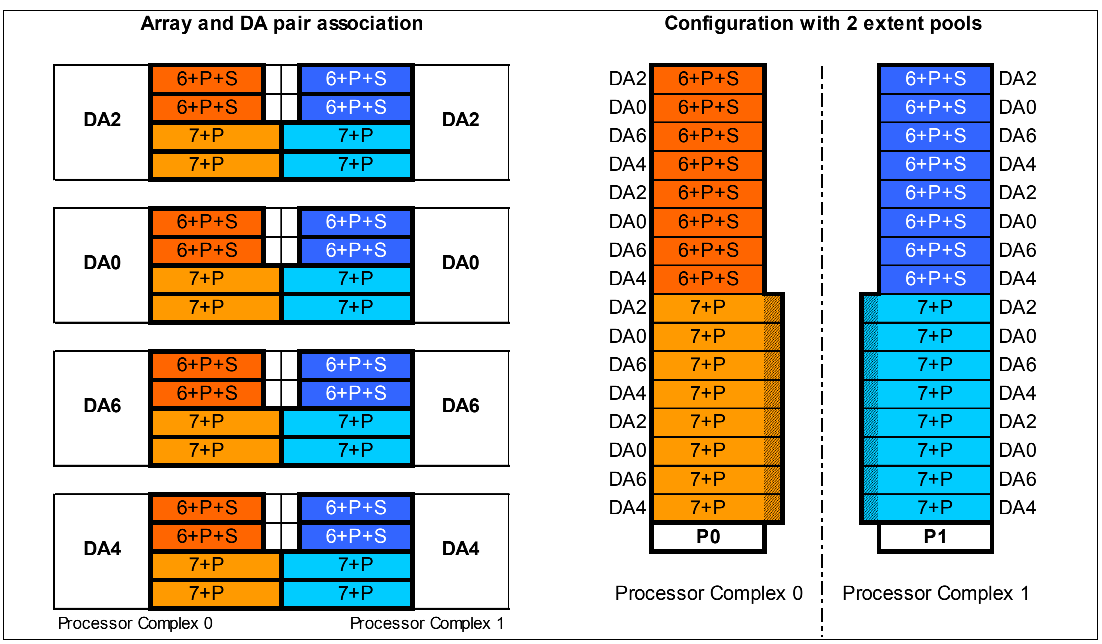


Figure 4-10 Example of homogeneously configured DS8700 with two extent pools

Another example for a homogeneously configured DS8700 system with four extent pools and one workload group sharing all resources across four extent pools or two isolated workload groups each sharing half of the resources can be seen in Figure 4-11.

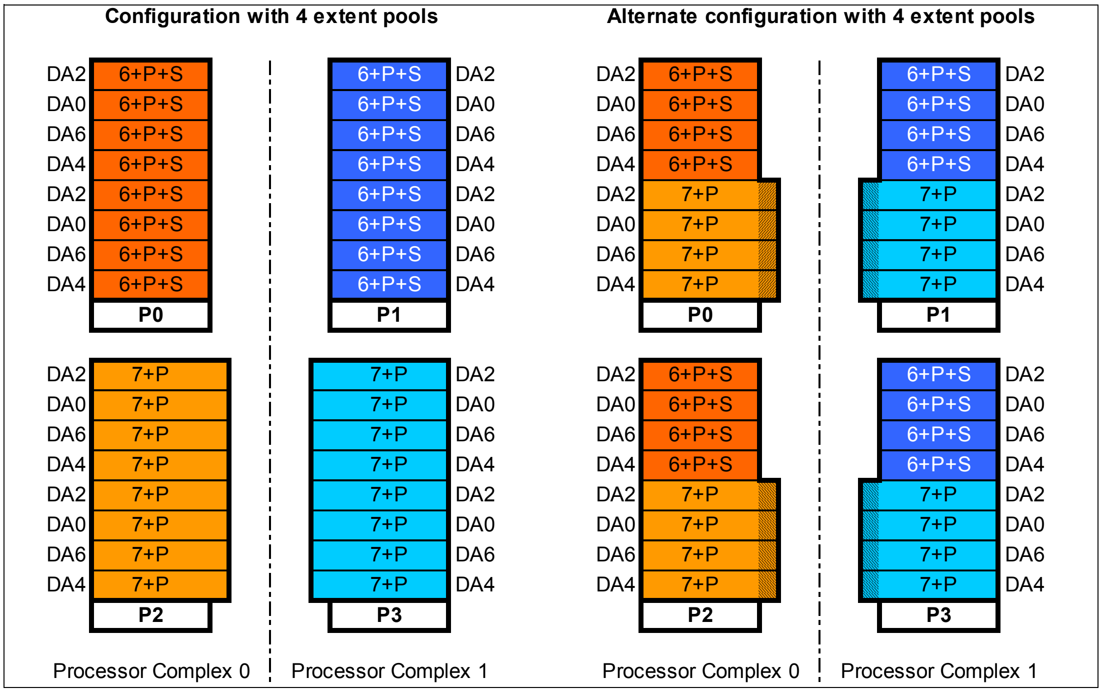


Figure 4-11 Example of homogeneously configured DS8700 with four extent pools

On the left in Figure 4-11, we have four strictly homogeneous extent pools, each one contains only ranks of the same RAID level and capacity for the spare drives. Storage pool striping efficiently distributes the extents in each extent pool across all ranks up to the last volume created in these pools, providing more ease of use. However, the extent pools with 6+P+S ranks and 7+P ranks differ in overall capacity, which might only be appropriate for two different workload groups.

Another configuration with four extent pools is shown on the right in Figure 4-11. We evenly distribute the 6+P+S and 7+P ranks from all DA pairs across all four extent pools to obtain the same overall capacity in each extent pool. However, the last capacity in these pools is only allocated on the 7+P ranks. So, ensure that you use this remaining capacity only for workloads with lower performance requirements in manually managed environments. Or, consider using Easy Tier automatic mode management (auto-rebalance) instead. Using four extent pools and storage pool striping instead of two can also reduce the failure boundary from one extent pool with 16 ranks (that is, if one rank fails all data in the pool is lost) to two distinct extent pools with only eight ranks per processor complex (for example, when physically separating tablespaces from logs).

You can also consider separating workloads by using different extent pools and the principles of workload isolation, as shown in Figure 4-12 on page 133. The isolated workload can either use storage pool striping as the EAM or *rotate volumes* combined with host-level or application-level striping. The workload isolation in this example is on the DA pair level (DA2). In addition, we have two pairs of extent pools for resource sharing workload groups. Furthermore, we can also consider using Easy Tier automatic management for all pools instead of manually spreading the workload.

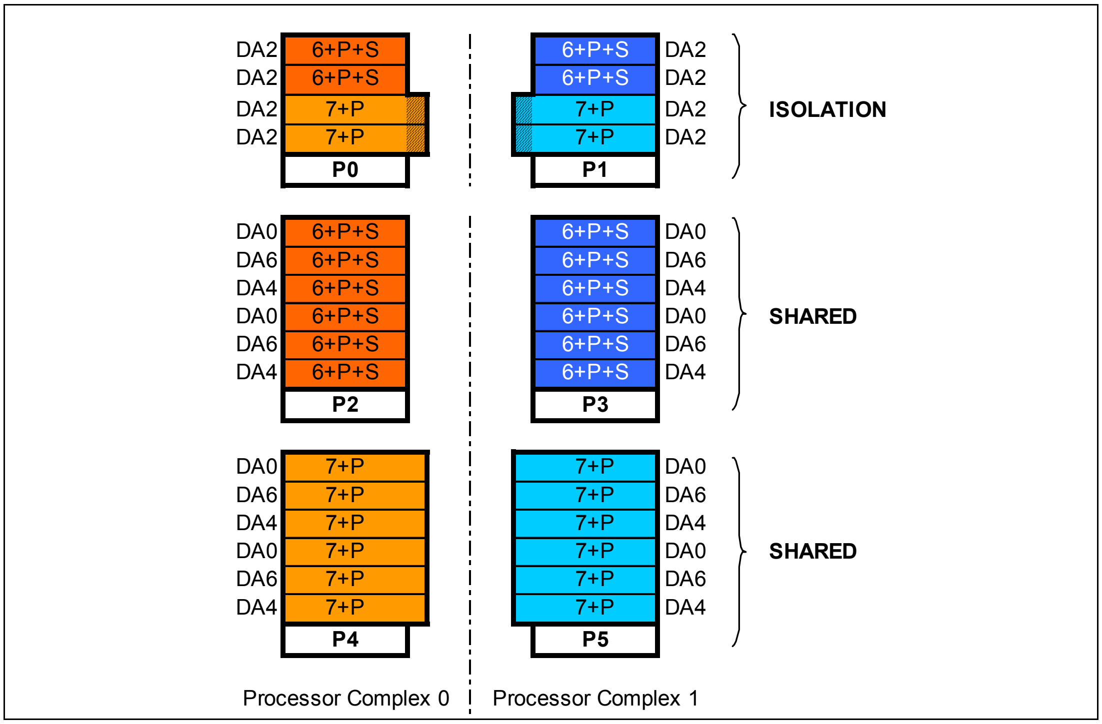


Figure 4-12 Example of DS8700 extent pool configuration with workload isolation on the DA pair level

Another consideration for the number of extent pools to create is the usage of Copy Services, such as FlashCopy SE. If you use FlashCopy SE, you also might consider a minimum of four extent pools with two extent pools per rank group or processor complex, as shown in Figure 4-13 on page 134. The FlashCopy SE repository for the space-efficient target volumes is distributed across all available ranks within the extent pool (comparable to using storage pool striping). Therefore, we suggest that you distribute the source and target volumes across different extent pools (that is, different ranks) from the same DS8000 processor complex (that is, the same rank group) for the best FlashCopy performance. Each extent pool can have FlashCopy source volumes, as well as repository space for space-efficient FlashCopy target volumes from source volumes, in the alternate extent pool. However, for certain environments, consider a dedicated set of extent pools that uses RAID 10 arrays for FlashCopy SE target volumes while the other extent pools that use RAID 5 arrays are only used for source volumes.

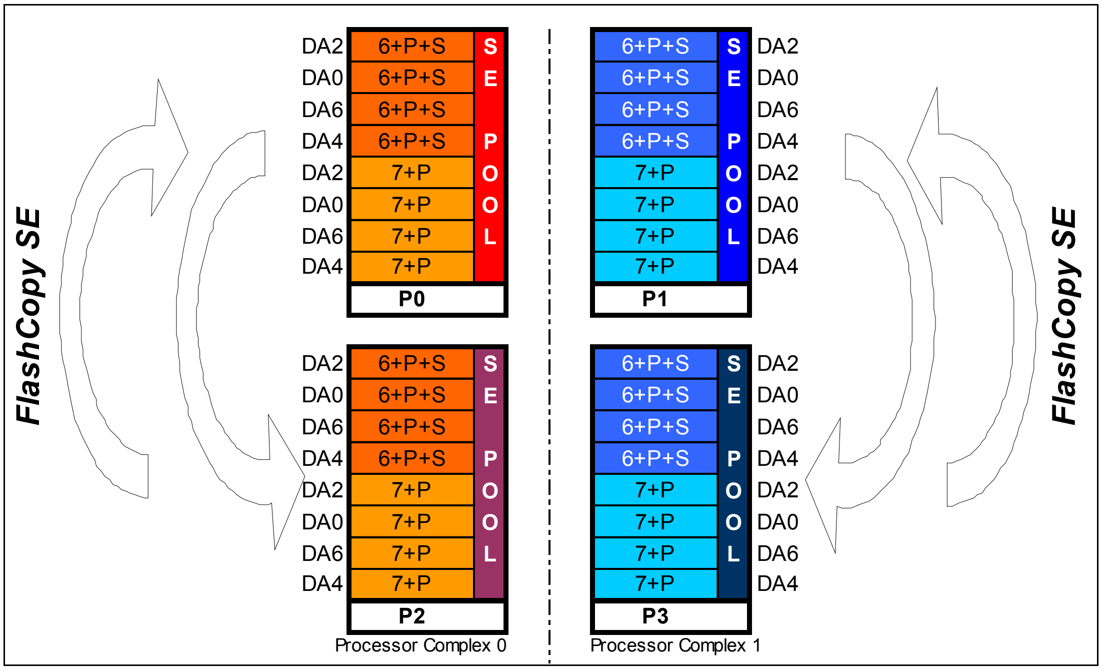


Figure 4-13 Example of a four-extent pool configuration on DS8700 using FlashCopy SE

Figure 4-14 gives an example of a balanced extent pool configuration with two extent pools on a DS8700 system with three different storage classes using Easy Tier. All arrays from each DA pair (especially the SSD arrays) are evenly distributed across both processor complexes to fully use the back-end DA pair bandwidth and I/O processing capabilities. All cross-tier and intra-tier data relocation in these pools is automatic by Easy Tier, constantly optimizing storage performance and storage economics.

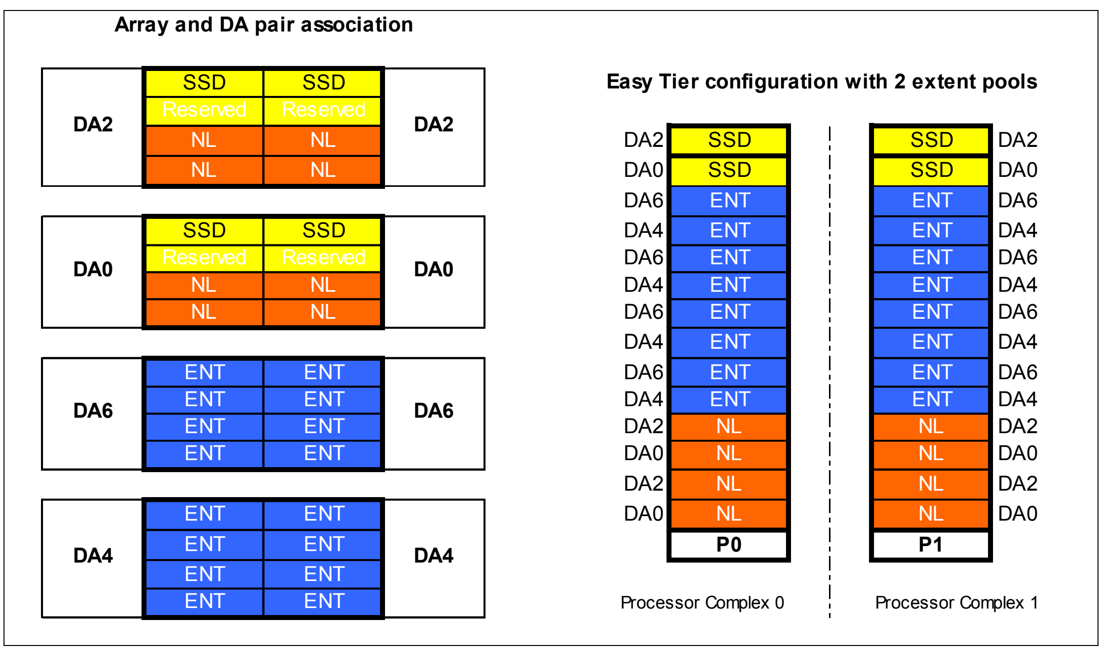


Figure 4-14 Example DS8700 configuration with two hybrid extent pools using Easy Tier

DS8800 extent pool configuration examples

This example represents a homogeneously configured DS8800 with four DA pairs and 24 ranks of the same SFF drive type that are all configured as RAID 5. Based on the specific DS8800 hardware and rank configuration, the scheme typically becomes more complex for the number of DA pairs, ranks, SFF or LFF drive enclosures, different RAID levels or drive classes, spare distribution, and storage types. Each DA pair is typically populated with up to six ranks before the next DA pair is used. Expect additional ranks on all DA pairs on a fully populated system with expansion frames.

A simple two-extent pool example, evenly distributing all ranks from each DA pair across both processor complexes for a homogeneously configured DS8800 system with one workload group sharing all resources and using storage pool striping, can be seen in Figure 4-15. The volumes created from the last extents in the pool are only distributed across the large 7+P ranks, because the capacity on the 6+P+S ranks is exhausted. Be sure to use this remaining capacity only for workloads with lower performance requirements in manually managed environments or consider using Easy Tier automatic mode management (auto-rebalance) instead.

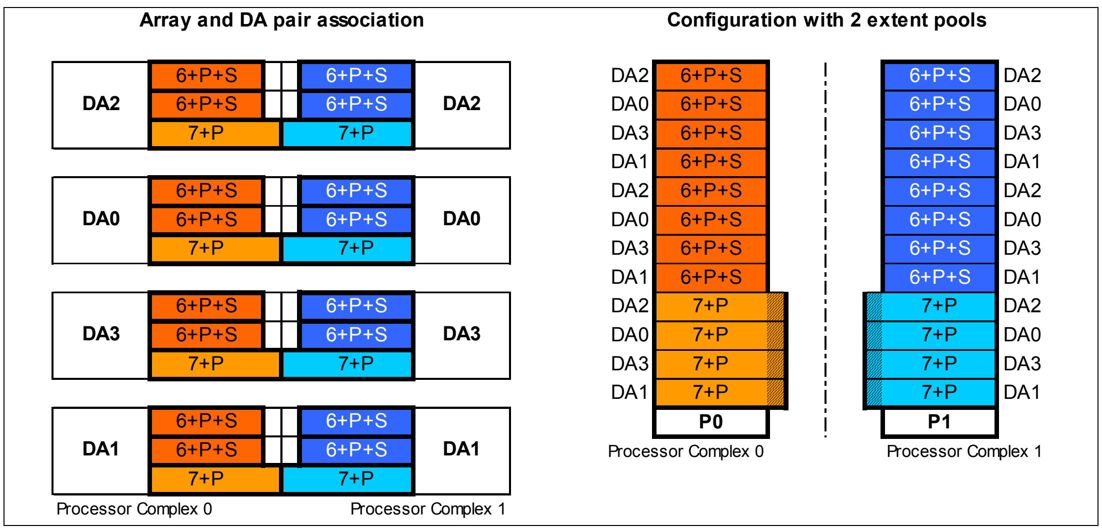


Figure 4-15 Example of homogeneously configured DS8800 with two extent pools

Another example for a homogeneously configured DS8800 system with four extent pools and one workload group, which shares all resources across four extent pools, or two isolated workload groups that each share half of the resources can be seen in Figure 4-16 on page 136.

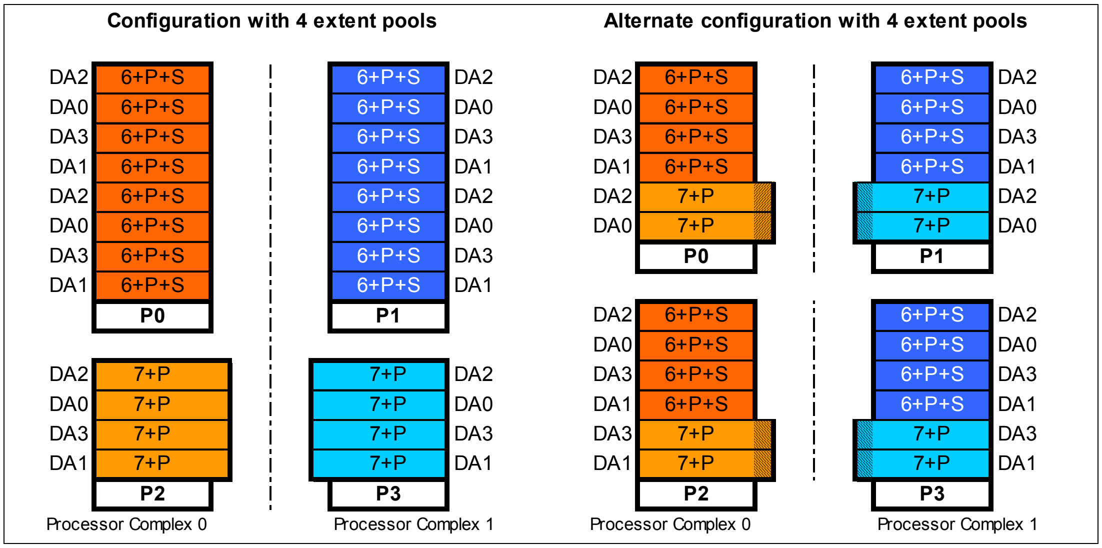


Figure 4-16 Example of homogeneously configured DS8800 with four extent pools

On the left in Figure 4-16, we have four strictly homogeneous extent pools, and each one contains only ranks of the same RAID level and capacity for the spare drives. Storage pool striping efficiently distributes the extents in each extent pool across all ranks up to the last volume created in these pools, providing more ease of use. However, the extent pools with 6+P+S ranks and 7+P ranks differ considerably in overall capacity and performance, which might only be appropriate for two workload groups with different overall capacity and performance requirements.

Another configuration with four extent pools is shown on the right in Figure 4-16. We evenly distribute the 6+P+S and 7+P ranks from all DA pairs across all four extent pools to obtain the same overall capacity in each extent pool. However, the last capacity in these pools is only allocated on the 7+P ranks. So, be sure to use this remaining capacity only for workloads with lower performance requirements in manually managed environments. Or, consider using Easy Tier automatic mode management (auto-rebalance) instead. Using four extent pools and storage pool striping instead of two can also reduce the failure boundary from one extent pool with 12 ranks (that is, if one rank fails, all data in the pool is lost) to two distinct extent pools with only six ranks per processor complex (for example, when physically separating tablespaces from logs).

Also consider separating workloads by using different extent pools with the principles of workload isolation, as shown in Figure 4-17 on page 137. The isolated workload can either use storage pool striping as the EAM or rotate volumes combined with host-level or application-level striping if desired. The workload isolation in this example is on the DA pair level (DA2). In addition, we have one pair of extent pools for resource sharing workload groups. Instead of manually spreading the workloads across the ranks in each pool, consider using Easy Tier automatic mode management (auto-rebalance) for all pools.

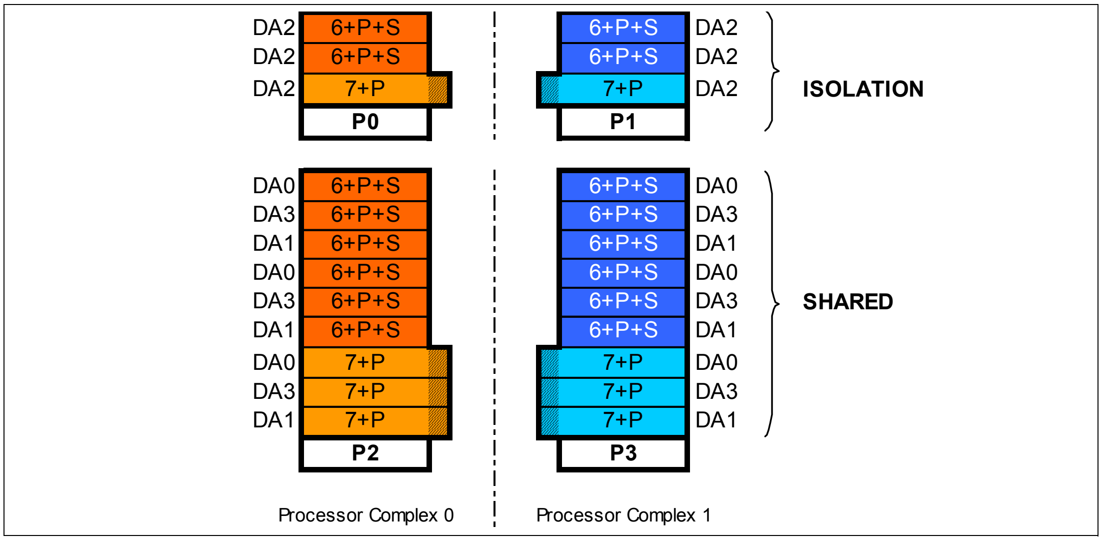


Figure 4-17 Example of DS8800 extent pool configuration with workload isolation on DA pair level

Another consideration for the number of extent pools to create is the usage of Copy Services, such as FlashCopy SE. If you use FlashCopy SE, you also might consider a minimum of four extent pools with two extent pools per rank group or processor complex, as shown in Figure 4-13 on page 134 for a DS8700 system. Because the FlashCopy SE repository for the space-efficient target volumes is distributed across all available ranks within the extent pool (comparable to using storage pool striping), we suggest that you distribute the source and target volumes across different extent pools (that is, different ranks) from the same DS8000 processor complex (that is, the same rank group) for the best FlashCopy performance. Each extent pool can have FlashCopy source volumes, as well as repository space for space-efficient FlashCopy target volumes from source volumes, in the alternate extent pool. However, for certain environments, consider dedicated extent pools that use RAID 10 arrays for FlashCopy SE target volumes while the other extent pools that use RAID 5 arrays are only used for source volumes.

Figure 4-18 on page 138 gives an example of a balanced extent pool configuration with two extent pools on a DS8800 system with three storage classes that use Easy Tier. All arrays from each DA pair (especially the SSD arrays) are evenly distributed across both processor complexes to fully use the back-end DA pair bandwidth and I/O processing capabilities. There is a difference in DA pair population with ranks from LFF disk enclosures for the 3.5 inch NL drives. A pair of LFF disk enclosure only contains 24 disk drives (three ranks) compared to a pair of SFF disk enclosures with 48 disk drives (six ranks). All cross-tier and intra-tier data relocation in these pools is automatic by Easy Tier. Easy Tier constantly optimizes storage performance and storage economics.

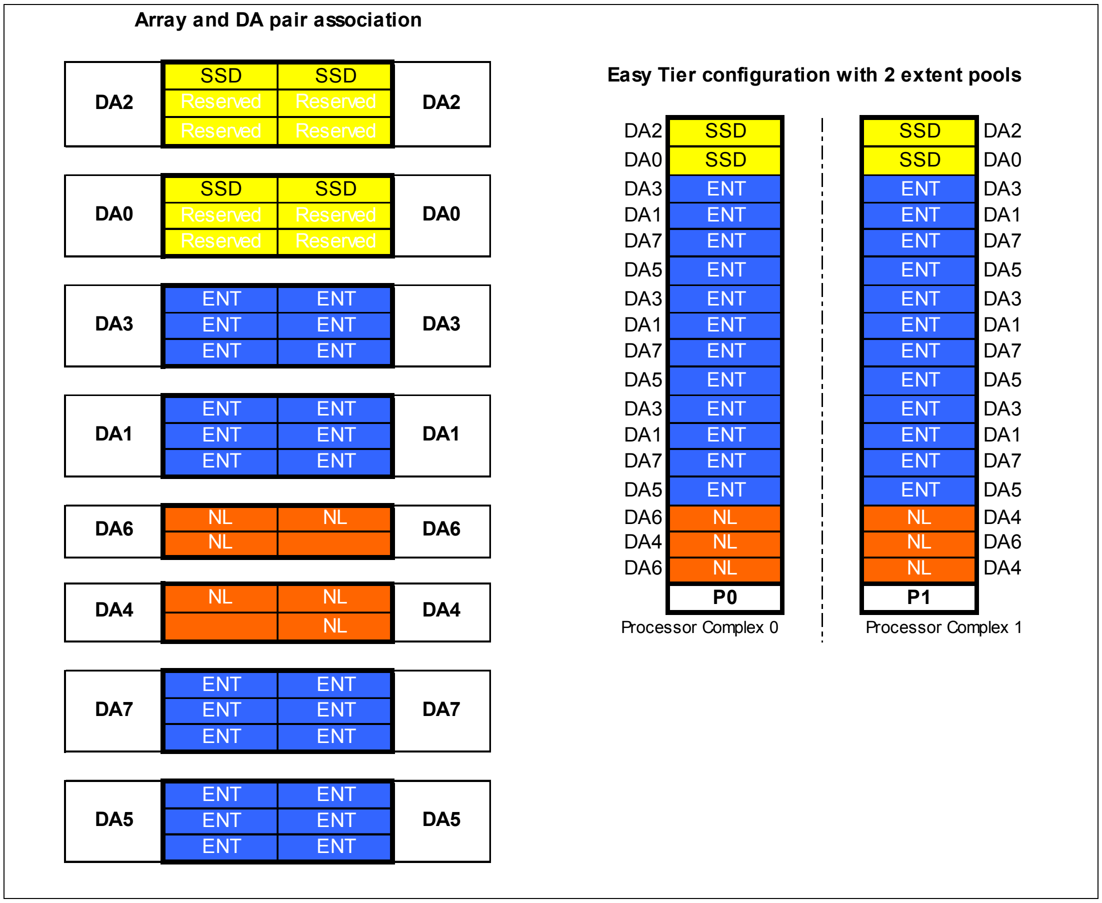


Figure 4-18 Example DS8800 configuration with two hybrid extent pools that use Easy Tier

4.8.8 Assigning workloads to extent pools

Extent pools can only contain ranks of the same storage type, either FB for Open Systems or System i or CKD for System z. You can have multiple extent pools in various configurations on a single DS8000 system, for example, managed or non-managed single-tier (homogeneous) extent pools with ranks of only one storage class. You can have multi-tier (hybrid) extent pools with any two storage tiers (SSD/ENT, SSD/NL, or ENT/NL) or up to three storage tiers (SSD/ENT/NL) managed by Easy Tier.

Multiple homogeneous extent pools, each with different storage classes, easily allow tiered storage concepts with dedicated extent pools and manual cross-tier management. For example, you can have extent pools with slow, large-capacity drives for backup purposes and other extent pools with high-speed, small capacity drives or solid-state drives (SSDs) for performance-critical transaction applications. Or you can use hybrid pools with Easy Tier and introduce fully automated cross-tier storage performance and economics management.

Using dedicated extent pools with an appropriate number of ranks and DA pairs for selected workloads is a suitable approach for *isolating* workloads.

The *minimum* number of required extent pools depends on the following considerations:

- The number of isolated and resource-sharing workload groups
- The number of different storage types, either FB for Open Systems or System i or CKD for System z

- Definition of failure boundaries (for example, separating logs and tablespaces to different extent pools)
- Copy Services considerations (for example, using a minimum of two pools per rank group for best FlashCopy SE performance)

Although you are not restricted from assigning all ranks to only one extent pool, the *minimum* number of extent pools, even with only one workload on a homogeneously configured DS8000, needs to be *two* (for example, P0 and P1). You need one extent pool for each rank group (or storage server), so that the overall workload is balanced across both processor complexes.

To optimize performance, the ranks for each workload group (either isolated or resource-sharing workload groups) need to be split across at least *two* extent pools with an equal number of ranks from each rank group. So, also at the workload level, each workload is balanced across both processor complexes. Typically, you need to assign an equal number of ranks from each DA pair to extent pools assigned to processor complex 0 (rank group 0: P0, P2, and P4, for example) and to extent pools assigned to processor complex 1 (rank group 1: P1, P3, and P5, for example). In environments with FB and CKD storage (Open Systems and System z), you additionally need separate extent pools for CKD and FB volumes. We advise a minimum of *four* extent pools to balance the capacity and I/O workload between the two DS8000 processor complexes. Additional extent pools might be desirable to meet individual needs, such as ease of use, implementing tiered storage concepts, or separating ranks for different DDM types, RAID types, clients, applications, performance, or Copy Services requirements.

The *maximum* number of extent pools, however, is given by the number of available ranks (that is, creating one extent pool for each rank).

Creating dedicated extent pools on the DS8000 with dedicated back-end resources for separate workloads allows individual performance management for business and performance-critical applications. Compared to easier to manage *share and spread everything* storage systems without the possibility to implement workload isolation concepts, creating dedicated extent pools on the DS8000 with dedicated back-end resources for separate workloads is an outstanding feature of the DS8000 as an enterprise-class storage system. With it, you can consolidate and manage various application demands with different performance profiles, which are typical in enterprise environments, on a single storage system. Unfortunately, the cost is a slightly higher administrative effort.

Before configuring the extent pools, we advise that you collect all the hardware-related information of each rank for the associated DA pair, disk type, available storage capacity, RAID level, and storage type (CKD or FB) in a spreadsheet. Then, plan the distribution of the workloads across the ranks and their assignments to extent pools.

As the first step, you can visualize the rank and DA pair association in a simple spreadsheet that is based on the graphical scheme that is shown in Figure 4-19 on page 140.

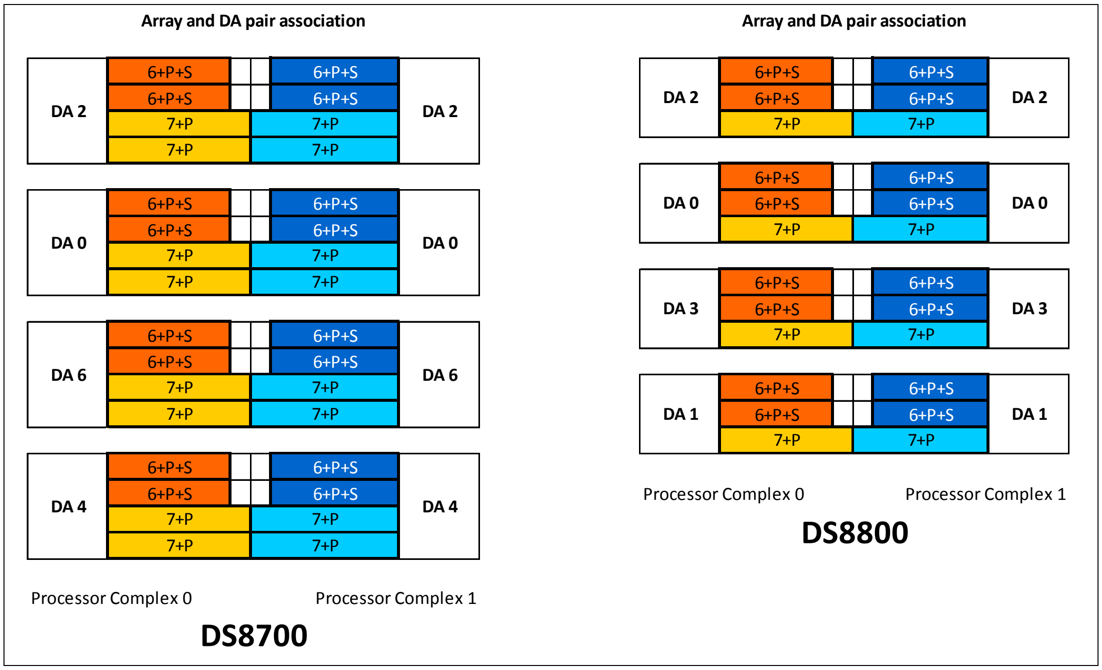


Figure 4-19 Basic scheme for rank and DA pair association in relation to extent pool planning

This example represents a homogeneously configured DS8700 with four DA pairs and 32 ranks that are all configured to RAID 5 and a DS8800 with four DA pairs and 24 ranks. Based on the specific DS8000 hardware and rank configuration, the scheme typically becomes more complex for the number of DA pairs, ranks, different RAID levels, drive classes, spare distribution, and storage types. The DS8700 with the LFF disk enclosures differs from the DS8800 with the SFF disk enclosures. Also, be aware of the difference in DA pair population on DS8800 with ranks from LFF disk enclosures for the 3.5 inch NL drives. A pair of LFF disk enclosures only contains 24 disk drives (three ranks) compared to a pair of SFF disk enclosures with 48 disk drives (six ranks).

On a homogeneous DS8700, the initial population of a DA pair typically increases to up to eight ranks before the next DA pair is populated. On a homogeneous DS8800 with only high-density enclosures (not considering the new LFF disk enclosures for the nearline drives), the initial population of a DA pair typically increases to up to six ranks before the next DA pair is populated. Expect additional ranks on the DA pairs on fully populated systems.

Based on this scheme, you can plan an initial assignment of ranks to your planned workload groups, either isolated or resource-sharing, and extent pools for your capacity requirements.

After this initial assignment of ranks to extent pools and appropriate workload groups, you can create additional spreadsheets to hold more details about the logical configuration and finally the volume layout of the array site IDs, array IDs, rank IDs, DA pair association, extent pools IDs, and volume IDs, as well as their assignments to volume groups and host connections.

4.9 Planning address groups, LSSs, volume IDs, and CKD PAVs

After creating the extent pools and evenly distributing the back-end resources (DA pairs and ranks) across both DS8000 processor complexes, you can create host volumes from these extent pools. When creating the host volumes, it is important to follow a volume layout scheme that evenly spreads the volumes of each application workload across all ranks and extent pools that are dedicated to this workload, to achieve a balanced I/O workload distribution across ranks, DA pairs, and the DS8000 processor complexes.

So, the next step is to plan the volume layout and thus the mapping of address groups and LSSs to volumes created from the various extent pools for the identified workloads and workload groups. For performance management and analysis reasons, it is crucial to be able to easily relate *volumes*, which are related to a specific I/O workload, to rank*s*, which finally provide the physical disk spindles for servicing the workload I/O requests and determining the I/O processing capabilities. Therefore, an overall logical configuration concept that easily relates volumes to workloads, extent pools, and ranks is desirable.

Each volume is associated with a hexadecimal 4-digit volume ID that must be specified when creating the volume, as shown, for example, in Table 4-2 for volume ID 1101.

Table 4-2 Understanding the volume ID relationship to address groups and LSSs/LCUs

Volume ID	Digits	Description
1101	First digit: **1***xxx*	Address group (0 - F) (16 address groups on a DS8000 subsystem)
	First and second digits: **11***xx*	Logical subsystem (LSS) ID for FB Logical control unit (LCU) ID for CKD (x0 - xF: 16 LSSs or LCUs per address group)
	Third and fourth digits: *xx***01**	Volume number within an LSS or LCU (00 - FF: 256 volumes per LSS or LCU)

The first digit of the hexadecimal volume ID specifies the address group, 0 - F, of that volume. Each address group can only be used by a single storage type, either FB or CKD. The first and second digit together specify the logical subsystem ID (LSS ID) for Open Systems volumes (FB) or the logical control unit ID (LCU ID) for System z volumes (CKD). There are 16 LSS/LCU IDs per address group. The third and fourth digits specify the volume number within the LSS/LCU, 00 - FF. There are 256 volumes per LSS/LCU. The volume with volume ID 1101 is the volume with volume number 01 of LSS 11, and it belongs to address group 1 (first digit).

The LSS/LCU ID is furthermore related to a rank group. LSS/LCU IDs are restricted to volumes that are created from rank group 0 and serviced by processor complex 0. Odd LSS/LCU IDs are restricted to volumes that are created from rank group 1 and serviced by processor complex 1. So, the volume ID also reflects the affinity of that volume to a DS8000 processor complex. All volumes, which are created from even-numbered extent pools (P0, P2, and P4, for example) have even LSS IDs and are managed by DS8000 processor complex 0. All volumes that are created from odd-numbered extent pools (P1, P3, and P5, for example) have odd LSS IDs and are managed by DS8000 processor complex 1.

There is no direct DS8000 performance implication as a result of the number of defined LSSs/LCUs. For the z/OS CKD environment, a DS8000 volume ID is also required for each PAV. The maximum of 256 addresses per LCU includes both CKD base volumes and PAVs, so the number of volumes and PAVs determines the number of required LCUs.

In the past, for performance analysis reasons, it was useful to easily identify the association of specific volumes to ranks or extent pools when investigating resource contention. But with the introduction of storage pool striping (rotate extents allocation method), the use of multi-rank extent pools is the preferred configuration approach today for most environments. And with the availability of Easy Tier automatic mode management for single-tier extent pools (auto-rebalance), which is based on the actual workload and rank utilization, we advise that you use Easy Tier automatic mode management also for single-tier extent pools. This method is easier than manually spreading the workload across the ranks by using EAMs, such as storage pool striping. Multi-tier extent pools are managed by Easy Tier automatic mode anyway, constantly providing automatic storage intra-tier and cross-tier performance and storage economics optimization. In managed pools, Easy Tier automatically relocates the data to the appropriate ranks and storage tiers based on the access pattern, so the extent allocation across the ranks for a specific volume is likely to change over time. With storage pool striping or extent pools managed by Easy Tier, we no longer have a fixed relationship between the performance of a specific volume and a single rank. Therefore, planning for a hardware-based LSS/LCU scheme and relating LSS/LCU IDs to hardware resources, such as ranks, is no longer reasonable. Performance management focus is shifted from ranks to extent pools. However, a numbering scheme that only relates to the extent pool might still be viable, but it is less common and less practical.

The common approach that is still valid today with Easy Tier and storage pool striping is to relate an LSS/LCU to a specific application workload with a meaningful numbering scheme for the volume IDs for the distribution across the extent pools. Each LSS can have 256 volumes, with volume numbers ranging from 00 - FF. So, relating the LSS/LCU to a certain application workload and additionally reserving a specific range of volume numbers for different extent pools is a reasonable choice especially in Open Systems environments. Because volume IDs are transparent to the attached host systems, this approach helps the administrator of the host system to easily determine the relationship of volumes to extent pools by the volume ID. Therefore, this approach helps you to easily identify physically independent volumes from different extent pools when setting up host-level striping across pools. This approach helps you when separating, for example, DB tablespaces from DB logs onto volumes from physically different drives in different pools.

This approach not only provides a logical configuration concept that provides ease of use for storage management operations, but it also reduces management efforts when using the DS8000 related Copy Services, because basic Copy Services management steps (such as establishing Peer-to-Peer Remote Copy (PPRC) paths and consistency groups) are related to LSSs. If Copy Services are not currently planned, plan the volume layout, because overall management is easier if you need to introduce Copy Services in the future (for example, when migrating to a new DS8000 storage system that uses Copy Services).

However, the actual strategy for the assignment of LSS/LCU IDs to resources and workloads can still vary depending on the particular requirements in an environment.

The following section introduces suggestions for LSS/LCU and volume ID numbering schemes to help to relate volume IDs to application workloads and extent pools. For other approaches to the LSS/LCU ID planning, including hardware-bound LSS/LCU configuration schemes down to the rank level, see *DS8000 Performance Monitoring and Tuning,* SG24-7146.

4.9.1 Volume configuration scheme using application-related LSS/LCU IDs

We show you several suggestions for a volume ID numbering scheme in a multi-rank extent pool configuration where the volumes are evenly spread across a set of ranks. These examples refer to a workload group where multiple workloads share the set of resources. The following suggestions apply mainly to Open Systems. For CKD environments, the LSS/LCU layout is defined at the operating system level with the Input/Output Configuration Program (IOCP) facility. The LSS/LCU definitions on the storage server must match exactly the IOCP statements. For this reason, any consideration on the LSS/LCU layout must be made first during the IOCP planning phase and afterward mapped to the storage server.

Typically, when using LSS/LCU IDs that relate to application workloads, the simplest approach is to reserve a suitable number of LSS/LCU IDs according to the total amount of volumes requested by the application. Then, populate the LSS/LCUs in sequence, creating the volumes from offset 00. Ideally, all volumes that belong to a certain application workload or a group of related host systems are within the same LSS. However, because the volumes need to also be spread evenly across both DS8000 processor complexes, at least two logical subsystems are typically required per application workload. One even LSS is for the volumes managed by processor complex 0, and one odd LSS is for volumes managed by processor complex 1 (for example, LSS 10 and LSS 11). Moreover, consider the future capacity demand of the application when planning the number of LSSs to be reserved for an application. So, for those applications that are likely to increase the number of volumes beyond the range of one LSS pair (256 volumes per LSS), we advise that you reserve a suitable number of LSS pair IDs for them from the beginning.

Figure 4-20 shows an example of an application-based LSS numbering scheme. This example shows three applications, application A, B, and C, that share two large extent pools. Hosts A1 and A2 both belong to application A and are assigned to LSS 10 and LSS 11, each using a different volume ID range from the same LSS range. LSS 12 and LSS 13 are assigned to application B, which runs on host B. Application C is likely to require more than 512 volumes, so we use LSS pairs 28/29 and 2a/2b for this application.

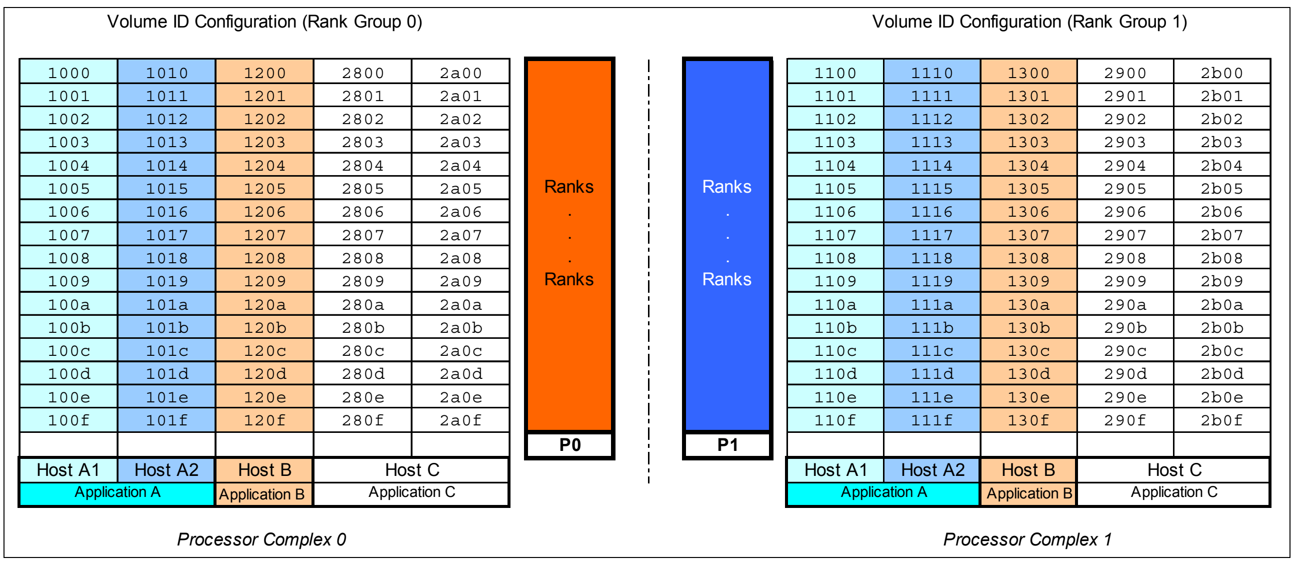


Figure 4-20 Application-related volume layout example for two shared extent pools

If an application workload is distributed across multiple extent pools on each processor complex, for example, in a four extent pool configuration, as shown in Figure 4-21 on page 144, you can expand this approach. Define a different volume range for each extent pool so that the system administrator can easily identify the extent pool behind a volume from the volume ID, which is transparent to the host systems.

In Figure 4-21 on page 144, we spread our workloads across four extent pools. Again, we assign two LSS/LCU IDs (one even, one odd) to each workload to spread the I/O activity evenly across both processor complexes (both rank groups). Additionally, we reserve a certain volume ID range for each extent pool based on the third digit of the volume ID. With this approach, you can quickly create volumes with successive volume IDs for a specific workload per extent pool with a single DSCLI **mkfbvol/mkckdvol** command.

Hosts A1 and A2 belong to the same application A and are assigned to LSS 10 and LSS 11. For this workload, we use volume IDs 1000 - 100f in extent pool P0 and 1010 - 101f in extent pool P2 on processor complex 0. And, we use volume IDs 1100 - 110f in extent pool P1 and 1110 - 111f in extent pool P3 on processor complex 1. In this case, the administrator on the host system can easily relate volumes to different extent pools and thus different physical resources on the same processor complex by looking at the third digit of the volume ID. This numbering scheme can be helpful when separating, for example, DB tablespaces from DB logs onto volumes from physically different pools.

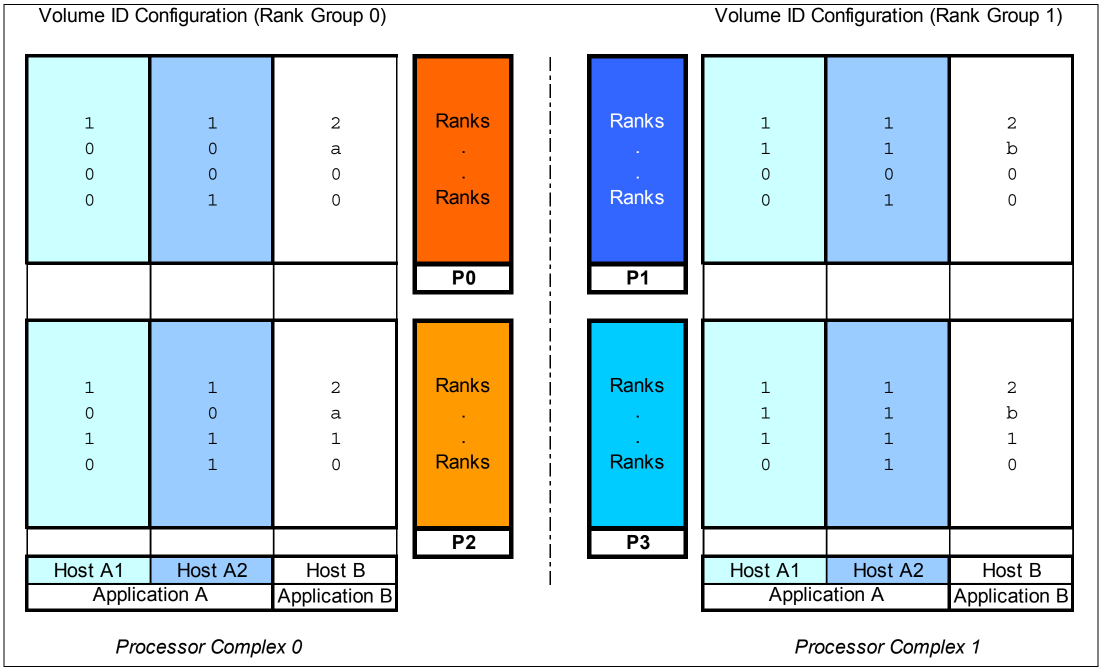


Figure 4-21 Application and extent pool-related volume layout example for four shared extent pools

In the example that is depicted in Figure 4-22 on page 145, we provide a numbering scheme that can be used in a FlashCopy or FlashCopy SE scenario. Two different pairs of LSS are used for source and target volumes. The address group identifies the role in the FlashCopy relationship: address group 1 is assigned to source volumes, and address group 2 is used for target volumes. This numbering scheme allows a symmetrical distribution of the FlashCopy relationships across source and target LSSs. For instance, source volume 1007 in P0 uses the volume 2007 in P2 as the FlashCopy target. We use the third digit of the volume ID within an LSS as a marker to indicate that source volumes 1007 and 1017 are from different extent pools. The same approach applies to the target volumes, for example, volumes 2007 and 2017 are from different pools. However for the simplicity of the Copy Services management, we chose a different extent pool numbering scheme for source and target volumes (so 1007 and 2007 are not from the same pool) to implement the recommended extent pool selection of source and target volumes in accordance with the FlashCopy guidelines. Source and target volumes must stay on the same rank group but different ranks or extent pools; see “Distribution of the workload: Location of source and target volumes” on page 589.

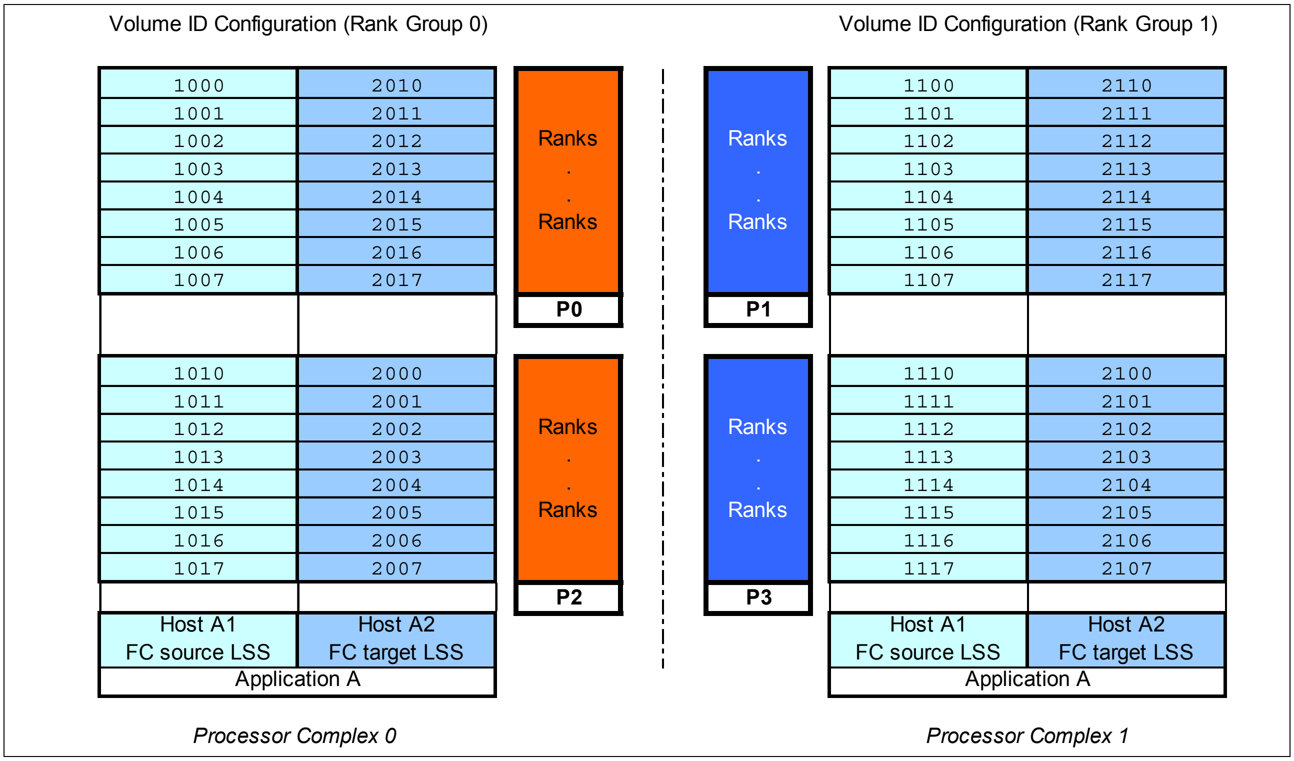


Figure 4-22 Application and extent pool-related volume layout example in a FlashCopy scenario

4.10 I/O port IDs, host attachments, and volume groups

Finally, when planning the attachment of the host system to the storage system HA I/O ports, you also need to achieve a balanced workload distribution across the available *front-end* resources for each workload with the appropriate isolation and resource-sharing considerations. Therefore, distribute the FC connections from the host systems evenly across the DS8800 HA ports, HA cards, and I/O enclosures.

For high availability, each host system must use a multipathing device driver, such as Subsystem Device Driver (SDD). Each host system must have a minimum of two host connections to HA cards in different I/O enclosures on the DS8800. Preferably, they are evenly distributed between left side (even-numbered) I/O enclosures and right side (odd-numbered) I/O enclosures. The number of host connections per host system is primarily determined by the required bandwidth. Use an appropriate number of HA cards to satisfy high throughput demands.

For DS8800, 4-port and 8-port HA card options are available. Each port can auto-negotiate a 2, 4, or 8 Gbps link speed. However, because the maximum available bandwidth is the same for 4-port and 8-port HA cards, the 8-port HA card provides additional connectivity and not more performance. Furthermore, the HA card maximum available bandwidth is less than the nominal aggregate bandwidth and depends on the workload profile. These specifications must be considered when planning the HA card port allocation and especially with workloads with high sequential throughputs. Be sure to contact your IBM representative or IBM Business Partner for an appropriate sizing, depending on your actual workload requirements. However, with typical transaction-driven workloads that show high numbers of random, small blocksize I/O operations, all ports in a HA card can be used likewise. For the best performance of workloads with different I/O characteristics, consider the isolation of large block sequential and small block random workloads at the I/O port level or the HA card level.

The preferred practice is to use dedicated I/O ports for Copy Services paths and host connections. For more information about performance aspects related to Copy Services, see 18.3.1, “Metro Mirror configuration considerations” on page 597.

To assign FB volumes to the attached Open Systems hosts by using LUN masking, these volumes need to be grouped in the DS8000 volume groups. A volume group can be assigned to multiple host connections, and each host connection is specified by the worldwide port name (WWPN) of the host FC port. A set of host connections from the same host system is called a *host attachment*. The same volume group can be assigned to multiple host connections; however, a host connection can only be associated with one volume group. To share volumes between multiple host systems, the most convenient way is to create a separate volume group for each host system and assign the shared volumes to each of the individual volume groups as required. A single volume can be assigned to multiple volume groups. Only if a group of host systems shares the same set of volumes, and there is no need to assign additional non-shared volumes independently to particular hosts of this group, can you consider using a single shared volume group for all host systems to simplify management. Typically, there are no significant DS8000 performance implications due to the number of DS8000 volume groups or the assignment of host attachments and volumes to the DS8000 volume groups.

Do not omit additional host attachment and host system considerations, such as SAN zoning, multipathing software, and host-level striping. For additional information, see Chapter 8, “Host attachment” on page 311, Chapter 9, “Performance considerations for UNIX servers” on page 327, “Chapter 12, “Performance considerations for Linux” on page 445, and Chapter 14, “Performance considerations for System z servers” on page 499.

After the DS8000 is installed, you can use the DSCLI **lsioport** command to display and document I/O port information, including the I/O ports, HA type, I/O enclosure location, and WWPN. Use this information to add specific I/O port IDs, the required protocol (FICON or FCP), and the DS8000 I/O port WWPNs to the plan of host and remote mirroring connections that is identified in 4.4, “Planning allocation of disk and host connection capacity” on page 98.

Additionally, the I/O port IDs might be required as input to the DS8000 host definitions if host connections need to be restricted to specific DS8000 I/O ports by using the **-ioport** option of the **mkhostconnect** DSCLI command. If host connections are configured to allow access to all DS8000 I/O ports, which is the default, typically, the paths must be restricted by SAN zoning. The I/O port WWPNs are required as input for SAN zoning. The **lshostconnect -login** DSCLI command might help to verify the final allocation of host attachments to the DS8000 I/O ports, because it lists host port WWPNs that are logged in, sorted by the DS8000 I/O port IDs for known connections. The **lshostconnect -unknown** DSCLI command might further help to identify host port WWPNs, which are not yet configured to host connections, when creating host attachments by using the **mkhostconnect** DSCLI command.

The DSCLI **lsioport** output identifies this information:

- The number of I/O ports on each installed HA
- The type of installed HAs (SW FCP/FICON-capable or LW FCP/FICON-capable)
- The distribution of HAs across I/O enclosures
- The WWPN of each I/O port

The DS8000 I/O ports use predetermined, fixed DS8000 *logical* port IDs in the form I0*xyz* where:

- *x* = I/O enclosure
- *y* = slot number within the I/O enclosure

- *z* = port within the adapter

For example, I0101 is the I/O port ID for these devices:

- I/O enclosure 1
- Slot 0
- Second port

Slot numbers: The slot numbers for logical I/O port IDs are one less than the physical location numbers for HA cards, as shown on the physical labels and in Tivoli Storage Productivity Center for Disk, for example, I0101 is R1-XI2-C1-T2.

A simplified example of spreading the DS8000 I/O ports evenly to two redundant SAN fabrics is shown in Figure 4-23. The SAN implementations can vary, depending on individual requirements, workload considerations for isolation and resource-sharing, and available hardware resources.

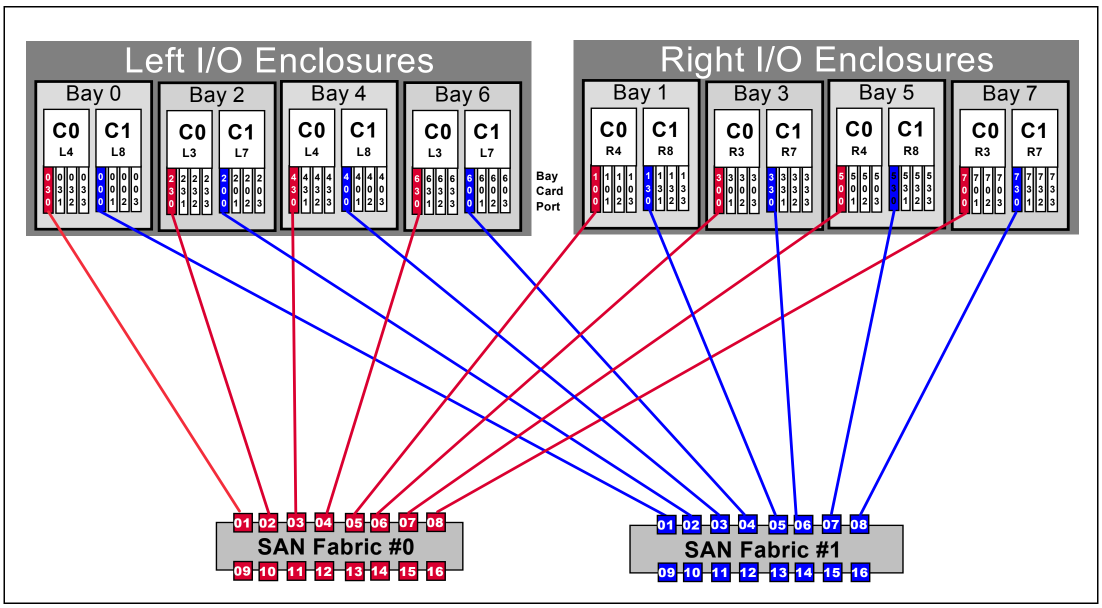


Figure 4-23 Example of spreading DS8000 I/O ports evenly across two redundant SAN fabrics

4.10.1 I/O port planning considerations

We offer several suggestions for the DS8000 I/O port planning and assignment:

- The ports on the 8 Gbps HA card can be connected in any order or position, and it delivers the same performance.
- The performance of an 8 Gbps HA card does not scale with more than four ports. So, for Fibre Channel and FICON paths with high utilization, do not use more than *four* ports on a DS8800 8 Gbps HA (more than four ports might increase the number of required HAs). Avoid using all eight ports of the 8-port HA cards for demanding workloads.
- Spread the paths from all host systems across the available I/O ports, HA cards, and I/O enclosures to optimize workload distribution across the available resources depending on your workload sharing and isolation considerations.
- Spread the host paths that access the same set of volumes as evenly as possible across the available HA cards and I/O enclosures. This approach balances workload across

hardware resources, and it ensures that a hardware failure does not result in a loss of access.

- Plan the paths for the attached host systems with a *minimum* of two host connections to different HA cards in different I/O enclosures on the DS8800. Preferably, evenly distribute them between left side (even-numbered) I/O enclosures and right side (odd-numbered) I/O enclosures for the highest availability and a balanced workload distribution across I/O enclosures and HA cards.
- Use separate DS8000 I/O ports for host attachment and Copy Services remote replication connections (such as Metro Mirror, Global Mirror, and zGM data mover). If additional HAs are available, consider using separate HAs for Copy Services remote replication connections and host attachments to avoid any possible interference between remote replication and host workloads.
- Spread Copy Services remote replication connections at least across two HA cards in different I/O enclosures.
- Consider using separate HA cards for FICON protocol and FCP. Although I/O ports on the same HA can be configured independently for the FCP protocol and the FICON protocol, it might be preferable to isolate your z/OS environment (FICON) from your Open Systems environment (FCP).

Consider a DS8800 single frame with four fully populated I/O enclosures. A schematic of the I/O enclosures is shown in Figure 4-24. It shows that each of the four base frame I/O enclosures (0 - 3) contains two 8-port FCP/FICON-capable HAs.

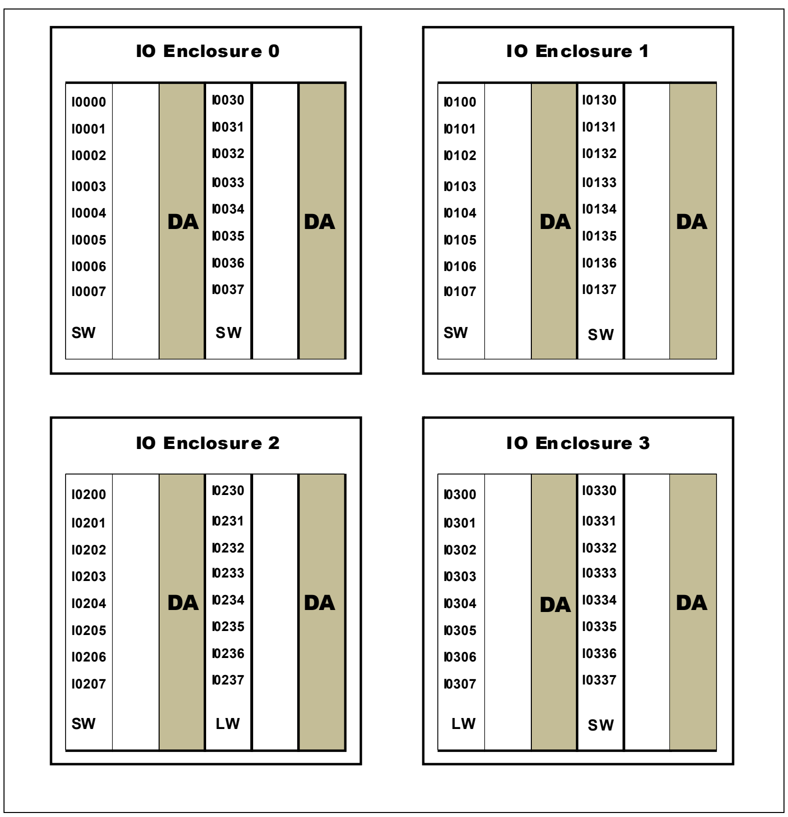


Figure 4-24 DS8800 example: I/O enclosures, HAs, and I/O ports

The DSCLI **lsioport** command output for the DS8800 in Example 4-6 shows a total of 64 available I/O ports:

- Six multi-mode (shortwave) 8-port FCP/FICON-capable adapters:
  - Two in I/O enclosures 0 and 1, and one in I/O enclosures 2 and 3
  - IDs I0000 - I0007, I0030 - I0037, I0100 - I0107, I0130 - I0137, I0200 - I0207, and I0330 - I0337
- Two single-mode (longwave) 8-port FCP/FICON-capable adapters:
  - One in each I/O enclosure 2 and 3
  - IDs I0230 - I0237 and I0330 - I0337

Example 4-6 DS8800 example 1: DSCLI lsioport command output

```
dscli> lsioport -l
ID    WWPN             State  Type             topo     portgrp Speed
=====================================================================
I0000 500507630A00029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0001 500507630A00429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0002 500507630A00829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0003 500507630A00C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0004 500507630A40029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0005 500507630A40429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0006 500507630A40829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0007 500507630A40C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0030 500507630A03029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0031 500507630A03429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0032 500507630A03829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0033 500507630A03C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0034 500507630A43029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0035 500507630A43429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0036 500507630A43829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0037 500507630A43C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0100 500507630A08029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0101 500507630A08429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0102 500507630A08829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0103 500507630A08C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0104 500507630A48029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0105 500507630A48429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0106 500507630A48829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0107 500507630A48C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0130 500507630A0B029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0131 500507630A0B429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0132 500507630A0B829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0133 500507630A0BC29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0134 500507630A4B029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0135 500507630A4B429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0136 500507630A4B829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0137 500507630A4BC29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0200 500507630A10029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0201 500507630A10429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0202 500507630A10829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0203 500507630A10C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0204 500507630A50029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0205 500507630A50429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0206 500507630A50829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0207 500507630A50C29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0230 500507630A13029F Online Fibre Channel-LW FICON    0       8 Gb/s
I0231 500507630A13429F Online Fibre Channel-LW FICON    0       8 Gb/s
```

```
I0232 500507630A13829F Online Fibre Channel-LW FICON    0       8 Gb/s
I0233 500507630A13C29F Online Fibre Channel-LW FICON    0       8 Gb/s
I0234 500507630A53029F Online Fibre Channel-LW FICON    0       8 Gb/s
I0235 500507630A53429F Online Fibre Channel-LW FICON    0       8 Gb/s
I0236 500507630A53829F Online Fibre Channel-LW FICON    0       8 Gb/s
I0237 500507630A53C29F Online Fibre Channel-LW FICON    0       8 Gb/s
I0300 500507630A18029F Online Fibre Channel-LW FICON    0       8 Gb/s
I0301 500507630A18429F Online Fibre Channel-LW FICON    0       8 Gb/s
I0302 500507630A18829F Online Fibre Channel-LW FICON    0       8 Gb/s
I0303 500507630A18C29F Online Fibre Channel-LW FICON    0       8 Gb/s
I0304 500507630A58029F Online Fibre Channel-LW FICON    0       8 Gb/s
I0305 500507630A58429F Online Fibre Channel-LW FICON    0       8 Gb/s
I0306 500507630A58829F Online Fibre Channel-LW FICON    0       8 Gb/s
I0307 500507630A58C29F Online Fibre Channel-LW FICON    0       8 Gb/s
I0330 500507630A1B029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0331 500507630A1B429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0332 500507630A1B829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0333 500507630A1BC29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0334 500507630A5B029F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0335 500507630A5B429F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0336 500507630A5B829F Online Fibre Channel-SW FC-AL    0       8 Gb/s
I0337 500507630A5BC29F Online Fibre Channel-SW FC-AL    0       8 Gb/s
```

I/O topology configurations: As seen in Example 4-6, the default I/O topology configurations are FICON for longwave (LW) HAs and Fibre Channel Arbitrated Loop (FC-AL) for shortwave (SW) HAs.

Consider the following connectivity requirements:

- Eight FICON ports
- Sixteen FCP ports for host connectivity
- Four FCP ports for PPRC connectivity
- Four FC-AL ports for direct connection to hosts

By assuming that the FICON attachments use LW ports (typical for FICON environments), we can reserve the LW cards for FICON attachments only and spread all the FCP attachments across all the SW cards. So, following the general guidelines described before, the following list is a possible IO port assignment:

- FICON connections:
  - I0230
  - I0232
  - I0234
  - I0236
  - I0300
  - I0302
  - I0304
  - I0306
- FCP host connections:
  - I0000
  - I0002
  - I0004
  - I0030
  - I0032
  - I0100
  - I0102

- I0104
- I0130
- I0132
- I0200
- I0202
- I0204
- I0330
- I0332
- I0334

- FCP PPRC connections:
  - I0036
  - I0106
  - I0206
  - I0336
- FC-AL connections:
  - I0006
  - I0034
  - I0104
  - I0136

In this case, the FCP PPRC ports share the adapter with FCP host ports. In general, sharing the adapter bandwidth between PPRC and the host workload might lead to performance issues especially during the PPRC full-copy operations. To avoid any possible interference between PPRC and host workloads, you must use dedicated HAs for PPRC (or zGM data mover) connections. For availability reasons, we advise that you spread the PPRC connection across at least two HAs. The drawback of isolating the PPRC connections is that, with a few PPRC connections, this approach leads to a waste of ports. In our example, we compromised by using only one port per adapter for PPRC connectivity. In this way, even in PPRC full copy, the PPRC port does not affect the performance of the other host ports significantly.

When planning the paths for the host systems, ensure that each host system uses a multipathing device driver, such as Subsystem Device Driver (SDD). Ensure that each host system has a minimum of two host connections to two different HA cards in different I/O enclosures on the DS8800. Preferably, they are evenly distributed between left side (even-numbered) I/O enclosures and the right side (odd-numbered) I/O enclosures for highest availability. Multipathing additionally optimizes workload spreading across the available I/O ports, HA cards, and I/O enclosures.

You must tune the SAN zoning scheme to balance both the oversubscription and the estimated total throughput for each I/O port to avoid congestion and performance bottlenecks.

4.11 Implement and document DS8000 logical configuration

For performance management and analysis reasons, it is crucial to be able to easily relate *volumes*, which are related to a specific I/O workload, to ranks, which finally provide the physical disk spindles for servicing the workload I/O requests and determining the I/O processing capabilities. An overall logical configuration concept that easily relates volumes to workloads, extent pools, and ranks is desirable.

After the logical configuration is planned, you can use either the DS Storage Manager or the DSCLI to implement it on the DS8000 in the following steps:

1. Change the password for the default user (*admin*) for DS Storage Manager and DSCLI.

2. Create additional user IDs for DS Storage Manager and DSCLI.
3. Apply the DS8000 authorization keys.
4. Create arrays.
5. Create ranks.
6. Create extent pools.
7. Assign ranks to extent pools.
8. Create CKD Logical Control Units (LCUs).
9. Create CKD volumes.
10. Create CKD PAVs.
11. Create FB LUNs.
12. Create Open Systems host definitions.
13. Create Open Systems DS8000 volume groups.
14. Assign Open Systems hosts and volumes to the DS8000 volume groups.
15. Configure I/O ports.
16. Implement SAN zoning, multipathing software, and host-level striping, as desired.

After the logical configuration is created on the DS8000, it is important to document it.

You can use the DS Storage Manager to export information in a spreadsheet format (that is, save it as a comma-separated values (CSV) file). You can use this information together with a planning spreadsheet to document the logical configuration. For more information and examples, see Appendix C, “Planning and documenting your logical configuration” on page 651.

The DSCLI provides a set of **`list (ls)`** and **`show`** commands, which can be redirected and appended into a plain text or CSV file. A list of selected DSCLI commands, as shown in Example 4-7, can be started as a DSCLI script (using the DSCLI command **`dscli -script`**) to collect the logical configuration of a DS8000 Storage Image. This output can be used as a text file or imported into a spreadsheet to document the logical configuration.

Example 4-7 only collects a minimum set of the DS8000 logical configuration information, but it illustrates a simple DSCLI script implementation and runs quickly within a single DSCLI command session. Depending on the environment, you can modify this script to include more commands to provide more information, for example, about Copy Services configurations and source/target relationships. The DSCLI script terminates with the first command that returns an error, which, for example, can be a simple **`lslcu`** command if no LCUs are defined. You can adjust the output of the **`ls`** commands in a DSCLI script to meet special formatting and delimiter requirements by using appropriate options for `format`, `delim`, or `header` in the specified DS8000 profile file or selected **`ls`** commands.

Example 4-7 Example of a minimum DSCLI script get_config.dscli to gather the logical configuration

```
> dscli -cfg profile/DEVICE.profile -script get_config.dscli > DEVICE_SN_config.out
CMMCI9029E showrank: rank R48 does not exist.

> cat get_config.dscli
ver -l
lssu -l
lssi -l

lsarraysite -l
lsarray -l
lsrank -l
lsextpool -l

lsaddressgrp
lslss                  # Use only if FB volumes have been configured
#lslcu                 # Use only if CKD volumes and LCUs have been configured
```

```
                      # otherwise the command returns an error and the script terminates.
lsioport -l
lshostconnect
lsvolgrp

lsfbvol -l            # Use only if FB volumes have been configured
#lsckdvol -l          # Use only if CKD volumes have been configured
                      # otherwise the command returns an error and the script terminates.
showrank R0           # Modify this list of showrank commands so that
showrank R1           # the showrank command is run on all available ranks!
showrank R2           # Note that an error is returned if the specified rank is not
showrank R3           # present. The script terminates on the first non-existing rank.
...                   # Check for gaps in the rank ID sequence.
showrank R192
```

The script in Example 4-7 on page 152 is focused on providing the relationship between volumes, ranks, and hosts, and can easily be used on different DS8000 systems without modification or the need to consider the particular storage image ID of the DS8000 system. However, you can further enhance the script by adding the commands that are shown in Example 4-8 to include hardware-specific information about the DS8000 system, which is helpful when performing a deeper performance analysis. In this case, you need to specify the storage unit or storage image ID correctly in the script for each DS8000 storage system.

Example 4-8 Additional DSCLI commands to include DS8000 machine-specific information

```
##NOTE: Please substitute the actual Serial# / Device-ID (assuming a single image machine)
##      of the DS8700/DS8800 for the “xxxx” below in the identified commands:
##
showsu    IBM.2107-75xxxx0
showsi    IBM.2107-75xxxx1
lskey     IBM.2107-75xxxx1
lsddm     IBM.2107-75xxxx1
lsda      IBM.2107-75xxxx1
lsframe   IBM.2107-75xxxx1
lshba     IBM.2107-75xxxx1
lsstgencl IBM.2107-75xxxx1
```

Part 2

DS8000 performance management

This part starts with a brief introduction of different workload types and introduces various tools for effective performance planning, monitoring, and management on the IBM System Storage DS8000 system.

This part includes the following topics:

- Understanding your workload
- Performance planning tools
- Practical performance management

5

Understanding your workload

In this chapter, we present and discuss the various workload types that an application can generate. This characterization can be useful for understanding performance documents and reports, as well as categorizing the various workloads in your installation.

Information in this chapter is not dedicated to the IBM System Storage DS8000. You can apply this information generally to other disk storage systems.

5.1 General workload types

The correct understanding of the existing or planned workload is the key element of the entire planning and sizing process. Understanding the workload means having the description of the workload pattern:

- Expected or existing number of IOPS
- Size of the I/O requests
- Read and write ratio
- Random and sequential access ratio
- General purpose of the application and the workload

You might also collect the following information:

- Expected cache hit ratio

  The number of requests serviced from cache. This number is important for read requests, because write requests always get into the cache first. If of 1000 requests, 100 of them are serviced from cache, you have a 10% of cache hit ratio. The higher this parameter, the lower the overall response time.

- Expected seek ratio

  The percentage of the I/O requests for which the disk arm must move from its current location. Moving the disk arm requires more time than rotating the disk, which rotates anyway and is fast enough. So, by not moving the disk arm, the whole track or a cylinder can be read, which generally means large amount of data. This parameter is mostly indicative of how disk subsystems worked a long time ago, and it is now turned into a sort of the quality value of the random nature of the workload. A random workload shows this value close to 100%. This parameter is not applicable to the solid state drives (SSDs) that have no disk arms by design.

- Expected write efficiency

  The write efficiency is a number that represents the number of times a block is written to before being destaged to the disk. Actual applications, especially databases, update the information: write-read again-change-write with changes. So, the data for the single disk block can be served several times from cache before it is written to the disk. A value of 0% means that a destage is assumed for every write operation and the characteristic of the "pure random small block write workload pattern", which is unlikely. A value of 50% means that a destage occurs after the track is written to twice. A value of 100% is unlikely also, because it means that writes come to the one track only and are never destaged to disk.

In general, you describe the workload in these terms. The following sections cover the details and describe the different workload types.

5.1.1 Typical online transaction processing (OLTP) workload

This workload is characterized by mostly the random access of small-sized I/O records (less than or equal to 16 KB) with a mix of 70% reads and 30% writes. This workload is also characterized by low read-hit ratios in the disk subsystem cache (less than 20%). This workload might be representative of various online applications, for example, the SAP R/3 application or many database applications. This type of workload is typical, because it is the basis for most of the benchmark and performance tests. However, in actuality, the following OLTP patterns are spread:

- 90% read, 8 - 16 - 32 - 64 KB blocks, 30% sequential, 30 - 40% cache hit
- 80% read, 16 - 32 - 64 KB blocks, 40 - 50% sequential, 50 - 60% cache hit

A 100% random access workload is rare, which you must remember when you size the disk system.

5.1.2 Microsoft Exchange Server workload

This type of workload can be similar to an OLTP workload, but it has a different read-write balance. It is characterized by many write operations, up to 60%, with high random numbers. Also, the size of the I/O can be high, with blocks up to 128 KB, which is explained by the nature of the application. It acts as a database and the data warehouse, as well. Size this type of workload with the Microsoft tools and advice. You might want to read the following documents to better understand the workload and storage configurations:

- http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101282
- http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101798
- http://technet.microsoft.com/en-us/exchange/dd203064

5.1.3 Sequential workload

Sequential access is one of the original workload types for data storage. Disks did not exist, only tapes. Tape is the best example of sequential access. Tape uses several large blocks at one read operation, and it uses buffers first. Sequential access does not change for the disks. Sequential workload is good for prefetch and putting data into cache, because blocks are accessed one by one and the disk system can read many blocks to unload the disks. Sequential write requests work well, because the disk system can optimize the access to the disks and write several tracks or cylinders at a time. Blocksizes are typically 256 KB or more and response time is high, but they do not matter. Sequential workload is about bandwidth, not response time. The following environments and applications are likely sequential workloads:

- Backup/restore applications
- Database log files
- Batch processing
- File servers
- Web servers
- Media streaming applications
- Graphical software

Important: Because of large block access and high response times, separate sequential workload physically from the random small-block workload. Do not mix random and sequential workloads on the same physical disk. If you do, large amounts of cache are required on the disk system. Typically, high response times with small-block random access mean the presence of the sequential write activity (foreground or background) on the same disks.

5.1.4 Batch jobs workload

Batch workloads have several common characteristics:

- Mixture of random database access, skip-sequential, pure sequential, and sorting.
- Large blocksizes up to 128 - 256 KB
- High volume of write activity, as well as read activity
- The same volume extents might scramble for write and read at the same time.
- Batch workloads include large data transfers and high path utilizations.
- Batch workloads are often constrained to operate within a particular window of time when online operation is restricted or shut down. Poor or improved performance is often not recognized unless it affects this window.

Plan when to run batch jobs: Plan all batch workload activity for the end of the day or to a slower time of day. Normal activity can be negatively affected with the batch activity.

5.1.5 Sort jobs workload

Most sorting applications, such as the z/OS DFSORT, are characterized by large transfers for input, output, and work datasets.

For more information about z/OS DFSORT, see these links:

- http://www.ibm.com/support/docview.wss?rs=114&uid=isg3T7000077
- http://publib.boulder.ibm.com/infocenter/zos/v1r11/index.jsp?topic=/com.ibm.zos.r11.iceg200/abstract.htm

5.1.6 Read-intensive cache friendly and unfriendly workloads

Use the cache hit ratio to estimate the cache-friendly read workload. If the ratio is more than 50%, the workload is cache friendly. If you have two serviced I/Os to the same data and one I/O is serviced from cache, it is a 50% cache hit ratio.

It is not as easy to divide known workload types into cache friendly and cache unfriendly. An application can change its behavior during the day several times. When users work with data, it is cache friendly. When the batch processing or reporting starts, it is not cache friendly. High random-access numbers mean a not cache-friendly workload type. However, if the amount of data that is accessed randomly is not large, 10% for example, it can be placed totally to the disk system cache and becomes cache friendly.

Sequential workloads are always cache friendly, because of prefetch algorithms that exist in the disk system. Sequential workload is easy to prefetch. You know that the next 10 or 100 blocks are definitely accessed, and you can read them in advance. For the random workloads, it is different. There are no purely random workloads in the actual applications and it is possible to predict some moments. DS8800 and DS8700 use the following powerful read-caching algorithms to deal with cache unfriendly workloads:

- Sequential Prefetching in Adaptive Replacement Cache
- Adaptive Multi-stream Prefetching

The write workload is always cache friendly, because every write request comes to the cache first and the application gets the reply as soon as the request is placed into cache. Write requests are served at least two times longer by the back end than read requests. You always need to wait for the write acknowledgment, which is why cache is used for every write request. However, improvement is possible. The DS8800 and DS8700 systems use the Intelligent Write Caching algorithm, which makes work with write requests more effective.

See the following links to learn more about the DS8000 caching algorithms:

- http://www.redbooks.ibm.com/abstracts/sg248886.html
- http://www.almaden.ibm.com/storagesystems/resources/amp.pdf
- http://www.almaden.ibm.com/cs/people/dmodha/sarc.pdf

Table 5-1 on page 161 provides a summary of the characteristics of the various types of workloads.

Table 5-1 Workload types

Workload type	Characteristics	Representative of this type of process
Sequential read	Sequential 128 - 1024 KB blocks Read hit ratio: 100%	Database backups Batch processing Media streaming
Sequential write	Sequential 128 - 1024 KB blocks Write ratio: 100%	Database restores and loads Batch processing File access
z/OS cache uniform	Random 4 KB record R/W ratio 3.4 Read hit ratio 84%	Average database CICS/VSAM IMS™
z/OS cache standard	Random 4 KB record R/W ratio 3.0 Read hit ratio 78%	Representative of typical database conditions
z/OS cache friendly	Random 4 KB record R/W ratio 5.0 Read hit ratio 92%	Interactive Existing software
z/OS cache hostile	Random 4 KB record R/W ratio 2.0 Read hit ratio 40%	DB2 logging
Open read-intensive	Random 4 - 8 - 16 - 32 KB record Read% = 70 - 90% Hit ratio 28%, 1%	Databases (Oracle and DB2) Large DB inquiry Decision support Warehousing
Open standard	Random 4 - 8 - 16 - 32 KB record Read% = 70% Hit ratio 50%	OLTP General Parallel File System (GPFS™)

5.2 Database workload

Database workload does not come with the database initially. It depends on the application that is written for the database and the type of work that this application performs. The workload can be an OLTP workload or a data warehousing workload in the same database. When we refer to a database in this section, we mean DB2 and Oracle databases mostly, but this section can apply to other databases also. For more information, see Chapter 17, “Database performance” on page 559.

The Database environment is often difficult to typify, because I/O characteristics differ greatly. Database query has a high read content and is of a sequential nature. It also can be random, depending on the query type and data structure. Transaction environments are more random in behavior and are sometimes cache unfriendly. At other times, they have good hit ratios. You can implement several enhancements in databases, such as sequential prefetch and the exploitation of I/O priority queuing, that affect the I/O characteristics. Users need to understand the unique characteristics of their database capabilities before generalizing the performance.

5.2.1 Database query workload

Database query is a common type of database workload. This term includes transaction processing that is typically random, as well as sequential data reading, writing, and updating. The query can have following properties:

- High read content
- Mix of write and read content
- Random access, as well as sequential access
- Small or large transfer size

A well-tuned database keeps characteristics of the queries closer to a sequential read workload. A database can use all available caching algorithms, both its own and the disk system algorithms. This functionality, which caches data that has the most probability to be accessed, provides performance improvements for most database queries.

5.2.2 Database logging workload

The logging system is an important part of the database. It is the main component to preserve the data integrity and provide a transaction mechanism. There are several types of log files in the database:

- Online transaction logs. This type of log is used to restore the last condition of the database when the latest transaction failed. A transaction can be complex and require several steps to complete. Each step means changes in the data. Because data in the database must be in a consistent state, an uncompleted transaction must be rolled back to the initial state of the data. Online transaction logs have a rotation mechanism that creates and uses several small files for about an hour each.
- Archive transaction logs. This type is used to restore a database state up to the specified date. Typically, it is used with incremental or differential backups. For example, if you identify a data error that occurred a couple of days ago, you can restore the data back to its prior condition with only the archive logs. This type of log uses a rotation mechanism also.

The workload pattern for the logging is sequential writes mostly. Blocksize is about 64 KB. Reads are rare and might not be considered. The write capability and location of the online transaction logs are most important. The entire performance of the database depends on the writes to the online transaction logs. If you expect high write rates to the database, plan for RAID 10 on to which to place the online transaction logs. Also, we strongly advise that log files must be physically separate from the disks on which the data and index files reside. For more information, see Chapter 17, “Database performance” on page 559.

5.2.3 Database transaction environment workload

Database transaction workloads have these characteristics:

- Low to moderate read hits, depending on the size of the database buffers
- Cache unfriendly for certain applications
- Deferred writes that cause low write-hit ratios, which means that cached write data is rarely required for reading
- Deferred write chains with multiple locate-record commands in chain
- Low read/write ratio due to reads that are satisfied in a large database buffer pool
- High random-read access values, which is cache unfriendly

The enhanced prefetch cache algorithms, together with the high storage back-end bandwidth, provide high subsystem throughput and high transaction rates for database transaction-based workloads.

A database can benefit from using a large amount of server memory for the large buffer pool. For example, the database large buffer pool, when managed properly, can avoid a large percentage of the accesses to disk. Depending on the application and the size of the buffer pool, this large buffer pool can translate poor cache hit ratios into synchronous reads in DB2. You can spread data across several RAID arrays to increase the throughput even if all accesses are read misses. DB2 administrators often require that tablespaces and their indexes are placed on separate volumes. This configuration improves both availability and performance.

5.2.4 Database utilities workload

Database utilities, such as loads, reorganizations, copies, and recovers, generate high read and write sequential and sometimes random operations. This type of workload takes advantage of the sequential bandwidth performance of the back-end storage connection, such as the PCI-Express bus for the device adapter (DA) pairs, and the use of higher rpm (15K) drives with SSDs and Easy Tier automatic mode enabled.

5.3 Application workload

This section categorizes various types of common applications according to their I/O behavior. There are four typical categories:

- Need for high throughput. These applications need more bandwidth (the more, the better). Transfers are large, read-only I/Os and, typically, sequential access. These applications use database management systems (DBMSs); however, random DBMS access might also exist.
- Need for high throughput and a mix of read/write (R/W), similar to the first category (large transfer sizes). In addition to 100% read operations, this category mixes reads and writes in 70/30 and 50/50 ratios. The DBMS is typically sequential, but random and 100% write operations also exist.
- Need for high I/O rate and throughput. This category requires both performance characteristics of IOPS and megabits per second (MBps). Depending on the application, the profile is typically sequential access, medium to large transfer sizes (16 KB, 32 KB, and 64 KB), and 100/0, 0/100, and 50/50 R/W ratios.
- Need for high I/O rate. With many users and applications that run simultaneously, this category can consist of a combination of small to medium-sized transfers (4 KB, 8 KB, 16 KB, and 32 KB), 50/50 and 70/30 R/W ratios, and a random DBMS.

Synchronous activities: Certain applications have synchronous activities, such as locking database tables during an online backup, or logging activities. These types of applications are highly sensitive to any increase in disk response time and must be handled with care.

Table 5-2 on page 164 summarizes these workload categories and common applications.

Table 5-2 Application workload types

Category	Application	Read/write ratio	I/O size	Access type
4	General file serving	Expect 50/50	64 - 256 KB	Sequential mostly due to good filesystem caching
4	Online transaction processing	50/50, 70/30	4 KB, 8 KB, 16 KB	Random mostly for writes and reads. Bad cache hits
4	Batch update	Expect 50/50	16 KB, 32 KB, 64 KB, 128 KB	Almost 50/50 mix of sequential and random. Moderate cache hits.
1	Data mining	90/10	32 KB, 64 KB, or larger	Mainly sequential, some random
1	Video on demand	100/0	256 KB and larger	Sequential, good caching
2	Data warehousing	90/10, 70/30, 50/50	64 KB or larger	Mainly sequential, rarely random, good caching
2	Engineering and scientific	100/0, 0/100, 70/30, 50/50	64 KB or larger	Sequential mostly, good caching
3	Digital video editing	100/0, 0/100, 50/50	128 KB, 256 - 1024 KB	Sequential, good caching
3	Image processing	100/0, 0/100, 50/50	64 KB, 128 KB	Sequential, good caching
1	Backup, restore	100/0, 0/100	256 - 1024 KB	Sequential, good caching

5.3.1 General file serving

This application type consist of many users who run many different applications, all with varying file access sizes and mixtures of read/write ratios, all occurring simultaneously. Applications can include file server, LAN storage, disk arrays, and even Internet/intranet servers. There is no standard profile, other than the “chaos” principle of file access. General file serving fits this application type, because this profile covers almost all transfer sizes and R/W ratios.

5.3.2 Online transaction processing

This application category typically has many users, all accessing the same disk storage subsystem and a common set of files. The file access typically is under the control of a DBMS, and each user might work on the same or unrelated activities. The I/O requests are typically spread across many files; therefore, the file sizes are typically small and randomly accessed. A typical application consists of a network file server or a disk subsystem that is accessed by a sales department that enters order information.

5.3.3 Data mining

Databases are the repository of most data, and every time that information is needed, a database is accessed. *Data mining* is the process of extracting valid, previously unknown, and ultimately comprehensive information from large databases to make crucial business decisions. This application category consists of a number of operations, each of which is supported by various techniques, such as rule induction, neural networks, conceptual clustering, and association discovery. In these applications, the DBMS only extracts large sequential or possibly random files, depending on the DBMS access algorithms.

5.3.4 Video on demand

Video on demand consists of video playback that can be used to broadcast quality video for either satellite transmission or a commercial application, such as in-room movies. Fortunately for the storage industry, the current data rates that are needed for this type of transfer are reduced dramatically due to data compression developments. A broadcast quality video stream, for example, Full HD video, now only needs about 4 - 5 Mbps bandwidth to serve a single user. These advancements reduce the need for higher speed interfaces and can be serviced with the current interface. However, these applications demand numerous concurrent users that interactively access multiple files within the same storage subsystem. This requirement changed the environment of video applications, because the storage subsystem is specified by a number of video streams that they can service simultaneously. In this application, the DBMS only extracts large sequential files.

5.3.5 Data warehousing

A data warehouse supports information processing by providing a solid platform of integrated, historical data from which to perform analysis. A data warehouse organizes and stores the data that is needed for informational and analytical processing over a long historical time period. A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data that is used to support the management decision making process. A data warehouse is always a physically separate store of data that spans a spectrum of time, and many relationships exist in the data warehouse.

An example of a data warehouse is a design around a financial institution and its functions, such as loans, savings, bank cards, and trusts for a financial institution. In this application, there are three kinds of operations: initial loading of the data, access to the data, and updating of the data. However, due to the fundamental characteristics of a warehouse, these operations can occur simultaneously. At times, this application can perform 100% reads when accessing the warehouse; 70% reads and 30% writes when accessing data while record updating occurs simultaneously; or even 50% reads and 50% writes when the user load is heavy. Remember that the data within the warehouse is a series of snapshots and after the snapshot of data is made, the data in the warehouse does not change. Therefore, there is typically a higher read ratio when using the data warehouse.

5.3.6 Engineering and scientific applications

The engineering and scientific arena includes hundreds of applications. Typical applications are computer-assisted design (CAD), Finite Element Analysis, simulations and modeling, and large scale physics applications. Transfers can consist of 1 GB of data for 16 users. Other transfers might require 20 GB of data and hundreds of users. The engineering and scientific areas of business are more concerned with the manipulation of spatial data and series data. This application typically goes beyond standard relational DBMSs, which manipulate only flat (two-dimensional) data. Spatial or multi-dimensional issues and the ability to handle complex data types are commonplace in engineering and scientific applications.

Object-Relational DBMS (ORDBMS) are now being developed, and they not only offer traditional relational DBMS features, but also support complex data types. Objects can be stored and manipulated, and complex queries at the database level can be run. Object data is data about real objects, including information about their location, geometry, and topology. Location describes their position, geometry relates to their shape, and topology includes their relationship to other objects. These applications essentially have an identical profile to that of the data warehouse application.

5.3.7 Digital video editing

Digital video editing is popular in the movie industry. The idea that a film editor can load entire feature films onto disk storage and interactively edit and immediately replay the edited clips has become a reality. This application combines the ability to store huge volumes of digital audio and video data onto relatively affordable storage devices to process a feature film.

Depending on the host and operating system that are used to perform this application, transfers are typically medium to large in size and access is always sequential. Image processing consists of moving huge image files for editing. In these applications, the user regularly moves huge high-resolution images between the storage device and the host system. These applications service many desktop publishing and workstation applications. Editing sessions can include loading large files of up to 16 MB into host memory, where users edit, render, modify, and eventually store data back onto the storage system. High interface transfer rates are needed for these applications, or the users waste huge amounts of time by waiting to see results. If the interface can move data to and from the storage device at over 32 MBps, an entire 16 MB image can be stored and retrieved in less than one second. The need for throughput is all important to these applications, and, along with the additional load of many users, I/O operations per second are also a major requirement.

5.4 Profiling workloads in the design phase

Assessing the I/O profile prior to the build and deployment of the application requires methods of evaluating the workload profile without measurement data. In these cases, we suggest using a combination of general rules based on application type and the development of an application I/O profile by the application architect or the performance architect. The following examples are basic examples that are designed to provide an idea how to approach workload profiling in the design phase.

For general rules for application types, see Table 5-1 on page 161.

To following requirements apply to developing an application I/O profile:

- User population

  Determining the user population requires understanding the total number of potential users, which for an online banking application might represent the total number of customers. From this total population, you need to derive the active population that represents the average number of persons using the application at any specific time, which is derived from experiences with other similar applications.

In Table 5-3, we use 1% of the total population. From the average population, we estimate the peak. The peak workload is some multiplier of the average and is typically derived based on experience with similar applications. In this example, we use a multiple of 3.

Table 5-3 User Population

Total potential users	Average active users	Peak active users
50000	500	1500

- Transaction distribution

  Table 5-4 breaks down the number of times that key application transactions are executed by the average user and how much I/O is generated per transaction. Detailed application and database knowledge is required to identify the number of I/Os and the type of I/Os per transaction. The following information is a sample.

Table 5-4 Transaction distribution

Transaction	Iterations per user	I/Os	I/O type
Look up savings account	1	4	Random read
Look up checking account	1	4	Random read
Transfer money to checking	.5	4 reads/4 writes	Random read/write
Configure new bill payee	.5	4 reads/4 writes	Random read/write
Submit payment	1	4 writes	Random write
Look up payment history	1	24 reads	Random read

- Logical I/O profile

  An I/O profile is created by combining the user population and the transaction distribution. Table 5-5 provides an example of a logical I/O profile.

Table 5-5 Logical I/O profile from user population and transaction profiles

Transaction	Iterations per user	I/Os	I/O type	Average user I/Os	Peak users
Look up savings account	1	4	Random read (RR) I/Os	2000	6000
Look up checking account	1	4	RR	2000	6000
Transfer money to checking	.5	4 reads/4 writes	RR, random write I/Os (RW)	1000, 1000	3000 R/W
Configure new bill payee	.5	4 reads/4 writes	RR, RW	1000, 1000	3000 R/W
Submit payment	1	4 writes	RW	2000	6000 R/W
Look up payment history	1	24 reads	RR	12000	36000

- Physical I/O profile

  The physical I/O profile is based on the logical I/O with the assumption that the database provides cache hits to 90% of the read I/Os. All write I/Os are assumed to require a physical I/O. This physical I/O profile results in a read miss ratio of (1 - 0.9) = .1 or 10%. Table 5-6 is an example, and every application has different characteristics.

Table 5-6 Physical I/O profile

Transaction	Average user logical I/Os	Average active users physical I/Os	Peak active users physical I/Os
Look up savings account	2000	200 RR	600 RR
Look up checking account	2000	200 RR	600 RR
Transfer money to checking	1000, 1000	100 RR, 1000 RW	300 RR, 3000 RW
Configure new bill payee	1000, 1000	100 RR, 1000 RW	300 RR, 3000 RW
Submit payment	2000	200 RR	600 RR
Look up payment history	12000	1200 SR	3600 RR
Totals	20000 R, and 2000 W	2000 RR 2000 RW	6000 RR 6000 RW

As you can see in Table 5-6, to meet the peak workloads, you need to design an I/O subsystem to support 6000 random reads/sec and 6000 random writes/sec:

Physical I/Os The number of physical I/Os per second from the host perspective
RR Random Read I/Os
RW Random Write I/Os

To determine the appropriate configuration to support your unique workload, see Appendix A, “Performance management process” on page 631.

5.5 Understanding your workload type

To understand the workload, you need the performance data from the operating system, as well as from the disk system. Combined analysis of these two sets of performance data can give you the entire picture and help you understand your workload. Separate analysis might be not accurate. In this section, we describe various performance monitoring tools.

5.5.1 Monitoring the DS8000 workload

The following performance monitoring tools are available for the IBM System Storage DS8800 and DS8700:

- IBM DS8800 and DS8700 performance reporting capability

  With Version 7.6.20.*xxx* of the IBM DS8000 command-line interface (DSCLI), you can retrieve performance data from a DS8800 or DS8700 system and provide it on a window or to a file. Use **lsperfgrprpt** and **lsperfrescrpt**. These commands provide reports for the performance group or the specified resource. See Example 5-1.

Example 5-1 Output of the performance reporting command

```
dscli> lsperfgrprpt pg0
time                  grp               resrc          avIO avMB avresp pri avQ tgtQ %hlpT %dlyT %impt
===================================================================================================
```

```
2011-11-11/14:00:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 1421 51.746  6.250    0 88  0    0     0     0
2011-11-11/14:05:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 2851 103.322 15.695   0 21  0    0     0     0
2011-11-11/14:10:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 3957 108.900 4.649    0 38  0    0     0     0
2011-11-11/14:15:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  708 20.079  0.206    0 69  0    0     0     0
2011-11-11/14:20:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  372 12.604  10.739   0 32  0    0     0     0
2011-11-11/14:25:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  490 17.082  9.948    0 61  0    0     0     0
2011-11-11/14:30:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181   57 0.294   0.177    0 112 0    0     0     0
2011-11-11/14:35:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  257 9.832   31.029   0 27  0    0     0     0
2011-11-11/14:40:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  229 7.875   0.711    0 89  0    0     0     0
2011-11-11/14:45:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  187 7.668   35.725   0 19  0    0     0     0
2011-11-11/14:50:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181  796 25.098  0.521    0 87  0    0     0     0
2011-11-11/14:55:00 IBM.2107-75TV181/PG0 IBM.2107-75TV181 8313 421.517 2.684    0 15  0    0     0     0
```

It is also possible to get the performance data for the DA pair or the rank. See Example 5-2.

Example 5-2 Output of the performance data for 20 hours for rank 17 (output truncated)

```
dscli> lsperfrescrpt -start 20h r17
time                resrc avIO avMB  avresp %Hutl %hlpT %dlyT %impt
===================================================================
2011-11-11/09:15:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:20:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:25:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:30:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:35:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:40:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:45:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:50:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/09:55:00 R17      0 0.000 0.000  0     0     0     0
2011-11-11/10:00:00 R17      0 0.000 0.000  0     0     0     0
```

By default, statistics are shown for one hour. You can use settings that are specified in days, hours, and minutes.

- IBM Tivoli Storage Productivity Center for Disk

  IBM Tivoli Storage Productivity Center for Disk is the tool to monitor the workload on your DS8000 for a long period of time and collect historical data. This tool can also create reports and provide alerts. See 7.2.1, “Tivoli Storage Productivity Center overview” on page 237.

5.5.2 Monitoring the host workload

The following sections list the host-based performance measurements and reporting tools under the UNIX, Linux, Windows, IBM i, and z/OS environments.

Open Systems servers

We list the most common tools that are available on Open Systems servers to monitor the workload.

UNIX and Linux Open Systems servers

To get host information about I/O subsystems, processor activities, virtual memory, and the use of the physical memory, you can use the following common UNIX and Linux tools:

- iostat
- vmstat
- sar
- nmon

- topas
- filemon

These commands are standard tools that are available with most UNIX and UNIX-like (Linux) systems. We suggest using **iostat** for the data that you need to evaluate your host I/O levels. Specific monitoring tools are also available for AIX, Linux, Hewlett-Packard UNIX (HP-UX), and Oracle Solaris.

For more information, see Chapter 9, “Performance considerations for UNIX servers” on page 327 and Chapter 12, “Performance considerations for Linux” on page 445.

Intel Open Systems servers

Common Microsoft Windows Server 2008 and Windows Server 2003 monitoring tools include the Windows Performance Monitor (**perfmon**). Performance Monitor gives you the flexibility to customize the monitoring to capture various categories of Windows server system resources, including processor and memory. You can also monitor disk I/O through **perfmon**.

For more information, see Chapter 10, “Performance considerations for Microsoft Windows servers” on page 397.

IBM i environment

IBM i provides a vast selection of performance tools that can be used in performance-related cases with external storage. Several of the tools, such as *Collection services,* are integrated in the IBM i system. Other tools are a part of an IBM i licensed product. The management of many IBM i performance tools is integrated into IBM i web graphical user interface *IBM System Director Navigator for i*, or into the product *iDoctor*.

The IBM i tools, such as *Performance Explorer* and iDoctor, are used to analyze the hot data in IBM i and to size solid-state drives (SSDs) for this environment. Other tools, such as Job Watcher, are used mostly in solving performance problems, together with the tools for monitoring the DS8000.

For more information about the IBM i tools and their usage, see 13.4.1, “IBM i performance tools” on page 484.

System z environment

The z/OS systems have proven performance monitoring and management tools that are available to use for performance analysis. *Resource Measurement Facility* (RMF), a z/OS performance tool, collects performance data and reports it for the desired interval. It also provides cache reports. The cache reports are similar to the disk-to-cache and cache-to-disk reports that are available in the Tivoli Storage Productivity Center for Disk, except that the RMF cache reports are in text format. RMF collects the performance statistics of the DS8000 that are related to the link or port and also to the rank and extent pool. The REPORTS(ESS) parameter in the RMF report generator produces the reports that are related to those resources.

The RMF Spreadsheet Reporter is an easy way to create Microsoft Excel Charts based on RMF postprocessor reports. It is used to convert your RMF data to spreadsheet format and generate representative charts for all performance charts for all performance relevant areas.

For more information, see Chapter 14, “Performance considerations for System z servers” on page 499.

5.5.3 Modeling the workload and sizing the system

Workload modeling is used to predict the behavior of the system under the workload, to be able to identify the limits and potential bottlenecks, and to model the growth of the system and plan for the future.

IBM Specialists and IBM Business Partner specialists use the IBM Disk Magic tool for modeling the workload on the systems. Disk Magic can be used to help to plan the DS8000 hardware configuration. With Disk Magic, you model the DS8000 performance when migrating from another disk subsystem or when making changes to an existing DS8000 configuration and the I/O workload. Disk Magic is for use with both System z and Open Systems server workloads.

When running the DS8000 modeling, you start from one of these scenarios:

- An existing, non-DS8000 model, which you want to migrate to a DS8000
- An existing DS8000 workload
- Modeling a planned new workload, even if you do not have the workload currently running on any disk subsystem

You can model the following major DS8000 components by using Disk Magic:

- DS8000 model: DS8100, DS8300, DS8700, and DS8800
- Cache size for the DS8000
- Number, capacity, and speed of disk drive modules (DDMs)
- Number of arrays and RAID type
- Type and number of DS8000 host adapters (HAs)
- Type and number of channels
- Remote Copy option

When working with Disk Magic, always ensure that you input accurate and representative workload information, because Disk Magic results depend on the input data that you provide. Also, carefully estimate the future demand growth that you input to Disk Magic for modeling projections. The hardware configuration decisions are based on these estimates.

More information about using Disk Magic is in 6.1, “Disk Magic” on page 176.

Sizing the system for the workload

With the performance data collected or estimated, and the model of the data created, you can size the planned system. Systems are sized with application-specific tools that are provided by the application vendors. We list several tools:

- General storage sizing: Disk Magic and Capacity Magic
- MS Exchange sizing: MS Exchange sizing tool

  See more information at these websites:

  – http://technet.microsoft.com/en-us/library/bb124558.aspx
  – http://technet.microsoft.com/en-us/library/ff367907.aspx
- Oracle, DB2, SAP R/3: IBM Techline and specific tools

  The following tools are useful also:

  – http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1931
  – http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1887
  – https://www.ibm.com/partnerworld/wps/servlet/ContentHandler/LLIE-6LLS4T
  – http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS3743

Workload testing

There are various reasons for conducting I/O load tests. They all start with a hypothesis and have well defined performance requirements. The objective of the test is to determine if the hypothesis is true or false. For example, a hypothesis might be that you think that a DS8800 with 18 disk arrays and 128 GB of cache can support 10,000 IOPS with a 70/30/50 workload and the following response time requirements:

- Read response times: 95th percentile < 10 ms
- Write response times: 95th percentile < 5 ms

Follow these generic steps for testing:

1. Define the hypothesis.
2. Simulate the workload by using an artificial or actual workload.
3. Measure the workload.
4. Compare workload measurements with objectives.
5. If the results support your hypothesis, publish the results and make recommendations. Or, if the results do not support your hypothesis, determine why and make adjustments.

Microsoft Windows environment

The following examples of tests might be appropriate for a Windows environment:

- Pre-deployment hardware validation. Ensure that the operating system, multipathing, and host bus adapter (HBA) drivers are at the latest levels and supported. Prior to deployment of any solution and especially a complex solution, such as Microsoft cluster servers, ensure that the configuration is supported. Check the interoperability site:

  http://www.ibm.com/systems/support/storage/ssic/interoperability.wss

- Application-specific requirements. Often, we receive inquiries about the DS8000 and Microsoft Exchange. Use the following tools to test and simulate the workload of MS Exchange: MS Exchange Jetstress and MS Exchange Workload Generator. For more information, see the following links about these tools:
  - http://technet.microsoft.com/en-us/library/ff706601.aspx
  - http://www.microsoft.com/download/en/details.aspx?id=20322

The universal workload generator and benchmark tool is the Iometer (http://www.iometer.org). Iometer is both a workload generator (it performs I/O operations in order to stress the system) and a measurement tool (it examines and records the performance of its I/O operations and their effect on the system). It can be configured to emulate the disk or network I/O load of any program or benchmark, or it can be used to generate entirely synthetic I/O loads. It can generate and measure loads on single or multiple (networked) systems.

Iometer can be used for the following measurements and characterizations:

- Performance of disk and network controllers
- Bandwidth and latency capabilities of buses
- Network throughput to attached drives
- Shared bus performance
- System-level hard drive performance
- System-level network performance

Iometer provides the ability to configure these settings:

- Read/write ratios
- Sequential/random
- Arrival rate and queue depth
- Blocksize

- Number of concurrent streams

With these configuration settings, you can simulate and test most types of workloads. Specify the workload characteristics to reflect the workload in your environment.

UNIX and Linux environment

The UNIX and Linux **dd** command is a great tool to drive sequential read workloads or sequential write workloads against the DS8000.

In this section, we discuss how to perform these tasks:

- Determine the sequential read speed that an individual vpath (logical unit number (LUN)) can provide in your environment.
- Measure sequential read and write speeds for filesystems.

To test the sequential read speed of a rank, you can run the command:

```
time dd if=/dev/rvpath0 of=/dev/null bs=128k count=781
```

The rvpath0 is the character or raw device file for the LUN presented to the operating system by SDD. This command reads 100 MB off of rvpath0 and reports how long it takes in seconds. Take 100 MB and divide by the number of seconds that is reported to determine the MB/s read speed.

> **Linux:** For Linux systems, use the appropriate /dev/sd*X* device or /dev/mpath/mpath*n* device if you use Device-Mapper multipath.

Issue the following command and start the **nmon** monitor or **iostat -k 1** command in Linux:

```
dd if=/dev/rvpath0 of=/dev/null bs=128k
```

Your **nmon** monitor (the **e** option) reports that this previous command imposed a sustained 100 MB/s bandwidth with a blocksize=128k on vpath0. Notice the xfers/sec column; xfers/sec is IOPS. Now, if your **dd** command did not already error out because it reached the end of the disk, press Ctrl+C to stop the process. Now, **nmon** reports idle. Next, issue the following **dd** command with a 4 KB blocksize and put it in the background:

```
dd if=/dev/rvpath0 of=/dev/null bs=4k &
```

For this command, **nmon** reports a lower MB/s but a higher IOPS, which is the nature of I/O as a function of blocksize. Try your **dd** sequential read command with a bs=1024 and you see a high MB/s but a reduced IOPS.

The following commands perform sequential writes to your LUNs:

```
dd if=/dev/zero of=/dev/rvpath0 bs=128k
dd if=/dev/zero of=/dev/rvpath0 bs=1024k
time dd if=/dev/zero of=/dev/rvpath0 bs=128k count=781
```

Try different blocksizes, different raw vpath devices, and combinations of reads and writes. Run the commands against the block device (/dev/vpath0) and notice that blocksize does not affect performance.

Because the **dd** command generates a sequential workload, you still need to generate the random workload. You can use a free open-source tool, Vdbench.

Vdbench is a disk and tape I/O workload generator for verifying data integrity and measuring the performance of direct-attached and network-connected storage on Windows, AIX, Linux, Solaris, OS X, and HP-UX. It uses workload profiles as the inputs for the workload modeling and has its own reporting system. All output is presented in HTML files as reports and can be analyzed later. For more information, see the following link:

http://sourceforge.net/projects/vdbench/

6

Performance planning tools

In this chapter, we present Disk Magic and the Storage Tier Advisor Tool (STAT), which you can use for the DS8000 capacity and performance planning. Disk Magic is a disk performance and capacity modeling tool that is provided by IntelliMagic. It is available for both System z and Open Systems. The STAT is a powerful reporting tool that provides a graphical representation of performance data that is collected by Easy Tier.

6.1 Disk Magic

We describe *Disk Magic* and its basic usage in this chapter. We include examples that show the required input data, how the data is fed into the tool, and also show the output reports and information that Disk Magic provides.

The examples in this chapter demonstrate the steps that are required to model a storage system for certain workload requirements. The examples and screen captures might refer to previous versions of Disk Magic and previous IBM storage systems, such as ESS800 and DS8300. However, they still provide guidance about the general steps that are involved in this process.

> **Disk Magic:** Disk Magic is available for use by IBM Business Partners, IBM representatives, and users. Clients must contact their IBM representative to run Disk Magic tool studies when planning for their DS8000 hardware configurations.
>
> Disk Magic for Windows is a product of IntelliMagic B.V. Although we continue to refer to this product as *Disk Magic* in this book, the product that is available from IntelliMagic for clients is renamed to *IntelliMagic Direction* and might contain more features.
>
> We provide basic examples only for the use of Disk Magic in this book. The version of the product used in these examples might be outdated and contain fewer features than the currently offered version for clients. For more information, see the product documentation and guides. Or, for more information about the latest client version of this product, go to the IntelliMagic website:
>
> `http://www.intellimagic.net`

6.1.1 The need for performance planning and modeling tools

Trying to apply the results of a lab benchmark to another environment and making extrapolations and other adjustments for the differences (and there are a significant number) generally lead to a low-confidence result.

One of the problems with general rules is that they are based on assumptions about the workload, which are lost or never documented in the first place. For example, a general rule for I/O rates that applies to 4 KB transfers does not necessarily apply to 256 KB transfers. A particular rule applies if the workloads are the same and all the hardware components are the same, which they seldom are. Disk Magic overcomes this inherent lack of flexibility in rules by allowing the person who runs the model to specify the details of the workload and the details of the hardware. Disk Magic then computes the result in the response time and resource utilization of running that workload on that hardware.

Disk Magic models are often based on estimates of the workload. For example, what is the maximum I/O rate that the storage server sees? This I/O rate depends on identifying the historical maximum I/O rate (which can require a bit of searching) and then possibly applying adjustment factors to account for anticipated changes. The more you can substitute hard data or even reasonable estimates to replace assumptions and guesses, the more accurate the results are. In any event, a Disk Magic model is likely to be far more accurate than results obtained by adjusting benchmark results or by applying general rules.

Disk Magic is calibrated to match the results of lab runs documented in sales materials and white papers. You can view it as an encoding of the data obtained in benchmarks and reported in white papers.

When the Disk Magic model is run, it is important to size each component of the storage server for its peak usage period, usually a 15 or 30 minute interval. Using a longer period tends to average out the peaks and non-peaks, which does not give a true reading of the maximum demand.

Different components can peak at different times. For example, a processor-intensive online application might drive processor utilization to a peak while users are actively using the system. However, disk utilization might be at a peak when the files are backed up during off-hours. So, you might need to model multiple intervals to get a complete picture of your processing environment.

6.1.2 Overview and characteristics

Disk Magic is a Microsoft Windows based disk subsystem performance modeling tool. Disk Magic can be used to help to plan the DS8000 hardware configuration. With Disk Magic, you *model* the DS8000 performance when migrating from another disk subsystem or changing an existing DS8000 configuration and the I/O workload. You can use Disk Magic with both System z and Open Systems server workloads.

When running the DS8000 modeling, you start from one of these scenarios:

- An existing model that is not a DS8000 that you want to migrate to a DS8000. This system can be an IBM product, such as an IBM 3990-6, an IBM Enterprise Storage Server (ESS), or a disk subsystem (other than IBM). Because a DS8000 can have much greater storage and throughput capacity than other disk storage subsystems, with Disk Magic you can *merge* the workload from several existing disk subsystems into a single DS8000.
- An existing DS8000 workload.
- Modeling a planned new workload, even if you do not have the workload currently running on any disk subsystem. You need an estimate of the workload characteristics, such as disk capacity, I/O rate, and cache statistics, which provide an estimate of the DS8000 performance results. Use an estimate for rough planning purposes only.

6.1.3 Input data needed for Disk Magic study

To perform the Disk Magic study, we need to get information about the characteristics of the current workload. Depending on the environment where the current workload is running, there are different methods of collecting data.

z/OS environment

For each control unit to be modeled (current and proposed), we need the following information:

- Control unit type and model
- Cache size
- Nonvolatile storage (NVS) size
- Disk drive module (DDM) size and speed
- Number, type, and speed of channels
- Parallel access volume (PAV), and whether it is used

For Remote Copy, we need the following information:

- Remote Copy type
- Distance
- Number of links

In a z/OS environment, running a Disk Magic model requires the System Management Facilities (SMF) record types 70 through 78. The easiest way to send the SMF data to IBM is through FTP. To avoid huge dataset sizes, separate the SMF data by SYSID or by date. The SMF dataset needs to be tersed before putting it on the FTP site. Example 6-1 shows the instructions to terse the dataset.

Example 6-1 How to terse the SMF dataset

```
Use TRSMAIN with the PACK option to terse the SMF dataset. Do NOT run TRSMAIN against an
SMF dataset that resides on tape, which causes problems with the terse process of the
dataset. The SMF record length on tape can be greater than 32760 bytes, which TRSMAIN is
not able to handle properly.
```

Example 6-2 shows how to FTP the tersed dataset.

Example 6-2 FTP instructions

```
The FTP site is: testcase.boulder.ibm.com
userid = anonymous
password = your e-mail user id, for example: abc@defg.com
directory to put in: eserver/toibm/zseries
notify IBM with the file name you use to create your FTP file
```

Open Systems environment

In an Open Systems environment, we need the following information for each control unit to include in this study:

- Storage controller make, machine type, model, and serial number
- The number, size, and speed of the disk drives installed on each controller
- The number, speed, and type of channels
- The cache size
- Whether the control unit is direct-attached or SAN-attached
- How many servers are allocated and sharing these disks

For Remote Copy, we need the following information:

- Remote Copy type
- Distance
- Number of links

Data collection

The preferred data collection method for a Disk Magic study is by using Tivoli Storage Productivity Center. For each control unit to be modeled, collect performance data, create a report for each control unit, and export each report as a comma-separated values (CSV) file. You can obtain the detailed instructions for this data collection from the IBM representative.

Other data collection techniques

If Tivoli Storage Productivity Center is not available or cannot be used for the existing disk systems, other data collection techniques are available. Contact your IBM representative.

The Help function in Disk Magic documents shows how to gather various Open Systems types of performance data by using commands, such as **iostat** in Linux/UNIX and **perfmon** in Windows. Disk Magic also can process PT reports from IBM i systems.

6.1.4 Output information

Disk Magic models the DS8000 performance based on the I/O workload and the DS8000 hardware configuration. Thus, it helps in the DS8000 capacity planning and sizing decisions. The following major DS8000 components can be modeled by using Disk Magic:

- DS8000 model: DS8800 2-way and 4-way, DS8700 2-way and 4-way, DS8300, DS8300 logical partition (LPAR) and DS8300 Turbo, and DS8100 and DS8100 Turbo
- Cache size for the DS8000
- Number, capacity, and speed of DDMs
- Number of arrays and RAID type
- Type and number of the DS8000 host adapters (HAs)
- Type and number of channels
- Remote Copy option

When working with Disk Magic, always ensure that you feed in accurate and representative workload information, because Disk Magic results depend on the input data that is provided. Also, carefully estimate future demand growth, which is fed into Disk Magic for modeling projections on which the hardware configuration decisions are made.

6.1.5 Workload growth projection

The workload that runs on any disk subsystem always grows over time, which is why it is important to project how the new disk subsystem performs as the workload grows. There are two growth projection options:

- I/O rate

  The I/O rate growth projection projects the disk subsystem performance when the I/O rate grows. Use this option when the I/O rate is expected to grow but without a simultaneous growth of cache or the backstore capacity of the subsystem. You can expect the Access Density (number of I/O/sec/GB) to increase. In this particular case, Disk Magic keeps the hit rate the same for each step.
- I/O rate with capacity growth

  The I/O rate with capacity growth projection projects the disk subsystem performance when both the I/O rate and the capacity grow. With this selection, Disk Magic grows the workload and the backstore capacity at the same rate (the Access Density remains constant) while the cache size remains the same. Automatic Cache modeling is used to compute the negative effect on the cache hit ratio.

6.1.6 Disk Magic modeling

The process of modeling with Disk Magic starts with creating a *base model* for the existing disk subsystems. Initially, you load the input data that describes the hardware configuration and workload information of those disk subsystems. When you create the base model, Disk Magic validates the hardware and workload information that you entered, and if everything is acceptable, a valid base model is created. If not, Disk Magic provides messages that explain why the base model cannot be created, and it shows the errors on the log.

After the valid base model is created, you proceed with your modeling. You change the hardware configuration options of the base model to determine the best DS8000 configuration for a certain workload. Or, you can modify the workload values that you initially entered, so that, for example, you can see what happens when your workload grows or its characteristics change.

Welcome to Disk Magic

When we launch the Disk Magic program, the Welcome to Disk Magic panel opens, as shown in Figure 6-1. We can select to create a SAN project or open an existing project that is saved to a Disk Magic file with the extension DM2.

Figure 6-1 Welcome to Disk Magic

When we create a model for a DS8000 storage system, we select **New SAN Project** and the panel, as shown in Figure 6-2 on page 181, opens. This panel shows the following options:

- **CreAte New Project Using Automated Input**:
  - zSeries or WLE Automated input (*.DMC) by using DMC input files:

    Disk Magic Control (DMC) files contain a description of both the configuration and workload of a z/OS environment for a specific time interval. This description is used as a starting point for a detailed and accurate disk subsystem modeling study with Disk Magic that is based on data collected with the z/OS Resource Management Facility. Use the Resource Management Facility (RMF) Loader option to process raw RMF data created with RMFPack on z/OS to create DMC files for Disk Magic.
  - Open and iSeries® Automated Input (*.IOSTAT, *.TXT, *.CSV):

    With this option, you can make Disk Magic process multiple UNIX/Linux **iostat**, Windows **perfmon**, or iSeries PT reports, that are generated by multiple servers. Disk Magic then consolidates these statistics across the servers, so that you can identify the interval with the highest I/O rate, MB transferred rate, and so on.
  - Tivoli Productivity Center Reports from disk subsystem (DSS) and SAN Volume Controller (SVC) configurations (*.CSV):

    This option creates a .CSV output file from Tivoli Storage Productivity Center.
- **Create New Project Using Manual Input**:
  - General Project: This option can be selected to create a project that initially consists of a single Project, Model, System, and Disk Subsystem:
    - Number of zSeries Servers
    - Number of Open Servers
    - Number of iSeries Servers

- A Transaction Processing Facility (TPF) project (TPF storage subsystem modeler)
- Storage Virtualization Wizard Project: Select this option when you intend to build a model that consists of a virtualization cluster (for example, SAN Volume Controller) and attached disk subsystems.

- **Create DMC Files with RMF Loader**

  This option can be used to read z/OS RMF/SMF data collected at the mainframe of the client and packed with RMFPACK. RMF Loader then creates the command files to be later used in Disk Magic.

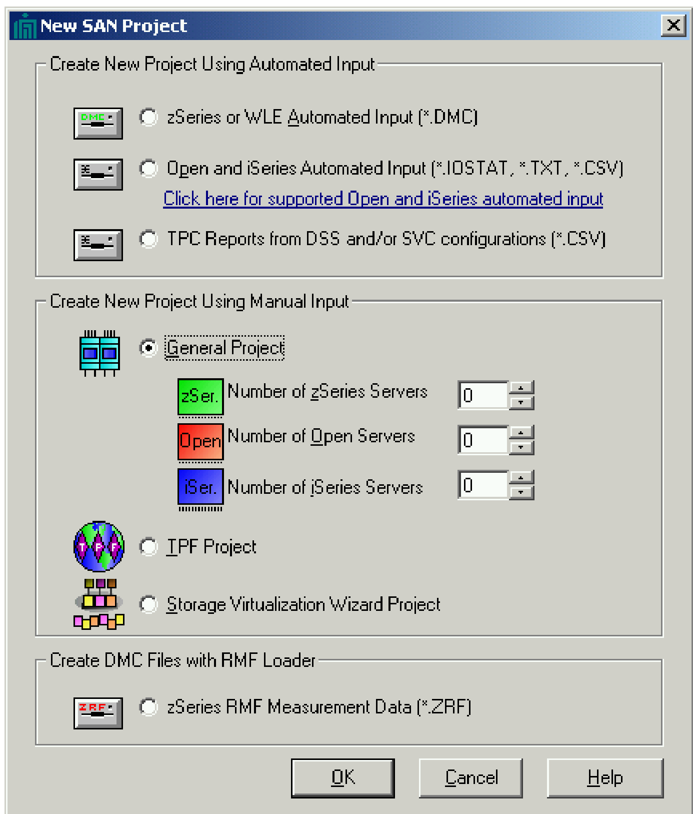


Figure 6-2 Creating a SAN project in Disk Magic

6.2 Disk Magic for System z

We explain how to use Disk Magic as a modeling tool for System z (still designated as zSeries in Disk Magic). In the example presented, we merge two ESS-800s into one DS8300. Although we typically consolidate DS8100 or DS8300 systems into one DS8800 system today, this example still demonstrates the basic steps generally involved in this process, which can easily be adapted to other storage models.

In a z/OS environment, disk subsystem measurement data is collected by the z/OS Resource Management Facility. For more information, see 14.11, “RMF” on page 522. The disk subsystem measurement data is stored in an SMF dataset. These zSeries I/O load statistics can be entered manually or automatically into Disk Magic.

Manual data entry requires you to format Device and Cache Activity reports with the RMF post processor, which creates reports with device-level statistics and a summary by logical control unit (LCU). The LCU-level data needs to be entered into Disk Magic. The format of the Disk Magic zSeries Workload page corresponds to the data items on these reports.

The preferred option is to use an automated input process to load the RMF data into Disk Magic by using a Disk Magic Control (DMC) file. DMC files contain a description of both the configuration and workload of a z/OS environment for a specific time interval. This DMC file is used as a starting point for a detailed and accurate disk subsystem modeling study with Disk Magic.

You can use the RMF Loader option in Disk Magic, as shown in Figure 6-2 on page 181, to process raw RMF data and create a Disk Magic Control (DMC) file.

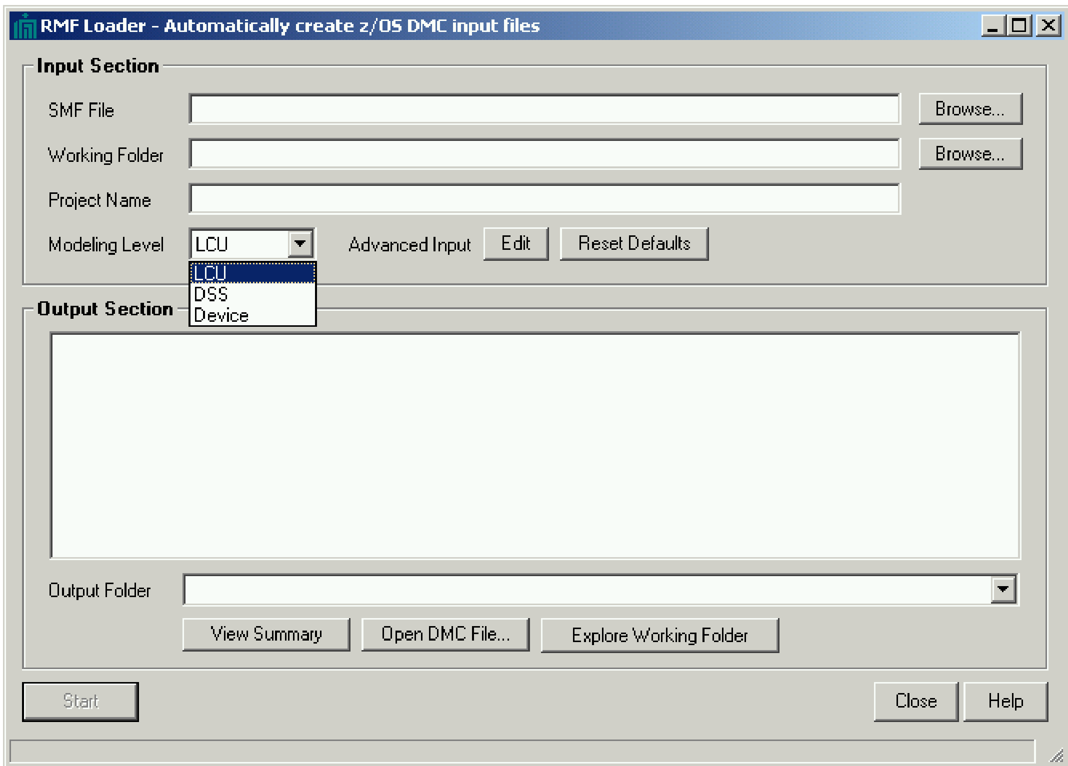


Figure 6-3 Using RMF Loader in Disk Magic to create z/OS DMC input files

To be able to process the z/OS SMF dataset on a Windows system with Disk Magic, it must be packed first with the RMFPack utility. RMFPack is part of the Disk Magic installation. The Disk Magic installation provides two XMIT files for installation of RMFPack on the z/OS system. The Disk Magic installation provides a PDF file that contains a detailed description of how to install and use RMFPack on z/OS. Use RMFPack to create the input data for the RMF Loader option. RMFPack creates an SMF file in ZRF format on z/OS to be downloaded in binary to your Windows system. You can then create your DMC files by processing the data with the RMF Loader option in Disk Magic, as shown in Figure 6-3, and determine the intervals to use for modeling.

To read the DMC file as automated input into Disk Magic, select **zSeries or WLE Automated input (*.DMC)** in the New SAN Project dialog panel, as shown in Figure 6-2 on page 181. By using automated input, you have the options to make the Disk Magic process the model at the disk subsystem (DSS), logical control unit (LCU), or device level (Figure 6-3). Considerations for using one or the other are provided in the Disk Magic help text under “How to Perform Device Level Modeling”.

We need the DMC file for the period for which we want to create the Disk Magic model. Typically, you model one of these periods:

- Peak I/O period
- Peak Read + Write throughput in MBps
- Peak Write throughput in MBps

6.2.1 Process the DMC file

DMC files typically use a *dmc* suffix for the file name and show the date and time of the corresponding RMF period by using the following naming convention:

- The first two characters represent an abbreviation of the month.
- The next two digits represent the date.
- The following four digits show the time period.

For example, `JL292059` means that the DMC file was created for the RMF period of July 29 at 20:59.

Follow these steps to process the DMC file:

1. From the panel in Figure 6-2 on page 181, first we select **zSeries or WLE Automated Input (*.DMC)** from the Create New Project Using Automated Input options. Figure 6-4 shows the window that opens and where you can select the DMC file that you want to use.

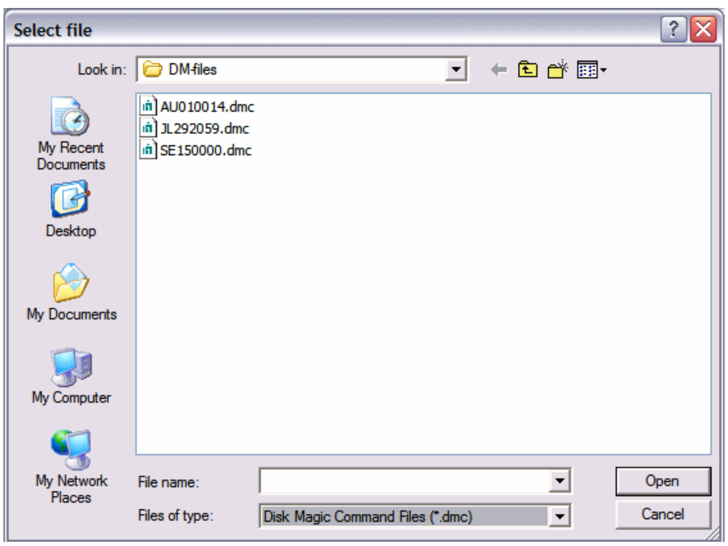


Figure 6-4 zSeries select DMC file

2. In this particular example, we select the **JL292059.dmc** file, which opens the following window, as shown in Figure 6-5 on page 184.

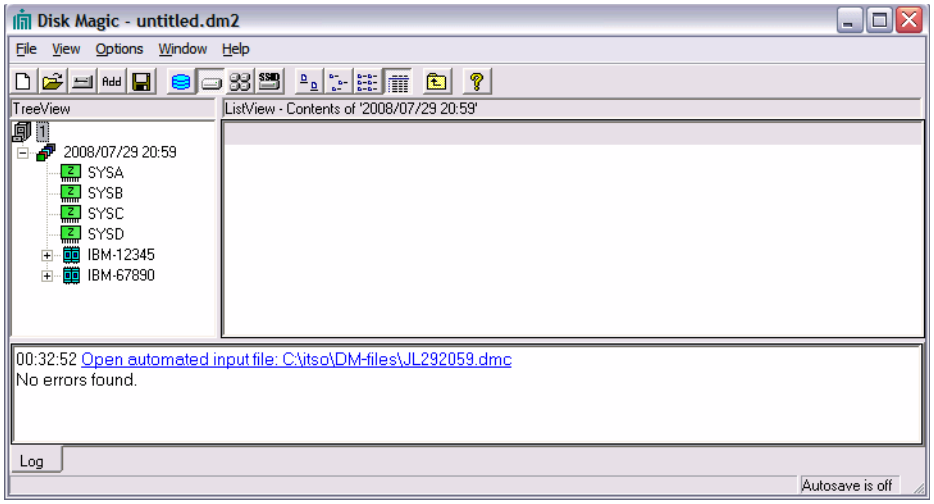


Figure 6-5 zSeries DMC file opened

3. We see that there are four LPARs (`SYSA`, `SYSB`, `SYSC`, and `SYSD`) and two disk subsystems (`IBM-12345` and `IBM-67890`).

   Clicking the IBM-12345 icon opens the general information that relates to this disk subsystem (Figure 6-6). It shows that this disk subsystem is an ESS-800 with 32 GB of cache that was created by using the subsystem identifier (SSID) or logical control unit (LCU) level. The number of subsystem identifiers (SSIDs) or LCUs is 12, as shown in the Number of zSeries LCUs field.

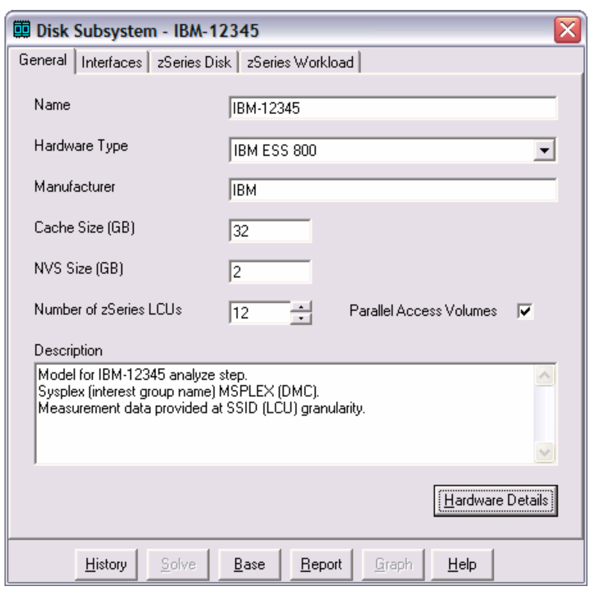


Figure 6-6 zSeries general information of the disk subsystem

4. Select **Hardware Details** on Figure 6-6 to open the window in Figure 6-7 on page 185. You can change the following features, based on the actual hardware configuration of the ESS-800:

   – SMP type
   – Number of host adapters
   – Number of device adapters
   – Cache size

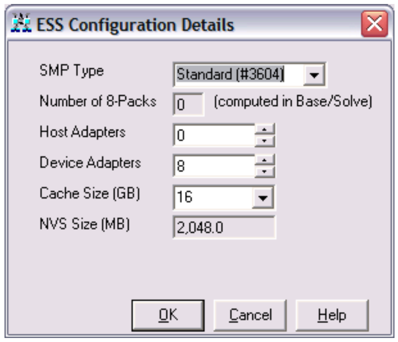


Figure 6-7 zSeries configuration details

5. Next, click the **Interfaces** tab, as shown in Figure 6-6 on page 184. We see that each LPAR connects to the disk subsystem through eight Fibre Channel connection (FICON) Express2 2 Gb channels. If this information is incorrect, you can change it by clicking **Edit**.

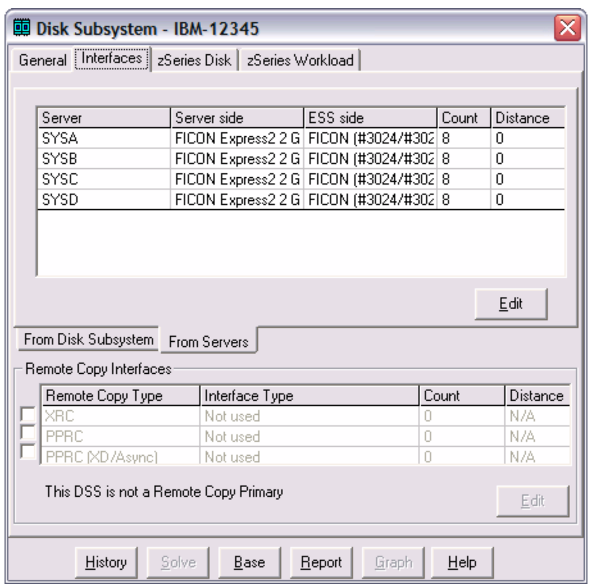


Figure 6-8 zSeries LPAR to disk subsystem interfaces

6. Select **From Disk Subsystem** in Figure 6-8 shows the interface that is used by the disk subsystem. Figure 6-9 on page 186 indicates that ESS IBM-12345 uses eight FICON ports.

   In this panel, you also indicate whether there is a Remote Copy relationship between this ESS-800 and a remote disk subsystem. You also can define the connections that are used between the primary site and the secondary site.

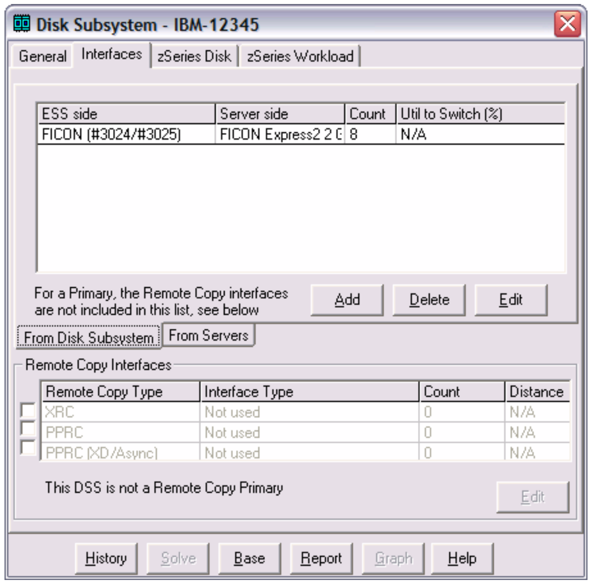


Figure 6-9 zSeries disk subsystem to LPAR interfaces

7. Next, look at the DDM by clicking the **zSeries Disk** tab. The DDM type is 36 GB/15K rpm. Because the DDM that we use is 73 GB/10K rpm, we update this information by clicking **Edit**. The 3390 types or models are 3390-3 and 3390-9 in Figure 6-10. Because any 3390 model that has a greater capacity than a 3390-9 model shows as a 3390-9 in the DMC file, we need to know the actual models of the 3390s. Generally, there is a mixture of 3390-9, 3390-27, and 3390-54 models.

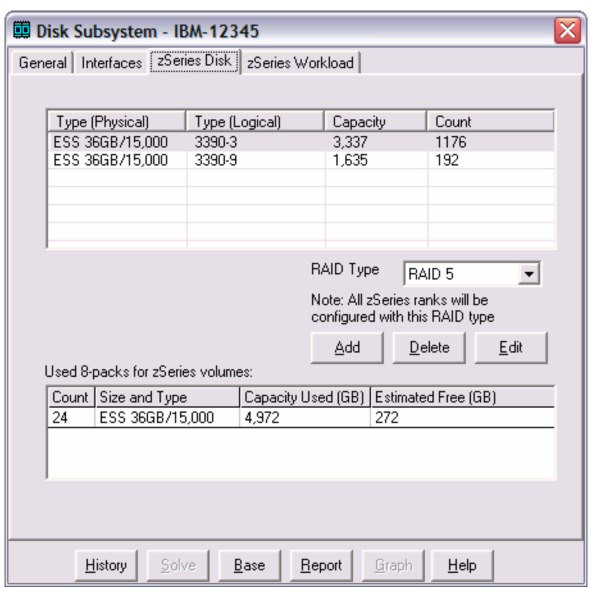


Figure 6-10 zSeries DDM option

8. To see the last option, select the **zSeries Workload** tab. Because this DMC file is created by using the SSID or LCU option, we see the I/O statistics for each LPAR by SSID (Figure 6-11 on page 187). Click the **Average** tab to the right of `SYSA` (the Average tab at the top in Figure 6-11 on page 187), scroll to the right of SSID `4010`, and click the **Average** tab (the Average tab at the bottom in Figure 6-12 on page 187). We see the total I/O rate from all four LPARs to this ESS-800, which is 9431.8 IOPS (Figure 6-12 on page 187).

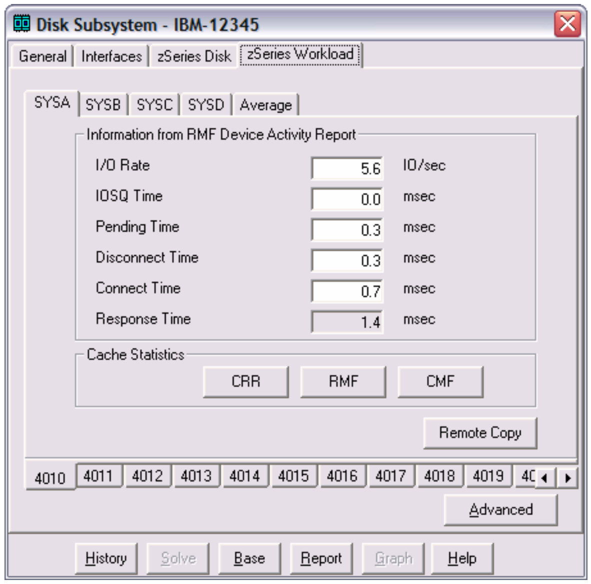


Figure 6-11 zSeries I/O statistics from SYSA on SSID 4010

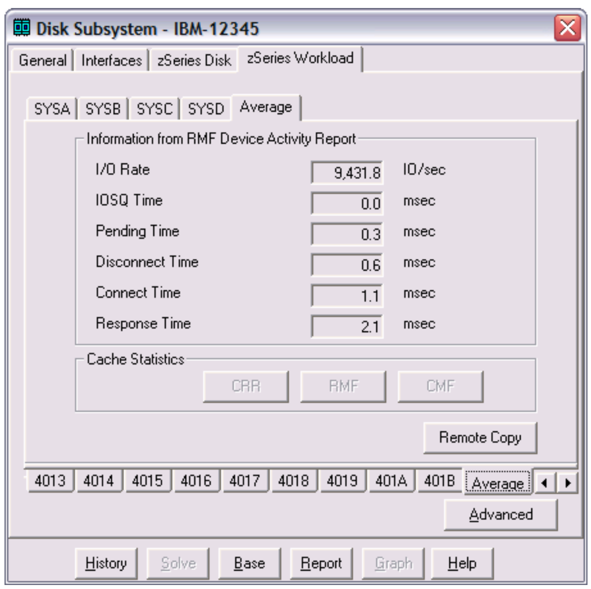


Figure 6-12 zSeries I/O statistics from all LPARs to this ESS-800

9. Click **Base** to create the base model for this ESS-800. It is possible that a base model cannot be created from the input workload statistics, for example, if there is excessive CONN time that Disk Magic cannot calibrate against the input workload statistics. In this case, we must identify another DMC from a different time period, and try to create the base model from that DMC file. After creating this base model for IBM-12345, we must also create the base model for IBM-67890 by following this same procedure.

6.2.2 zSeries model to merge the two ESS-800s to a DS8300

Now, we start the merge procedure to merge the two ESS-800s to a DS8300:

1. In Figure 6-13, right-click **IBM-12345** to open a new window, and click **Merge**, which opens yet another window, and select **Add to Merge Source Collection and create New Target**. This option creates the `Merge Target1`, which is the new disk subsystem that we use as the merge target (Figure 6-14).

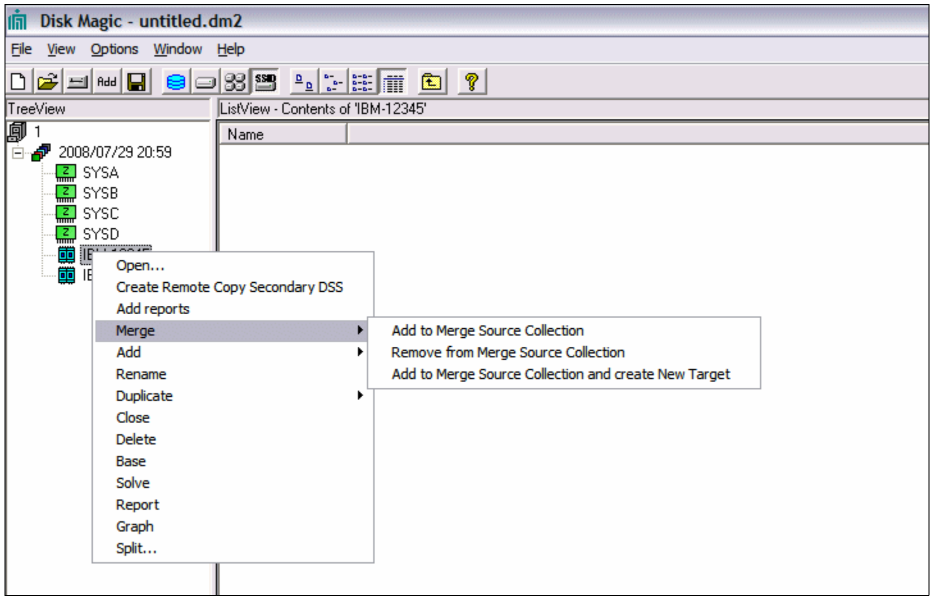


Figure 6-13 zSeries merge and create new target disk subsystem

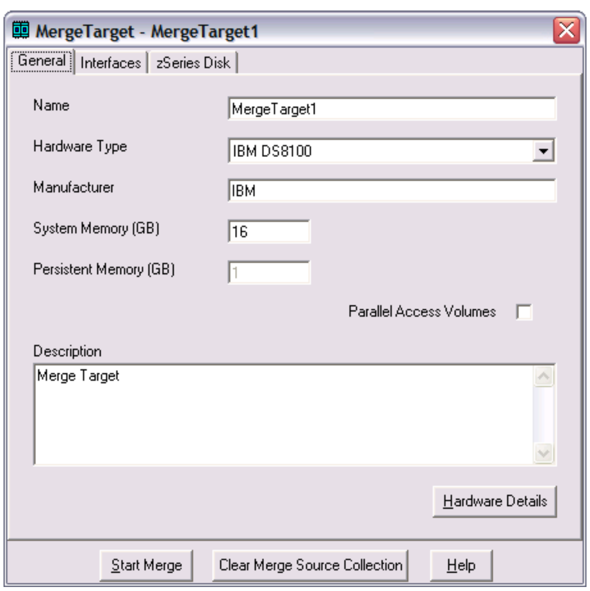


Figure 6-14 zSeries merge target disk subsystem

2. Because we want to merge the ESS-800s to a DS8300, we need to modify this Merge Target1. Click **IBM DS8100** on the Hardware Type option to open a window that presents choices, where we can select the IBM DS8300 Turbo. We also select **Parallel Access Volumes** so that Disk Magic can model the DS8300 to take advantage of this feature.

3. Select **Hardware Details** to open the window in Figure 6-15. With the Failover Mode option, you can model the performance of the DS8000 when one processor server with its associated processor storage is lost.

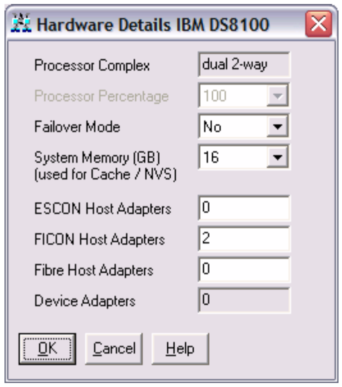


Figure 6-15 zSeries target hardware details option

We can select the cache size. In this case, we select 64 GB, because each of the two ESS-800s has 32 GB cache. In a DS8300, this selection automatically also determines the nonvolatile storage (NVS) size.

Disk Magic computes the number of HAs on the DS8000 based on the specification on the Interfaces page, but you can, to a certain extent, override these numbers. We suggest that you use one host adapter (HA) for every two ports, for both the Fibre Channel connection (FICON) ports and the Fibre ports. The Fibre ports are used for Peer-to-Peer Remote Copy (PPRC) links. We enter 4 FICON Host Adapters, because we are using eight FICON ports on the DS8300 (see the Count column in Figure 6-17 on page 190).

4. Click the **Interfaces** tab to open the From Servers dialog (Figure 6-16 on page 190). Because the DS8300 FICON ports are running at 4 Gbps, we need to update this option on all four LPARs and also on the From Disk Subsystem (Figure 6-17 on page 190) dialog. If the Host CEC uses different FICON channels than the FICON channels that are specified, it also needs to be updated.

   Select and determine the Remote Copy Interfaces. Select the Remote Copy type and the connections that are used for the Remote Copy links.

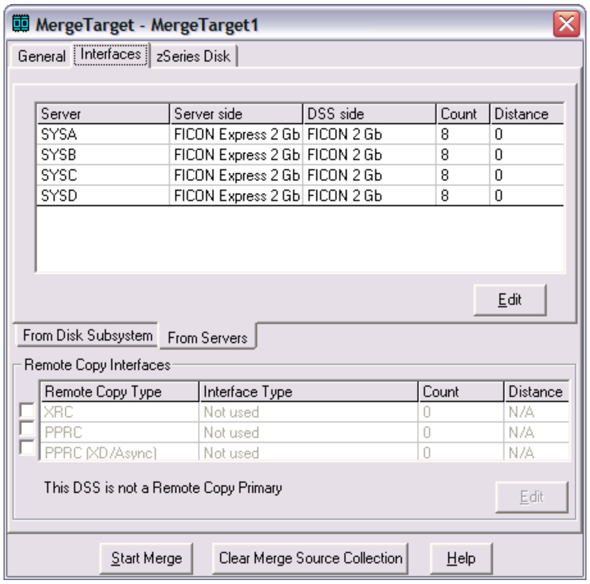


Figure 6-16 zSeries target interfaces from LPARs

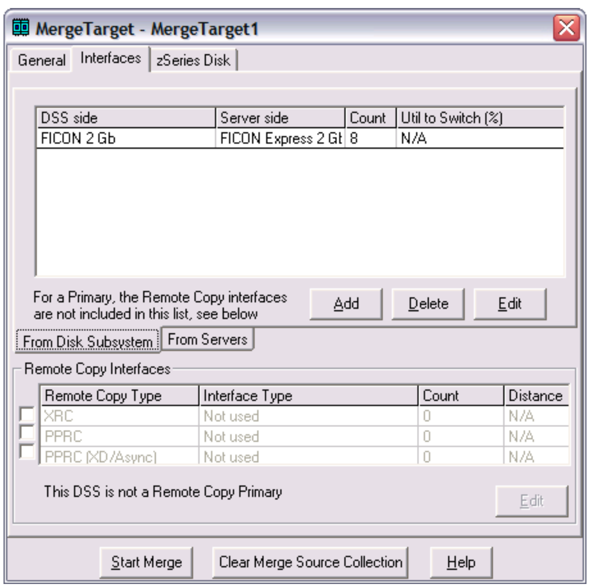


Figure 6-17 zSeries target interfaces from disk subsystem

5. To select the DDM capacity and rpm used, click the **zSeries Disk** tab in Figure 6-17. Now, you can select the DDM type that is used by clicking **Edit** in Figure 6-18 on page 191. In our example, we select the **DS8000 146GB/15k** DDM.

   Usually, you do not specify the number of volumes used. Disk Magic determines the number by adding all the 3390s coming from the merge source disk subsystems. If you know the configuration that is used as the target subsystem and want the workload to be spread over all the DDMs in that configuration, you can select the number of volumes on the target subsystem so that it reflects the number of configured ranks. You can also specify the RAID type that is used for this DDM set.

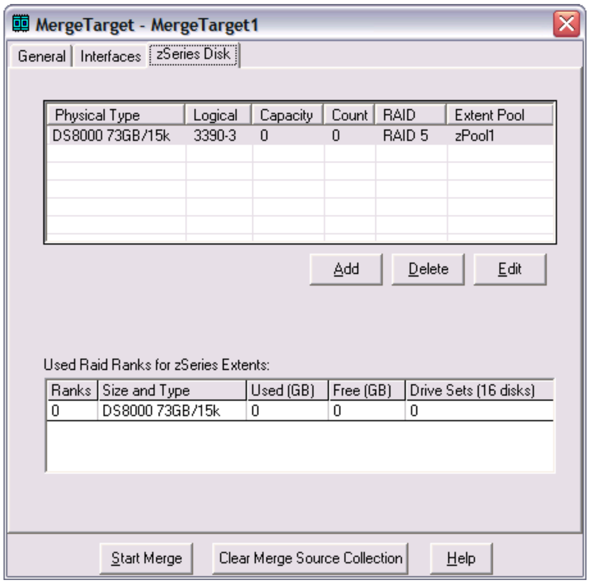


Figure 6-18 zSeries target DDM option

6. Merge the second ESS onto the target subsystem. In Figure 6-19, right-click **IBM-67890**, select **Merge**, and then, select **Add to Merge Source Collection**.

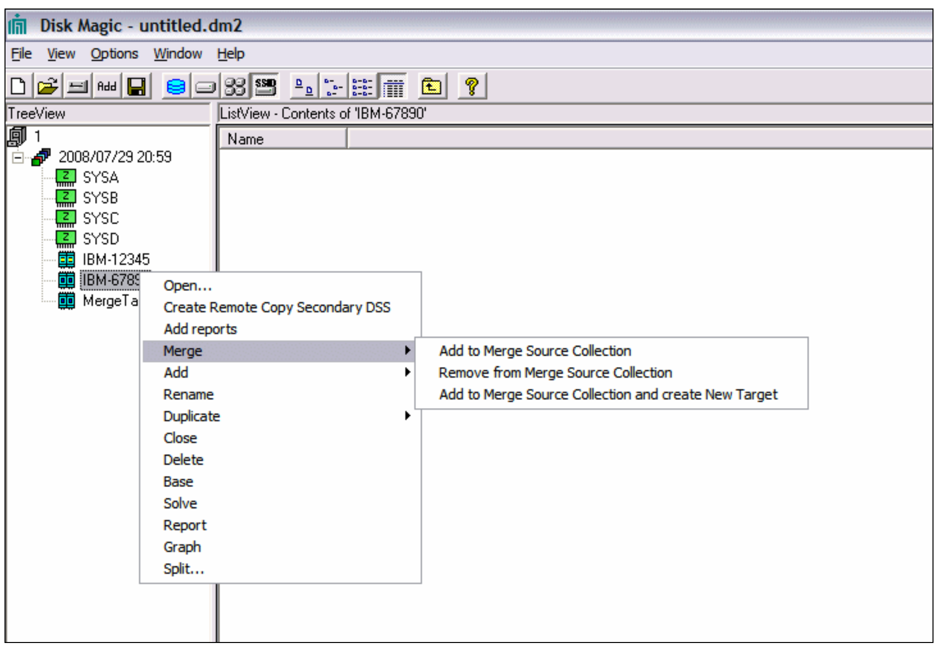


Figure 6-19 zSeries merge second ESS-800

7. Perform the merge procedure. From the Merge Target window (Figure 6-20 on page 192), click **Start Merge**.

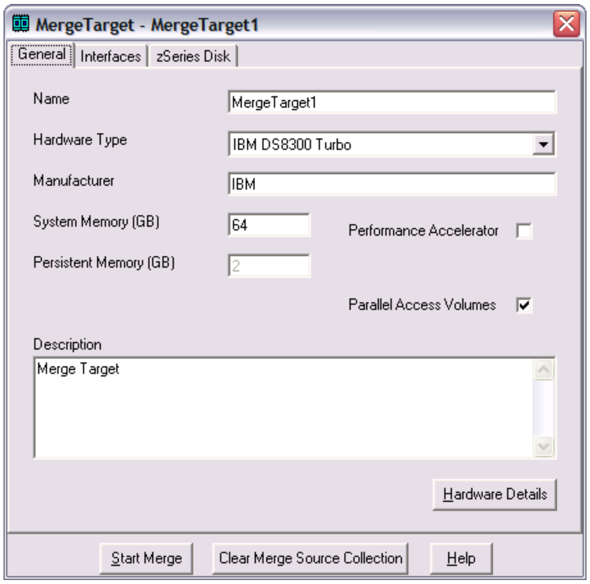


Figure 6-20 zSeries start the merge

8. This selection initiates Disk Magic to merge the two ESS-800s onto the new DS8300 and creates `Merge Result1` (Figure 6-21).

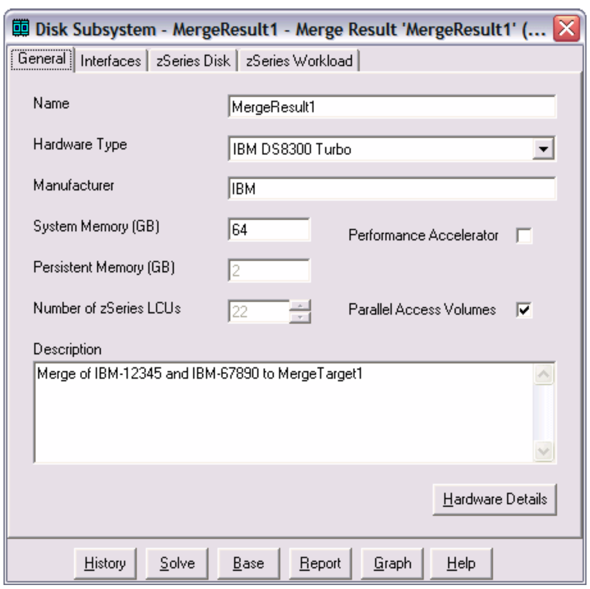


Figure 6-21 zSeries disk subsystem created as the merge result

9. To see the DDM configured for the DS8300, select **zSeries Disk** on MergeResult1. You can see the total capacity that is configured based on the total number of volumes on the two ESS-800s (Figure 6-22 on page 193). There are 11 ranks of 146 GB/15K rpm DDM required.

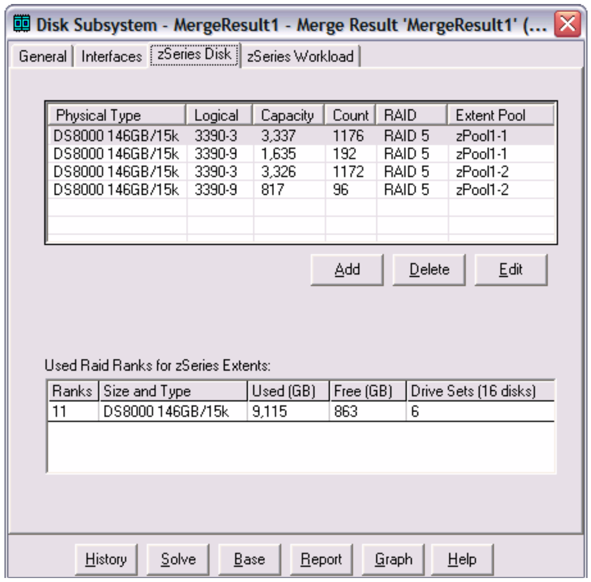


Figure 6-22 zSeries DDM of the new DS8300

10. Select **zSeries Workload** to show the Disk Magic predicted performance of the DS8300. You can see that the modeled DS8300 has an estimated response time of 1.1 msec. Disk Magic assumes that the workload is spread evenly among the ranks within the extent pool that is configured for the workload (Figure 6-23).

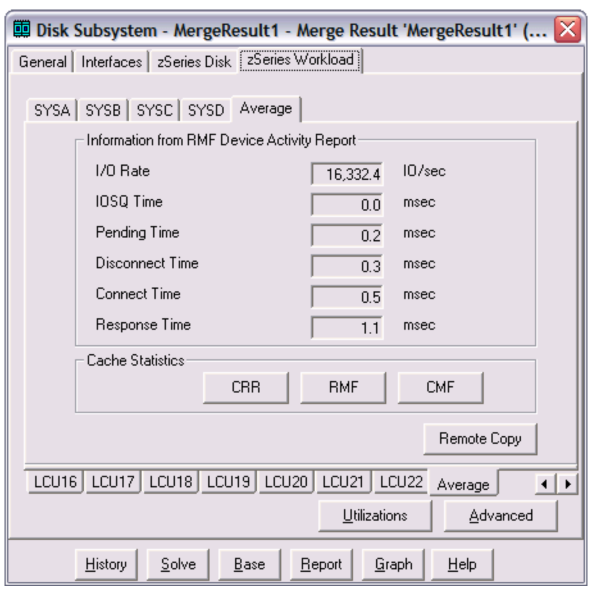


Figure 6-23 zSeries performance statistics of the DS8300

11. Click **Utilization** to display the utilization statistics of the various DS8300 components. In Figure 6-24 on page 194, you can see that the Average FICON HA Utilization is 39.8% and has a darker (amber) background color. This amber background is an indication that the utilization of that resource is approaching its limit. This percentage is still acceptable, but it is a warning that workload growth might push this resource to its limit.

    Any resource that is a bottleneck is shown with a red background. If a resource has a red background, you need to increase the size of that resource to resolve the bottleneck.

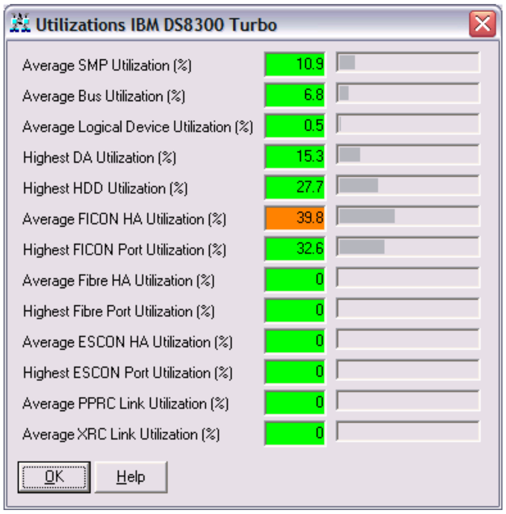


Figure 6-24 zSeries DS8300 utilization statistics

12. Figure 6-25 on page 195 can be used as a guideline for the various resources in a DS8000. The middle column has an amber background color, and the rightmost column has a red background color. The amber number indicates a warning that if the resource utilization reaches this number, an increase in the workload might soon cause the resource to reach its limit. The red numbers are the utilization numbers, which indicate that the resource is already saturated and can cause an increase in one of the components of the response time.
13. It is better if the merge result shows that none of the resource utilization falls into the amber color category.

Metric	Amber	Red
Average SMP Utilization (%)	60	80
Average Internal Bus Utilization (%)	70	90
Average Cross-Cluster Bus Utilization (%)		
Average Logical Device Utilization (%)		
- iSeries	30-40	50
- iSeries with Multipath	60-80	100
- zSeries, DSS or LCU level model	60	80
- zSeries with PAV - refer to ListView details	n/a	n/a
Highest DA Utilization (%)	60	80
Highest HDD Utilization (%)		
- iSeries	40	50
- zSeries	60	80
- Open	50	70
Average FICON HA Utilization (%)	35	50
Average FICON Port Utilization (%)	35	50
Average Fibre HA Utilization (%)	60	80
Average Fibre Port Utilization (%)	60	80
Average ESCON HA Utilization (%)	60	80
Average ESCON Port Utilization (%)	50	70
Average PPRC Link Utilization (%)	60	80
Average XRC Link Utilization (%)	60	80

Figure 6-25 Disk Magic utilization guideline

6.2.3 Disk Magic performance projection for the zSeries model

Based on the previous modeling results, we can create a chart that compares the performance of the original ESS-800s with the performance of the new DS8300:

1. Click **2008/07/29 20:59** to put the LPAR and disk subsystems in the right column (Figure 6-26).

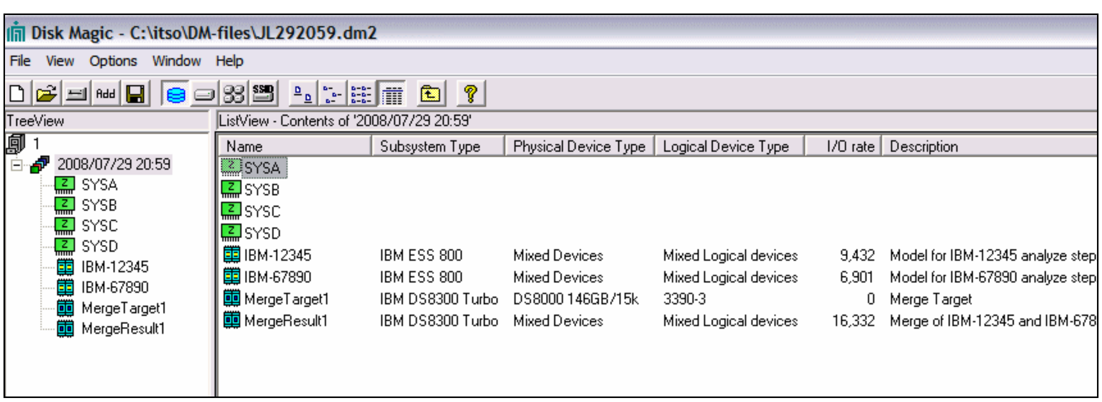


Figure 6-26 zSeries panel for performance comparison

2. Hold Ctrl down on the keyboard and select **IBM-12345**, **IBM-67890**, and **MergeResult1**. Right-click any of them and a small window opens. Select **Graph** from this window. In the panel that opens (Figure 6-27), select **Clear** to clear any prior graph option settings.

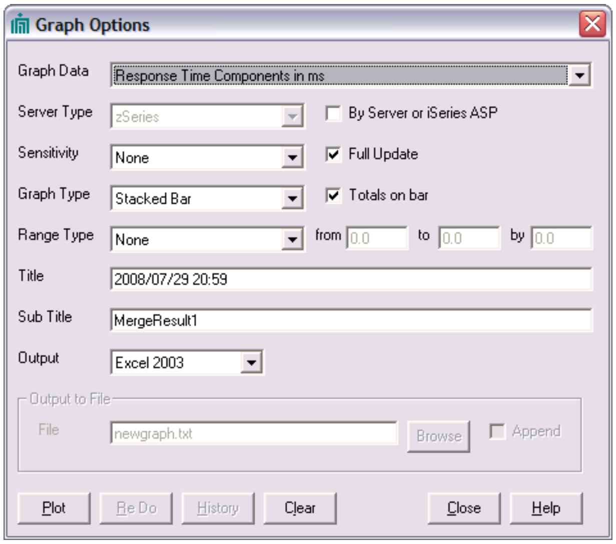


Figure 6-27 zSeries graph option panel

3. Click **Plot** to produce the response time components graph of the three disk subsystems that you selected in a Microsoft Excel spreadsheet. Figure 6-28 is the graph that is created based on the numbers from the Excel spreadsheet.

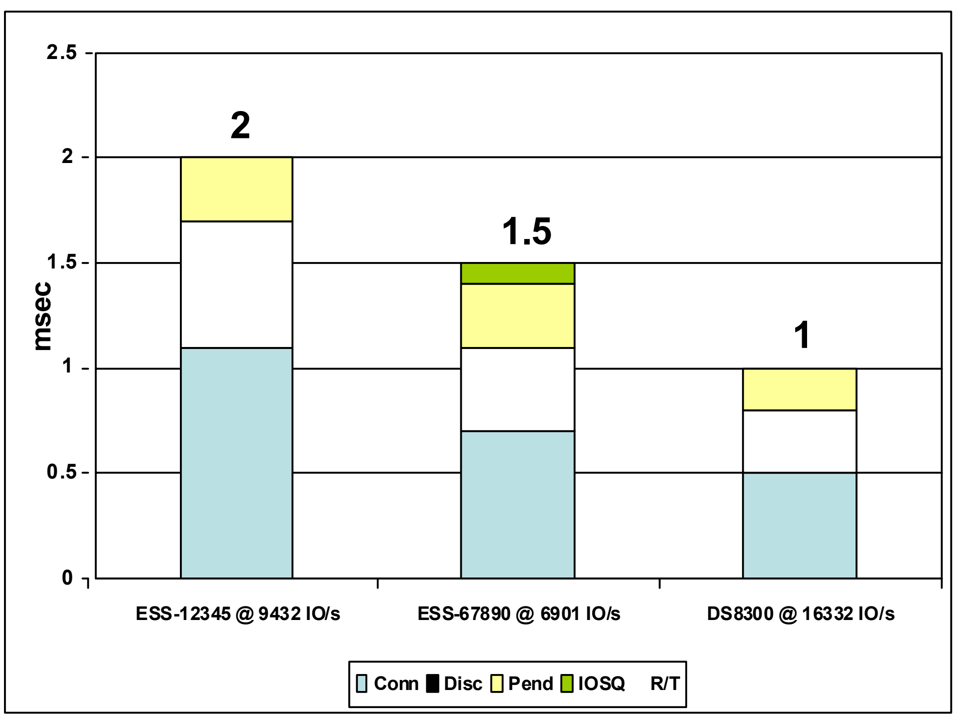


Figure 6-28 zSeries response time comparison

The DS8300 Response Time on the chart that is shown in Figure 6-28 on page 196 is 1.0 msec. The Disk Magic projected Response Time of the DS8300 in Figure 6-30 on page 198 is 1.1 msec. They differ, because Disk Magic rounds up to one decimal point on the performance statistics.

6.2.4 Workload growth projection for zSeries model

A useful feature of Disk Magic is the capability to create a workload growth projection and observe the impact of this growth on the various DS8000 resources and also on the response time. To run this workload growth projection, click **Graph** in Figure 6-23 on page 193 to open the panel that is shown in Figure 6-29. Click **Range Type**, and choose **I/O Rate**, which fills up the from field with the I/O Rate of the current workload, which is 16,332.4 I/O per second, the to field with 20,332.4 and the by field with 1,000. We can change these numbers. In our case, we change the numbers to `16000`, `50000`, and `2000`. Select **Clear** to clear up any graph option that was set up before.

Now, select **Plot**. An error message indicates a host adapter `Utilization > 100%`. This message means that you cannot increase the I/O rate up to 50000 I/O per second because of a FICON host adapter bottleneck. Click **OK** to complete the graph creation. The graph is based on the I/O rate increase as shown in Figure 6-30 on page 198.

Next, select **Utilization Overview** in the Graph Data choices, then click **Clear** and **Plot** to produce the chart shown in Figure 6-31 on page 198.

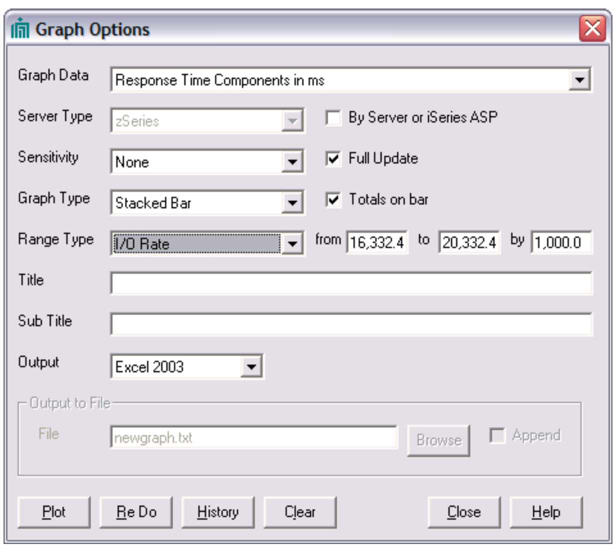


Figure 6-29 zSeries workload growth projection dialog

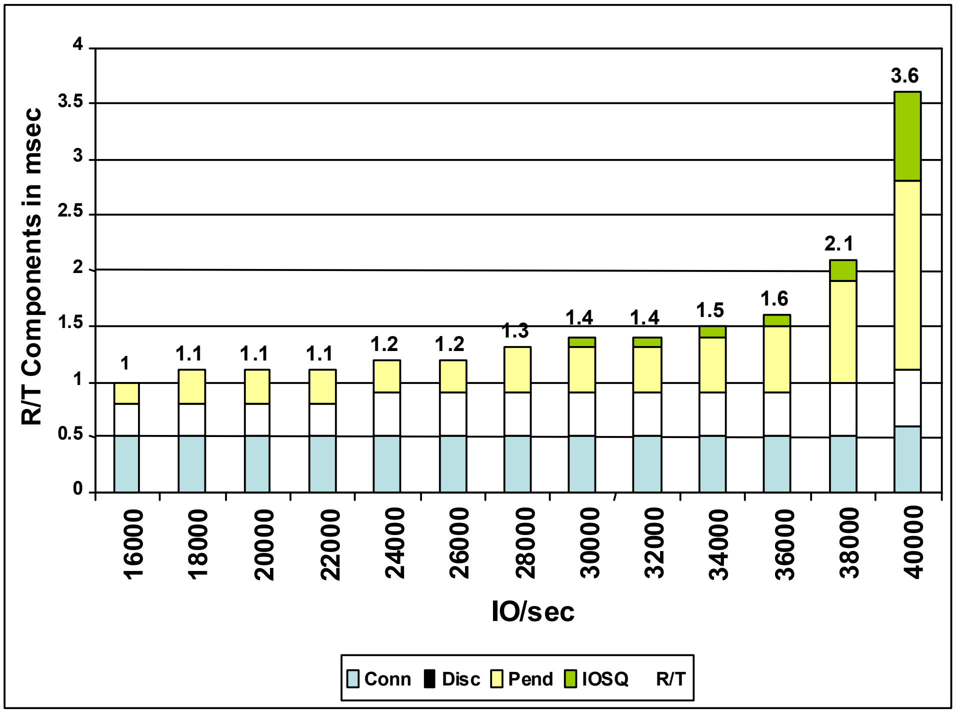


Figure 6-30 zSeries workload growth impact on response time components

In Figure 6-31, observe that the FICON host adapter started to reach the red area at 22000 I/O per second. The workload growth projection stops at 40000 I/O per second, because the FICON host adapter reaches 100% utilization when the I/O rate is greater than 40000 I/O per second.

			Total I/O Rate (I/Os per second)												
Utilizations	Amber Threshold	Red Threshold	16000	18000	20000	22000	24000	26000	28000	30000	32000	34000	36000	38000	40000
Average SMP	60%	80%	10.7%	12.0%	13.4%	14.7%	16.0%	17.4%	18.7%	20.1%	21.4%	22.7%	24.1%	25.4%	26.7%
Average Bus	70%	90%	6.7%	7.5%	8.4%	9.2%	10.0%	10.9%	11.7%	12.5%	13.4%	14.2%	15.0%	15.9%	16.7%
Average Logical Device	n/a	n/a	0.5%	0.6%	0.7%	0.7%	0.8%	0.9%	1.0%	1.1%	1.2%	1.3%	1.4%	1.5%	1.6%
Highest DA	60%	80%	14.9%	16.8%	18.7%	20.5%	22.4%	24.3%	26.1%	28.0%	29.9%	31.8%	33.6%	35.5%	37.4%
Highest HDD	60%	80%	27.2%	30.6%	34.0%	37.3%	40.7%	44.1%	47.5%	50.9%	54.3%	57.7%	61.1%	64.5%	67.9%
Average FICON HA	35%	50%	39.0%	43.9%	48.8%	53.7%	58.5%	63.4%	68.3%	73.2%	78.0%	82.9%	87.8%	92.7%	97.6%
Highest FICON Port	35%	50%	31.9%	35.9%	39.9%	43.9%	47.9%	51.9%	55.9%	59.9%	63.9%	67.9%	71.9%	75.8%	79.8%

Figure 6-31 zSeries workload growth impact on resource utilization

6.3 Disk Magic for Open Systems

In this section, we show how to use Disk Magic as a modeling tool for Open Systems. We illustrate the example of merging two ESS-800s into one DS8300 system. Although we typically consolidate DS8100 or DS8300 systems into one DS8800 system today, this example still demonstrates the basic steps generally involved in this process, which can easily be adapted to other storage models.

In this example, we use the Tivoli Storage Productivity Center comma-separated values (CSV) output file for the period that we want to model with Disk Magic. Typically, clients model these periods:

- Peak I/O period
- Peak Read + Write throughput in MBps
- Peak Write throughput in MBps

Tivoli Storage Productivity Center for Disk creates the Tivoli Storage Productivity Center csv output files.

6.3.1 Process the Tivoli Storage Productivity Center CSV output file

We use these steps to process the CSV files:

1. From Figure 6-1 on page 180, we select **Open and iSeries Automated Input** and click **OK**. This selection opens a window where you can select the csv files to use. In this case, we select **ESS11_TPC.csv**, and then, while holding Ctrl down, we select **ESS14_TPC.csv**. Then, click **Open** as shown in Figure 6-32.

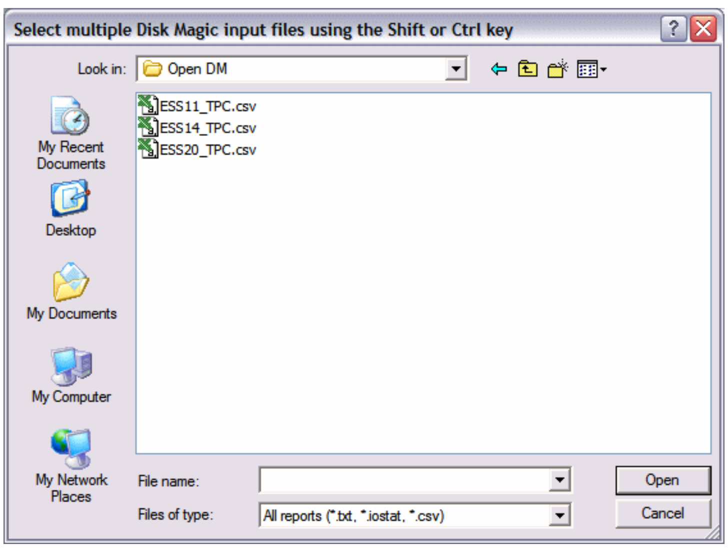


Figure 6-32 Open Systems select input file

2. The result is shown in Figure 6-33 on page 200. To include both csv files, click **Select All** and then click **Process** to display the I/O Load Summary by Interval table (see Figure 6-37 on page 202). This table shows the combined load of both ESS11 and ESS14 for all the intervals recorded in the Tivoli Storage Productivity Center CSV file.

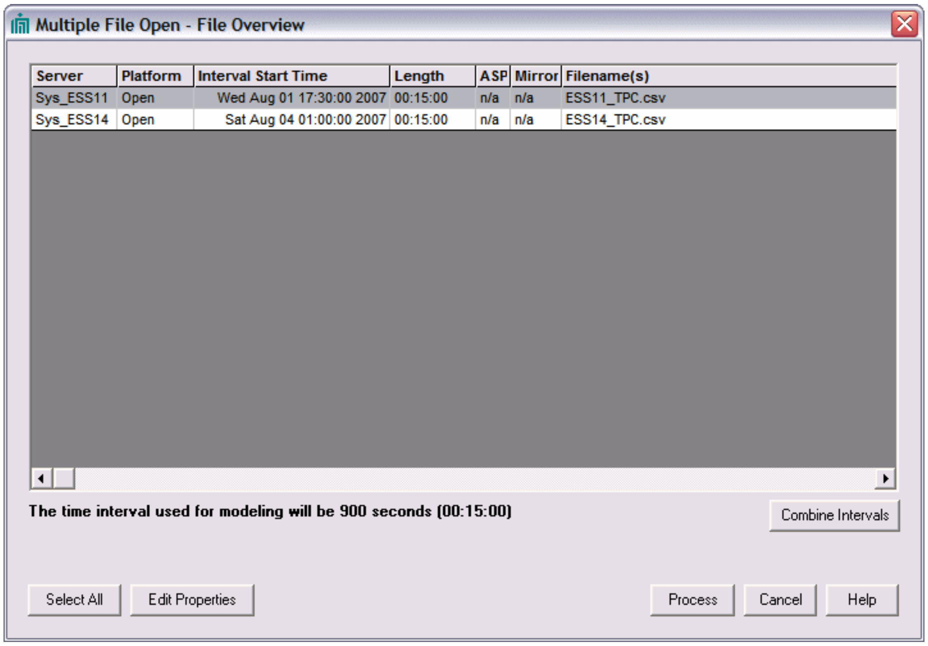


Figure 6-33 Open Systems Multiple File Open panel

3. Select **Excel** in Figure 6-37 on page 202 to create a spreadsheet with graphs for the I/O rate (Figure 6-34), Total MBps (Figure 6-35 on page 201), and Write MBps (Figure 6-36 on page 201) for the combined workload on both of the ESS-800s by time interval.

   Figure 6-34 shows the I/O Rate graph and that the peak is approximately 18000+ IOPS.

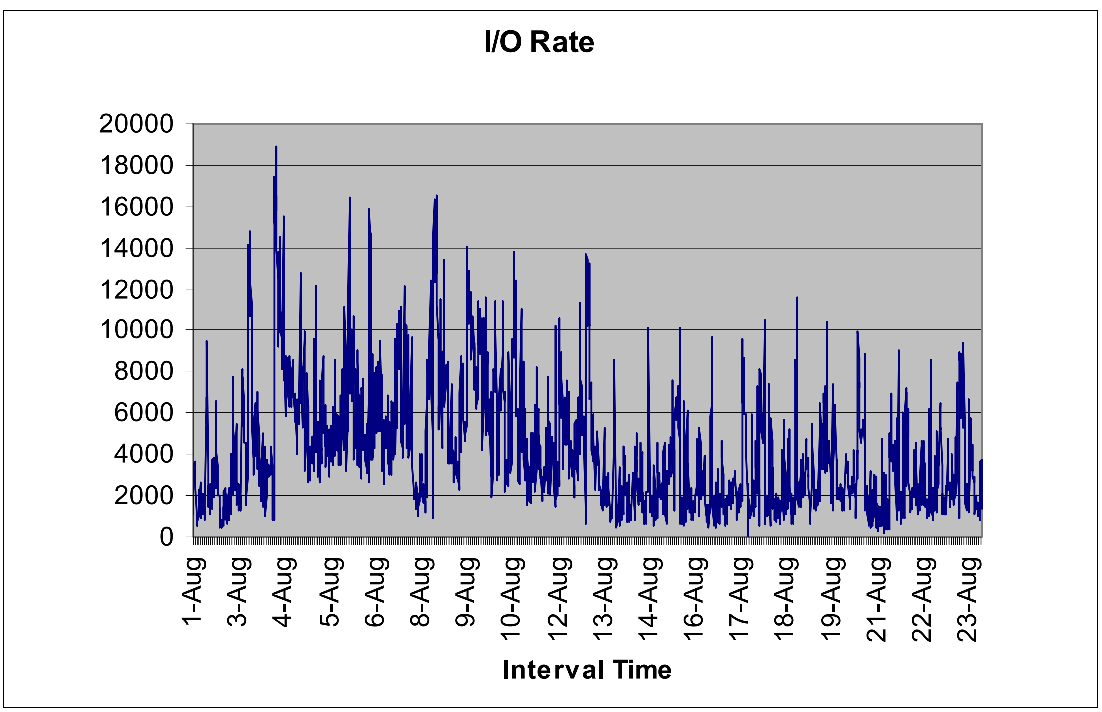


Figure 6-34 Open Systems I/O rate by time interval

Figure 6-35 shows the total MBps graph and that the peak is approximately 4700 MBps. This graph shows that this peak looks out of line compared to the other total MBps numbers from the other periods. Before using this interval to model the peak total MBps, investigate to learn whether this peak is real or if something unusual happened during this period that might cause this anomaly.

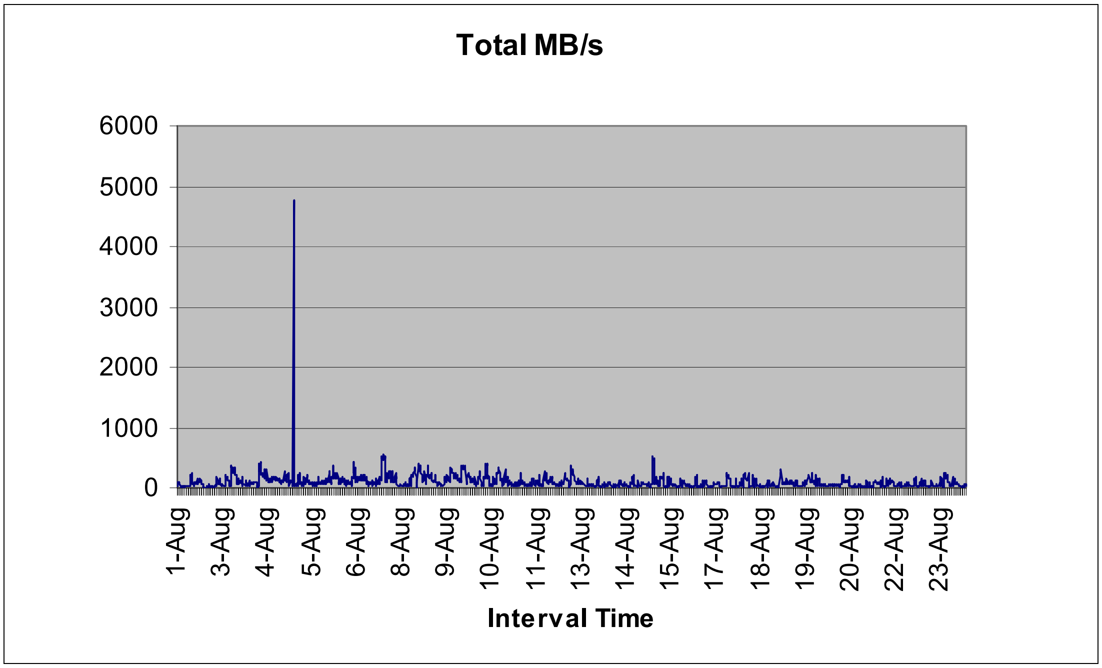


Figure 6-35 Open Systems total MBps by time interval

Also, investigate the situation for the peak write MBps on Figure 6-36. The peak period, as expected, coincides with the peak period of the total MBps.

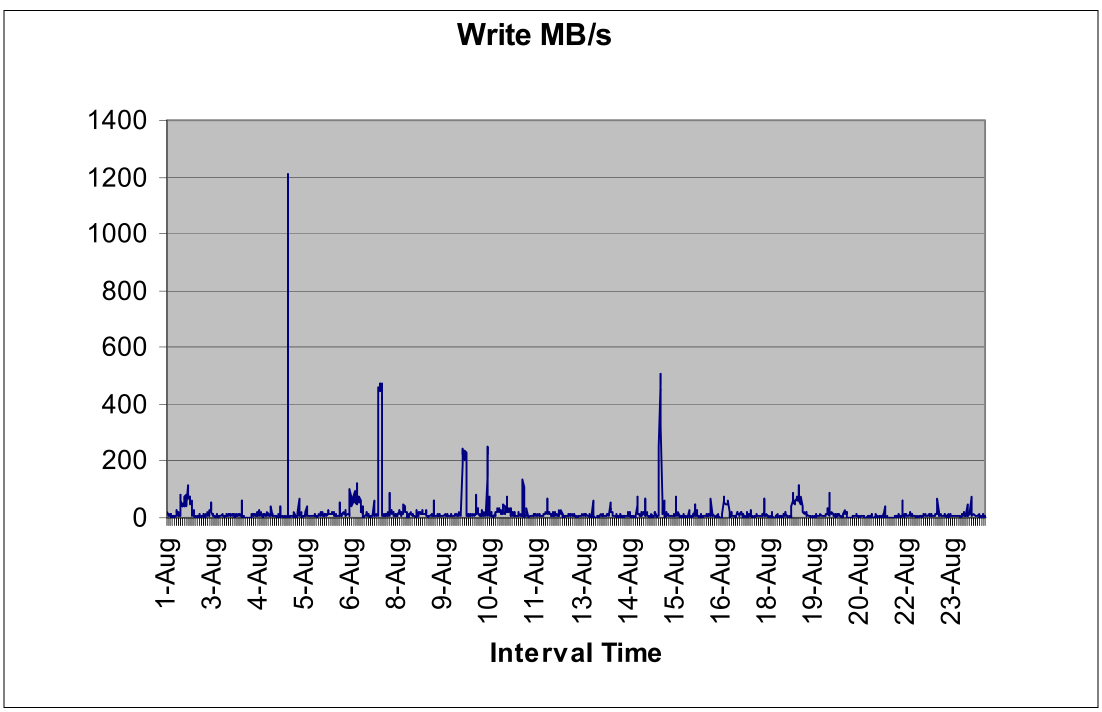
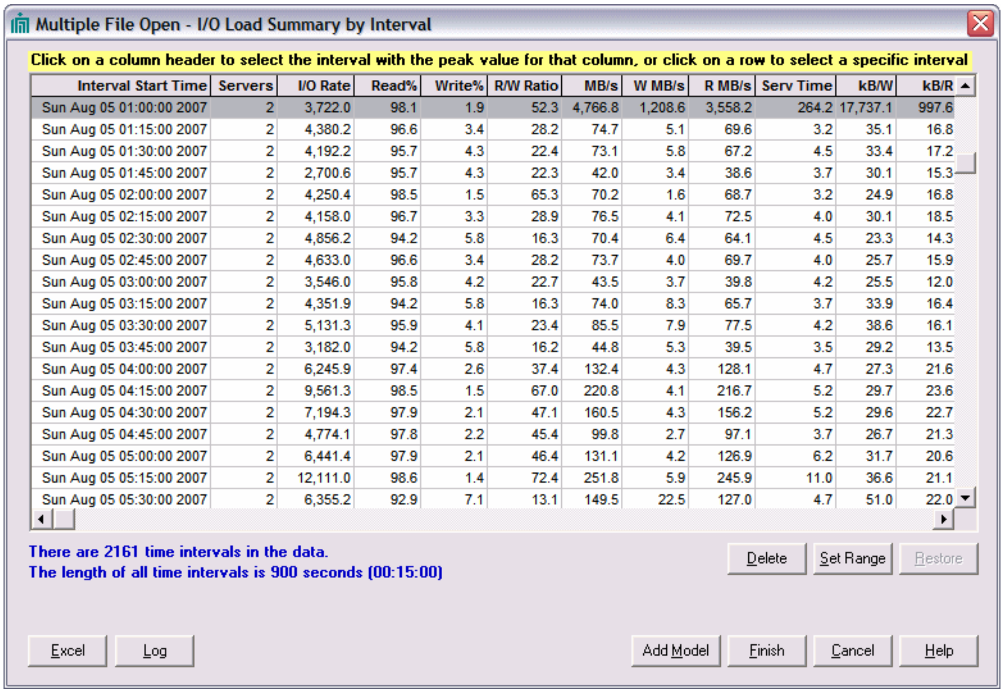


Figure 6-36 Open Systems write MBps by time interval

4. Click the **I/O Rate** column header in Figure 6-37 to highlight the peak I/O rate for the combined ESS11 and ESS14 disk subsystems (Figure 6-37).



Interval Start Time	Servers	I/O Rate	Read%	Write%	R/W Ratio	MB/s	W MB/s	R MB/s	Serv Time	kB/W	kB/R
Sun Aug 05 01:00:00 2007	2	3,722.0	98.1	1.9	52.3	4,766.8	1,208.6	3,558.2	264.2	17,737.1	997.6
Sun Aug 05 01:15:00 2007	2	4,380.2	96.6	3.4	28.2	74.7	5.1	69.6	3.2	35.1	16.8
Sun Aug 05 01:30:00 2007	2	4,192.2	95.7	4.3	22.4	73.1	5.8	67.2	4.5	33.4	17.2
Sun Aug 05 01:45:00 2007	2	2,700.6	95.7	4.3	22.3	42.0	3.4	38.6	3.7	30.1	15.3
Sun Aug 05 02:00:00 2007	2	4,250.4	98.5	1.5	65.3	70.2	1.6	68.7	3.2	24.9	16.8
Sun Aug 05 02:15:00 2007	2	4,158.0	96.7	3.3	28.9	76.5	4.1	72.5	4.0	30.1	18.5
Sun Aug 05 02:30:00 2007	2	4,856.2	94.2	5.8	16.3	70.4	6.4	64.1	4.5	23.3	14.3
Sun Aug 05 02:45:00 2007	2	4,633.0	96.6	3.4	28.2	73.7	4.0	69.7	4.0	25.7	15.9
Sun Aug 05 03:00:00 2007	2	3,546.0	95.8	4.2	22.7	43.5	3.7	39.8	4.2	25.5	12.0
Sun Aug 05 03:15:00 2007	2	4,351.9	94.2	5.8	16.3	74.0	8.3	65.7	3.7	33.9	16.4
Sun Aug 05 03:30:00 2007	2	5,131.3	95.9	4.1	23.4	85.5	7.9	77.5	4.2	38.6	16.1
Sun Aug 05 03:45:00 2007	2	3,182.0	94.2	5.8	16.2	44.8	5.3	39.5	3.5	29.2	13.5
Sun Aug 05 04:00:00 2007	2	6,245.9	97.4	2.6	37.4	132.4	4.3	128.1	4.7	27.3	21.6
Sun Aug 05 04:15:00 2007	2	9,561.3	98.5	1.5	67.0	220.8	4.1	216.7	5.2	29.7	23.6
Sun Aug 05 04:30:00 2007	2	7,194.3	97.9	2.1	47.1	160.5	4.3	156.2	5.2	29.6	22.7
Sun Aug 05 04:45:00 2007	2	4,774.1	97.8	2.2	45.4	99.8	2.7	97.1	3.7	26.7	21.3
Sun Aug 05 05:00:00 2007	2	6,441.4	97.9	2.1	46.4	131.1	4.2	126.9	6.2	31.7	20.6
Sun Aug 05 05:15:00 2007	2	12,111.0	98.6	1.4	72.4	251.8	5.9	245.9	11.0	36.6	21.1
Sun Aug 05 05:30:00 2007	2	6,355.2	92.9	7.1	13.1	149.5	22.5	127.0	4.7	51.0	22.0

Figure 6-37 Open Systems I/O Load Summary by Interval

5. From this panel, select **Add Model** and then select **Finish**. A pop-up window prompts you with “`Did you add a Model for all the intervals you need?`” because you can include multiple workload intervals in the model. However, we model one workload interval, so we respond **Yes**. The window in Figure 6-38 opens.

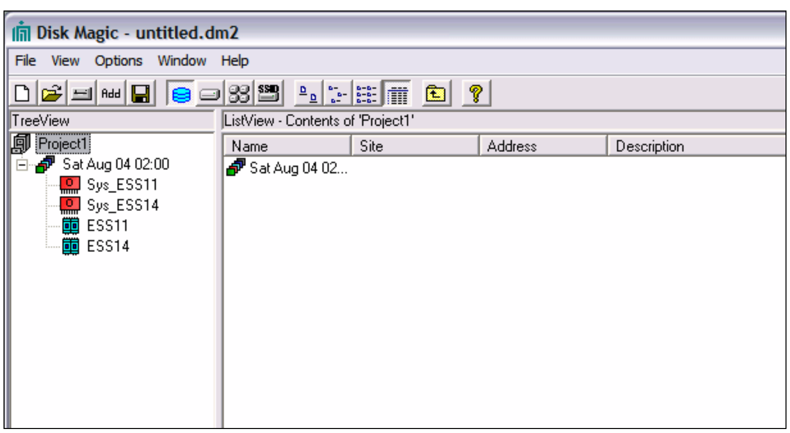


Figure 6-38 Open Systems disk subsystems

6. In Figure 6-38, double-click **ESS11** to see the general information related to this disk subsystem as shown in Figure 6-39 on page 203. Figure 6-39 on page 203 shows that this disk subsystem is an ESS-800 with 16 GB of cache and 2 GB of NVS.

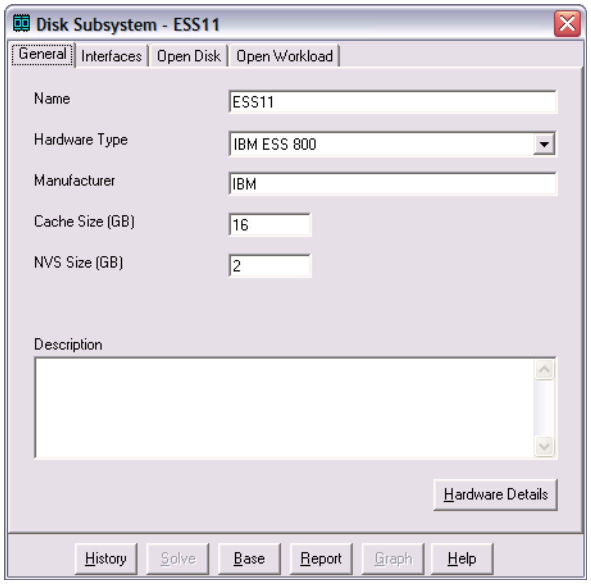


Figure 6-39 Open Systems general information about the ESS11 disk subsystem

7. Select **Hardware Details** in Figure 6-39 to change the following features, based on the actual hardware configuration of the ESS-800:
   - SMP type
   - Number of host adapters
   - Number of device adapters
   - Cache size

   In this example (Figure 6-40), we change the cache size to the actual cache size of the ESS11, which is 32 GB.

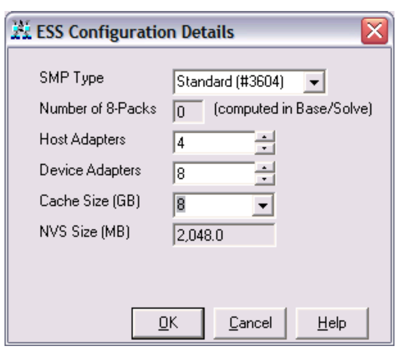


Figure 6-40 Open Systems configuration details

8. Next, click the **Interface** tab in Figure 6-39. The From Servers panel (Figure 6-41 on page 204) shows that each server connects to the disk subsystem through four 2 Gb Fibre Channels. If this information is not correct, you can change it by clicking **Edit**.

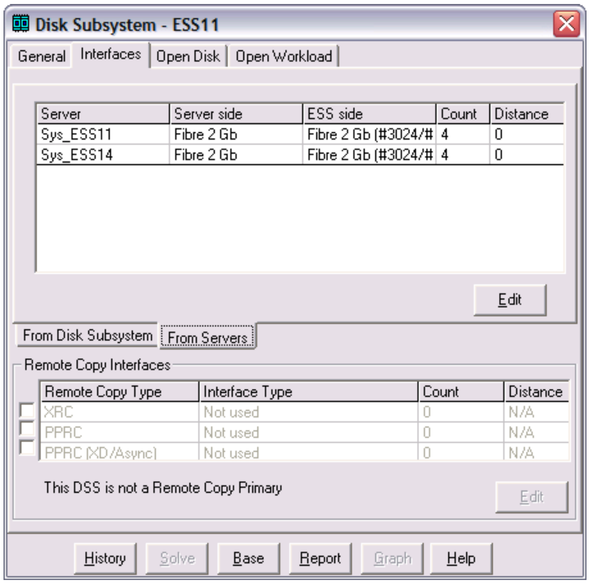


Figure 6-41 Open Systems server to disk subsystem interfaces

9. Select the **From Disk Subsystem** option in Figure 6-41 to display the interface used by the disk subsystem. Figure 6-42 shows that ESS11 uses 8 Fibre 2 Gb Ports. We need to know how many Fibre ports are used, because there are two servers that access this ESS-800. Each server uses four Fibre Channels, so there can be up to eight Fibre ports on the ESS-800. In this particular case, there are eight Fibre ports on the ESS-800. If there are more (or fewer) FICON ports, you can update this information by clicking **Edit**.
10. On this panel, you also indicate whether there is a Remote Copy relationship between this ESS-800 and a remote disk subsystem. On this panel, you can choose to define the connections used between the Primary site and the Secondary site.

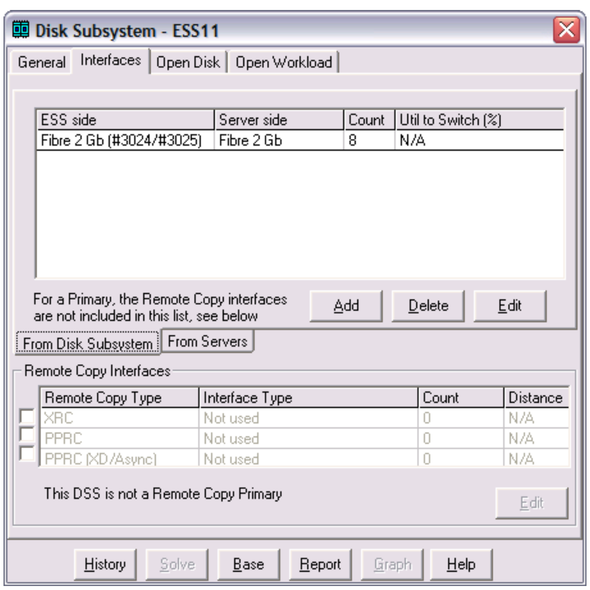


Figure 6-42 Open Systems disk subsystem to server interfaces

11. To see the DDM, click the **Open Disk** tab. Figure 6-43 on page 205 shows DDM options by server. We enter or select the actual configuration specifics of ESS11. ESS11 is accessed by server Sys_ESS11.

The configuration details are:

– Total capacity: `12000 GB`
– DDM type: `73 GB/10K rpm`
– RAID type: `RAID 5`

12. Next, click **Sys_ESS14** in Figure 6-43 and leave the Total Capacity at `0`, because ESS11 is accessed by Sys_ESS11 only.

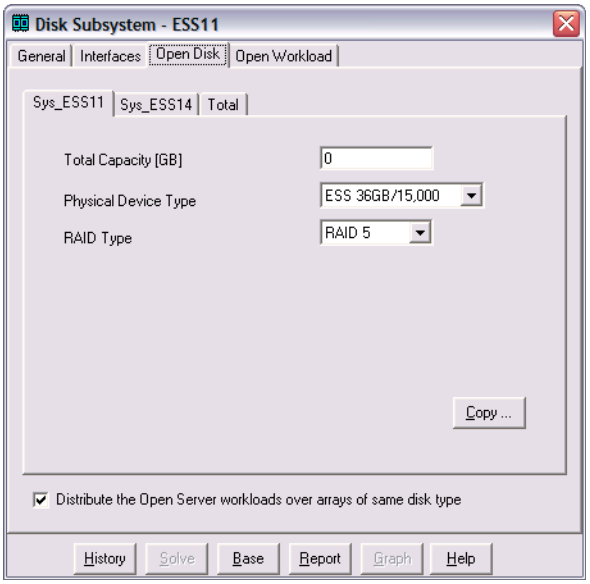


Figure 6-43 Open Systems DDM option by server

13. Select the **Total** tab in Figure 6-43 to display the total capacity of the ESS-800, which is 12 TB on 28 RAID ranks of 73GB/10K rpm DDMs, as shown in Figure 6-44.

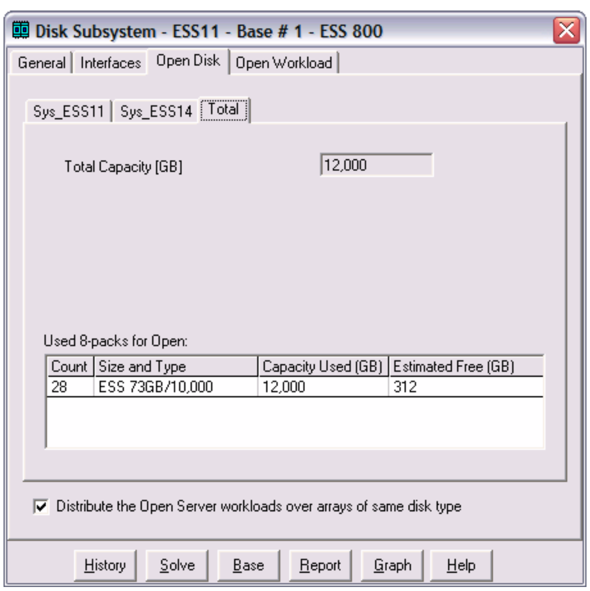


Figure 6-44 Open Systems disk subsystem total capacity

14. To see the last option, select the **Open Workload** tab in Figure 6-44. Figure 6-45 on page 206 shows that the I/O rate from Sys_ESS11 is 6376.8 IOPS and the service time is

4.5 msec. If we click **Average**, we observe the same I/O statistics, because ESS11 is accessed by Sys_ESS11 only. Click **Base** to create the base model for this ESS-800.

15. We can now also create the base model for ESS14 by following the same procedure.

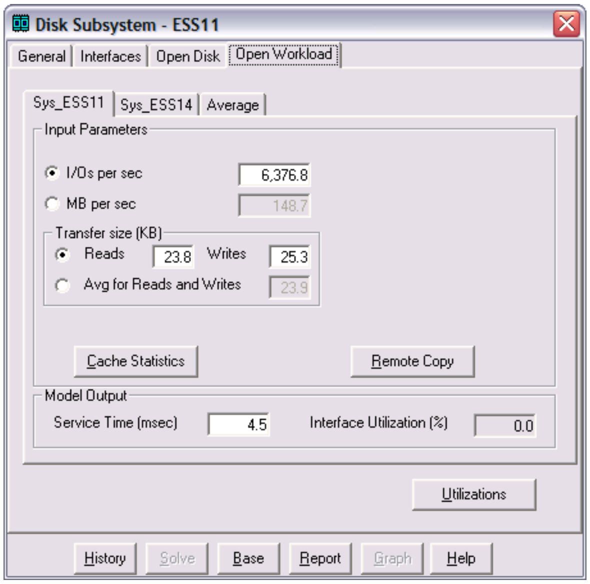


Figure 6-45 Open Systems workload statistics

6.3.2 Open Systems model to merge the two ESS-800s to a DS8300

Now, we start the merge procedure:

1. In Figure 6-46, right-click **ESS11**, and click **Merge** from the pop-up menu. Then, select **Add to Merge Source Collection and create New Target** from the cascading menu. This selection creates `Merge Target1`, which is the new disk subsystem that we use as the merge target (Figure 6-47 on page 207).

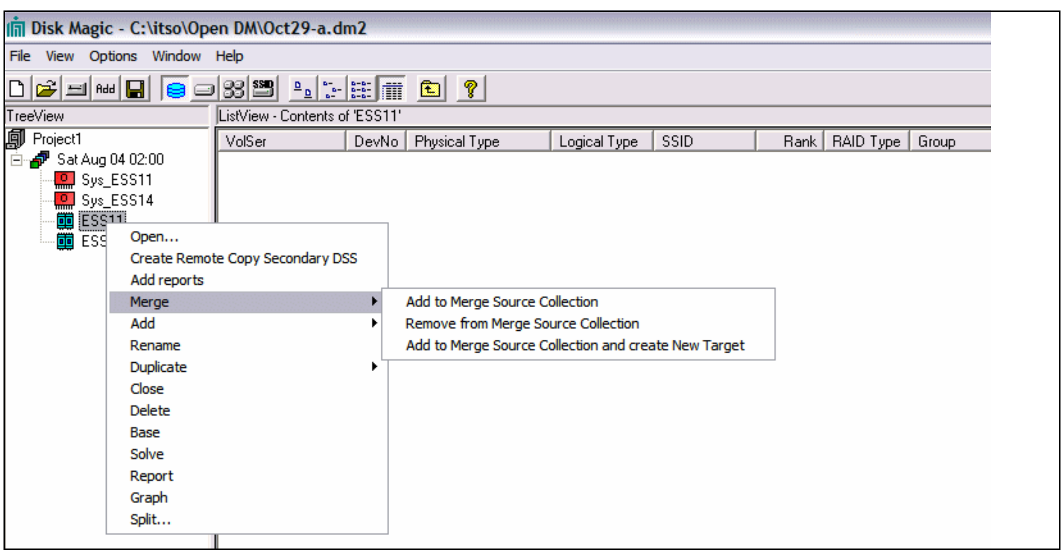


Figure 6-46 Open Systems merge and the creation of a target disk subsystem

2. Because we want to merge the ESS-800s to a DS8300, we need to modify Merge Target1. Click **Hardware Type** in Figure 6-47 to open a list box where we select **IBM DS8300 Turbo**.

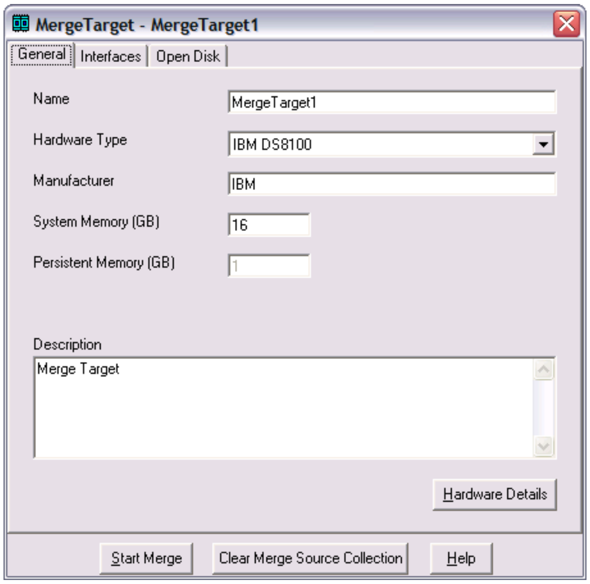


Figure 6-47 Open Systems merge target disk subsystem

3. Selecting Hardware Details opens the window that is shown in Figure 6-48. With the Failover Mode option, you can model the performance of the DS8000 when one processor server with its associated processor storage is lost.

   We can select the cache size, in this case 64 GB, which is the sum of the cache sizes of ESS11 and ESS14. In a DS8300, this selection automatically also determines the NVS size.

   Disk Magic computes the number of HAs on the DS8000 based on the numbers that are specified on the Interfaces page, but you can, to a certain extent, override these numbers. We suggest that you use one HA for every two Fibre ports. In this case, we select four Fibre HAs, because we use eight Fibre ports.

4. Click the **Interfaces** tab in Figure 6-47 to open the dialog that is shown in Figure 6-48.

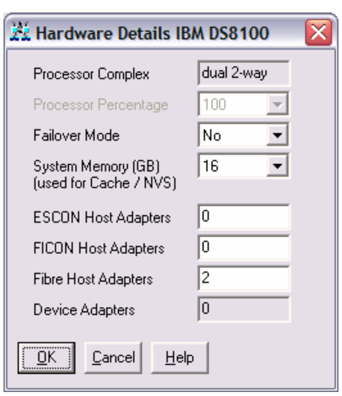


Figure 6-48 Open Systems target hardware details option

5. Because the DS8300 Fibre ports run at 4 Gbps, we need to update this option on both servers (Figure 6-49 on page 208) and also on the From Disk Subsystem dialog

(Figure 6-50). If the servers use different Fibre Channels than the Fibre Channels that are specified, update this information.

Select and determine the Remote Copy Interfaces. You need to select the Remote Copy type and the connections used for the Remote Copy links.

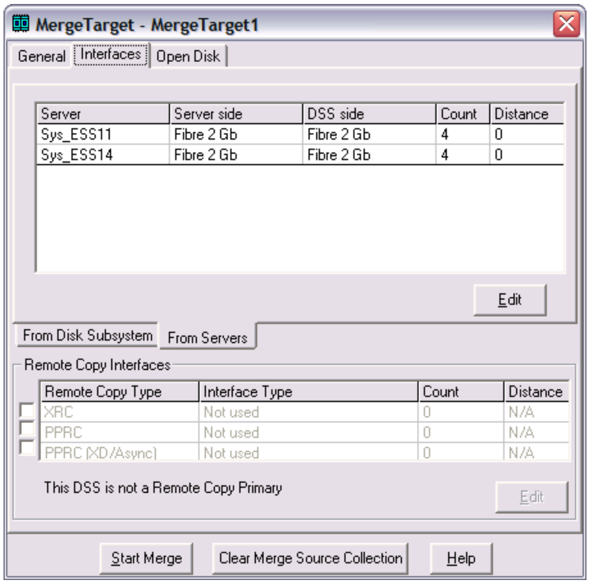


Figure 6-49 Open Systems target interfaces from server

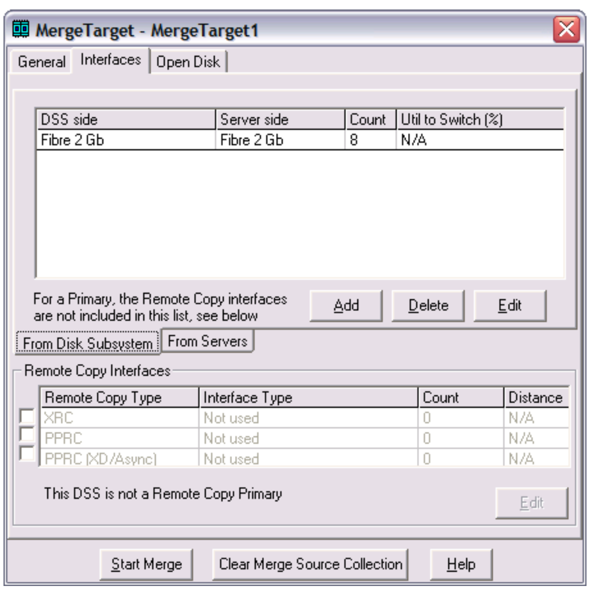


Figure 6-50 Open Systems target interfaces from disk subsystem

6. To select the DDM capacity and rpm used, click the **Open Disk** tab in Figure 6-51 on page 209. Then, select the DDM used by clicking **Add**. Select the HDD type used (**146GB/15K rpm**) and the RAID type (**RAID 5**), and enter capacity in GB (`24000`). Now, click **OK**.

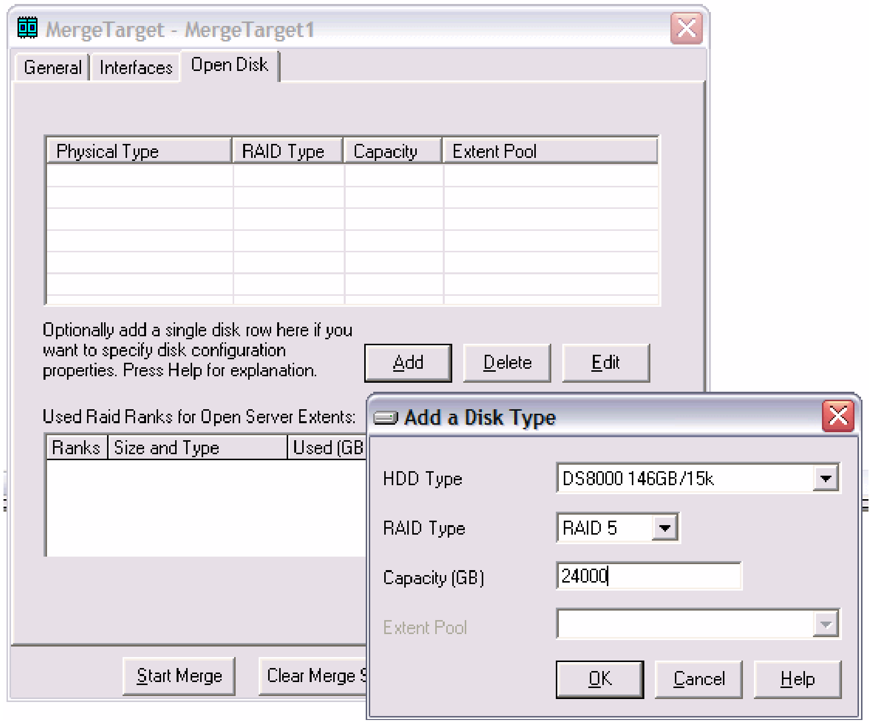


Figure 6-51 Open Systems target DDM option

7. Now, merge the second ESS onto the target subsystem. In Figure 6-52, right-click **ESS14**, select **Merge**, and then, select **Add to Merge Source Collection**.

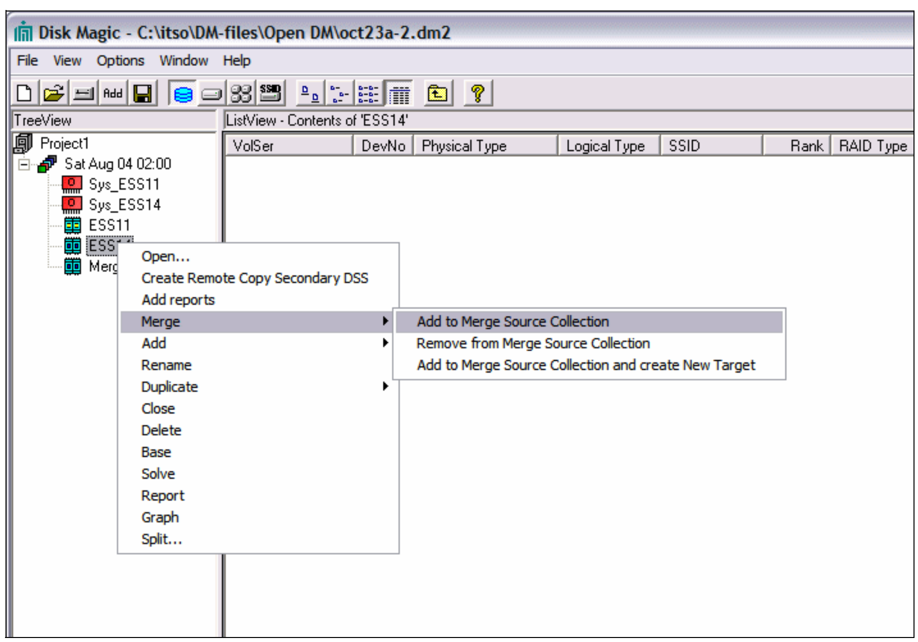


Figure 6-52 Open Systems merge second ESS-800

8. To start the merge, in the Merge Target window that is shown in Figure 6-53 on page 210, click **Start Merge**. This selection initiates Disk Magic to merge the two ESS-800s onto the new DS8300. Use the pop-up window to select whether to merge all workloads or only a

subset of the workloads. We select **I want to merge all workloads on the selected DSSs**, which creates Merge Result1 (see Figure 6-54).

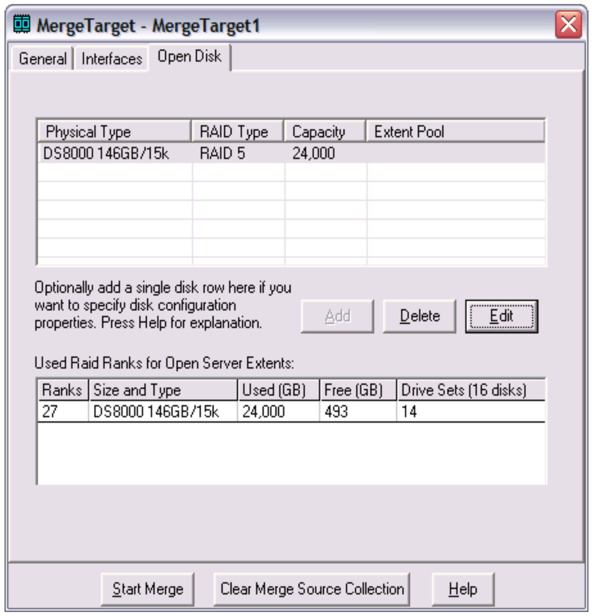


Figure 6-53 Open Systems start the merge

9. Click the **Open Disk** tab in Figure 6-54 to show the disk configuration. In this case, it is 24 TB on 28 ranks of 146 GB/15K rpm DDMs (Figure 6-55 on page 211).

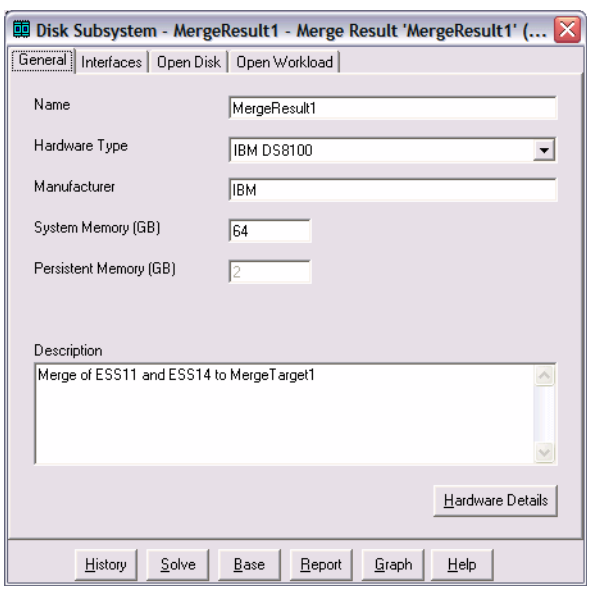


Figure 6-54 Open Systems merge result disk subsystem

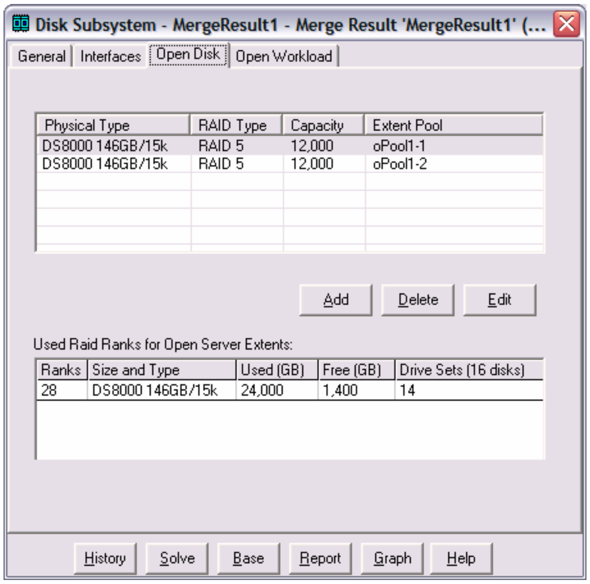


Figure 6-55 Open Systems DDM of the new DS8300

10.Select the **Open Workload** tab in Figure 6-55 to show the Disk Magic predicted performance of the DS8300. We see that the modeled DS8300 has an estimated service time of 5.9 msec (Figure 6-56).

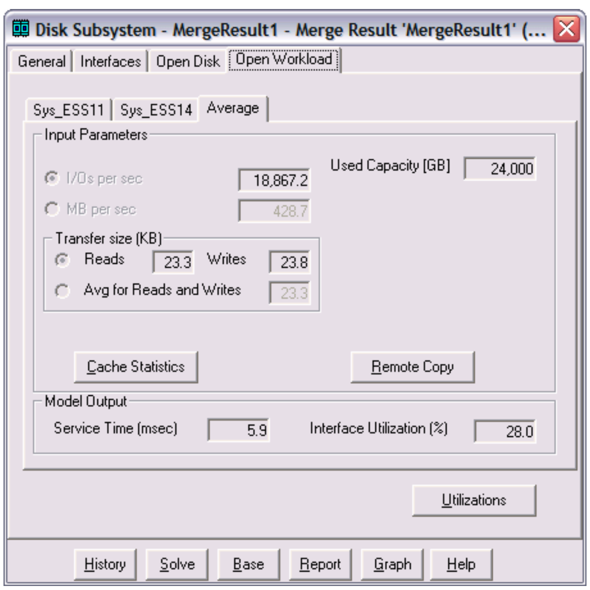


Figure 6-56 Open Systems performance statistics on the DS8300

11.Click **Utilizations** in Figure 6-56 to show the utilization statistics of the various components of the DS8300. In Figure 6-57 on page 212, we see that the **Highest HDD Utilization** is 60.1% and has a darker (amber) background color. This amber background is an indication that the utilization of that resource is approaching its limit. It is still acceptable, but the color is a warning that a workload increase might push this resource to its limit.

Any resource that is a bottleneck is shown with a red background. If a resource shows a red background, you need to increase that resource to resolve the bottleneck.

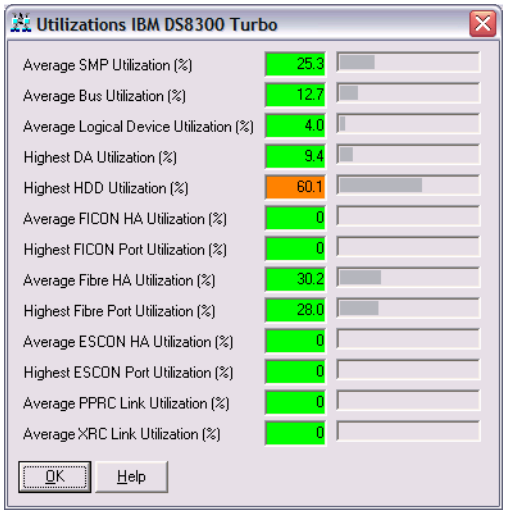


Figure 6-57 Open Systems DS8300 utilization statistics

Use Figure 6-25 on page 195 as a guideline for the various resources in a DS8000. The middle column has an amber background color and the rightmost column has a red background color. The amber number indicates a warning that if the resource utilization reaches this number, a workload increase might soon cause the resource to reach its limit. The red numbers are utilization numbers that cause an increase in one of the components of the response time.

6.3.3 Disk Magic performance projection for an Open Systems model

Based on the previous modeling results, we can create a chart that compares the performance of the original ESS-800s with the performance of the new DS8300. In Figure 6-58, clicking **Sat Aug 04 02:00** copies the server and disk subsystems in the right column.

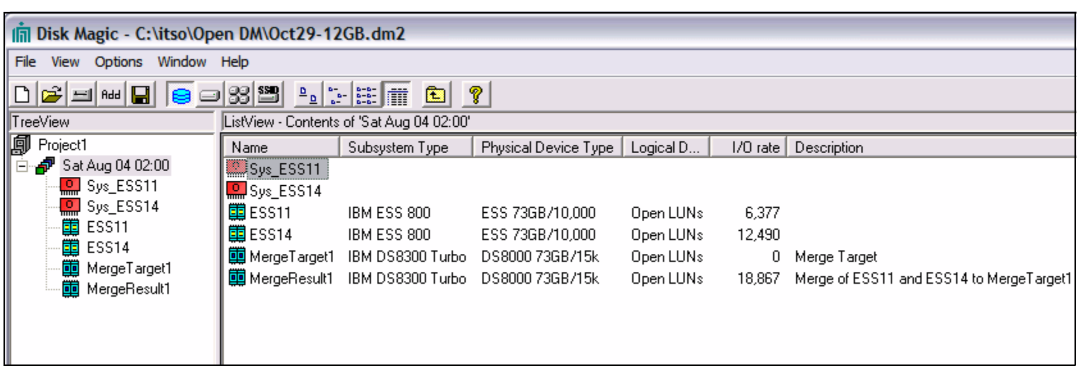


Figure 6-58 Open Systems panel for performance comparison

Hold Ctrl down and select **ESS11**, **ESS14**, and **MergeResult1**. Right-click any of them, and a small window appears. Select **Graph** from this window. On the panel that appears (Figure 6-59 on page 213), select **Clear** to clear any prior graph option settings. Click **Plot** to produce the service time graph of the three selected disk subsystems (Figure 6-60 on page 213).

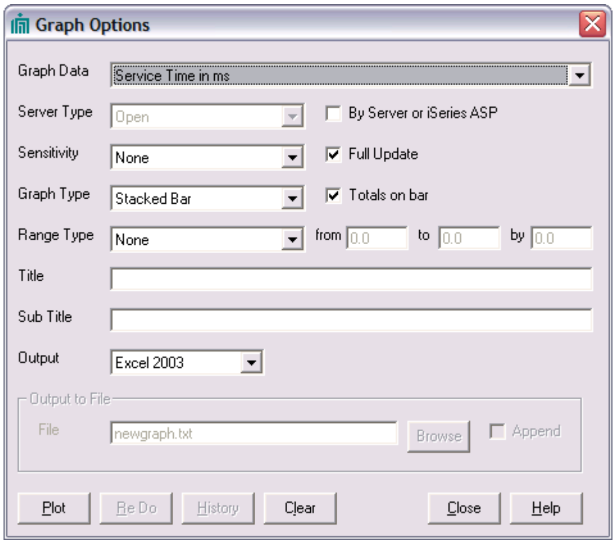


Figure 6-59 Open Systems graph option panel

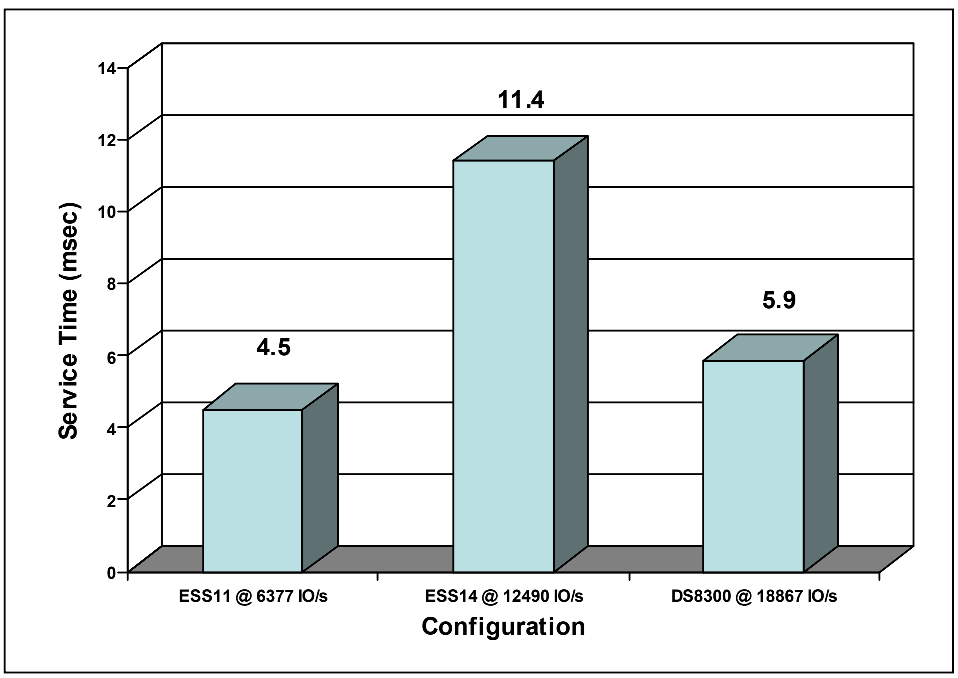


Figure 6-60 Open Systems service time comparison

6.3.4 Workload growth projection for an Open Systems model

A useful feature of Disk Magic is the capability to project workload growth and observe the impact of this growth on the various DS8000 resources and also on the service time. To create this workload growth projection, click **Graph** in Figure 6-56 on page 211, which opens the panel that is shown in Figure 6-61 on page 214.

Click **Range Type**, and choose **I/O Rate**. This selection fills the from field with the I/O Rate of the current workload, which is 18,867.2 IOPSec, the to field with 22,867.2, and the by field with 1,000. You can change these numbers. In our case, we change the numbers to `18000`, 40000, and 1000. Select **Line** for the graph type, and then, select **Clear** to clear up any prior graph settings.

Now, select **Plot**. An error message appears that shows that the HDD utilization > 100%. This message indicates that we cannot increase the I/O rate up to 40000 IOPSec because of the DDM bottleneck. Click **OK** to complete the graph creation.

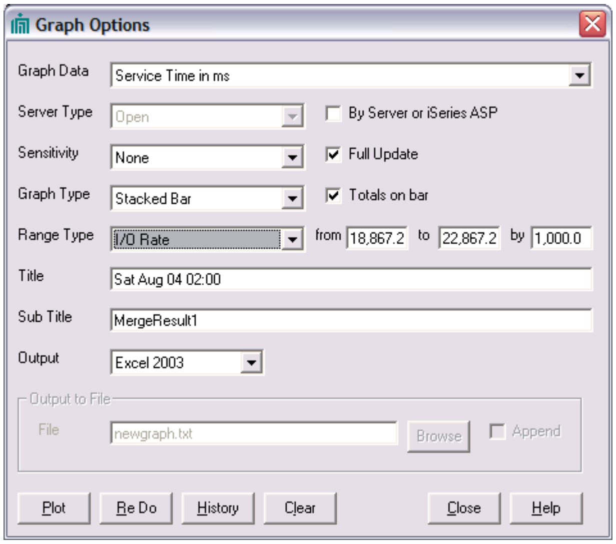


Figure 6-61 Open Systems workload growth projection dialog

The graph shows the service time plotted against the I/O rate increase as shown in Figure 6-62.

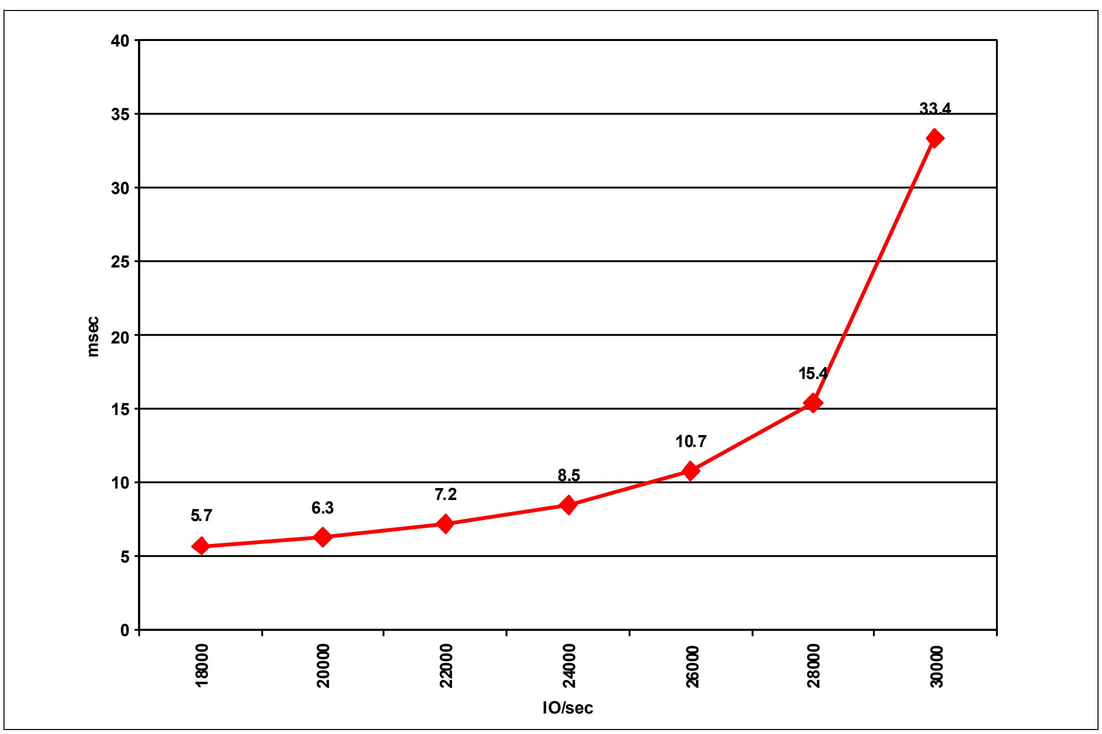


Figure 6-62 Open Systems service time projection with workload growth

This graph shows the DS8300 resource utilization growth with the increase in I/O rate. We can observe that the HDD utilization starts to reach the *red* area at an I/O rate > 24000 IOPS. This utilization impacts the service time and can be seen in Figure 6-62 on page 214, where at greater than 24000 IOPS, the service time increases more rapidly.

After selecting **Utilization Overview** in the Graph Data choices, click **Clear** and click **Plot**, which produces the resource utilization table in Figure 6-63.

			Total I/O Rate (I/Os per second)						
Utilizations	Amber Threshold	Red Threshold	18000	20000	22000	24000	26000	28000	30000
Average SMP	60%	80%	24.1%	26.8%	29.5%	32.2%	34.8%	37.5%	40.2%
Average Bus	70%	90%	12.1%	13.5%	14.8%	16.2%	17.5%	18.9%	20.2%
Average Logical Device	n/a	n/a	3.6%	4.6%	5.8%	7.7%	10.8%	17.4%	42.2%
Highest DA	60%	80%	8.9%	9.9%	10.9%	11.9%	12.9%	13.9%	14.9%
Highest HDD	60%	80%	57.3%	63.7%	70.1%	76.5%	82.8%	89.2%	95.6%
Average FICON HA	35%	50%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Highest FICON Port	35%	50%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Average Fibre HA	60%	80%	28.8%	32.0%	35.2%	38.4%	41.6%	44.8%	48.1%
Highest Fibre Port	60%	80%	26.7%	29.7%	32.7%	35.6%	38.6%	41.6%	44.5%

Figure 6-63 Open Systems workload growth impact on resource utilization

6.4 Disk Magic SSD modeling

Disk Magic provides the capability to advise data placement on SSDs. Supplied with DS8000 configurations with SSDs and multiple servers, Disk Magic can select specific server workloads that benefit the most from running on SSDs. This feature is called *SSD Advisor* and applies both to System z and Open Systems.

6.4.1 SSD advisor example

In the following section, we describe how to use the SSD Advisor to predict the performance improvement after you move a selected workload to SSDs. The base model for this example consists of a DS8800 subsystem attached to two AIX systems. The workload statistics for systems AIX1, AIX2, and the combined systems are depicted in Figure 6-64 on page 216, Figure 6-65 on page 216, and Figure 6-66 on page 217.

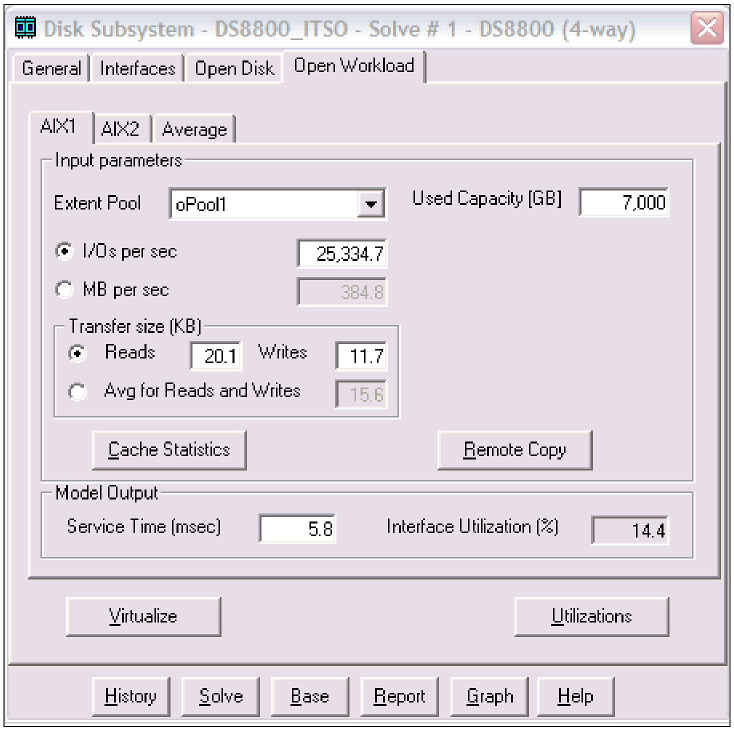


Figure 6-64 Server AIX1 workload summary in the base model

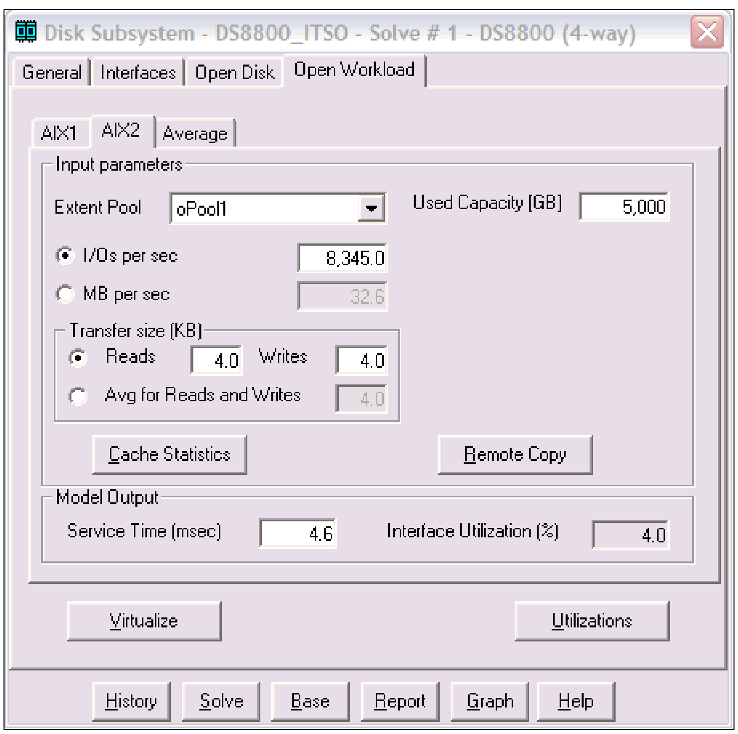


Figure 6-65 Server AIX2 workload summary in the base model

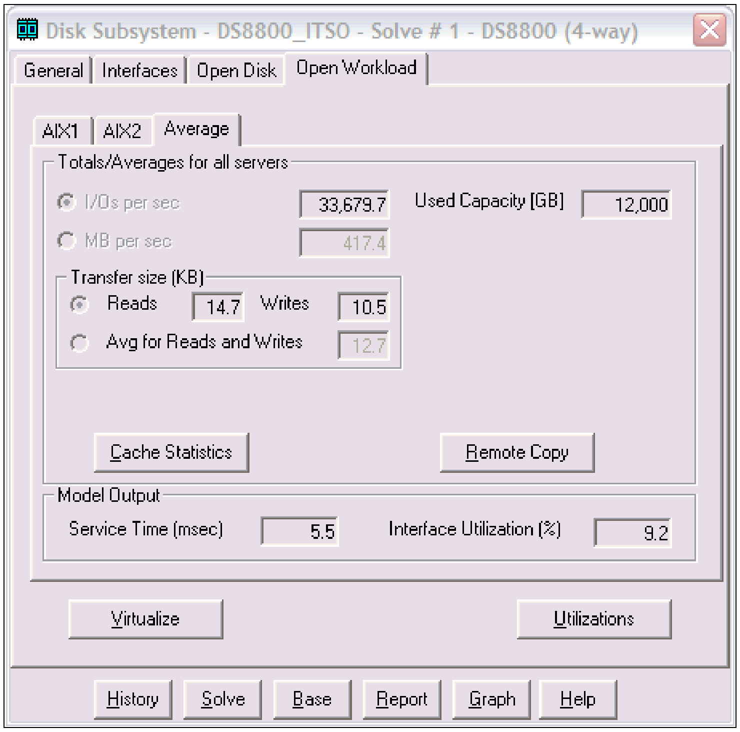


Figure 6-66 Combined AIX1 and AIX2 workload in the base model

The resource utilizations are shown in Figure 6-67.

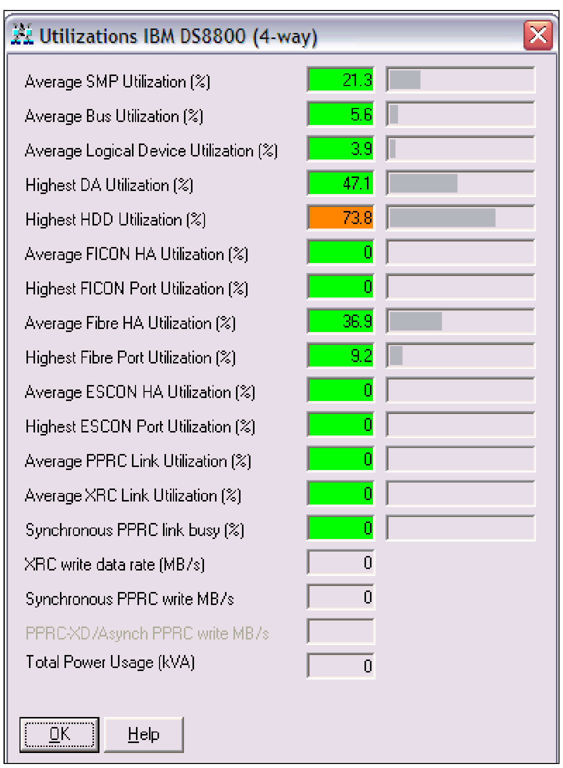


Figure 6-67 Resource utilization in the base model

Follow these steps:

1. To start the SSD modeling, go to the Disk Subsystem main panel, select the **Open Disk** tab and click **extent pools**. The dialog shown in Figure 6-68 opens.

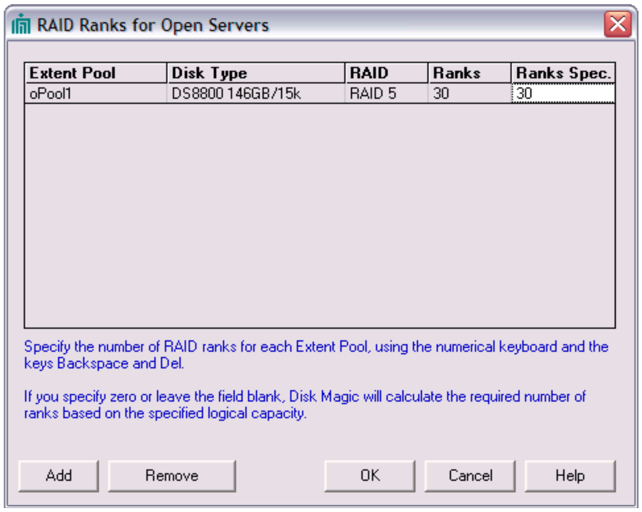


Figure 6-68 RAID Ranks for Open Servers dialog

2. Click **Add** to add the new extent pool `oPool2` with `4` SSD ranks, as depicted in Figure 6-69.

Figure 6-69 Adding four SSD ranks

3. Click **OK** in the Add RAID Ranks dialog to complete the new extent pool creation. By clicking **OK** in the RAID Ranks for Open Systems dialog, Disk Magic asks whether you want to start the SSD advisor, as shown in Figure 6-70.

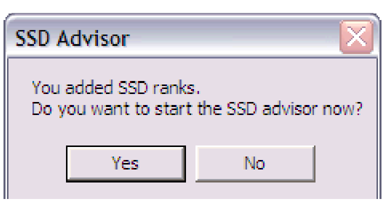


Figure 6-70 SSD advisor

4. Answer **Yes** to the question and Disk Magic tries to identify a suitable server workload to move onto SSDs. If it succeeds, the window that is depicted in Figure 6-71 on page 219 opens.

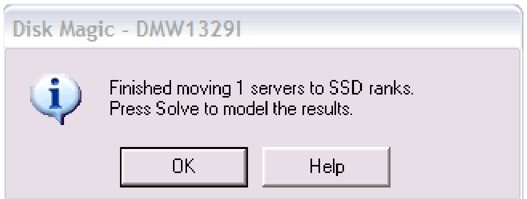


Figure 6-71 One workload selected for SSDs

5. In this example, Disk Magic identified the AIX2 workload to move to SSDs. On the main Disk Subsystem dialog, select **Solve** to create the model. Figure 6-72 and Figure 6-73 on page 220 show the predicted workload statistics for AIX2 and AIX1.

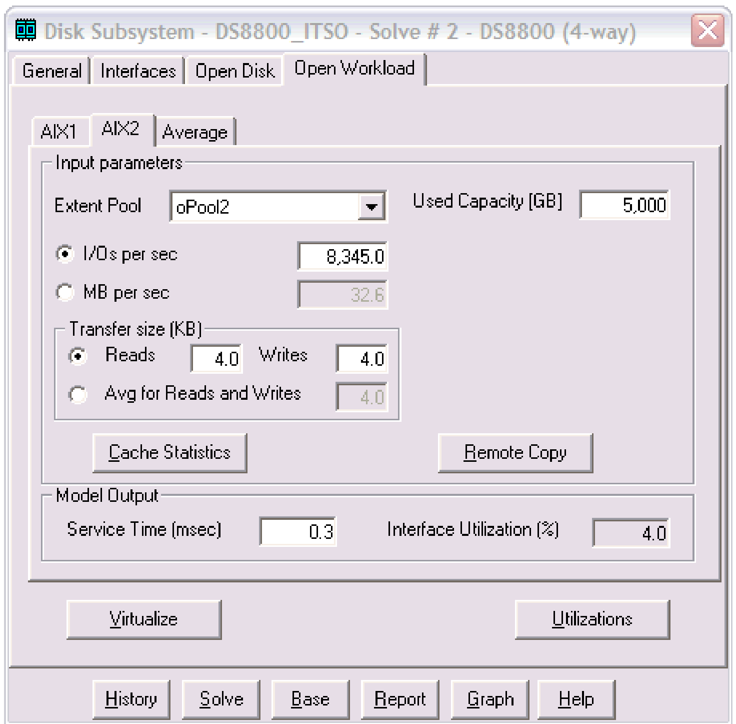


Figure 6-72 Server AIX2 workload summary in the new model

6. The extent pool field now shows oPool2 for AIX2, because Disk Magic decided to move this workload to SSDs.

Figure 6-73   Server AIX1 workload summary in the new model

7. The AIX1 workload also benefits from moving the AIX2 workload to SSDs, because the HDDs are now less utilized, as shown in Figure 6-74.

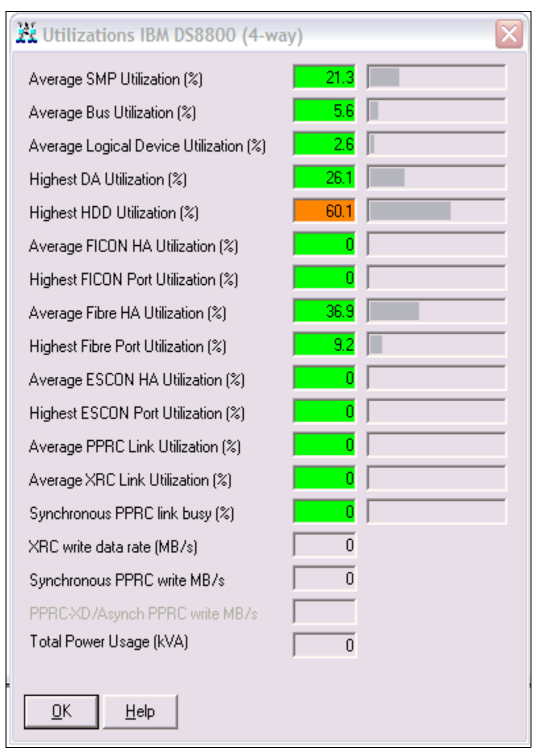


Figure 6-74   Resource utilization in the new model

6.5 Disk Magic Easy Tier modeling

Disk Magic supports DS8000 Easy Tier modeling for DS8800 and DS8700 multi-tier configurations. The performance modeling that is provided by Disk Magic is based on the workload *skew level* concept. The skew level describes how the I/O activity is distributed across the capacity for a specific workload. A workload with a low skew level has the I/O activity distributed evenly across the available capacity. A heavily skewed workload has many I/Os to only a small portion of the data. The workload skewing affects the Easy Tier effectiveness, especially if there is a limited number of high performance ranks. The heavily skewed workloads benefit most from the Easy Tier capabilities, because even when moving a small amount of data, the overall performance improves. With lightly skewed workloads, Easy Tier is less effective, because the I/O activity is distributed in such a large amount of data that it cannot be completely moved to a higher performance tier.

Predefined skew levels

Disk Magic supports three[1] predefined skew levels to use in Easy Tier predictions: *heavily skewed*, *medium skewed,* and *lightly skewed*. Disk Magic uses this setting to predict the number of I/Os that are serviced by the higher performance tier. In Figure 6-75, the three curves represent the predefined skew levels in respect to the capacity and I/Os.

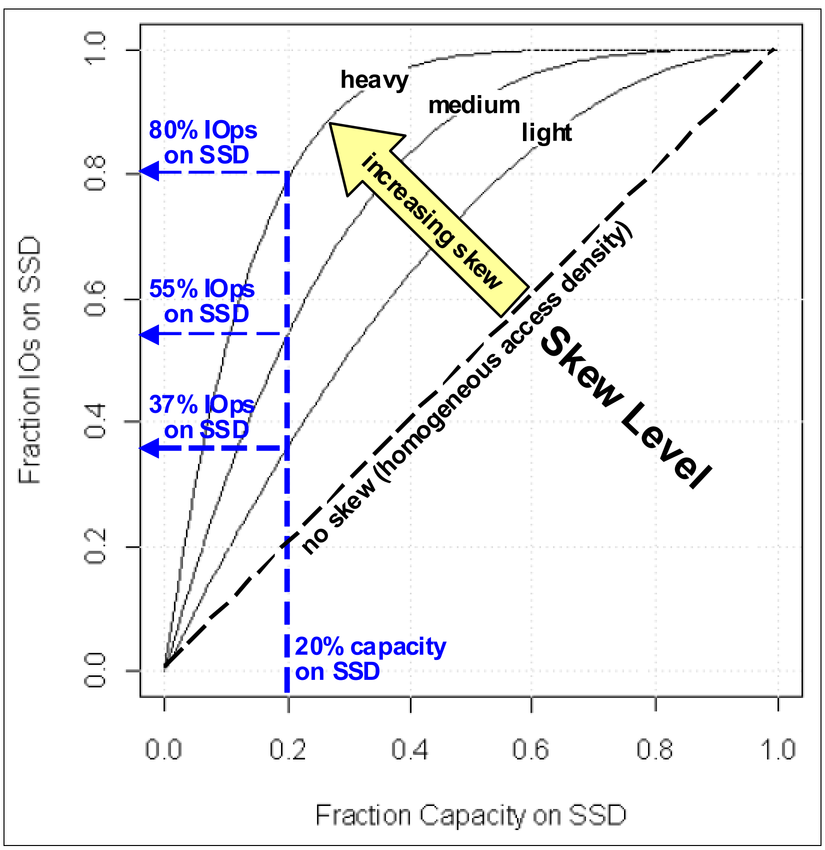


Figure 6-75 Disk Magic skew level curves

[1] Release 9.1.*x* of Disk Magic introduced an extended set of five predefined skew levels: *very low skew*, *low skew*, *intermediate skew*, *high skew,* and *very high skew*. The previous skew levels *light*, *medium,* and *heavy* proved to be too conservative in general. Many workloads showed even higher skew levels. The new skew levels are adopted to address this increase by introducing new *high* and *very high* skew levels. The previous skew levels now map to light=very low skew, medium=low skew, and heavy=intermediate skew.

So, the top curve represents the heavy skew level. The lower curve represents the light skew level. In this chart, the value for the heavily skewed curve for 20% SSD capacity is 0.8. Disk Magic assumes that if there is an extent pool where 20% of the capacity is on SSD ranks, Easy Tier manages to fill this 20% of the capacity with data that handles 80% of all the I/Os. For the medium skewed workload, the 20% SSD ranks get 55% of the I/Os. For the lightly skewed workload, the SSD ranks get 37% of the I/Os. The skew level settings affect the Disk Magic predictions: a high skew level selection results in a more aggressive sizing of the higher performance tier. A low skew level selection provides a conservative prediction. It is important to understand which skew level best matches the actual workload before you start the modeling.

Calculated skew levels

Disk Magic also provides capabilities to estimate the skew level of a specific workload. These estimations are based on the determination of *skew level value*, which is a numeric representation of the skew level. A skew level value of 1.00 represents a theoretical case without any workload skew. The predefined skew levels described previously correspond with skew level values of 2.00, 3.50, and 7.00.

For Open Systems and System z workloads, Disk Magic uses the IBM TotalStorage Productivity Center for Disk data that is provided as input. By examining the workload distribution over the logical volumes, it estimates a skew level value. For a System z multi-tier configuration, Disk Magic can also determine a skew level value so that the computed average disconnect times for the tiers match the measured values.

6.5.1 Easy Tier prediction

In the following section, we describe how to use Disk Magic to predict the performance improvement after the introduction of Easy Tier in a multi-tier count key data (CKD) configuration. The same settings apply also to fixed block (FB) multi-tier models. Our base model consists of a DS8800 that is connected to one System z LPAR. The key performance statistics and the resources utilization are depicted in Figure 6-76 on page 223 and Figure 6-77 on page 223.

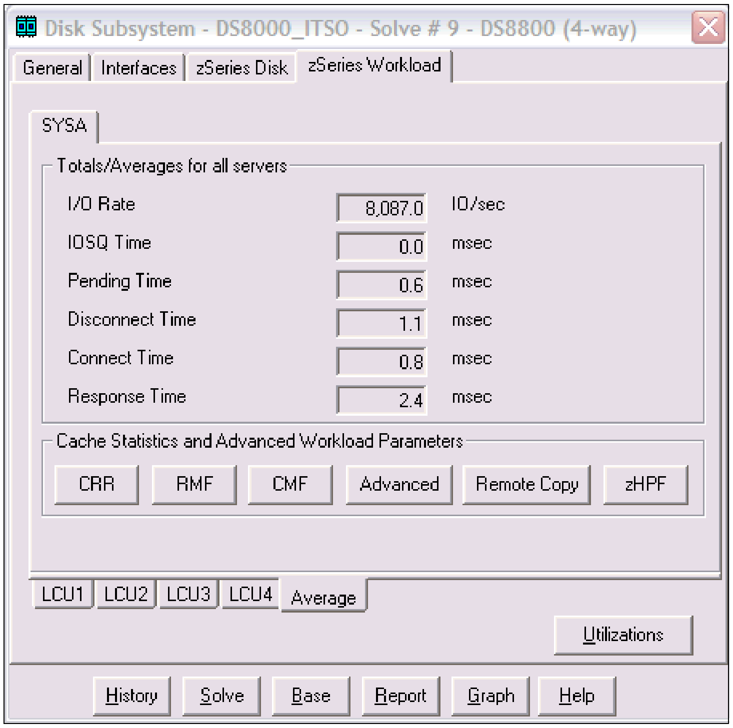


Figure 6-76 Base model workload statistics

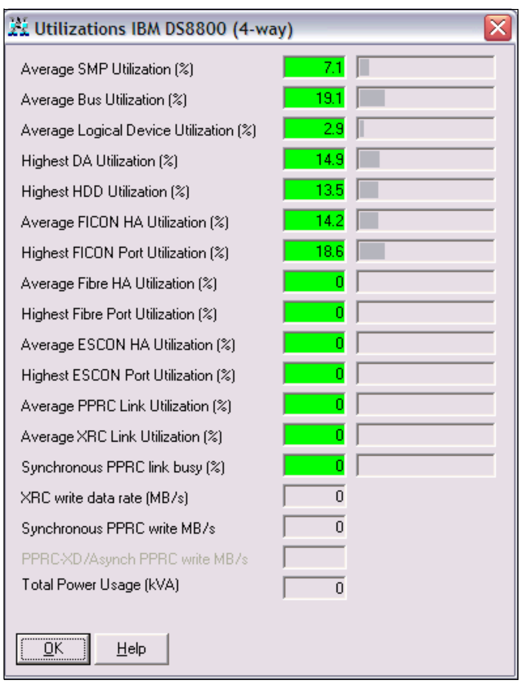


Figure 6-77 Base model resources utilization

To enable the Easy Tier modeling, Disk Magic provides a special settings dialog. On the General tab of the Disk Subsystem dialog, you can see Easy Tier Settings, as depicted in Figure 6-78 on page 224.

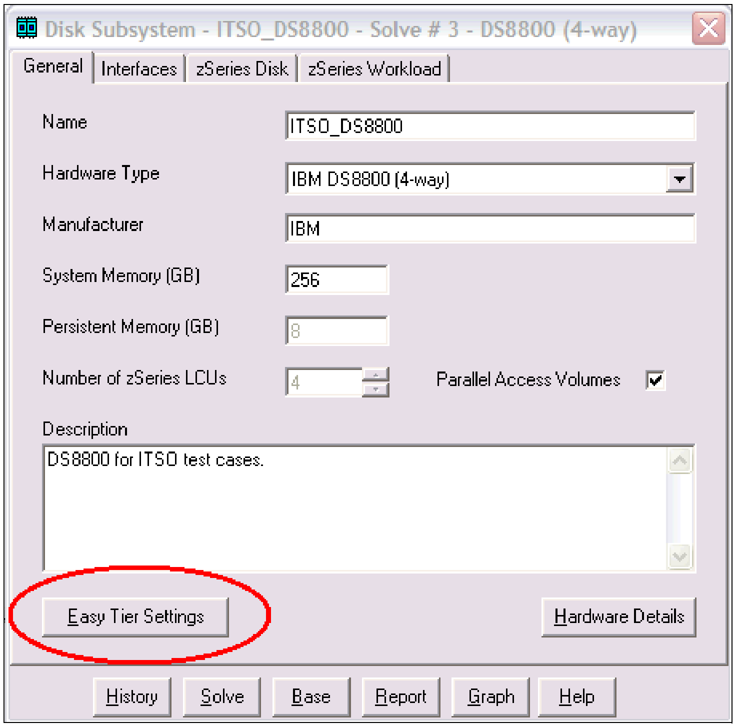


Figure 6-78 Easy Tier Settings

Follow these steps to use the Easy Tier settings:

1. The Easy Tier Settings dialog appears, as shown in Figure 6-79. Select **Enable Easy Tier** to enable the Easy Tier modeling. To use the predefined skew levels, select **Use Predefined Skew Level** and then choose the desired skew level. In this example, we are modeling a highly skewed workload. Click **OK**.

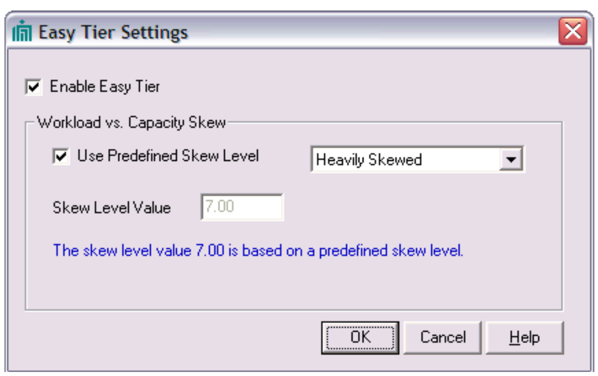


Figure 6-79 Easy Tier Settings dialog

2. Select the **zSeries Disk** tab in the Disk Subsystem dialog. The new tab format appears. Now, you can mix different drive types in the same extent pool, as shown in Figure 6-80 on page 225. Click **Add** in the RAID Rank Definitions section to add new ranks into an extent pool.

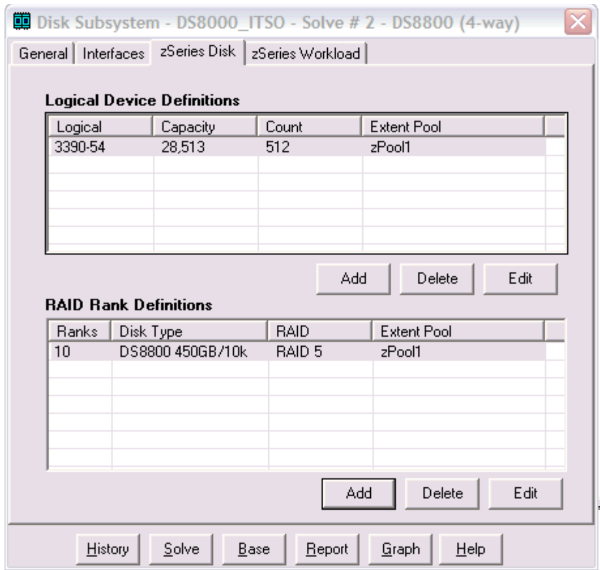


Figure 6-80 zSeries disk dialog

3. In our example, we add two SSD 300 GB ranks to extent pool zPool1, as depicted in Figure 6-81.

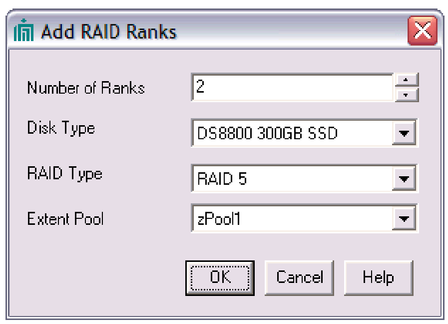


Figure 6-81 Add RAID Ranks dialog

4. Now, we are ready to start the modeling with a multi-tier configuration. In the main Disk Subsystem dialog, click **Solve** to start the modeling. After the modeling completes, the zSeries Workload tab reports the new workload statistics for the model, as depicted in Figure 6-82 on page 226. The predicted Response Time improves from 2.8 msec to 1.8 msec.

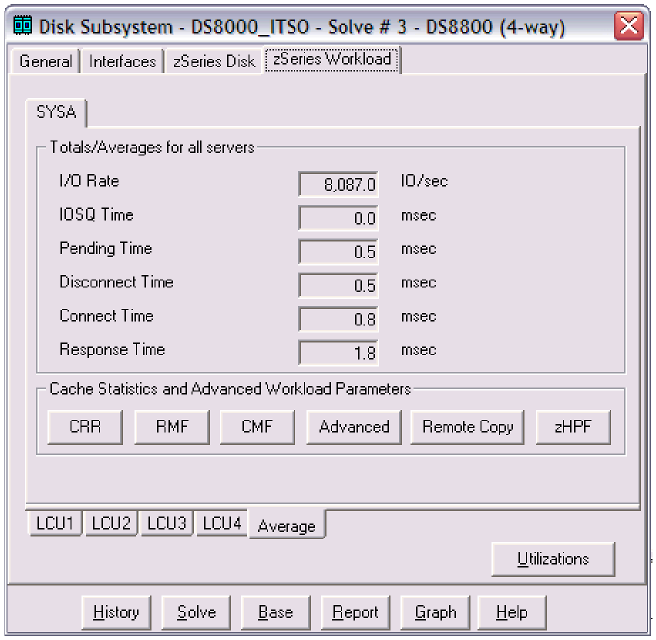


Figure 6-82 New model workload statistics

5. Also, the utilization statistics change, as shown in Figure 6-83.

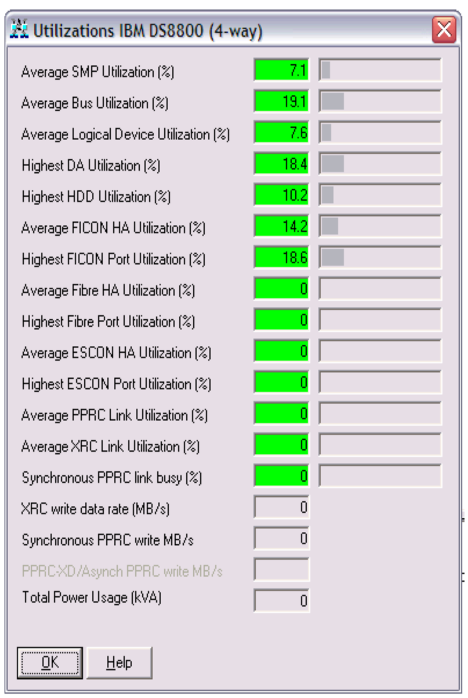


Figure 6-83 New model utilization statistics

6. As result of introducing the SSD, we can see a higher DA utilization and a lower HDD utilization. Click **Report** on the main Disk Subsystem dialog (Figure 6-82). The report for the new model is created. We can see the predicted I/O distribution across the tiers, as reported in Figure 6-84 on page 227.

```
Extent Pool      Type      HDD        RAID     #Ranks Log.Type
zPool1[tr0]      CKD       300GB SSD  RAID 5        2 3390-54
zPool1[tr1]      CKD       450GB/10k  RAID 5       10 3390-54

Extent Pool           I/O IOSQ Pend Conn Disc Resp     Highest
                     Rate                     Time    HDD Util
zPool1[tr0]        4867.8  ---  ---  0.8  0.2  ---    10.2%
zPool1[tr1]        3219.2  ---  ---  0.8  1.1  ---     5.8%
```

Figure 6-84 Disk Magic report tiering information

7. In this example, Disk Magic estimates that about 60% of I/Os are to be performed on SSD ranks, which are about 13% of the total capacity. These figures agree with the high skew level curve that is reported in Figure 6-75 on page 221.

   In the previous example, we use the predefined skew level settings. However, in certain cases, we can ask Disk Magic to estimate the skew level value directly from the workload data. This estimate can be created for both System z and Open Systems.

8. For System z workloads that run in multi-tier configurations, Disk Magic calibrates the skew level value for each LCU on the tiered extent pools that best matches the LCU measured disconnect time. For non-LCU models, a single skew level is determined similarly for the whole DSS. This feature is called *Skew Value Calibration*. To activate the automatic skew level value calibration, in Figure 6-85, enable **Easy Tier** modeling and clear the **Use Predefined Skew Level** check box. To restart the modeling, click **Solve** in the main Disk Subsystem panel.

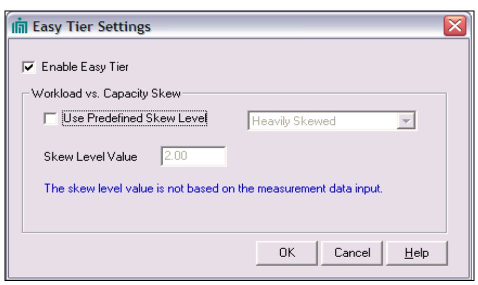


Figure 6-85 Easy Tier Settings panel with cleared Use Predefined Skew Level option

9. In the report that is produced by Disk Magic, you can see the skew values that are estimated at the LCU level, as shown in Figure 6-86.

```
Extent Pool           I/O IOSQ Pend Conn Disc Resp     Highest
                     Rate                     Time    HDD Util
zPool1[tr0]        1871.3  ---  ---  0.8  0.2  ---     3.9%
zPool1[tr1]        6215.7  ---  ---  0.8  1.1  ---    10.6%

   LCU name  Calibrated Skew
   LCU1       2.20
   LCU2       2.60
   LCU3       1.80
   LCU4       1.40
```

Figure 6-86 Disk Magic report tiering information with calibrated skews

Single-tier configurations: The skew value calibration for System z is available with workload measurements of multi-tier configurations only. For single-tier configurations, the calibration reports a skew value of 0.5.

Hint: When the workload measurements are inadequate for a skew value calibration, Disk Magic returns the default skew value (2.00) for all the LCUs or DSS. Use the skew value calibration feature only with real workloads that are provided through RMF data (RMF Loader).

Disk Magic provides another skew value estimation capability with the Tivoli Productivity Center Loader facility. This skew value calculation method applies both to System z and Open Systems. The following inputs are needed by the Tivoli Productivity Center Loader to estimate the skew level:

- CSV files that contain the Tivoli Productivity Center data at the volume level
- CSV file that contains the output of the DS8000 command-line interface (DSCLI) command:
  - `lsfbvol -l -fmt delim -delim ; -fullid` for Open Systems
  - `lsckdvol -l -fmt delim -delim ; -fullid` for System z

The Tivoli Productivity Center Loader tries to calculate the skew value by analyzing the logical volume I/O density. If a skew value is successfully calculated, it is reported on the Easy Tier Settings dialog.

Disk Magic performs skew value estimations by making assumptions that are based on measured performance statistics, such as disconnect times and I/O density. Although these assumptions are generally valid, in certain cases, they cannot be accepted. For example, large LUNs with high I/O density can bring Disk Magic to overestimate the amount of extents to be placed into the higher performance tier. This situation results in an aggressive sizing.

6.6 General configuration planning guidelines

The following general guidelines are designed to assist your storage management planning in estimating the required hardware components for a specific workload. The purpose of this guideline is to estimate the required configuration to support a workload with known I/O characteristics. For more information about workload types and profiles, see Chapter 5, “Understanding your workload” on page 157.

These guidelines do not represent any type of guarantee of performance and do not replace the more accurate estimation techniques that can be obtained from a Disk Magic study.

Workload

The example is based on an online workload with the assumption that the transfer size is 4K and all the read and write operations are random I/Os. The workload is a 70/30/50 online transaction processing (OLTP) workload, which is an online workload with 70% reads, 30% writes, and a 50% read-hit-ratio. We estimate these workload characteristics:

- Maximum host I/O rate is 10000 IOPS.
- Write efficiency is 33%, which means that 67% of the writes are destaged.
- Ranks use a RAID 5 configuration.

DS8000 DDM configuration

We use the following DDM characteristics.

DDM speed

For I/O intensive workloads, consider 15K rpm DDMs, or a mix with SSDs.

DDM capacity

You must estimate the capacity. Choose from among 146 GB, 300 GB, 450 GB, 600 GB, or 900 GB DDMs based on these factors:

- Total capacity that is needed in GBs
- Estimated Read and Write I/O rates
- RAID type used

For a discussion about RAID levels, space efficiency, and write penalty, see 4.7, "Planning RAID arrays and ranks" on page 103.

Based on the workload characteristics, we use this calculation for RAID 5:

- Reads: Read misses are 10000 x 70% x 50% = 3500 IOPS
- Writes: 10000 x 30% x 67% x 4 = 8040 IOPS
- Total = 11540 IOPS

RAID 5: The RAID 5 write penalty of four I/O operations per write is shown in Table 4-1 on page 106 in the Performance Write Penalty column.

Calculate the number of required ranks based on the following information:

- A 15K rpm DDM can sustain a maximum of 200 IOPS/DDM.

  For a 10K rpm DDM, reduce this number by 33%.
- For planning purposes, use a DDM utilization of 50%:
  - The (6+P) rank can sustain 200 x 7 x 50% = 700 IOPS.
  - The (7+P) rank can handle a higher IOPS. To be conservative, calculate these estimates based on the throughput of a (6+P) rank.
- Based on the 11540 total IOPS, this calculation yields 11540 / 700 = 17 ranks.
- Because you can order DDMs based on a multiple of two ranks only, if you require a capacity with 17 ranks, you actually require a DS8000 configuration with 18 ranks.

Depending on the DDM size, the table in Figure 6-87 shows how much capacity you can get with 18 ranks for three selected drive types. Knowing the total GB capacity that is needed for this workload, use this chart to select the DDM size that can meet the capacity requirement.

	146 GB	300 GB	450 GB
Total GB capacity for 18 Ranks			
System z	16,092	32,792	49,544
Open	16,295	33,224	50,202

Figure 6-87 Total rank capacity in GB by DDM size (DS8800 2-way RAID 5)

Tip: Use larger DDM sizes only for applications that are less I/O intensive, or as part of a tiered configuration.

DS8000 FICON or Fibre ports and host adapters

If the workload runs on a System z and the CEC is a z10™ with FICON Express4 channels, perform the following estimate:

- The maximum throughput on the FICON Express4 channel is 14000 IOPS (without zHPF).
- At 50% utilization, you get a throughput of 7000 IOPS.
- Divide the total IOPS by this throughput:
  - 10000/7000 = 1.43 and round this number up to 2.
  - This result means that you need two FICON Express4 channels.
  - For connectivity reasons, increase this number to four FICON Express4 channels.
- For availability reasons, spread the four channels across a minimum of two host adapters.

For Open Systems loads, make sure that the DS8000 adapters and ports do not reach their throughput limits (MB/s values). If you must fulfill specific throughput requirements, ensure that each port under peak conditions is only loaded with a maximum of approximately 70% of its nominal throughput. Also, consider the guidelines in 4.10.1, "I/O port planning considerations" on page 147.

If you do not have throughput requirements, use the following rule to obtain an initial estimate of the number of host adapters:

- For each TB of capacity that is used, configure a nominal throughput of 100 MB/s.

  For instance, 64 TB disk capacity then leads to 6400 MB/s required nominal throughput.
- With 8-Gbps ports assumed, you need eight DS8000 I/O ports.

In 4.10.1, "I/O port planning considerations" on page 147, we learned that the performance of an 8 Gbps host adapter (HA) does not scale with more than four ports. Therefore, you need to plan for a minimum of two 8 Gbps host adapters in this example, which is based on an estimate for the throughput.

With an actual throughput requirement of this size, consider that the actual throughput that can be sustained by a host adapter also depends on the workload profile and, therefore, is typically less than the sum of the nominal 8 Gbps Fibre Channel port throughputs. In general, plan for a utilization that is no more than approximately 70% of the nominal throughput. In this case, we suggest more than two host adapters. Contact your IBM representative for an appropriate sizing that is based on your actual workload requirements.

DS8000 cache

Use Figure 6-88 as a guide for the DS8000 cache size if you do not have workload history that you can use.

Useable Capacity (TB)	Minimum Recommended System Memory (GB)				
	Open and System i		z/OS		Both
	High Performance	Standard Performance	High Performance	Standard Performance	Low Access or Archival
Up to 5 TB	16 GB	16 GB	32 GB	16 GB	16 GB
5 TB to 11 TB	32 GB	16 GB	64 GB	32 GB	16 GB
11 TB to 23 TB	64 GB	32 GB	128 GB	64 GB	16 GB
23 TB to 51 TB	128 GB	64 GB	256 GB	128 GB	32 GB
51 TB to 118 TB	256 GB	128 GB	384 GB	256 GB	64 GB
over 118 TB*)	384 GB	256 GB	384 GB	384 GB	128 GB

Figure 6-88 Cache guideline that is based on the DS8000 capacity

Validation

To finalize your DS8000 configuration, contact your IBM representative to further validate the workload against the selected configuration. You especially need to validate the workload against the selected configuration for larger capacities or high throughput requirements.

6.7 Storage Tier Advisor Tool

The Storage Tier Advisor Tool (STAT) is a Microsoft Windows application that can be used to analyze the characteristics of the workload that runs on the storage facility. The STAT provides capacity planning information that is associated with the current or future use of the Easy Tier facility.

STAT overview

The advisor tool processes data that is collected by the Easy Tier monitor. The DS8000 monitoring capabilities are available regardless of whether you install and activate the Easy Tier license feature on your DS8000. The monitoring capability of the DS8000 enables it to monitor the usage of storage at the volume extent level. Monitoring statistics are gathered and analyzed every 24 hours. The results of this data are summarized in summary monitor data that can be downloaded from a DS8000 for reporting with the advisor tool.

The advisor tool provides a graphical representation of performance data that is collected by the Easy Tier monitor over a 24-hour operational cycle. You can view the information that is displayed by the advisor tool to analyze workload statistics and evaluate which logical volumes might be candidates for Easy Tier management. If the Easy Tier feature is not installed and enabled, you can use the performance statistics that are gathered by the monitoring process to help you determine whether to use Easy Tier to enable potential performance improvements in your storage environment and to determine optimal SSD or HDD configurations and benefits.

After you know your microcode version (DSCLI command **`ver -l`**), you can download the suitable STAT version from the IBM FTP site:

`ftp://ftp.software.ibm.com/storage/ds8000/updates/DS8K_Customer_Download_Files/Storage_Tier_Advisor_Tool/`

To extract the summary performance data that is generated by the Storage Tier Advisor Tool, you can use either the DSCLI or DS Storage Manager. When you extract summary data, two files are provided, one for each processor complex in the Storage Facility Image (SFI server). The download operation initiates a long running task to collect performance data from both selected storage facility images. This information can be provided to IBM if performance analysis or problem determination is required.

> **Availability:** Easy Tier monitor is available on any DS8000 system, starting with LMC level 6.5.1.*xx* (DS8700) or LMC 7.6.10.*xx* (DS8800), with or without the Easy Tier LIC feature or SSDs.

STAT improvement

In its initial version, the STAT estimated performance improvement and SSD configuration guidelines were performed for the storage system as a whole.

The newer version of the STAT that is available with DS8000 microcode R6.2 is more granular and it provides a broader range of recommendations and benefit estimations. The recommendations are available for all the supported multi-tier configurations and they are on a per-extent pool basis. The tool estimates the performance improvement for each pool by using the existing SSD ranks, and it provides guidelines on the type and number of SSD ranks to configure. The STAT also provides recommendations for sizing a Nearline tier so you can evaluate the advantage of demoting extents to either existing or additional nearline ranks, and the cold data capacity that results from this cold demotion.

Another improvement in the STAT is a better way to calculate the performance estimation. Previously, the STAT took the heat values of each bucket as linear, which resulted in an inaccurate estimation when the numbers of extents in each bucket are disproportional. The STAT now uses the average heat value that is provided at the sub-LUN level to provide a more accurate estimate of the performance improvement.

6.7.1 STAT output samples

We provide several STAT output samples with short descriptions. For a full description of the STAT capabilities and their usage, see *IBM System Storage DS8000 Easy Tier*, REDP-4667.

The STAT (Figure 6-89) describes the Easy Tier statistical data that is collected by the DS8000 in detail and it produces reports in HTML format that can be viewed by using a standard browser. These reports provide information at the levels of a DS8000 storage system, the extent pools, and the volumes. The reports also show whether certain pools are bandwidth (BW)-overloaded, IOPS-overloaded, or skewed. Sizing recommendations and estimated benefits are also in the STAT reports.

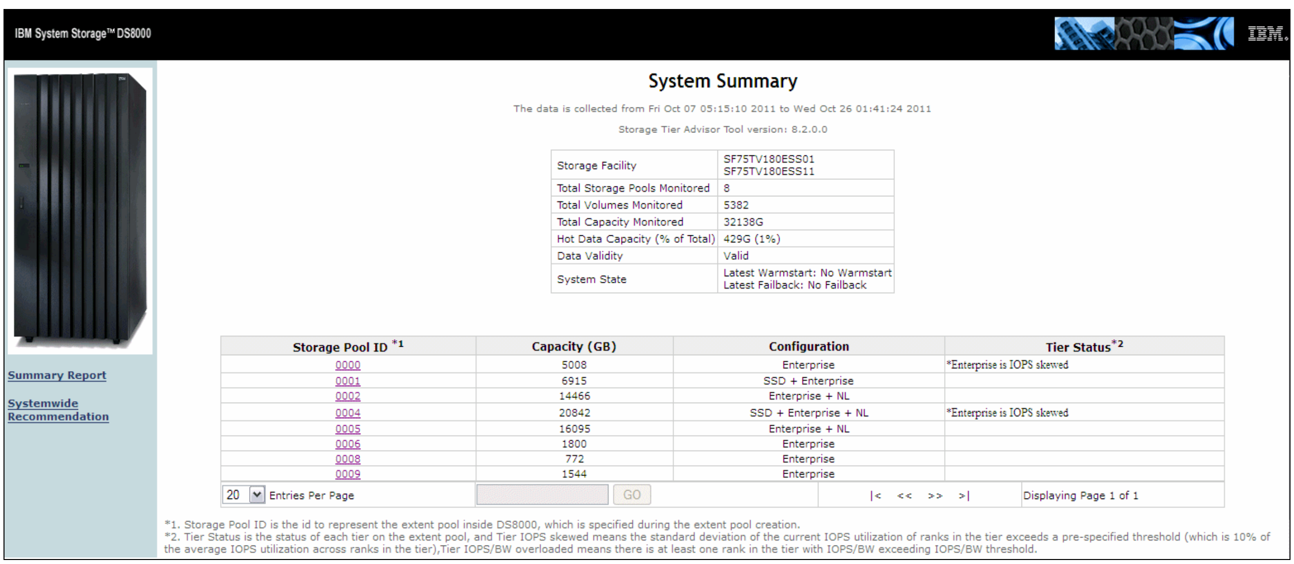
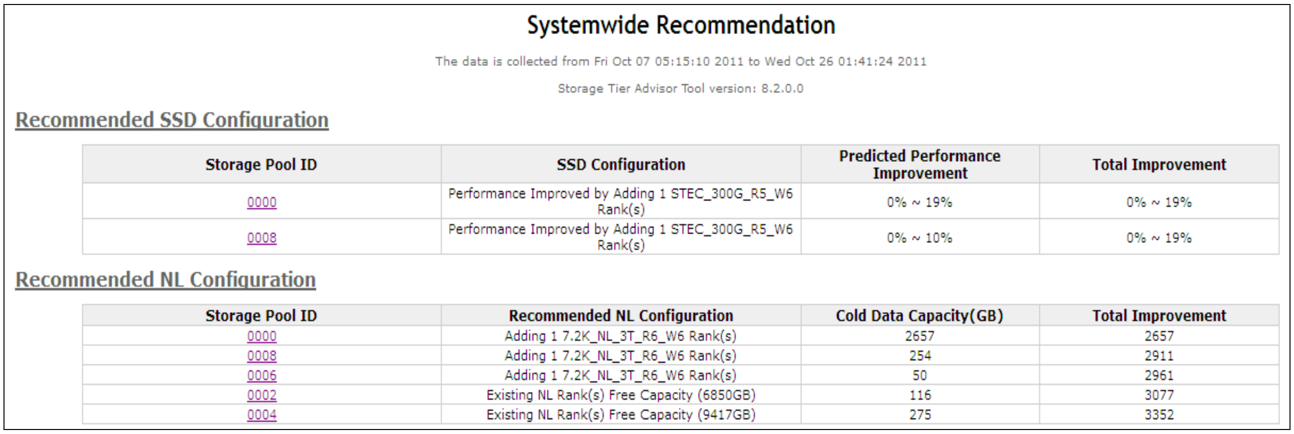


Figure 6-89 System Summary for a DS8800

The STAT provides information about the amount of hot data, the length of time that it takes to apply automatic tiering, and suggestions about the amount of SSD capacity (higher tier) that can be added beneficially. Also, the STAT provides information about the nearline capacity that can be added to save costs when a large amount of cold data is detected.

Figure 6-89 shows the first output view of a STAT analysis, which is the System Summary. This report provides information about all the extent pools, how many tiers make up each extent pool, and the total amount of data monitored, and hot data.

When the STAT detects a significant amount of cold data, the STAT provides suggestions (Figure 6-90 on page 233) to add SSDs and nearline drives to a DS8000 to cold-demote data that is infrequently accessed onto slower and less costly storage space.



Systemwide Recommendation

The data is collected from Fri Oct 07 05:15:10 2011 to Wed Oct 26 01:41:24 2011

Storage Tier Advisor Tool version: 8.2.0.0

Recommended SSD Configuration

Storage Pool ID	SSD Configuration	Predicted Performance Improvement	Total Improvement
0000	Performance Improved by Adding 1 STEC_300G_R5_W6 Rank(s)	0% ~ 19%	0% ~ 19%
0008	Performance Improved by Adding 1 STEC_300G_R5_W6 Rank(s)	0% ~ 10%	0% ~ 19%

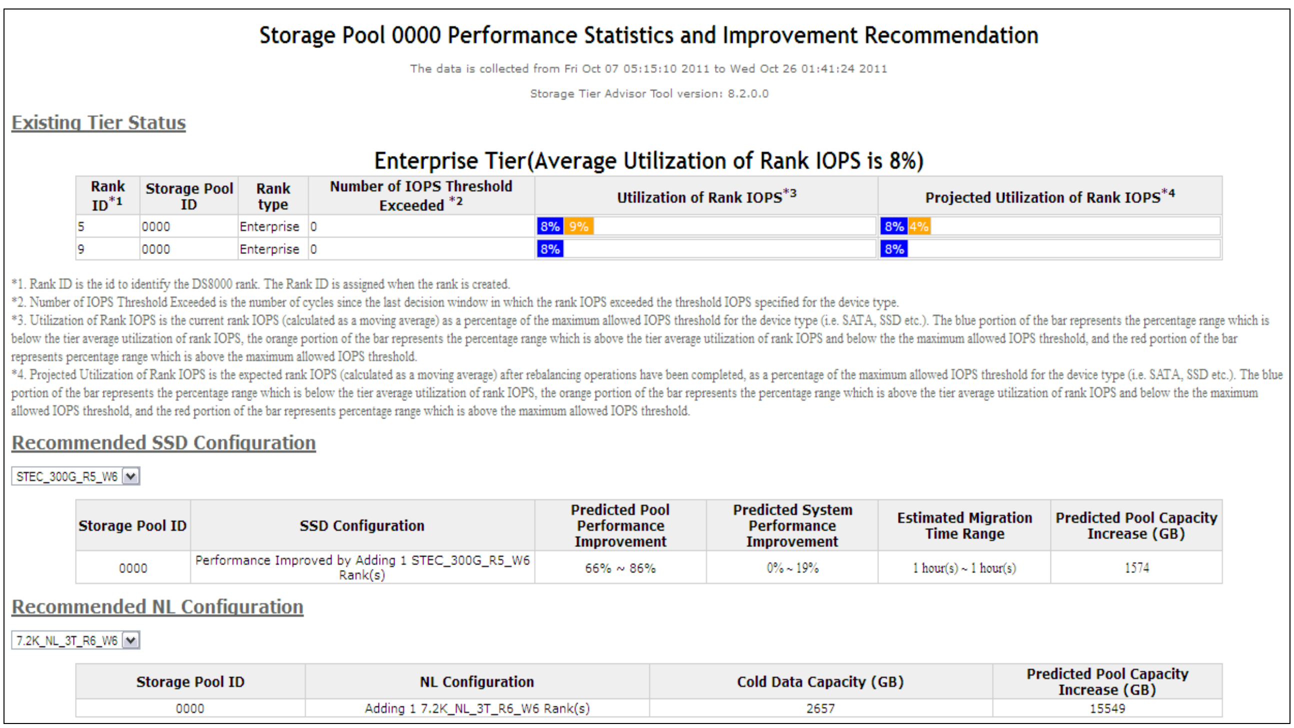
Recommended NL Configuration

Storage Pool ID	Recommended NL Configuration	Cold Data Capacity(GB)	Total Improvement
0000	Adding 1 7.2K_NL_3T_R6_W6 Rank(s)	2657	2657
0008	Adding 1 7.2K_NL_3T_R6_W6 Rank(s)	254	2911
0006	Adding 1 7.2K_NL_3T_R6_W6 Rank(s)	50	2961
0002	Existing NL Rank(s) Free Capacity (6850GB)	116	3077
0004	Existing NL Rank(s) Free Capacity (9417GB)	275	3352

Figure 6-90 Systemwide Recommendation report that shows possible improvements for all pools

When your system consists of a limited number of available SSD ranks, or nearline ranks, the STAT recommendation shows you which pool might benefit most by adding those ranks to existing Enterprise HDD ranks.

The next series of views is by extent pool. When you click a certain storage pool, you can see additional and detailed recommendations for improvements at the level of each extent pool. See Figure 6-91.



Storage Pool 0000 Performance Statistics and Improvement Recommendation

The data is collected from Fri Oct 07 05:15:10 2011 to Wed Oct 26 01:41:24 2011

Storage Tier Advisor Tool version: 8.2.0.0

Existing Tier Status

Enterprise Tier(Average Utilization of Rank IOPS is 8%)

Rank ID *1	Storage Pool ID	Rank type	Number of IOPS Threshold Exceeded *2	Utilization of Rank IOPS *3	Projected Utilization of Rank IOPS *4
5	0000	Enterprise	0	8% 9%	8% 4%
9	0000	Enterprise	0	8%	8%

*1. Rank ID is the id to identify the DS8000 rank. The Rank ID is assigned when the rank is created.
*2. Number of IOPS Threshold Exceeded is the number of cycles since the last decision window in which the rank IOPS exceeded the threshold IOPS specified for the device type.
*3. Utilization of Rank IOPS is the current rank IOPS (calculated as a moving average) as a percentage of the maximum allowed IOPS threshold for the device type (i.e. SATA, SSD etc.). The blue portion of the bar represents the percentage range which is below the tier average utilization of rank IOPS, the orange portion of the bar represents the percentage range which is above the tier average utilization of rank IOPS and below the the maximum allowed IOPS threshold, and the red portion of the bar represents percentage range which is above the maximum allowed IOPS threshold.
*4. Projected Utilization of Rank IOPS is the expected rank IOPS (calculated as a moving average) after rebalancing operations have been completed, as a percentage of the maximum allowed IOPS threshold for the device type (i.e. SATA, SSD etc.). The blue portion of the bar represents the percentage range which is below the tier average utilization of rank IOPS, the orange portion of the bar represents the percentage range which is above the tier average utilization of rank IOPS and below the the maximum allowed IOPS threshold, and the red portion of the bar represents percentage range which is above the maximum allowed IOPS threshold.

Recommended SSD Configuration

STEC_300G_R5_W6

Storage Pool ID	SSD Configuration	Predicted Pool Performance Improvement	Predicted System Performance Improvement	Estimated Migration Time Range	Predicted Pool Capacity Increase (GB)
0000	Performance Improved by Adding 1 STEC_300G_R5_W6 Rank(s)	66% ~ 86%	0% ~ 19%	1 hour(s) ~ 1 hour(s)	1574

Recommended NL Configuration

7.2K_NL_3T_R6_W6

Storage Pool ID	NL Configuration	Cold Data Capacity (GB)	Predicted Pool Capacity Increase (GB)
0000	Adding 1 7.2K_NL_3T_R6_W6 Rank(s)	2657	15549

Figure 6-91 Report that shows that storage pool 0000 needs SSDs and skew

Figure 6-91 shows an extent pool view, for a pool that currently consists of two Enterprise (15K/10K) HDD ranks, with both hot and cold extents. This pool can benefit from adding one solid-state drive (SSD) rank and one nearline rank. You can select the types of drives that you want to add for a certain tier through the small pull-down menus on the left side. These menus contain all the drive and RAID types for a certain type of tier. For instance, when adding more Enterprise drives is suggested, the STAT can calculate the benefit of adding drives in

RAID 10 instead of RAID 5. Or, the STAT can calculate the benefit of using additional 900 GB/10K HDDs instead of the 300 GB/15K drives.

If adding multiple ranks of a certain tier is beneficial for a certain pool, the STAT modeling offers improvement predictions for the expected performance gains when adding two, three, or more ranks up to the recommended number.

Another view within each pool is the Volume Heat Distribution report. Figure 6-92 shows all the volumes of a certain pool. For each volume, the view shows the amount of capacity that is allocated to each tier and the distribution within each tier among hot, warm, and cold data.

Volume Heat Distribution

Volume Heat Distribution

Volume ID *5	Configured Size *6	Tier	Capacity on Tier *7	Heat Distribution *8
0x80c0	100G	SSD Tier	44G	44G
		Enterprise Tier	29G	15G 14G
		NL Tier	27G	27G
0x80c1	100G	SSD Tier	49G	44G 5G
		Enterprise Tier	7G	4G 3G
		NL Tier	44G	37G 7G
0x80c2	100G	SSD Tier	38G	37G 1G
		Enterprise Tier	10G	5G 5G
		NL Tier	52G	42G 10G
0x80c3	100G	SSD Tier	38G	38G
		Enterprise Tier	18G	9G 9G
		NL Tier	44G	43G 1G
0x80c4	100G	SSD Tier	61G	3G 58G
		Enterprise Tier	0G	
		NL Tier	39G	39G
0x80c5	100G	SSD Tier	2G	2G
		Enterprise Tier	0G	
		NL Tier	98G	98G
0x80c6	100G	SSD Tier	8G	8G
		Enterprise Tier	0G	
		NL Tier	92G	92G
0x80c7	100G	SSD Tier	5G	5G
		Enterprise Tier	0G	
		NL Tier	95G	95G
0x80d0	100G	SSD Tier	73G	73G
		Enterprise Tier	0G	
		NL Tier	27G	27G

Figure 6-92   Storage pool statistics and recommendations: Volume Heat Distribution view

In this view, three heat classes are visible externally. Internally, however, DS8000 Easy Tier monitoring uses a more granular extent temperature in heat buckets. This detailed Easy Tier data can be retrieved by IBM support for extended studies of a client workload situation.

7

Practical performance management

This chapter describes the tools, data, and activities available for supporting the DS8000 performance management processes.

In this chapter, we present these topics:

- Introduction to practical performance management
- Performance management tools
- Tivoli Storage Productivity Center data collection
- Key performance metrics
- Tivoli Storage Productivity Center reporting options
- Monitoring performance of a SAN switch or director
- End-to-end analysis of I/O performance problems
- Tivoli Storage Productivity Center for Disk in mixed environments

7.1 Introduction to practical performance management

In Appendix A, “Performance management process” on page 631, we describe performance management processes and inputs, actors, and roles. Performance management processes include operational processes, such as data collection and alerting, tactical processes, such as performance problem determination and analysis, and strategic processes, such as long term trending. This chapter defines the tools, metrics, and processes required to support the operational, tactical, and strategic performance management processes.

7.2 Performance management tools

Tools for collecting, monitoring, and reporting on the DS8000 performance are critical to the performance management processes. At the time of the writing of this book, the storage resource management tool with the most DS8000 performance management capabilities is IBM Tivoli Storage Productivity Center for Disk. Tivoli Storage Productivity Center for Disk provides support for the DS8000 performance management processes with the following features, which are shown in Table 7-1, that are described in greater detail in the remainder of this chapter.

Table 7-1 Tivoli Storage Productivity Center supported activities for performance processes

Process	Activities	Feature
Operational	Performance data collection for port, array, volume, and switch metrics	Performance monitor jobs that use Native API (NAPI)
Operational	Alerting	Alerts and constraint violations
Tactical/ Strategic	Performance reporting of port, array, volume, and switch	Predefined, ad hoc, Batch, TPCTOOL, and BIRT
Tactical	Performance analysis and tuning	Tool facilitates through data collection and reporting
Tactical	Short term trending	GUI charting facilitates trend lines, and reporting options facilitate export to analytical tools
Tactical	Workload profiling	Volume Planner
Strategic	Long term trending	GUI charting facilitates trend lines, and reporting options facilitate export to analytical tools

Certain features that are required to support the performance management processes are not provided in Tivoli Storage Productivity Center. These features are shown in Table 7-2 on page 237.

Table 7-2 Other tools required

Process	Activity	Alternative
Strategic	Sizing	Disk Magic, general rules, see Chapter 6, “Performance planning tools” on page 175.
Strategic	Planning	Logical configuration performance considerations, see Chapter 4, “Logical configuration performance considerations” on page 87.
Operational	Host data collection performance and alerting	Native host tools. See the OS chapters for more detail.
Tactical	Host performance analysis and tuning	Native host tools. See the OS chapters for more detail.

7.2.1 Tivoli Storage Productivity Center overview

IBM Tivoli Storage Productivity Center Standard Edition (SE) is designed to reduce the complexity of managing SAN storage devices by allowing administrators to configure, manage, and monitor storage devices and switches from a single console. Tivoli Storage Productivity Center includes the following components for managing your storage infrastructure:

- Tivoli Storage Productivity Center for Disk/Disk Manager

  This version of Tivoli Storage Productivity Center provides the critical elements needed to provide storage performance management. Tivoli Storage Productivity Center enables the features to collect, analyze, and report on storage devices configured to Tivoli Storage Productivity Center.

- Tivoli Storage Productivity Center for Data/Data Manager

  This version of Tivoli Storage Productivity Center supports the asset capacity, Storage Resource Management (SRM) along with many of the server reporting features available within Tivoli Storage Productivity Center. Tivoli Storage Productivity Center for Data does not provide any performance management capabilities.

- Tivoli Storage Productivity Center Basic Edition/Fabric Manager

  This version of Tivoli Storage Productivity Center provides volume management that includes discovery, provisioning, and basic reporting for all storage devices supported by Tivoli Storage Productivity Center. In addition, Tivoli Storage Productivity Center Basic Edition provides fabric management that includes in-band and out-of-band discovery, alert notification, and SAN zoning of volumes provisioned for all fabric vendors supported by Tivoli Storage Productivity Center.

For a full list of the features that are provided in each of the Tivoli Storage Productivity Center SE components, visit the IBM website:

http://www.ibm.com/systems/storage/software/center/standard/index.html

7.2.2 Tivoli Storage Productivity Center data collection

With many disk systems, Tivoli Storage Productivity Center for Disk uses the Common Information Model (CIM) Agent as an interface to the disk subsystem. Tivoli Storage Productivity Center uses the CIM Agent for discovery, probes, and performance monitoring. For the DS8000, SAN Volume Controller, and IBM XIV, starting with Tivoli Storage Productivity Center 4.2, Tivoli Storage Productivity Center does not use the CIM for these products anymore, but instead uses the Native APIs (NAPI) of these storage systems, to achieve faster detection and handling speeds and get access to more internal counters. We partially continue describing the CIM, because it explains the main principles of data collection for this book. Tivoli Storage Productivity Center also gets CIM indications from the CIM Agent. In past, we used Common Information Model Object Manager (CIMOM) and Common Information Model (CIM) agent interchangeably. Figure 7-1 shows a high-level design overview of a Storage Management Initiative Specification (SMI-S) Tivoli Storage Productivity Center implementation in older versions.

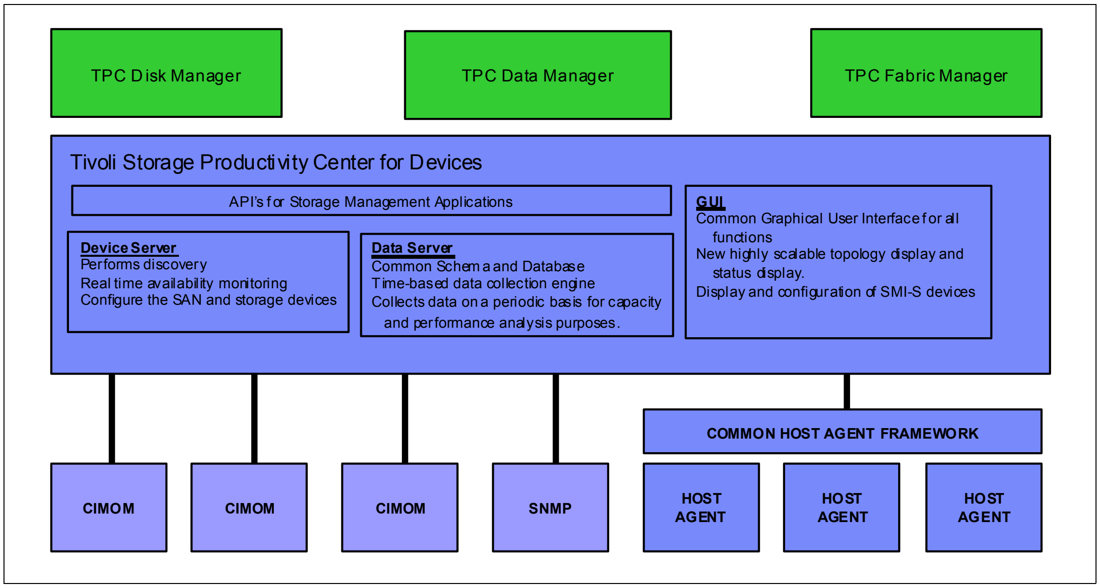


Figure 7-1 Tivoli Storage Productivity Center: SMI-S design overview

Tivoli Storage Productivity Center V4.2 provides a new access method to gather information from devices. This method is called the Native API (NAPI) and is at this time available for only a limited number of disk storage systems.

With the introduction of Native API, another architectural change is introduced: the External Process Manager (EPM). This process manager is the link between the devices used by NAPI and Tivoli Storage Productivity Center. It is called External Process Manager, because now the jobs for the NAPI devices are started as external processes in the operating system, and are no longer running as threads within the Device server process. The advantage is that the scalability and reliability is increased. Figure 7-2 on page 239 shows the high-level architecture of the EPM. You can see that the EPM starts external processes for each type of device and each type of job.

For the devices we listed, Tivoli Storage Productivity Center V4.2 uses only the Native API. When you upgrade to Tivoli Storage Productivity Center V4.2, an update/migration is required to switch to the NAPI, which can be done before or during the installation (or even later, but you cannot the use the device until you complete the migration). For that reason, the Supported Storage Products Matrix does not list any provider versions or Interop Namespace for the IBM supported devices that are listed. In addition to this new interface, the device server is modified, so that together with the NAPI, the scalability and reliability are enhanced. Tivoli Storage Productivity Center is still not trying to replace the management tools for those devices, but at the same time, clients asked for better integration of IBM devices. As an example, for the DS8000, specifying the logical subsystem (LSS) when provisioning volumes was not possible. It is now possible with Tivoli Storage Productivity Center V4.2.

The Storage Management Initiative (SMI) standard does not include this level of detail, because the intention of SMI-S is too abstract from the actual hardware devices.

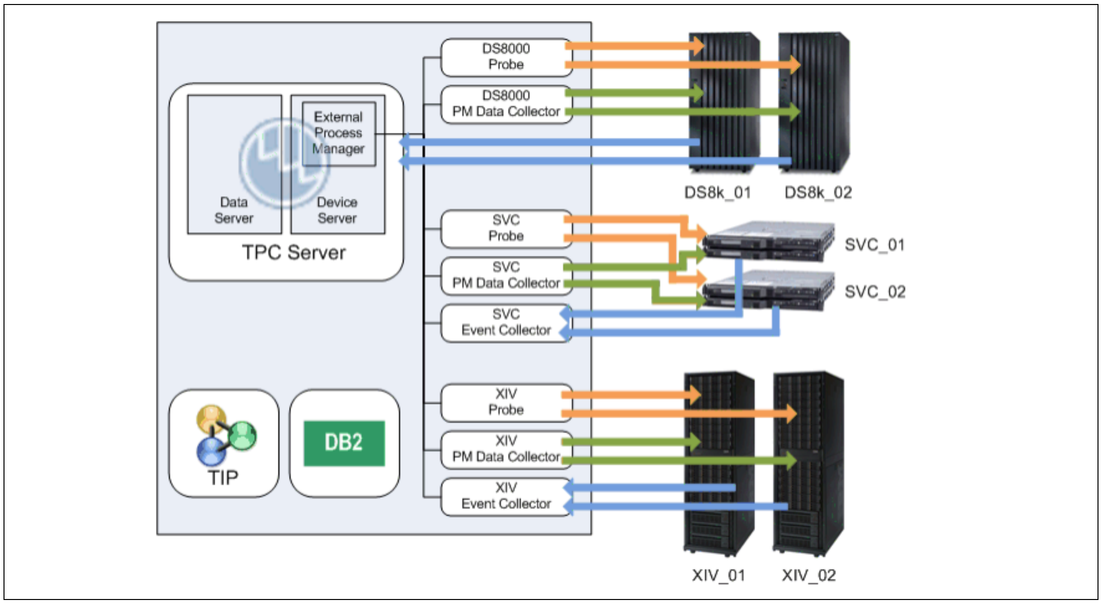


Figure 7-2 External Process Manager (EPM)

There are many components in a Tivoli Storage Productivity Center environment. An example of the complexity of a Tivoli Storage Productivity Center environment is provided in Figure 7-3.

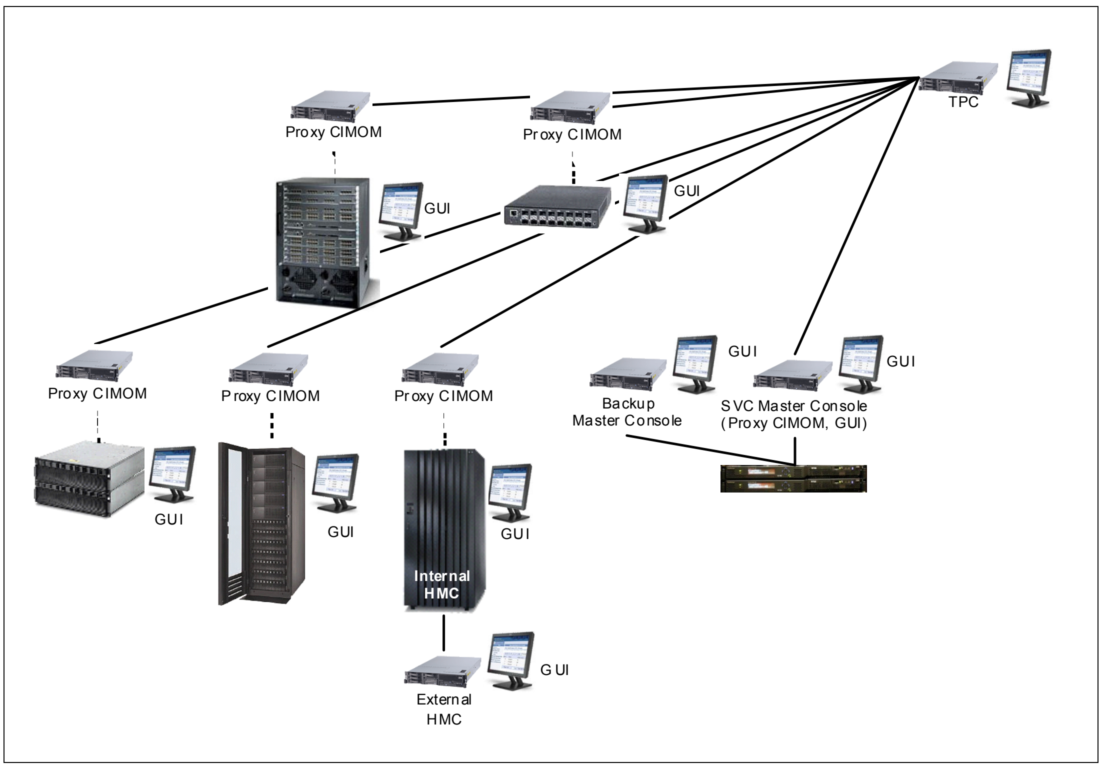


Figure 7-3 Tivoli Storage Productivity Center physical layout

This architecture was designed by the Storage Networking Industry Association (SNIA), an industry workgroup. The architecture is not simple, but it is “open,” which means that any company can use SMI-S standard CIMOMs to manage and monitor storage and switches.

For additional information about the configuration and deployment of Tivoli Storage Productivity Center, see these publications:

- *Tivoli Storage Productivity Center 4.2 Release Update,* SG24-7894
- *SAN Storage Performance Management Using Tivoli Storage Productivity Center*, SG24-7364

7.2.3 Tivoli Storage Productivity Center measurement of DS8000 components

To gather performance data, you first need to set up a job that is called a *Subsystem Performance Monitor* after you successfully discover and probe your subsystems. When this job starts, Tivoli Storage Productivity Center uses NAPI to start collecting data. Later, Tivoli Storage Productivity Center regularly queries the performance counters on that NAPI for a specific device and stores the information in its database as a sample. After two samples are in the database, Tivoli Storage Productivity Center can calculate the differences between the counters for that interval. It uses the differences to derive metrics, such as IOPS and average response times, but also advanced metrics that are not natively provided by the DS8000 system, such as disk utilization. After calculating the metrics from the gathered data, Tivoli Storage Productivity Center provides various metrics to display the performance data in a meaningful way.

Tivoli Storage Productivity Center can collect the DS8000 data, but it also can collect SAN fabric performance data when using Tivoli Storage Productivity Center Standard Edition. Tivoli Storage Productivity Center can gather information about the component levels, as shown in Figure 7-4 on page 241 for the DS8000 systems. Displaying a metric within Tivoli Storage Productivity Center depends on the ability of the storage subsystem and the NAPI to provide the performance data and related information, such as the values that are assigned to processor complexes. We guide you through the diagram in Figure 7-4 on page 241 by drilling down from the overall subsystem level.

Metrics: A *metric* is a numerical value that is derived from the information that is provided by a device. It is not only the raw data, but a calculated value. For example, the raw data is the transferred bytes, but the metric uses this value and the interval to show the bytes/second.

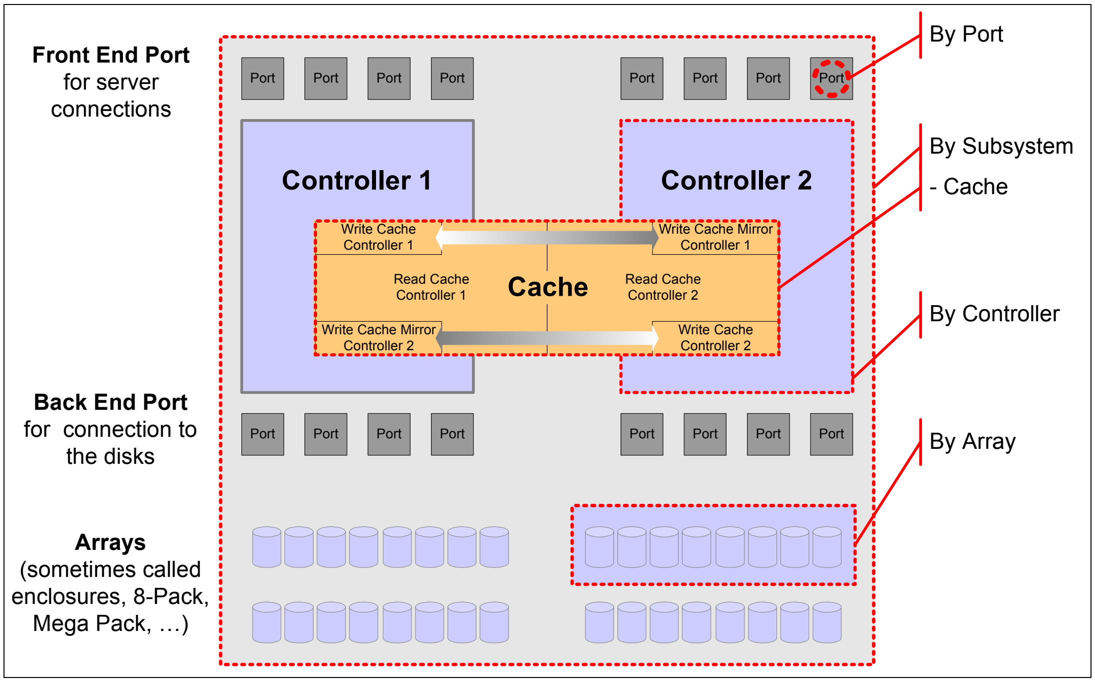


Figure 7-4 Storage device: Physical view

The amount of available information or metrics depends on the type of subsystem involved. The SMI-S standard does not require vendors to provide detailed performance data. For the DS8000, IBM provides extensions to the standard that include much more information than the information that is required by the SMI-S standard. Therefore, the NAPI is used.

The DS8000 interacts with the NAPI in the following ways:

- Access method used: Enterprise Storage Server Network Interface (ESSNI)
- Failover:
  - For the communication with a DS8000, Tivoli Storage Productivity Center uses the ESSNI client. This library is basically the same library that is included in any DS8000 command-line interface (CLI). Because this component has built-in capabilities to fail over from one hardware management console (HMC) to another HMC, a good approach is to specify the secondary HMC IP address if your DS8000 has one.
  - The failover might still cause errors in a Tivoli Storage Productivity Center job, but the next command that is sent to the device uses the redundant connection.
- Network: No special network considerations exist. Tivoli Storage Productivity Center needs to be able to talk to the HMC as it did before the embedded CIMOM was used.
- Tivoli Storage Productivity Center is currently unable to provide specific messages for the vast majority of ESSNI error codes. You can look up the errors in the DS8000 Information Center, which provides useful information (for example, that the user ID is wrong or that the password expired) that is not in any Tivoli Storage Productivity Center logs.

Important: If you migrate a NAPI device either before or as part of the upgrade to Tivoli Storage Productivity Center V4.2, any embedded DS8000 CIMOMs, SAN Volume Controller CIMOMs, and XIV CIMOMs are automatically deleted from Tivoli Storage Productivity Center. Proxy DS CIMOMs are not automatically deleted, even if Tivoli Storage Productivity Center knows of no other devices configured on that CIMOM.

Subsystem

On the subsystem level, metrics are aggregated from multiple records to a single value per metric to give the performance of a storage subsystem from a high-level view, based on the metrics of other components. This aggregation is done by adding values, or calculating average values, depending on the metric.

Cache

The cache in Figure 7-4 on page 241 is a subcomponent of the subsystem, because the cache plays a crucial role in the performance of any storage subsystem. You do not find the cache as a selection in the navigation tree in Tivoli Storage Productivity Center, but there are available metrics that provide information about cache.

Cache metrics for the DS8000 are available in the following report types:

- Subsystem
- Controller
- Array
- Volume

Cache metrics

Metrics, such as disk-to-cache operations, show the number of data transfer operations from disks to cache. The number of data transfer operations from disks to cache is called *staging* for a specific volume. Disk-to-cache operations are directly linked to read activity from hosts. When data is not found in the DS8000 cache, the data is first staged from back-end disks into the cache of the DS8000 server and then transferred to the host.

Read hits occur when all the data requested for a read data access is in cache. The DS8000 improves the performance of read caching by using Sequential Prefetching in Adaptive Replacement Cache (SARC) staging algorithms. For more information about the SARC algorithm, see 1.3.1, “Advanced caching techniques” on page 8. The SARC algorithm seeks to store those data tracks that have the greatest probability of being accessed by a read operation in cache.

The cache-to-disk operation shows the number of data transfer operations from cache to disks, which is called as *destaging* for a specific volume. Cache-to-disk operations are directly linked to write activity from hosts to this volume. Data written is first stored in the persistent memory (also known as nonvolatile storage (NVS)) at the DS8000 server and then destaged to the back-end disk. The DS8000 destaging is enhanced automatically by striping the volume across all the disk drive modules (DDMs) in one or several ranks (depending on your configuration). This striping, or volume management done by Easy Tier, provides automatic load balancing across DDMs in ranks and an elimination of the hot spots.

The Write-cache Delay I/O Rate or Write-cache Delay Percentage due to persistent memory allocation gives us information about the cache usage for write activities. The DS8000 stores data in the persistent memory before sending an acknowledgement to the host. If the persistent memory is full of data (no space available), the host receives a retry for its write request. In parallel, the subsystem must destage the data that is stored in its persistent memory to the back-end disk before accepting new write operations from any host.

If a volume experiences *write operation delayed due to persistent memory constraint* delays, consider moving the volume to a lesser used rank or spread this volume on multiple ranks (increase the number of DDMs used). If this solution does not fix the persistent memory constraint problem, consider adding cache capacity to your DS8000.

Controller

Tivoli Storage Productivity Center refers to the DS8000 processor complexes as *controllers*. A DS8000 has two processor complexes, and each processor complex independently provides major functions for the disk subsystem. Examples include directing host adapters for data transfer to and from host processors, managing cache resources, and directing lower device interfaces for data transfer to and from physical disks. To analyze performance data, you need to know that most volumes can be “assigned/used” by only one controller at a time.

You can use the controller reports to identify if the DS8000 processor complexes are busy and persistent memory is sufficient. Write delays can occur due to write performance limitations on the back-end disk (at the rank level) or limitation of the persistent memory size.

Ports

The port information reflects the performance metrics for the front-end DS8000 ports that connect the DS8000 to the SAN switches or hosts. Additionally, port error rate metrics, such as Error Frame Rate, are also available. The DS8000 host adapter (HA) card has four or eight ports. The SMI-S standards do not reflect this aggregation so Tivoli Storage Productivity Center does not show any group of ports that belong to the same HA. Monitoring and analyzing the ports that belong to the same card are beneficial, because the aggregate throughput is less that the sum of the stated bandwidth of the individual ports. For more information about the DS8000 port cards, see 2.5.1, “Fibre Channel and FICON host adapters” on page 46.

Port metrics: Tivoli Storage Productivity Center reports on many port metrics; therefore, the ports on the DS8000 are the front-end part of the storage device.

Tivoli Storage Productivity Center array reports

The array number that is shown in the Tivoli Storage Productivity Center for Disk *performance* reports, as shown in Figure 7-5 on page 244, directly refers to the array on the DS8000 storage system as listed in the DS GUI or DSCLI. For example, array A10 on a DS8000 system with S/N 75-TV180 and storage image ID IBM.2107-75TV181 is referred to as array 2107.75TV181-A10 in the Tivoli Storage Productivity Center performance reports. However, in the Tivoli Storage Productivity Center storage subsystem reports about the storage subsystem configuration, Tivoli Storage Productivity Center refers to the DS8000 array sites as shown in Figure 7-6 on page 245. For example, the identifier 2107.75TV181-S10 as listed in the Tivoli Storage Productivity Center configuration report refers to the DS8000 array site S10. The DS8000 array numbering starts from A0, A1, A2, and so on. The DS8000 array site numbering starts from S1, S2, S3, and so on.

Volumes on the DS8000 storage systems are primarily associated with an extent pool and an extent pool relates to a set of ranks. To quickly associate all arrays with their related ranks and extent pools, use the output of the DSCLI `lsarray -l` and `lsrank -l` commands, as shown in Example 7-1 on page 245.

Important: In Tivoli Storage Productivity Center releases before Tivoli Storage Productivity Center V4.2, the identifier 2107.*xxxxxxx-yy* in the Tivoli Storage Productivity Center storage subsystem performance reports is called the DS8000 array site S*yy* (starting from S1, S2, and so on) rather than the DS8000 array number (starting from A0, A1, and so on). So, the identifier 2107.75TV181-10 in Tivoli Storage Productivity Center performance reports before Version 4.2 refers to array site S10 instead of array A10. In current Tivoli Storage Productivity Center releases, the identifier uses an 'S' or 'A' to clearly reference a DS8000 array A*yy* or an array site S*yy*.

Certain Tivoli Storage Productivity Center examples and figures in this book might refer to Tivoli Storage Productivity Center identifiers used in versions before Version 4.2 and refer to the DS8000 array sites instead of arrays.

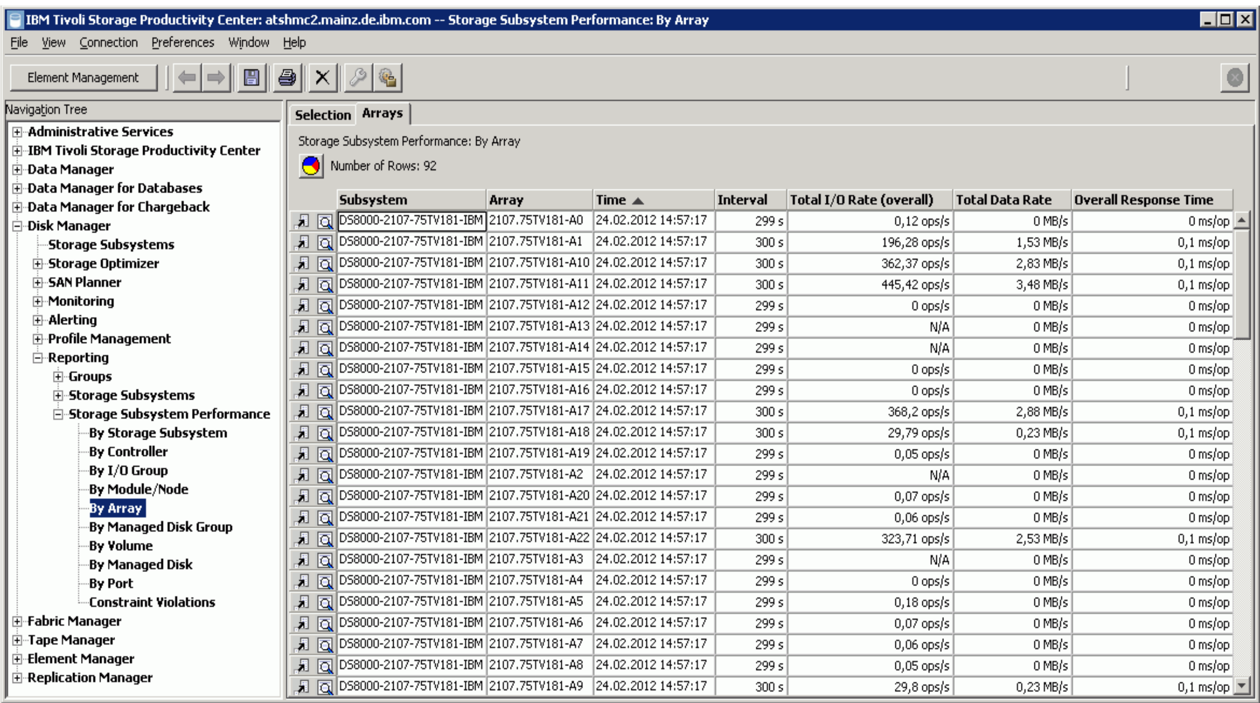


Figure 7-5 TPC 4.2.x storage subsystem performance report by array that shows statistics for DS8000 arrays (Axy)

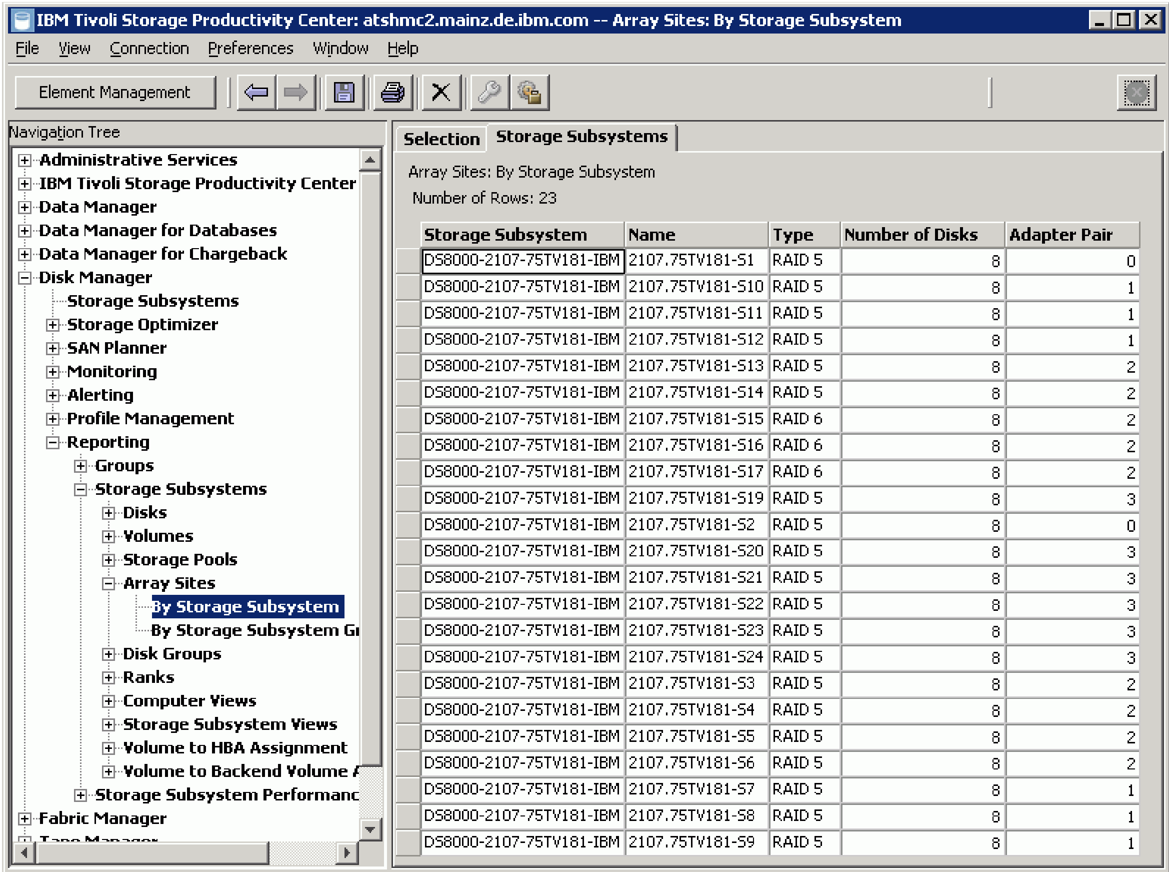


Figure 7-6 TPC 4.2.x storage subsystem report that shows configuration information for DS8000 array sites (Sxy)

A DS8000 array site consists of a group of eight DDMs. A DS8000 *array* is defined on an array site with a specific RAID type. A *rank* is a logical construct to which an array is assigned. A *rank* provides a number of extents that are used to create one or several volumes. A volume can use the DS8000 extents from one or several *ranks*. For more information, see 3.2.1, “Array sites” on page 58, 3.2.2, “Arrays” on page 58, and 3.2.3, “Ranks” on page 59.

Associations: On a configured DS8000 system, there is a 1:1 relationship between an array site, an array, and a rank. However, the numbering sequence typically differs for arrays, ranks, and array sites, for example, array site S16 = array A15 = rank R23.

Example 7-1 shows the relationships among a DS8000 rank, an array, and an array site with a typical divergent numbering scheme by using DSCLI commands. Use the **showrank** command to show which volumes have extents on the specified rank.

Example 7-1 DS8000 array site, array, and rank association

```
dscli> showrank r23
ID             R23
SN             -
Group          1
State          Normal
datastate      Normal
Array          A15
RAIDtype       6
```

```
extpoolID      P5
extpoolnam     rd6_fb_1
volumes        1100,1101,1102,1103,1104,1105
stgtype        fb
exts           12922
usedexts       174
widearrays     0
nararrays      1
trksize        128
strpsize       384
strpesize      0
extsize        16384
encryptgrp     -
migrating(in)  0
migrating(out) 0

dscli> lsrank -l R23
ID  Group State  datastate Array RAIDtype extpoolID extpoolnam stgtype exts  usedexts
=====================================================================================
R23     1 Normal Normal    A15          6 P5        rd6_fb_1   fb      12922      174

dscli> lsarray -l A15
Array State    Data   RAIDtype    arsite Rank DA Pair DDMcap (10^9B) diskclass encrypt
==========================================================================================
A15   Assigned Normal 6 (5+P+Q+S) S16    R23  2              3000.0 NL        unsupported

dscli> lsarraysite S16
arsite DA Pair dkcap (10^9B) State    Array
===========================================
S16    2             3000.0 Assigned A15
```

The Tivoli Storage Productivity Center array performance reports include both front-end and back-end metrics. The back-end metrics are specified by the keyword *Backend*. They provide metrics from the perspective of the controller to the back-end array sites. The front-end metrics relate to the activity between the server and the controller.

There is a relationship between array operations, cache hit ratio, and percentage of read requests. When the cache hit ratio is low, the DS8000 has frequent transfers from DDMs to cache (*staging*).

When the percentage of read requests is high and the cache hit ratio is also high, most of the I/O requests can be satisfied without accessing the DDMs due to the cache management prefetching algorithm.

When the percentage of read requests is low, the DS8000 write activity to the DDMs can be high. The DS8000 has frequent transfers from cache to DDMs (*destaging*).

Comparing the performance of different arrays shows whether the global workload is equally spread on the DDMs of your DS8000. Spreading data across multiple arrays increases the number of DDMs used and optimizes the overall performance.

Important: Back-end write metrics do not include the RAID overhead. In reality, the RAID 5 write penalty adds additional unreported I/O operations.

Tivoli Storage Productivity Center might not be able to calculate some of the volume-based metrics in the array report in multi-rank extent pools, and multi-rank extent pools that contain space-efficient volumes. In this case, the columns for volume-based metrics display the value

N/A in the *Storage Subsystem Performance - By Array* report for the arrays associated with these extent pools.

The following message might be reported by Tivoli Storage Productivity Center in conjunction with multi-rank extent pools:

```
HWNPM2060W The device does not support performance management for segment pool pool ID.
Only incomplete performance data can be collected for array array ID.

Explanation
The specified segment pool contains multiple ranks, which makes it impossible to
accurately manage the performance for those ranks, the arrays associated with those
ranks, and the device adapters associated with those arrays.
For DS6000 and DS8000 devices whenever a segment pool contains multiple ranks, any
volumes allocated in that segment pool might be spread across those ranks in an
unpredictable manner. This makes it impossible to determine the performance impact of
the volumes on the individual ranks. To avoid presenting the user with inaccurate or
misleading performance data, the Performance Manager does not attempt to compute the
performance metrics for the affected arrays and device adapters.
```

Volumes

The *volumes*, which are also called logical unit numbers (LUNs), are shown in Figure 7-7 on page 247. The host server sees the volumes as physical disk drives and treats them as physical disk drives.

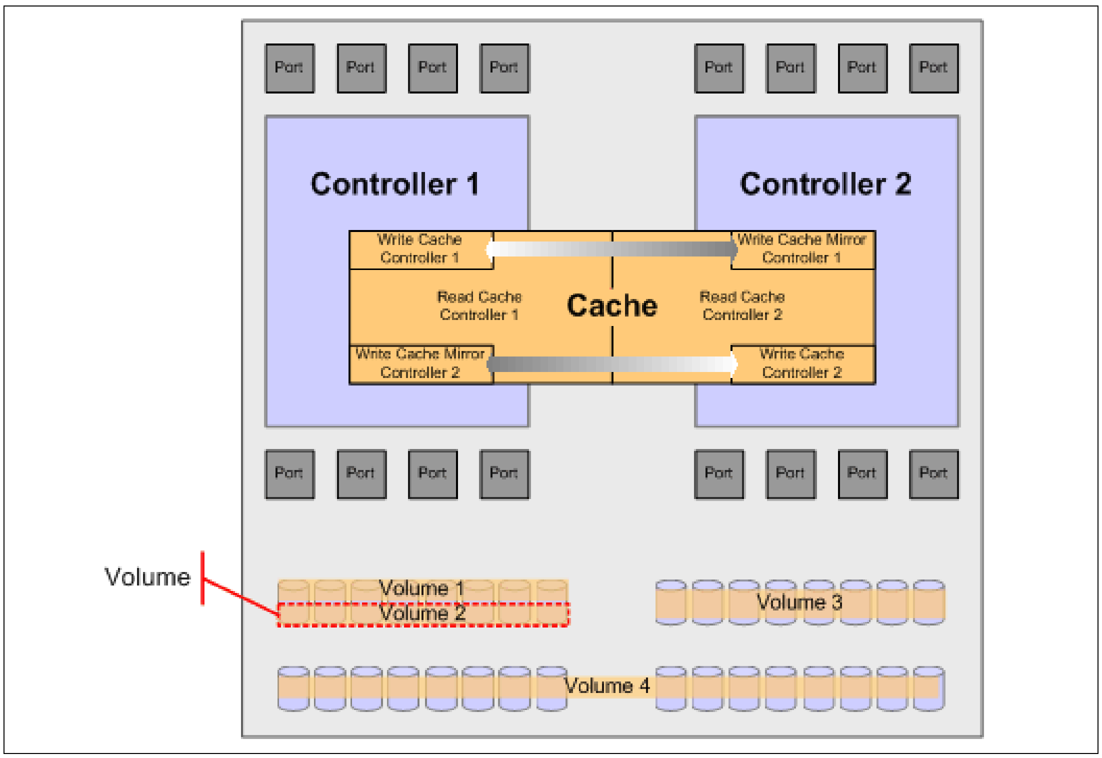


Figure 7-7 DS8000 volume

Analysis of volume data facilitates the understanding of the I/O workload distribution among volumes, as well as workload characteristics (random or sequential and cache hit ratios). A DS8000 volume can belong to one or several ranks, as shown in Figure 7-7 (for more information, see 3.2.7, "Extent allocation methods (EAMs)" on page 68). Especially in managed multi-rank extent pools with Easy Tier automatic data relocation enabled, the distribution of a certain volume across the ranks in the extent pool can change over time. The

STAT with its volume heat distribution report provides additional information about the heat of the data and the data distribution across the tiers within a pool for each volume. For more information about the STAT and its report, see 6.7, “Storage Tier Advisor Tool” on page 231.

The analysis of volume metrics shows the activity of the volumes on your DS8000 and can help you perform these tasks:

- Determine where the most accessed data is located and what performance you get from the volume.
- Understand the type of workload that your application generates (sequential or random and the read or write operation ratio).
- Determine the cache benefits for the read operation (cache management prefetching algorithm SARC).
- Determine cache bottlenecks for write operations.
- Compare the I/O response observed on the DS8000 with the I/O response time observed on the host.

The relationship of certain array sites and ranks to the DS8000 volumes can be derived from the Tivoli Storage Productivity Center configuration report, as shown in Figure 7-8 on page 248. In addition, to quickly associate the DS8000 arrays to array sites and ranks, you might use the output of the DSCLI commands `lsrank -l` and `lsarray -l`, as shown in Example 7-1 on page 245.

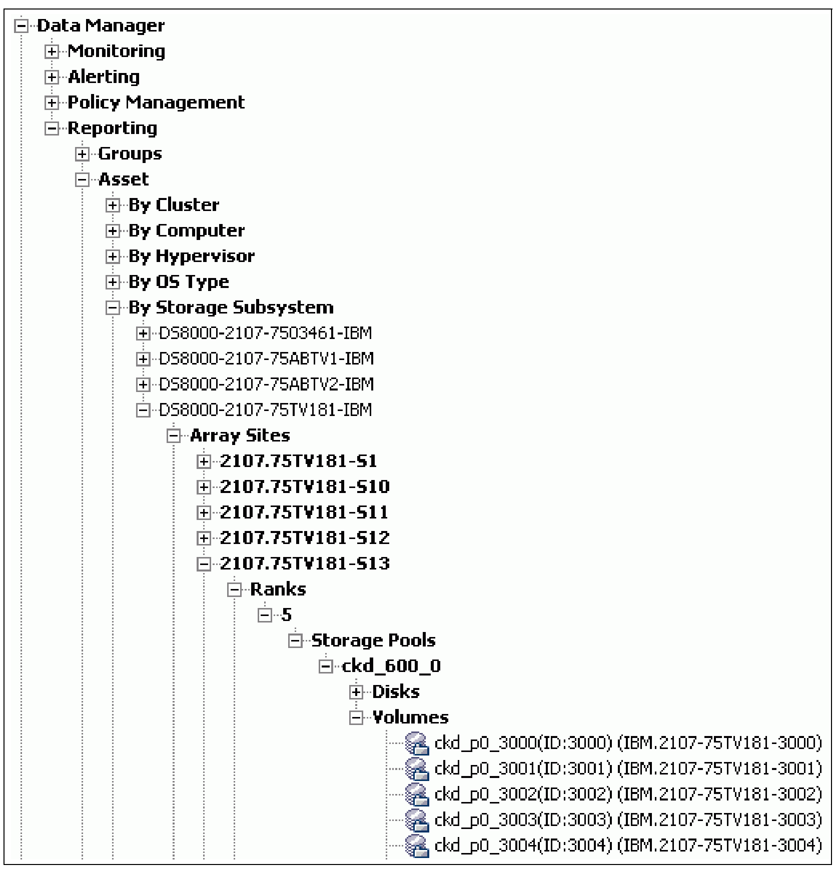


Figure 7-8 Array site to volume breakdown using Tivoli Storage Productivity Center

7.2.4 General Tivoli Storage Productivity Center measurement considerations

To understand the Tivoli Storage Productivity Center measurements of the DS8000 components, it is helpful to understand the context for the measurement. The measurement facilitates insight into the behavior of the DS8000 and its ability to service I/O requests. The DS8000 handles various types of I/O requests differently. Table 7-3 shows the behavior of the DS8000 for various I/O types.

Table 7-3 DS8000 I/O types and behavior

I/O type	DS8000 high-level behavior
Sequential read	Pre-stage reads in cache to increase cache hit ratio.
Random read	Attempt to find data in cache. If not present in cache, read from back end.
Sequential write	Write data to NVS of processor complex owning volume and send copy of data to cache in other processor complex. Upon back-end destaging, perform prefetching of read data and parity into cache to reduce the number of disk operations on the back end.
Random write	Write data to NVS of processor complex owning volume and send copy of data to cache in other processor complex. Destage modified data from NVS to disk as determined by microcode.

Understanding writes to a DS8000

When the DS8000 accepts a write request, it processes it without writing to the DDMs physically. The data is written into both the processor complex to which the volume belongs and the persistent memory of the second processor complex in the DS8000. Later, the DS8000 asynchronously destages the modified data out to the DDMs. In cases where back-end resources are constrained, NVS delays might occur. Tivoli Storage Productivity Center reports on these conditions with the following front-end metrics: Write Cache Delay I/O Rate and Write Cache Delay I/O Percentage.

The DS8000 lower interfaces use switched Fibre Channel (FC) connections, which provide a high data transfer bandwidth. In addition, the destage operation is designed to avoid the write penalty of RAID 5, if possible. For example, there is no write penalty when modified data to be destaged is contiguous enough to fill the unit of a RAID 5 stride. A *stride* is a full RAID 5 stripe. However, when all of the write operations are random across a RAID 5 array, the DS8000 cannot avoid the write penalty.

Understanding reads on a DS8000

If the DS8000 cannot satisfy the read I/O requests within the cache, it transfers data from the DDMs. The DS8000 suspends the I/O request until it reads the data. This situation is called *cache-miss*. If an I/O request is cache-miss, the response time includes the data transfer time between host and cache, and also the time that it takes to read the data from DDMs to cache before sending it to the host. The various read hit ratio metrics show how efficiently cache works on the DS8000.

The read hit ratio depends on the characteristics of data on your DS8000 and applications that use the data. If you have a database and it has a high locality of reference, it shows a high cache hit ratio, because most of the data referenced can remain in the cache. If your database has a low locality of reference, but it has the appropriate sets of indexes, it might also have a high cache hit ratio, because the entire index can remain in the cache.

A database can be *cache-unfriendly* by nature. An example of a cache-unfriendly workload is a workload that consists of large sequential reads to a highly fragmented filesystem. If an application reads this file, the cache hit ratio is low, because the application never reads the same data due to the nature of sequential access. In this case, defragmentation of the filesystem improves the performance. You cannot determine whether increasing the size of cache improves the I/O performance without knowing the characteristics of the data on your DS8000.

We suggest that you monitor the read hit ratio over an extended period of time:

- If the cache hit ratio is historically low, it is most likely due to the nature of the data access patterns. Defragmenting the filesystem and making indexes if none exist might help more than adding cache.
- If you have a high cache hit ratio initially and it decreases as the workload increases, adding cache or moving part of the data to volumes associated with the other processor complex might help.

Interpreting read-to-write ratio

The read-to-write ratio depends on how the application programs issue I/O requests. In general, the overall average read-to-write ratio is in the range of 75% - 80% reads.

For a logical volume that has sequential files, it is key to understand the application types that access those sequential files. Normally, these sequential files are used for either read only or write only at the time of their use. The DS8000 cache management prefetching algorithm (SARC) determines whether the data access pattern is sequential. If the access is sequential, contiguous data is prefetched into cache in anticipation of the next read request.

Tivoli Storage Productivity Center for Disk reports the reads and writes through various metrics. See 7.3, “Tivoli Storage Productivity Center data collection” on page 250 for a description of these metrics in greater detail.

7.3 Tivoli Storage Productivity Center data collection

In this section, we describe the performance data collection considerations, such as time stamps, durations, and intervals.

7.3.1 Time stamps

Tivoli Storage Productivity Center server uses the time stamp of the source devices when it inserts data into the database. If the Tivoli Storage Productivity Center server clock is not synchronized with the rest of your environment, it does not include any additional offset, because you might need to compare the performance data of the DS8000 with the data gathered on a server.

Although the time information of the device is written to the database, reports are always based on the time of the Tivoli Storage Productivity Center server. Tivoli Storage Productivity Center receives the time zone information from the devices (or the NAPIs) and uses this information to adjust the time in the reports to the local time. Certain devices might convert the time into Greenwich mean time (GMT) time stamps and not provide any time zone information.

This complexity is necessary to be able to compare the information from two subsystems in different time zones from a single administration point. This administration point is the GUI, not the Tivoli Storage Productivity Center server. If you open the GUI in different time zones, a performance diagram might show a distinct peak at different times, depending on its local time zone.

When using Tivoli Storage Productivity Center to compare data from a server (for example, iostat data) with the data of the storage subsystem, it is important to know the time stamp of the storage subsystem. Unfortunately, Tivoli Storage Productivity Center does not provide a report to see the time zone information for a device. Most likely because the devices or NAPIs convert the time stamps into GMT time stamps before they are sent.

To ensure that the time stamps on the DS8000 are synchronized with the other infrastructure components, the DS8000 provides features for configuring a Network Time Protocol (NTP) server. To modify the time and configure the hardware management console (HMC) to use an NTP server, the following steps are required:

1. Log on to **HMC**.
2. Select **HMC Management**.
3. Select **Change Date and Time**.
4. A dialog box similar to Figure 7-9 on page 251 opens. Change the Time to match the current time for the time zone.

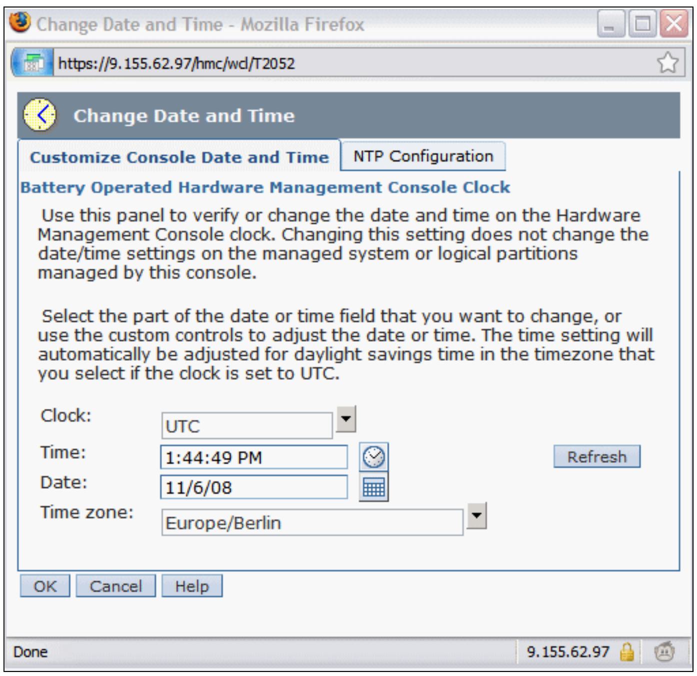


Figure 7-9 DS8000 Change Date and Time

5. To configure an NTP server, select the **NTP Configuration** tab. A dialog box similar to Figure 7-10 opens.

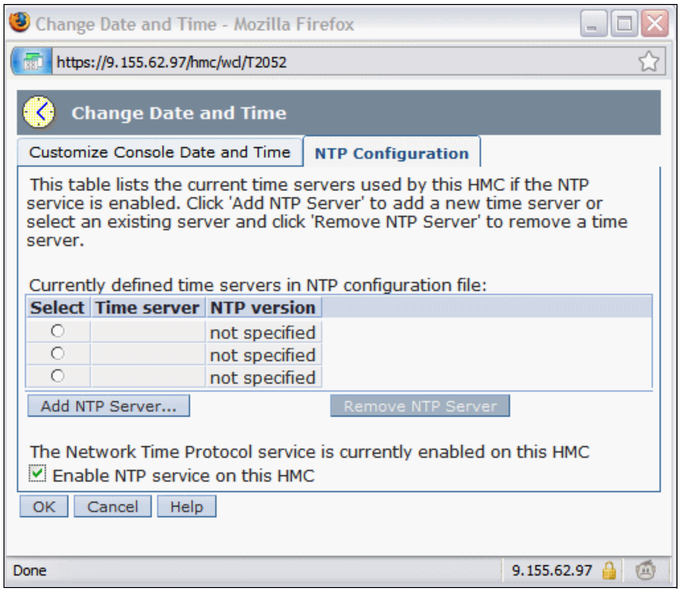


Figure 7-10 DS8000 NTP Configuration tab

6. Select **Add NTP Server** and provide the IP address and the NTP version.
7. Check **Enable NTP service on this HMC** and click **OK**.

> **Reboot:** These configuration changes require a reboot of the HMC. These steps were tested on DS8000 code Version 4.0 and later.

7.3.2 Duration

Tivoli Storage Productivity Center can collect data continuously. From a performance management perspective, collecting data continuously means that performance data exists to facilitate reactive, proactive, and even predictive processes, as described in Chapter 7, “Practical performance management” on page 235. For ongoing performance management of the DS8000, we suggest one of the following approaches to data collection:

- Run continuously. The benefit to this approach is that at least in theory data always exists. The downside is that if a component of Tivoli Storage Productivity Center goes into a bad state, it does not always generate an alert. In these cases, data collection might stop with only a warning, and a Simple Network Management Protocol (SNMP) alert is not generated. In certain cases, the only obvious indication of a problem is a lack of performance data.
- Restart collection every *n* number of hours. In this approach, configure the collection to run for somewhere between 24 - 168 hours. For larger environments, a significant delay period might need to be configured between the last interval and the first interval in the next data collection. The benefit to this approach is that data collection failures result in an alert every time that the job fails. You can configure this alert to go to an operational monitoring tool, such as Tivoli Enterprise Console® or IBM Tivoli Netcool/OMNIbus operations management software. In this case, performance data loss is limited to the configured duration. The downside to this approach is that there is always data that is missing for a period of time as Tivoli Storage Productivity Center starts the data collection on all devices. For large environments, this technique might not be tenable for an interval less than 72 hours, because the start-up costs related to starting the collection on many devices can be significant.

In Table 7-4, we list the advantages and disadvantages of these methods.

Table 7-4 Scheduling considerations

Situation	Scheduled with a duration of *n* hours, repeating every *n* hours	Set to run indefinitely
Job startup behavior	Job tries to connect to any NAPI to which the device is connected. If the connection fails, the job fails, and an alert, if defined, is generated. In any case, the job is scheduled to run after *n* hours, so *n* number of hours is the maximum that you lose.	Job tries to connect to any NAPI to which the device is connected. If the connection fails, the job fails, and an alert, if defined, is generated. If you do not fix the problem and restart the job manually, the job never automatically restarts, even though the problem might be temporary.
NAPI fails after successful start of job	Performance data collection job fails, and an alert, if defined, is generated. You lose up to *n* number of hours of information in addition to the one hour pause, depending on when this failure happens.	The job tries to reconnect to the NAPI within the defined intervals and recover if the communication can be reestablished. But, there might be situations where this recovery does not work. For example, if the NAPI is restarted, it might not be able to resume the performance collection from the device until Tivoli Storage Productivity Center restarts the data collection job.
Data completeness	There is at least a gap of one hour every *n* number of hours.	Data is gathered as completely as possible.
Alerts	You get alerts for every job that fails.	You get an alert only one time.
Manual restart	Manually restarted jobs can cause trouble, because the jobs can easily overlap with the next scheduled job, which prevents the scheduled job from starting.	No problems.
Log files	A log file is created for each scheduled run. The navigation tree shows the status of the current and past jobs, and whether the job was successful in the past.	Usually, you see a only single log file. You see multiple log files only if you stop and restart the job manually.

7.3.3 Intervals

In Tivoli Storage Productivity Center, the data collection interval is referred to as the *sample interval*. The sample interval for the DS8000 performance data collection tasks is from 5 - 60 minutes. A shorter sample interval results in a more granular view of performance data at the expense of requiring additional database space. The appropriate sample interval depends on the objective of the data collection. Table 7-5 displays example data collection objectives and reasonable values for a sample interval.

Table 7-5 Sample interval examples

Objective	Sample interval minutes
Problem determination/service-level agreement (SLA)	5
Ongoing performance management	15
Baseline or capacity planning	60

To reduce the growth of the Tivoli Storage Productivity Center database while watching for potential performance issues, Tivoli Storage Productivity Center can store only samples in which an alerting threshold is reached. This skipping function is useful for SLA reporting and longer term capacity planning.

In support of ongoing performance management, a reasonable sample interval is 15 minutes. An interval of 15 minutes usually provides enough granularity to facilitate reactive performance management. In certain cases, the level of granularity required to identify the performance issue is less than 15 minutes. In these cases, you can reduce the sample interval. Tivoli Storage Productivity Center also provides reporting at higher intervals, including hourly and daily. Tivoli Storage Productivity Center provides these views automatically.

7.4 Key performance metrics

Tivoli Storage Productivity Center for Disk has a significant number of metrics available for reporting the health and performance of the DS8000. The metrics provided can be categorized as either front-end or back-end metrics. Front-end metrics relate to the activity between the server and the storage cache. Back-end metrics relate to the activity between the storage controller and the disk arrays. In Tivoli Storage Productivity Center, back-end statistics are clearly delineated with the key word "*Backend*" and front-end metrics do not contain any keyword. Table 7-6 on page 254 provides a matrix of key DS8000 metrics for each component.

Table 7-6 Tivoli Storage Productivity Center subsystem, controller, port, volume, and array metrics

Key DS8000 metrics	Subsystem	Controller	Port	Volume	Array	Definition
Read I/O Rate (overall)	✓	✓		✓	✓	Average number of read operations per second for the sample interval.
Write I/O Rate (overall)	✓	✓		✓	✓	Average number of write operations per second for the sample interval.
Total I/O Rate (overall)	✓	✓		✓	✓	Average number of read and write operations per second for the sample interval.
Read Cache Hits Percentage (overall)	✓	✓		✓	✓	Percentage of reads during the sample interval that are found in the cache. A storage subsystem-wide target is 50%, although this percentage varies depending on the workload.
Write Cache Hits Percentage (overall)	✓	✓		✓	✓	Percentage of writes handled in cache. This number needs to be 100%

Key DS8000 metrics	Subsystem	Controller	Port	Volume	Array	Definition
Write-cache Delay I/O Rate	✓	✓		✓	✓	The rate of I/Os (writes) that are delayed during the sample interval because of write cache. This rate must be 0.
Read Data Rate	✓	✓		✓	✓	Average read data rate in megabytes per second during the sample interval.
Write Data Rate	✓	✓		✓	✓	Average write data rate in megabytes per second during the sample interval.
Total Data Rate	✓	✓		✓	✓	Average total (read+write) data rate in megabytes per second during the sample interval.
Read Response Time	✓	✓		✓	✓	Average response time in milliseconds for reads during the sample interval. For this report, this metric is an average of read hits in cache, as well as read misses.
Write Response Time	✓	✓		✓	✓	Average response time in milliseconds for writes during the sample interval.
Overall Response Time	✓	✓		✓	✓	Average response time in milliseconds for all I/O in the sample interval, including both cache hits, as well as misses to back-end storage if required.
Read Transfer Size	✓	✓		✓	✓	Average transfer size in kilobytes for reads during the sample interval.
Write Transfer Size	✓	✓		✓	✓	Average transfer size in kilobytes for writes during the sample interval.
Overall Transfer Size	✓	✓		✓	✓	Average transfer size in kilobytes for all I/O during the sample interval.
Backend Read I/O Rate	✓	✓			✓	The average read rate in reads per second caused by read misses. This rate is the read rate to the back-end storage for the sample interval.
Backend Write I/O Rate	✓	✓			✓	The average write rate in writes per second caused by front-end write activity. This rate is the write rate to the back-end storage for the sample interval. These writes are logical writes, and the actual number of physical I/O operations depends on the type of RAID architecture.
Total Backend I/O Rate	✓	✓			✓	The average write rate in writes per second caused by front-end write activity. This rate is the write rate to the back-end storage for the sample interval. These writes are logical writes and the actual number of physical I/O operations depends on the type of RAID architecture.
Backend Read Data Rate	✓	✓			✓	Average number of megabytes per second read from back-end storage during the sample interval.

Key DS8000 metrics	Subsystem	Controller	Port	Volume	Array	Definition
Backend Write Data Rate	✓	✓			✓	Average number of megabytes per second written to back-end storage during the sample interval.
Total Backend Data Rate	✓	✓			✓	Sum of the Backend Read and Write Data Rates for the sample interval.
Backend Read Response Time	✓	✓			✓	Average response time in milliseconds for read operations to the back-end storage.
Backend Write Response Time	✓	✓			✓	Average response time in milliseconds for write operations to the back-end storage. This time might include several physical I/O operations, depending on the type of RAID architecture.
Overall Backend Response Time	✓	✓			✓	The weighted average of Backend Read and Write Response Times during the sample interval.
Disk Utilization Percentage					✓	Average disk utilization during the sample interval. This percentage is also the utilization of the RAID array, because the activity is uniform across the array.
Port Send I/O Rate	✓		✓			The average rate per second for operations that send data from an I/O port, typically to a server. This operation is typically a read from the server perspective.
Port Receive I/O Rate	✓		✓			The average rate per second for operations where the storage port receives data, typically from a server. This operation is typically a write from the server perspective.
Total Port I/O Rate	✓		✓			Average read plus write I/O rate per second at the storage port during the sample interval.
Port Send Data Rate	✓		✓			The average data rate in megabytes per second for operations that send data from an I/O port, typically to a server.
Port Receive Data Rate	✓		✓			The average data rate in megabytes per second for operations where the storage port receives data, typically from a server.
Total Port Data Rate	✓		✓			Average (read+write) data rate in megabytes per second at the storage port during the sample interval.
Port Send Response Time	✓		✓			Average number of milliseconds that it took to service each port send (server read) operation for a particular port over the sample interval.
Port Receive Response Time	✓		✓			Average number of milliseconds that it took to service each port receive (server write) operation for a particular port over the sample interval.

Key DS8000 metrics	Subsystem	Controller	Port	Volume	Array	Definition
Total Port Response Time	✓		✓			Weighted average port send and port receive time over the sample interval.
Port Send Transfer Size	✓		✓			Average size in kilobytes per Port Send operation during the sample interval.
Port Receive Transfer Size	✓		✓			Average size in kilobytes per Port Receive operation during the sample interval.
Total Port Transfer Size	✓		✓			Average size in kilobytes per port transfer during the sample interval.

Performance Metrics in TotalStorage Productivity Center Performance Reports, REDP-4347, provides a list of Tivoli Storage Productivity Center metrics for the DS8000:

http://www.redbooks.ibm.com/abstracts/redp4347.html?Open&pdfbookmark

7.4.1 DS8000 key performance indicator thresholds

As seen in Table 7-6 on page 254, Tivoli Storage Productivity Center for Disk provides an overwhelming number of metrics for performance management. In this section, we provide additional information about a subset of critical metrics. We provide suggested threshold values as general rules for the constraint of various DS8000 components. As with any rules, you must adjust them for the performance requirements of your environment.

Colors are used to distinguish the components.

Table 7-7 DS8000 key performance indicator thresholds

Component	Metric	Threshold	Comment
Controller	Cache Holding Time	< 60	Indicates high cache track turnover and possibly cache constraint.
Controller	Write Cache Delay I/O Rate	>1	Indicates writes delayed due to insufficient memory resources.
Array	Disk Utilization Percentage	> 50%	Indicates disk saturation.
Array	Write I/O Rate (overall)	> 250	RAID 5 with 4 operations per write indicates saturated array.
Array	Total I/O Rate (overall)	> 1000	Even if all I/Os are reads, this metric indicates busy disks.
Array	Overall Backend Response Time	> 35	Indicates busy disks.
Array	Backend Write Response Time	> 35	Indicates busy disks.
Array	Backend Read Response Time	> 35	Indicates busy disks.

Component	Metric	Threshold	Comment
Volume	Read Cache Hits Percentage (overall)	> 90	Look for opportunities to move volume data to application or database cache.
Volume	Write Cache Hits Percentage (overall)	< 100	Cache misses can indicate busy back end or need for additional cache.
Volume	Read I/O Rate (Overall)	N/A	Look for high rates.
Volume	Write I/O Rate (overall)	N/A	Look for high rates.
Volume	Read Response Time	> 20	Indicates disk or port contention.
Volume	Write Response Time	> 5	Indicates cache misses, busy back end, and possible front-end contention.
Volume	Write Cache Delay I/O Rate	> 1	Cache misses can indicate busy back end or need for additional cache.
Volume	Read Transfer Size	> 100	Indicates throughput intensive workload.
Volume	Write Transfer Size	> 100	Indicates throughput intensive workload.
Port	Total Port I/O Rate	> 2500	Indicates transaction intensive load.
Port	Total Port Data Rate	~= 2/4 Gb	If port data rate is close to bandwidth, this rate indicates a saturation.
Port	Port Send Response Time	> 20	Indicates contention on I/O path from the DS8000 to host.
Port	Port Receive Response Time	> 20	Indicates potential issue on I/O path or the DS8000 back end.
Port	Total Port Response Time	> 20	Indicates potential issue on I/O path or the DS8000 back end.

7.5 Tivoli Storage Productivity Center reporting options

Tivoli Storage Productivity Center provides numerous ways of reporting about the DS8000 performance data. In this section, we provide an overview of the various options and their appropriate usage in ongoing performance management of a DS8000 (Table 7-8). The categories of usage are based on definitions in “Tactical performance subprocess” on page 639.

Table 7-8 Report category, usage, and considerations

Report type	Performance process	Advantages	Disadvantages
Alerting/Constraints	Operational	Facilitates operational reporting for certain failure conditions and threshold exception reporting for support of SLAs and service-level objectives (SLOs)	Requires thorough understanding of workload to configure appropriate thresholds
Predefined performance reports	Tactical	Ease of use Top 10 reports provide method for quickly viewing entire environment	Limited metrics Lacks scheduling Inflexible charting Limited to 2500 rows displayed
Ad hoc reports	Tactical, Strategic	Ease of use Flexible	Can export only multiple metrics of same data type at a time Lack of scheduling Inflexible charting Limited to 2500 rows displayed
Batch reports	Tactical, Strategic	Ease of use Ability to export all metrics available Schedule Drill downs with preestablished relationships	Time stamps are in AM/PM format Volume data does not contain array correlation No charting
TPCTOOL	Tactical, Strategic	Flexible Programmable	Non-intuitive Output to flat files that must be post-processed in spreadsheet or other reporting tool
Custom Reports (SQL, BIRT, and ODBC,)	Tactical, Strategic	Highly customizable	Requires some DB and reporting skills
Analytical	Tactical, Strategic	Easy to use Detailed	Does not take into account future or potential changes to the environment

All of the reports use the metrics available for the DS8000 as described in Table 7-6 on page 254. In the remainder of this section, we describe each of the report types in detail.

The STAT can also provide additional performance information for your DS8800/DS8700 system based on the DS8000 Easy Tier performance statistics collection by revealing data heat information at the system and volume level in addition to configuration recommendations. For more information about the STAT, see 6.7, “Storage Tier Advisor Tool” on page 231.

The next topics include brief descriptions of new Tivoli Storage Productivity Center features, such as SAN Planner, Storage Optimizer, and Tivoli Storage Productivity Center Reporter for Disk.

7.5.1 Alerts and constraints

Tivoli Storage Productivity Center provides support for the performance management operational subprocesses through performance alerts and constraint violations. In this section, we describe the difference between the alerts and constraint violations and how to implement them.

Tivoli Storage Productivity Center is not an online performance monitoring tool. However, it uses the term *performance monitor* for the name of the job that is set up to gather data from a subsystem. The performance monitor is a performance data collection task. Tivoli Storage Productivity Center collects information at certain intervals and stores the data in its database. After inserting the data, the data is available for analysis by using several methods that we describe in this section. Because the intervals are usually 5 - 15 minutes, Tivoli Storage Productivity Center is not an online or real-time monitor.

You can use Tivoli Storage Productivity Center to define performance-related alerts that can trigger an event when the defined thresholds are reached. Even though Tivoli Storage Productivity Center works in a similar manner to a monitor without user intervention, the actions are still performed at the intervals specified during the definition of the performance monitor job.

Before describing alerts, we must clarify the terminology.

Alerts

Generally, *alerts* are the notifications defined for different jobs. Tivoli Storage Productivity Center creates an alert on certain conditions, for example, when a probe or scan fails. There are various ways to be notified. Simple Network Management Protocol (SNMP) traps, IBM Tivoli Enterprise Console or IBM Tivoli Netcool® OMNIbus events, and email are the most common methods.

All the alerts are stored in the Alert Log, even if notification is not set up yet. This log is in the navigation tree at **IBM Tivoli Storage Productivity Center** → **Alerting** → **Alert Log**.

In addition to the alerts that you set up when you define a certain job, you can also define alerts that are not directly related to a job, but instead to specific conditions, such as a new subsystem that is discovered. This type of alert is defined in **Disk Manager** → **Alerting** → **Storage Subsystem Alerts**.

These types of alerts are either condition-based or threshold-based. When we describe setting up a threshold, we mean setting up an alert that defines a threshold. The same is true if an individual sets up a constraint. The individual sets up an alert to define a constraint. These values or conditions need to be exceeded or met for an alert to be generated.

Constraints

In contrast to the alerts that are defined with a probe or a scan job, the alerts defined in the Alerting navigation subtree are kept in a special constraint report available in the **Disk Manager** → **Reporting** → **Storage Subsystem Performance** → **Constraint Violation** navigation subtree. This report lists all the threshold-based alerts, which can be used to identify hot spots within the storage environment. You can think of a constraint report as a statistic about how often each alert is triggered in the specified time frame. To effectively use thresholds, the analyst must be familiar with the workloads. Figure 7-11 shows all the available constraint violations. Unfortunately, most of them are not applicable to the DS8000.

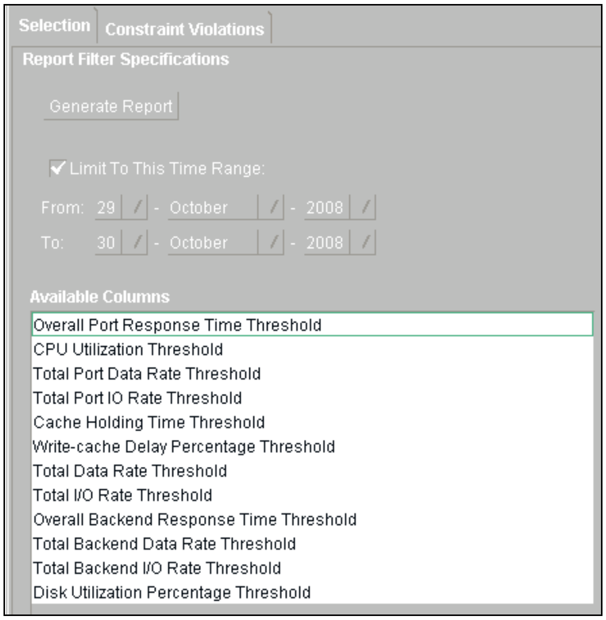


Figure 7-11 Available constraint violations

Table 7-9 shows the constraint violations applicable to the DS8000. For those constraints without predefined values, we provide suggestions. You need to configure the exact values appropriately for the environment. Most of the metrics that are used for constraint violations are I/O rates and I/O throughput. It is difficult to configure thresholds based on these metrics, because absolute threshold values depend on the hardware capabilities and the workload. It might be perfectly acceptable for a tape backup to use the full bandwidth of the storage subsystem ports during backup periods. If the thresholds are configured to identify a high data rate, a threshold is generated. In these cases, the thresholds are exceeded, but the information does not necessarily indicate a problem. These types of exceptions are called "false positives." Other metrics, such as Disk Utilization Percentage, Overall Port Response Time, Write Cache Delay Percentage, and perhaps Cache Hold Time, tend to be more predictive of actual resource constraints and need to be configured in every environment. These constraints are highlighted in green in Table 7-9.

Table 7-9 DS8000 constraints

Condition	Critical stress	Warning stress	Comment
Disk Utilization Percentage Threshold	80	50	Can be effective in identifying consistent disk hot spots.
Overall Port Response Time Threshold	20	10	Can be used to identify hot ports.
Write Cache Delay Percentage Threshold	10	3	Percentage of total I/O operations per processor complex delayed due to write cache space constraints.
Cache Holding Time Threshold	30	60	Amount of time in seconds that the average track persisted in cache per processor complex.

Condition	Critical stress	Warning stress	Comment
Total Port I/O Rate Threshold	Depends	Depends	Indicates highly active ports.
Total Port Data Rate Threshold	Depends	Depends	Indicates highly active port.
Total I/O Rate Threshold	Depends	Depends	Difficult to use, because I/O rates vary depending on workload and configuration.
Total Data Rate Threshold (MB)	Depends	Depends	Difficult to use, because data rates vary depending on workload and configuration.

For information about the exact meaning of these metrics and thresholds, see 7.3, "Tivoli Storage Productivity Center data collection" on page 250.

Figure 7-12 on page 263 is a diagram to illustrate the four thresholds that create five "regions." Stress alerts define levels that, when exceeded, trigger an alert. An *idle threshold level* triggers an alert when the data value drops beneath the defined idle boundary. There are two types of alerts for both the stress category and the idle category:

- Critical Stress: No warning stress alert is created, because both (warning and critical) levels are exceeded with the interval.
- Warning Stress: It does not matter that the metric shows a lower value than in the last interval. An alert is triggered, because the value is still above the warning stress level.
- Normal workload and performance: No alerts are generated.
- Warning Idle: The workload drops, and this drop might indicate a problem (does not have to be performance-related).
- Critical Idle: The same applies as for critical stress.

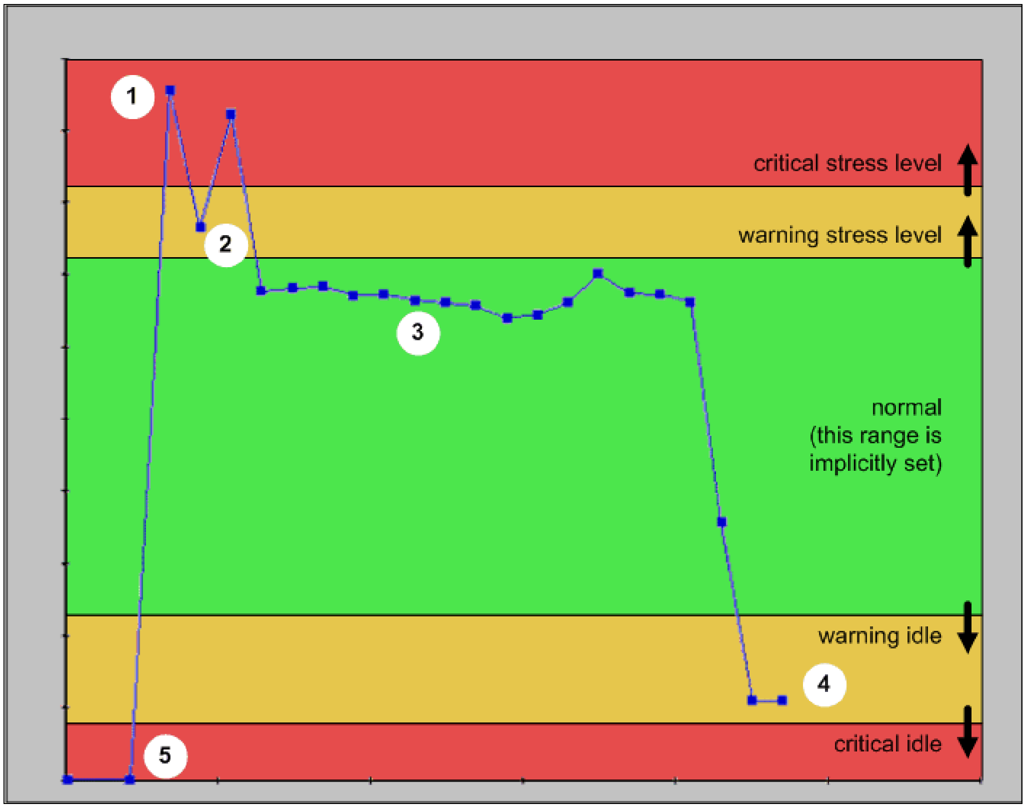


Figure 7-12 Alert levels

It is unnecessary to specify a threshold value for all levels. However, for a growing perspective of your system and effective monitoring of the usage of resources, you might configure the most important thresholds and also some nearby thresholds for good planning.

In order to configure a constraint, perform the following steps:

1. Go to **Disk Manager** → **Alerting**.
2. Right-click **Storage Subsystems Alert**.
3. Select **Create Storage Subsystem Alert**.
4. A window opens that is similar to Figure 7-13 on page 264. Select the triggering condition from the list box and scroll down until you see the desired metrics.

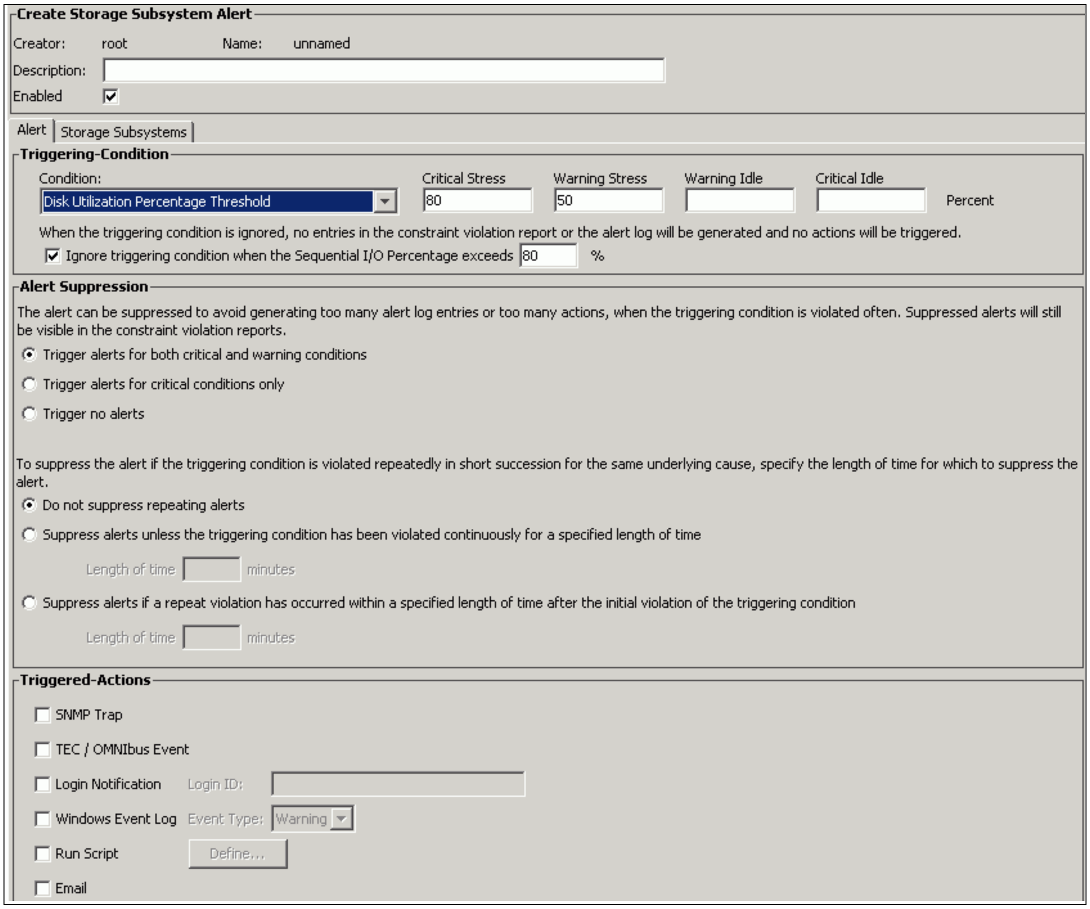


Figure 7-13 Create Storage Subsystem Alert

5. Select the Condition to configure.
6. Set the Critical Stress and Warning Stress levels.
7. On the Storage Subsystems tab, select the systems to which apply the Constraint.
8. Configure any Triggered Actions, such as SNMP Trap, Tivoli Enterprise Console Event, Login Notification, Windows Event Log, Run Script, or Email.
9. Save the alert and provide a name.
10. You can view the alerts in the **Disk Manager** → **Reporting** → **Storage Subsystem Performance - Constraint Violation** navigation subtree.

Limitations to alert definitions

There are a few limitations to alert levels:

- There are only a few constraints that apply to the DS8000.
- Thresholds are always active. They cannot be set to exclude specific time periods.
- Detailed knowledge of the workload is required to use the constraints effectively.
- An alert is reissued for every sample that exceeds the threshold.
- If an idle threshold and a stress threshold are both exceeded, only the stress alert is generated.

False positive alerts: Configuring constraint violations too conservatively can lead to an excessive number of false positive alerts. Be aware to take results compared in a Timeline. Especially if Easy Tier is enabled, it might change the situation after it applies plans after an analysis.

7.5.2 Predefined performance reports in Tivoli Storage Productivity Center

Tivoli Storage Productivity Center has several predefined reports to facilitate tactical performance management processes. To navigate to these reports, select **IBM Tivoli Storage Productivity Center** → **Reporting** → **System Reports** → **Disk**. See Figure 7-14.

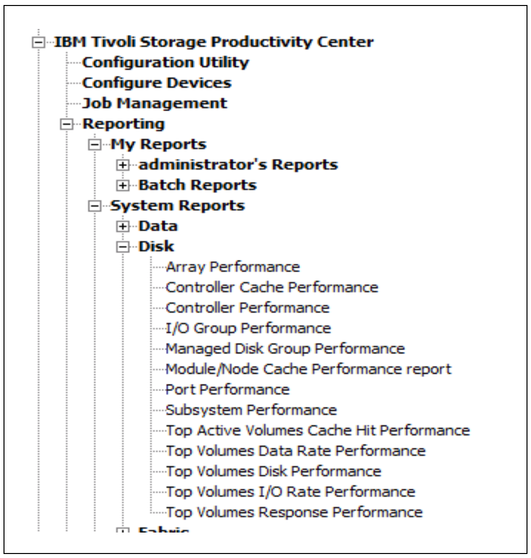


Figure 7-14 Tivoli Storage Productivity Center predefined reports

The predefined Tivoli Storage Productivity Center performance reports are customized reports. The Top Volume reports show only a single metric over a specific time period. These reports provide a way to identify the busiest volumes in the entire environment or by storage subsystem. You can use Selection and Filter options for these reports. We describe the Selection and Filter options in detail in 7.5.4, “Batch reports” on page 269.

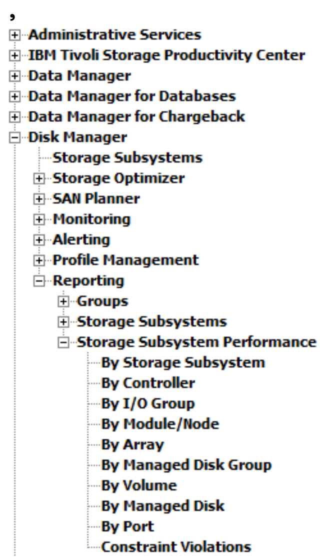
7.5.3 Ad hoc reports

In contrast to the predefined reports, the reports generated in Disk Manager contain all the metrics that apply to that component of the subsystem (for example, controller, array, and volume). Use the Selection and Filter options to eliminate unnecessary data. We suggest that you create and save reports for reuse later.

Create reports in the Reporting panel:

1. Select **Disk Manager** → **Reporting** → **Storage Subsystem Performance** (or in the corresponding Fabric Manager navigation tree) as shown in Figure 7-15 on page 266. Regard these reports as toolkits for building reusable reports.
2. Include only key columns and save them by selecting the **Save** icon or **File** → **Save**. They now show up under the navigation tree in **IBM Tivoli Storage Productivity Center** → **My Reports** → ***User ID*** in My Reports.
3. To generate a chart, select the chart icon as shown .

4. Select the metric to display as shown in Figure 7-16. Click **OK**. A chart, such as the chart shown in Figure 7-17 on page 267, is displayed.
5. Often the report is difficult to analyze due to the layout and the space occupied by the trend lines. Removing the trend lines improves the usability of the chart. Remove trends by right-clicking in the chart area and selecting **Customize Chart** and then clearing the check mark in **Show Trends.**
6. Another option for viewing the data is to export the data to a comma-separated values (csv) file. Select **File** → **Export Data**. Provide a file name as shown in Figure 7-18 on page 267.




```
Select any of the available metrics for DS8000 under
Storage Subsystem Performance:

    By Storage Subsystem
    By Controller
    By array
    By Volume
    By Port

Use the selection and filter criteria as described in
7.5.4, “Batch reports” on page 269.
```

Figure 7-15 Storage Subsystem Performance reports

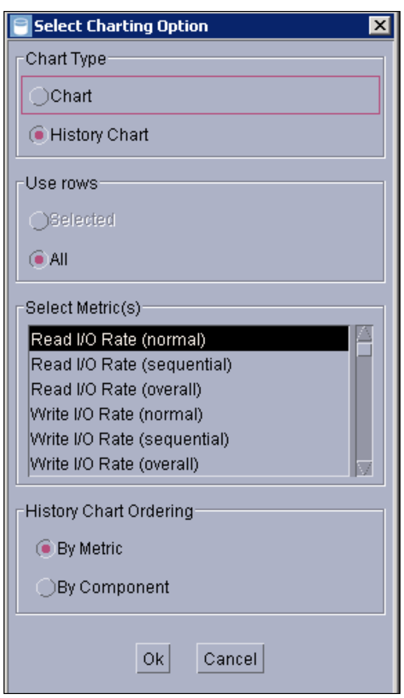


Figure 7-16 Chart metrics

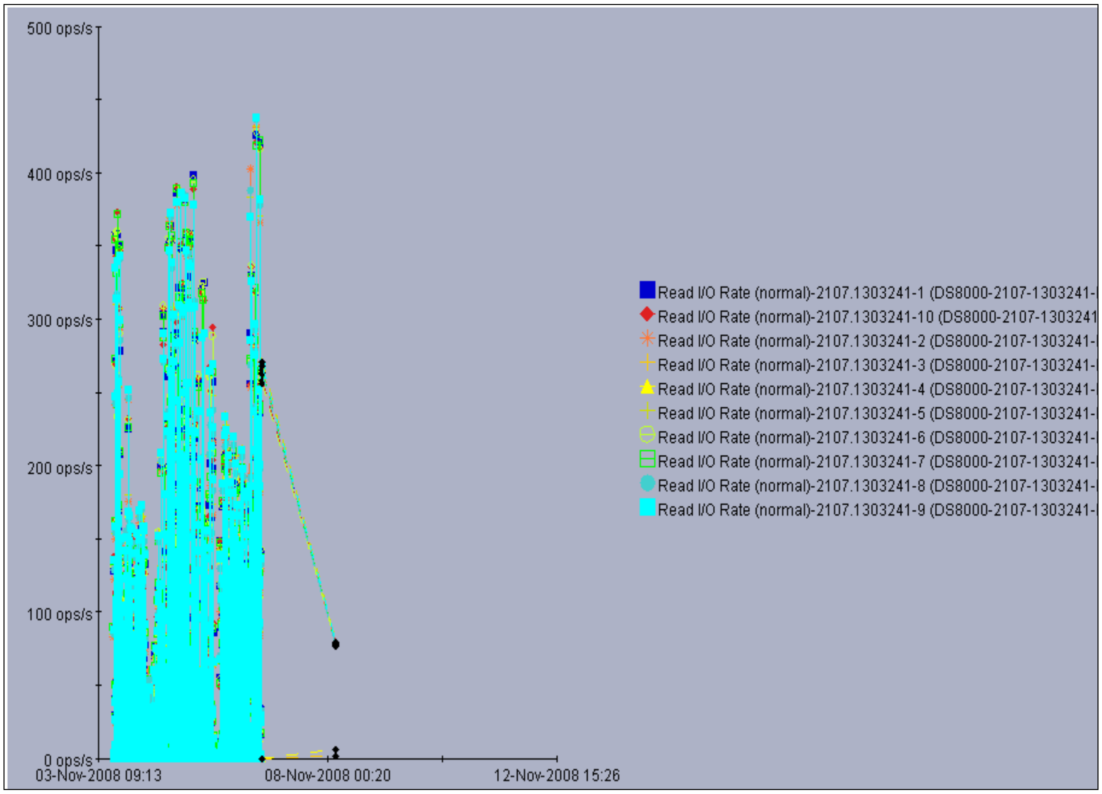


Figure 7-17 Read I/O rate chart

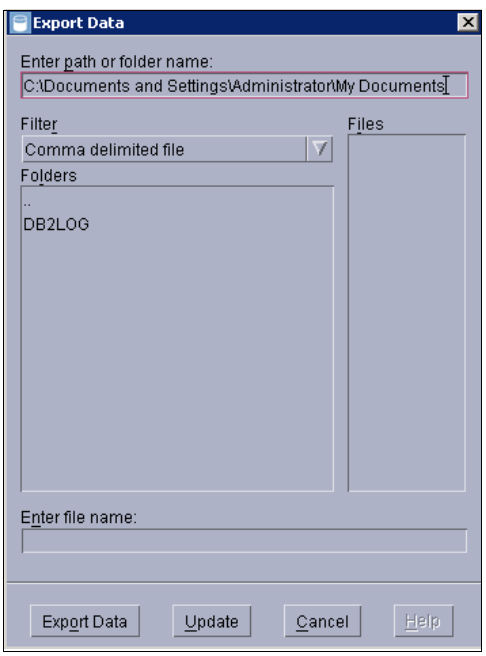
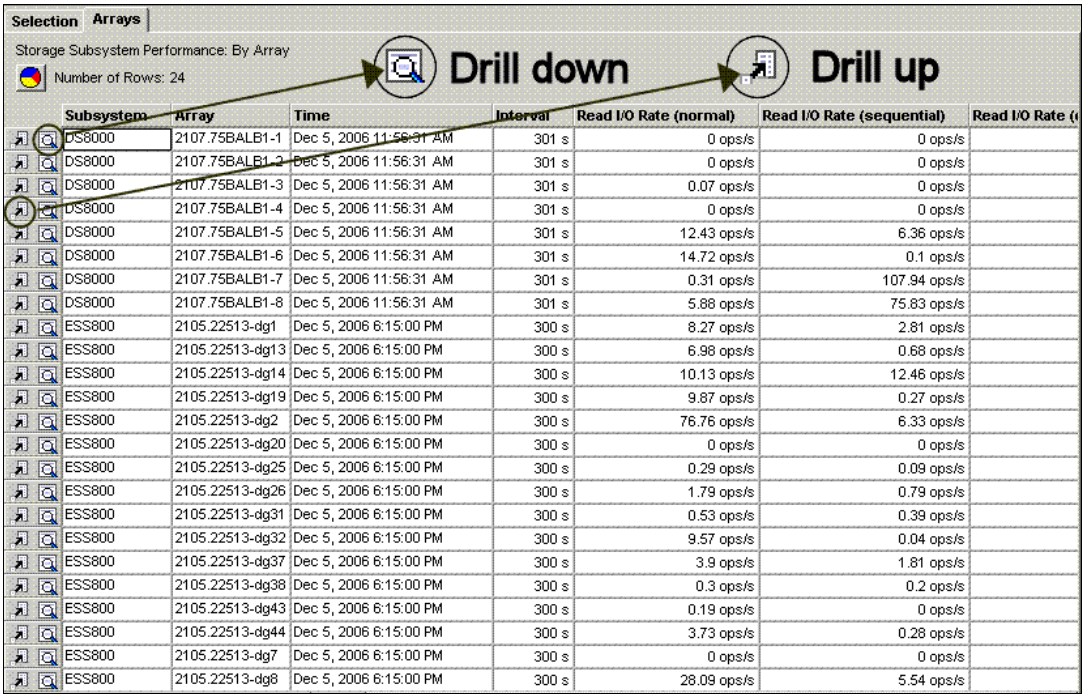


Figure 7-18 Export chart

Drill up and drill down

When you see the report as displayed in Figure 7-19 on page 268, there are two small icons displayed on the left side of the table (we copied and resized the icons on the top of the picture).



Selection | Arrays

Storage Subsystem Performance: By Array

Number of Rows: 24

Subsystem	Array	Time	Interval	Read I/O Rate (normal)	Read I/O Rate (sequential)	Read I/O Rate ((
DS8000	2107.75BALB1-1	Dec 5, 2006 11:56:31 AM	301 s	0 ops/s	0 ops/s	
DS8000	2107.75BALB1-2	Dec 5, 2006 11:56:31 AM	301 s	0 ops/s	0 ops/s	
DS8000	2107.75BALB1-3	Dec 5, 2006 11:56:31 AM	301 s	0.07 ops/s	0 ops/s	
DS8000	2107.75BALB1-4	Dec 5, 2006 11:56:31 AM	301 s	0 ops/s	0 ops/s	
DS8000	2107.75BALB1-5	Dec 5, 2006 11:56:31 AM	301 s	12.43 ops/s	6.36 ops/s	
DS8000	2107.75BALB1-6	Dec 5, 2006 11:56:31 AM	301 s	14.72 ops/s	0.1 ops/s	
DS8000	2107.75BALB1-7	Dec 5, 2006 11:56:31 AM	301 s	0.31 ops/s	107.94 ops/s	
DS8000	2107.75BALB1-8	Dec 5, 2006 11:56:31 AM	301 s	5.88 ops/s	75.83 ops/s	
ESS800	2105.22513-dg1	Dec 5, 2006 6:15:00 PM	300 s	8.27 ops/s	2.81 ops/s	
ESS800	2105.22513-dg13	Dec 5, 2006 6:15:00 PM	300 s	6.98 ops/s	0.68 ops/s	
ESS800	2105.22513-dg14	Dec 5, 2006 6:15:00 PM	300 s	10.13 ops/s	12.46 ops/s	
ESS800	2105.22513-dg19	Dec 5, 2006 6:15:00 PM	300 s	9.87 ops/s	0.27 ops/s	
ESS800	2105.22513-dg2	Dec 5, 2006 6:15:00 PM	300 s	76.76 ops/s	6.33 ops/s	
ESS800	2105.22513-dg20	Dec 5, 2006 6:15:00 PM	300 s	0 ops/s	0 ops/s	
ESS800	2105.22513-dg25	Dec 5, 2006 6:15:00 PM	300 s	0.29 ops/s	0.09 ops/s	
ESS800	2105.22513-dg26	Dec 5, 2006 6:15:00 PM	300 s	1.79 ops/s	0.79 ops/s	
ESS800	2105.22513-dg31	Dec 5, 2006 6:15:00 PM	300 s	0.53 ops/s	0.39 ops/s	
ESS800	2105.22513-dg32	Dec 5, 2006 6:15:00 PM	300 s	9.57 ops/s	0.04 ops/s	
ESS800	2105.22513-dg37	Dec 5, 2006 6:15:00 PM	300 s	3.9 ops/s	1.81 ops/s	
ESS800	2105.22513-dg38	Dec 5, 2006 6:15:00 PM	300 s	0.3 ops/s	0.2 ops/s	
ESS800	2105.22513-dg43	Dec 5, 2006 6:15:00 PM	300 s	0.19 ops/s	0 ops/s	
ESS800	2105.22513-dg44	Dec 5, 2006 6:15:00 PM	300 s	3.73 ops/s	0.28 ops/s	
ESS800	2105.22513-dg7	Dec 5, 2006 6:15:00 PM	300 s	0 ops/s	0 ops/s	
ESS800	2105.22513-dg8	Dec 5, 2006 6:15:00 PM	300 s	28.09 ops/s	5.54 ops/s	

Figure 7-19 Drill down and drill up

If you click the drill-down icon in Figure 7-19, you get a report that contains all the volumes that are stored on that specific array. If you click the drill up icon, you get a performance report at the controller level. In Figure 7-20, we show you the DS8000 components and levels to which you can drill down. Tivoli Storage Productivity Center refers to the DS8000 processor complexes as *controllers*.

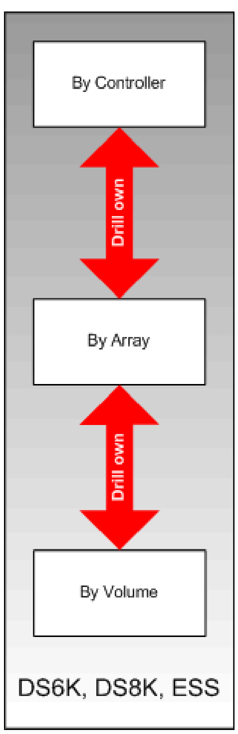


Figure 7-20 Drill-down path and drill-up path

7.5.4 Batch reports

The Tivoli Storage Productivity Center predefined and adhoc reports provide a basic level of reporting, but they have a few limitations. Use batch reports if you want to perform additional analysis and reporting or to regularly schedule performance data extracts. In the following section, we describe the necessary steps for generating batch reports:

1. Expand the **Batch Reports** subtree as shown Figure 7-21.

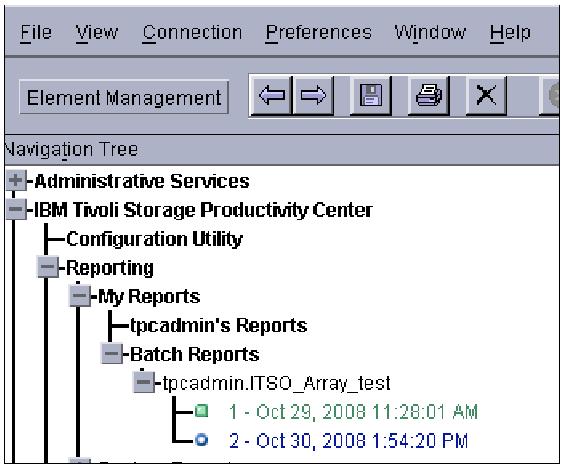


Figure 7-21 Batch Reports

2. Right-click **Batch Reports** and Select **Create Batch Report.**
3. Select the report type and provide a description as shown in Figure 7-22.

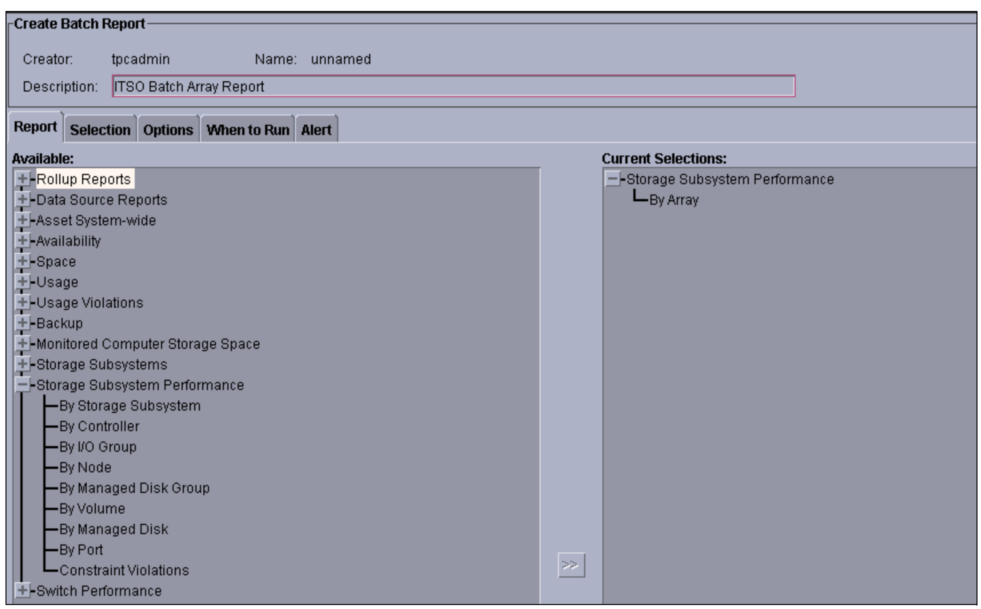


Figure 7-22 Batch report type

4. On the Selection tab, select the date and time range, the interval, and the Subsystem components as shown in Figure 7-23 on page 270.

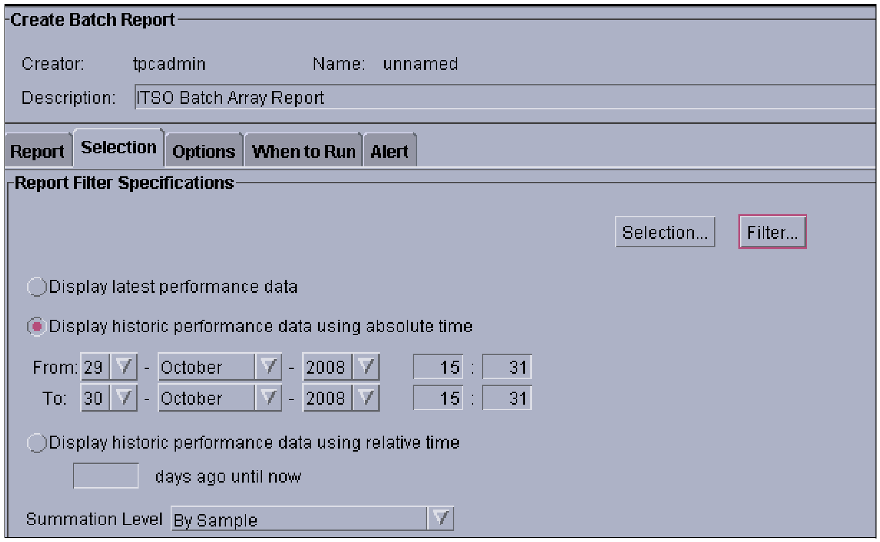


Figure 7-23 Batch selection options

Important: Do not click Selection when extracting volume data. In this case, we suggest that you click Filter. Use the following syntax to gather volumes for only the subsystem of interest: DS8000-2107-#######-IBM. See Figure 7-24 for an example. Replace *#######* with the seven character DS8000 storage image ID. Remember that the number of results might be limited. By filtering, you can also obtain a greater number of results in an interval for the specified filter.

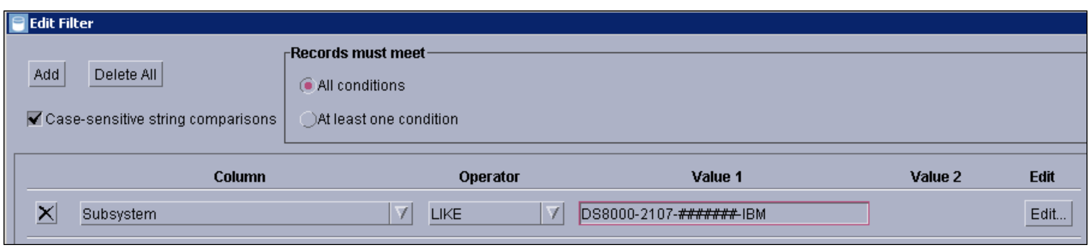


Figure 7-24 Filter on Subsystem

5. To reduce the amount of data, we suggest creating a filter that requires the selected component to contain at least 1 for the Total I/O Rate (overall) (ops/s) as shown in Figure 7-25.

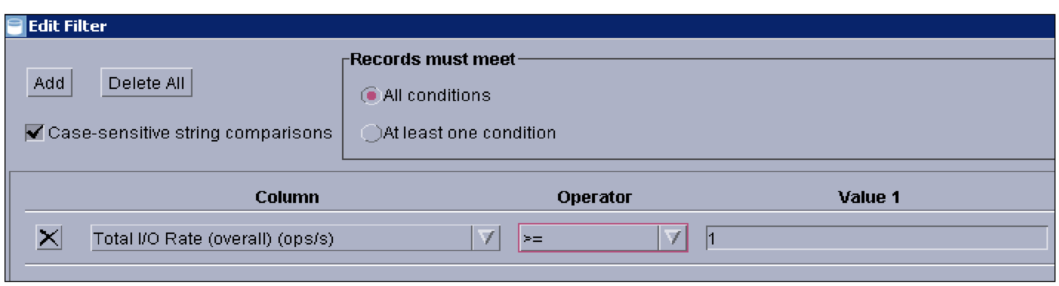


Figure 7-25 Filter on Total I/O Rate

6. Click the **Options** tab and select **Include Headers**.
7. Leave the radio button selected for **CSV File**. This option exports the data to a comma-separated values file that can be analyzed with spreadsheet software.
8. Select an agent computer. Usually, this batch report runs on the Tivoli Storage Productivity Center Server. See Figure 7-26 for an example.

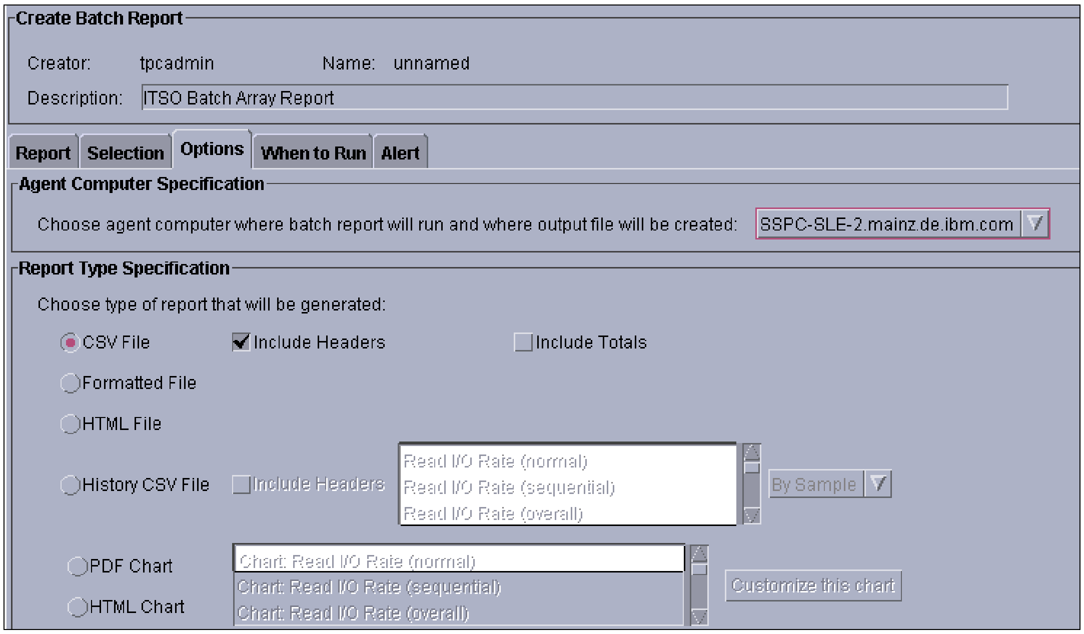


Figure 7-26 Batch options

9. Another consideration is When to Run. Click **When to Run** to see the available options. The default is **Run Now**. This option is fine for ad hoc reporting, but you might also schedule the report to Run Once at a certain time or Run Repeatedly. This tab also contains an option for setting the time zone for the report. The default is to use the local time in each time zone. For more information, see 7.3.1, "Time stamps" on page 250.
10. Before running the job, configure any desired alerts in the Alert tab, which provides a means for sending alerts if the job fails. This feature can be useful if the job is a regularly scheduled job.
11. In order to run the batch report, immediately click the **Save** icon (diskette) in the toolbar as shown in Figure 7-27.

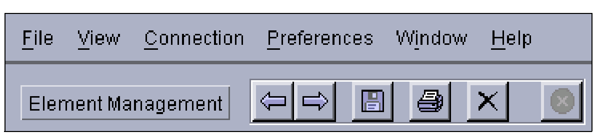


Figure 7-27 Batch report run

12. When clicking the Save icon, a prompt displays `Specify a Batch Report name`. Enter a name that is descriptive enough for later reference.
13. After submitting the job, it is either successful or unsuccessful. Examine the log under the Batch Reports to perform problem determination on the unsuccessful jobs.

> **Reports:** The location of the batch file reports is not intuitive. It is in the Tivoli Storage Productivity Center installation directory as shown in Figure 7-28.

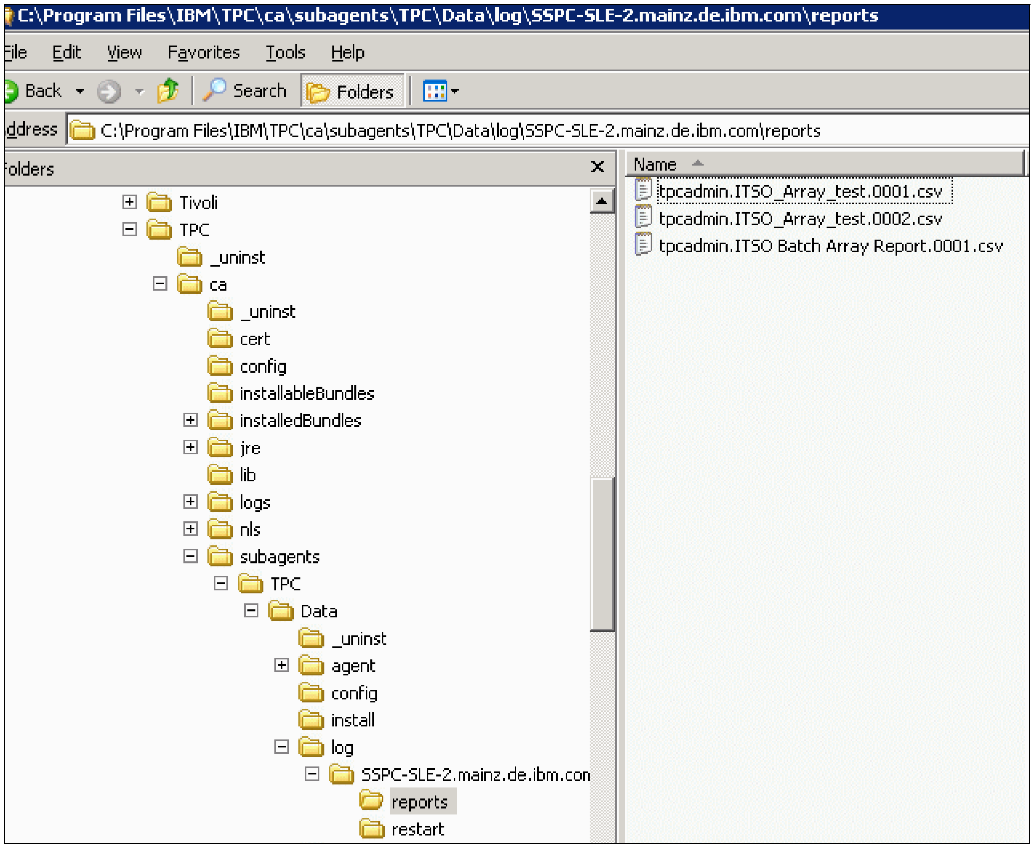


Figure 7-28 Batch file output location

7.5.5 TPCTOOL

You can use the TPCTOOL command-line interface (CLI) to extract data from the Tivoli Storage Productivity Center database. It requires no knowledge of the Tivoli Storage Productivity Center schema or SQL query skills, but you need to understand how to use the tool. It is not obvious. Nevertheless, it has advantages over the Tivoli Storage Productivity Center GUI:

- Multiple components

  Extract information about multiple components, such as volumes and arrays by specifying a list of component IDs. If the list is omitted, every component for which data is gathered is returned.

- Multiple metrics

  The multiple metrics feature is probably the most important feature of the TPCTOOL reporting function. Exporting data from a history chart can include data from multiple samples for multiple components, but it is limited to a single metric type. In TPCTOOL, the metrics are specified by the *columns* parameter.

- The data extraction can be automated. TPCTOOL, when used in conjunction with shell scripting, can provide an excellent way to automate the Tivoli Storage Productivity Center data extracts. This capability can be useful for loading data into a consolidated performance history repository for custom reporting and data correlation with other data sources.

- TPCTOOL can be useful if you need to create your own metrics by using supplied metrics or counters. For example, you can create a metric that shows the access density: the number of I/Os per GB. For this metric, you also need information from other Tivoli Storage Productivity Center reports that include the volume capacity. Manipulating the data requires additional work.

Nevertheless, TPCTOOL also has a few limitations:

- Single subsystem or fabric

  Reports can include data of a single subsystem or a single fabric only, regardless of the components, ctypes, and metrics that you specify.

- Identification

  The identification of components, subsystems, and fabrics is not so easy, because TPCTOOL uses worldwide name (WWN) and Globally Unique Identifiers (GUIDs) instead of the user-defined names or labels. At least for certain commands, you can instruct TPCTOOL to return more information by using the **`-l`** parameter. For example, **`lsdev`** also returns the user-defined label when using the **`-l`** parameter.

- Correlation

  The drill-down relationships provided in the GUI are not maintained in the TPCTOOL extracts. Manual correlation of volume data with the Tivoli Storage Productivity Center array can be done or a script can be used to automate this process.

TPCTOOL has one command for creating reports and several list commands (starting with **`ls`**) for querying information needed to generate a report.

Follow these steps to generate a report with TPCTOOL:

1. Launch TPCTOOL by clicking **tpctool.bat** in the installation directory. Typically, `tpctool.bat` is in `C:\Program Files\IBM\TPC\cli`.
2. List the devices by using **`lsdev`** as shown in Figure 7-29. Note the devices from which to extract data. In this example, we use `2107.1303241+0`.

```
tpctool> lsdev -url localhost:9550  -user tpcadmin -pwd [illegible] -subsys -l -header
2107.7503461+0 DS8000-2107-7503461-IBM - DS8000 UNREACHABLE 2008.11.01:00:29:08
2107.7520781+0 DS8000-2107-7520781-IBM - DS8000 UNREACHABLE 2008.10.27:10:33:24
2107.75GB191+0 DS8000-2107-75GB191-IBM - DS8000 UNREACHABLE 2008.11.05:00:29:08
2107.75GB192+0 DS8000-2107-75GB192-IBM - DS8000 UNREACHABLE 2008.11.05:00:29:08
2107.75DY531+0 DS8000-2107-75DY531-IBM - DS8000 NORMAL      2008.11.06:00:29:11
2107.75DY532+0 DS8000-2107-75DY532-IBM - DS8000 NORMAL      2008.11.06:00:29:16
2105.25688+0   ESS-2105-25688-IBM      - ESS    NORMAL      2008.11.06:00:29:09
2107.1303241+0 DS8000-2107-1303241-IBM - DS8000 NORMAL      2008.11.06:00:29:10
tpctool> _
```

Figure 7-29 TPCTOOL lsdev output

3. Determine the component type to report by using the **`lstype`** command as shown in Figure 7-30.

```
tpctool> lstype
Name        Type Description
=======================================
subsystem   1    Subsystem
subsys_port 2    HBA port
controller  3    Controller
stor_pool   4    Storage Pool
svc_iogrp   5    SVC I/O Group
ds_rio      6    RIO Loop
svc_mdgrp   7    SVC Managed Disk Group
da          8    Device Adapter
ds_rank     9    Rank
array       10   Array
svc_mdisk   11   SVC Managed Disk
vol         12   Volume
switch      13   Switch
switch_port 14   Switch Port
svc_node    15   SVC Node
```

Figure 7-30 TPCTOOL lstype command output list

4. Next, decide which metrics to include in the report. The metrics returned by the **lsmetrics** command are the same as the columns in the Tivoli Storage Productivity Center GUI. Figure 7-31 provides an example of the **lsmetrics** command.

```
tpctool> lsmetrics -user <USERID> -pwd <PASSWORD> -ctype subsystem  -url
localhost:9550 -subsys 2107.1303241+0
Metric                                       Value
================================================
Read I/O Rate (normal)                       801
Read I/O Rate (sequential)                   802
Read I/O Rate (overall)                      803
Write I/O Rate (normal)                      804
Write I/O Rate (sequential)                  805
Write I/O Rate (overall)                     806
Total I/O Rate (normal)                      807
Total I/O Rate (sequential)                  808
Total I/O Rate (overall)                     809
Read Cache Hit Percentage (normal)           810
Record Mode Read I/O Rate                    828
Read Cache Hits Percentage (sequential)      811
Read Cache Hits Percentage (overall)         812
Write Cache Hits Percentage (normal)         813
Write Cache Hits Percentage (sequential)     814
Write Cache Hits Percentage (overall)        815
Total Cache Hits Percentage (normal)         816
Total Cache Hits Percentage (sequential)     817
Total Cache Hits Percentage (overall)        818
Cache Holding Time                           834
Read Data Rate                               819
Write Data Rate                              820
Total Data Rate                              821
Read Response Time                           822
Write Response Time                          823
Overall Response Time                        824
Read Transfer Size                           825
Write Transfer Size                          826
Overall Transfer Size                        827
Record Mode Read Cache Hit Percentage        829
Disk to Cache Transfer Rate                  830
Cache to Disk Transfer Rate                  831
NVS Full Percentage                          832
NVS Delayed I/O Rate                         833
Backend Read I/O Rate                        835
Backend Write I/O Rate                       836
Total Backend I/O Rate                       837
Backend Read Data Rate                       838
Backend Write Data Rate                      839
Total Backend Data Rate                      840
Backend Read Response Time                   841
Backend Write Response Time                  842
Overall Backend Response Time                843
Backend Read Transfer Size                   847
Backend Write Transfer Size                  848
Overall Backend Transfer Size                849
```

Figure 7-31 The lsmetrics command output

5. Determine the start date and time and put in the following format: *YYYY.MM.DD:HH:MM:SS.*
6. Determine the data collection interval in seconds: `86400 (1 day)`.
7. Determine the summarization level: sample, hourly, or daily.
8. Run the report by using the **getrpt** command as shown in Figure 7-32. The command output can be redirected to a file for analysis in a spreadsheet. The <*USERID*> and <*PASSWORD*> variables need to be replaced with the correct values for your environment.

```
tpctool> getrpt -user <USERID> -pwd <PASSWORD> -ctype array -url localhost:9550 -subs
2107.1303241+0 -level hourly -start 2008.11.04:10:00:00 -duration 86400 -columns 801,
Timestamp            Interval Device                   Component       801   802
=================================================================================
2008.11.04:00:00:00      3600 DS8000-2107-1303241-IBM 2107.1303241-10 78.97 26.31
2008.11.04:01:00:00      3600 DS8000-2107-1303241-IBM 2107.1303241-10 54.73 14.85
2008.11.04:02:00:00      3600 DS8000-2107-1303241-IBM 2107.1303241-10 43.72 11.13
2008.11.04:03:00:00      3600 DS8000-2107-1303241-IBM 2107.1303241-10 40.92  8.36
2008.11.04:04:00:00      3600 DS8000-2107-1303241-IBM 2107.1303241-10 50.92 10.03
```

Figure 7-32 The getrpt command sample

> **Tip:** If you want to import the data into Excel later, we suggest that you use a semi-colon as the field separator (**`-fs`** parameter). A comma can easily be mistaken as a decimal or decimal grouping symbol.

The book titled *Tivoli Storage Productivity Center Advanced Topics,* SG24-7438, contains instructions for importing TPCTOOL data into Excel. The book also provides a Visual Basic macro that can be used to modify the time stamp to the international standard.

The **`lstime`** command is helpful, because it provides information that can be used to determine whether performance data collection is running. It provides three fields:

Start The date and time of the start of performance data collections
Duration The number of seconds that the job ran
Option The location

Example 7-2 Using the TPCTOOL lstime command

```
tpctool> lstime -user <USERID> -pwd <PASSWORD> -ctype array -url localhost:9550
 -level hourly -subsys 2107.1303241+0
Start               Duration Option
===================================
2008.10.23:13:00:00 370800   server
2008.10.27:20:00:00 928800   server
```

In order to identify whether the performance job is still running, use the following logic:

1. Identify the start time of the last collection (**2008.10.27 at 20:00:00**).
2. Identify the duration (**928800**).
3. Add the start time to the duration (Use Excel =Sum(2008.10.27 20:00:00+(928800/86400)).
4. Compare the result to the current time. The result is `2008.11.07` at `14:00`, which happens to be the current time. This result indicates that data collection is running.

7.5.6 SAN Planner

With the SAN Planner, you can plan end-to-end for these objects and devices:

- Fabrics
- Hosts
- Storage controllers
- Volumes
- Paths
- Ports
- Zones
- Zone sets
- Storage resource groups (SRGs)
- Replication.

After a plan is made, it can be implemented by SAN Planner. After a plan is made, the user can select to implement the plan with the SAN Planner.

SAN Planner supports TotalStorage Enterprise Storage Server (ESS), IBM System Storage DS6000, IBM System Storage DS8000, and IBM System Storage SAN Volume Controller. SAN Planner supports the Space Only workload profile option for any other storage system supported by Tivoli Storage Productivity Center.

SAN Planner consists of three panels: the Configuration panel, the Planner Section panel, and the Planner Recommendation panel. The Configuration panel contains the Volume Planner, Path Planner, and Zone Planner. The Planner Selection panel contains the Topology Viewer. When you want to provision storage to the host, you can use SAN Planner to provide the best recommendation for creating volumes in the storage subsystems. You can also plan for multipaths and zoning by using SAN Planner. In addition, if you want to replicate the volumes, you can use SAN Planner to provide a recommendation for replicating the volumes.

After you create the plan recommendation, which is also called the *planner output*, you can review it and choose to execute the plan recommendation. The planner can create a job to change the environment based on the plan output. Alternatively, you can vary the input provided to the planner to get multiple recommendations.

For additional information about SAN Planner and more details about how to use it, see Chapter 12 in *SAN Planner* on*Tivoli Storage Productivity Center 4.2 Release Update,* SG24-7894.

7.5.7 Storage Optimizer

Starting with Tivoli Storage Productivity Center Standard Edition 4.2.*x,* the *Storage Optimizer* feature can be used to analyze storage subsystem performance and create an optimization report that includes recommendations to improve the performance. It provides one heat map for each storage subsystem that is included in the analysis and shows the heat for each storage pool. The heat map uses different colors to represent the actual performance as measured against a defined performance threshold (default 80%).

> **Important:** The Storage Optimizer does not modify subsystem configurations. Its primary purpose is to provide you with a performance analysis and optimization recommendations that you can choose to implement at your discretion. It might be useful especially if you use Easy Tier in manual mode to move volumes manually across different extent pools and storage tiers. However, before applying any data migrations, double-check the results and carefully plan your data migrations according to your application performance requirements and growth considerations.

To use the Storage Optimizer, enable analysis on your specified storage subsystems for collecting data (**Disk Manager** → **Storage Optimizer**, see Figure 7-33 on page 277). Later, you need to create an analytics report to obtain results (**Analytics** → **Storage Optimizer**, see Figure 7-34 on page 277). Export it to a PDF file, for example (click the job execution under **Analytics** → **Storage Optimizer** and configure your report).

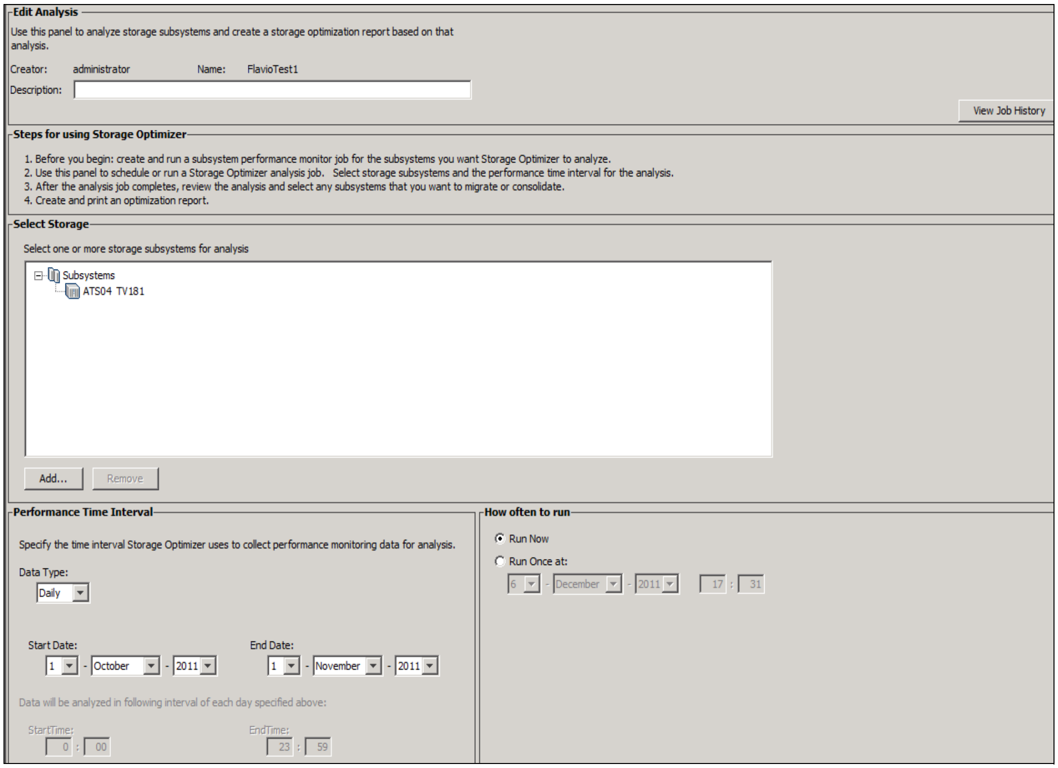


Figure 7-33 Configure a Storage Optimizer analysis job

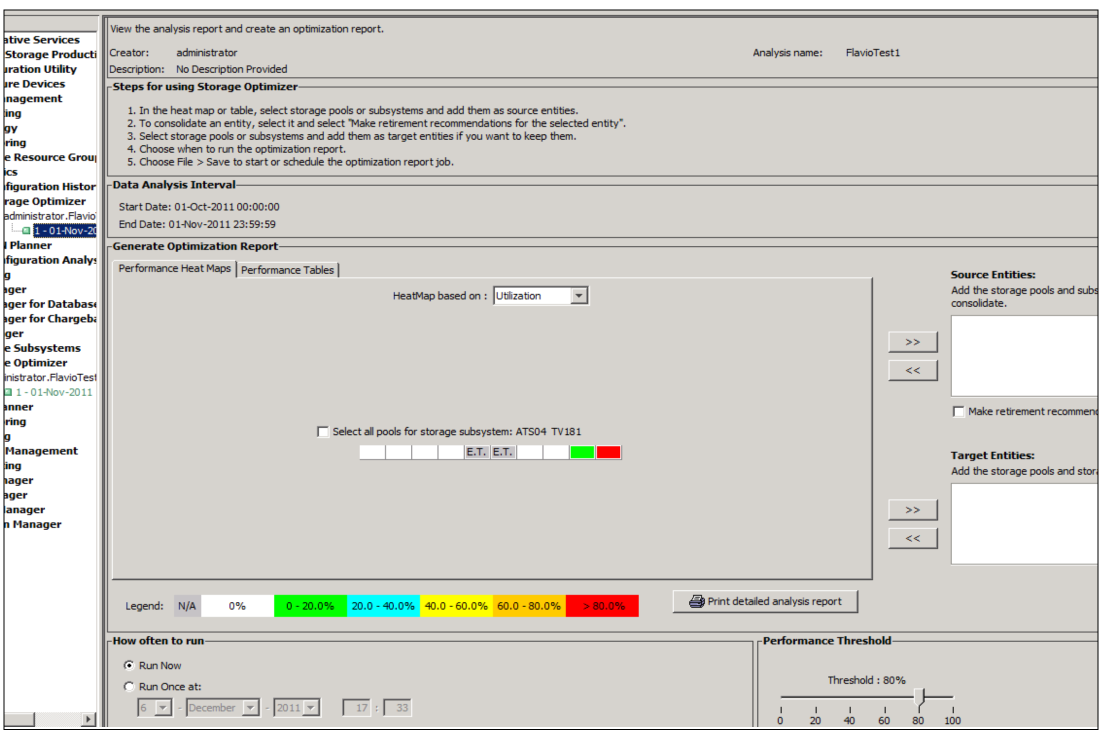


Figure 7-34 Create a Storage Optimizer optimization report

An example Storage Optimizer report for a DS8000 system is shown in Figure 7-35 on page 278.

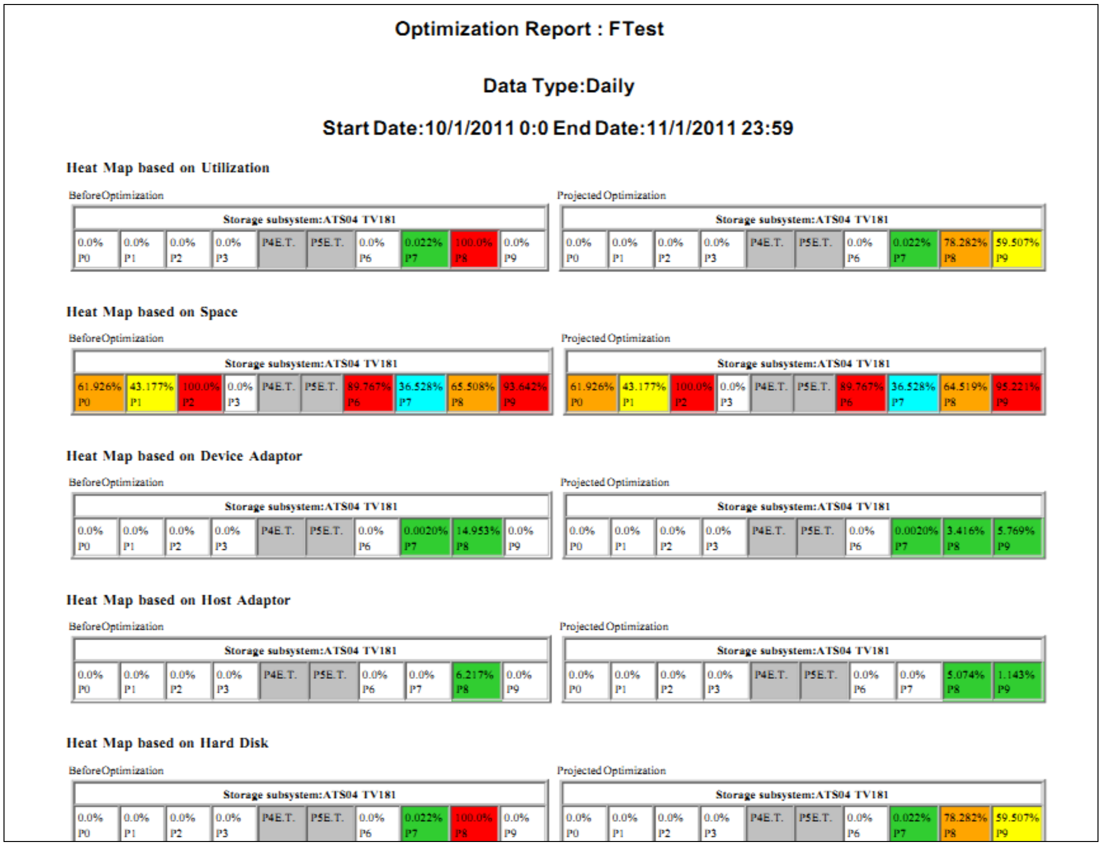


Figure 7-35 Storage Optimizer report example 1

An example of the detailed report is shown in Figure 7-36 on page 279. The detailed report includes some recommendations for moving data manually across storage pools to help improve performance and avoid hot spots.

Report Description

1	• Move storage volume ckd_p0_321E(ID:321e) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
2	• Move storage volume ckd_p0_320D(ID:320d) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
3	• Move storage volume ckd_p0_320A(ID:320a) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
4	• Move storage volume ckd_p0_3211(ID:3211) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
5	• Move storage volume ckd_p0_3202(ID:3202) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
6	• Move storage volume ckd_p0_3206(ID:3206) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
7	• Move storage volume ckd_p0_321B(ID:321b) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
8	• Move storage volume ckd_p0_3205(ID:3205) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.
9	• Move storage volume ckd_p0_322B(ID:322b) from source pool ckd_600_0 of subsystem ATS04 TV181 to target pool ITSO_SmallP of subsystem ATS04 TV181. • Datapath to be added [Initiator WWN: N/A, Target WWN: 500507630A5BC29F, Volume: ITSO_ckd_2(ID:91a9)] • No zoning changes are necessary • The reason is Hard disk utilization is 100%.

Figure 7-36 Storage Optimizer Report Example 2

7.5.8 TPC Reporter for Disk

IBM TPC Reporter for Disk V2 is a Microsoft Windows (.NET 4.0 Framework) application that connects remotely to a server that is running IBM Tivoli Storage Productivity Center software, Version 4.1 and later. The TPC Reporter extracts storage system information and hourly performance statistics from the DB2 database of the Tivoli Storage Productivity Center server. Extracted statistics are compiled locally and transcribed into a white paper-style PDF file, which is saved on the local machine. The report contains information that describes your storage server use in detail.

The automatically generated report contains an overview of the subsystem information, basic attributes of each subsystem component, a performance summary of each component, aggregate statistics of each component, and charts that describe information about each component instance in detail. The DS8000 component types reported are subsystem, ports, arrays, and volumes.

Configuring TPC Reporter for Disk

Follow these steps to run the TPC Reporter for Disk:

1. Download and install the package on a system connected to the same network as the Tivoli Storage Productivity Center server. Download the software from this website:

   `http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS2618`

2. Install software per the accompanying readme file. Check for the specific DB2 runtime fix pack that you probably still need to install, and .NET 4.0 if not already installed on the server that you select for the TPC Reporter. TPC Reporter can also be installed on your System Storage Productivity Center (SSPC).

3. Start the program by clicking the **TPC Reporter for Disk** icon. A start window opens as shown in Figure 7-37. Click **Start**.

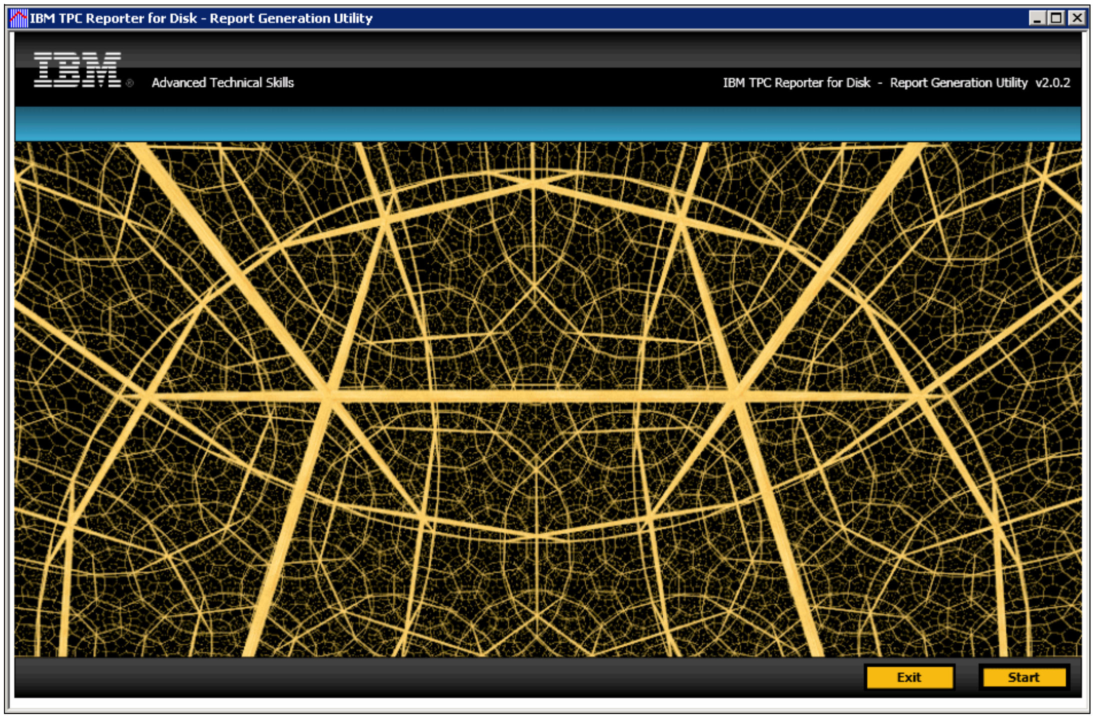


Figure 7-37 TPC Reporter V2: Start window

4. On the next page, the software prompts you for Customer Contact Information and IBM/Business Partner Contact Information. Because these values become part of the PDF and are permanently stored, enter them fully, and click **Next**.
5. The TPC Reporter asks for the TPC for Disk database server IP address, user ID, and password to use to connect. Enter this information as shown in Figure 7-38 on page 281, and click **Next**.

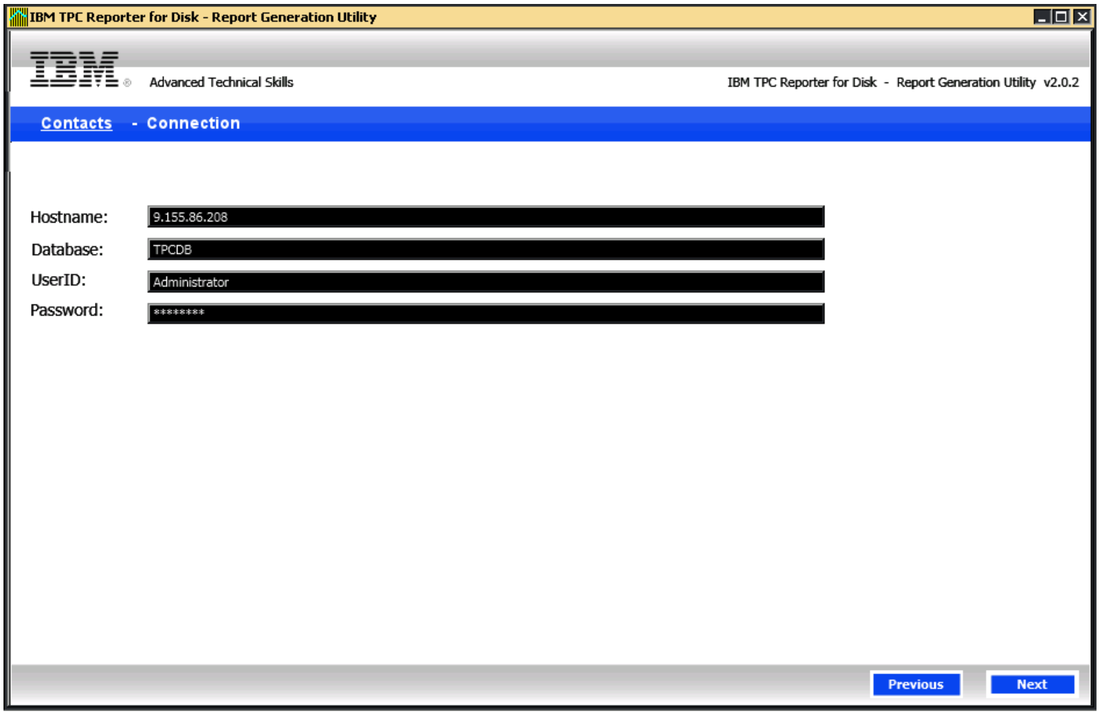


Figure 7-38 TPC Reporter V2: Entering the TPC for Disk server IP address, user ID, and password

6. On the next page, TPC Reporter asks which counters must be present in the PDF. You can use the volume threshold to exclude volumes with less or no IOPS activity from the report. When using TPC Reporter with SAN Volume Controller, there are more counters available. For DS (the DS8000), unless you want to reduce the PDF size, click all counters, specify the export directory as in Figure 7-39, and click **Next**.

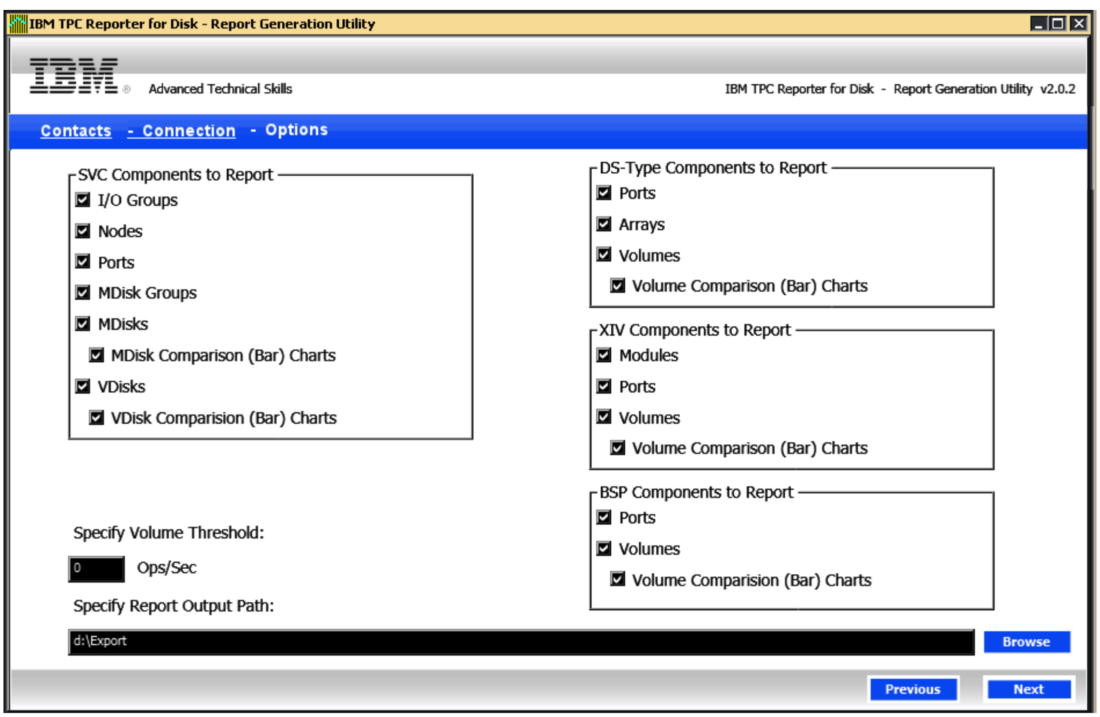


Figure 7-39 TPC Reporter: Specify counters and output path for the PDF report

7. After the TPC Reporter can connect to this database, it shows the various IBM disk systems that are currently actively monitored for performance by your TPC for Disk installation. Specify your disk system and specify the time range that you want (if you encounter problems when you enter the time range, try to change regional localization to English-US) as in Figure 7-40, then click **Next**.

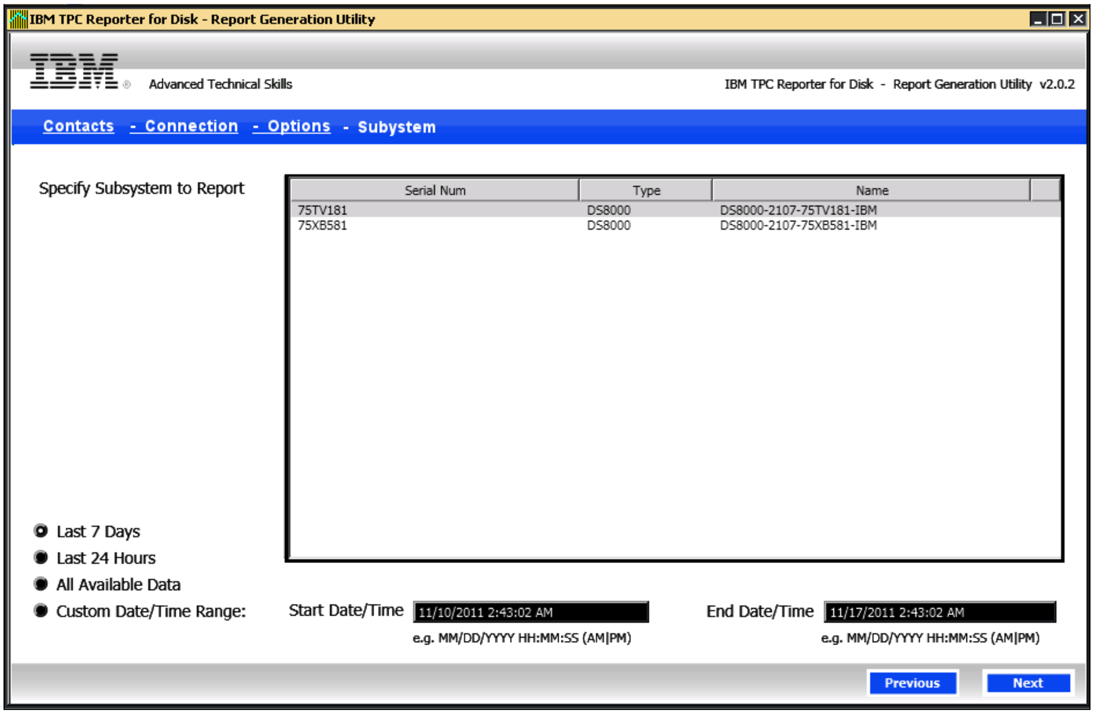


Figure 7-40 TPC Reporter: Specify your disk system and the time range that you want

8. On the next page, as depicted in Figure 7-41 on page 283, click **Generate Report** to start processing. After it completes, click **Finish**.

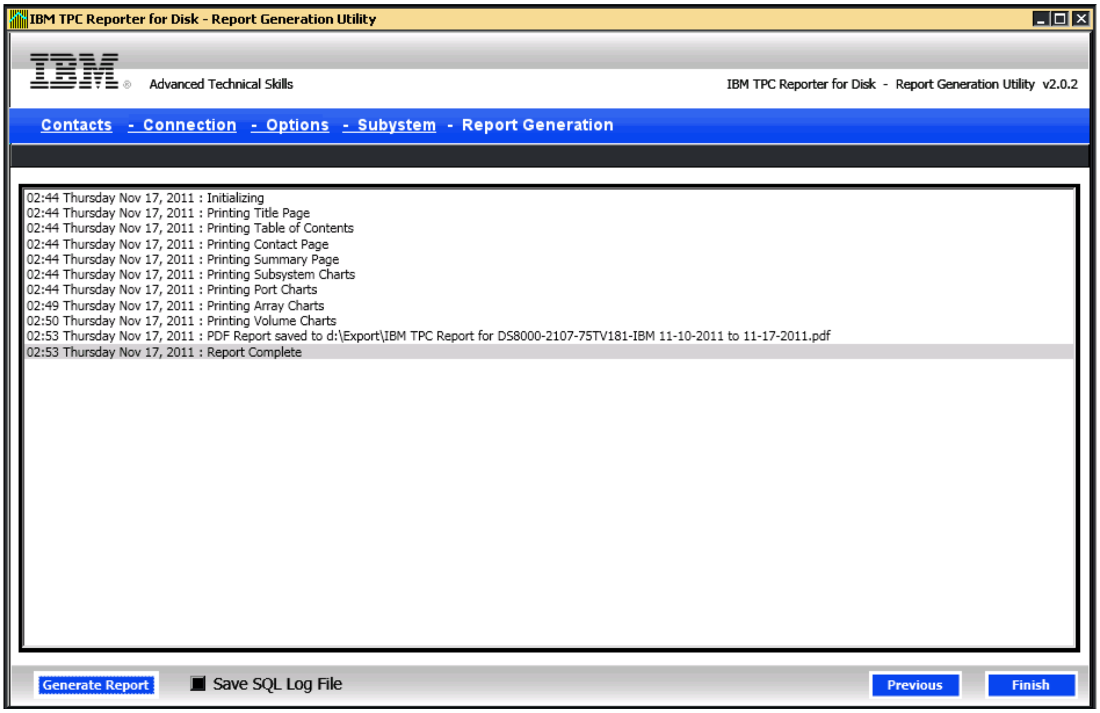


Figure 7-41 TPC Reporter: Generating the PDF report

A report is now in the selected location in the previous steps. The report contains the following information:

- Subsystem Summary
- Subsystem Statistics
- Port Information
- Port Performance Summary
- Port Comparison Statistics
- Port Statistics
- Array Information
- Array Performance Summary
- Array Comparison Statistics
- Array Statistics
- Volume Information
- Volume Performance Summary
- Volume Comparison Statistics

TPC Reporter for Disk is an excellent way to generate regular performance healthcheck reports, especially for ports and arrays. You might want to exclude, for example, the volume information, to make the report shorter, if this information is not necessary for deeper analysis.

For long-term observation, the TPC Reporter can provide a quick view while the performance situation on the system develops. Keep the PDFs for historical reference. Also, configuration information is contained, for instance, the number of type of disk arrays currently installed, and their RAID formats.

Figure 7-42 on page 284 and Figure 7-43 on page 284 are examples.

Subsystem Summary

Information

Subsystem Name:	DS8000-2107-75TV181-IBM
System Type:	DS8000
Vendor:	IBM
Model ID:	951
Code Level:	7.6.20.187

Capacity

Disk Space:	97.4 TB
Available Disk Space:	52.5 TB
Volume Space:	46.9 TB
Consumable Volume Space:	46.9 TB
Unavailable Disk Space:	9.7 TB
Formatted Space:	63.7 TB
Cache Size:	235.8 GB
Nonvolatile Storage:	8.0 GB

Components

Number of Ports:	64
Number of Arrays:	22
Number of Disks:	184
Number of Volumes:	7248

Performance Summary

Max. Total I/O Rate (overall)	29,374.4 Ops/Sec
Avg. Total I/O Rate (overall)	22,597.1 Ops/Sec
Max. Total Data Rate	276.3 MB/Sec
Avg. Total Data Rate	207.4 MB/Sec
Max. Total Response Time	5.5 MSec/Op
Avg. Total Response Time	2.7 MSec/Op
Max. Total Transfer Size	10.4 KB/Ops
Avg. Total Transfer Size	9.5 KB/Ops
Max. Write-cache Delay	14.1 %
Avg. Write-cache Delay	4.1 %
Max. Total Cache Hit (overall)	98.6 %
Avg. Total Cache Hit (overall)	96.5 %

Array Backend Performance Summary

	Total Backend I/O Rate (Ops/Sec)		Total Backend Data Rate (MB/Sec)		Overall Backend Response Time (MSec/Op)	
Name	Avg	Max	Avg	Max	Avg	Max
2107.75TV181-A11	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A22	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A23	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A6	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A12	0.0	0.6	0.0	0.0	0.0	6.4
2107.75TV181-A0	1,130.6	2,048.3	28.2	74.8	1.3	2.7
2107.75TV181-A13	367.1	1,137.8	18.9	57.3	48.5	228.1
2107.75TV181-A8	0.2	0.6	0.0	0.0	20.5	79.2
2107.75TV181-A19	0.1	17.8	0.0	0.1	0.6	18.7
2107.75TV181-A9	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A1	3,575.3	5,199.6	100.9	148.9	5.8	36.6
2107.75TV181-A16	328.4	476.8	13.8	48.6	36.3	67.0
2107.75TV181-A2	838.0	1,029.1	48.0	86.0	64.2	112.5
2107.75TV181-A3	778.5	1,008.5	38.9	74.0	47.5	93.8
2107.75TV181-A10	0.2	17.7	0.3	30.8	0.2	12.5
2107.75TV181-A18	0.2	17.6	0.3	30.7	0.2	15.2
2107.75TV181-A15	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A21	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A5	0.0	0.0	0.0	0.0	0.0	0.0
2107.75TV181-A14	0.0	0.0	0.0	0.0	0.3	21.3
2107.75TV181-A20	0.0	0.1	0.0	0.0	0.2	40.0
2107.75TV181-A7	0.1	17.8	0.0	0.1	0.7	16.0

Figure 7-42 TPC Reporter V2 Summary view examples: Overall machine and Array Backend Performance summaries

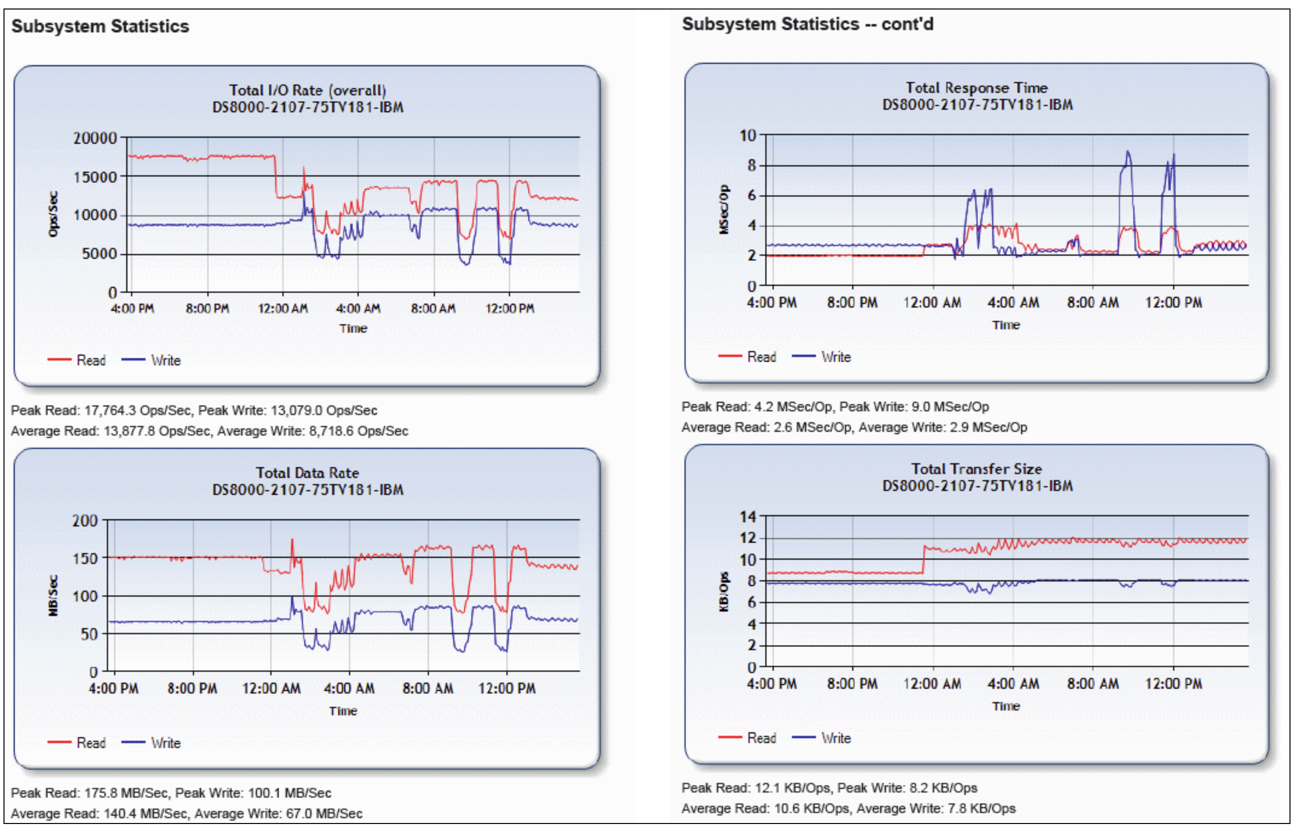


Figure 7-43 TPC Reporter Summary statistics: Subsystem view examples

7.6 Monitoring performance of a SAN switch or director

All SAN switch and director vendors provide management software that includes performance monitoring capabilities. The real-time SAN statistics, such as port utilization and throughput information available from SAN management software, can be used to complement the performance information provided by host servers or storage subsystems.

For additional information about monitoring performance through a SAN switch or director point product, see these websites:

http://www.ibm.com/systems/networking/
http://www.brocade.com
http://www.cisco.com

Most SAN management software includes options to create SNMP alerts based on performance criteria, and to create historical reports for trend analysis. Certain SAN vendors offer advanced performance monitoring capabilities, such as measuring I/O traffic between specific pairs of source and destination ports, and measuring I/O traffic for specific LUNs.

In addition to the vendor point products, the Tivoli Storage Productivity Center Fabric Manager can be used as a central data repository and reporting tool for switch environments. It lacks real-time capabilities, but Tivoli Storage Productivity Center Fabric Manager collects and reports on data at a 5 - 60 minute interval for later analysis.

The Fabric Manager of Tivoli Storage Productivity Center provides facilities to report on fabric topology, configuration and configuration changes, and switch and port performance and errors. In addition, you can use it to configure alerts or constraints for Total Port Data Rate and Total Port Packet Rate. Configuration options allow the creation of events to be triggered if the constraints are exceeded. Although Tivoli Storage Productivity Center does not provide real-time monitoring, it offers several advantages over traditional vendor point products:

- Ability to store performance data from multiple switch vendors in a common database
- Advanced reporting and correlation between host data and switch data through custom reports
- Centralized management and reporting
- Aggregation of port performance data for the entire switch

In general, you need to analyze SAN statistics for these reasons:

- Ensure that there are no SAN bottlenecks that limit the DS8000 I/O traffic, for example, analyze any link utilization over 80%.
- Confirm that multipathing/load balancing software operates as expected.
- Isolate the I/O activity contributed by adapters on different host servers that share storage subsystem I/O ports.
- Isolate the I/O activity contributed by different storage subsystems accessed by the same host server.

7.6.1 SAN configuration examples

We look at four example SAN configurations where SAN statistics might be beneficial for monitoring and analyzing the DS8000 performance.

The first example configuration, which is shown in Figure 7-44 on page 286, has host server Host_1 that connects to DS8000_1 through two SAN switches or directors (SAN Switch/Director_1 and SAN Switch/Director_2). There is a single inter-switch link (ISL) between the two SAN switches. In this configuration, the performance data available from the host and from the DS8000 cannot show the performance of the ISL. For example, if the Host_1 adapters and the DS8000_1 adapters do not achieve the expected throughput, the SAN statistics for utilization of the ISL must be checked to determine whether the ISL is limiting I/O performance.

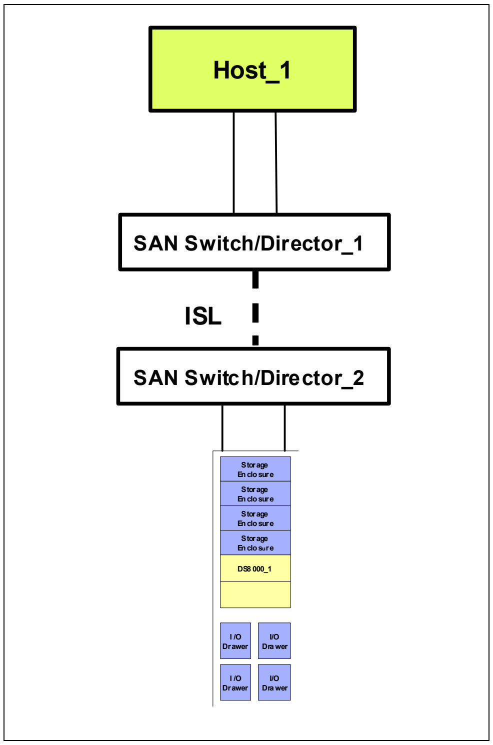


Figure 7-44 Inter-switch link (ISL) configuration

A second type of configuration in which SAN statistics can be useful is shown in Figure 7-45 on page 287. In this configuration, host bus adapters or channels from multiple servers access the same set of I/O ports on the DS8000 (server adapters 1 - 4 share access to the DS8000 I/O ports 5 and 6). In this environment, the performance data available from only the host server or only the DS8000 might not be enough to confirm load balancing or to identify the contributions of each server to I/O port activity on the DS8000, because more than one host is accessing the same DS8000 I/O ports.

If DS8000 I/O port 5 is highly utilized, it might not be clear whether Host_A, Host_B, or both hosts are responsible for the high utilization. Taken together, the performance data available from Host_A, Host_B, and the DS8000 might be enough to isolate the contribution of each server connection to I/O port utilization on the DS8000. However, the performance data available from the SAN switch or director might make it easier to see load balancing and relationships between I/O traffic on specific host server ports and the DS8000 I/O ports at a glance. The performance data can provide real-time utilization and traffic statistics for both host server SAN ports and DS8000 SAN ports in a single view, with a common reporting interval and metrics. The Tivoli Storage Productivity Center Fabric Manager can be used for analysis of historical data, but it does not collect data in real time.

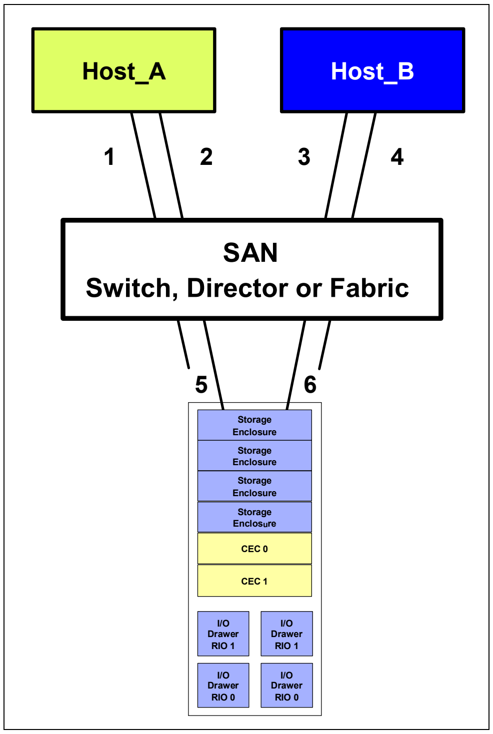


Figure 7-45 Shared DS8000 I/O ports

SAN statistics can also be helpful in isolating the individual contributions of multiple DS8000s to I/O performance on a single server. In Figure 7-46 on page 288, host bus adapters or channels 1 and 2 from a single host (Host_A) access I/O ports on multiple DS8000s (I/O ports 3 and 4 on DS8000_1 and I/O ports 5 and 6 on DS8000_2).

In this configuration, the performance data available from either the host server or from the DS8000 might not be enough to identify the contribution of each DS8000 to adapter activity on the host server, because the host server is accessing I/O ports on multiple DS8000s. For example, if adapters on Host_A are highly utilized or if I/O delays are experienced, it might not be clear whether this situation is due to traffic that is flowing between Host_A and DS8000_1, between Host_A and DS8000_2, or between Host_A and both DS8000_1 and DS8000_2.

The performance data available from the host server and from both DS8000s can be used together to identify the source of high utilization or I/O delays. Additionally, you can use the Tivoli Storage Productivity Center Fabric Manager or vendor point products to gather performance data for both host server SAN ports and DS8000 SAN ports.

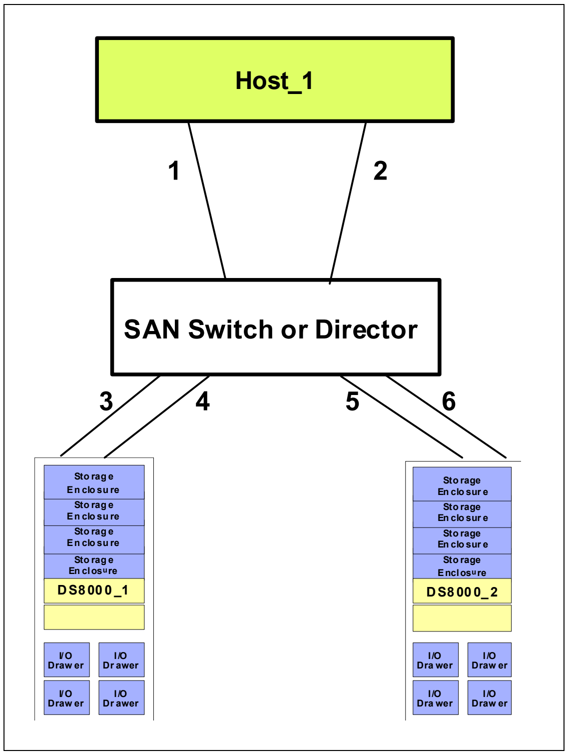


Figure 7-46 Single server that accesses multiple DS8000s

Another configuration in which SAN statistics can be important is a remote mirroring configuration, such as the configuration shown in Figure 7-47. Two DS8000s are connected through a SAN for synchronous or asynchronous remote mirroring or remote copying, and the SAN statistics can be collected to analyze traffic for the remote mirroring links.

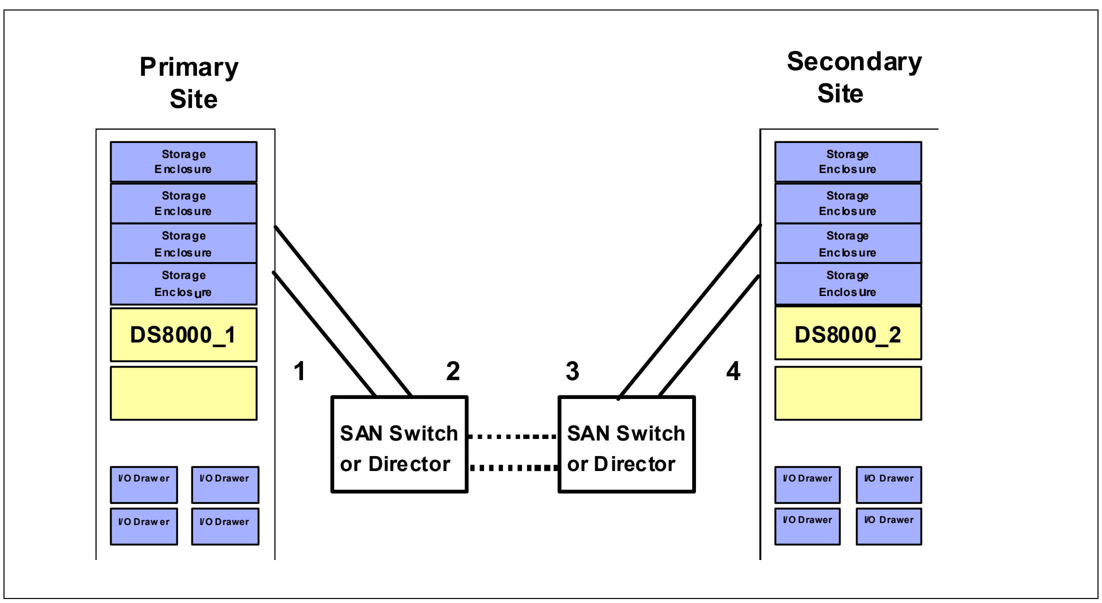


Figure 7-47 Remote mirroring configuration

You must check SAN statistics to determine whether there are SAN bottlenecks that limit DS8000 I/O traffic. You can also use SAN link utilization or throughput statistics to break down the I/O activity contributed by adapters on different host servers to shared storage subsystem

I/O ports. Conversely, you can use SAN statistics to break down the I/O activity contributed by different storage subsystems accessed by the same host server. SAN statistics can also highlight whether multipathing/load-balancing software is operating as desired, or whether there are performance problems that need to be resolved.

7.6.2 Tivoli Storage Productivity Center Fabric Manager alerts

Tivoli Storage Productivity Center Fabric Manager helps you to configure two types of alerts, for the components Switch, Fabric, and Endpoint Device:

- Discovery

  An alert due to a type of change in the environment. These alerts are relevant to performance from the perspective of identifying changes in the environment.

- Threshold

  An alert due to a type of performance threshold exception or constraint violation.

The following Switch Threshold alerts might matter most:

- Total Port Data Rate Threshold
- Link Failure Rate Threshold
- Error Frame Rate Threshold
- Total Port Packet Rate Threshold
- Invalid Transmission Word Rate Threshold
- CRC Error Rate Threshold
- Port Receive Bandwidth Percentage Threshold
- Port Send Bandwidth Percentage Threshold

It is not only the formal bandwidth that can be exceeded (for example, the Bandwidth percentages are preselected to be set to 75% Warning level and 85% Critical). Also, in bad-quality SAN connections, due to many retries and resent packages, we might see performance degradations that we can track.

Configuring Tivoli Storage Productivity Center Fabric Manager alerts

The discovery thresholds are enabled by default and reported during a probe. The threshold alerts require configuration. The following steps illustrate how to configure a new threshold alert:

1. In the navigation tree, expand **Fabric Manager** and **Alerting**. Right-click **Switch Alerts** and select **Create Switch Alerts**. A window similar to Figure 7-48 on page 290 opens.

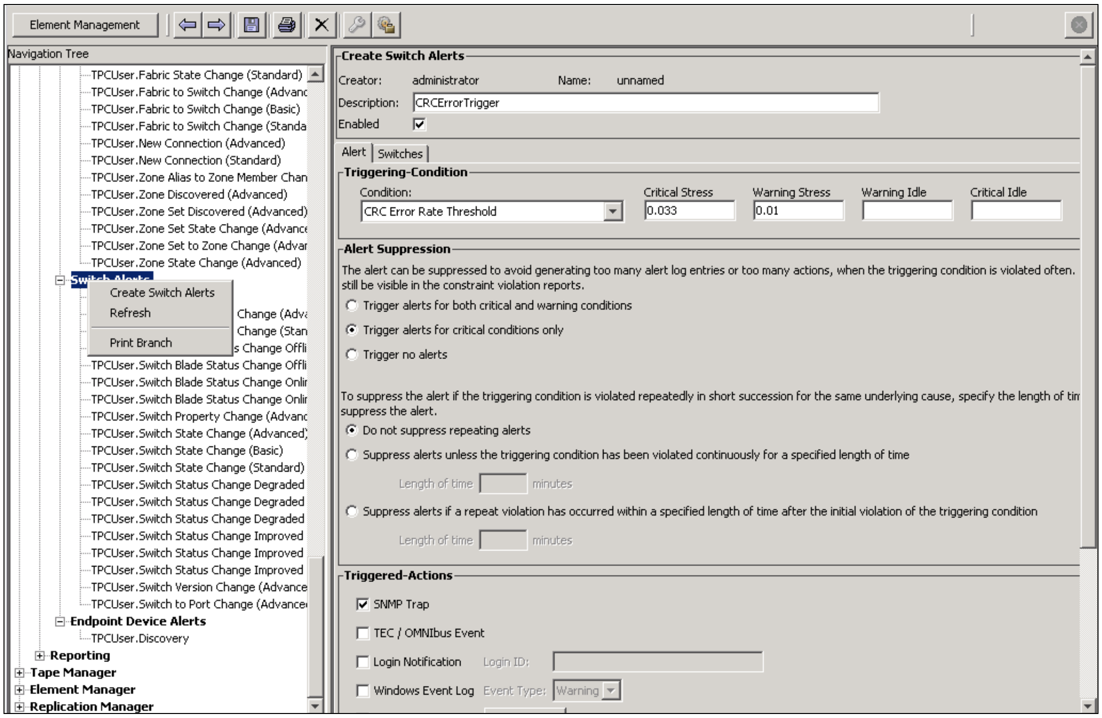


Figure 7-48 Tivoli Storage Productivity Center Fabric Manager: Switch alert threshold

2. Configure the Critical Stress and Warning Stress rates.
3. Enable any Triggered Actions and save.

7.6.3 Tivoli Storage Productivity Center Fabric Manager reporting

The Tivoli Storage Productivity Center Fabric Manager can create several types of performance reports, which are shown in Table 7-10.

Table 7-10 Tivoli Storage Productivity Center Fabric Manager reports

Report type	Report name	Comments
Predefined	Switch Performance	Aggregate of all ports for a switch.
Predefined	Total Switch Port Data Rate	Graph individual switch port metrics.
Predefined	Total Switch Port Packet Rate	Graph individual switch port metrics.
Ad hoc	Create line chart with up to 10 ports and any supported metric	Useful for identifying port hot spots over time.
Batch	Export port performance data	Useful for exporting data for analysis in spreadsheet software.
TPCTOOL	Command-line tool for extracting data from TPC	Extract data for analysis in spreadsheet software. Can be automated.
Custom	Create custom queries by using BIRT	Useful for creating reports not available in Tivoli Storage Productivity Center.

The process of using Tivoli Storage Productivity Center Fabric Manager to create reports is similar to the process that is used to create reports in Tivoli Storage Productivity Center Disk

Manager as described in 7.5, “Tivoli Storage Productivity Center reporting options” on page 258.

7.6.4 Tivoli Storage Productivity Center Fabric Manager metrics

Table 7-11 shows a selection of the Tivoli Storage Productivity Center Fabric Manager metrics that you can collect.

Table 7-11 Tivoli Storage Productivity Center for Fabric Manager metrics

Metric	Definition
Port Send Packet Rate	Average number of packets per second for send operations, for a particular port during the sample interval
Port Receive Packet Rate	Average number of packets per second for receive operations, for a particular port during the sample interval
Total Port Packet Rate	Average number of packets per second for send and receive operations, for a particular port during the sample interval
Port Send Data Rate	Average number of mebibytes (2^{20} bytes) per second that were transferred for send (write) operations, for a particular port during the sample interval
Port Receive Data Rate	Average number of mebibytes (2^{20} bytes) per second that were transferred for receive (read) operations, for a particular port during the sample interval
Total Port Data Rate	Average number of mebibytes (2^{20} bytes) per second that were transferred for send and receive operations, for a particular port during the sample interval
Port Peak Send Data Rate	Peak number of mebibytes (2^{20} bytes) per second that were sent by a particular port during the sample interval
Port Peak Receive Data Rate	Peak number of mebibytes (2^{20} bytes) per second that were received by a particular port during the sample interval
Port Send Packet Size	Average number of KB sent per packet by a particular port during the sample interval
Port Receive Packet Size	Average number of KB received per packet by a particular port during the sample interval
Overall Port Packet Size	Average number of KB transferred per packet by a particular port during the sample interval
Error Frame Rate	The average number of frames per second that were received in error during the sample interval
Dumped Frame Rate	The average number of frames per second that were lost due to a lack of available host buffers during the sample interval
Link Failure Rate	The average number of link errors per second during the sample interval
Loss of Sync Rate	The average number of times per second that synchronization was lost during the sample interval
Loss of Signal Rate	The average number of times per second that the signal was lost during the sample interval

Metric	Definition
CRC Error Rate	The average number of frames received per second in which the cyclic redundancy check (CRC) in the frame did not match the CRC computed by the receiver during the sample interval
Port Send Bandwidth Percentage	The approximate bandwidth utilization percentage for send operations by a port based on its current negotiated speed
Port Receive Bandwidth Percentage	The approximate bandwidth utilization percentage for receive operations by this port, based on its current negotiated speed
Overall Port Bandwidth Percentage	The approximate bandwidth utilization percentage for send and receive operations by this port

The most important metrics for determining whether a SAN bottleneck exists are the Total Port Data Rate, and the Port Bandwidth Percentages (Send, Receive, Overall).

Tivoli Storage Productivity Center Topology Viewer

Tivoli Storage Productivity Center provides a tool called *Topology Viewer* for viewing the connectivity in an environment. Figure 7-49 shows an example of this feature. This example shows the connectivity between the host x346-tic-5 to the switch and from the switch the back-end connections. Additional detail can be provided by drilling down on each of the paths. Alerts for the component are indicated by a red exclamation point (!).

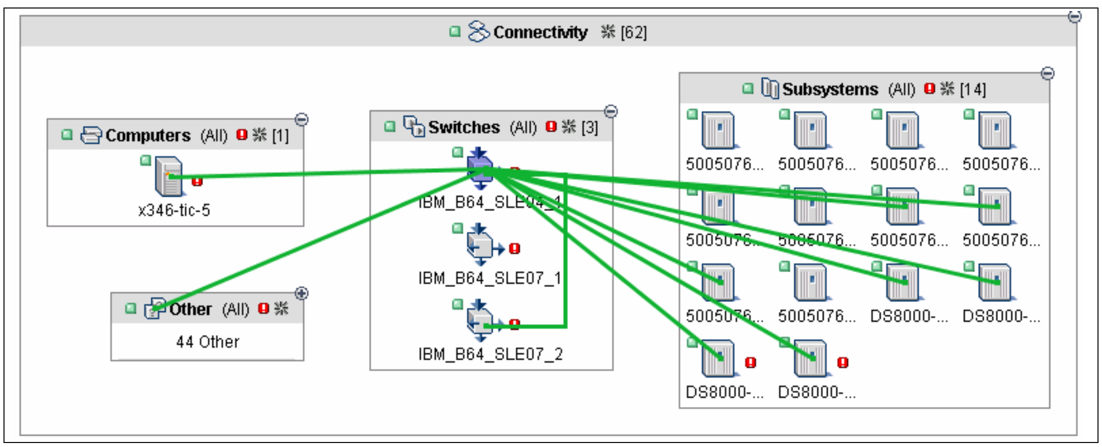


Figure 7-49 Tivoli Storage Productivity Center Topology Viewer

7.7 End-to-end analysis of I/O performance problems

To support tactical performance management processes, problem determination skills and processes must exist. In this section, we explain the logical required steps to perform successful problem determination for I/O performance issues. The process of I/O performance problem determination consists of the following logical steps:

- Define the problem.
- Classify the problem.
- Identify the I/O bottleneck.
- Implement changes to remove the I/O bottleneck.
- Validate that changes resolved the issue.

Perceived or actual I/O bottlenecks can result from hardware failures on the I/O path, contention on the server, contention on the SAN Fabric, contention on the DS8000 front-end ports, or contention on the back-end disk adapters or disk arrays. In this section, we provide a process for diagnosing these scenarios by using Tivoli Storage Productivity Center and external data. This process was developed for identifying specific types of problems and is not a substitute for common sense, knowledge of the environment, and experience. Figure 7-50 shows the high-level process flow.

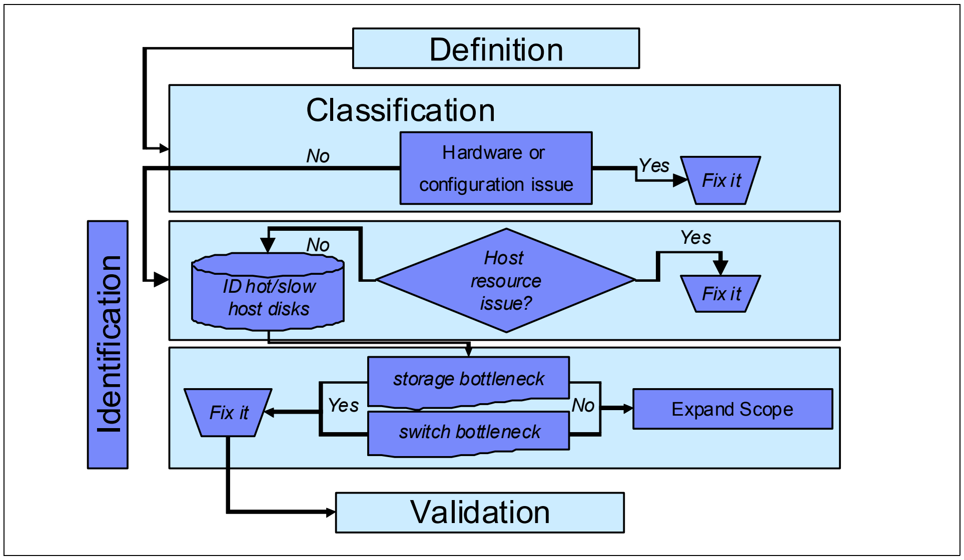


Figure 7-50 I/O performance analysis process

I/O bottlenecks as referenced in this section relate to one or more components on the I/O path that reached a saturation point and can no longer achieve the I/O performance requirements. I/O performance requirements are typically throughput-oriented or transaction-oriented. Heavy sequential workloads, such as tape backups or data warehouse environments, might require maximum bandwidth and use large sequential transfers. However, they might not have stringent response time requirements. Transaction-oriented workloads, such as online banking systems, might have stringent response time requirements but have no requirements for throughput.

If a server processor or memory resource shortage is identified, it is important to take the necessary remedial actions. These actions might include but are not limited to adding additional processors, optimizing processes or applications, or adding additional memory. In general, if there are not any resources constrained on the server but the end-to-end I/O response time is higher than expected for the DS8000 (see “Rules” on page 410), a resource constraint is likely in one or more of the SAN components.

In order to troubleshoot performance problems, Tivoli Storage Productivity Center Disk Manager and Fabric Manager data must be augmented with host performance and configuration data. Figure 7-51 on page 294 shows a logical end-to-end view from a measurement perspective.

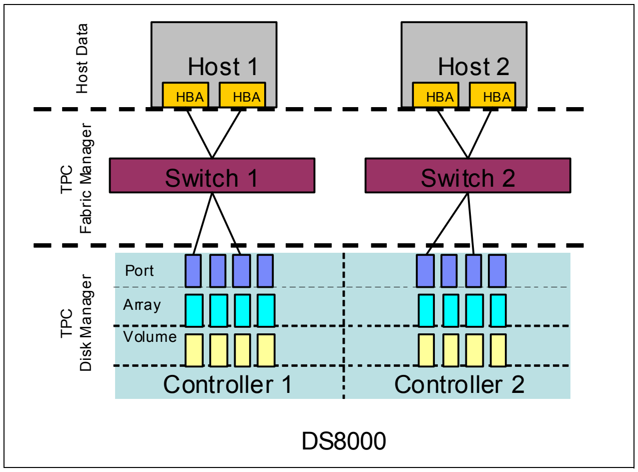


Figure 7-51 End-to-end measurement

As shown in Figure 7-51, Tivoli Storage Productivity Center does not provide host performance, configuration, or error data. Tivoli Storage Productivity Center Fabric Manager provides performance and error log information about SAN switches. Tivoli Storage Productivity Center Disk Manager provides the DS8000 storage performance and configuration information.

Tip: Performance analysis and troubleshooting must always start top-down, starting with the application (for example, database design and layout), then operating system, server hardware, SAN, and then storage. The tuning potential is greater at the “higher” levels. The best I/O tuning is never carried out, because server caching or a better database design eliminated the need for it.

Process assumptions

This process assumes that the following conditions exist:

- The server is connected to the DS8000 natively.
- Tools exist to collect the necessary performance and configuration data for each component along the I/O path (server disk, SAN fabric, and the DS8000 arrays, ports, and volumes).
- Skills exist to use the tools, extract data, and analyze data.
- Data is collected in a continuous fashion to facilitate performance management.

Process flow

The order in which you conduct the analysis is important. We suggest the following process:

1. Define the problem. A sample questionnaire is provided in “Sample questions for an AIX host” on page 638. The goal is to assist in determining the problem background and understand how the performance requirements are not being met.

   Changes: Before proceeding any further, ensure that adequate discovery is pursued to identify any changes that occurred in the environment. In our experience, there is a significant correlation between changes in the environment and sudden “unexpected” performance issues.

2. Consider checking the application level first. Has all potential tuning on the database level been performed? Does the layout adhere to the vendor recommendations, and is the server adequately sized (RAM, processor, and buses) and configured?
3. Correctly classify the problem by identifying hardware or configuration issues. Hardware failures often manifest themselves as performance issues, because I/O is degraded on one or more paths. If a hardware issue is identified, all problem determination efforts must focus on identifying the root cause of the hardware errors:
   a. Gather any errors on any of the host paths.

   > **Physical component:** If you notice significant errors in the “datapath query device” or the “pcmpath query device” and the errors increase, likely there is a problem with a physical component on the I/O path.

   b. Gather the host error report and look for Small Computer System Interface (SCSI) or FIBRE errors.

   > **Hardware:** Often a hardware error that relates to a component on the I/O path shows as a `TEMP` error. A TEMP error does not exclude a hardware failure. You must perform diagnostics on all hardware components in the I/O path, including the host bus adapter (HBA), SAN switch ports, and the DS8000 HBA ports.

   c. Gather the SAN switch configuration and errors. Every switch vendor provides different management software. All of the SAN switch software provides error monitoring and a way to identify whether there is a hardware failure with a port or application-specific integrated circuit (ASIC). For more information about identifying hardware failures, see your vendor-specific manuals or contact vendor support.

   > **Patterns:** As you move from the host to external resources, remember any patterns. A common error pattern that you see involves errors that affect only those paths on the same HBA. If both paths on the same HBA experience errors, the errors are a result of a common component. The common component is likely to be the host HBA, the cable from the host HBA to the SAN switch, or the SAN switch port. Ensure that all of these components are thoroughly reviewed before proceeding.

   d. If errors exist on one or more of the host paths, determine whether there are any DS8000 hardware errors. Log on to the HMC as *customer/cust0mer* and look to ensure that there are no hardware alerts. Figure 7-52 provides a sample of a healthy DS8000. If there are any errors, you might need to open a problem ticket (PMH) with DS8000 hardware support (2107 engineering).

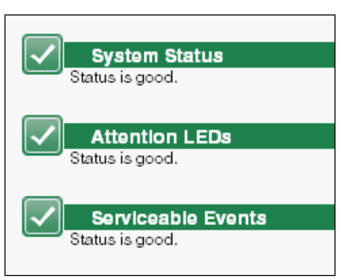


Figure 7-52 DS8000 healthy HMC

4. After validating that no hardware failures exist, analyze server performance data and identify any disk bottlenecks. The fundamental premise of this methodology is that I/O performance degradation that relates to SAN component contention can be observed at the server through analysis of the key server-based I/O metrics.

   Degraded end-to-end I/O response time is the strongest indication of I/O path contention. Typically, server physical disk response times measure the time that a physical I/O request takes from the moment that the request was initiated by the device driver until the device driver receives an interrupt from the controller that the I/O completed. The measurements are displayed as either service time or response time. They are usually averaged over the measurement interval. Typically, server wait or queue metrics refer to time spent waiting at the HBA, which is usually an indication of HBA saturation. In general, you need to interpret the service times as response times, because they include potential queuing at various storage subsystem components, for example:

   – Switch
   – Storage HBA
   – Storage cache
   – Storage back-end disk controller
   – Storage back-end paths
   – Disk drives

   Important: Subsystem-specific load-balancing software usually does not add any performance overhead and can be viewed as a pass-through layer.

   In addition to the disk response time and disk queuing data, gather the disk activity rates, including read I/Os, write I/Os, and total I/Os, because they show which disks are active:

   a. Gather performance data as shown in Table 7-12.

Table 7-12 Native tools and key metrics

OS	Native tool	Command/Object	Metric/Counter
AIX	iostat, filemon	`iostat -D`, `filemon -o /tmp/fmon.log -O all`	read time(ms) write time(ms) reads, writes queue length
HP-UX	sar	`sar -d`	avserv(ms) avque blks/s
Linux	iostat	`iostat -d`	svctm(ms) avgqu-sz tps
Solaris	iostat	`iostat -xn`	svc_t(ms) Avque blks/s
Microsoft Windows Server	perfmon	Physical Disk	Avg Disk Sec/Read Avg Disk Sec/Write Read Disk Queue Length Write Disk Queue Length Disk Reads/sec Disk Writes/sec

OS	Native tool	Command/Object	Metric/Counter
System z	Resource Measurement Facility (RMF)/System Management Facilities (SMF)	Chapter 14, "Performance considerations for System z servers" on page 499	N/A

I/O-intensive disks: The number of total I/Os per second indicates the relative activity of the device. This relative activity provides a metric to prioritize the analysis. Those devices with high response times and high activity are more important to understand than devices with high response time and infrequent access. If analyzing the data in a spreadsheet, consider creating a combined metric of Average I/Os × Average Response Time to provide a method for identifying the most I/O-intensive disks. You can obtain additional detail about OS-specific server analysis in the OS-specific chapters.

b. Gather configuration data (Subsystem Device Driver (SDD)/Subsystem Device Driver Path Control Module (SDDPCM)) as shown in Table 7-13. In addition to the multipathing configuration data, you need to collect configuration information for the host and DS8000 HBAs that includes the bandwidth of each adapter.

Table 7-13 Path configuration data

OS	Tool	Command	Key	Other information
All UNIX	SDD/SDDPCM	datapath query essmap pcmpath query essmap	LUNserial	Rank[a], logical subsystem (LSS), Storage subsystem
Windows	SDD/Subsystem Device Driver Device Specific Module (SDDDSM)	datapath query essmap	LUN serial	Rank[a], LSS, Storage subsystem

a. The rank column is not meaningful for multi-rank extent pools on the DS8000.

Multipathing: Ensure that multipathing works as designed. For example, if there are two paths zoned per HBA to the DS8000, there must be four active paths per LUN. Both SDD and SDDPCM use an active/active configuration of multipathing, which means that traffic flows across all the traffic fairly evenly. For native DS8000 connections, the absence of activity on one or more paths indicates a problem with the SDD behavior.

c. Format the data and correlate the host LUNs with their associated DS8000 resources. Formatting the data is not required for analysis, but it is easier to analyze formatted data in a spreadsheet.

The following steps represent the logical steps that are required to format the data and do not represent literal steps. You can codify these steps in scripts. You can obtain examples of these scripts in Appendix E, "Post-processing scripts" on page 677:

i. Read the configuration file.

    ii. Build hdisk hash with key = hdisk and value = LUN SN.

    iii. Read I/O response time data.

    iv. Create hashes for each of the following values with hdisk as the key: Date, Start time, Physical Volume, Reads, Avg Read Time, Avg Read Size, Writes, Avg Write Time, and Avg Write Size.

    v. Print the data to a file with headers and commas to separate the fields.

    vi. Iterate through hdisk hash and use the common hdisk key to index into the other hashes and print those hashes that have values.

  d. Analyze the host performance data:

    i. Determine whether I/O bottlenecks exist by summarizing the data and analyzing key performance metrics for values in excess of the thresholds described in "Rules" on page 410. Identify those vpaths/LUNs with poor response time. We show an example in 10.8.6, "Analyzing performance data" on page 419. Hardware errors and multipathing configuration issues must already be excluded. The hot LUNs must already be identified. Proceed to step 5 to determine the root cause of the performance issue.

    ii. If no degraded disk response times exist, the issue is likely not internal to the server.

5. If there are disk constraints identified, continue the identification of the root cause by collecting and analyzing the DS8000 configuration and performance data:

  a. Gather the configuration information. Tivoli Storage Productivity Center can also be used to gather configuration data through the topology viewer or from selecting **Data Manager** → **Reporting** → **Asset** → **By Storage Subsystem** as shown in Figure 7-53.

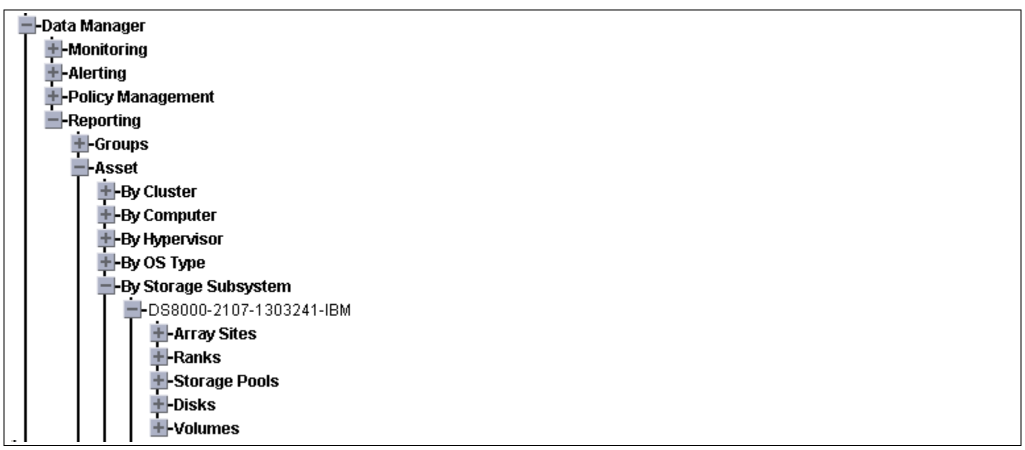


Figure 7-53 By Storage Subsystem Asset report

> **Analyze the DS8000 performance data first:** Analysis of the SAN fabric and the DS8000 performance data can be completed in either order. However, SAN bottlenecks occur less frequently than disk bottlenecks, so it can be more efficient to analyze the DS8000 performance data first.

  b. Use Tivoli Storage Productivity Center to gather the DS8000 performance data for subsystem ports, arrays, and volumes. Compare the key performance indicators from Table 7-7 on page 257 with the performance data. Follow these steps to analyze the performance:

i. For those server LUNs that show poor response time, analyze the associated volumes during the same period. If the problem is on the DS8000, a correlation exists between the high response times observed on the host and the volume response times observed on the DS8000.

> **Compare the same time period:** Meaningful correlation with the host performance measurement and the previously identified hot LUNs requires analysis of the DS8000 performance data for the same time period that the host data was collected. For more information about time stamps, see 7.3.1, “Time stamps” on page 250.

ii. Correlate the hot LUNs with their associated disk arrays. When using the Tivoli Storage Productivity Center GUI, the relationships are provided automatically in the drill-down feature. If using batch exports and you want to correlate the volume data to the rank data, you can correlate the volume data to the rank data manually or by using the script. If multiple ranks per extent pool and storage pool striping, or Easy Tier managed pools are used, one volume can exist on multiple ranks.

iii. Analyze storage subsystem ports for the ports associated with the server in question.

6. Continue the identification of the root cause by collecting and analyzing SAN fabric configuration and performance data:

   a. Gather the connectivity information and establish a visual diagram of the environment. If you use the Tivoli Storage Productivity Center Fabric Manager, you can use the Topology Viewer to quickly create a visual representation of your SAN environment as shown in Figure 7-49 on page 292.

   > **Visualize the environment:** Sophisticated tools are not necessary for creating this type of view; however, the configuration, zoning, and connectivity information must be available to create a logical visual representation of the environment.

   b. Gather the SAN performance data. Each vendor provides SAN management applications that provide the alerting capability and some level of performance management. Often, the performance management software is limited to real-time monitoring and historical data collection features require additional licenses. In addition to the vendor-provided solutions, Tivoli Storage Productivity Center Fabric Manager can collect further metrics that are shown in Table 7-11 on page 291.

   c. Consider graphing the Overall Port Response Time, Port Bandwidth Percentage, and Total Port Data Rate metrics to determine whether any of the ports along the I/O path are saturated during the time when the response time is degraded. If the Total Port Data Rate is close to the maximum expected throughput for the link or the bandwidth percentages that exceed their thresholds, this situation is likely a contention point. You can add additional bandwidth to mitigate this type of issue either by adding additional links or by adding faster links. Adding links might require upgrades of the server HBAs and the DS8000 host adapters to take advantage of the additional switch link capacity.

In addition to the ability to create ad hoc reports by using Tivoli Storage Productivity Center Fabric Manager metrics, Tivoli Storage Productivity Center provides predefined reports. Select **IBM Tivoli Storage Productivity Center** → **Reporting** → **System Reports** → **Fabric** or **Disk**.

7.7.1 Performance analysis examples

This section provides sample performance data, analysis, and recommendations for the following performance scenarios by using the process described in 7.7, "End-to-end analysis of I/O performance problems" on page 292. The examples highlight the key performance data that is appropriate for each problem type. We provide the host configuration or errors only in the cases where that information is critical to determine the outcome.

DS8000 disk array bottleneck example

The most common type of performance problem is a disk array bottleneck. Similar to other types of I/O performance problems, a disk array bottleneck usually manifests itself in high disk response time on the host. In many cases, the write response times are excellent due to cache hits, but reads often require immediate disk access.

Problem definition

The application owner complains of poor response time for transactions during certain times of the day.

Problem classification

There are no hardware errors, configuration issues, or host performance constraints.

Identification

Figure 7-54 shows the average read response time for a Windows Server 2008 server that performs a random workload in which the response time increases steadily over time.

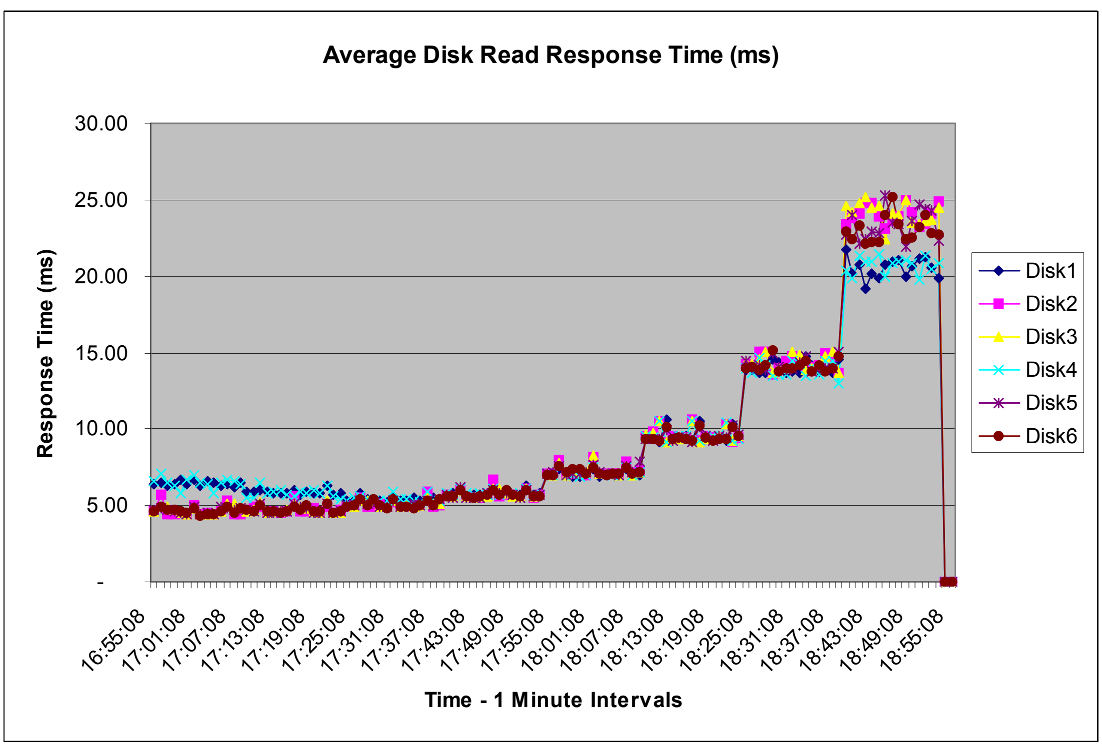


Figure 7-54 Windows Server perfmon average physical disk read response time

At approximately 18:39 hours, the average read response time jumps from approximately 15 ms to 25 ms. Further investigation on the host reveals that the increase in response time correlates with an increase in load as shown in Figure 7-55.

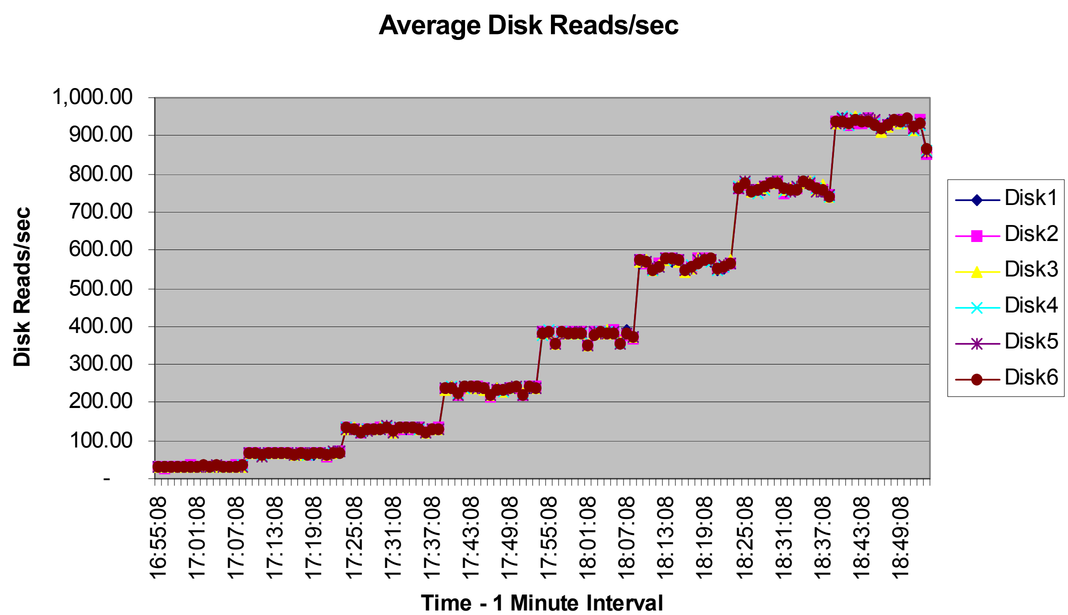


Figure 7-55 Average Disk Reads/sec

As described in 7.7, “End-to-end analysis of I/O performance problems” on page 292, there are several possibilities for high average disk read response time:

- DS8000 array contention
- DS8000 port contention
- SAN fabric contention
- Host HBA saturation

Because the most probable reason for the elevated response times is the disk utilization on the array, gather and analyze this metric first. Figure 7-56 on page 302 shows the disk utilization on the DS8000.

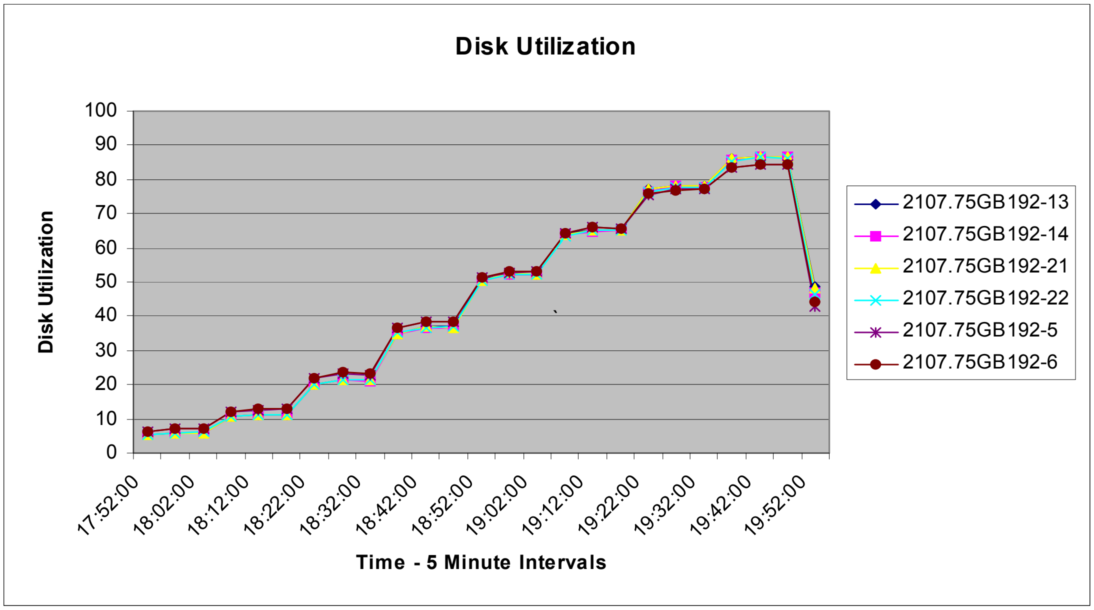


Figure 7-56 Tivoli Storage Productivity Center array disk utilization

Recommend changes

We recommend that you add volumes on additional disks. For environments where host striping is configured, you might need to re-create the host volumes to spread the I/O from an existing workload across the new volumes.

Validate the problem resolution

Gather performance data to determine whether the issue is resolved.

Hardware connectivity example one

Infrequent connectivity issues occur as a result of broken or damaged components in the I/O path. The following example illustrates the required steps to identify and resolve these types of issues.

Problem definition

The online transactions for a Windows Server 2008 SQL server appear to take longer than normal and time out in certain cases.

Problem classification

After reviewing the hardware configuration and the error reports for all hardware components, we determined that there are errors on the paths associated with one of the host HBAs, as shown in Figure 7-57 on page 303. This output shows the errors on path 0 and path 1, which are both on the same HBA (SCSI port 1). For a Windows Server 2008 server that runs SDDDSM, additional information about the host adapters is available through the **`gethba.exe`** command. The command that you use to identify errors depends on the multipathing software installation.

```
DEV#:   4  DEVICE NAME: Disk6 Part0  TYPE: 2107900    POLICY: OPTIMIZED
SERIAL: 75GB1924814
============================================================================
Path#            Adapter/Hard Disk         State  Mode      Select     Errors
    0     Scsi Port1 Bus0/Disk6 Part0     CLOSE  NORMAL    5284328       2366
    1     Scsi Port1 Bus0/Disk6 Part0     CLOSE  NORMAL    5289305       2276
    2     Scsi Port2 Bus0/Disk6 Part0     CLOSE  NORMAL    5310124          1
    3     Scsi Port2 Bus0/Disk6 Part0     CLOSE  NORMAL    5304164          2

DEV#:   5  DEVICE NAME: Disk7 Part0  TYPE: 2107900    POLICY: OPTIMIZED
SERIAL: 75GB1924815
============================================================================
Path#            Adapter/Hard Disk         State  Mode      Select     Errors
    0     Scsi Port1 Bus0/Disk7 Part0     CLOSE  NORMAL    5285870       2331
    1     Scsi Port1 Bus0/Disk7 Part0     CLOSE  NORMAL    5287774       2308
    2     Scsi Port2 Bus0/Disk7 Part0     CLOSE  NORMAL    5302395          1
    3     Scsi Port2 Bus0/Disk7 Part0     CLOSE  NORMAL    5306109          1
```

Figure 7-57 Example of datapath query device

Identify the root cause

A further review of the switch software revealed significant errors on the switch port associated with the paths in question. A visual inspection of the environment revealed a kink in the cable from the host to the switch.

Implement changes to resolve the problem

Replace the cable.

Validate the problem resolution

After we implement the change, the error counts do not increase and the nightly backups complete within the backup window.

Hardware connectivity example two

Infrequent connectivity issues occur as a result of broken or damaged components in the I/O path. The following example illustrates the required steps to identify and resolve these types of issues.

Define the problem

Users report that the data warehouse application on an AIX server does not complete jobs in a reasonable amount of time. Online transactions also time out.

Classify the problem

A review of the host error log shows a significant number of hardware errors. We provide an example of the errors in Figure 7-58 on page 304.

```
---------------------------------------------------------------------------
LABEL:          FSCSI_ERR4
IDENTIFIER:     3074FEB7

Date/Time:       Tue Nov 27 19:36:05 CST 2007
Sequence Number: 285162
Machine Id:      00C13E9E4C00
Node Id:         G01010VS1000
Class:           H
Type:            TEMP
Resource Name:   fscsi2
Resource Class:  driver
Resource Type:   efscsi
Location:        U5791.001.9920BH0-P2-C03-T1

Description
ADAPTER ERROR

Probable Causes
ADAPTER HARDWARE OR CABLE
ADAPTER MICROCODE
FIBRE CHANNEL SWITCH OR FC-AL HUB
Standard input_
```

Figure 7-58 AIX error log

Identify the root cause

The IBM service support representative (SSR) ran the IBM diagnostics tests on the host HBA, and the card did not pass the diagnostics tests.

> **Disabling a path:** In the cases where there is a path with significant errors, you can disable the path with the multipathing software, which allows the non-working paths to be disabled without causing performance degradation to the working paths. With SDD, disable the path by using **`datapath set device # path # offline`**.

Implement changes to resolve the problem

Replace the card.

Validate the problem resolution

The errors did not persist after the card is replaced and the paths are brought online.

Performance analysis example: DS8000 port bottleneck

The DS8000 port bottlenecks do not occur often, but they are a component that is typically oversubscribed.

Define the problem

The production server batch runs exceed their batch window.

Classify the problem

There are no hardware errors, configuration issues, or host performance constraints.

Identify the root cause

The production server throughput diminishes at approximately 18:30 hours daily. At the same time, development workloads that run on the same DS8000 ports increase. Figure 7-59 on page 305 demonstrates the overall workload from both the production server and the development server.

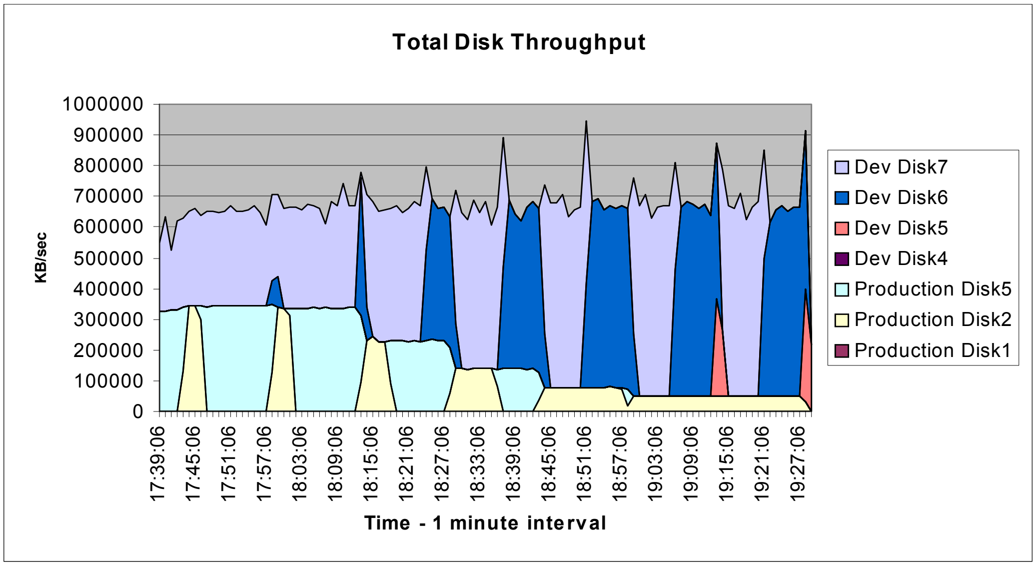


Figure 7-59 Production throughput compared to development throughput

The DS8000 port data reveals a peak throughput of around 300 MBps per 4-Gbps port.

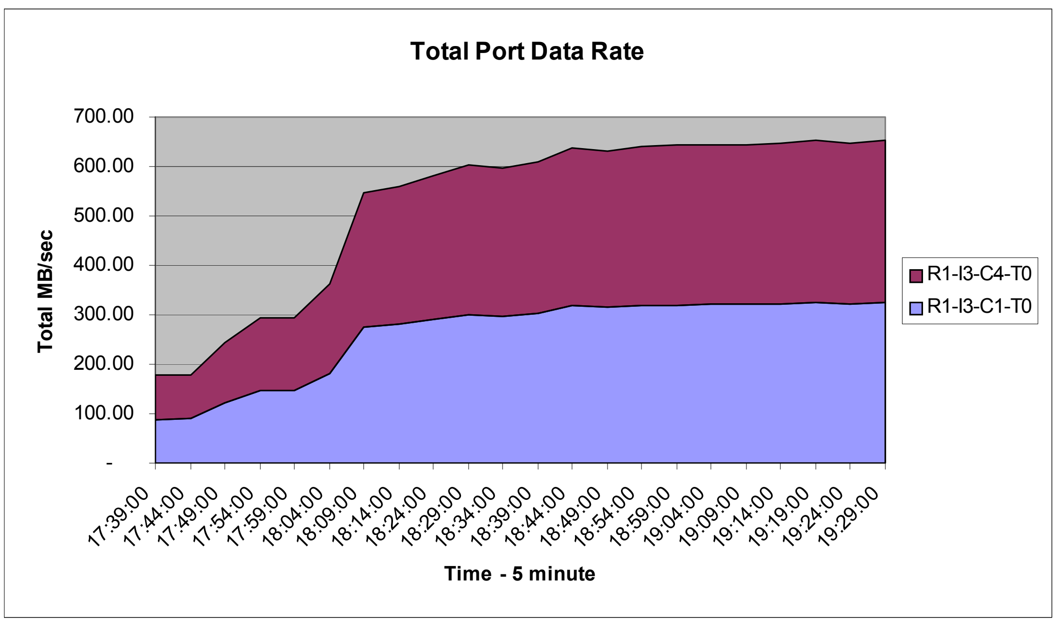


Figure 7-60 Total port data rate

Implement changes to resolve the problem

Rezone ports for production servers and development servers so that they do not use the same DS8000 ports. Add additional ports so that each server HBA is zoned to two DS8000 ports.

Validate the problem resolution

After implementing the new zoning that separated the production server and the development server, the storage ports are no longer the bottleneck.

Performance analysis example: Server HBA bottleneck

Although rare, server HBA bottlenecks occur, usually as result of a highly sequential workload with under-configured HBAs. We describe an example of the type of workload and configuration that lead to this type of problem in 10.8.6, "Analyzing performance data" on page 419.

7.8 Tivoli Storage Productivity Center for Disk in mixed environments

A benefit of IBM Tivoli Storage Productivity Center for Disk is the capability to analyze both Open Systems fixed block (FB) and System z Count Key Data (CKD) workloads. When the DS8000 subsystems are attached to multiple hosts that run on different platforms, Open Systems hosts might affect your System z workload, and the System z workload might affect the Open Systems workloads. If you use a mixed environment, looking at the RMF reports is insufficient. You also need the information about the Open Systems. The IBM Tivoli Storage Productivity Center for Disk informs you about the cache and I/O activity.

Before beginning the diagnostic process, you must understand your workload and your physical configuration. You need to know how your system resources are allocated, as well as understand your path and channel configuration for all attached servers.

Assume that you have an environment with a DS8000 attached to a z/OS host, an AIX Power Systems host, and several Windows Server 2008 hosts. You noticed that your z/OS online users experience a performance degradation between 07:30 and 08:00 hours each morning.

You might notice that there are 3390 volumes that indicate high *disconnect* times, or high *device busy delay* time for several volumes in the RMF device activity reports. Unlike UNIX or Windows Server 2008, you might notice *response time* and its breakdown to *connect, disconnect, pending*, and IOS *queuing*.

Disconnect time is an indication of cache-miss activity or destage wait (due to persistent memory high utilization) for logical disks behind the DS8000s.

Device busy delay is an indication that another system locks up a volume, and an extent *conflict* occurs among z/OS hosts or applications in the same host when using Parallel Access Volumes (PAVs). The DS8000 multiple allegiance or PAVs capability allows it to process multiple I/Os against the same volume at the same time. However, if a read or write request against an extent is pending while another I/O is writing to the extent, or if a write request against an extent is pending while another I/O is reading or writing data from the extent, the DS8000 delays the I/O by queuing. This condition is referred as extent *conflict.* Queuing time due to extent conflict is accumulated to *device busy* (DB) *delay* time. An *extent* is a sphere of access; the unit of increment is a *track*. Usually, I/O drivers or system routines decide and declare the sphere.

To determine the possible cause of high *disconnect* times, check the read cache hit ratios, read-to-write ratios, and bypass I/Os for those volumes. If you see that the cache hit ratio is lower than usual and you did not add other workload to your System z environment, I/Os against Open Systems FB volumes might be the cause of the problem. Possibly, FB volumes that are defined on the same server have a cache-unfriendly workload, thus affecting your System z volumes hit ratio.

To get more information about cache usage, you can check the cache statistics of the FB volumes that belong to the same server. You might be able to identify the FB volumes that have a low read hit ratio and short cache holding time. Moving the workload of these Open Systems logical disks or the System z CKD volumes about which you are concerned to the other side of the cluster, so that you can concentrate cache-friendly I/O workload to either cluster, improves the situation. If you cannot or if the condition does not improve after this move, consider balancing the I/O distribution on more ranks, or solid-state drives (SSDs). Balancing the I/O distribution on more ranks optimizes the staging and destaging operation.

The approaches that use IBM Tivoli Storage Productivity Center for Disk as described in this chapter might not cover all the possible situations that you can encounter. You might need to include more information, such as application and host operating system-based performance statistics, the STAT reports, or other data collections to analyze and solve a specific performance problem. But if you have a basic understanding of how to interpret the DS8000 performance reports and how the DS8000 works, you can develop your own ideas about how to correlate the DS8000 performance reports with other performance measurement tools when you approach specific situations in your production environment.

Part 3

Performance considerations for host systems

This part provides performance considerations for various host systems or appliances that are attached to the IBM System Storage DS8000 system.

This part includes the following topics:

- Host attachment
- Performance considerations for UNIX servers
- Performance considerations for Microsoft Windows servers
- Performance considerations for VMware
- Performance considerations for Linux
- Performance considerations for IBM i
- Performance considerations for System z servers
- IBM System Storage SAN Volume Controller attachment
- IBM ProtecTIER deduplication

8

Host attachment

This chapter describes the following attachment topics and considerations between host systems and the DS8000 series for availability and performance:

- DS8000 attachment types
- Attaching Open Systems hosts
- Attaching System z hosts

We provide detailed information about performance tuning considerations for specific operating systems in later chapters of this book.

8.1 DS8000 host attachment

The DS8000 enterprise storage solution provides various host attachments that allow exceptional performance and superior data throughput. We suggest a minimum of two connections to any host, and the connections need to be on different host adapters in different I/O enclosures. You can consolidate storage capacity and workloads for Open Systems hosts and IBM System z hosts by using the following adapter types and protocols:

- Fibre Channel Protocol (FCP)-attached Open Systems hosts
- Fibre Channel Protocol (FCP)/Fibre Connection (FICON)-attached System z hosts

The DS8800 4-way standard model supports a maximum of 16 FCP/FICON host adapters with four or eight ports each, so a maximum of 128 FCP/FICON ports are supported. The DS8800 2-way standard and Business Class models support a maximum of four FCP/FICON host adapters with four or eight ports each, so a maximum of 32 FCP/FICON ports are supported. All the ports can be intermixed and independently configured. The DS8800 host adapters support 2, 4, or 8 Gbps link speed, but 1 Gbps is no longer supported. Enterprise Systems Connection (ESCON®) adapters are not supported in DS8800.

The DS8000 can support host systems and remote mirroring links by using Peer-to-Peer Remote Copy (PPRC) on the same I/O port. However, we advise that you use dedicated I/O ports for remote mirroring links.

Planning and sizing the host adapters for performance are not easy tasks and we strongly suggest that you use modeling tools, such as Disk Magic (see 6.1, “Disk Magic” on page 176). The factors that might affect the performance at the host adapter level are typically the aggregate throughput and the workload mix that the adapter can handle. All connections on a host adapter share bandwidth in a balanced manner. Therefore, host attachments that require maximum I/O port performance must be connected to HAs that are not fully populated. You must allocate host connections across I/O ports, host adapters, and I/O enclosures in a balanced manner (workload spreading).

8.2 Attaching Open Systems hosts

This section describes the host system requirements and attachment considerations for Open Systems hosts that run AIX, Linux, Hewlett-Packard UNIX (HP-UX), Novell, Oracle Solaris, and Microsoft Windows to the DS8000 series with Fibre Channel adapters.

> **No SCSI:** There is no direct Small Computer System Interface (SCSI) attachment support for the DS8000.

8.2.1 Fibre Channel

Fibre Channel is a 100, 200, 400, and 800 MBps, full-duplex, serial communications technology to interconnect I/O devices and host systems that might be separated by tens of kilometers. The DS8800 supports 8, 4, and 2 Gbps connections and it negotiates the link speed automatically.

Supported Fibre Channel-attached hosts

For specific considerations that apply to each server platform, as well as for the *most current* information about supported servers (the list is updated periodically), see this website:

http://www.ibm.com/systems/support/storage/config/ssic

Fibre Channel topologies

The DS8000 architecture supports all three Fibre Channel interconnection topologies:

- Direct connect
- Arbitrated loop
- Switched fabric

For maximum flexibility and performance, we suggest a switched fabric topology.

We describe recommendations for implementing a switched fabric in more detail in the next section.

8.2.2 Storage area network implementations

In this section, we describe a basic storage area network (SAN) network and how to implement it for maximum performance and availability. We show examples of a correctly connected SAN network to maximize the throughput of disk I/O.

Description and characteristics of a SAN

With a SAN, you can connect heterogeneous Open Systems servers to a high-speed network and share storage devices, such as disk storage and tape libraries. Instead of each server having its own locally attached storage and tape drives, a SAN shares centralized storage components and you can easily allocate storage to hosts.

SAN cabling for availability and performance

For availability and performance, you need to connect to different adapters in different I/O enclosures whenever possible. You must use multiple Fibre Channel switches or directors to avoid a potential single point of failure. You can use inter-switch links (ISLs) for connectivity.

Importance of establishing zones

For Fibre Channel attachments in a SAN, it is important to establish *zones* to prevent interaction from host adapters. Every time that a host adapter joins the fabric, it issues a Registered State Change Notification (RSCN). An RSCN does not cross zone boundaries, but it affects every device or host adapter in the same zone.

If a host adapter fails and starts logging in and out of the switched fabric, or a server must be rebooted several times, you do not want it to disturb the I/O to other hosts. Figure 8-1 on page 315 shows zones that include only a single host adapter and multiple DS8000 ports (*single initiator zone*). This approach is the suggested way to create zones to prevent interaction between server host adapters.

> **Tip:** Each zone contains a single host system adapter with the desired number of ports attached to the DS8000.

By establishing zones, you reduce the possibility of interactions between system adapters in switched configurations. You can establish the zones by using either of two zoning methods:

- Port number
- Worldwide port name (WWPN)

You can configure switch ports that are attached to the DS8000 in more than one zone, which enables multiple host system adapters to share access to the DS8000 host adapter ports. Shared access to a DS8000 host adapter port might be from host platforms that support a combination of bus adapter types and operating systems.

Important: A DS8000 host adapter port configured to run with the FICON topology cannot be shared in a zone with non-z/OS hosts. Ports with non FICON topology cannot be shared in a zone with z/OS hosts.

LUN masking

In Fibre Channel attachment, logical unit number (LUN) affinity is based on the *worldwide port name* (WWPN) of the adapter on the host, which is independent of the DS8000 host adapter port to which the host is attached. This *LUN masking* function on the DS8000 is provided through the definition of DS8000 volume groups. A *volume group* is defined by using the DS Storage Manager or DS8000 command-line interface (DSCLI), and host WWPNs are connected to the volume group. The LUNs to be accessed by the hosts that are connected to the volume group are defined to reside in that volume group.

Although it is possible to limit through *which* DS8000 host adapter ports a certain WWPN connects to volume groups, we suggest that you define the WWPNs to have access to all available DS8000 host adapter ports. Then, by using the recommended process of creating Fibre Channel zones as discussed in "Importance of establishing zones" on page 313, you can limit the desired host adapter ports through the Fibre Channel zones. In a switched fabric with multiple connections to the DS8000, this concept of LUN affinity enables the host to see the same LUNs on different paths.

Configuring logical disks in a SAN

In a SAN, carefully plan the configuration to prevent many disk device images from being presented to the attached hosts. Presenting many disk devices to a host can cause longer failover times in cluster environments. Also, boot times can take longer, because the device discovery steps take longer.

The number of times that a DS8000 logical disk is presented as a disk device to an open host depends on the number of paths from each host adapter to the DS8000. The number of paths from an open server to the DS8000 is determined by these factors:

- The number of host adapters installed in the server
- The number of connections between the SAN switches and the DS8000
- The zone definitions created by the SAN switch software

Physical paths: Each physical path to a logical disk on the DS8000 is presented to the host operating system as a disk device.

Consider a SAN configuration as shown in Figure 8-1 on page 315:

- The host has two connections to the SAN switches, and each SAN switch in turn has four connections to the DS8000.
- Zone A includes one Fibre Channel card (FC0) and two paths from SAN switch A to the DS8000.
- Zone B includes one Fibre Channel card (FC1) and two paths from SAN switch B to the DS8000.
- This host uses only four of the eight possible paths to the DS8000 in this zoning configuration.

By cabling the SAN components and creating zones as shown in Figure 8-1 on page 315, each logical disk on the DS8000 is presented to the host server four times, because there are four unique physical paths from the host to the DS8000. As you can see the picture, Zone A shows that FC0 has access through DS8000 host ports I0000 and I0130. Zone B shows that

FC1 has access through DS8000 host ports I0230 and I0300. So, in combination, this configuration provides four paths to each logical disk presented by the DS8000. If Zone A and Zone B are modified to include four paths each to the DS8000, the host has a total of eight paths to the DS8000. In that case, each logical disk assigned to the host is presented as eight physical disks to the host operating system. Additional DS8000 paths are shown as connected to Switch A and Switch B, but they are not in use for this example.

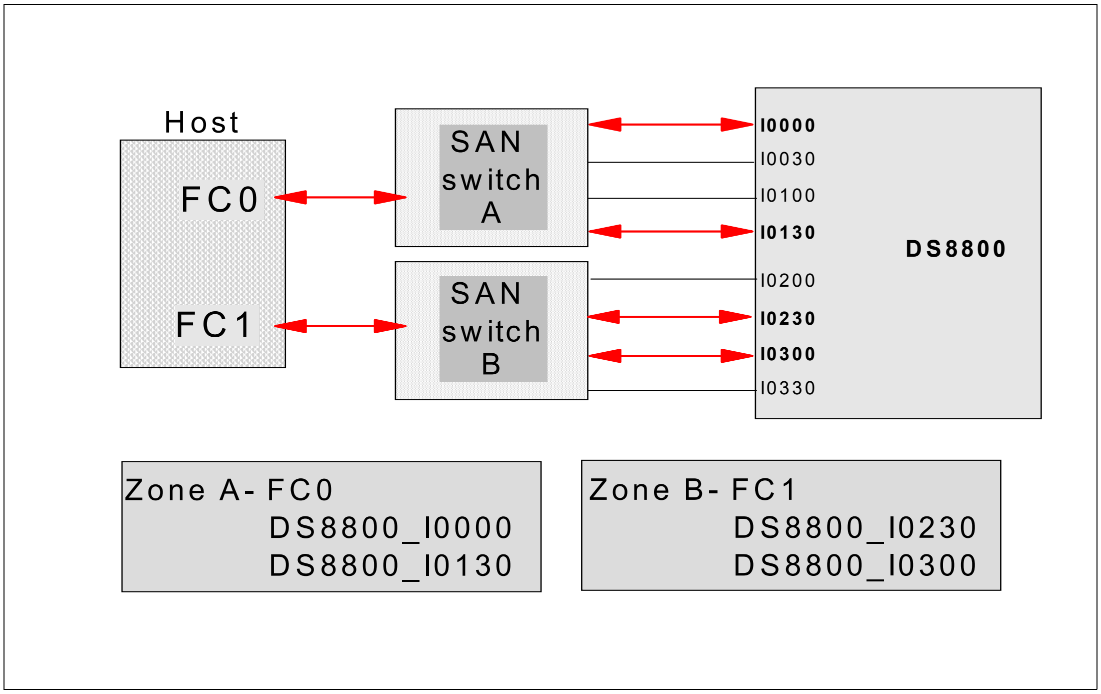


Figure 8-1 Zoning in a SAN environment

You can see how the number of logical devices presented to a host can increase rapidly in a SAN environment if you are not careful about selecting the size of logical disks and the number of paths from the host to the DS8000.

Typically, we suggest that you cable the switches and create zones in the SAN switch software for dual-attached hosts so that each server host adapter has two to four paths from the switch to the DS8000. With hosts configured this way, you can allow the multipathing module to balance the load across the four host adapters in the DS8000.

Zoning more paths, such as eight connections from the host to the DS8000, generally does not improve SAN performance and causes twice as many devices to be presented to the operating system.

8.2.3 Multipathing

Multipathing describes a technique to attach one host to an external storage device through more than one path. Multipathing can improve fault-tolerance and the performance of the overall system, because the fault of a single component in the environment can be tolerated without an impact to the host. Also, you can increase the overall system bandwidth, which positively influences the performance of the system.

As illustrated in Figure 8-2 on page 316, attaching a host system by using a single-path connection implements a solution that depends on several single points of failure. In this example, as a single link, failure either between the host system and the switch, between the switch and the storage system, or a failure of the host adapter in the host system, the DS8000 storage system or even a failure of the switch leads to a loss of access of the host system. Additionally, the path performance of the whole system is reduced by the slowest component in the link.

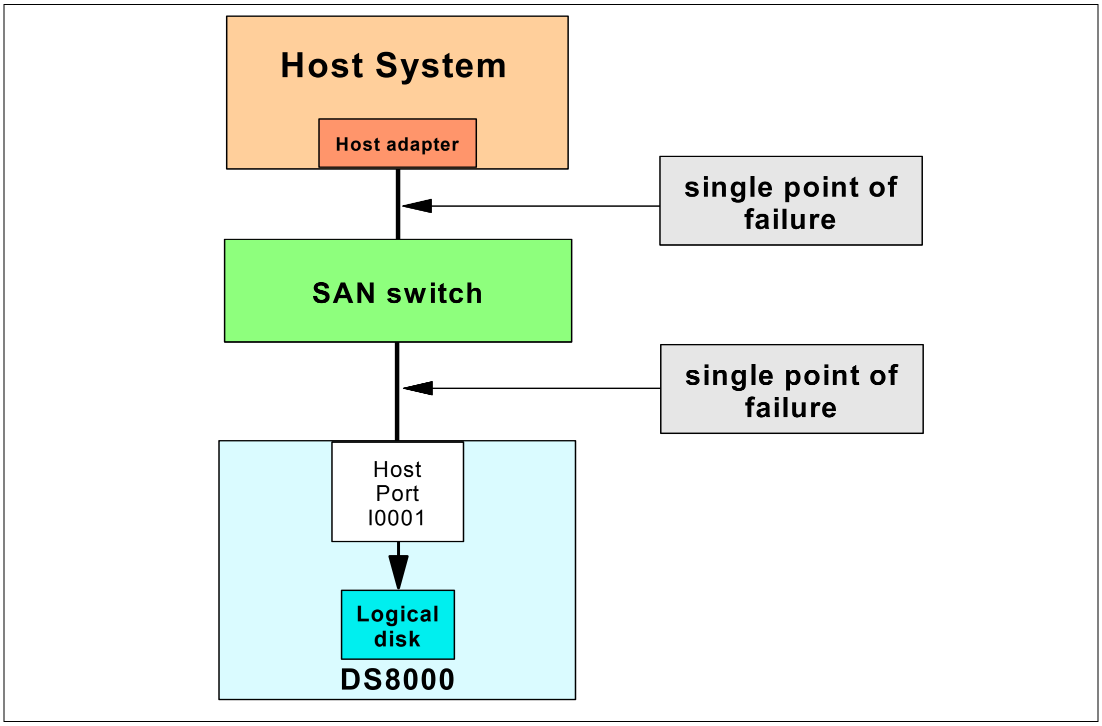


Figure 8-2 SAN single-path connection

Adding additional paths requires you to use multipathing software (Figure 8-3 on page 317). Otherwise, the same LUN behind each path is handled as a separate disk from the operating system side, which does not allow failover support.

Multipathing provides the DS8000 attached Open Systems hosts that run Windows, AIX, HP-UX, Oracle Solaris, or Linux with these capabilities:

- Support for several paths per LUN
- Load balancing between multiple paths when there is more than one path from a host server to the DS8000. This approach might eliminate I/O bottlenecks that occur when many I/O operations are directed to common devices via the same I/O path, thus improving the I/O performance.
- Automatic path management, failover protection, and enhanced data availability for users that have more than one path from a host server to the DS8000. It eliminates a potential single point of failure by automatically rerouting I/O operations to the remaining active paths from a failed data path.
- Dynamic reconsideration after changing the configuration environment, including zoning, LUN masking, and adding or removing physical paths.

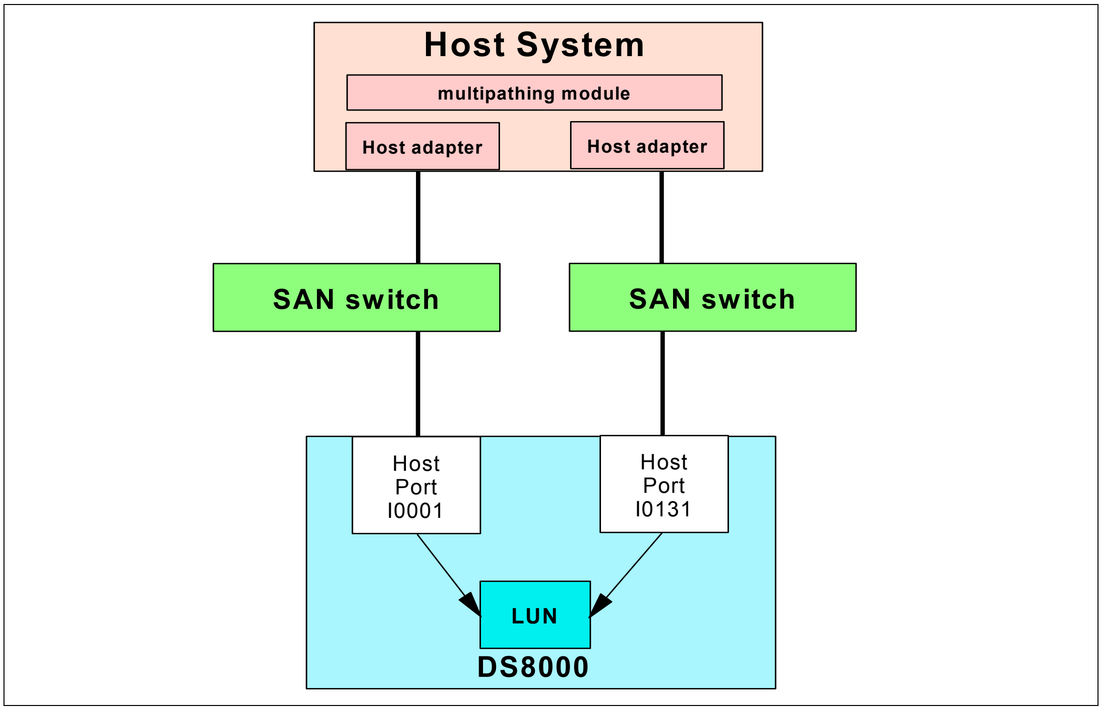


Figure 8-3 DS8000 multipathing implementation that uses two paths

The DS8000 supports several multipathing implementations. Depending on the environment, host type, and operating system, only a subset of the multipathing implementations is available. This section introduces the multipathing concepts and provides general information about implementation, usage, and specific benefits.

Important: Do not intermix several multipathing solutions within one host system. Usually, the multipathing software solutions cannot coexist.

Subsystem Device Driver

The IBM Subsystem Device Driver (SDD) software is a generic host-resident pseudo device driver that is designed to support the multipath configuration environments in the DS8000. The SDD resides on the host system with the native disk device driver and manages redundant connections between the host server and the DS8000. The solid-state drive (SSD) provides enhanced performance and data availability. The SDD is provided by and maintained by IBM for the AIX, Linux, HP-UX, Oracle Solaris, and Windows host operating systems.

The Subsystem Device Driver can operate under different modes or configurations:

- Concurrent data access mode: A system configuration where simultaneous access to data on common LUNs by more than one host is controlled by system application software. Examples are Oracle Parallel Server or file access software that can handle address conflicts. The LUN is not involved in access resolution.
- Non-concurrent data access mode: A system configuration where there is no inherent system software control of simultaneous accesses to the data on a common LUN by more than one host. Therefore, access conflicts must be controlled at the LUN level by a hardware-locking facility, such as Small Computer System Interface (SCSI) Reserve/Release.

> **Persistent Reserve:** Do not share LUNs among multiple hosts without the protection of Persistent Reserve (PR). If you share LUNs among hosts without PR, you are exposed to data corruption situations. You must also use PR when using FlashCopy.

The IBM Subsystem Device Driver does not support booting from or placing a system primary paging device on an SDD pseudo device.

For certain servers that run AIX, booting off the DS8000 is supported. In that case, LUNs used for booting are manually excluded from the SDD configuration by using the **`querysn`** command to create an exclude file.

For more information about installing and using SDD, see *IBM System Storage Multipath Subsystem Device Driver User's Guide*, GC52-1309. This publication and other information are available at this website:

http://www.ibm.com/servers/storage/support/

SDD load balancing

SDD automatically adjusts data routing for optimum performance. Multipath load balancing of data flow prevents a single path from becoming overloaded. If a single path is overloaded, it can cause the I/O congestion that occurs when many I/O operations are directed to common devices along the same I/O path.

The policy that is specified for the device determines the path that is selected to use for an I/O operation. The following policies are available:

- Load balancing (default). The path to use for an I/O operation is chosen by estimating the load on the adapter to which each path is attached. The load is a function of the number of I/O operations currently in process. If multiple paths have the same load, a path is chosen at random from those paths.
- Round-robin. The path to use for each I/O operation is chosen at random from those paths not used for the last I/O operation. If a device has only two paths, SDD alternates between the two paths.
- Failover only. All I/O operations for the device are sent to the same (preferred) path until the path fails because of I/O errors. Then, an alternate path is chosen for later I/O operations.

Normally, path selection is performed on a global rotating basis; however, the same path is used when two sequential write operations are detected.

Single-path mode

SDD does not support concurrent download and installation of the Licensed Machine Code (LMC) to the DS8000 if hosts use a single-path mode. However, SDD supports a single-path Fibre Channel connection from your host system to a DS8000. It is possible to create a volume group or a vpath device with only a single path.

> **Important:** With a single-path connection, which we do not advise, the SDD *cannot* provide failure protection and load balancing.

Single FC adapter with multiple paths

A host system with a single Fibre Channel (FC) adapter that connects through a switch to multiple DS8000 ports is considered to have multiple Fibre Channel paths.

From an availability point of view, we discourage this configuration because of the single fiber cable from the host to the SAN switch. However, this configuration is better than a single path from the host to the DS8000, and this configuration can be useful for preparing for maintenance on the DS8000.

Path failover and online recovery

SDD automatically and non-disruptively can redirect data to an alternate data path.

When a path failure occurs, the IBM SDD automatically reroutes the I/O operations from the failed path to the other remaining paths. This action eliminates the possibility of a data path becoming a single point of failure.

SDD datapath command

The SDD provides commands that you can use to display the status of adapters that are used to manage disk devices or to display the status of the disk devices. You can also set individual paths online or offline, and you can also set all paths connected to an adapter online or offline at one time.

Multipath I/O

Multipath I/O (MPIO) summarizes native multipathing technologies that are available in several operating systems, such as AIX, Linux, and Windows. Although the implementation differs for each of the operating systems, the basic concept is almost the same:

- The multipathing module is delivered with the operating system.
- The multipathing module supports failover and load balancing for standard SCSI devices, such as simple SCSI disks or SCSI arrays.
- To add device-specific support and functions for a specific storage device, each storage vendor might provide a device-specific module that implements advanced functions for managing the specific storage device.

IBM currently provides a device-specific module for the DS8000 for AIX, Linux, and Windows according to the information in Table 8-1.

Table 8-1 Available DS8000 specific MPIO path control modules

Operating system	Multipathing solution	Device-specific module	Acronym
AIX	MPIO	SDD Path Control Module	SDDPCM
Windows	MPIO	SDD Device Specific Module	Subsystem Device Driver Device Specific Module (SDDDSM)
Linux	Device-Mapper Multipath	DM-MPIO configuration file	DM-MPIO

External multipathing software

In addition to the SDD and MPIO solutions, third-party multipathing software is available for specific host operating systems and configurations.

For example, Symantec provides an alternative to the IBM provided multipathing software. The Veritas Volume Manager (VxVM) relies on the Microsoft implementation of MPIO and Device Specific Modules (DSMs) that rely on the Storport driver. The Storport driver is not available for all versions of Windows.

The Veritas Dynamic MultiPathing (DMP) software is also available for UNIX versions, such as Oracle Solaris.

Check the System Storage Interoperation Center (SSIC) website for your specific hardware configuration:

http://www.ibm.com/systems/support/storage/config/ssic/

8.3 Attaching IBM System z hosts

This section describes the host system requirements and attachment considerations for attaching the IBM System z hosts (z/OS, z/VM®, z/VSE®, Linux on System z, and Transaction Processing Facility (TPF)) to the DS8000 series. Starting with the DS8800, the ESCON attachment is discontinued. In the following sections, we discuss attachment through FICON adapters only.

> **FCP:** z/VM, z/VSE, and Linux for System z can also be attached to the DS8000 series with FCP.

8.3.1 FICON

FICON is a Fibre Connection used with System z servers. Each storage unit host adapter has either four or eight ports, and each port has a unique WWPN. You can configure the port to operate with the FICON upper layer protocol. When configured for FICON, the storage unit provides the following configurations:

- Either fabric or point-to-point topology
- A maximum of 128 host ports for DS8800 4-way model and a maximum of 32 host ports for DS8800 2-way model (either Standard or Business Class)
- A maximum of 2048 logical paths on each Fibre Channel port
- Access to all 255 control unit images (65280 count key data (CKD) devices) over each FICON port. The connection speeds are 200, 400, and 800 MB/s, which is similar to Fibre Channel for Open Systems.

IBM introduced FICON channels in the IBM 9672 G5 and G6 servers with the capability to run at 1 Gbps. Since that time, IBM introduced several generations of FICON channels. The FICON Express8S channels make up the latest generation of FICON channels. They are designed to support 8 Gbps link speeds and can also auto-negotiate to 2 or 4 Gbps link speeds. These speeds depend on the capability of the director or control unit port at the other end of the link. Operating at 8 Gbps speeds, FICON Express8S channels are designed to achieve up to 620 MBps for a mix of large sequential read and write I/O operations, as depicted in the following charts. Figure 8-4 on page 321 shows a comparison of the overall throughput capabilities of various generations of channel technology.

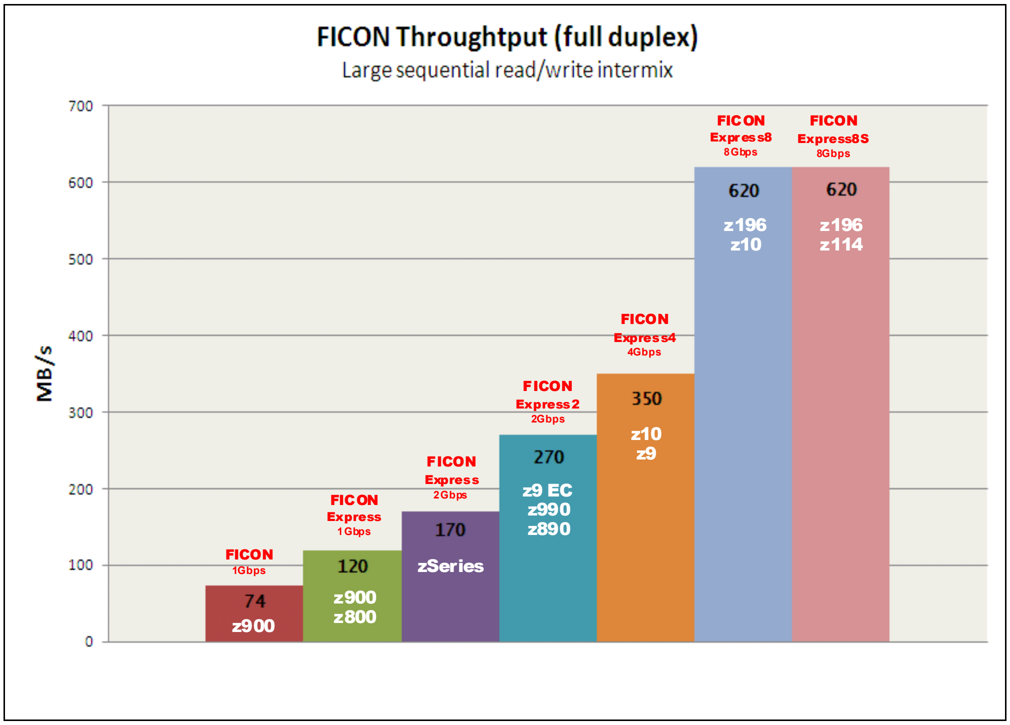


Figure 8-4 Measurements of channel throughput over several generations of channels

The FICON Express8S channel on the IBM System zEnterprise 196 and z114 represents a significant improvement in maximum bandwidth capability compared to FICON Express4 channels and previous FICON offerings. The response time improvements are expected to be noticeable for large data transfers. The speed at which data moves across a 4 Gbps link is effectively 800 MBps compared to 400 MBps with a 4 Gbps link.

As shown in Figure 8-5 on page 322, the maximum number of I/Os per second (IOPS) measured on a FICON Express8S channel that runs an I/O driver benchmark with a 4 KB per I/O workload is approximately 23000 IOPS. This maximum is more than 10% greater than the maximum number of I/Os measured with a FICON Express8 channel. The greater performance capabilities of the FICON Express8S channel make it a good match with the performance characteristics of the new DS8000 host adapters.

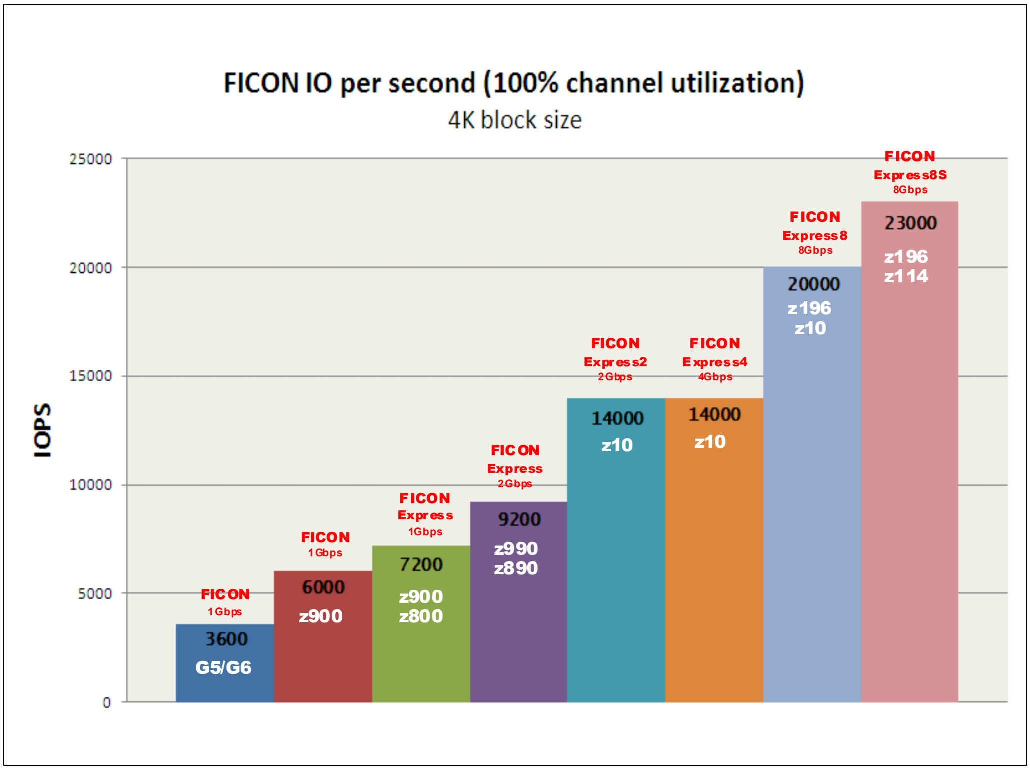


Figure 8-5 Measurements of IOPS over several generations of channels

Support: The FICON Express8S SX and LX are supported on zEnterprise 196 and z114 servers only. FICON Express8 SX and LX are available on z10, zEnterprise 196, and z114 servers.

The System zEnterprise 196 and System z114 servers offer FICON Express8S SX and LX features that have two independent channels. Each feature occupies a single I/O slot and uses one CHPID per channel. Each channel supports 2 Gbps, 4 Gbps, and 8 Gbps link data rates with auto-negotiation to support existing switches, directors, and storage devices.

FICON Express4: FICON Express4 was the last feature to support 1 Gbps link data rates.

For any generation of FICON channels, you can attach directly to a DS8000 or you can attach through a FICON capable Fibre Channel switch.

When you use a Fibre Channel/FICON host adapter to attach to FICON channels, either directly or through a switch, the port is dedicated to FICON attachment and cannot be simultaneously attached to FCP hosts. When you attach a DS8000 to FICON channels through one or more switches, the maximum number of FICON *logical paths* is 2048 per DS8000 host adapter port. The directors provide high availability with redundant components and no single points of failure.

8.3.2 FICON configuration and sizing considerations

Usually in System z environments, a one-to-one connection between FICON channels and storage host adapters is preferred, because the FICON channels are usually shared among multiple logical partitions (LPARs) and heavily utilized. Carefully plan the oversubscription of host adapter ports to avoid any bottlenecks. Figure 8-6 shows an example of FICON attachment that connects a System z server through FICON switches. This example uses 16 FICON channel paths to eight host adapter ports on the DS8000, and addresses eight logical control units (LCUs). This channel consolidation might be possible when your aggregate host workload does not exceed the performance capabilities of the DS8000 host adapter.

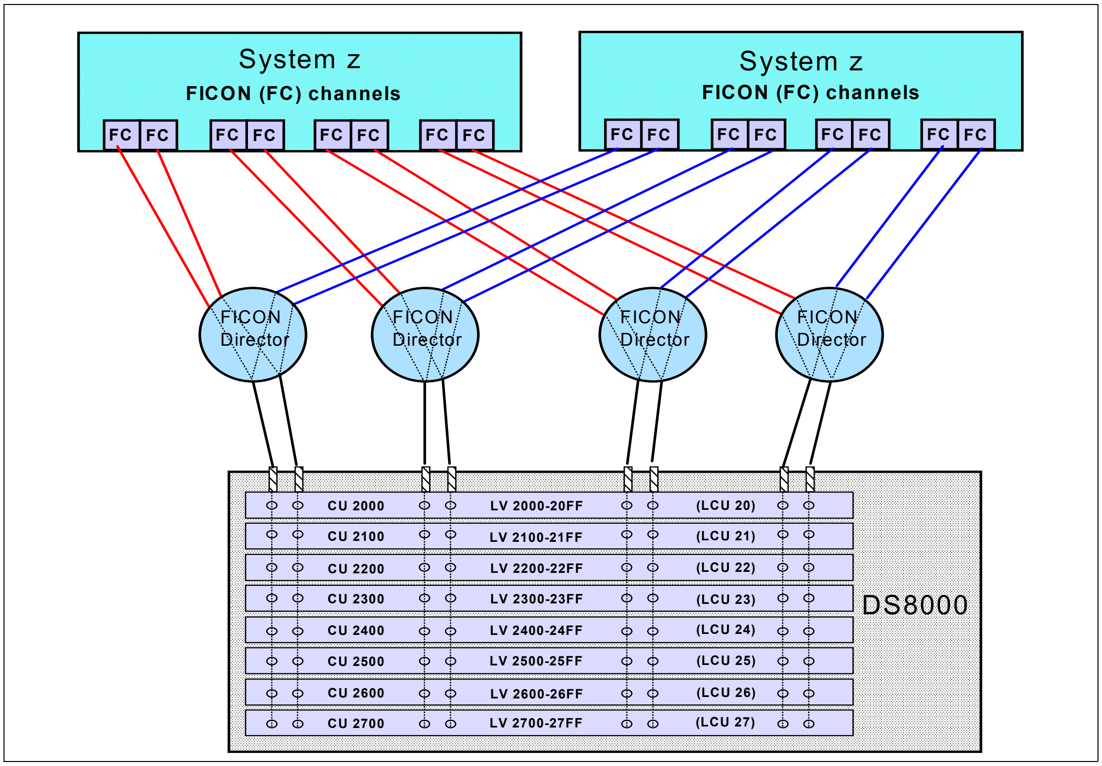


Figure 8-6 Many-to-one FICON attachment

A one-to-many configuration is also possible, as shown in Figure 8-6, but, again, careful planning is needed to avoid performance issues.

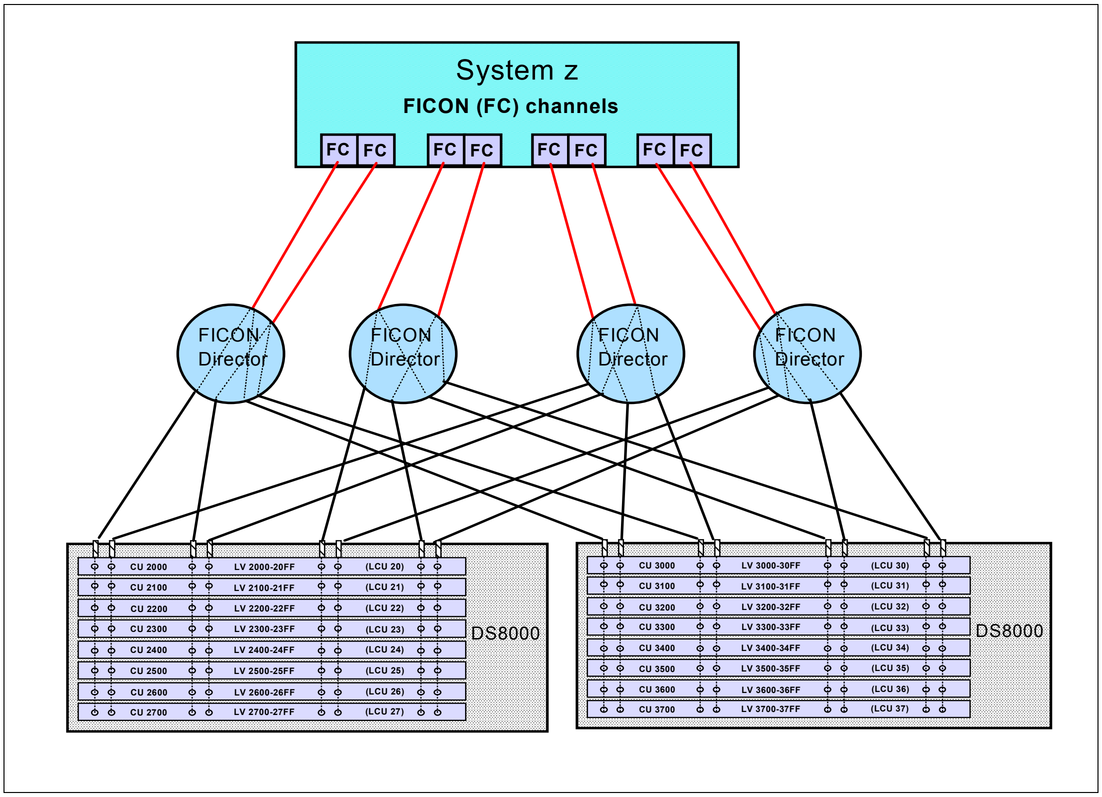


Figure 8-7 One-to-many FICON attachment

Sizing FICON connectivity is not an easy task. You must consider many factors. We strongly advise that you create a detailed analysis of the specific environment. Use these guidelines before you begin sizing the attachment environment:

- For FICON Express CHPID utilization, the recommended maximum utilization level is 50%.
- For the FICON Bus busy utilization, the recommended maximum utilization level is 40%.
- For the FICON Express Link utilization with an estimated link throughput of 2 Gbps, 4 Gbps, or 8 Gbps, the recommended maximum utilization threshold level is 70%

For more information about the DS8000 FICON support, see *IBM System Storage DS8000 Host Systems Attachment Guide*, SC26-7917, and *FICON Native Implementation and Reference Guide,* SG24-6266.

8.3.3 z/VM, z/VSE, and Linux on System z attachment

FICON channels in Fibre Channel Protocol (FCP) mode provide full fabric and point-to-point attachments of Fixed Block (FB) devices to the operating system images. With this attachment, z/VM, z/VSE, and Linux on System z can access industry-standard FCP storage controllers and devices. This capability can facilitate the consolidation of UNIX server farms onto System z servers and help protect investments in Small Computer System Interface (SCSI)-based storage.

The FICON features provide support of Fibre Channel and SCSI devices in z/VM, z/VSE, and Linux on System z. FCP allows z/VM, z/VSE, and Linux on System z to access industry-standard SCSI devices. For disk applications, these FCP storage devices use FB 512-byte sectors rather than Extended Count Key Data (ECKD™) format. Each FICON

Express8S, FICON Express8, FICON Express4, FICON Express2, and FICON Express channel can be defined in FCP mode.

Linux FCP connectivity

You can use either direct or switched attachment to attach a storage unit to a System z host system that runs SUSE SLES 9, 10, or 11, or Red Hat Enterprise Linux 4.8, and later with current maintenance updates for FICON.

For more information about Linux on System z connectivity, see the IBM developerWorks® website:

http://www.ibm.com/developerworks/linux/linux390/development_documentation.html

9

Performance considerations for UNIX servers

This chapter describes performance considerations for attaching the DS8000 to several of the supported UNIX operating systems.

This chapter includes these topics:

- Planning and preparing UNIX servers for performance
- UNIX disk I/O architecture
- AIX disk I/O components
- AIX performance monitoring tools
- Solaris disk I/O components
- Solaris performance monitoring tools
- Hewlett-Packard UNIX (HP-UX) disk I/O components
- HP-UX performance monitoring tools
- Testing and verifying the DS8000 storage

9.1 Planning and preparing UNIX servers for performance

Planning and configuring a UNIX system for performance is never a simple task. There are numerous factors to consider before tuning parameters and deciding the ideal setting. To help you answer these questions, consider the following factors:

- Type of application: There are thousands of applications available on the market. But, it is possible to group them into a few types, based on their I/O profile. The I/O profile helps you decide the best configuration at the operating system level. For more details about identifying and classifying applications, see “Logical configuration performance considerations” on page 87.
- Platform and version of operating system: Although, in general terms, they have the same performance characteristics, there can be differences in how each operating system implements these functions. In addition, newer versions or releases of an operating system bring or already have performance parameters with optimized default values for certain workload types.
- Type of environment: Another significant factor is whether the environment is for production, testing, or development. Normally, production environments are hosted in servers with many processors, hundreds of gigabytes of memory, and terabytes of disk space. They demand high levels of availability and therefore are difficult to schedule for downtime, hence, the need to make a more detailed plan. Quality and assurance (or testing) environments normally are smaller in size, but they need to sustain the performance tests. Development environments are typically much smaller than their respective production environments, and normally performance is not a concern. In this chapter, we pay attention to the performance tuning of the production environments.

Before planning for performance, validate the configuration of your environment. See the System Storage Interoperation Center (SSIC):

http://www.ibm.com/systems/support/storage/config/ssic/index.jsp

Also, for host bus adapter (HBA) interoperability, see this website:

http://www.ibm.com/systems/support/storage/config/hba/index.wss

Check the IBM Support website to download the latest version of firmware for the Fibre Channel (FC) adapters for AIX servers:

http://www14.software.ibm.com/webapp/set2/firmware/gjsn

Download the latest fix packs for your AIX version from the following site:

http://www.ibm.com/eserver/support/fixes/fixcentral/main/pseries/aix

The Oracle Solaris software family requires patches to ensure that the host and the DS8000 function correctly. See the following website for the most current list of Oracle Solaris-SPARC patches and the Oracle Solaris-x86 patch for recent and current versions of Oracle Solaris:

http://www.oracle.com/us/products/servers-storage/solaris/solaris11/overview/index.html

For HP-UX servers, download device drivers from this website:

http://www.hp.com/country/us/eng/support.html

Also, for detailed information about how to attach and configure a host system to a DS8000, see the *IBM System Storage DS8000 Host System Attachment Guide,* GC27-2298:

http://www.ibm.com/support/docview.wss?rs=1114&context=HW2B2&uid=ssg1S7001161

9.1.1 UNIX disk I/O architecture

It is fundamental to understand the UNIX I/O subsystem to adequately tune your system. The I/O subsystem can be represented by a set of layers. Figure 9-1 provides an overview of those layers.

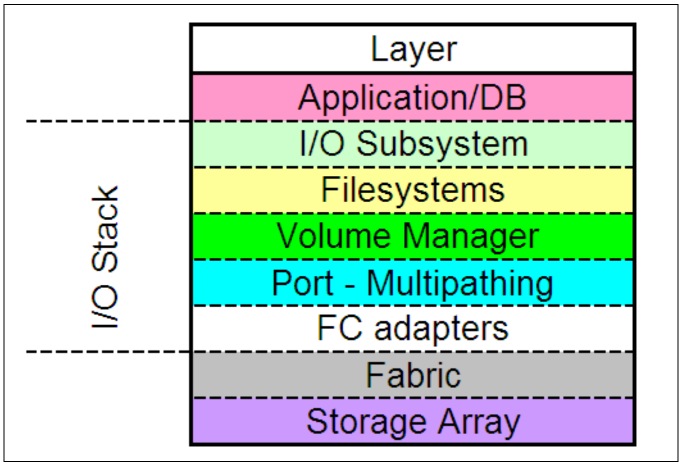


Figure 9-1 UNIX disk I/O architecture

I/O requests normally go through these layers:

- Application/DB layer: This layer is the top-level layer where many of the I/O requests start. Each application generates several I/Os that follow a pattern or a profile. An application I/O profile has these characteristics:
  - IOPS: IOPS is the number of I/Os (reads and writes) per second.
  - Throughput: How much data is transferred in a certain sample time? Typically, the throughput is measured in MB/s or KB/s.
  - I/O size: This I/O size is the result of MB/s divided by IOPS.
  - Read ratio: The read ratio is the percentage of I/O reads compared to the total I/Os.
  - Disk space: This amount is the total amount of disk space needed by the application.
- I/O system calls layer: Through the system calls provided by the operating system, the application issues I/O requests to the storage. By default, all I/O operations are synchronous. Many operating systems also provide asynchronous I/O, which is a facility that allows an application to overlap processing time while it issues I/Os to the storage. Typically, the databases take the advantage of this feature.
- Filesystem layer: The filesystem is the operating system's way to organize and manage the data in the form of files. Many filesystems support buffered and unbuffered I/Os. If your application has its own caching mechanism and supports a type of direct I/O, we suggest that you enable it, because it avoids double-buffering and reduces the processor utilization. Otherwise, your application can take advantage of features, such as file caching, read-ahead, and write-behind.
- Volume manager layer: A *volume manager* was a key component to distribute the I/O workload over the logical unit numbers (LUNs) of the DS8000. The situation differs with the implementation of Easy Tier V3 and I/O Priority Manager in DS8800. You need to use a logical volume manager (LVM) to provide LUN concatenation only if it is the cluster software requirement. For any other case, it is better to use DS8800 or DS8700 capabilities of managing the workload, because additional striping at the LVM level might deform the skew factor and lead to an incorrect heat calculation. Also, because I/O Priority

Manager works at the storage system back-end level, it improves the usage of the internal resources. Managing priority at the operating system level might be less effective.

- Multipathing/Disk layer: Today, there are several multipathing solutions available: hardware multipathing, software multipathing, and operating system multipathing. It is better to adopt the operating system multipathing solutions. However, depending on the environment, you might face limitations and prefer to use a hardware or software multipathing solution. Always try *not* to exceed a maximum of four paths for each LUN unless required.
- FC adapter layer: The need to make configuration changes in the FC adapters depends on the operating system and vendor model. Always consult the *IBM System Storage DS8000 Host Systems Attachment Guide,* GC27-2298, for specific instructions about how to set up the FC adapters. Also, check the compatibility matrix for dependencies among the firmware level, operating system patch levels, and adapter models.
- Fabric layer: The storage area network (SAN) is used to interconnect storage devices and servers.
- Array layer: The array layer is the DS8000 in our case.

Normally in each of these layers, there are performance indicators that help you to assess how that particular layer affects performance.

Typical performance indicators

The following indicators are the typical performance indicators:

- The first performance indicator used to assess whether there is an I/O bottleneck is the wait I/O time (wio). It is essential to realize that the wio is calculated differently depending on the operating system:
  - Current versions of IBM AIX increment wio if the processor is not busy in user or system mode and there is outstanding I/O started by that processor. In this way, systems with several processors have a less inflated wio. Moreover, wio from filesystems mounted via Network Filesystem (NFS) is also recorded.
  - In Solaris systems, wio is calculated with the idle time. In addition, the counter wio in Solaris is incremented by disk I/O. It also is incremented by I/Os from filesystems and Small Computer System Interface (SCSI) tape devices.

  For more details about the wait I/O time in AIX, see the following link:

  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.prftungd%2Fdoc%2Fprftungd%2Fwait_io_time_reporting.htm

  For more details about the wait I/O time in Solaris, see the following link:

  http://sunsite.uakom.sk/sunworldonline/swol-08-1997/swol-08-insidesolaris.html

  The wio might be an indication that there is a disk I/O bottleneck, but it is not enough to assume from only the wio that there is a constraint of disk I/O. We must observe other counters, such as the blocked process in the kernel threads column and statistics generated by iostat or an equivalent tool.
- The technology for disk is improved. In the past, disks only were capable of 120 I/Os per second (IOPS) and had no cache memory. Therefore, utilization levels of 10 - 30% were considered high. Today, with arrays of the DS8000 class (supporting tens or hundreds of gigabytes of cache memory and hundreds of physical disks at the back end), even utilization levels higher than 80 or 90% still might not indicate an I/O performance problem.
- It is fundamental to check the queue length, the service time, and the I/O size averages that are reported in the disk statistics:
  - If the queue length and the service time are low, there is no performance problem.

   - If the queue length is low and the service time and the I/O size are high, it is also not evidence of a performance problem.
- Performance thresholds: They might indicate a change in the system. However, they are not able to explain why or how it changed. Only a good interpretation of data is able to answer these types of questions.

  We describe an actual case: A user was complaining that the transactional system showed a performance degradation of 12% in the average transaction response time. The Database Administrator (DBA) argued that the database spent a good part of the time in disk I/O operations. At the operating system and the storage level, all performance indicators were excellent. The only curious fact was that the system showed a hit cache of 80%, which did not make much sense, because the access characteristic of a transactional system is random. High levels of hit cache indicated that somehow the system was reading sequentially. By analyzing the application, we discovered that about 30% of the database workload related to 11 queries. These 11 queries accessed the tables without the help of an index. The fix was the creation of those indexes with specific fields to optimize the access to disk.

9.1.2 Modern approach to the Logical Volume Management in UNIX Systems

Logical Volume Managers (LVMs) were and are the main part of UNIX system management and performance tuning. Having a rich functionality, they allowed administrators to precisely adopt servers and disk systems to the particularities of the applications and the workloads. The important role of the LVMs explains the way that the DS8000 systems were configured before. It was the method of workload isolation at the rank and array level and the hardware resource management approach. It was the administrator's decision about how to use each of the hardware components in the DS8000 system and when. LVMs played a key role in the disk management process; they allow administrators to add a flexible level of management of the hardware resources. There were two main techniques of disk management in LVMs: volume striping with different levels of granularity and implementation of the RAID technology. Administrators overcome the imperfection of the disk systems with these techniques.

Modern DS8000 systems have many improvements in the data management that can change LVM usage. Easy Tier V3 functionality moves the method of data isolation from the rank level to the extent pool, volume, and application levels. It is important that a disk system of today must be planned from the application point of view, not the hardware resource allocation point of view. Plan the logical volumes on the extent pool that is for one type of application or workload. This method of disk system layout eliminates the necessity of LVM usage.

Moreover, by using LVM striping of the Easy Tier managed volumes, you eliminate most of the technology benefits, because striping shadows the real skew factor and changes the real picture of the hot extents allocation. This method might lead to improper extent migration plan generation, which leads to continuous massive extent migration. Performance analysis becomes complicated and makes I/O bottleneck allocation a complicated task. In general, the basic approach for most of the applications is to use one or two hybrid extent pools with three tiers and Easy Tier in automated mode for group of applications of the same kind or the same workload type. To prevent bandwidth consumption by one or several applications, the I/O Priority Manager functionality must be used.

The extended RAID functionality of the LVM (RAID 5, 6, and 10) must not be used at all, except the RAID 1 (Mirroring) function which might be required in high availability (HA) and disaster recovery (DR) solutions.

However, there are still several major points of LVM usage:

- Definite and complete understanding of the workload nature, which requires isolation at the rank level and disk management at the LVM level. This situation is a rare case for today, but it might be required.
- Usage of the large volumes in the operating system that are not recommended to create in the disk system (typically volumes larger than 2 TB). The concatenation functionality of the LVM must be used in this case. Concatenation itself does not change the skew factor.
- Any cluster software requirement of LVM usage that might be the key component of the HA or DR solution.

Consider these points when you read the LVM description in the following sections. Also, see Chapter 4, "Logical configuration performance considerations" on page 87.

9.1.3 Queue depth parameter (qdepth)

The operating system can enforce limits on the number of I/O requests that can be outstanding from the SCSI adapter to a specific SCSI bus or disk drive. These limits are intended to exploit the hardware ability to handle multiple requests while ensuring that the seek-optimization algorithms in the device drivers are able to operate effectively. There are also queue depth limits to the FC adapter and FC path. These parameters have a common meaning: the length of the queue of the SCSI commands, which a device can keep unconfirmed while maintaining the I/O requests. A device (disk or FC adapter) sends a successful command completion notification to the operating system or a driver before it is completed, which allows the operating system or a driver to send another command or I/O request. Remember that one I/O request can consist of two or more SCSI commands.

There are two methods for implementation of queuing: tagged and untagged. *Tagged queuing* allows a target to accept multiple I/O processes from each initiator for each logical unit. *Untagged queuing* allows a target to accept one I/O process from each initiator for each logical unit or target routine. Untagged queuing might be supported by SCSI-1 or SCSI-2 devices. Tagged queuing is new in SCSI-2. For more information, see this link:

`http://ldkelley.com/SCSI2/SCSI2-07.html`

The qdepth parameter might affect the disk I/O performance. Values that are too small can make the device ineffective to use. Values that are too high might lead to the `QUEUE FULL` status of the device, reject the next I/O, and lead to data corruption or a system crash.

Another important reason why queue depth parameters must be set correctly is the queue limits of the host ports of the disk system. The host port might be flooded with the SCSI commands if there is no correct limit set in the operating system. When this situation happens, a host port refuses to accept any I/Os, then resets, and then starts the loop initialization primitive (LIP) procedure. This situation leads to the inactivity of the port for up to several minutes and might initiate path failover or an I/O interruption. Moreover, in highly loaded environments, this situation leads to the overload of the other paths and might lead to the complete I/O interruption for the application or buffer overflow in the operating system, which causes paging activity.

Queue depth parameter settings

Each operating system has its own settings for the disk I/O, because many factors affect the formation of the disk I/O in the operating system: kernel I/O threads, device driver-specific factors, host bus adapter (HBA) driver specific factors, and filesystem specifics. Pay close attention to the parameters of the queue depth for the disks and FC HBAs specified in the *IBM System Storage DS8000 Host System Attachment Guide,* SC26-7917-02, for every supported operating system. Not observing these suggestions can lead to the DS8000 host port buffer overflow and the data corruption with interrupts in the data flow. For example, the Oracle Solaris host attachment section instructs you to set lun-queue-depth to 30 and tgt-queue-depth to 0 when you use the Emulex LP9002DC, LP9002L, LP9002S, LP9802, LP10000, LP10000, LP11000, LP11002, LPe11000, LPe11002, LPe12000, and LPe12002 DC adapters. These settings are needed to limit the number of commands that the driver sends to the FC target and FCP LUN to prevent buffer overflow.

In addition to the settings specified in the *DS8000 Host System Attachment Guide,* SC26-7917-02, there can be additional suggestions for multipathing software. For example, the DM-Multipath driver in Linux requires that you have the following settings for QLogic FC HBAs:

- ql2xmaxqdepth. This parameter defines the maximum queue depth reported to SCSI Mid-Level per device. The Queue depth specifies the number of outstanding requests per LUN. The default is 32. The recommended number is 16.
- qfull_retry_count. The number of retries to perform on Queue Full status on the device. The default is 16. The recommended number is 32.

The DM-Multipath driver has following queue depth-related settings:

- queue_length path selection algorithm, which selects the next path for the I/O based on the queue length of the each path. So, by changing the maximum queue length of the path or the device, you might set the priorities of use of the path.
- queue_if_no_path, which is the same as setting no_path_retry to queue. A numeric value for this attribute specifies the number of times that the system attempts to use a failed path before disabling queueing. A value of fail indicates immediate failure, without queuing. A value of queue indicates that queuing must not stop until the path is fixed. If feature “1 queue_if_no_path" is specified, any process that issues I/O stops until one or more paths are restored.
- queue_without_daemon. If set to no, the multipathd daemon disables queuing for all devices when it is shut down.
- flush_on_last_del. If set to yes, the multipathd daemon disables queuing when the last path to a device is deleted.

IBM Subsystem Device Driver (SDD) and SDD Path Control Module (SDDPCM) multipathing drivers have their own queue depth settings to manage FC targets and the vpaths.

If qdepth_enable=yes (the default), I/Os that exceed the queue_depth queue at SDD. And if qdepth_enable=no, I/Os that exceed the queue_depth queue in the hdisk wait queue. SDD with qdepth_enable=no and SDDPCM do not queue I/Os and instead merely pass them to the hdisk drivers.

The datapath command: With SDD 1.6, it is preferable to use the **`datapath`** command to change qdepth_enable, rather than by using **`chdev`**, because then it is a dynamic change. For example, **`datapath set qdepth disable`** sets it to no. Certain releases of SDD do not include SDD queueing, and other releases include SDD queueing. Certain releases do not show the qdepth_enable attribute. Either check the manual for your version of SDD or try the **`datapath`** command to see whether it supports turning this feature off.

If you used both SDD and SDDPCM, you remember that with SDD, each LUN has a corresponding vpath and an hdisk for each path to the vpath or LUN. And with SDDPCM, you have only one hdisk per LUN. Thus, with SDD, you can submit `queue_depth x # paths` to a LUN, while with SDDPCM, you can submit `queue_depth IOs` only to the LUN. If you switch from SDD that uses four paths to SDDPCM, you must set the SDDPCM hdisks to 4x that of the SDD hdisks for an equivalent effective queue depth. And, we advise that you migrate to SDDPCM, because SDDPCM over time might replace SDD.

Both the hdisk and adapter drivers have “in process” and “wait” queues. After the queue limit is reached, the I/Os wait until an I/O completes and opens a slot in the service queue. The in-process queue is also sometimes called the “service” queue. Many applications do not generate many in-flight I/Os, especially single-threaded applications that do not use asynchronous I/O. Applications that use asynchronous I/O are likely to generate more in-flight I/Os.

Queue depths with VIO server

When using a VIO server, configure VSCSI adapters (for each virtual adapter in a Virtual I/O Server (VIOS), which is known as a *vhost device*, there is a matching VSCSI adapter in a VIOC). These adapters have a fixed queue depth that varies depending on how many VSCSI LUNs are configured for the adapter. There are 512 command elements of which two are used by the adapter, three are reserved for each VSCSI LUN for error recovery, and the rest are used for I/O requests. Thus, with the default queue_depth of 3 for VSCSI LUNs, there are up to 85 LUNs to use at an adapter: (512 - 2)/(3 + 3) = 85 (rounding down). So, if we need higher queue depths for the devices, the number of LUNs per adapter is reduced. For example, if we want to use a queue_depth of 25, we can have 510/28 = 18 LUNs. We can configure multiple VSCSI adapters to handle many LUNs with high queue depths and each one requires additional memory. You can have more than one VSCSI adapter on a VIOC that is connected to the same VIOS if you need more bandwidth. See Figure 9-2 for reference.

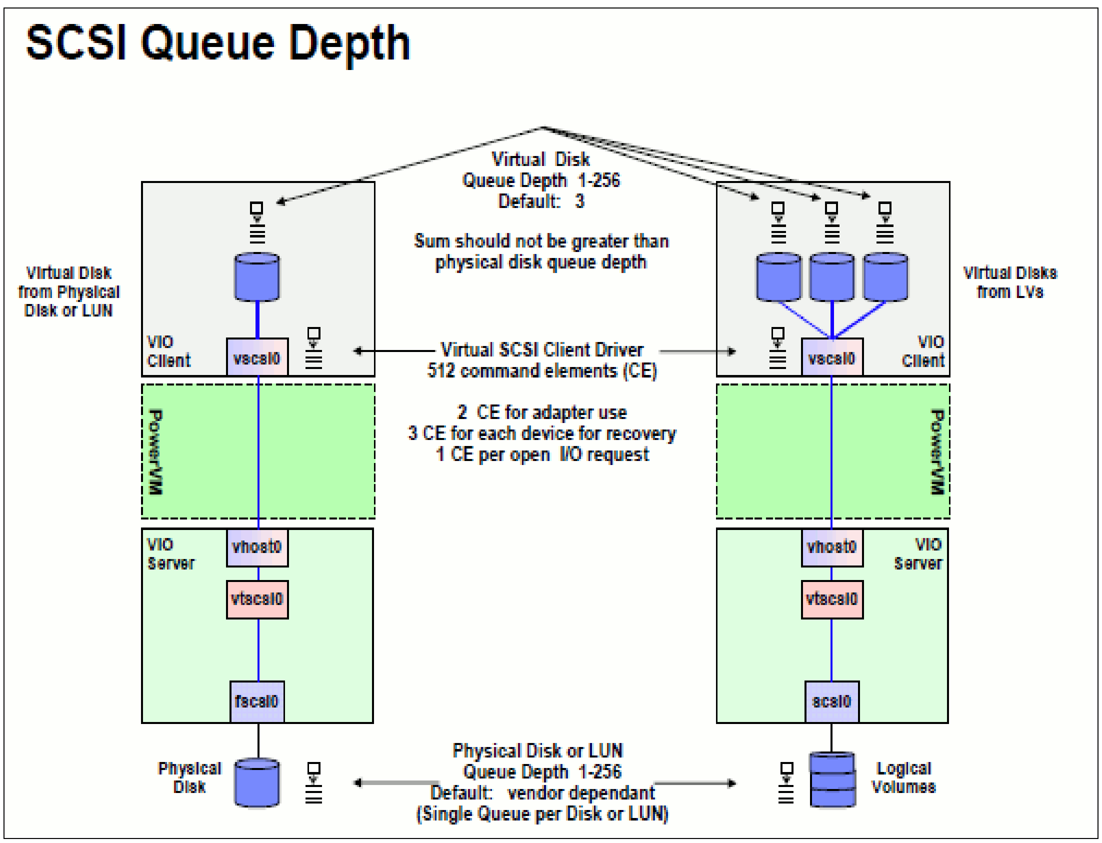


Figure 9-2 VIO server queue depth

As it is shown in Figure 9-2 on page 334, you need to set the queue_depth attribute on the VIOC hdisk to match that of the mapped hdisk queue_depth on the VIOS. For a formula, the maximum number of LUNs per virtual SCSI adapter (vhost on the VIOS or vscsi on the VIOC) is =INT(510/(*Q*+3)) where *Q* is the queue_depth of all the LUNs (which assumes that they are all the same).

> **Important:** To change the queue_depth on an hdisk at the VIOS, you are required to unmap the disk from the VIOC and remap it back.

If you use N_Port ID Virtualization (NPIV), if you increase num_cmd_elems on the virtual FC (vFC) adapter, you must also increase the setting on the real FC adapter.

For more information about the queue depth settings for VIO Server, see *IBM System Storage DS8000 Host Attachment and Interoperability,* SG24-8887-00:

http://www.redbooks.ibm.com/abstracts/sg248887.html

Queue depth and the performance

There is no need to merely increase these values. It is possible to overload the disk subsystem or to cause problems with device configuration at boot. So, the approach of adding the hdisk queue_depths to get a total to determine the num_cmd_elems is not wise. Instead, it is better to use the maximum I/Os to each device for tuning. When you increase the queue_depths and number of in-flight I/Os that are sent to the disk subsystem, the I/O service times are likely to increase, but throughput increases. If the I/O service times start approaching the disk timeout value, you are submitting more I/Os than the disk subsystem can handle. If you start seeing I/O timeouts and errors in the error log that indicate that there are problems when completing I/Os, look for hardware problems or make the pipe smaller.

A good general rule for tuning queue_depths is to increase queue_depths until I/O service times start exceeding 15 ms for small random reads or writes or you are not filling the queues. After I/O service times start increasing, you push the bottleneck from the AIX disk and adapter queues to the disk subsystem. There are two approaches to tuning queue depth:

- Use your application and tune the queues from it.
- Use a test tool to see what the disk subsystem can handle and tune the queues from that information based on what the disk subsystem can handle.

We prefer to tune based on the application I/O requirements, especially when the disk system is shared with other servers.

When you examine the devstats, if you see that the Maximum field = queue_depth x # paths and qdepth_enable=yes for SDD, then increasing the queue_depth for the hdisks might help performance. At least the I/Os queue on the disk subsystem rather than in AIX. It is reasonable to increase queue depths about 50% at a time.

Regarding the qdepth_enable parameter, the default is yes, which essentially has SDD handling the I/Os beyond queue_depth for the underlying hdisks. Setting it to `no` results in the hdisk device driver handling them in its wait queue. With qdepth_enable=yes, SDD handles the wait queue; otherwise, the hdisk device driver handles the wait queue. There are error handling benefits that allow SDD to handle these I/Os, for example, by using LVM mirroring across two DS8000s. With heavy I/O loads and much queuing in SDD (when qdepth_enable=yes), it is more efficient to allow the hdisk device drivers to handle relatively shorter wait queues rather than SDD handling a long wait queue by setting qdepth_enable=no. SDD queue handling is single threaded and there is a thread for handling each hdisk queue. So, if error handling is of primary importance (for example, when LVM mirroring across disk subsystems), leave qdepth_enable=yes. Otherwise, setting qdepth_enable=no more efficiently handles the wait queues when they are long. Set the qdepth_enable parameter by using the **`datapath`** command, because it is a dynamic change that way (**`chdev`** is not dynamic for this parameter).

For the adapters, look at the adaptstats column. And set num_cmd_elems=Maximum or 200, whichever is greater. Unlike devstats with qdepth_enable=yes, Maximum for adaptstats can exceed num_cmd_elems.

It is also reasonable to use the **iostat -D** command or **sar -d** to provide an indication if the queue_depths need to be increased (Example 9-1).

Example 9-1 Iostat -D output (output truncated)

```
Paths/Disks:
hdisk0              xfer:  %tm_act      bps      tps      bread      bwrtn
                              1.4     30.4K      3.6       23.7K       6.7K
                    read:       rps  avgserv  minserv  maxserv   timeouts      fails
                              2.8       5.7      1.6     25.1           0          0
                   write:       wps  avgserv  minserv  maxserv   timeouts      fails
                              0.8       9.0      2.1     52.8           0          0
                   queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                             11.5       0.0     34.4      0.0        0.0        0.9
```

The avgwqsz is the average wait queue size, and avgsqsz is the average service queue size. The average time spent in the wait queue is avgtime. The sqfull value changed from initially being a count of the times that we submitted an I/O to a full queue, to now where it is the rate of I/Os submitted to a full queue. The example report shows the prior case (a count of I/Os submitted to a full queue). Newer releases typically show decimal fractions that indicate a rate. It is good that **iostat -D** separates reads and writes, because we expect the I/O service times to differ when we have a disk subsystem with cache. The most useful report for tuning is to run **iostat -D**. This command shows statistics since system boot, and it assumes that the system is configured to continuously maintain disk IO history (run # **lsattr -El sys0**, or **smitty chgsys** to see whether the iostat attribute is set to true).

The **sar -d** generates an output as shown in Figure 9-3.

```
# sar -d 1 2
System configuration: lcpu=2 drives=1 ent=0.30


10:01:37 device     %busy  avque  r+w/s  Kbs/s  avwait  avserv
10:01:38 hdisk0     100    36.1   363    46153  51.1    8.3
10:01:39 hdisk0     99     38.1   350    44105  58.0    8.5
Average  hdisk0     99     37.1   356    45129  54.6    8.4
```

Figure 9-3 Output of sar -d command

The avwait is the average time spent in the wait queue. The avserv is the average time spent in the service queue. And, avserv corresponds to avgserv in the **iostat** output. The avque value represents the average number of I/Os in the wait queue.

SDD provides the **datapath query devstats** and **datapath query adaptstats** commands to show hdisk and adapter queue statistics. SDDPCM similarly has **pcmpath query devstats** and **pcmpath query adaptstats**. You can refer to the SDD manual for syntax, options, and explanations of all the fields. We show **devstats** output for a single LUN. See Example 9-2 on page 336 for details.

Example 9-2 Pcmpath query output (output truncated)

```
# pcmpath query devstats
```

```
DEV#:   1  DEVICE NAME: hdisk1
===============================
               Total Read  Total Write  Active Read  Active Write   Maximum
I/O:                69227        49414            0             0        40
SECTOR:           1294593      1300373            0             0      1737

Transfer Size:      <= 512        <= 4k        <= 16K        <= 64K     > 64K
                       118        20702         80403         12173      5245

# pcmpath query adaptstats
Adapter #:  0
=============
               Total Read  Total Write  Active Read  Active Write   Maximum
I/O:                88366       135054            0             0       258
SECTOR:           2495579      4184324            0             0      3897
```

We are interested in the Maximum field, which indicates the maximum number of I/Os submitted to the device since system boot. The Maximum for devstats does not exceed queue_depth x # paths for SDD when qdepth_enable=yes. But Maximum for adaptstats can exceed num_cmd_elems, because it represents the maximum number of I/Os submitted to the adapter driver and includes I/Os for both the service and wait queues. If, in this case, we have two paths and use the default queue_depth of 20, the 40 indicates that we filled the queue at least once and increasing queue_depth can help performance. For SDDPCM, if the Maximum value equals the hdisk queue_depth, the hdisk driver queue filled during the interval and increasing queue_depth is appropriate.

You can similarly monitor adapter queues and IOPS: for adapter IOPS, run **iostat -at *<interval> <# of intervals>*** and for adapter queue information, run **iostat -aD**, optionally with an interval and number of intervals.

The downside of setting queue depths too high is that the disk subsystem cannot handle the I/O requests in a timely fashion and might even reject the I/O or ignore it. This situation can result in an I/O timeout, and an I/O error recovery code is called. This situation is bad, because the processor ends up performing more work to handle I/Os than necessary. If the I/O eventually fails, this situation can lead to an application crash or worse.

Lower the queue depth per LUN when using multipathing. With multipathing, this default value is magnified, because it equals the default queue depth of the adapter multiplied by the number of active paths to the storage device. For example, because QLogic uses a default queue depth of 32, the suggested queue depth value to use is 16 when using two active paths and 8 when using four active paths. Directions for adjusting the queue depth are specific to each HBA driver and are available in the documentation for the HBA.

For more information about AIX, see this technote, *AIX disk queue depth tuning for performance*, available at this website:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105745

9.2 AIX disk I/O components

Since AIX Version 6, a set of tunable parameters from six tuning commands (**vmo**, **ioo**, **schedo**, **raso**, **no**, and **nfso**) is preset with default values that are optimized for most types of workloads, and the tunable parameters are classified as “restricted use” tunables. Change them only if instructed to do so by IBM support. For more information, see 6.2 “Performance management” in the *IBM AIX Version 7.1 Differences Guide,* SG24-7910:

http://www.redbooks.ibm.com/abstracts/sg247910.html

With AIX 7 (with the DS8000), you need to use the default parameters and install the filesets for multipathing and host attachment that already provide basic performance defaults for queue length and SCSI timeout. For additional information about setting up the volume layout, see 9.2.4, "IBM Logical Volume Manager" on page 343.

For AIX 5.3 or earlier, if you want additional information about the tunable commands and their parameters for a specific configuration, see the following link:

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds3/ioo.htm

For a complete discussion of AIX tuning, see the following links:

- *AIX 7.1 Performance tools guide and reference*, SC23-6733:

  http://publib.boulder.ibm.com/infocenter/aix/v7r1/topic/com.ibm.aix.prftools/doc/prftools/prftools.pdf

- *AIX 6.1 Performance Tuning Manual,* SC23-5253:

  http://publib.boulder.ibm.com/infocenter/systems/scope/aix/topic/com.ibm.aix.prftungd/doc/prftungd/file_sys_perf.htm?tocNode=int_214554

- *The Performance Tuning Manual* discusses the relationships between Virtual Memory Manager (VMM) and the buffers that are used by the filesystems and LVM:

  http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/com.ibm.aix.prftungd/doc/prftungd/vmm_page_replace_tuning.htm

- Tuning recommendations for Oracle on AIX paper:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100883

- Performance improvement of an Oracle database with concurrent I/O (CIO) paper:

  http://www.ibm.com/systems/resources/systems_p_os_aix_whitepapers_db_perf_aix.pdf

- How to optimize Sybase Adaptive Server Enterprise (ASE) on AIX paper:

  http://www.ibm.com/servers/enable/site/peducation/wp/b78a/b78a.pdf

9.2.1 AIX Journaled File System (JFS) and Journaled File System 2 (JFS2)

JFS and JFS2 are AIX standard filesystems. JFS was created for the 32-bit kernels and implements the concept of a transactional filesystem where all of the I/O operations of the metadata information are kept in a log. The practical impact is that in the recovery of a filesystem, the **fsck** command looks at that log to see what I/O operations completed and rolls back only those operations that were not completed. From a performance point of view, there is overhead. However, it is generally an acceptable compromise to ensure the recovery of a corrupted filesystem. Its file organization method is a linear algorithm. You can mount the filesystems with the Direct I/O option. You can adjust the mechanisms of sequential read ahead, sequential and random write behind, delayed write operations, and others. You can tune its buffers to increase the performance. It also supports asynchronous I/O.

JFS2 was created for 64-bit kernels. Its file organization method is a B+ tree algorithm. It supports all the features that are described for JFS, with exception of "delayed write operations." It also supports concurrent I/O (CIO).

AIX filesystem caching

In AIX, you can limit the amount of memory used for filesystem cache and the behavior of the page replacement algorithm. Configure the following parameters and use these suggested values:

- minperm% = 3
- maxperm% = 90
- maxclient% = 90
- strict_maxperm = 0
- strict_maxclient = 1
- lru_file_repage = 0
- lru_poll_interval = 10

Filesystem I/O buffers

AIX tracks I/O to disk by using pinned memory buffers (pbufs). In AIX 7.1, filesystem buffers are controlled by the following system-wide tuning parameters:

- numfsbufs
- lvm_bufcnt
- pd_npages
- v_pinshm
- j2_nBufferPerPagerDevice
- j2_dynamicBufferPreallocation

For more information, see the following link:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.prftungd%2Fdoc%2Fprftungd%2Ffs_buff_tuning.htm

Read ahead

JFS and JFS2 have read-ahead algorithms that can be configured to buffer data for sequential reads into the filesystem cache before the application requests it. Ideally, this feature reduces the percent of I/O wait (%iowait) and increases I/O throughput as seen from the operating system. Configuring the read ahead algorithms too aggressively results in unnecessary I/O. The following VMM tunable parameters control read-ahead behavior:

- For JFS:
  - minpgahead = max(2, <*application's blocksize*> / <*filesystem's blocksize*>)
  - maxpgahead = max(256, (<*application's blocksize*> / <*filesystem's blocksize*> * <*application's read ahead block count*>))
- For JFS2:
  - j2_minPgReadAhead = max(2, <*application's blocksize*> / <*filesystem's blocksize*>)
  - j2_maxPgReadAhead = max(256, (<*application's blocksize*> / <*filesystem's blocksize*> * <*application's read ahead block count*>))

I/O pacing

I/O pacing manages the concurrency to files and segments by limiting the processor resources for processes that exceed a specified number of pending write I/Os to a discrete file or segment. When a process exceeds the maxpout limit (high-water mark), it is put to sleep until the number of pending write I/Os to the file or segment is less than minpout (low-water mark). This pacing allows another process to access the file or segment.

Disabling I/O pacing improves backup times and sequential throughput. Enabling I/O pacing ensures that no single process dominates the access to a file or segment. AIX 6.1 and higher enables I/O pacing by default. In AIX 5.3, you needed to explicitly enable this feature. The feature is enabled by setting the sys0 settings of the minpout and maxpout parameters to 4096 and 8193 (**`lsattr -El sys0`**). To disable I/O pacing, simply set them both to zero. You can also limit the effect of setting global parameters by mounting filesystems by using an explicit 0 for minput and maxpout: **`mount -o minpout=0,maxpout=0 /u`**. Tuning the maxpout and minpout parameters might prevent any thread that performs sequential writes to a file from dominating system resources.

In certain circumstances, it is appropriate to enable I/O pacing:

- For PowerHA (HACMP™), we advise that you enable I/O pacing to ensure that heartbeat activities complete. If you enable I/O pacing, start with settings of maxpout=321 and minpout=240 by increasing the default values of maxpout=33 and minpout=24.
- Beginning with AIX 5.3, I/O pacing can be enabled at the filesystem level with **`mount`** command options.
- In AIX Version 6, I/O pacing is technically enabled but with such high settings that the I/O pacing is not active except under extreme situations.
- In AIX Version 7, I/O pacing parameters can be changed on mounted filesystems with the **`-o remount`** option.

Enabling I/O pacing improves user response time at the expense of throughput. For more information about I/O pacing, see the following link:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.prftungd%2Fdoc%2Fprftungd%2Fdisk_io_pacing.htm

Write behind

This parameter enables the operating system to initiate I/O that is normally controlled by the syncd. Writes are triggered when a specified number of sequential 16 KB clusters are updated:

- Sequential write behind:
  - numclust for JFS
  - j2_nPagesPerWriteBehindCluster and j2_nRandomCluster for JFS2
- Random write behind:
  - maxrandwrt for JFS
  - j2_maxRandomWrite

Setting j2_nPagesPerWriteBehindCluster to 0 disables JFS2 sequential write behind, and setting j2_maxRandomWrite=0 also disables JFS2 random write behind.

Mount options

Use *release behind* mount options when appropriate:

- The release behind mount option can reduce syncd and lrud overhead. This option modifies the filesystem behavior in such a way that it does not maintain data in JFS2 cache. You use these options if you know that data that goes into or out of certain filesystems is not requested again by the application before the data is likely to be paged out. Therefore, the lrud daemon has less work to do to free up cache and eliminates any syncd overhead for this filesystem. One example of a situation where you can use these options is if you have a Tivoli Storage Manager Server with disk storage pools in filesystems and you configured the read ahead mechanism to increase the throughput of data, especially when a migration takes place from disk storage pools to tape storage pools:
  - **`-rbr`** for release behind after a read
  - **`-rbw`** for release behind after a write
  - **`-rbrw`** for release behind after a read or a write
- Direct I/O (DIO):

- Bypass JFS/JFS2 cache.
- No read ahead.
- An option of the **mount** command.
- Useful for databases that use filesystems rather than raw logical volumes. If an application has its own cache, it does not make sense to also have the data in filesystem cache.
- Direct I/O is not supported on compressed filesystems.

For more information about DIO, see this website:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.prftungd%2Fdoc%2Fprftungd%2Fdirect_io_tuning.htm

- Concurrent I/O (CIO):
  - Same as DIO but without inode locking, so the application must ensure data integrity for multiple simultaneous I/Os to a file.
  - An option of the **mount** command.
  - Not available for JFS.
  - When possible, consider the use of a no cache option at the database level when it is available rather than at the AIX level. An example is to for IBM DB2 to control the 'no_filesystem_cache' option. IBM DB2 can control it at the tablespace level. With a 'no filesystem caching' policy enabled in DB2, a specific OS call is made that uses CIO regardless of the mount. By setting CIO as a mount option, *all* files in the filesystem are CIO-enabled, which might not benefit certain files.

Asynchronous I/O

Asynchronous I/O (AIO) is the AIX facility that allows an application to issue an I/O request and continue processing without waiting for the I/O to finish:

- With AIX Version 6, the tunables **fastpath** and **fsfastpath** are classified as restricted tunables and now are set to a value of 1, by default. Therefore, all asynchronous I/O requests to a raw logical volume are passed directly to the disk layer by using the corresponding strategy routine (legacy AIO or POSIX-compliant AIO). Or, all asynchronous I/O requests for files opened with cio are passed directly to LVM or disk by using the corresponding strategy routine.

  Also, there are no more AIO devices in Object Data Manager (ODM) and all their parameters now become tunables using the **ioo** command. The newer **aioo** command is removed. For additional information, see the *IBM AIX Version 6.1 Differences Guide,* SG24-7559, and the *IBM AIX Version 7.1 Differences Guide,* SG24-7910:

  - http://www.redbooks.ibm.com/abstracts/sg247559.html
  - http://www.redbooks.ibm.com/abstracts/sg247910.html

9.2.2 Symantec Veritas File System (VxFS) for AIX

Veritas developed the Symantec Veritas File System. Implementation and administration are similar to the same versions on Oracle Solaris and HP-UX. For more information, see the *Veritas File System for AIX Administrator's Guide*:

http://sfdoccentral.symantec.com/sf/5.1/aix/pdf/vxfs_admin.pdf

9.2.3 General Parallel File System (GPFS)

GPFS is a concurrent filesystem that can be shared among the nodes that compose a cluster through a SAN or through a high-speed TCP/IP network. Beginning with Version 2.3, GPFS does not require an LVM or a PowerHA cluster. You use it for sharing data among the servers of a cluster concurrently without losing the facility to manipulate files that a standard filesystem provides. It implements Direct I/O and filesystem caching among other features.

GPFS memory buffers

The pagepool sets the size of buffer cache on each node. For a database system, 100 MB might be enough. For an application with many small files, you might need to increase this setting to 2 GB - 4 GB of RAM memory.

The maxFilesToCache parameter sets the number of inodes to cache for recently used files. The default value is 1000. If the application has many small files, consider increasing this value. There is a limit of 300000 tokens.

The maxStatCache sets the number of inodes to keep in stat cache. This value needs to be four times the size of the maxFilesToCache value.

Number of threads

The workerThreads parameter controls the maximum number of concurrent file operations at any instant. The suggested value is the same number of maxservers in AIX 5.3. There is a limit of 550 threads.

The prefetchThreads parameter controls the maximum number of threads dedicated to prefetching data for files that are read sequentially or to handle sequential write behind. For Oracle RAC, set this value for 548. There is a limit of 550 threads.

maxMBpS

Increase the maxMBpS to 80% of the total bandwidth for all HBAs in a single host. The default value is 150 MB/s.

maxblocksize

Configure the GPFS blocksize (maxblocksize) to match the application I/O size, the RAID stripe size, or a multiple of the RAID stripe size. For example, if you use an Oracle database, it is better to adjust a value that matches the product of the value of the DB_BLOCK_SIZE and DB_FILE_MULTIBLOCK_READ_COUNT parameters. If the application performs many sequential I/Os, it is better to configure a blocksize from 8 to 16 MB to take advantage of the sequential prefetching algorithm on the DS8000.

For additional information, see the following links:

- GPFS manuals

  http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfsbooks.html

- Tuning considerations in the *Concepts, Planning, and Installation Guide V3.4,* GA76-0413-04:

  http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=%2Fcom.ibm.cluster.gpfs.v3r4.gpfs100.doc%2Fbl1ins_BookMap_xtoc.html

- *Deploying Oracle 10g RAC on AIX V5 with GPFS,* SG24-7541:

  http://www.redbooks.ibm.com/abstracts/sg247541.html?Open

- *Configuration and Tuning GPFS for Digital Media Environments,* SG24-6700:

http://www.redbooks.ibm.com/abstracts/sg246700.html?Open

9.2.4 IBM Logical Volume Manager

IBM Logical Volume Manager (LVM) is the standard LVM that comes with AIX. It is an abstraction layer that allows storage virtualization at the operating system level. It is possible to implement RAID 0 or RAID 1, or a combination of both RAID types. It is also possible to spread the data over the LUNs in a round-robin manner. You can configure the buffer sizes to optimize performance. Figure 9-4 shows an overview.

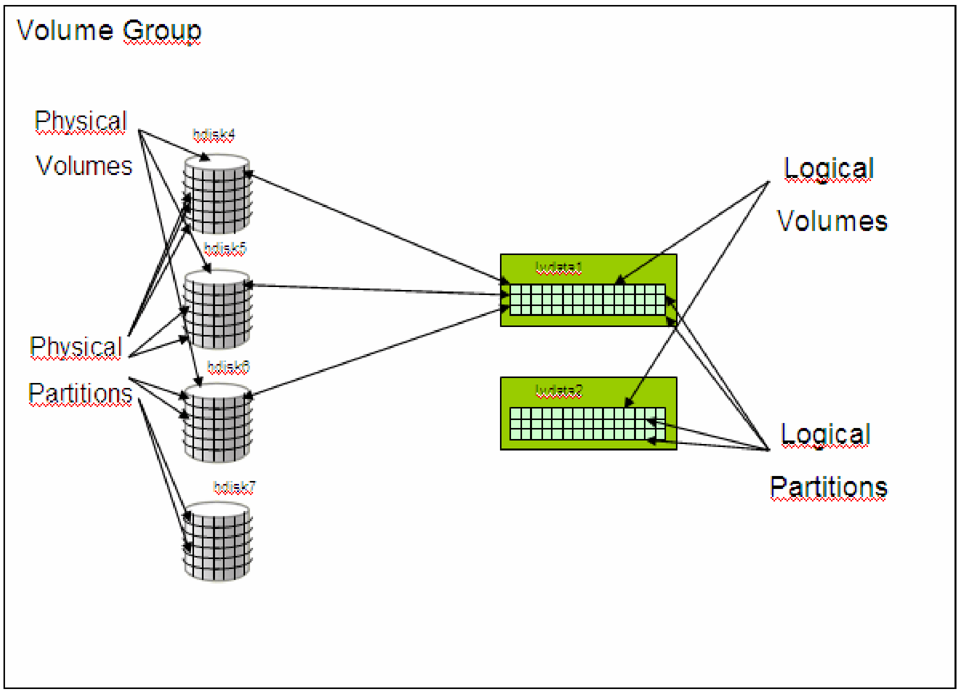


Figure 9-4 IBM LVM overview

In Figure 9-4, the DS8000 LUNs that are under the control of the LVM are called *physical volumes* (PVs). The LVM splits the disk space into smaller pieces, which are called *physical partitions* (PPs). A *logical volume* (LV) consists of several *logical partitions* (LPs). A filesystem can be mounted over an LV, or it can be used as a raw device. Each LP can point to up to three corresponding PPs. The ability of the LV to point a single LP to multiple PPs is the way that LVM implements mirroring (RAID 1).

To set up the volume layout with the DS8000 LUNs, you can adopt one of the following strategies:

- Storage pool striping: In this case, you are spreading the workload at the storage level. At the operating system level, you need to create the LVs with the inter-policy attribute set to minimum, which is the default option when creating an LV.
- PP Striping: A set of LUNs are created in different ranks inside of the DS8000. When the LUNs are recognized in AIX, a volume group (VG) is created. The LVs are spread evenly over the LUNs by setting the inter-policy to maximum, which is the most common method used to distribute the workload. The advantage of this method compared to storage pool striping is the granularity of data spread over the LUNs. With storage pool striping, the data is spread in chunks of 1 GB. In a VG, you can create PP sizes from 8 MB to 16 MB. The advantage of this method compared to LVM Striping is that you have

more flexibility to manage the LVs, such as adding more disks and redistributing evenly the LVs across all disks by reorganizing the VG.

- LVM Striping: As in the PP Striping, a set of LUNs is created in different ranks inside of the DS8000. After the LUNs are recognized in AIX, a VG is created with larger PP sizes, such as 128 MB or 256 MB. And the LVs are spread evenly over the LUNs by setting the stripe size of LV from 8 MB to 16 MB. From a performance standpoint, LVM Striping and PP Striping provide the same performance. You might see an advantage in a scenario of PowerHA with LVM Cross-Site and VGs of 1 TB or more when you perform cluster verification. Or, you see that operations related to creating, modifying, or deleting LVs are faster.

Volume group limits

When creating the volume group, there are LVM limits to consider along with the potential expansion of the volume group. The key LVM limits for a volume group are shown in Table 9-1.

Table 9-1 Volume group characteristics

Limit	Standard VG	Big VG	Scalable VG
Maximum PVs/VG	32	128	1024
Maximum LVs/VG	256	512	4096
Maximum PPs/VG	32512	130048	2097152

Suggestion: Use AIX scalable volume groups whenever possible.

PP Striping

Figure 9-5 on page 345 shows an example of PP Striping. The volume group contains four LUNs and created 16 MB physical partitions on the LUNs. The logical volume in this example consists of a group of 16 MB physical partitions from four logical disks: hdisk4, hdisk5, hdisk6, and hdisk7.

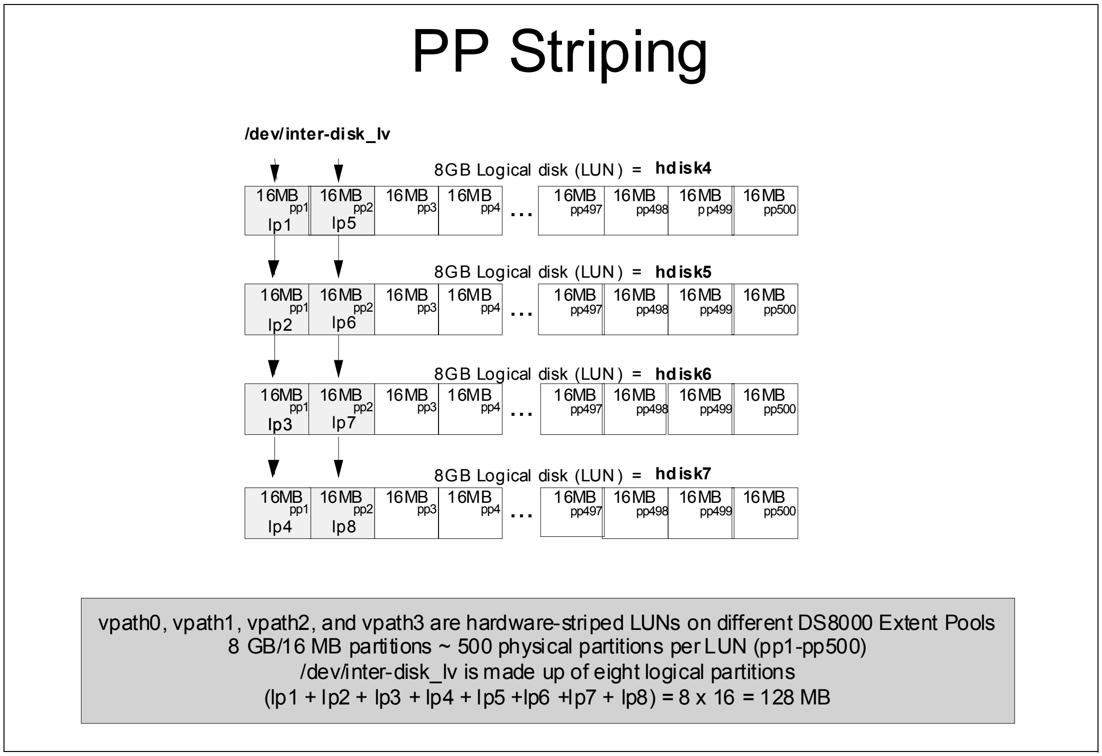


Figure 9-5 Inter-disk policy logical volume

The first step is to create a volume group. We suggest that you create a VG with a set of DS8000 LUNs where each LUN is in a separate extent pool. If you plan to add a set of LUNs to a host, define another VG. For you to create a VG, execute the following command to create the data01vg and a PP size of 16 MB:

```
mkvg -S -s 16 -y data01vg hdisk4 hdisk5 hdisk6 hdisk7
```

> **Commands:** To create the volume group, if you use SDD, you use the **mkvg4vp** command. If you use SDDPCM, you use the **mkvg** command. All the flags for the **mkvg** command apply to the **mkvg4vp** command.

After you create the VG, the next step is to create the LVs. To create a VG with four disks (LUNs), we suggest that you create the LVs as a multiple of the number of disks in the VG, times the PP size. In our case, we create the LVs in multiples of 64 MB. You can implement the PP Striping by using the option **-e x**. By adding the **-a e** option, the Intra-Physical Volume Allocation Policy changes the allocation policy from *middle* (default) to *edge* so that the LV physical partitions are allocated beginning at the outer edge and continuing to the inner edge. This method ensures that all physical partitions are sequentially allocated across the physical disks. To create an LV of 1 GB, execute the following command:

```
mklv -e x -a e -t jfs2 -y inter-disk_lv data01vg 64 hdisk4 hdisk5 hdisk6 hdisk7
```

Preferably, use inline logs for JFS2 logical volumes, which results in one log for every filesystem and it is automatically sized. Having one log per filesystem improves performance, because it avoids the serialization of access when multiple filesystems make metadata changes. The disadvantage of inline logs is that they cannot be monitored for I/O rates, which can provide an indication of the rate of metadata changes for a filesystem.

LVM Striping

Figure 9-6 shows an example of a striped logical volume. The logical volume called /dev/striped_lv uses the same capacity as /dev/inter-disk_lv (shown in Figure 9-5 on page 345), but it is created differently.

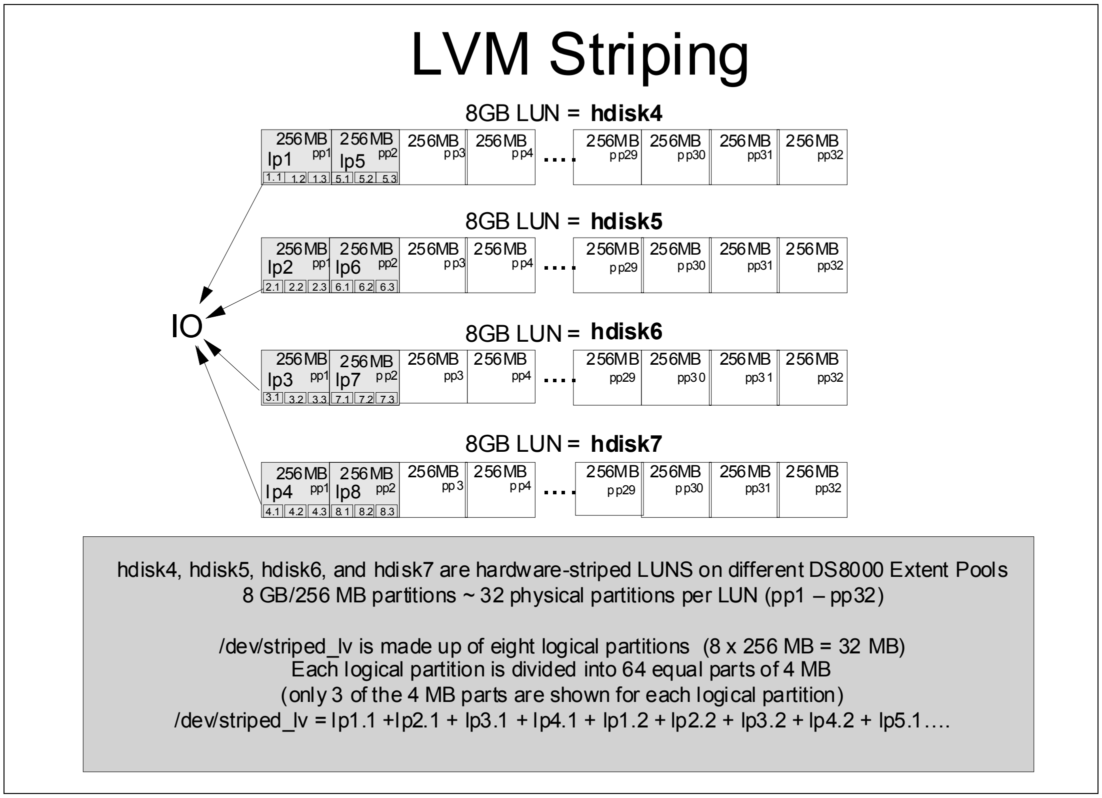


Figure 9-6 Striped logical volume

Notice that /dev/striped_lv is also made up of eight 256 MB physical partitions, but each partition is then subdivided into 32 chunks of 8 MB; only three of the 8 MB chunks are shown per logical partition for space reasons.

Again, the first step is to create a VG. To create a VG for LVM Striping, execute the following command:

```
mkvg -S -s 256 -y data01vg hdisk4 hdisk5 hdisk6 hdisk7
```

For you to create a striped LV, you need to combine the following options when you use LVM Striping:

- Stripe width (**-C**): This option sets the maximum number of disks to spread the data. The default value is used from the option upperbound (**-u**).
- Copies (**-c**): This option is only required when you create mirrors. You can set from 1 to 3 copies. The default value is 1.
- Strict allocation policy (**-s**): This option is required only when you create mirrors and it is necessary to use the value “s” (superstrict).
- Stripe size (**-S**): This option sets the size of a chunk of a sliced PP. Since AIX 5.3, the valid values include 4K, 8K, 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M, 32M, 64M, and 128M.

- Upperbound (**-u**): This option sets the maximum number of disks for a new allocation. If you set the allocation policy to superstrict, the upperbound value must be the result of the stripe width times the number of copies that you want to create.

Important: Do not set the option **-e** with LVM Striping.

Execute the following command to create an striped LV:

```
mklv -C 4 -c 1 -s s -S 8 -t jfs2 -u 4 -y striped_lv data01vg 4 hdisk4 hdisk5
hdisk6 hdisk7
```

AIX 5.3 implemented a new feature, the *striped column*. With this feature, you can extend an LV in a new set of disks after the current disks where the LV is spread is full.

Memory buffers

Adjust the memory buffers (pv_min_pbuf) of LVM to increase the performance. Set it to 2048 for AIX 7.1.

Scheduling policy

If you have a dual-site cluster solution that uses PowerHA with LVM Cross-Site, you can reduce the link requirements among the sites by changing the scheduling policy of each LV to parallel write/sequential read (ps). You must remember that the first copy of the mirror needs to point to the local storage.

9.2.5 Symantec Veritas Volume Manager (VxVM)

VxVM is another LVM. With AIX, VxVM can replace the IBM LVM for rootvg. The Veritas Volume Manager for AIX is similar to other platforms, such as Oracle Solaris and HP-UX. See the following links:

http://www.ibm.com/developerworks/aix/library/au-aixveritas/

http://sfdoccentral.symantec.com/sf/5.1/aix/pdf/vxvm_admin.pdf

9.2.6 IBM Subsystem Device Driver (SDD) for AIX

SDD is an IBM proprietary multipathing software that works with the DS8000 and other IBM storage devices only. It was popular, but with SDDPCM and IBM AIX MPIO, a more modern alternative is available now. For more information, see the following link:

http://www.ibm.com/support/docview.wss?rs=540&uid=ssg1S7001350#AIXSDD

9.2.7 MPIO with SDDPCM

MPIO is another multipathing device driver. It was introduced in AIX 5.2. The reason for providing your own multipathing solution is that in a SAN environment you might want to connect to several storage subsystems from a single host. Each storage vendor has its own multipathing solution that is not interoperable with the multipathing solution of other storage vendors. This restriction increases the complexity of managing the compatibility of operating system fix levels, HBA firmware levels, and multipathing software versions. AIX provides the base MPIO device driver; however, it is still necessary to install the MPIO device driver that is provided by the storage vendor to take advantage of all of the features of a multipathing solution. We prefer to use MPIO with SDDPCM rather than SDD with AIX whenever possible. MPIO with SDDPCM is not supported with PowerHA/XD.

For additional information about MPIO, see the following links:

- *Multipath I/O description in AIX 7.1*:

  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.prftungd%2Fdoc%2Fprftungd%2Fwait_io_time_reporting.htm

- A paper from Oracle that provides the best practices of Automatic Storage Management (ASM) with Multipathing:

  http://www.oracle.com/technetwork/database/asm.pdf

  http://docs.oracle.com/cd/B28359_01/server.111/b31107/asmprepare.htm

- A paper from Oracle titled *ASM Overview and Technical Best Practices*:

  http://www.oracle.com/technology/products/database/asm/pdf/asm_10gr2_bestpractices%209-07.pdf

- Check the interoperability matrix of SDDPCM (MPIO) to see which version is supported:

  http://www.ibm.com/support/docview.wss?rs=540&uid=ssg1S7001350#AIXSDDPCM

- MPIO is only supported with PowerHA (HACMP) if you configure the VGs in Enhanced Concurrent Mode. For additional information about PowerHA with MPIO, see this link:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10504

- If you use a multipathing solution with Virtual I/O Server (VIOS), use MPIO. There are several limitations when you use SDD with VIOS. See 9.2.10, "Virtual I/O Server (VIOS)" on page 350 and the VIOS support site for additional information:

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html#multipath

9.2.8 Symantec Veritas Dynamic MultiPathing (DMP) for AIX

Symantec Veritas Dynamic MultiPathing (DMP) is a device driver that is provided by Symantec to work with VxVM. If you use VxVM, use DMP instead of SDD. Also, there is an option for you to use VxVM with SDD. For additional information, see the following links:

http://www.symantec.com/business/support/index?page=content&id=DOC2728

http://sfdoccentral.symantec.com/sf/5.1SP1PR1/aix/pdf/dmp_admin_51sp1pr1_aix.pdf

9.2.9 FC adapters

FC adapters or host bus adapters (HBAs) provide the connection between the host and the storage devices. We suggest that you configure the following four important parameters:

- num_cmd_elems: This parameter sets the maximum number of commands to queue to the adapter. When many supported storage devices are configured, you can increase this attribute to improve performance. The range of supported values depends on the FC adapter. Check with **`lsattr -Rl fcs0 -a num_cmd_elems`**.

- max_xfer_size: This attribute for the fscsi device, which controls the maximum I/O size that the adapter device driver can handle, also controls a memory area used by the adapter for data transfers. When the default value is used (**`max_xfer_size=0x100000`**), the memory area is 16 MB in size. When setting this attribute to any other allowable value (for example, 0x200000), the memory area is 128 MB in size. For typical DS8000 environments, this setting must remain unchanged and use its default value. Any other change might imply risks and not lead to performance improvements.

  The **`fcstat`** command can be used to examine whether increasing num_cmd_elems or max_xfer_size can increase performance. In selected environments with heavy I/O and especially large I/Os (such as for backups), carefully consider increasing this setting to max_xfer_size=0x200000 and verify whether you improve performance (see the following note and take appropriate precautions when testing this setting).

At AIX 6.1 TL2 or later, a change was made for virtual FC adapters so that the DMA memory area is always 128 MB even with the default max_xfer_size. This memory area is a DMA memory area, but it is different from the DMA memory area that is controlled by the lg_term_dma attribute (which is used for I/O control). The default value for lg_term_dma of 0x800000 is adequate.

> **Changing the max_xfer_size:** Changing max_xfer_size uses memory in the PCI Host Bridge chips attached to the PCI slots. The sales manual, regarding the dual-port 4 Gbps PCI-X FC adapter states that "*If placed in a PCI-X slot rated as SDR compatible and/or has the slot speed of 133 MHz, the AIX value of the max_xfer_size must be kept at the default setting of 0x100000 (1 megabyte) when both ports are in use. The architecture of the DMA buffer for these slots does not accommodate larger max_xfer_size settings*." Issues occur when configuring the LUNs if there are too many FC adapters and too many LUNs attached to the adapter. Errors, such as `DMA_ERR` might appear in the error report. If you get these errors, you need to change the max_xfer_size back to the default value. Also if you boot from SAN, if you encounter this error, you cannot boot, so be sure to have a back-out plan if you plan to change the max_xfer_size and boot from SAN.

- dyntrk: IBM AIX supports dynamic tracking of FC devices. Previous releases of AIX required a user to unconfigure FC storage device and adapter device instances before changing the system area network (SAN), which can result in an N_Port ID (SCSI ID) change of any remote storage ports. If dynamic tracking of FC devices is enabled, the FC adapter driver detects when the Fibre Channel N_Port ID of a device changes. The FC adapter driver then reroutes traffic destined for that device to the new address while the devices are still online. Events that can cause an N_Port ID to change include moving a cable between a switch and storage device from one switch port to another, connecting two separate switches that use an inter-switch link (ISL), and possibly rebooting a switch. Dynamic tracking of FC devices is controlled by a new fscsi device attribute, dyntrk. The default setting for this attribute is no. To enable dynamic tracking of FC devices, set this attribute to **`dyntrk=yes`**, as shown in the example: **`chdev -l fscsi0 -a dyntrk=yes`**.

> **Important:** The disconnected cable must be reconnected within 15 seconds.

- fc_err_recov: IBM AIX supports Fast I/O Failure for FC devices after link events in a switched environment. If the FC adapter driver detects a link event, such as a lost link between a storage device and a switch, the FC adapter driver waits a short time, approximately 15 seconds, so that the fabric can stabilize. At that point, if the FC adapter driver detects that the device is not on the fabric, it begins failing all I/Os at the adapter driver. Any new I/O or future retries of the failed I/Os are failed immediately by the adapter until the adapter driver detects that the device rejoins the fabric. Fast Failure of I/O is controlled by a new fscsi device attribute, fc_err_recov. The default setting for this attribute is delayed_fail, which is the I/O failure behavior seen in previous versions of AIX. To enable Fast I/O Failure, set this attribute to **`fast_fail`**, as shown in the example: **`chdev -l fscsi0 -a fc_err_recov=fast_fail`**.

> **Important:** Change the attributes fc_err_recov to fast_fail and dyntrk to yes only if you use a multipathing solution with more than one path.

Example 9-3 on page 350 is the output of the attributes of an fcs device.

Example 9-3 Output of an fcs device

```
# lsattr -El fcs0
bus_intr_lvl  8673       Bus interrupt level                           False
bus_io_addr   0xffc00    Bus I/O address                               False
bus_mem_addr  0xfffbf000 Bus memory address                            False
init_link     al         INIT Link flags                               True
intr_priority 3          Interrupt priority                            False
lg_term_dma   0x800000   Long term DMA                                 True
max_xfer_size 0x100000   Maximum Transfer Size                         True
num_cmd_elems 200        Maximum number of COMMANDS to queue to the adapter True
pref_alpa     0x1        Preferred AL_PA                               True
sw_fc_class   2          FC Class for Fabric                           True
```

Example 9-4 is the output of the attributes of an fscsi device.

Example 9-4 Output of an fscsi device

```
# lsattr -El fscsi0
attach       switch       How this adapter is CONNECTED         False
dyntrk       no           Dynamic Tracking of FC Devices        True
fc_err_recov delayed_fail FC Fabric Event Error RECOVERY Policy True
scsi_id      0x490e00     Adapter SCSI ID                       False
sw_fc_class  3            FC Class for Fabric                   True
```

Regarding the Fast I/O Failure (fc_err_recov) and Dynamic Tracking (dyntrk) options, for more information, see the following links:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.prftungd%2Fdoc%2Fprftungd%2Ffast_io_failure.htm

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.kernelext%2Fdoc%2Fkernextc%2Ffastfail_dyntracking.htm

Also, see the information about **num_cmd_elems** and **max_xfer_size** in *AIX disk queue depth tuning for performance*:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105745

For additional information, see the *SDD User's Guide* at the following link:

http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303

9.2.10 Virtual I/O Server (VIOS)

The Virtual I/O Server (VIOS) is an appliance that provides virtual storage and Shared Ethernet adapter capability to client logical partitions (LPARs) on POWER7® and POWER6. It is built on top of AIX and uses a default AIX user *padmin* that runs in a restricted shell to execute the available commands provided by the ioscli command-line interface (CLI) only.

The VIOS allows a physical adapter with disk attached at the VIOS partition level to be shared by one or more partitions and enables clients to consolidate and potentially minimize the number of required physical adapters. See Figure 9-7 on page 351 for an illustration.

The VIOS works in the following manner:

- On the DS8000 side, the same set of LUNs for the VIOSs is assigned to the corresponding volume groups for LUN masking.

- At the VIOS, there are at least two LPARs defined. They are considered SCSI servers that provide access to the DS8000 LUNs for the other LPARs. We advise that you have two or more HBAs installed on each VIOS.

  For every LPAR VSCSI client, there is a *Virtual SCSI Server* (VSCSI target) defined in both VIOSs through the Hardware Management Console (HMC). Similarly, for every VIOS, there is a *VSCSI Client device* (VSCSI Initiator) defined for the corresponding LPAR VSCSI Client. The host attachments for SDDPCM and SDDPCM for AIX are installed in the VIOSs through the **oem_setup_env** command, as with an ordinary AIX server. For all DS8000 LUNs that you map directly to the corresponding LPAR Client vscsi device, you also need to disable the SCSI reservation first.

- The LPAR that is the VSCSI Client needs the basic MPIO device driver only (MPIO works in failover mode only). You can use the VSCSI disk devices like any other ordinary disk in the AIX.

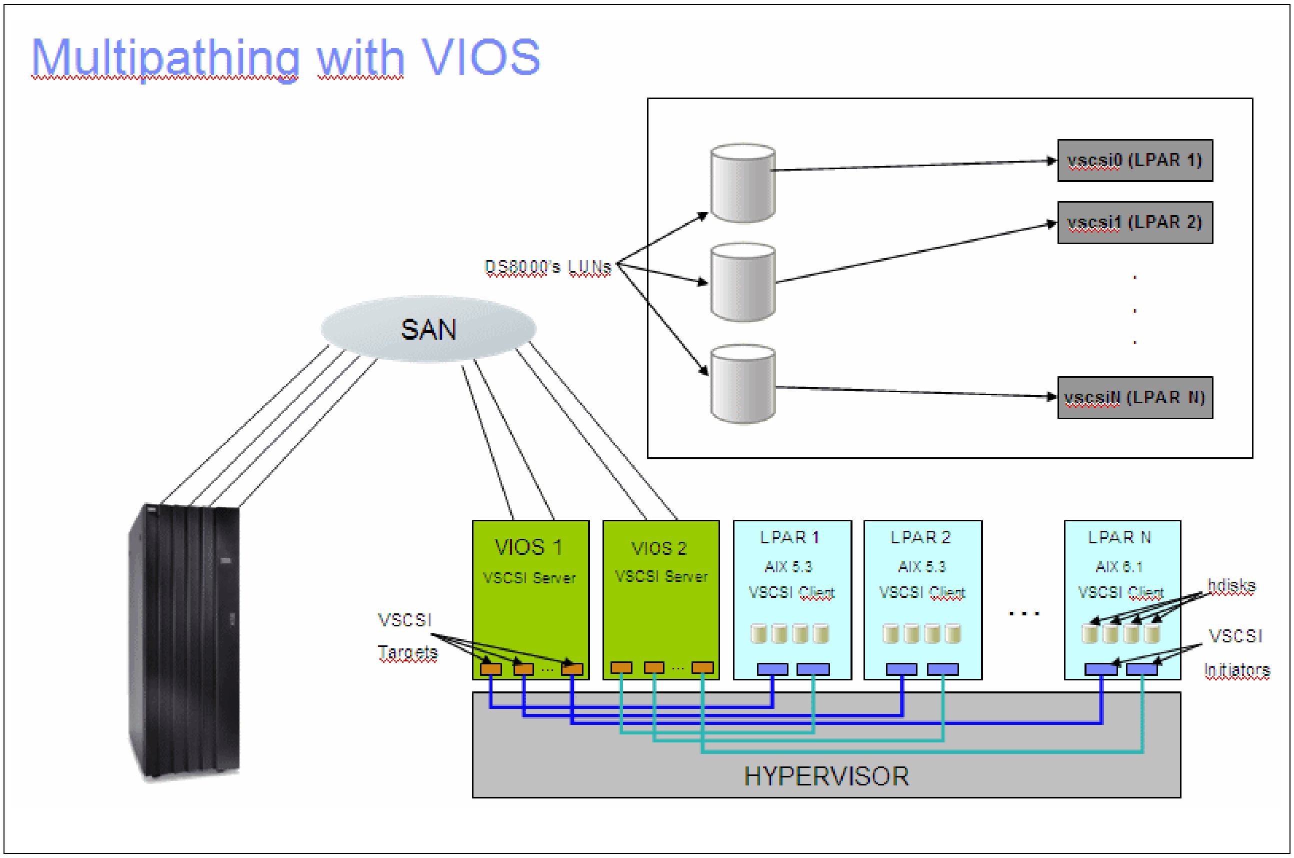


Figure 9-7 Multipathing with VIOS overview

When you assign several LUNs from the DS8000 to the VIOS and then map those LUNs to the LPAR clients with the time, trivial activities, such as upgrading the SDDPCM device driver, can become challenging. For additional information about VIOS, see the following links:

- *IBM PowerVM Virtualization Introduction and configuration*:

  http://www.redbooks.ibm.com/redbooks/pdfs/sg247940.pdf

- *IBM PowerVM Virtualization Managing and Monitoring*:

  http://www.redbooks.ibm.com/abstracts/sg247590.html

- *Virtual I/O Server and Integrated Virtualization Manager command descriptions*:

  http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/iphb1/iphb1_vios_commandslist.htm

- Also, check the VIOS frequently asked questions (FAQs) that explain in more detail several restrictions and limitations, such as the lack of the load balancing feature for AIX VSCSI MPIO devices:

  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/faq.html

Performance suggestions

We suggest these performance settings when configuring Virtual SCSI for performance:

- Processor:
  - Typical entitlement is 0.25
  - Virtual processor of 2
  - Always run uncapped
  - Run at higher priority (weight factor >128)
  - More processor power with high network loads
- Memory:
  - Typically >= 10 GB (at least 1 GB of memory is required. The minimum is 2 GB + 20 MB per hdisk.) This type of a configuration is needed to avoid any paging activity in the VIOS, which might lead to performance degradation in all LPARs.
  - Add more memory if there are high device (vscsi and hdisk) counts.
  - Small LUNs drive up the memory requirements.
- For multipathing with VIOS, check the configuration of the following parameters:
  - fscsi devices on VIOS:
    - The attribute fc_err_recov is set to fast_fail
    - The attribute dyntrk is set to yes with the command **`chdev -l fscsiX -a dyntrk=yes`**
  - hdisk devices on VIOS:
    - The attribute algorithm is set to load_balance
    - The attribute reserve_policy is set to no_reserve
    - The attribute hcheck_mode is set to nonactive
    - The attribute hcheck_interval is set to 20
  - vscsi devices in client LPARs:

    The attribute vscsi_path_to is set to 30
  - hdisk devices in client:
    - The attribute algorithm is set to failover
    - The attribute reserv_policy is set to no_reserve
    - The attribute hcheck_mode is set to nonactive
    - The attribute hcheck_interval is set to 20

Important: Change the reserve_policy parameter to no_reserve only if you are going to map the LUNs of the DS8000 directly to the client LPAR.

For additional information, see the following link:

http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/p7hb1/iphb1_vios_planning_vscsi.htm

9.3 AIX performance monitoring tools

In this section, we briefly review the various AIX commands and utilities that are useful for performance monitoring.

9.3.1 AIX vmstat

The **vmstat** utility is a useful tool for taking a quick snapshot of the system performance. It is the first step toward understanding the performance issue and specifically in determining whether the system is I/O bound.

Example 9-5 shows how **vmstat** can help you monitor filesystem activity by using the command **vmstat -I**.

Example 9-5 The vmstat -I utility output for filesystem activity analysis

```
[root@p520-tic-3]# vmstat -I 1 5

System Configuration: lcpu=30 mem=3760MB

kthr      memory             page              faults        cpu
-------- ----------- ------------------------ ------------ -----------
 r  b  p   avm   fre  fi  fo  pi  po  fr  sr   in   sy  cs us sy id wa
10  1 42 5701952 10391285 338 715   0   0 268  541 2027 46974 13671 18  5 60 16
10  0 88 5703217 10383443   0 6800   0   0   0    0 9355 72217 26819 14 10 15 62
 9  0 88 5697049 10381654   0 7757   0   0   0    0 10107 77807 28646 18 11 11 60
 8  1 70 5692366 10378376   0 8171   0   0   0    0 9743 80831 26634 20 12 13 56
 5  0 74 5697938 10365625   0 6867   0   0   0    0 11986 63476 28737 13 10 13 64
10  0 82 5698586 10357280   0 7745   0   0   0    0 12178 66806 29431 14 11 12 63
12  0 80 5704760 10343915   0 7272   0   0   0    0 10730 74279 29453 16 11 11 62
 6  0 84 5702459 10337248   0 9193   0   0   0    0 12071 72015 30684 15 12 11 62
 6  0 80 5706050 10324435   0 9183   0   0   0    0 11653 72781 31888 16 10 12 62
 8  0 76 5700390 10321102   0 9227   0   0   0    0 11822 82110 31088 18 14 12 56
```

In an I/O-bound system, look for these indicators:

- A high I/O wait percentage as shown in the “cpu” column under the “wa” subcolumn. Example 9-5 shows that a majority of CPU cycles are waiting for I/O operations to complete.
- A high number of blocked processes as shown in the “kthr” column under the subcolumns “b” and “p”, which are the wait queue (b) and wait queue for raw devices (p). A high number of blocked processes normally indicate I/O contention among the process.
- Paging activity as seen under the column “page.” High first in first out (FIFO) indicates intensive file caching activity.

Example 9-6 on page 354 shows you another option that you can use, **vmstat -v**, from which you can understand whether the blocked I/Os are due to a shortage of buffers.

Example 9-6 The vmstat -v utility output for filesystem buffer activity analysis

```
[root@p520-tic-3]# vmstat -v | tail -7
                    0 pending disk I/Os blocked with no pbuf
                    0 paging space I/Os blocked with no psbuf
                 2484 filesystem I/Os blocked with no fsbuf
                    0 client filesystem I/Os blocked with no fsbuf
                    0 external pager filesystem I/Os blocked with no fsbuf
                    0 Virtualized Partition Memory Page Faults
                 0.00 Time resolving virtualized partition memory page faults
```

In Example 9-6, notice these characteristics:

- Filesystem buffer (fsbuf) and LVM buffer (pbuf) space are used to hold the I/O request at the filesystem and LVM levels.
- If a substantial number of I/Os are blocked due to insufficient buffer space, both buffers can be increased by using the **ioo** command, but a larger value results in overall poor system performance. We suggest that you increase buffers incrementally and monitor the system performance with each increase.

For the preferred practice values, see the application papers listed under “AIX filesystem caching” on page 338.

By using **lvmo**, you can also check whether contention is happening due to a lack of LVM memory buffer, which is illustrated in Example 9-7.

Example 9-7 Output of lvmo -a

```
[root@p520-tic-3]# lvmo -a -v rootvg
vgname = rootvg
pv_pbuf_count = 512
total_vg_pbufs = 1024
max_vg_pbuf_count = 16384
pervg_blocked_io_count = 0
pv_min_pbuf = 512
global_blocked_io_count = 0
```

As you can see in Example 9-7, there are two incremental counters: pervg_blocked_io_count and global_blocked_io_count. The first counter indicates how many times an I/O block happened because of a lack of LVM pinned memory buffer (pbufs) on that VG. The second incremental counter counts how many times an I/O block happened due to the lack of LVM pinned memory buffer (pbufs) in the whole OS. Other indicators for I/O bound can be seen with the disk xfer part of the vmstat output when run against the physical disk as shown in Example 9-8.

Example 9-8 Output of vmstat for disk xfer

```
# vmstat hdisk0 hdisk1 1 8
kthr     memory             page              faults        cpu     disk xfer
----  ----------  ----------------------- ------------ ----------- ------
r  b   avm   fre  re  pi  po  fr  sr  cy  in   sy  cs  us sy id wa  1 2 3 4
0  0  3456 27743   0   0   0   0   0   0 131  149  28  0  1 99  0  0 0
0  0  3456 27743   0   0   0   0   0   0 131   77  30  0  1 99  0  0 0
1  0  3498 27152   0   0   0   0   0   0 153 1088  35  1 10 87  2  0 11
0  1  3499 26543   0   0   0   0   0   0 199 1530  38  1 19  0 80  0 59
0  1  3499 25406   0   0   0   0   0   0 187 2472  38  2 26  0 72  0 53
0  0  3456 24329   0   0   0   0   0   0 178 1301  37  2 12 20 66  0 42
0  0  3456 24329   0   0   0   0   0   0 124   58  19  0  0 99  0  0 0
0  0  3456 24329   0   0   0   0   0   0 123   58  23  0  0 99  0  0 0
```

The disk xfer part provides the number of transfers per second to the specified physical volumes that occurred in the sample interval. This count does not imply an amount of data that was read or written.

For additional details, see the man page of vmstat:

http://publib.boulder.ibm.com/infocenter/systems/scope/aix/topic/com.ibm.aix.cmds/doc/aixcmds6/vmstat.htm?tocNode=int_216867

9.3.2 pstat

The **pstat** command counts how many legacy asynchronous I/O servers are used in the server. There are two asynchronous I/O subsystems (AIOs):

- Legacy AIO
- Posix AIO

You can use the command **psat -a | grep 'aioserver' | wc -l** to get the number of legacy AIO servers that are running. You can use the command **pstat -a | grep posix_aioserver | wc -l** to see the number of Posix AIO servers.

Example 9-9 pstat -a output to measure the legacy AIO activity

```
[root@p520-tic-3]# pstat -a | grep ' aioserver' | wc -l
       0
[root@p520-tic-3]#
```

> **Important:** If you use raw devices, you have to use **ps -k** instead of **pstat -a** to measure the legacy AIO activity.

Example 9-9 shows that the host does not have any AIO servers that are running. This function is not enabled, by default. You can enable this function with **mkdev -l aio0** or by using **SMIT**. For Posix AIO, substitute **posix_aio** for **aio0**.

In AIX Version 6 and Version 7, both AIO subsystems are loaded by default but are activated only when an AIO request is initiated by the application. Use the command **pstat -a | grep aio** to see the AIO subsystems that are loaded, as shown in Example 9-10.

Example 9-10 pstat -a output to show the AIO subsystem defined in AIX 6

```
[root@p520-tic-3]# pstat -a | grep aio
18 a    1207c       1  1207c     0     0     1  aioLpool
33 a    2104c       1  2104c     0     0     1  aioPpool
```

In AIX Version 6 and Version 7, you can use the new **ioo** tunables to show whether the AIO is used. An illustration is given in Example 9-11.

Example 9-11 ioo -a output to show the AIO subsystem activity in AIX 6

```
[root@p520-tic-3]# ioo -a | grep aio
                   aio_active = 0
                  aio_maxreqs = 65536
                   aio_maxservers = 30
                   aio_minservers = 3
                   aio_server_inactivity = 300
                   posix_aio_active = 0
                   posix_aio_maxreqs = 65536
                   posix_aio_maxservers = 30
```

```
posix_aio_minservers = 3
posix_aio_server_inactivity = 300
```

From Example 9-11 on page 355, **aio_active** and **posix_aio_active** show whether the AIO is used. The parameters **aio_server_inactivity** and **posix_aio_server_inactivity** show how long an AIO server sleeps without servicing an I/O request.

To check the Asynchronous I/O configuration in AIX 5.3, type the following commands that are shown in Example 9-12.

Example 9-12 lsattr -El aio0 output to list the configuration of legacy AIO

```
[root@p520-tic-3]# lsattr -El aio0
autoconfig defined STATE to be configured at system restart True
fastpath   enable  State of fast path                       True
kprocprio  39      Server PRIORITY                          True
maxreqs    4096    Maximum number of REQUESTS               True
maxservers 10      MAXIMUM number of servers per cpu        True
minservers 1       MINIMUM number of servers                True
```

Notes: If you use AIX Version 6, there are no more Asynchronous I/O devices in the Object Data Manager (ODM), and the command **aioo** is removed. You must use the **ioo** command to change them.

If your AIX 5.3 is between TL05 and TL08, you can also use the **aioo** command to list and increase the values of maxservers, minservers, and maxreqs.

The rule is to monitor the I/O wait by using the **vmstat** command. If the I/O wait is more than 25%, consider enabling AIO, which reduces the I/O wait but does not help disks that are busy. You can monitor busy disks by using **iostat**, which we explain in the next section.

9.3.3 AIX iostat

The **iostat** command is used for monitoring system I/O device load. It can be used to determine and balance the I/O load between physical disks and adapters.

The **lsattr -E -l sys0 -a iostat** command option indicates whether the iostat statistic collection is enabled. To enable the collection of iostat data, use **chdev -l sys0 -a iostat=true**.

The disk and adapter-level system throughput can be observed by using the **iostat -aDR** command.

The option **a** retrieves the adapter-level details, and the option **D** retrieves the disk-level details. The option **R** resets the min* and max* values at each interval. See Example 9-12.

Example 9-13 Disk-level and adapter-level details by using iostat -aDR

```
[root@p520-tic-3]# iostat -aDR 1 1
System configuration: lcpu=2 drives=1 paths=1 vdisks=1 tapes=0

Vadapter:
vscsi0              xfer:     Kbps      tps   bkread     bkwrtn partition-id
                             29.7      3.6      2.8        0.8            0
                    read:      rps  avgserv  minserv  maxserv
                              0.0     48.2S     1.6     25.1
                   write:      wps  avgserv  minserv  maxserv
```

```
                   30402.8      0.0       2.1     52.8
         queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                     0.0      0.0      0.0      0.0        0.0        0.0


Paths/Disks:
hdisk0            xfer:  %tm_act      bps      tps      bread      bwrtn
                     1.4    30.4K      3.6      23.7K       6.7K
          read:      rps  avgserv  minserv  maxserv   timeouts      fails
                     2.8      5.7      1.6     25.1          0          0
         write:      wps  avgserv  minserv  maxserv   timeouts      fails
                     0.8      9.0      2.1     52.8          0          0
         queue:  avgtime  mintime  maxtime  avgwqsz    avgsqsz     sqfull
                    11.5      0.0     34.4      0.0        0.0        0.9
```

Check for the following situations when analyzing the output of **iostat**:

- Check whether the number of I/Os is balanced among the disks. If not, it might indicate that you have problems in the distribution of PPs over the LUNs. With the information provided by **lvmstat** or **filemon**, select the most active LV, and with the **lslv -m** command, check whether the PPs are distributed evenly among the disks of the VG. If not, check the inter-policy attribute on the LVs to see whether they are set to maximum. If the PPs are not distributed evenly and the (LV inter-policy attribute is set to minimum, you need to change the attribute to maximum and reorganize the VG.
- Check in the read section that the avgserv is larger than 15 ms. It might indicate that your bottleneck is in a lower layer, which can be the HBA, SAN, or even in the storage. Also, check whether the same problem occurs with other disks of the same VG. If yes, you need to add up the number of I/Os per second, add up the throughput by vpath (if it is the case), rank, and host, and compare with the performance numbers from Tivoli Storage Productivity Center for Disk.
- Check in the write section whether the avgserv is larger than 3 ms. Writes that average significantly and consistently higher indicate that write cache is full, and there is a bottleneck in the disk.
- Check in the queue section whether avgwqsz is larger than avgsqsz. Compare with other disks in the storage. Check whether the PPs are distributed evenly in all disks in the VG. If avgwqsz is smaller than avgsqsz, compare with other disks in the storage. If there are differences and the PPs are distributed evenly in the VG, it might indicate that the unbalanced situation is at the rank level.

The following example shows how multipath needs to be considered to interpret the **iostat** output.

In this example on Figure 9-8 on page 358, a server has two FC adapters and is zoned so that it uses four paths to the DS8000.

To determine the I/O statistics for the example in Figure 9-8 on page 358, you need to add up the iostats for hdisk1 - hdisk4. One way to establish relationships between the hdisks and the DS8000 LVs is to use the **pcmpath query device** command that is included with SDDPCM.

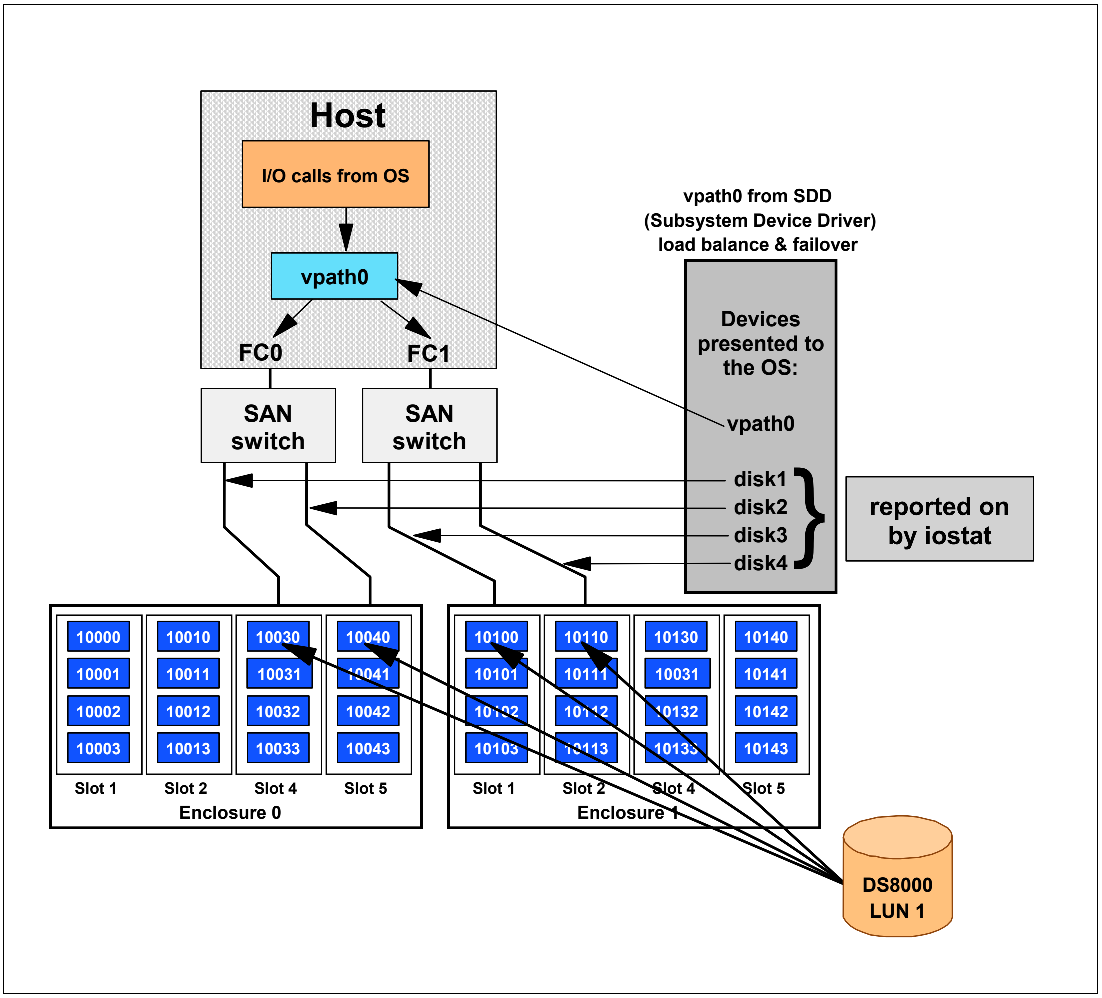


Figure 9-8 Devices presented to iostat

Another way is shown in Example 9-14. The command, **pcmpath query device 1**, lists the *paths* (hdisks). In this example, the logical disk on the DS8000 has LUN serial number 75065513000. The disk devices presented to the operating system are hdisk4, hdisk2, hdisk3, and hdisk4, so we can add up the iostats for these four hdisk devices.

Example 9-14 The pcmpath query device command

```
{CCF-part2:root}/ -> pcmpath query device 1

DEV#:   0  DEVICE NAME: hdisk1 TYPE: 2107900         POLICY:    Optimized
SERIAL: 75065513000
===========================================================================
Path#       Adapter/Hard Disk          State     Mode      Select      Errors
    0           fscsi0/hdisk4           OPEN    NORMAL        155           0
    1          fscsi0/hdisk12           OPEN    NORMAL        151           0
    2          fscsi1/hdisk20           OPEN    NORMAL        144           0
    3          fscsi1/hdisk28           OPEN    NORMAL        131           0
```

The option shown in Example 9-15 on page 359 provides details in a record format, which can be used to sum up the disk activity.

Example 9-15 Output of the iostat command (output truncated)

```
# iostat -alDRT 1 2

System configuration: lcpu=16 drives=5 paths=17 vdisks=22 tapes=0
Adapter:                    xfers               time
-------------------- --------------------------- ---------
                       bps    tps  bread  bwrtn
fcs0                 40.6M 8815.9   40.6M   0.0   15:10:03

Disks:                        xfers                              read                              write
queue                    time
---------- ------------------------------- ----------------------------------- ----------------------------------- ------------------
---------
                      %tm    bps   tps  bread  bwrtn   rps    avg    min    max time fail   wps    avg    min    max time fail    avg
min    max   avg   avg  serv
                      act                                      serv   serv   serv outs              serv   serv   serv outs        time
time   time  wqsz  sqsz qfull
hdisk2                9.9  11.1M 2406.4  11.1M   0.0  4643.8   0.1    0.1    0.4     0    0   0.0   0.0    0.0    0.0     0    0   0.0
0.0    0.0    0.0   0.0   0.0  15:10:03
hdisk1                5.9   8.2M 1778.5   8.2M   0.0  3665.3   0.1    0.1    8.5     0    0   0.0   0.0    0.0    0.0     0    0   0.0
0.0    0.0    0.0   0.0   0.0  15:10:03
hdisk3               12.9  11.0M 2387.3  11.0M   0.0  4889.2   0.1    0.1   10.6     0    0   0.0   0.0    0.0    0.0     0    0   0.0
0.0    0.0    0.0   0.0   0.0  15:10:03
hdisk4                5.9  10.3M 2243.8  10.3M   0.0  4519.5   0.1    0.1    9.6     0    0   0.0   0.0    0.0    0.0     0    0   0.0
0.0    0.0    0.0   0.0   0.0  15:10:03

Adapter:                    xfers               time
-------------------- --------------------------- ---------
                       bps    tps  bread  bwrtn
sissas1             561.4K 137.1    0.0  561.4K  15:10:03
```

It is not unusual to see a device reported by **iostat** as 90% - 100% busy, because a DS8000 volume that is spread across an array of multiple disks can sustain a much higher I/O rate than a single physical disk. A device that is 100% busy is generally a problem for a single device, but it is probably not a problem for a RAID 5 device.

Further Asynchronous I/O can be monitored through **iostat -A** for legacy AIO and **iostat -P** for Posix AIO.

Because the asynchronous I/O queues are assigned by filesystem, it is more interesting to measure the queues per filesystem. If you have several instances of the same application where each application uses a set of filesystems, you can see which instances consume more resources. Execute the **iostat -AQ** command to see the legacy AIO, which is shown in Example 9-16. Similarly for POSIX-compliant AIO statistics, use **iostat -PQ**.

Example 9-16 iostat -AQ output to measure legacy AIO activity by filesystem

```
[root@p520-tic-3]# iostat -AQ 1 2

System configuration: lcpu=4

aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait
        0    0    0    0 16384             0.0   0.1   99.9      0.0

Queue#         Count          Filesystems
129            0              /
130            0              /usr
132            0              /var
133            0              /tmp
136            0              /home
137            0              /proc
138            0              /opt

aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait
        0    0    0    0 16384             0.0   0.1   99.9      0.0
```

```
Queue#          Count           Filesystems
129             0               /
130             0               /usr
132             0               /var
133             0               /tmp
136             0               /home
137             0               /proc
138             0               /opt
```

If your AIX system is in a SAN environment, you might have so many hdisks that **iostat** does not provide much information. We suggest that you use **nmon**, which can report iostats based on vpaths or ranks, as discussed in “Interactive nmon options for DS8000 performance monitoring” on page 363.

For detailed information about the enhancements of the **iostat** tool in AIX Version 7, see 6.4 “Iostat command enhancement” in the *IBM AIX Version 7.1 Differences Guide*, SG24-7910:

http://www.redbooks.ibm.com/abstracts/sg247910.html

Or, see the **iostat** man pages:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.cmds%2Fdoc%2Faixcmds3%2Fiostat.htm

9.3.4 lvmstat

The **lvmstat** command reports input and output statistics for logical partitions, logical volumes, and volume groups. This command is useful in determining the I/O rates to LVM volume groups, logical volumes, and logical partitions. This command is useful for dealing with unbalanced I/O situations where the data layout was not considered initially.

Enabling volume group I/O by using lvmstat

By default, the statistics collection is not enabled. If the statistics collection is not enabled for the volume group or logical volume that you want to monitor, **lvmstat** reports this error:

```
#lvmstat -v rootvg
0516-1309 lvmstat:Statistics collection is not enabled for this logical device.
Use -e option to enable.
```

To enable statistics collection for all logical volumes in a volume group (in this case, the rootvg volume group), use the **-e** option together with the **-v *<volume group>*** flag as the following example shows:

```
#lvmstat -v rootvg -e
```

When you do not need to continue to collect statistics with **lvmstat**, disable it, because it affects the performance of the system. To disable the statistics collection for all logical volumes in a volume group (in this case, the rootvg volume group), use the **-d** option together with the **-v *<volume group>*** flag as the following example shows:

```
#lvmstat -v rootvg -d
```

This command disables the collection of statistics on all logical volumes in the volume group.

The first report section generated by **lvmstat** provides statistics that concern the time since the statistical collection was enabled. Each later report section covers the time since the previous report. All statistics are reported each time that **lvmstat** runs. The report consists of a header row, followed by a line of statistics for each logical partition or logical volume, depending on the flags specified.

Monitoring volume group I/O by using lvmstat

After a volume group is enabled for **lvmstat** monitoring, such as rootvg in this example, you need to run **lvmstat -v rootvg** only to monitor all activity to rootvg. An example of the **lvmstat** output is shown in Example 9-17.

Example 9-17 The lvmstat command example

```
#lvmstat -v rootvg
Logical Volume iocnt Kb_read Kb_wrtn Kbps
lv05           682478 16      8579672 16.08
loglv00        0      0       0       0.00
datalv         0      0       0       0.00
lv07           0      0       0       0.00
lv06           0      0       0       0.00
lv04           0      0       0       0.00
lv03           0      0       0       0.00
```

You can see that lv05 is busy performing writes.

The **lvmstat** tool has powerful options, such as reporting on a specific logical volume or reporting busy logical volumes in a volume group only. For additional information about usage, see the following links:

- *IBM AIX Version 7.1 Differences Guide*, SG24-7910-00:

  http://www.redbooks.ibm.com/abstracts/sg247910.html
- *AIX 5L Performance Tools Handbook*, SG24-6039

  http://www.redbooks.ibm.com/abstracts/sg246039.html?Open
- The man page of **lvmstat**:

  http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds3/lvmstat.htm&tocNode=int_215986

9.3.5 topas

The interactive AIX tool, **topas**, is convenient if you want to get a quick overall view of the current activity of the system. A fast snapshot of memory usage or user activity can be a helpful starting point for further investigation. Figure 9-9 on page 362 contains a sample **topas** output.

```
Topas Monitor for host:    p650-tic-1b          EVENTS/QUEUES    FILE/TTY
Tue Dec 18 11:26:06 2007   Interval:  2         Cswitch    2662  Readch     0.0G
                                                Syscall   71499  Writech    0.0G
Kernel    77.5   |######################     |  Reads     35583  Rawin         0
User      22.5   |#######                    |  Writes    35581  Ttyout      215
Wait       0.0   |                           |  Forks         0  Igets         0
Idle       0.0   |                           |  Execs         0  Namei       119
                                                Runqueue    3.0  Dirblk        0
Network  KBPS    I-Pack  O-Pack   KB-In  KB-Out Waitqueue   0.0
Total      0.4      2.0     1.0     0.1     0.3                  MEMORY
                                                PAGING           Real,MB    2560
Disk    Busy%      KBPS     TPS KB-Read KB-Writ Faults        0  % Comp     43.3
Total      0.0     41.1K 1000.5    32.0   41.1K Steals     4450  % Noncomp  55.5
                                                PgspIn        0  % Client   55.5
FileSystem         KBPS     TPS KB-Read KB-Writ PgspOut       0
Total           34.8K   69.6K   17.4K   17.4K   PageIn        0  PAGING SPACE
                                                PageOut   10041  Size,MB     512
Name            PID  CPU%  PgSp Owner           Sios      10037  % Used      1.1
blast        311460  77.8   1.1 root                             % Free     99.9
syncd         69724   7.3   0.5 root            NFS (calls/sec)
lrud          16392   5.1   0.1 root            SerV2         0  WPAR Activ    0
j2pg          86070   2.6   0.1 root            CliV2         0  WPAR Total    0
topas        335984   0.1   1.5 root            SerV3         0  Press: "h"-help
getty        249990   0.1   0.5 root            CliV3         0         "q"-quit
```

Figure 9-9 topas output in AIX 6.1

With AIX6.1, the **topas** monitor offers enhanced monitoring capabilities and now also provides I/O statistics for filesystems:

- Enter `ff` (first f turns it off, the next f expands it) to expand the filesystem I/O statistics.
- Type `F` to get an exclusive and even more detailed view of the filesystem I/O statistics.
- Expanded disk I/O statistics can be obtained by typing `dd` or `D` in the topas initial window.

For more details, see the **topas** manual page:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds5/topas.htm&tocNode=int_123659

9.3.6 nmon

The **nmon** tool and analyzer for AIX and Linux is a great storage performance analysis resource, and it is no charge. It was written by Nigel Griffiths who works for IBM in the UK. We use this tool, among others, when we perform client benchmarks. It is available at this website:

http://www.ibm.com/developerworks/aix/library/au-analyze_aix/

> **The nmon tool:** The **nmon** tool is not supported. No warranty is given or implied, and you cannot obtain help or maintenance from IBM.

The **nmon** tool currently is available in two versions to run on different levels of AIX:

- The **nmon** Version 12e for AIX 6.1 and later
- The **nmon** Version 9 for previous versions of AIX

The interactive **nmon** tool is similar to **monitor** or **topas**, which you perhaps used before to monitor AIX, but it offers more features that are useful for monitoring the DS8000 performance. We explore these interactive options.

Unlike **topas**, the **nmon** tool can also record data that can be used to establish a baseline of performance for comparison later. Recorded data can be saved in a file and imported into the **nmon** analyzer (a spreadsheet format) for easy analysis and graphing.

Interactive nmon options for DS8000 performance monitoring

The interactive **nmon** tool is an excellent way to show comprehensive AIX system monitoring information, of your choice, on one display. When used interactively, **nmon** updates statistics every 2 seconds. You can change the refresh rate. To run the tool, type **nmon** and press Enter. Then, press the keys corresponding to the areas of interest. For this book, we are interested in monitoring storage. The options that relate to storage are a, d, D, and e. For example, type **nmon** to start the tool, then select A (adapter), then D (disk I/O graphs or disk statistics, but not both at the same time), and then E (DS8000 vpath statistics). It is also helpful to view only the busiest disk devices, so type a period (.) to turn on this viewing feature. We also want to look at C (CPU utilization) and sometimes T (top processes). The **nmon** tool has its own way of ordering the topics that you choose on the display.

The different options you can select when you run **nmon** Version 12 are shown in Example 9-18.

Example 9-18 The nmon tool options

```
lqHELPqqqqqqqqqmost-keys-toggle-on/offqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
xh = Help information     q = Quit nmon            0 = reset peak counts      x
x+ = double refresh time  - = half refresh         r = ResourcesCPU/HW/MHz/AIXx
xc = CPU by processor     C=upto 128 CPUs          p = LPAR Stats (if LPAR)   x
xl = CPU avg longer term  k = Kernel Internal      # = PhysicalCPU if SPLPAR  x
xm = Memory & Paging      M = Multiple Page Sizes   P = Paging Space          x
xd = DiskI/O Graphs       D = DiskIO +Service times o = Disks %Busy Map       x
xa = Disk Adapter         e = ESS vpath stats      V = Volume Group stats     x
x^ = FC Adapter (fcstat)  O = VIOS SEA (entstat)   v = Verbose=OK/Warn/Danger x
xn = Network stats        N=NFS stats (NN for v4)  j = JFS Usage stats        x
xA = Async I/O Servers    w = see AIX wait procs   "="= Net/Disk KB<-->MB     x
xb = black&white mode     g = User-Defined-Disk-Groups (see cmdline -g)       x
xt = Top-Process --->     1=basic 2=CPU-Use 3=CPU(default) 4=Size 5=Disk-I/O  x
xu = Top+cmd arguments    U = Top+WLM Classes      . = only busy disks & procsx
xW = WLM Section          S = WLM SubClasses)                                 x
xNeed more details?  Then stop nmon and use: nmon -?                          x
xOr try http://www.ibm.com/collaboration/wiki/display/WikiPtype/nmon          x
x                                                                             x
xnmon version 12e build=5300-06 - written by Nigel Griffiths, nag@uk.ibm.com  x
x                                                                             x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
```

The nmon adapter performance

Example 9-19 displays the ability of **nmon** to show I/O performance based on system adapters and FC statistics. The output shows one SCSI controller and two FC adapters. The real advantage of these FC statistics is that **nmon** shows actual FC statistics, which are useful to measure the throughput of a tape drive attached to the FC adapter.

Example 9-19 The nmon tool adapter statistics

```
lqnmon12eqqqqqS=WLMsubclassesqqqqHost=p520-tic-3qqqqqRefresh=1 secsqqq12:57.11qk
x Fibre-Channel-Adapter qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqx
x       Adapter  Receive Transmit        -Requests-       --Size KB--         x
xNumber Name      KB/s    KB/s           In     Out        In       Out       x
x    1      fcs0     0.0     0.0          0.0     0.0       0.0      0.0      x
x    2      fcs1     0.0     0.0          0.0     0.0       0.0      0.0      x
x         Totals     0.0     0.0 MB/s     0.0     0.0                         x
```

```
x Disk-Adapter-I/O qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqx
xName          %busy     read    write        xfers Disks Adapter-Type        x
xscsi0           0.0      0.0      0.0 KB/s     0.0 2     PCI-X Dual Channel Ulx
xide0            0.0      0.0      0.0 KB/s     0.0 1     ATA/IDE Controller Dex
xsisscsia0       9.8      0.0    109.9 KB/s    26.5 2     PCI-X Dual Channel Ulx
xfcs0            0.0      0.0      0.0 KB/s     0.0 4     FC Adapter          x
xTOTALS  4 adapters       0.0    109.9 KB/s    26.5 9     TOTAL(MB/s)=0.1     x
xqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqx
```

The nmon tool vpath performance

The **e** option of **nmon** shows I/O activity based on vpaths as shown in Example 9-20.

Example 9-20 The nmon tool vpath option

```
¦Name     Size(GB) AvgBusy  read-KB/s write-KB/s  TotalMB/s xfers/s vpaths=2   ¦
¦vpath0       10.0  22.9%    1659.2     796.3        2.4      4911.0           ¦
¦vpath1       10.0  23.5%    1673.0     765.5        2.4      4877.0           ¦
¦TOTALS             23.2%    3332.2    1561.8        4.8      9788.1           ¦
+------------------------------------------------------------------------------+
```

The nmon tool disk group performance

The **nmon** Version 10 tool has a feature called *disk grouping*. For example, you can create a disk group based on your AIX volume groups. First, you need to create a file that maps hdisks to nicknames. For example, you can create a map file, as shown in Example 9-21.

Example 9-21 The nmon tool disk group mapping file

```
vi /tmp/vg-maps

rootvg hdisk0 hdisk1
6000vg hdisk2 hdisk3 hdisk4 hdisk5 hdisk6 hdisk7 hdisk8 hdisk9 hdisk10 hdisk11 hdisk1
2 hdisk13 hdisk14 hdisk15 hdisk16 hdisk17
8000vg hdisk26 hdisk27 hdisk28 hdisk29 hdisk30 hdisk31 hdisk32 hdisk33
```

Then, type **nmon** with the **-g** flag to point to the map file:

```
nmon -g /tmp/vg-maps
```

When **nmon** starts, press the G key to view statistics for your disk groups. An example of the output is shown in Example 9-22.

Example 9-22 The nmon tool disk-group output

```
--nmon-v10p---N=NFS--------------Host=san5198b-------Refresh=1 secs---14:02.10-----
  Disk-Group-I/O
 Name          Disks AvgBusy Read|Write-KB/s  TotalMB/s   xfers/s   R:W-SizeKB
 rootvg            2   0.0%       0.0|0.0          0.0       0.0    0.0
 6000vg           16  45.4%     882.6|93800.1     92.5    2131.5   44.4
 8000vg            8  95.3%    1108.7|118592.0   116.9    2680.7   44.7
 Groups= 3 TOTALS  26   5.4%    1991.3|212392.2   209.4    4812.2
```

The **nmon** tool has these characteristics:

- The **nmon** tool reports real-time iostats for different disk groups.
- In this case, the disk groups that we created are for volume groups.
- You can create logical groupings of hdisks for any group that you want.
- You can make multiple disk-group map files and start **nmon -g *<map-file>*** to report on different groups.

To enable **nmon** to report iostats based on ranks, you can make a disk-group map file that lists ranks with the associated hdisk members.

Use the SDDPCM command **pcmpath query device** to provide a view of your host system logical configuration on the DS8000. You can, for example, create a **nmon** disk group of storage type (DS8000), logical subsystem (LSS), rank, and port to show you unique views into your storage performance.

Recording nmon information for import into the nmon analyzer tool

A *great* benefit that the **nmon** tool provides is the ability to collect data over time to a file and then to import the file into the *nmon analyzer* tool:

http://www.ibm.com/developerworks/aix/library/au-nmon_analyser/

To collect **nmon** data in comma-separated value (csv) file format for easy spreadsheet import:

1. Run **nmon** with the **-f** flag. See **nmon -h** for the details, but as an example, to run **nmon** for an hour to capture data snapshots every 30 seconds, use this command:

   ```
   nmon -f -s 30 -c 120
   ```

2. This command creates the output file in the current directory called `<hostname>_date_time.nmon`.

The *nmon analyzer* is a macro-customized Microsoft Excel spreadsheet. After transferring the output file to the machine that runs the **nmon** analyzer, simply start the **nmon** analyzer, enabling the macros, and click **Analyze nmon data**. You are prompted to select your spreadsheet and then to save the results.

Many spreadsheets have fixed numbers of columns and rows. We suggest that you collect up to a maximum of 300 snapshots to avoid experiencing these issues.

> **Tip:** The use of the CHARTS setting instead of PICTURES for graph output simplifies the analysis of the data, which makes it more flexible.

When you capture data to a file, the **nmon** tool disconnects from the shell to ensure that it continues running even if you log out, which means that **nmon** can appear to fail, but it is still running in the background until the end of the analysis period.

9.3.7 fcstat

The **fcstat** command displays statistics from an specific FC adapter. Example 9-23 shows the output of the **fcstat** command.

Example 9-23 The fcstat command output

```
# fcstat fcs0
FIBRE CHANNEL STATISTICS REPORT: fcs0
skipping.........
FC SCSI Adapter Driver Information
  No DMA Resource Count: 0
  No Adapter Elements Count: 0
  No Command Resource Count: 99023
skipping.........
```

The "No Command Resource Count" indicates how many times the *num_cmd_elems* value was exceeded since AIX was booted. You can continue to take snapshots every 3 - 5 minutes during a peak period to evaluate whether you need to increase the value of *num_cmd_elems*. For additional information, see the man pages of the **fcstat** command:

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds2/fcstat.htm&tocNode=int_122561

9.3.8 filemon

The **filemon** command monitors a trace of filesystem and I/O system events, and reports performance statistics for files, virtual memory segments, logical volumes, and physical volumes. The **filemon** command is useful to individuals whose applications are thought to be disk-bound, and who want to know where and why.

The **filemon** command provides a quick test to determine whether there is an I/O problem by measuring the I/O service times for reads and writes at the disk and logical volume level.

The **filemon** command resides in `/usr/bin` and is part of the `bos.perf.tools` file set, which can be installed from the AIX base installation media.

filemon measurements

To provide a complete understanding of filesystem performance for an application, the **filemon** command monitors file and I/O activity at four levels:

- Logical filesystem

  The **filemon** command monitors logical I/O operations on logical files. The monitored operations include all read, write, open, and seek system calls, which might result in actual physical I/O, depending on whether the files are already buffered in memory. I/O statistics are kept on a per-file basis.

- Virtual memory system

  The **filemon** command monitors physical I/O operations (that is, paging) between segments and their images on disk. I/O statistics are kept on a per segment basis.

- Logical volumes

  The **filemon** command monitors I/O operations on logical volumes. I/O statistics are kept on a per-logical volume basis.

- Physical volumes

  The **filemon** command monitors I/O operations on physical volumes. At this level, physical resource utilizations are obtained. I/O statistics are kept on a per-physical volume basis.

filemon examples

A simple way to use **filemon** is to run the command that is shown in Example 9-24, which performs these actions:

- Run **filemon** for 2 minutes and stop the trace.
- Store output in `/tmp/fmon.out`.
- Collect only logical volume and physical volume output.

Example 9-24 The filemon command

```
#filemon -o /tmp/fmon.out -T 500000 -PuvO lv,pv; sleep 120; trcstop
```

Tip: To set the size of the buffer of option **-T**, in general, start with 2 MB per logical CPU.

For additional information about **filemon**, check the man pages of **filemon**:

http://publib.boulder.ibm.com/infocenter/systems/scope/aix/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds2/filemon.htm&tocNode=int_215661

To produce sample output for **filemon**, we ran a sequential write test in the background, and started a **filemon** trace, as shown in Example 9-25. We used the **lmktemp** command to create a 2 GB file full of nulls while **filemon** gathered I/O statistics.

Example 9-25 The filemon command with a sequential write test

```
cd /interdiskfs
time lmktemp 2GBtest 2000M &
filemon -o /tmp/fmon.out -T 500000 -PuvO lv,pv; sleep 120; trcstop
```

In Example 9-26 on page 368, we look at parts of the `/tmp/fmon.out` file. When analyzing the output from **filemon**, focus on these areas:

- Most active physical volume:
  - Look for balanced I/O across disks.
  - Lack of balance might be a data layout problem.
- Look at I/O service times at the physical volume layer:
  - Writes to cache that average less than 3 ms are good. Writes averaging significantly and consistently longer times indicate that write cache is full, and there is a bottleneck in the disk.
  - Reads that average less than 10 ms - 20 ms are good. The disk subsystem read cache hit rate affects this value considerably. Higher read cache hit rates result in lower I/O service times, often near 5 ms or less. If reads average greater than 15 ms, it can indicate that something between the host and the disk is a bottleneck, although it indicates a bottleneck in the disk subsystem.
  - Look for consistent I/O service times across physical volumes. Inconsistent I/O service times can indicate unbalanced I/O or a data layout problem.
  - Longer I/O service times can be expected for I/Os that average greater than 64 KB in size.
  - Look at the difference between the I/O service times between the logical volume and the physical volume layers. A significant difference indicates queuing or serialization in the AIX I/O stack.

The following fields show in the filemon report of the **filemon** command:

util Utilization of the volume (fraction of time busy). The rows are sorted by this field, in decreasing order. The first number, 1.00, means 100 percent.

#rblk Number of 512-byte blocks read from the volume.

#wblk Number of 512-byte blocks written to the volume.

KB/sec Total transfer throughput in kilobytes per second.

volume Name of volume.

description Contents of the volume: either a filesystem name or a logical volume type (jfs2, paging, jfslog, jfs2log, boot, or sysdump). Also indicates whether the filesystem is fragmented or compressed.

Example 9-26 The filemon most active logical volumes report

```
Thu Oct  6 21:59:52 2005
System: AIX CCF-part2 Node: 5 Machine: 00E033C44C00

Cpu utilization:  73.5%

Most Active Logical Volumes
------------------------------------------------------------------------
  util  #rblk  #wblk   KB/s  volume                  description
------------------------------------------------------------------------
  0.73      0 20902656 86706.2  /dev/305glv             /interdiskfs
  0.00      0    472    2.0  /dev/hd8                jfs2log
  0.00      0     32    0.1  /dev/hd9var             /var
  0.00      0     16    0.1  /dev/hd4                /
  0.00      0    104    0.4  /dev/jfs2log01          jfs2log

Most Active Physical Volumes
------------------------------------------------------------------------
  util  #rblk  #wblk   KB/s  volume                  description
------------------------------------------------------------------------
  0.99      0 605952 2513.5  /dev/hdisk39            IBM FC 2107
  0.99      0 704512 2922.4  /dev/hdisk55            IBM FC 2107
  0.99      0 614144 2547.5  /dev/hdisk47            IBM FC 2107
  0.99      0 684032 2837.4  /dev/hdisk63            IBM FC 2107
  0.99      0 624640 2591.1  /dev/hdisk46            IBM FC 2107
  0.99      0 728064 3020.1  /dev/hdisk54            IBM FC 2107
  0.98      0 612608 2541.2  /dev/hdisk38            IBM FC 2107

skipping...........

------------------------------------------------------------------------
Detailed Logical Volume Stats   (512 byte blocks)
------------------------------------------------------------------------

VOLUME: /dev/305glv  description: /interdiskfs
writes:                 81651   (0 errs)
  write sizes (blks):   avg   256.0 min     256 max     256 sdev     0.0
  write times (msec):   avg   1.816 min   1.501 max   2.409 sdev   0.276
  write sequences:      6
  write seq. lengths:   avg 3483776.0 min  423936 max 4095744 sdev 1368402.0
seeks:                  6       (0.0%)
  seek dist (blks):     init  78592,
                        avg 4095744.0 min 4095744 max 4095744 sdev     0.0
time to next req(msec): avg   1.476 min   0.843 max 13398.588 sdev  56.493
throughput:             86706.2 KB/sec
utilization:            0.73

skipping...........
------------------------------------------------------------------------
Detailed Physical Volume Stats   (512 byte blocks)
------------------------------------------------------------------------

VOLUME: /dev/hdisk39  description: IBM FC 2107
writes:                 2367    (0 errs)
  write sizes (blks):   avg   256.0 min     256 max     256 sdev     0.0
  write times (msec):   avg   1.934 min   0.002 max   2.374 sdev   0.524
  write sequences:      2361
  write seq. lengths:   avg   256.7 min     256 max     512 sdev    12.9
seeks:                  2361    (99.7%)
```

```
  seek dist (blks):     init 14251264,
                        avg  1928.4 min     256 max  511232 sdev 23445.5
  seek dist (%tot blks):init 10.61802,
                        avg 0.00144 min 0.00019 max 0.38090 sdev 0.01747
time to next req(msec): avg  50.666 min   1.843 max 14010.230 sdev 393.436
throughput:             2513.5 KB/sec
utilization:            0.99

VOLUME: /dev/hdisk55  description: IBM FC 2107
writes:                 2752    (0 errs)
  write sizes (blks):   avg   256.0 min     256 max     256 sdev     0.0
  write times (msec):   avg   1.473 min   0.507 max   1.753 sdev   0.227
  write sequences:      2575
  write seq. lengths:   avg   273.6 min     256 max     512 sdev    64.8
seeks:                  2575    (93.6%)
  seek dist (blks):     init 14252544,
                        avg  1725.9 min     256 max  511232 sdev 22428.8
  seek dist (%tot blks):init 10.61897,
                        avg 0.00129 min 0.00019 max 0.38090 sdev 0.01671
time to next req(msec): avg  43.573 min   0.844 max 14016.443 sdev 365.314
throughput:             2922.4 KB/sec
utilization:            0.99

skipping to end....................
```

In the **filemon** output in Example 9-26 on page 368, we notice these characteristics:

- The most active logical volume is /dev/305glv (/interdiskfs); it is the busiest logical volume with an average data rate of 87 MBps.
- The Detailed Logical Volume Status field shows an average write time of 1.816 ms for /dev/305glv.
- The Detailed Physical Volume Stats show an average write time of 1.934 ms for the busiest disk, /dev/hdisk39, and 1.473 ms for /dev/hdisk55, which is the next busiest disk.

The **filemon** command is a useful tool to determine where a host spends I/O. More details about the **filemon** options and reports are available in the publication *AIX 5L Performance Tools Handbook*, SG24-6039:

http://www.redbooks.ibm.com/abstracts/sg246039.html?Open

9.4 Oracle Solaris disk I/O components

Oracle Solaris 11 has several performance enhancements and new features. The following sections give configuration advice according to a typical environment, the type of application, and the setup of the volume layout. See the following link for the information:

http://www.oracle.com/technetwork/server-storage/solaris11/documentation/solaris-express-whatsnew-201011-175308.pdf

9.4.1 UFS

UFS is the standard filesystem of Solaris. You can configure a journaling feature, adjust cache filesystem parameters, implement Direct I/O, and adjust the mechanism of sequential read ahead.

For more information about UFS, see the following links:

- Introduction to the Solaris filesystem:

  http://www.solarisinternals.com/si/reading/sunworldonline/swol-05-1999/swol-05-filesystem.html

  http://www.solarisinternals.com/si/reading/fs2/fs2.html

  http://www.solarisinternals.com/si/reading/sunworldonline/swol-07-1999/swol-07-filesystem3.html

- You can obtain the *Oracle Solaris Tunable Parameters Reference Manual* at this website:

  http://docs.oracle.com/cd/E23824_01/html/821-1450/index.html

- For more information about Oracle Solaris commands and tuning options, see the following website:

  http://www.solarisinternals.com

9.4.2 Symantec Veritas File System (VxFS) for Solaris

The Veritas File System supports 32-bit and 64-bit kernels and is a journaled filesystem. Its method of file organization is based on a hash algorithm. It implements Direct I/O and Discovered Direct I/O.

With the Veritas File System, you can adjust the mechanisms of *sequential read ahead* and *sequential and random write behind*. You can tune its buffers to increase performance, and it supports asynchronous I/O.

Filesystem blocksize

The smallest allocation unit of a filesystem is the blocksize. In VxFS, you can choose from 512 bytes to 8192 bytes. To decide which size is best for your application, consider the average size of the application files. If the application is a file server and the average size is about 1 KB, choose a blocksize of 1 KB. But if the application is a database with a few large files, choose the maximum size of 8 KB. The default blocksize is 2 KB. In addition, when creating and allocating file space inside of VxFS and by using standard tools, such as **mkfile** (Solaris only) or database commands, you might see performance degradations. For additional information, see these websites:

http://www.symantec.com/business/support/index?page=answerlink&url=http%3A%2F%2Fwww.symantec.com%2Fbusiness%2Fsupport%2Fresources%2Fsites%2FBUSINESS%2Fcontent%2Fstaging%2FTECHNICAL_SOLUTION%2F10000%2FTECH10174%2Fen_US%2F2.0%2Ffsed_sag_sol_50_235203.pdf&answerid=16777220&searchid=1322121458096

http://www.symantec.com/business/support/resources/sites/BUSINESS/content/staging/TECHNICAL_SOLUTION/61000/TECH61155/en_US/3.0/305161.doc

Direct I/O and Discovered Direct I/O

Direct I/O works in a similar manner to AIX Direct I/O and Solaris UFS Direct I/O. It is a way to execute an I/O request by bypassing the filesystem cache. A Discovered Direct I/O is similar to a Direct I/O request, but it does not need a synchronous commit of the inode when the file is extended or blocks are allocated. Whenever the filesystem receives an I/O request larger than the attribute discovered_direct_iosz, it tries to use Direct I/O on the request. The default value is 256 KB.

Quick I/O

Quick I/O is a licensed feature from Storage Foundation for Oracle, DB2, or Sybase databases. However, the binaries come in the VRTSvxfs package. It allows a database access to pre-allocated VxFS files as raw character devices. It combines the performance of a raw device with the convenience to manipulate files that a filesystem can provide. There is an interesting document that describes the use of Quick I/O compared to other I/O access methods, such as Direct I/O:

http://eval.veritas.com/webfiles/docs/qiowp.pdf

Read Ahead and Write Behind

VxFS uses the following parameters to adjust the configuration of Read Ahead and Write Behind:

- read_pref_io: It sets the preferred read request size. The default is 64 KB.
- write_pref_io: It sets the preferred write request size. The default is 64 KB.
- read_nstream: It sets the number of parallel read requests at one time. The default is 1.
- write_nstream: It sets the number of parallel write requests at one time. The default is 1.

For additional usage information, see the following links:

- *Veritas File System 5.0 - Administrator's Guide for Solaris:*

  http://sfdoccentral.symantec.com/sf/5.1/solaris/pdf/vxfs_admin.pdf

- *Veritas File System 5.0 - Administrator's Guide for AIX:*

  http://sfdoccentral.symantec.com/sf/5.1/aix/pdf/vxfs_admin.pdf

- *Veritas File System 5.0 - Administrator's Guide for HP-UX:*

  http://sfdoccentral.symantec.com/sf/5.0.1/hpux/pdf/vxfs_admin.pdf

- *Veritas File System 5.0 - Administrator's Guide for Linux:*

  http://sfdoccentral.symantec.com/sf/5.1/linux/pdf/vxfs_admin.pdf

- In addition, there is an IBM Redbooks publication about VERITAS Storage Foundation Suite called *Introducing VERITAS Foundation Suite for AIX*, SG24-6619:

  http://www.redbooks.ibm.com/abstracts/sg246619.html?Open

9.4.3 Oracle Solaris ZFS

ZFS is the third generation of UNIX Filesystem, developed by SUN. It was introduced in Solaris 10. It is an entirely redesigned filesystem. It is a 128-bit filesystem and does not need a Volume Manager. It is a transactional filesystem, and it implements dynamic striping, multiple blocksizes, and intelligent prefetch, among other features.

For the latest updates in ZFS and the administrator guide, see this link:

http://docs.oracle.com/cd/E19963-01/html/821-1448/gbscy.html

Storage pools

ZFS implements built-in features of volume management. You define a storage pool and add disks for that storage pool. It is not necessary to partition the disks and create filesystems on top of those partitions. Instead, you simply define the filesystems, and ZFS allocates disk space dynamically. ZFS abstracts the real memory through a virtual memory address space, similar to virtual memory. You also can implement quotas and reserve space for a specific filesystem.

> **Support:** ZFS is not supported with SDD, only with MPxIO.

Data Integrity Model

ZFS implements its transactional model. In case a system crashes, there is no need to run an **fsck**. It also continues to check the integrity of each data block every time that it executes an I/O read. If it detects any error in a mirrored storage pool, ZFS automatically reads the data from a mirror and fixes the corrupted data. There are five options to configure the checksum:

- on: This option enables checksum and uses the fletcher2 method of checksumming. This option is the default.
- off: This option disables checksum. We do not advise that you disable checksumming.
- fletcher2: This option is a 256-bit hash method used for checksumming. This option is the fastest and simplest checksum option available in ZFS.
- fletcher4: This option is another 256-bit hash method for checksumming. It is stronger than fletcher2, but it still is not a cryptographic hash algorithm.
- sha256: This option is a 256-bit cryptographically strong hash method used for checksumming. This option is the highest level of checksumming available in ZFS.

For detailed information about how to check and configure the checksumming method, see this link:

http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide#Checksums

> **Important:** Every time that ZFS issues an I/O write to the disk, it does not know whether the storage has a nonvolatile random access memory (NVRAM). Therefore, it requests to flush the data from cache to disk in the storage. If your Solaris Release is 10 or later, you can disable the flush by setting the following parameter in the `/etc/system` file: `set zfs:zfs_nocacheflush = 1`. Additional details are provided at the following link:
>
> http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide#FLUSH

Dynamic striping

With the storage pool model, after you add more disks to the storage pool, ZFS automatically redistributes the data among the disks.

Multiple blocksize

In ZFS, there is no need to define a blocksize for each filesystem. ZFS tries to match the blocksize with the application I/O size. However, if your application is a database, we suggest that you enforce the blocksize to match the database blocksize. The parameter is recordsize and can range from 512 bytes to 128 KB. For example, if you want to configure a blocksize of 8 KB, you type the command **zfs set recsize=16384 *<userpool name>*** or ***<filesystem>***.

Cache management

Cache management is implemented by a modified version of an Adaptive Replacement Cache (ARC) algorithm. By default, it tries to use the real memory of the system while the utilization increases and there is free memory in the system. Therefore, if your application also uses a large amount of memory, such as a database, you might need to limit the amount of memory available for the ZFS ARC. For detailed information and instructions about how to limit the ZFS ARC, see the following link:

http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide#ARCSIZE

For more information about ZFS, see the following links:

- A website with ZFS documentation and a link to a guide about ZFS best practices:

  http://hub.opensolaris.org/bin/view/Community+Group+zfs/docs

- The *ZFS Evil Tuning Guide* is a website with the latest recommendations about how to tune ZFS:

  http://www.solarisinternals.com/wiki/index.php/ZFS_Evil_Tuning_Guide

- A blog with performance recommendations for tuning ZFS with a database:

  http://blogs.oracle.com/realneel/entry/zfs_and_databases

- *The Solaris ZFS Administrator's Guide*:

  http://docs.oracle.com/cd/E19253-01/819-5461/index.html

9.4.4 Solaris Volume Manager (formerly Solstice DiskSuite)

Solaris Volume Manager (SVM) is an LVM developed by SUN. Since Solaris 9, it is bundled with the operating system. It comes with a new feature called *soft partition* and you can implement RAID 0, RAID 1, and RAID 5.

For detailed information, see the following links:

- Performance best practices with SVM:

  http://docs.oracle.com/cd/E19082-01/819-2789/best-practices-1/index.html

- *The Solaris Volume Manager Administration Guide - Solaris 10*:

  http://docs.oracle.com/cd/E19253-01/816-4520/

9.4.5 Veritas Volume Manager (VxVM) for Solaris

The Symantec Veritas Volume Manager (VxVM) for Solaris is a third-party LVM that was developed by Veritas. It supports RAID 0, RAID 1, and RAID 5, and you can spread data over subdisks that make up *plexes* (you can create over 32 plexes). Figure 9-10 on page 374 gives an overview.

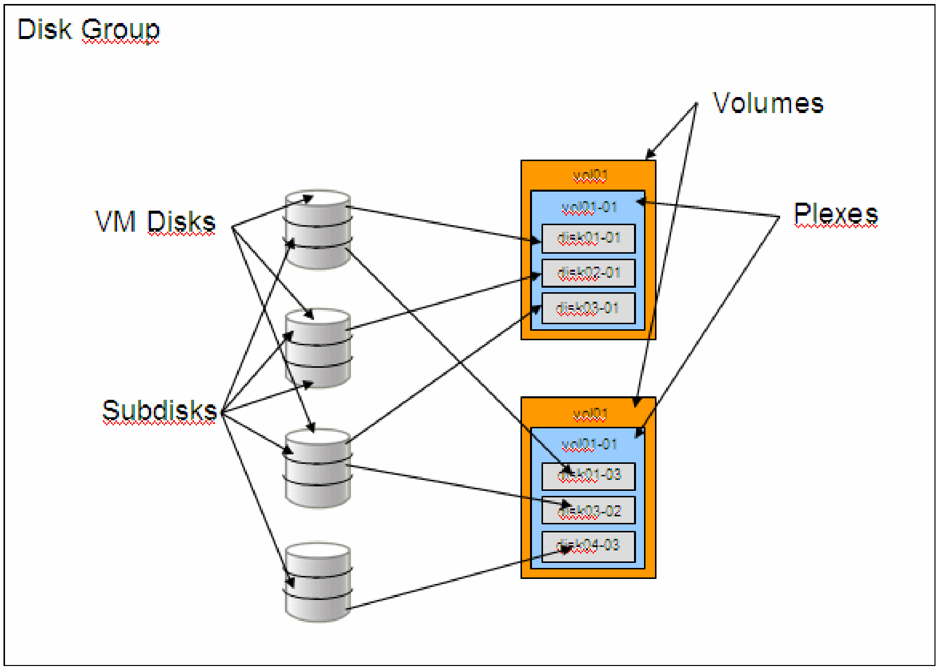


Figure 9-10 Veritas Volume Manager (VxVM) overview

The DS8000 LUNs that are under the control of VxVM are called *VM Disks*. You can split those VM Disks in smaller pieces that are called *subdisks*. A plex looks like a mirror and consists of a set of subdisks. It is at the plex level that you can configure RAID 0, RAID 5, or simply concatenate the subdisks. A volume consists of one or more plexes. When you add more than one plex to a volume, you can implement RAID 1. On top of that volume, you can create a filesystem or simply use it as a raw device.

To set up the volume layout with the DS8000 LUNs, you can adopt one of the following strategies:

- Storage pool striping: In this case, you spread the workload at the storage level. At the operating system level, you need to create the plexes with the layout attribute set to concat, which is the default option when creating a plex.
- Striped Plex: A set of LUNs is created in different ranks inside of the DS8000. After the LUNs are recognized in Solaris, a Disk Group (DG) is created, the plexes are spread evenly over the LUNs, and the stripe size of a plex is set from 8 MB to 16 MB.

RAID considerations

When using VxVM with the DS8000 LUNs, spread the workload over the several DS8000 LUNs by creating RAID 0 plexes. The stripe size is based on the I/O size of your application. If your application has I/O sizes of 1 MB, define the stripe sizes as 1 MB. If your application performs many sequential I/Os, it is better to configure stripe sizes of 4 MB or more to take advantage of the DS8000 prefetch algorithm. See Chapter 9, “Performance considerations for UNIX servers” on page 327 for details about RAID configuration.

vxio:vol_maxio

When you use VxVM on the DS8000 LUNs, you must set the VxVM maximum I/O size parameter (vol_maxio) to match the I/O size of your application or the stripe size of VxVM RAID 0. If the I/O size of your application is 1 MB and you use the Veritas Volume Manager on your DS8000 LUNs, edit the `/etc/system` and add the entry `set vxio:vol_maxio=2048`. The value is in blocks of 512 bytes.

For detailed information about how to configure and use VxVM in several platforms, see the links in “Read Ahead and Write Behind” on page 371.

9.4.6 IBM Subsystem Device Driver for Oracle Solaris

IBM Subsystem Device Driver for Solaris is the IBM multipathing device driver that is available for Oracle Solaris machines. See the following link for details:

http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303

9.4.7 MPxIO

MPxIO is the multipathing device driver that comes with Oracle Solaris and is required when implementing Oracle Solaris Clusters. For additional information, see the following links:

- A presentation providing an overview of MPxIO:

  http://opensolaris.org/os/project/mpxio/files/mpxio_toi_sio.pdf

- *The Solaris SAN Configuration and Multipathing Guide*:

  http://docs.oracle.com/cd/E19253-01/820-1931/820-1931.pdf

9.4.8 Symantec Veritas Dynamic MultiPathing (DMP) for Oracle Solaris

Veritas Dynamic MultiPathing (DMP) for Solaris is a device driver provided by Symantec to work with VxVM for Solaris.

General performance suggestions or tips for multipathing software:

- For SDD, check the compatibility matrix to download the correct version with Solaris:

  http://www.ibm.com/support/docview.wss?rs=540&uid=ssg1S7001350#SolarisSDD

- When configuring the DS8000 with MPxIO, follow the instructions in the *Host System Attachment Guide,* SG26-7628:

  http://www.ibm.com/support/docview.wss?uid=ssg1S7001161&aid=1

- Configuring and tuning DMP paper:

  http://www.symantec.com/content/en/us/enterprise/media/stn/pdfs/Articles/dynamic_multi_pathing.pdf

9.4.9 Array Support Library (ASL)

The ASL is a software package that is used by Device Discovery Layer, which is a component of VxVM to support third-party arrays.

For additional information, check the following links:

- There is a technical note about ASL that provides additional information:

  http://www.symantec.com/business/support/index?page=content&id=TECH21351

- Check the compatibility matrix in the Symantec website to find the right package:

  http://www.symantec.com/business/products/otherresources.jsp?pcid=pcat_storage&pvid=203_1

- Check the following guide for a list of the updates that Oracle Solaris needs for attachment to the DS8000: *Host System Attachment Guide,* GC27-2298-02. Download this guide from this website:

  http://www.ibm.com/support/docview.wss?uid=ssg1S7001161&aid=1

- Adjust the maximum I/O size for each device driver if you must work with I/Os larger than 1 MB. For example, perform these tasks to change to an I/O size of 4 MB:
  - sd (SCSI Device Driver): Edit the file `/kernel/drv/sd.conf` and add a line with `sd_max_xfer_size=0x400000`
  - ssd (Fibre Channel Device Driver): Edit the file `/kernel/drv/ssd.conf` and add a line with `ssd_max_xfer_size=0x400000`
  - st (Tape Device Driver): Edit the file `/kernel/drv/st.conf` and add a line with `st_max_xfer_size=0x400000`

> **Important:** For Solaris 10 SPARC, the sd and ssd transfer size setting defaults to `maxphys`. Certain Fibre Channel HBAs do not support requests greater than 8 MB. Do not forget to test the new values before you put them in production.

9.4.10 FC adapter

AMCC (formerly JNI), Emulex, QLogic, and SUN FC adapters are described in the DS8000 *Host System Attachment Guide,* GC27-2298-02, with suggested performance parameters. For more information, see the following link:

http://publib.boulder.ibm.com/infocenter/dsichelp/ds8000ic/index.jsp?topic=/com.ibm.storage.ssic.help.doc/f2c_agrs62105inst_1atzyy.html

9.5 Oracle Solaris performance monitoring tools

You can use the following tools or commands.

9.5.1 fcachestat and directiostat

Two useful tools to measure performance at the filesystem level are **`fcachestat`** and **`directiostat`**.

fcachestat

The **`fcachestat`** command and its output are illustrated in Example 9-27.

Example 9-27 The fcachestat command output

```
[root@v480-1]# fcachestat 1
  --- dnlc ---     -- inode ---     -- ufsbuf --     -- segmap --     -- segvn ---
  %hit   total     %hit   total     %hit   total     %hit   total     %hit   total
 99.68  145.4M    17.06  512720    99.95   17.6M    42.05   11.0M    80.97    9.7M
100.00       1     0.00       0    99.99    8653     0.00    4288    50.01    4289
100.00       8     0.00       0    99.99    7747     0.00    3840    50.00    3840
100.00       1     0.00       0    99.99    9653     0.00    4780    50.92    4694
100.00       1     0.00       0    99.99    8533     0.00    4215    49.15    4279
100.00       1     0.00       0    99.99    8651     0.00    4289    51.18    4195
100.00       4     0.00       0    99.98    9358     0.00    4636    48.82    4752
100.00       1     0.00       0    99.99    8781     0.00    4352    50.00    4352
100.00       7     0.00       0    99.99    8387     0.00    4154    50.00    4154
  0.00       0     0.00       0    99.99    9215     0.00    4566    50.00    4566
  0.00       0     0.00       0    99.99    8237     0.00    4080    50.14    4069
  0.00       0     0.00       0    99.99    7789     0.00    3842    49.86    3853
```

With this tool, you can measure the filesystem buffer utilization. You must ignore the first line, because it is an accumulation of statistics since the server was booted:

- Directory Name Lookup Cache (DNLC) and inode cache: Every time that a process tries to look for a directory, inodes, and file metadata, it first looks in the cache memory. If the information is not there, it must go to the disk. If you are not reaching higher than 90%, you might need to increase the size of bufhwm.
- UFS buffer cache: Whenever a process accesses a file, it first checks in the buffer cache to see whether the file pages of that file are still there. If not, it has to get the data from disk. If you see a hit cache percentage under 90%, you might have problems with data buffering in memory.

You can check the actual buffer size with the **sysdef** command that is shown in Example 9-28.

Example 9-28 The sysdef command output

```
[root@v480-1]# sysdef
skipping…
*
* Tunable Parameters
*
85114880        maximum memory allowed in buffer cache (bufhwm)
   30000        maximum number of processes (v.v_proc)
      99        maximum global priority in sys class (MAXCLSYSPRI)
   29995        maximum processes per user id (v.v_maxup)
      30        auto update time limit in seconds (NAUTOUP)
      25        page stealing low water mark (GPGSLO)
       1        fsflush run rate (FSFLUSHR)
      25        minimum resident memory for avoiding deadlock (MINARMEM)
      25        minimum swapable memory for avoiding deadlock (MINASMEM)
skipping...
```

To change bufhwm, you need to edit the `/etc/system` and look for the parameter bufhwm.

For additional information and to download this tool, see the following link:

http://www.brendangregg.com/cachekit.html

You can also use **sar -a**, **sar -b**, and **sar -v** to check the DNLC and inode cache utilizations. Check the following links for more details about how to use **sar** in Solaris:

- The **sar -a** and **sar -b** command:

  http://dennis_caparas.tripod.com/Configuring_sar_for_your_system.html

directiostat

The **directiostat** command and its output are illustrated in Example 9-29.

Example 9-29 The directiostat command output

```
[root@v480-1]# ./directiostat 1 5
  lreads lwrites  preads pwrites     Krd      Kwr holdrds  nflush
       0       0       0       0       0        0       0       0
       0       0       0       0       0        0       0       0
       0       0       0       0       0        0       0       0
       0       0       0       0       0        0       0       0
       0       0       0       0       0        0       0       0
```

With this tool, you can measure the I/O request that is executed in filesystems with the Direct I/O mount option enabled.

For additional information and to download this tool, see the following link:

http://www.solarisinternals.com/wiki/index.php/Direct_I/O

9.5.2 Solaris vmstat

The **vmstat** tool reports virtual memory activity. If intense I/O activity occurs, you can use this tool to determine whether you have a memory shortage issue or whether the server is experiencing a large amount of file cache in or file cache out. With certain operating systems, you cannot distinguish the two types of paging activity. Execute the command that is shown in Example 9-30.

Example 9-30 The vmstat -p command output

```
[root@v480-1]# vmstat -p 1 5
     memory           page          executable      anonymous      filesystem
   swap  free  re  mf  fr  de  sr  epi  epo  epf  api  apo  apf  fpi  fpo  fpf
 3419192 3244504 3  3   0   0   0    0    0    0    0    0    0    0    0    0
 3686632 3669336 8 19   0   0   0    0    0    0    0    0    0    0    0    0
 3686632 3669392 0  0   0   0   0    0    0    0    0    0    0    0    0    0
 3686632 3669392 0  0   0   0   0    0    0    0    0    0    0    0    0    0
 3686632 3669392 0  0   0   0   0    0    0    0    0    0    0    0    0    0
```

The **vmstat** output has five major columns (memory, page, executable, anonymous, and filesystem). The filesystem column contains three subcolumns:

- fpi: It means file pages in. It tells how many file pages were copied from disk to memory.
- fpo: It means file pages out. It tells how many file pages were copied from memory to disk.
- fpf: It means file pages free. It tells how many file pages are being freed at every sample of time.

If you see no page activity in anonymous (api/apo) and only in file page activity (fpi/fpo), it means that you do not have memory constraints but that too many file page activities occur and you might need to optimize it. One way is by enabling Direct I/O in the filesystems of your application. Another way is by adjusting the read ahead mechanism if that mechanism is enabled, or adjusting the scanner parameters of virtual memory. We suggest the following values:

- fastscan: This parameter sets how many memory pages are scanned per second. Configure it for 1/4 of real memory with a limit of 1 GB.
- handspreadpage: This parameter sets the distance between the two-handled clock algorithm to look for candidate memory pages to be reclaimed when memory is slow. Configure it with the same value set for the fastscan parameter.
- maxpgio: This parameter sets the maximum number of pages that can be queued by the Virtual Memory Manager. Configure it for 1024 if you use eight or more ranks in the DS8000 and if you use a high-end server.

9.5.3 Solaris iostat

The **iostat** tool reports I/O activities from disks and slices in the operating system. You must ignore the first line, because it is an accumulation of statistics since the server was booted. Execute the command as shown in Example 9-31.

Example 9-31 The iostat command output

```
[root@v480-1]# iostat -xn 1 5
                    extended device statistics
```

```
 r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.9   0   0 c0t0d0
 0.0    0.4    0.1    1.4  0.0  0.0    0.6   23.0   0   0 c1t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t1d0
                 extended device statistics
 r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c0t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t1d0
                 extended device statistics
 r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c0t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t1d0
                 extended device statistics
 r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c0t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t1d0
                 extended device statistics
 r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c0t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t0d0
 0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c1t1d0
```

When analyzing the output, look at the following situations:

- Look to see whether the number of I/Os is balanced among the disks. If not, it might indicate that you have problems in the distribution of subdisks or plexes (VxVM), or volumes (SVM) over the LUNs. You can use the option **-p** in **iostat** to see the workload among the slices of a LUN.
- If r/s is larger than w/s and if the asvc_t is larger than 15 ms, it means that your bottleneck is in a lower layer, which can be the HBA, SAN, or even in the storage. For some reason, it is taking too much time to process the I/O request. Also, check whether the same problem occurs with other disks of the same Disk Set or Disk Group (DG). If yes, you need to add the number of I/Os per second, add the throughput by vpath (if it is the case), rank, and host, and compare them with the performance numbers from TotalStorage Productivity Center for Disk.
- If r/s is smaller than w/s and if the asvc_t is larger than 3 ms, writes that average significantly and consistently higher indicate that the write cache is full, and there is a bottleneck in the disk. Confirm this information with the Tivoli Storage Productivity Center for Disk reports.
- If the wait is greater than actv, compare with other disks of the storage subsystem. See whether the distribution of subdisks or plexes (VxVM) or volumes (SVM) is evenly distributed among all disks in the Disk Set or DG. If the wait value is smaller than actv, compare it with other disks of the storage. If there are differences and the subdisks, plexes (VxVM), or volumes (SVM) are distributed evenly in the Disk Set or DG, it might indicate that the unbalance is at the rank level. Confirm this information with the Tivoli Storage Productivity Center for Disk reports.

For detailed information about the options of **iostat**, see the following link:

http://docs.oracle.com/cd/E23824_01/html/821-1451/spmonitor-4.html

9.5.4 vxstat

The **vxstat** performance tool is part of VxVM. With this tool, you can collect performance data related to VM disks, subdisks, plexes, and volumes. It can provide the following information:

- Operations (reads/writes): The number of I/Os over the sample interval
- Blocks (reads/writes): The number of blocks in 512 bytes over the sample interval
- Avg time (reads/writes): The average response time for reads and writes over the sample interval

With the DS8000, we suggest that you collect performance information from VM disks and subdisks. To display 10 sets of disk statistics, with intervals of one second, use **vxstat -i 1 -c 10 -d**. To display 10 sets of subdisk statistics, with intervals of one second, use **vxstat -i 1 -c 10 -s**. You need to dismiss the first sample, because it provides statistics since the boot of the server.

When analyzing the output of **vxstat**, focus on the following areas:

- Whether the number of I/Os is balanced among the disks. If not, it might indicate that you have problems in the distribution of subdisks or plexes (VxVM) over the LUNs.
- If operations/read is bigger than operations/write and if the Avg time/read is longer than 15 ms, it means that your bottleneck is in a lower layer, which can be the HBA, SAN, or even in the storage. For some reason, it takes too much time to process the I/O request. Also, check whether the same problem occurs with other disks of the same Disk Group (DG) or LUNs of the DS8000. If yes, you need to add the number of I/Os per second, add the throughput by VM disk, rank, and host, and compare with the performance numbers from Tivoli Storage Productivity Center for Disk.
- If operations/read is smaller than operations/write and if the Avg time/write is greater than 3 ms, writes that average significantly and consistently higher might indicate that write cache is full, and that there is a bottleneck in the disk. Confirm this information with the TotalStorage Productivity Center for Disk reports.
- Whether the distribution of subdisks or plexes (VxVM) is even in all disks in the DG. If there are differences, and the subdisks or plexes (VxVM) are distributed evenly in the DG, it can indicate that the unbalance might be at the rank level.

9.5.5 dtrace

The **dtrace** is not only a trace tool. DTrace is also a framework for tracking dynamically the operating system's kernel and also applications that run on top of Solaris 10. You can write your own tools for performance analysis by using the D programming language. The syntax is based on the C programming language with several specific commands for tracing instrumentation. There are many scripts already developed that you can use for performance analysis. You can start by downloading the DTrace Toolkit from the following link:

http://www.brendangregg.com/dtrace.html#DTraceToolkit

Follow the instructions at the website to install the DTrace Toolkit. When installed, set your PATH environment variable to avoid having to type the full path every time, as shown in Example 9-32.

Example 9-32 Setting PATH environment variable for DTrace Toolkit

```
[root@v480-1]# export PATH=$PATH:/opt/DTT:/opt/DTT/Bin
```

One example is a large sequential I/O that might be reaching a limit. See the script in Example 9-33 on page 381.

Example 9-33 bitesize.d script

```
[root@v480-1]# dd if=/dev/zero of=/export/home/test_file.dd bs=2048k cou>
[1]     6516
[root@v480-1]# bitesize.d
Tracing... Hit Ctrl-C to end.
1000+0 records in
1000+0 records out
^C

     PID  CMD
       0  sched\0

           value  ------------- Distribution ------------- count
             256 |                                         0
             512 |@@@@@@@@@                                59
            1024 |@@@                                      22
            2048 |                                         1
            4096 |                                         0
            8192 |@@@@@@@@@@@@@@@@@@@@@@@@@@@              191
           16384 |                                         0

       3  fsflush\0

           value  ------------- Distribution ------------- count
             512 |                                         0
            1024 |                                         1
            2048 |                                         1
            4096 |                                         0
            8192 |@@@@@@@                                  15
           16384 |@@@@@@                                   14
           32768 |@@@@@                                    11
           65536 |@@@@@@                                   12
          131072 |@@                                       5
          262144 |@@                                       4
          524288 |@@@@@@@@@@@                              24
         1048576 |                                         0

    6516  dd if=/dev/zero of=/export/home/test_file.dd bs=2048k count=1000\0

           value  ------------- Distribution ------------- count
            2048 |                                         0
            4096 |                                         2
            8192 |@@                                       118
           16384 |@                                        29
           32768 |@                                        47
           65536 |@@                                       142
          131072 |@                                        62
          262144 |@                                        47
          524288 |@@@                                      150
         1048576 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@           1692
         2097152 |                                         0


[1] +  Done                    dd if=/dev/zero of=/export/home/test_file.dd bs=2048k
count=1000&
```

In the previous example, we executed a **dd** command with a blocksize of 2 MB, but when we measure the I/O activity, we can see that in fact the maximum I/O size is 1 MB and not 2 MB, which might be related to the maximum physical I/O size that can be executed in the operating system. Let us confirm the maxphys value (Example 9-34).

Example 9-34 Checking the maxphys parameter

```
[root@v480-1]# grep maxphys /etc/system
[root@v480-1]#
```

As you can see, the maxphys parameter is not set in the `/etc/system` configuration file. It means that Solaris 10 is using the default value, which is 1 MB. You can increase the value of maxphys to increase the size of I/O requests.

For additional information about how to use DTrace, check the following links:

- Introduction about DTrace:

  http://www.solarisinternals.com/wiki/index.php/DTrace_Topics_Intro

- The DTrace Toolkit:

  http://www.solarisinternals.com/wiki/index.php/DTraceToolkit

9.6 HP-UX disk I/O architecture

HP-UX 11i v3 comes with several performance enhancements and new features. The two most important from an I/O perspective are the LVM Version 2 and the new Mass Storage Subsystem.

When planning the DS8000 volume layout for HP-UX host systems, due to limitations in the host operating system, the DS8000 LUN IDs greater than x'3FFF' are not supported. When you create or assign LUNs and volumes, only LUN and volume IDs less than x'3FFF' can be used, which limits the maximum number of volumes that can be used on a single DS8000 system for HP-UX hosts to 16384.

See the DS8000 Information Center:

http://publib.boulder.ibm.com/infocenter/dsichelp/ds8000ic/index.jsp

Select **Configuring** → **Attaching hosts** → **HP-UX host attachment** for initial information about configuring HP-UX hosts for attachment to the DS8000 systems.

9.6.1 HP-UX High Performance File System (HFS)

High Performance File System (HFS) is the HP-UX product that is similar to the UNIX File System (UFS). It was designed for small files. It is not a journaled filesystem, and it also does not support Direct I/O.

Blocksize, fragment size, cylinders per cylinder group, and minfree

The default blocksize in HFS is 1 KB. You must adjust this value based on the average size of the application files. For database applications, for example, files can easily be bigger than 2 GB. In that case, set the filesystem blocksize equal to the database blocksize. Use the same value defined in blocksize for fragment size. For cylinders per cylinder group, if your application is a database, set it to 32. For minfree, if your application is a database, set it to 2% or zero.

HFS read ahead

The read ahead mechanism in HFS is set by the parameter hfs_ra_per_disk, and the value is in KB. This value needs to match the I/O size of your application. For example, if you use the Oracle database, it is better use a value that matches the product of the values of the DB_BLOCK_SIZE and DB_FILE_MULTIBLOCK_READ_COUNT parameters. For DB2, the read ahead value needs to match the product of the database blocksize and extent size.

Asynchronous I/O

Asynchronous I/O is a feature of HP-UX that is not enabled by default. It allows the application to keep processing while issuing I/O requests without needing to wait for a reply, therefore, reducing the application response time. Normally, database applications take advantage of that feature. If your application supports asynchronous I/O, enable it in the operating system as well. For detailed information about how to configure asynchronous I/O, see the appropriate application documentation:

- For Oracle 11g with HP-UX using asynchronous I/O:

  http://download.oracle.com/docs/cd/B28359_01/server.111/b32009/appb_hpux.htm#BABBFDCI

- For Sybase Adaptive Server Enterprise (ASE) 15.0 using asynchronous I/O:

  http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.dc35823_1500/html/uconfig/BBCBEAGF.htm

Tip: If there is no suggestion for the max_async_ports parameter, set it to 1024.

Dynamic buffer cache

Other important options for HP-UX and I/O performance are dbc_min_pct and dbc_max_pct. These kernel parameters, dbc_min_pct and dbc_max_pct, control the lower and upper limit, as a percentage of system memory that can be allocated for buffer cache.

The number of pages of memory allocated for buffer cache use at any specific time is determined by the system needs, but the two parameters ensure that allocated memory never drops under dbc_min_pct and does not exceed dbc_max_pct of the total system memory.

The default value for dbc_max_pct is 50%, which is usually too much. If you want to use a dynamic buffer cache, set the dbc_max_pct value to 25%. If you have 4 GB of memory or more, start with an even smaller value.

With a large buffer cache, the system is likely to need to page out or shrink the buffer cache to meet the application memory needs. These actions cause I/Os to lose paging space. You need to avoid that situation from happening and set memory buffers to favor applications over cached files.

9.6.2 HP-UX Journaled File System (JFS)

Journaled File System (JFS) is the HP-UX version of Veritas File System (VxFS). It provides almost the same features as VxFS. Also, OnlineJFS is an optional product that you can use to to change online the size of the JFS filesystem. The latest version of VxFS is 5.0. For additional information about VxFS, see 9.4.2, “Symantec Veritas File System (VxFS) for Solaris” on page 370. Also, see the following links:

- JFS tuning and performance V3.5 in *HP-UX Performance Cookbook*:

  http://h21007.www2.hp.com/portal/download/files/unprot/devresource/docs/techpapers/uxperfcookbook.pdf

- Supported file and filesystem sizes for HFS and JFS:

  http://docs.hp.com/en/5992-4023/5992-4023.pdf

- Veritas documentation at the HP website:

  http://docs.hp.com/en/oshpux11iv3.html#VERITAS%20Volume%20Manager%20and%20File%20System

- A paper about the most common misconfigured resources, among them, the filesystem cache mechanisms:

  http://docs.hp.com/en/5992-0732/5992-0732.pdf

- A paper that provides the suggested server tuning parameters for most types of workloads:

  http://docs.hp.com/en/5992-4222ENW/5992-4222ENW.pdf

9.6.3 HP Logical Volume Manager (LVM)

There are two versions of the HP LVM: Version 1 and Version 2. Version 2 was introduced with the March 2008 release of HP-UX 11i v3 (11.31). LVM supports RAID 0 or RAID 1, or, with Version 2, a combination of both RAID 0 and RAID 1. It is also possible to spread the data among the disks through LVM Distributed Allocation Policy. Figure 9-11 gives an overview.

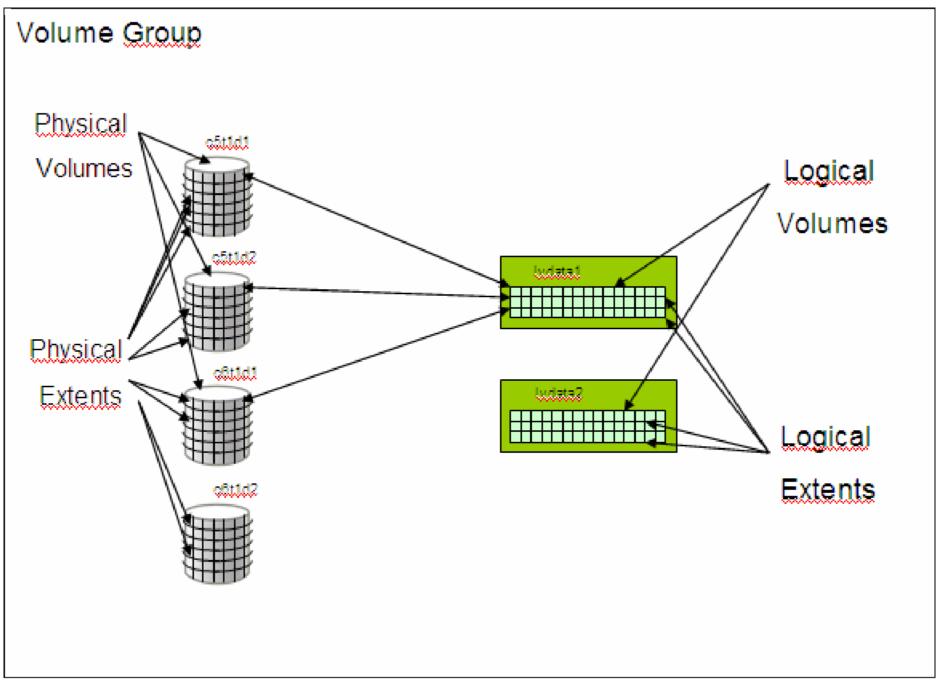


Figure 9-11 HP-UX LVM overview

The DS8000 LUNs that are under control of LVM are called *physical volumes* (PVs). The LVM splits the disk space in smaller pieces that are called *physical extents* (PEs). A *logical volume* (LV) is composed of several *logical extents* (LEs). A filesystem is created on top of an LV or simply used as a raw device. Each LE can point to up to two corresponding PEs in LVM Version 1.0 and up to five corresponding PEs in LVM Version 2.0/2.1, which is how LVM implements mirroring (RAID 1).

Requirement: To implement mirroring (RAID 1), it is necessary to install an optional product, HP MirrorDisk/UX.

To set up the volume layout with the DS8000 LUNs, you can adopt one of the following strategies:

- Storage pool striping: In this case, you spread the workload at the storage level. At the operating system level, you need to create the LVs with the inter-policy attribute set to minimum, which is the default option when creating an LV.
- Distributed Allocation Policy: A set of LUNs is created in different ranks inside the DS8000. When the LUNs are recognized in the HP-UX, a VG is created and the LVs are spread evenly over the LUNs with the option `-D`. The advantage of this method compared to storage pool striping is the granularity of data spread over the LUNs. With storage pool striping, the data is spread in chunks of 1 GB. With the Distributed Allocation Policy, you can create PE sizes from 8 MB to 16 MB in a VG.
- LVM Striping: As with Distributed Allocation Policy, a set of LUNs is created in different ranks inside the DS8000. When the LUNs are recognized in the HP-UX, a VG is created with larger PE sizes, such as 128 MB or 256 MB. And the LVs are spread evenly over the LUNs by setting the stripe size of LV from 8 MB to 16 MB. From a performance standpoint, LVM Striping and PP Striping provide the same results.

Volume group limits

When creating the volume group, there are LVM limits to consider along with the potential expansion of the volume group. The main LVM limits for a volume group are shown in Table 9-2.

Table 9-2 HP-UX volume group characteristics

	LVM 1.0	LVM 2.0	LVM 2.1
Maximum PVs/VG	255	2047	2047
Maximum LVs/VG	255	2047	2047
Maximum PEs/PV	65535	16777216	16777216

Tip: We suggest that you use LVM Version 2.0 or later.

LVM Distributed Allocation Policy

Figure 9-12 on page 386 shows an example of the inter-disk policy logical volume. The LVM created a volume group that contains four LUNs and created 16 MB physical partitions on the LUNs. The logical volume in this example is a group of 16 MB physical partitions from four different logical disks: disk4, disk5, disk6, and disk7.

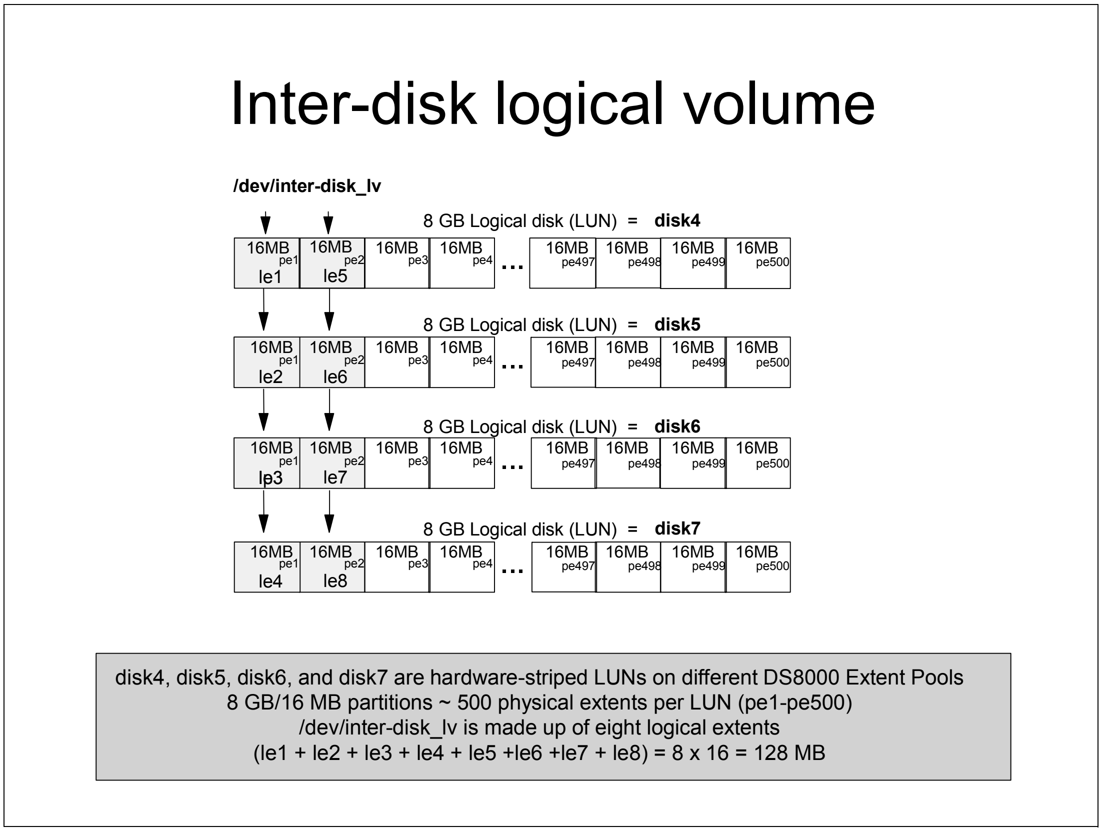


Figure 9-12 Inter-disk policy logical volume

The first step is to initialize the PVs with the following commands:

- `pvcreate /dev/rdisk/disk4`
- `pvcreate /dev/rdisk/disk5`
- `pvcreate /dev/rdisk/disk6`
- `pvcreate /dev/rdisk/disk7`

The next step is to create the VG. We suggest that you create a VG with a set of DS8000 LUNs, and each LUN is in a different extent pool. If you add a set of LUNs to a host, define another VG, and so on. Follow these steps to create the VG data01vg and the PE size of 16 MB:

1. Create the directory `/dev/data01vg` with a special character file called `group`:

```
mkdir /dev/data01vg
mknod /dev/data01vg/group c 70 0x020000
```

2. Create the VG with the following command:

```
vgcreate /dev/data01vg -g data01pvg01 -s 16 /dev/data01vg /dev/disk/disk4
/dev/disk/disk5 /dev/disk/disk6 /dev/disk/disk7
```

Then, you can create the LVs with the option **`-D`**, which stripes the logical volume from one LUN to the next LUN in chunks the size of the physical partition size of the volume group, for instance:

```
lvcreate -D y -l 16 -m 1 -n inter-disk_lv -s g /dev/data01vg
```

LVM striping

An example of a striped logical volume is shown in Figure 9-13. The logical volume called /dev/striped_lv uses the same capacity as /dev/inter-disk_lv (shown in Figure 9-12 on page 386), but it is created differently.

Notice that /dev/striped_lv is also made up of eight 256 MB physical partitions, but each partition is then subdivided into 64 chunks of 4 MB (only three of the 4 MB chunks are shown per logical partition for space reasons).

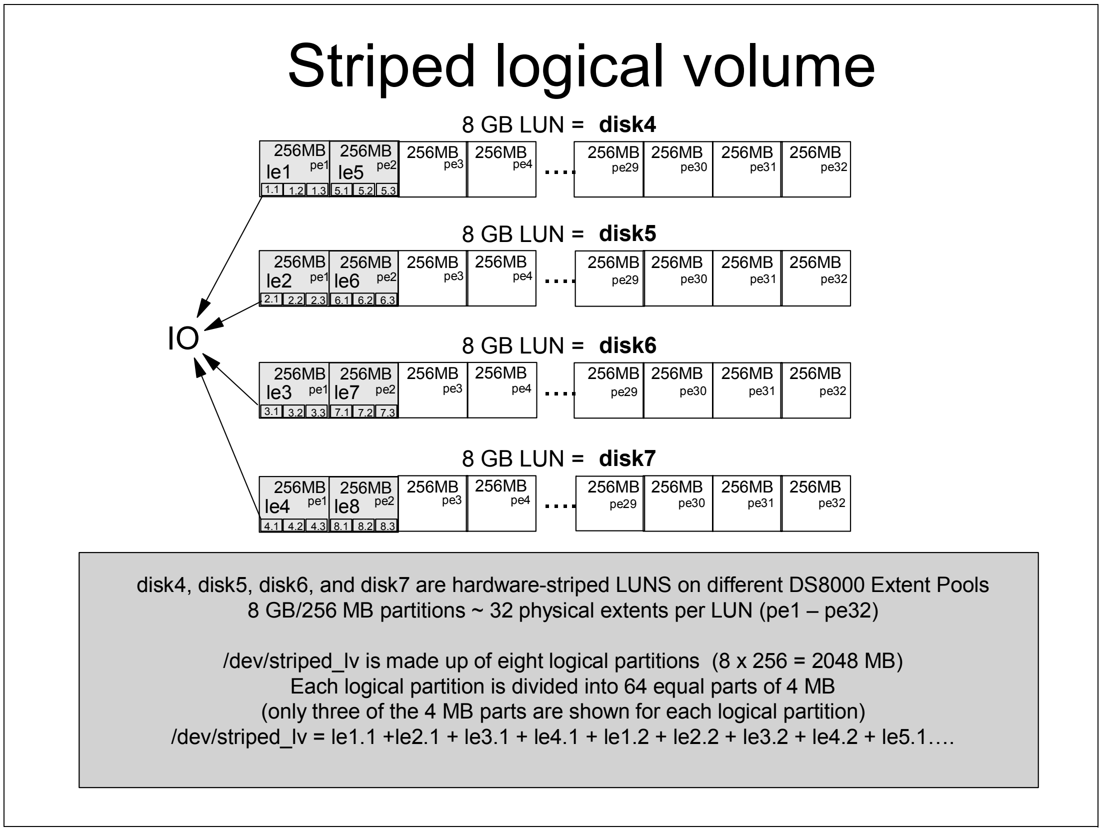


Figure 9-13 Striped logical volume

As with LVM Distributed Allocation Policy, the first step is to initialize the PVs with the following commands:

- `pvcreate /dev/rdisk/disk4`
- `pvcreate /dev/rdisk/disk5`
- `pvcreate /dev/rdisk/disk6`
- `pvcreate /dev/rdisk/disk7`

Then, you need to create the VG with a PE size of 256 MB:

1. Create the directory `/dev/data01vg` with a special character file called `group`:

   ```
   mkdir /dev/data01vg
   mknod /dev/data01vg/group c 70 0x020000
   ```

2. Create the VG with the following command:

   ```
   vgcreate /dev/data01vg -g data01pvg01 -s 256 /dev/data01vg /dev/disk/disk4
   /dev/disk/disk5 /dev/disk/disk6 /dev/disk/disk7
   ```

The last step is to create all of the needed LVs with a stripe size of 8 MB. To create a striped LV, you need to combine the following options:

- Number of LEs (**-l**): This option sets the size of your LV. In our case, we want to create a 2 GB LV. Knowing that the PE size is 256 MB, we need to divide 2048 by 256 to find out how many LEs are needed.
- Stripes (**-i**): This option sets the number of disks where the data needs to be spread.
- Stripe size (**-I**): This option sets the size in kilobytes of the stripe.
- Name of LV (**-n**): This option sets the name of the LV.

For each LV, execute the following command:

```
lvcreate -l 8 -i 4 -I 8192 -n striped_lv /dev/data01vg
```

For additional information, see these resources:

- *LVM Limits* white paper:

  http://docs.hp.com/en/6054/LVM_Limits_White_Paper_V4.pdf
- *LVM Version 2.0 Volume Groups in HP-UX 11i v3*:

  http://docs.hp.com/en/lvm-v2/L2_whitepaper_8.pdf
- *LVM New Features in HP-UX 11i v3*:

  http://docs.hp.com/en/LVM-11iv3features/LVM_New_Features_11iv3_final.pdf
- LVM documentation:

  http://docs.hp.com/en/oshpux11iv3.html#LVM%20Volume%20Manager

9.6.4 Veritas Volume Manager (VxVM) for HP-UX

The Veritas Volume Manager (VxVM) is another LVM that is available for HP-UX. It is similar to VxVM for Solaris and AIX. See 9.2.5, “Symantec Veritas Volume Manager (VxVM)” on page 347 for a brief description of features and additional information.

9.6.5 PV Links

PV Links is the multipathing solution that comes with HP-UX. It primarily provides a failover capability, but if the storage allows it, you can use the alternate path for load balancing.

For more information, see the following link:

http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01919369/c01919369.pdf

9.6.6 Native multipathing in HP-UX

Native multipathing is the multipathing solution that is imbedded in HP-UX 11i v3. Native multipathing uses a new way to represent storage devices, known as the *agile view*. Instead of having the device names of disks and tape drives represented by a hardware path, the system uses device names based on objects. In the old way (now called the “legacy” view), the name of a LUN used to look like /dev/dsk/c4t2d0 or /dev/rdsk/c4t2d0 and used to correspond to controller 5, SCSI target 2, and SCSI LUN 0. With the new way, the name of a LUN looks like /dev/disk/disk7 or /dev/rdisk/disk7. This new way of representing storage devices resulted in a series of necessary modifications to existing commands, the implementation of new commands, modifications in LVM, a new hardware addressing model, and changes in tunable parameters. In addition, the storage subsystem needs to be

Asymmetric Logical Unit Access (ALUA) compliant. For additional information, see the following links:

- *The Next Generation Mass Storage Stack HP-UX 11i v3:*

  http://docs.hp.com/en/MassStorageStack/The_Next_Generation_Mass_Storage_Stack.pdf

- *HP-UX 11i v3 Native Multi-Pathing for Mass Storage:*

  http://docs.hp.com/en/native-multi-pathing/native_multipathing_wp_AR0709.pdf

- *HP-UX 11i v3 Mass Storage I/O Performance Improvements:*

  http://docs.hp.com/en/11iv3IOPerf/IOPerformanceWhitePaper.pdf

- *LVM Migration from Legacy to Agile Naming Model:*

  http://docs.hp.com/en/LVMmigration1/LVM_Migration_to_Agile.pdf

9.6.7 Subsystem Device Driver (SDD) for HP-UX

SDD is the IBM multipathing device driver that is available for HP-UX machines. For the correct version to use with HP-UX, see the following link:

http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303

9.6.8 Veritas Dynamic MultiPathing (DMP) for HP-UX

Veritas DMP is the device driver provided by Symantec to work with VxVM. For additional information, see 9.4.8, "Symantec Veritas Dynamic MultiPathing (DMP) for Oracle Solaris" on page 375.

9.6.9 Array Support Library (ASL) for HP-UX

The Array Support Library (ASL) is a software package used by Device Discovery Layer, which is a component of VxVM to support third-party arrays. For additional information, see 9.4.9, "Array Support Library (ASL)" on page 375.

9.6.10 FC adapter

The FC adapter provides the connection between the host and the storage devices.

9.7 HP-UX performance monitoring tools

Next, we describe the commands available to monitor the performance of each layer of the HP-UX I/O subsystem architecture.

9.7.1 HP-UX System Activity Report (SAR)

System Activity Report (SAR) is a tool that reports the contents of certain cumulative activity counters within the UNIX operating system. The **sar** command has numerous options, providing paging, TTY, CPU busy, and many other statistics. Used with the appropriate command flag (**-u**), the **sar** command provides a quick way to identify whether a system is I/O-bound. There are three possible modes in which to use the **sar** command:

- Real-time sampling and display
- System activity accounting through cron
- Displaying previously captured data

Real-time sampling and display

One way to run **sar** is to specify a sampling interval and the number of times that you want it to run. To collect and display system statistic reports immediately, run **sar -u 1 5**. Example 9-35 shows an example of **sar** output.

Example 9-35 HP-UX sar sample output

```
[root@rx4640-1]# sar -u 1 5

HP-UX rx4640-1 B.11.31 U ia64    11/13/08

10:57:56    %usr    %sys    %wio   %idle
10:57:57       3      23      12      62
10:57:58       1       6      42      51
10:57:59       1       9      36      54
10:58:00       1      10      37      52
10:58:01       1       1      70      28

Average        1      10      39      50
```

Not all **sar** options are the same for AIX, HP-UX, and Solaris, but the **sar -u** output is the same. The output in Example 9-35 shows CPU information every 1 second, five times.

To check whether a system is I/O-bound, the important column to check is the %wio column. The %wio column includes the time that is spent waiting on I/O from *all* drives, including internal and DS8000 logical disks. If %wio values exceed 40, you need to investigate to understand the storage I/O performance. The next action is to look at I/O service times reported by the **sar -Rd** command (Example 9-36).

Example 9-36 HP-UX sar -d output

```
[root@rx4640-1]# sar -Rd 1 5

HP-UX rx4640-1 B.11.31 U ia64    11/13/08

11:01:11    device   %busy   avque     r/s     w/s  blks/s  avwait  avserv
11:01:12     disk3    9.90    0.50      11       6    1006    0.00   28.01
11:01:13     disk3   15.00    3.46      27      28     792    9.11   11.45
11:01:14     disk3    9.00    4.35       4      23     402   16.18   22.43
11:01:15     disk3    9.09    2.38       6      18     598    7.86   19.53
11:01:16     disk3   11.88    2.05       9      20     388    8.36   16.87

Average      disk3   10.98    2.85      11      19     638    9.00   17.56
```

The avwait and avserv columns show the average times spent in the wait queue and service queue. The avque column represents the average number of I/Os in the queue of that device. With HP-UX 11i v3, the **sar** command has new options to monitor the performance:

- **-H** reports I/O activity by HBA
- **-L** reports I/O activity by lunpath
- **-R** with option **-d** splits the number of I/Os per second between reads and writes
- **-t** reports I/O activity by tape device

Analyze the output of the **sar** command:

- Check whether the number of I/Os is balanced among the disks. If not, it might indicate that you have problems in the distribution of Logical Volumes (LVs) over the LUNs.

- Check whether the I/O is balanced among the controllers (HBAs). If not, certain paths might be in failed status.
- If r/s is larger than w/s and if the avserv is larger than 15 ms, your bottleneck is in a lower layer, which can be the HBA, SAN, or even in the storage. It takes too much time to process the I/O request. Also, check whether the same problem occurs with other disks of the same volume group (VG) or DS8000 LUNs. If yes, you need to add the number of I/Os per second, add the throughput by LUN, rank, and host, and compare with the performance numbers from Tivoli Storage Productivity Center for Disk.
- If r/s is smaller than w/s, the avserv is greater than 3 ms, and the writes are averaging significantly and consistently higher, it might indicate that write cache is full, and there is a bottleneck in the disk. Confirm this information with the TotalStorage Productivity Center for Disk reports.
- Check that LVs are evenly distributed across all disks in the VG. If there are differences and the LVs are distributed evenly in the VG, it might indicate that the unbalance is at the rank level.

For additional information about the HP-UX **sar** command, see this website:

http://docs.hp.com/en/B2355-60130/sar.1M.html

System activity accounting through cron

The **sar** data collection program is an unintrusive program, because it extracts data from information collected by the system. You need to configure a system to collect data. The frequency of the data collection can affect performance and the size of the data files that are collected.

To configure a system to collect data for **sar**, you can run the **sadc** command or the modified **sa1** and **sa2** commands. The following list includes more information about the **sa** commands and how to configure **sar** data collection:

- The **sa1** and **sa2** commands are shell procedure variants of the **sadc** command.
- The **sa1** command collects and stores binary data in the `/var/adm/sa/sadd` file, where *dd* is the day of the month.
- The **sa2** command is designed to be run automatically by the **cron** command and run concurrently with the **sa1** command. The **sa2** command generates a daily report called /var/adm/sa/sardd. It also removes a report more than one week old.
- The `/var/adm/sa/sadd` file contains the daily data file, and *dd* represents the day of the month. And, `/var/adm/sa/sardd` contains the daily report file, and *dd* represents the day of the month. Note the `r` in `/var/adm/sa/sardd` for **sa2** output.

To configure a system to collect data, edit the root crontab file. For our example, if we want to run **sa1** every 15 minutes every day, and we want the **sa2** program to generate ASCII versions of the data immediately before midnight, we change the cron schedule to look like this example:

```
0,15,30,45 * * * 0-6 /usr/lib/sa/sa1
55 23 * * 0-6 /usr/lib/sa/sa2 -A
```

Display previously captured data

After the **sa1** and **sa2** commands are configured in cron and data collection starts, you see binary report files in `/var/adm/sa/sadd`, where *dd* represents the day of the month.

You can view performance information files from these files with this command:

`sar -f /var/adm/sa/sa`*`dd`* (*dd* is the day in which you are interested)

You can also focus on a certain period, for example, 8 a.m. to 5:15 p.m. with this command:

```
sar -s 8:00 -e 17:15 -f  /var/adm/sa/sadd
```

Remember, **sa2** removes the data collection files that are over a week old as scheduled in cron.

You can save **sar** information to view later with the commands:

`sar -A -o data.file interval count > /dev/null &` (SAR data saved to `data.file`)

`sar -f data.file` (Read SAR information back from saved file)

All data is captured in binary form and saved to a file (`data.file`). The data can then be selectively displayed with the **sar** command by using the **-f** option.

sar summary

The **sar** tool helps you to tell quickly if a system is I/O-bound. Remember though that a busy system can mask I/O issues, because io_wait counters are not increased if the CPUs are busy.

The **sar** tool can help you to save a history of I/O performance so that you have a baseline measurement for each host. You can then verify whether tuning changes make a difference. You might want, for example, to collect **sar** data for a week and create reports: 8 a.m. - 5 p.m. Monday - Friday if that time is the prime time for random I/O, and 6 p.m. - 6 a.m. Saturday - Sunday if those times are batch/backup windows.

9.7.2 vxstat

The **vxstat** tool is a performance tool that comes with VxVM. For additional information, see 9.5.4, “vxstat” on page 380.

9.7.3 HP Perfview/Measureware

HP Perfview/Measurware is good for recording performance measurements and maintaining a baseline of system performance data for reference. The HP Perfview/Measurware tool can show statistics for each physical disk in graphical format and you can change the time scale easily.

9.8 Verifying your system

At each step along the way, there are tests that you must run to test the storage infrastructure. We describe these tests to provide you with testing techniques as you configure your storage for the best performance.

Verify the storage subsystem

After the LUNs are assigned to the host system, and multipathing software, such as SDD, discovers the LUNs, it is important to test the storage subsystem. The storage subsystem includes the SAN infrastructure, the host system HBAs, and the DS8000.

Follow these steps to test the storage subsystem:

1. Run the **datapath query essmap** command to determine whether your storage allocation from the DS8000 works well with SDD. Or, for SDDPCM, run **pcmpath query essmap**.
2. Make sure that the LUNs are set up in the manner that you expected. Is the number of paths to the LUNs correct? Are all of the LUNs from different ranks? Are the LUN sizes correct? The output from the command looks like the output in Example 9-37.

Example 9-37 The datapath query essmap command output

```
{CCF-part2:root}/tmp/perf/scripts -> datapath query essmap
 Disk          Path  P     Location    adapter    LUN SN        Type         Size   LSS     Vol  Rank  C/A   S   Connection  port RaidMode
-------       -----  -   -----------  ------   -----------  ------------   ----   ----    ---  ----- ----  -   ----------- ---- --------
vpath0        hdisk4     7V-08-01[FC] fscsi0   75065513000  IBM 2107-900   10.7    48      0   0000   0e   Y   R1-B2-H1-ZA  100 RAID10
vpath0       hdisk12     7V-08-01[FC] fscsi0   75065513000  IBM 2107-900   10.7    48      0   0000   0e   Y   R1-B2-H1-ZB  101 RAID10
vpath0       hdisk20     7k-08-01[FC] fscsi1   75065513000  IBM 2107-900   10.7    48      0   0000   0e   Y   R1-B1-H3-ZC   32 RAID10
vpath0       hdisk28     7k-08-01[FC] fscsi1   75065513000  IBM 2107-900   10.7    48      0   0000   0e   Y   R1-B1-H3-ZD   33 RAID10
vpath1        hdisk5     7V-08-01[FC] fscsi0   75065513001  IBM 2107-900   10.7    48      1   fffe   0e   Y   R1-B2-H1-ZA  100 RAID10
vpath1       hdisk13     7V-08-01[FC] fscsi0   75065513001  IBM 2107-900   10.7    48      1   fffe   0e   Y   R1-B2-H1-ZB  101 RAID10
vpath1       hdisk21     7k-08-01[FC] fscsi1   75065513001  IBM 2107-900   10.7    48      1   fffe   0e   Y   R1-B1-H3-ZC   32 RAID10
vpath1       hdisk29     7k-08-01[FC] fscsi1   75065513001  IBM 2107-900   10.7    48      1   fffe   0e   Y   R1-B1-H3-ZD   33 RAID10
vpath2        hdisk6     7V-08-01[FC] fscsi0   75065513002  IBM 2107-900   10.7    48      2   fffc   0e   Y   R1-B2-H1-ZA  100 RAID10
vpath2       hdisk14     7V-08-01[FC] fscsi0   75065513002  IBM 2107-900   10.7    48      2   fffc   0e   Y   R1-B2-H1-ZB  101 RAID10
vpath2       hdisk22     7k-08-01[FC] fscsi1   75065513002  IBM 2107-900   10.7    48      2   fffc   0e   Y   R1-B1-H3-ZC   32 RAID10
vpath2       hdisk30     7k-08-01[FC] fscsi1   75065513002  IBM 2107-900   10.7    48      2   fffc   0e   Y   R1-B1-H3-ZD   33 RAID10
vpath3        hdisk7     7V-08-01[FC] fscsi0   75065513003  IBM 2107-900   10.7    48      3   fffa   0e   Y   R1-B2-H1-ZA  100 RAID10
vpath3       hdisk15     7V-08-01[FC] fscsi0   75065513003  IBM 2107-900   10.7    48      3   fffa   0e   Y   R1-B2-H1-ZB  101 RAID10
vpath3       hdisk23     7k-08-01[FC] fscsi1   75065513003  IBM 2107-900   10.7    48      3   fffa   0e   Y   R1-B1-H3-ZC   32 RAID10
vpath3       hdisk31     7k-08-01[FC] fscsi1   75065513003  IBM 2107-900   10.7    48      3   fffa   0e   Y   R1-B1-H3-ZD   33 RAID10
vpath4        hdisk8     7V-08-01[FC] fscsi0   75065513100  IBM 2107-900   10.7    49      0   ffff   17   Y   R1-B2-H1-ZA  100 RAID10
vpath4       hdisk16     7V-08-01[FC] fscsi0   75065513100  IBM 2107-900   10.7    49      0   ffff   17   Y   R1-B2-H1-ZB  101 RAID10
vpath4       hdisk24     7k-08-01[FC] fscsi1   75065513100  IBM 2107-900   10.7    49      0   ffff   17   Y   R1-B1-H3-ZC   32 RAID10
vpath4       hdisk32     7k-08-01[FC] fscsi1   75065513100  IBM 2107-900   10.7    49      0   ffff   17   Y   R1-B1-H3-ZD   33 RAID10
vpath5        hdisk9     7V-08-01[FC] fscsi0   75065513101  IBM 2107-900   10.7    49      1   fffd   17   Y   R1-B2-H1-ZA  100 RAID10
vpath5       hdisk17     7V-08-01[FC] fscsi0   75065513101  IBM 2107-900   10.7    49      1   fffd   17   Y   R1-B2-H1-ZB  101 RAID10
vpath5       hdisk25     7k-08-01[FC] fscsi1   75065513101  IBM 2107-900   10.7    49      1   fffd   17   Y   R1-B1-H3-ZC   32 RAID10
vpath5       hdisk33     7k-08-01[FC] fscsi1   75065513101  IBM 2107-900   10.7    49      1   fffd   17   Y   R1-B1-H3-ZD   33 RAID10
vpath6       hdisk10     7V-08-01[FC] fscsi0   75065513102  IBM 2107-900   10.7    49      2   fffb   17   Y   R1-B2-H1-ZA  100 RAID10
vpath6       hdisk18     7V-08-01[FC] fscsi0   75065513102  IBM 2107-900   10.7    49      2   fffb   17   Y   R1-B2-H1-ZB  101 RAID10
vpath6       hdisk26     7k-08-01[FC] fscsi1   75065513102  IBM 2107-900   10.7    49      2   fffb   17   Y   R1-B1-H3-ZC   32 RAID10
vpath6       hdisk34     7k-08-01[FC] fscsi1   75065513102  IBM 2107-900   10.7    49      2   fffb   17   Y   R1-B1-H3-ZD   33 RAID10
```

3. Next, run sequential reads and writes (using the **dd** command, for example) to all of the vpath devices (raw or block) for about an hour. Then, look at your SAN infrastructure to see how it performs.

   Look at the UNIX error report. Problems show up as storage errors, disk errors, or adapter errors. If there are problems, they are not to hard to identify in the error report, because there are many errors. The source of the problem can be hardware problems on the storage side of the SAN, Fibre Channel cables or connections, down-level device drivers, or device (HBA) microcode. If you see errors similar to the errors shown in Example 9-38, stop and fix them.

   Example 9-38 SAN problems reported in UNIX error report

```
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
3074FEB7   0915100805 T H fscsi0         ADAPTER ERROR
3074FEB7   0915100805 T H fscsi3         ADAPTER ERROR
3074FEB7   0915100805 T H fscsi3         ADAPTER ERROR
825849BF   0915100705 T H fcs0           ADAPTER ERROR
3074FEB7   0915100705 T H fscsi3         ADAPTER ERROR
3074FEB7   0915100705 T H fscsi0         ADAPTER ERROR
3074FEB7   0914175405 T H fscsi0         ADAPTER ERROR
3074FEB7   0914175405 T H fscsi0         ADAPTER ERROR
3074FEB7   0914175305 T H fscsi0         ADAPTER ERROR
```

   Ensure that after you run **dd** commands on *all* your vpaths for one hour, there are no storage errors in the UNIX error report.

4. Next, issue the following command to see whether SDD correctly balances the load across paths to the LUNs:

```
datapath query device
```

Or, use **pcmpath query device** when you use SDDPCM. The output from this command looks like Example 9-39.

Example 9-39 The datapath query device command output

```
{CCF-part2:root}/tmp/perf/scripts -> datapath query device|more

Total Devices : 16


DEV#:   0  DEVICE NAME: vpath0  TYPE: 2107900         POLICY:    Optimized
SERIAL: 75065513000
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk4           OPEN   NORMAL     220544          0
    1         fscsi0/hdisk12           OPEN   NORMAL     220396          0
    2         fscsi1/hdisk20           OPEN   NORMAL     223940          0
    3         fscsi1/hdisk28           OPEN   NORMAL     223962          0

DEV#:   1  DEVICE NAME: vpath1  TYPE: 2107900         POLICY:    Optimized
SERIAL: 75065513001
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk5           OPEN   NORMAL     219427          0
    1         fscsi0/hdisk13           OPEN   NORMAL     219163          0
    2         fscsi1/hdisk21           OPEN   NORMAL     223578          0
    3         fscsi1/hdisk29           OPEN   NORMAL     224349          0

DEV#:   2  DEVICE NAME: vpath2  TYPE: 2107900         POLICY:    Optimized
SERIAL: 75065513002
==========================================================================
Path#      Adapter/Hard Disk          State     Mode     Select     Errors
    0          fscsi0/hdisk6           OPEN   NORMAL     218881          0
    1         fscsi0/hdisk14           OPEN   NORMAL     219835          0
    2         fscsi1/hdisk22           OPEN   NORMAL     222697          0
    3         fscsi1/hdisk30           OPEN   NORMAL     223918          0

DEV#:   3  DEVICE NAME: vpath3  TYPE: 2107900         POLICY:    Optimized
SERIAL: 75065513003
```

Check to ensure that for *every* LUN, the counters under the *Select* column are the same and that there are no errors.

5. Next, randomly check the sequential read speed of the raw vpath device. The following command is an example of the command run against a LUN called vpath0. For the LUNs that you test, ensure that they each yield the *same results*:

```
time dd if=/dev/rvpath0 of=/dev/null bs=128k count=781
```

> **Tip:** For this **dd** command, for the first time that it is run against rvpath0, the I/O must be read from disk and staged to the DS8000 cache. The second time that this **dd** command is run, the I/O is already in cache. Notice the shorter read time when we get an I/O cache hit.

If any of these LUNs are on ranks that are also used by another application, you see a variation in the throughput. If there is a large variation in the throughput, you need to give that LUN back to the storage administrator; trade for another LUN. You want all your LUNs to have the same performance.

If everything looks good, continue with the configuration of volume groups and logical volumes.

Verify the logical volumes

The next time that you stop and look at how your DS8000 storage performs is after the logical volumes are created. Remember that after volume groups and logical volumes are created, it is a disastrous idea to use the **dd** command to perform sequential writes to the raw vpath device, so do *not* use the **dd** command to perform sequential writes to the raw vpath device. It is a great idea to create a temporary raw logical volume on the vpaths to use for testing:

1. Put the **nmon** monitor up for a quick check on I/O throughput performance and vpath balance.
2. Test the sequential read speed on every raw logical volume device, if practical, or at least a decent sampling if you have too many to test each one. The following command is an example of the command that is run against a logical volume called 38glv. Perform this test against all your logical volumes to ensure that they each yield the *same results:*

   ```
   time dd if=/dev/r38glv of=/dev/null bs=128k count=781
   ```

3. Use the **dd** command without the **time** or **count** options to perform sequential reads and writes against all your logical volumes, raw devices, or block devices. Watch **nmon** for the Mb/s and IOPS of each LUN. Monitor the adapter. Look at the following characteristics:
   – Performance is the same for all the logical volumes.
   – Raw logical volumes devices (/dev/rlvname) are faster than the counterpart block logical volume devices (/dev/lvname) as long as the blocksize specified is more than 4 KB.
   – Larger blocksizes result in higher MB/s but reduced IOPS for raw logical volumes.
   – The blocksize does not affect the throughput of a block (not raw) logical volume, because, in AIX, the LVM imposes an I/O blocksize of 4 KB. Verify this size by running the **dd** command against a raw logical volume with a blocksize of 4 KB. This performance is the same as running the **dd** command against the non-raw logical volume.
   – Reads are faster than writes.
   – With inter-disk logical volumes, **nmon** does not report that all the LUNs have input at the same time, as with a striped logical volume. This result is normal and has concerns the **nmon** refresh rate and the characteristics of inter-disk logical volumes.
4. Ensure that the UNIX errorlog is clear of storage-related errors.

Verify the filesystems and characterize performance

After the filesystems are created, it is a good idea to take time to characterize and document the filesystem performance. A simple way to test sequential write/read speeds for filesystems is to time how long it takes to create a large sequential file and then how long it takes to copy the same file to /dev/null. After creating the file for the write test, you need to be careful that the file is not still cached in host memory, which invalidates the read test because the data comes from RAM instead of disk. This **lmktemp** command, which is used next, creates a file and you control the size of the file. It does not appear to be supported by any AIX documentation and therefore might disappear in future releases of AIX. We show examples of the tests:

- A simple sequential write test:

```
cd /interdiskfs
time lmktemp 2GBtestfile 2000M
real    0m18.83s
user    0m0.15s
sys     0m18.68s
```

Divide 2000/18.83 seconds = 107 MB/s sequential write speed.

- Sequential read speed:

  `cd /`

  `umount /interdiskfs` (this command flushes the file from the operating system (jfs, jfs2) memory)

  `mount /interdiskfs`

  `cd -` (cd back to the previous directory, /interdiskfs)

  `time dd if=/interdiskfs/2GBtestfile of=/dev/null bs=128k`

  ```
  real    0m11.19s
  user    0m0.39s
  sys     0m10.31s
  ```

  Divide 2000/11.19 seconds = 178 MB/s read speed.

  Now that the DS8000 cache is primed, run the test again. When we ran the test again, we got 4.08 seconds. Priming the cache is a good idea for isolated application read testing. If you have an application, such as a database, and you perform several isolated fixed reads, ignore the first run and measure the second run to take advantage of read hits from the DS8000 cache, because these results are a more realistic measurement of how the application performs.

For HP-UX, use the **`prealloc`** command instead of **`lmktemp`** for AIX to create large files. For Oracle Solaris, use the **`mkfile`** command.

> **Tip:** The **`prealloc`** command for HP-UX and the **`lmktemp`** command for AIX have a 2 GB size limitation. Those commands are not able to create a file greater than 2 GB in size. If you want a file larger than 2 GB for a sequential read test, concatenate a couple of 2 GB files.

10

Performance considerations for Microsoft Windows servers

This chapter describes performance considerations for supported Microsoft Windows servers attached to the IBM System Storage DS8700 or DS8800. In the context of this chapter, the term *Windows servers* refers to native servers as opposed to Windows servers that run as guests on VMware. You can obtain the most current list of supported Windows servers (at the time of writing this book) from the IBM System Storage Interoperation Center:

http://www.ibm.com/systems/support/storage/ssic/interoperability.wss

Disk throughput and I/O response time for any server connected to a DS8000 are affected by the workload and configuration of the server and the DS8000, data layout and volume placement, connectivity characteristics, and the performance characteristics of the DS8000. While the health and tuning of all of the system components affect the overall performance management and tuning of a Windows server, this chapter limits its discussion to the following topics:

- General Windows performance tuning
- I/O architecture overview
- Filesystem
- Volume management
- Multipathing and the port layer
- Host bus adapter (HBA) settings
- Windows Server 2008 I/O enhancements
- I/O performance measurement
- Problem determination
- Load testing

10.1 General Windows performance tuning

Microsoft Windows Server 2003 and Windows Server 2008 are largely self-tuning. Typically, leaving the system defaults is reasonable from a performance perspective. In this section, we describe the general considerations for improving the disk throughput or response time for either a file server or a database server:

- The Windows environment is memory hungry and benefits from having as much memory as possible. Provide enough memory for database, application, and filesystem cache. As a rule, database buffer hit ratios must be greater than 90%. Increasing the cache hit ratios is the most important tuning consideration for the I/O performance of databases, because it reduces the amount of physical I/O that is required.
- Monitor file fragmentation regularly, which is important for database files and Microsoft Exchange database files. After starting with the fixed block allocation, a database file can become fragmented later. This fragmentation increases reading and writing response time dramatically, because fragmented files are cached badly by filesystem cache.
- Schedule processes that are processor-intensive, memory-intensive, or disk-intensive during after-hours operations. Examples of these processes are virus scanners, backups, and disk fragment utilities. These types of processes must be scheduled to run when the server is least active.
- Optimize the amount and priority of the services running. It is better to have most of the secondary services started manually than running all the time. Critical application-related services must be planned with the highest priority and resource allocation, that is, you must start main services first and then add secondary services one-by-one if needed. For example, the following services can be stopped: Alerter, Clipbook Server, Computer Browser, Messenger, Network dynamic data exchange (DDE), Object Linking and Embedding (OLE) Schedule, and spooler.
- Follow the Microsoft recommendation that large dedicated file servers or database servers are configured as backup domain controllers (BDC) due to the overhead associated with the *netlogon* service.
- Optimize the paging file configuration. The paging file fragments if there is not enough contiguous hard disk space to hold the entire paging file. By default, the operating system configures the paging file to allocate space for itself dynamically. During the dynamic resizing, the file can end up fragmented because of a lack of contiguous disk space. You can obtain more details about the paging file in 10.4.3, “Paging file” on page 401.
- Set the New Technology File System (NTFS) log file to 64 MB to reduce the frequency of the NTFS log file expansion. Log file expansion is costly, because it locks the volume for the duration of the log file expansion operation. Set the NTFS log file to 64 MB through the following command-line entry: **`chkdsk x: /L:65536`**

For detailed instructions about these tuning suggestions, see the following publications:

- *Tuning IBM System x Servers for Performance*, SG24-5287
- *Tuning Windows Server 2003 on IBM System x Servers*, REDP-3943

Also, see these websites:

http://download.microsoft.com/download/2/8/0/2800a518-7ac6-4aac-bd85-74d2c52e1ec6/tuning.doc

http://msdn.microsoft.com/en-us/windows/hardware/gg463394

http://msdn.microsoft.com/en-us/windows/hardware/gg463392 http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/Perf-tun-srv.docx

10.2 I/O architecture overview

At a high level, the Windows I/O architecture is similar to the I/O architecture of most Open Systems. Figure 10-1 shows a generic view of the I/O layers and examples of how they are implemented.

	Layer	Implementations
	Application/DB	SQL Server
I/O Stack	I/O Subsystem	Microsoft I/O Manager
I/O Stack	Filesystems	NTFS, FAT, VxFS
I/O Stack	Volume Manager	Logical Disk Manager
I/O Stack	Port - Multipathing	Storport + miniport + DSM
I/O Stack	FC adapters	Qlogic
	Fabric	Brocade
	Storage Array	DS8000

Figure 10-1 Windows I/O stack

To initiate an I/O request, an application issues an I/O request by using one of the supported I/O request calls. The I/O manager receives the application I/O request and passes the I/O request packet (IRP) from the application to each of the lower layers that route the IRP to the appropriate device driver, port driver, and adapter-specific driver.

Windows server filesystems can be configured as file allocation table (FAT), FAT32, or NTFS. The file structure is specified for a particular partition or logical volume. A logical volume can contain one or more physical disks. All Windows volumes are managed by the Windows Logical Disk Management utility.

For additional information relating to the Windows Server 2003 and Windows Server 2008 I/O stacks and performance, see the following documents:

- http://download.microsoft.com/download/5/6/6/5664b85a-ad06-45ec-979e-ec4887d715eb/Storport.doc
- http://download.microsoft.com/download/5/b/9/5b97017b-e28a-4bae-ba48-174cf47d23cd/STO089_WH06.ppt

10.3 Windows Server 2008 I/O Manager enhancements

There are several I/O performance enhancements to the Windows Server 2008 I/O subsystem. We summarize them in this section.

I/O priorities

The Windows Server 2008 I/O subsystem provides a mechanism to specify I/O processing priorities. Windows primarily uses this mechanism to prioritize critical I/O requests over background I/O requests. API extensions exist to provide application vendors file-level I/O priority control. The prioritization code has some processing overhead and can be disabled for disks that are targeted for similar I/O activities, such as databases.

I/O completion and cancellation

The Windows Server 2008 I/O subsystem provides a more efficient way to manage the initiation and completion of I/O requests, resulting in a reduced number of context switches, lower CPU utilization, and reduced overall I/O response time.

I/O request size

The maximum I/O size was increased from 64 KB per I/O request in Windows Server 2003 to 1024 KB in Windows Server 2008. For large sequential workloads, such as file shares or backups, this increase can improve the disk throughput.

You can obtain additional information at this website:

http://support.microsoft.com/kb/2160852

http://blogs.technet.com/b/askperf/archive/2008/02/07/ws2008-memory-management-dynamic-kernel-addressing-memory-priorities-and-i-o-handling.aspx

10.4 Filesystem

A *filesystem* is a part of the operating system that determines how files are named, stored, and organized on a volume. A filesystem manages files, folders, and the information needed to locate and access these files and folders for local or remote users.

10.4.1 Windows filesystem overview

Microsoft Windows 2000 Server, Windows Server 2003, and Windows Server 2008 all support the FAT/FAT32 filesystem and NTFS. However, we advise that you use NTFS for the following reasons:

- NTFS provides considerable performance benefits by using a B-tree structure as the underlying data structure for the filesystem. This type of structure improves performance for large filesystems by minimizing the number of times that the disk is accessed, which makes it faster than FAT/FAT32.
- NTFS provides significant scalability over FAT/FAT32 in the maximum volume size. In theory, the maximum file size is 2^{64}. However, on a Windows 32-bit system that uses 64 KB clusters, the maximum volume size is 256 TB, and the maximum file size is 16 TB.
- NTFS provides recoverability through a journaled filesystem functionality.
- NTFS fully supports the Windows NT security model and supports multiple data streams. No longer is a data file a single stream of data. Additionally, under NTFS, a user can add user-defined attributes to a file.

10.4.2 NTFS guidelines

Follow these guidelines:

Allocation Block allocation size must be selected based on the application recommendations and preferred practices. Allocations with 64 KB blocks work in most of the cases and improve the efficiency of the NTFS filesystem. This allocation reduces the fragmentation of the filesystem and reduces the number of allocation units required for large file allocations.

Defragment disks Over time, files become fragmented in noncontiguous clusters across disks, and disk response time suffers as the disk head jumps between tracks to seek and reassemble the files when they are required. We advise regularly defragmenting volumes.

Block alignment Use `diskpar.exe` for Windows 2000 Server servers and use `diskpart.exe` for Windows Server 2003 servers to force sector alignment. Windows Server 2008 automatically enforces a 1 MB offset for the first sector in the partition, which negates the need for `diskpart.exe`. For additional information, see the following documents:

http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/Perf-tun-srv.docx

http://support.microsoft.com/kb/929491

Important: NTFS filesystem compression seems to be the easiest way to increase the amount of the available capacity. However, it is strictly not advised for use in enterprise environments. Filesystem compression consumes too much of disk and processor resources and dramatically increases response times of reading and writing. For better capacity utilization, consider the DS8000 Thin Provisioning technology and IBM Data Deduplication technologies.

Start sector offset: The start sector offset must be 256 KB due to the stripe size on the DS8000. Workloads with small, random I/Os (<16 KB) are unlikely to experience any significant performance improvement from sector alignment on the DS8000 logical unit numbers (LUNs).

10.4.3 Paging file

Windows servers use paging file technology as the extension to the existing random access memory. We use the term *paging file*, because the operating system writes memory pages there when main memory is full and it keeps them for future reading. Because the paging file is on the disk drives, it might degrade the performance of the entire system if the paging file is maintained improperly. Following these basic rules might help you to avoid this situation:

- The paging file must be on physically separate disk drives and those drives must have no other activity. The paging file must be on physically separate disk drives for monitoring.
- The paging file must have a static size, which eliminates the necessity of its growth to avoid the fragmentation of the paging file. The paging file fragments if there is not enough contiguous hard disk space to hold the entire paging file. So, plan paging file volumes for enough capacity.
- The activity on the paging file depends on the amount of memory in the system. Consider more memory in the system if the activity and usage of the paging file are high.
- Monitor the activity on the paging file volumes regularly. The monitored values must be included in the alert settings and generate alerts if the values are exceeded.
- The access pattern to the paging file is highly unpredictable. It can be random with the memory page size (typically 4 KB) or sequential with blocks up to 1 MB. Read/write balance is close to 50%. So, if you are expecting high activity on the paging file, it is better to put it on a separate RAID 10 rank.

- For systems with a large amount of memory, you might be tempted to disable the paging file, which is *not advised*. The paging file is used by all services and device drivers in the operating system and disabling it can lead to a system crash. If the system has enough memory, it is better to have a paging file of a minimum required static size.

For additional information about the paging file, see the following links:

http://support.microsoft.com/kb/889654

http://technet.microsoft.com/en-us/magazine/ff382717.aspx

10.5 Volume management

Volume managers provide an additional abstraction layer between the physical resources and the filesystem and allow administrators to group multiple physical resources into a single volume. There are two volume managers that are supported for the Microsoft Windows server environment:

- Microsoft Logical Disk Manager
- Veritas Volume Manager

10.5.1 Microsoft Logical Disk Manager (LDM)

The LDM provides an abstraction layer between the NTFS filesystem layer and the physical storage layer with the following functionality:

- Support of dynamic and basic volumes.
- Support of concatenated, striped (Raid 0), mirrored (Raid 1), and RAID 5 volumes.
- Dynamic expansion of the dynamic volumes. Volumes are expanded with disk system logical disk capacity increase.
- Support for Microsoft Cluster Service (might require additional hardware and software).

10.5.2 Veritas Volume Manager (VxVM)

Veritas Storage Foundation for Windows provides the Veritas Volume Manager (VxVM), a comprehensive solution for managing Windows server volumes.

VxVM includes the following features:

- Support of concatenated, striped (Raid 0), mirrored (Raid 1), mirrored striped (Raid 1+0), and RAID 5 volumes
- Dynamic expansion of all volume types
- Dynamic MultiPathing (DMP) as an optional component
- Support for Microsoft Cluster Service (might require additional hardware and software)
- Support for up to 256 physical disks in a dynamic volume

For additional information that relates to VxVM, see Veritas Storage Foundation High Availability for Windows:

http://www.symantec.com/business/storage-foundation-for-windows

Important: Despite the logical volume manager functional benefits, we do not advise that you use any LVM striping in DS8000 Easy Tier and I/O Priority Manager environments. DS8800 and DS8700 systems offer improved algorithms and methods of managing the data and performance at the lower level and do not require any additional volume management. Combined usage of these technologies might lead to unexpected results and performance degradation that makes the search for bottlenecks impossible.

10.5.3 Determining volume layout

From a performance perspective, there are two approaches to volume layout. The first approach is to spread everything everywhere. The second approach is to isolate volumes based on the nature of the workload or application. We discussed these approaches in great detail in Chapter 4, "Logical configuration performance considerations" on page 87. Today, we suggest the consolidation of these approaches with the benefits of Easy Tier v3 and I/O Priority Manager in DS8800 or DS8700. From the isolation on the rank and extent level, you switch to isolation on the application level and go deeper with priority management and micro-tiering. There are several areas to consider when planning the volume layout:

- Workloads that are highly sensitive to increases of I/O response time are better to isolate. Today, we do not suggest isolation on the rank level. We suggest isolation on the extent pool level and the volume level.
- Isolation can be done also on the functional group of the application level. For example, you might have several mail server instances that use their own databases and locate them on one or several extent pools that maintain different levels of I/O priority for each of them. This functional group of mail servers is isolated from the other applications but shares the cache memory and I/O ports.
- "Home tier" for all hybrid pool volumes is the Enterprise tier[1]. The goal of the tiering is to keep it performing well by moving hot extents to the solid-state drive (SSD) tier and free enough for new volumes by moving cold extents to the NearLine tier.
- Easy Tier technology and I/O Priority Manager technologies are unaware of the work of each other. It is likely not possible to make extents "hotter" or "colder" by setting different priority levels. Easy Tier looks for I/O density, and priority management looks for the quality of service (QoS).
- Skew factor is the main determinant whether to use SSDs. Low skew factor workloads might benefit more from the Micro-tier option than SSDs. The Micro-tier option is more attractive from the cost point of view.
- Micro-tier can be useful for high performance applications but not for capacity-oriented applications. By using 15K rpm and 10K rpm drives for these applications, you might improve the cost of 1 GB of storage.
- For applications that consistently demand high bandwidth for I/O throughput, consider Enterprise 10K rpm drives and NearLine drives to keep the level of performance high and the Enterprise drive space optimized.

Table 10-1 on page 404 demonstrates examples of typical Windows server workloads, categorizes them as potential candidates for hybrid pool usage, and describes the goals of priority management.

[1] Hybrid pools in this context means pools with Enterprise disks.

Table 10-1 Windows application tier requirements and priorities

Application type	Tier levels	Priority level	Goals of tiering
Database Server (online transaction processing (OLTP))	SSD+ENT+NL SSD+ENT(15,10)+NL	Highest	Maintain a low response time and the highest priority for business-critical applications. Keep ENT level free enough for growth and benefits from Micro-tiering.
Database server (data mining, test and development, and data warehousing)	SSD+ENT+NL ENT(15,10)+NL	High Medium	Micro-tiering is favorable for these applications. Non-business-critical applications might still require low response times as the backup option for the main applications, but they might also require a decreased I/O priority level.
Web servers	ENT+NL	Medium High Low	Keep ENT level free and high performing for the database part of the application. NL tier is used for storing the data. If it is not a business-critical application, it can have a low priority level on the requests.
Mail servers	ENT(15,10)+NL ENT+NL	High Medium	Keep high priority to the DB part requests and maintain low read and write request response time. Keep ENT tier free enough to expand.
File servers	NL mostly ENT(10)+NL	Low	Try to avoid the use of the ENT tier and keep the I/O requests at a low priority.

Suggestions: The example applications listed in Table 10-1 are examples only and not specific rules.

Volume planning for Windows applications

You must determine the number of drive types in each hybrid pool for each application group. It is not easily defined and predicted. To be accurate, you must consider many factors, such as skew factor, application behavior, access pattern, and active/passive user density. Also, workload must be evenly spread across all the DS8000 resources to eliminate or reduce contention. Follow these rules:

- Collect the workload statistics when possible to be able to predict the behavior of the systems and analyze the skew factor.
- Database applications often have a high skew factor and benefit from the SSDs in the pool. Plan at least 10% of pool capacity for SSDs, which might be increased up to 30%. Nearline drives can help to preserve free extents on the Enterprise tier. Plan to have at least 30% of the capacity in nearline drives.
- If you do not expect cold extents to appear in the database application and plan to have a heavy workload pattern, use the Micro-tiering option and have 15K rpm drives and 10K

rpm drives in the Enterprise tier. This plan frees the 15K rpm drives from the large block sequential workload so that the 15K rpm drives provide lower response times. The ratio is about two thirds for the 15K rpm drives.

- Electronic mail applications have a database part for indexing data for rapid random access and the data is stored for later access. This application can benefit from SSDs, but the amount of SSDs must not be more than 2% - 5% due to the high write ratio, which can be up to 50% - 60%. You can benefit from the micro-tiering option with 15K and 10K Enterprise drives. Mail servers can have many large block sequential writes and 10K drives work well. Plan for at least 90% of capacity on Enterprise drives with 50% of 10K rpm drives. The Nearline tier provides a plan to keep the archives in the same instances of the mail servers.
- Web servers are likely to keep the data for a long time and cache the data often, so the size of the Enterprise tier must be about 5% to provide random access if it happens. Most of the storage must be nearline drives. Use Thin Provisioning technology for the initial data allocation, which helps to keep the Enterprise extents free at the initial data allocation.
- File servers handle the cold data and have a sequential access pattern. The performance of the file servers depends more on the performance of the network adapters and not the disk subsystem. Plan to have 99% of the capacity on nearline drives and thin-provisioned.
- Run the I/O Priority Manager on both the inter-application and intra-application sides. You need the QoS of the overall system and the scope of the resources that are dedicated to each application.
- Remember the device adapter (DA) pair limits when planning the extent pools.
- Include ranks from both odd and even CECs in one extent pool.
- Monitor the resource load regularly to avoid high peaks and unexpected delays.

The prior approach of workload isolation on the rank level might work for the low skew factor workloads or some specific workloads. Also, you can use this approach if you are confident in planning the workload and volume layout.

10.6 Multipathing and the port layer

The multipathing, storage port, and adapter drivers exist in three separate logical layers; however, they function together to provide access and multipathing facilities to the DS8000. Multipathing provides redundancy and scalability. Redundancy is facilitated through multiple physical paths from the server to a DS8000. Scalability is implemented by allowing the server to have multiple paths to the storage and to balance the traffic across the paths. There are several methods available for configuring multipathing for Windows servers attached to an IBM DS8800 or DS8700:

- Windows Server 2003

  IBM SDD, IBM Subsystem Device Driver Device Specific Module (SDDDSM), and Veritas DMP
- Windows Server 2008

  IBM SDDDSM and Veritas DMP

On Windows servers, the implementations of multipathing rely on either native multipathing (Microsoft Multipath I/O (MPIO) + Storport driver) or non-native multipathing and the Small Computer System Interface (SCSI) port driver or SCSIport. The following sections describe the performance considerations for each of these implementations.

10.6.1 SCSIport scalability issues

Microsoft originally designed the SCSIport storage driver for parallel SCSI interfaces. IBM SDD and older versions of Veritas DMP still rely on it. Host bus adapters (HBA) miniport device drivers compliant with the SCSIport driver have a number of performance and scalability limitations. The following section summarizes the key scalability issues with this architecture:

- Adapter limits

  SCSIport is limited to 254 outstanding I/O requests per adapter regardless of the number of physical disks associated with the adapter. SCSIport does not provide a means for managing the queues in high loads. One of the possible results of this architecture is that one highly active device can dominate the adapter queue, resulting in latency for other non-busy disks.

- Serialized I/O requests processing

  The SCSIport driver cannot fully take advantage of the parallel processing capabilities available on modern enterprise class servers and the DS8000.

- Elevated Interrupt Request Levels (IRQLs)

  There is a high probability that other higher priority processes might run on the same processors as the device interrupts, which on I/O-intensive systems can cause a significant queuing of interrupts, resulting in slower I/O throughput.

- Data buffer processing overhead

  The SCSIport exchanges physical address information with the miniport driver one element at a time instead of in a batch, which is inefficient, especially with large data transfers, and results in the slow processing of large requests.

10.6.2 Storport scalability features

In response to significant performance and scalability advances in storage technology, such as hardware RAID and high performing storage arrays, Microsoft developed a new storage driver called *Storport*. The architecture and capabilities of this driver address most of the scalability limitations that exist in the SCSIport driver.

The Storport driver offers these key features:

- Adapter limits removed

  There are no adapter limits. There is a limit of 254 requests queued per device.

- Improvement in I/O request processing

  Storport decouples the StartIo and Interrupt processing, enabling parallel processing of start and completion requests.

- Improved IRQL processing

  Storport provides a mechanism to perform part of the I/O request preparation work at a low priority level, reducing the number of requests queued at the same elevated priority level.

- Improvement in data buffer processing

  Lists of information are exchanged between the Storport driver and the miniport driver as opposed to single element exchanges.

- Improved queue management

  Granular queue management functions provide HBA vendors and device driver developers the ability to improve management of queued I/O requests.

For additional information about the Storport driver, see the following document:

http://download.microsoft.com/download/5/6/6/5664b85a-ad06-45ec-979e-ec4887d715eb/Storport.doc

10.6.3 Subsystem Device Driver

The IBM Subsystem Device Driver (SDD) provides path failover/failback processing for the Windows server attached to the IBM System Storage DS8000. SDD relies on the existing Microsoft SCSIport system-supplied port driver and HBA vendor-provided miniport driver.

It also provides I/O load balancing. For each I/O request, SDD dynamically selects one of the available paths to balance the load across all possible paths.

To receive the benefits of path balancing, ensure that the disk drive subsystem is configured so that there are multiple paths to each LUN. By using multiple paths to each LUN, you can benefit from the performance improvements from SDD path balancing. This approach also prevents the loss of access to data in the event of a path failure.

We describe the Subsystem Device Driver in further detail in “Subsystem Device Driver” on page 317.

10.6.4 Subsystem Device Driver Device Specific Module

Subsystem Device Driver Device Specific Module (SDDDSM) provides multipath I/O support based on Microsoft MPIO technology for Windows Server 2003 and Windows Server 2008 servers. A Storport-based driver is required for the Fibre Channel adapter. SDDDSM uses a device-specific module designed to provide support of specific storage arrays. The DS8000 supports most versions of Windows Server 2003 and Windows Server 2008 servers as specified at the System Storage Interoperation Center (SSIC):

http://www.ibm.com/systems/support/storage/config/ssic/displayesssearchwithoutjs.wss?start_over=yes

You can obtain additional information about SDDDSM in the *SDD User’s Guide*:

http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S7000303

In Windows Server 2003, the MPIO drivers are provided as part of the SDDDSM package. On Windows Server 2008, they ship with the OS.

> **SDDDSM:** For non-clustered environments, we suggest that you use SDDDSM for its performance and scalability improvements.

10.6.5 Veritas Dynamic MultiPathing (DMP) for Windows

For enterprises with significant investment in Veritas software and skills, Veritas provides an alternative to the multipathing software provided by IBM. Veritas relies on the Microsoft implementation of MPIO and Device Specific Modules (DSMs), which rely on the Storport driver. This implementation is not available for all versions of Windows. See the System Storage Interoperation Center (SSIC) for your specific hardware configuration:

http://www.ibm.com/systems/support/storage/config/ssic/displayesssearchwithoutjs.wss?start_over=yes

10.7 Host bus adapter (HBA) settings

For each HBA, there are BIOS and driver settings that are suitable for connecting to your DS8000. Configuring these settings incorrectly can affect performance or cause the HBA to not work correctly.

To configure the HBA, see the *IBM System Storage DS8700 and DS8800 Introduction and Planning Guide,* GC27-2297-07. This guide contains detailed procedures and recommended settings. You also need to read the readme file and manuals for the driver, BIOS, and HBA.

Obtain a list of supported HBAs, firmware, and device driver information at this website:

http://www.ibm.com/systems/support/storage/config/ssic/displayesssearchwithoutjs.wss?start_over=yes

> **Newer versions:** When configuring the HBA, we strongly advise that you install the newest version of driver and the BIOS. The newer version includes more effective function and problem fixes so that the performance or reliability, availability, and service (RAS) can improve.

10.8 I/O performance measurement

Regular performance monitoring and measurement are critical for the normal daily work of enterprise environments. This section covers methods, tools, and approaches to performance management. Figure 10-2 demonstrates the I/O layer on the left side and an example of its corresponding method of control on the right side.

	Layer	Measurement Tools
	Application/DB	Perfmon App. Specific
I/O Stack	I/O Manager	Event Trace
I/O Stack	Filesystems	Event Trace
I/O Stack	Volume Manager	Perfmon Disk
I/O Stack	Port - Multipathing	SDD/SDDDSM/DMP
I/O Stack	FC adapters	Adapter Vendor Tools
	Fabric	Fabric
	DS8000	TPC

Figure 10-2 I/O layer and example implementations

- The application layer is monitored and tuned with the application-specific tools and metrics available for monitoring and analyzing application performance on Windows servers. Application-specific objects and counters are outside the scope of this book.
- The I/O Manager and filesystem levels are controlled with the built-in Windows tool, which is available in Windows Performance Console (**`perfmon`**).
- The volume manager level can be also monitored with **`perfmon`**. However, we do not suggest that you use any logical volume management in Windows servers.
- Fibre Channel port level multipathing is monitored with the tools provided by the multipathing software: IBM SDD, SDDPCM, or Veritas DMP drivers.

- The Fibre Channel adapter level can be monitored with the adapter-specific original software that is provided by each vendor. See the following sites for the support software of the adapters that are compatible with IBM DS8800 and DS8700:

  http://www.brocade.com/services-support/index.page

  http://www.emulex.com/support.html

  http://solutions.qlogic.com/KanisaSupportSite/supportcentral/supportcentral.do?id=m1

- The SAN fabric level and the D8000 level are monitored with the IBM Tivoli Storage Productivity Center and the DS8000 built-in tools. Because Tivoli Storage Productivity Center provides more functions to monitor the DS8000 systems, we suggest that you use it for monitoring and analysis.

We describe several more complex approaches to performance monitoring:

- Overview of I/O metrics
- Overview of **perfmon**
- Overview of logs
- Mechanics of logging
- Mechanics of exporting data
- Collecting multipath data
- Correlating the configuration and performance data
- Analyzing the performance data

10.8.1 Key I/O performance metrics

In Windows servers, there are two kinds of disk counters: *PhysicalDisk* object counters and *LogicalDisk* object counters. PhysicalDisk object counters are used to monitor single physical disks and are enabled by default. LogicalDisk object counters are used to monitor logical disks or software RAID arrays created with Logical Disk Manager. LogicalDisk object counters are enabled by default on Windows Server 2003 and Windows Server 2008 servers.

PhysicalDisk object counters: When attempting to analyze disk performance bottlenecks, always use physical disk counters to identify performance issues with individual DS8000 LUNs. We do not use LogicalDisk counters in this book.

Table 10-2 describes the key I/O-related metrics that are reported by **perfmon**.

Table 10-2 Performance monitoring counters for PhysicalDisk and other objects

Counter	Normal values	Critical values	Description
%Disk Time	~ 70 - 90%	Depends on the situation	Percentage of elapsed disk serviced read or write requests.
Average Disk sec/Read	5 - 15 ms	16 - 20 ms	The average amount of time in seconds to complete an I/O read request. These results are end-to-end disk response times.
Average Disk sec/Transfer	1 - 15 ms	16 - 20 ms	The average amount of time in seconds to complete an I/O request. These results are end-to-end disk response times.

Counter	Normal values	Critical values	Description
Average Disk sec/Write	0 - 3 ms	5 - 10 ms	The average amount of time in seconds to complete an I/O write request. These results are end-to-end disk response times. Write requests must be services with cache. If not, there is a problem.
Disk Transfers/sec	According to the workload	Close to the limits of volume, rank, and extent pool	The momentary number of disk transfers per second during the collection interval.
Disk Reads/sec	According to the workload	Close to the limits of volume, rank, and extent pool	The momentary number of disk reads per second during the collection interval.
Disk Bytes/sec	According to the workload	Close to the limits of volume, rank, and extent pool	The momentary number of bytes per second during the collection interval.
Disk Read Bytes/sec	According to the workload	Close to the limits of volume, rank, and extent pool	The momentary number of bytes read per second during the collection interval.
Current Disk Queue Length	5 - 100, depends on the activity	1000>	Indicates the momentary number of read and write I/O requests waiting to be serviced. *When you have I/O activity, Queue Length is not zero.*
Object Paging file, counter%Usage	0 - 1	40% and more	The amount of paging file instance in use in a percentage.
Current Disk Queue Length	5 - 100, depends on the activity	1000>	Indicates the momentary number of read and write I/O requests waiting to be serviced. *When you have I/O activity, Queue Length is not zero.*
Object Processor, counter% User Time	80 - 90%	1 - 10%	It is the amount of time the processor spends servicing user requests, that is, user mode. Applications are considered user requests, too.

Rules

We provide the following rules based on our field experience. Before using these rules for anything specific, such as a contractual service-level agreement (SLA), you must carefully analyze and consider these technical requirements: disk speeds, RAID format, workload variance, workload growth, measurement intervals, and acceptance of response time and throughput variance. We suggest these rules:

- Write and read response times in general must be as specified in the Table 10-2 on page 409.
- There must be a definite correlation between the counter values; therefore, the increase of one counter value needs to lead to the increase to the others connected to it. For example, the increase of the Transfers/sec counter leads to the increase of the Average sec/Transfer

counter, because the increased number of IOPS leads to an increase in the response time of each I/O.

- If one counter has a high value and the related parameter value is low, pay attention to this area. It can be a hardware or software problem or a bottleneck.
- A Disk busy counter close to 100% does not mean that the system is out of its disk performance capability. The disk is busy with I/Os. Problems occur when there are 100% disk busy counters with close to zero counters of the I/Os at the same time.
- Shared storage environments are more likely to have a variance in disk response time. If your application is highly sensitive to variance in response time, you need to isolate the application at either the processor complex, device adapter (DA), or rank level.
- With the **`perfmon`** tool, you monitor only the front-end activity of the disk system. To see the complete picture, monitor the back-end activity, also. Use the Tivoli Storage Productivity Center console and Storage Tier Advisor Tool (STAT).

> **Conversion to milliseconds:** By default, Windows provides the response time in seconds. To convert to milliseconds, you must multiply by 1000, which is automatic in the `perfmon-essmap.pl` script that is provided in Appendix E, “Post-processing scripts” on page 677.

10.8.2 Windows Performance console (perfmon)

The Windows Performance console (**`perfmon`**) is one of the most valuable monitoring tools available to Windows server administrators. It is commonly used to monitor server performance and to isolate disk bottlenecks. The tool provides real-time information about server subsystem performance. It also can log performance data to a file for later analysis. The data collection interval can be adjusted based on your requirements. The **`perfmon`** tool offers two ways to monitor performance:

- Real-time monitoring
- Monitoring with automatic data collection for a period

Monitoring disk performance in real time

Monitoring disk activity in real time permits you to view current disk activity on local or remote disk drives. The current, and not the historical, level of disk activity is shown in the chart. You cannot analyze the data for any performance problems, because the window size for real-time monitoring is about 2 minutes. See Figure 10-3 on page 412. The data is displayed for 1 minute and 40 seconds. It is indicated in the Duration box of the window.

Monitoring disk performance with data collection

If you want to determine whether excessive disk activity on your system is slowing performance, log the disk activity of the desired disks to a file, over a period that represents the typical use of your system. View the logged data in a chart and export it to a spreadsheet-readable file to see whether disk activity affects system performance. Collected data can be exported to electronic spreadsheet software for future analysis.

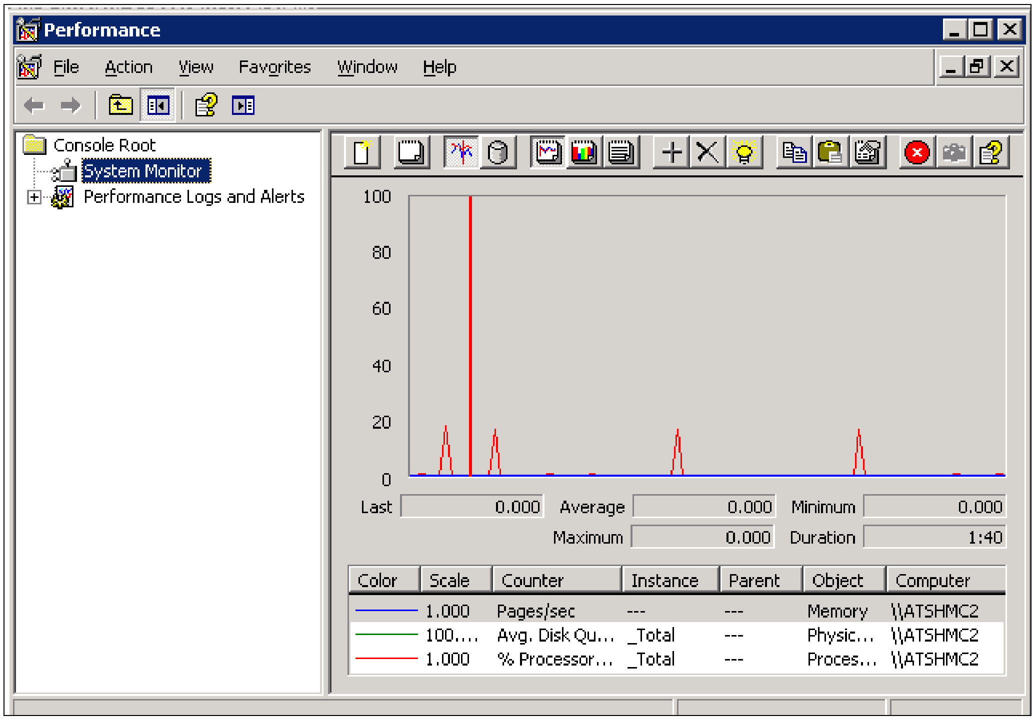


Figure 10-3 Main Performance console window in Windows Server 2003

The Performance console is a snap-in tool for the Microsoft Management Console (MMC). You use the Performance console to configure the System Monitor and Performance Logs and Alerts tools.

You can open the Performance console by clicking **Start** → **Programs** → **Administrative Tools** → **Performance** or by typing `perfmon` on the command line.

Windows Server 2008 perfmon

The Windows Server 2008 performance console (`perfmon`) has additional features, including these key new features:

Data Collector Sets Use to configure data collection templates for the collection of system and trace counters.

Resource Overview Shows a high-level view of key system resources, including CPU% total usage, disk aggregate throughput, network bandwidth utilization, and memory hard faults/sec (see Figure 10-4 on page 413).

Reports Integration of SPA functionality. This feature can quickly report on collected counter and trace data in a way that is both user friendly and that provides substantial detail.

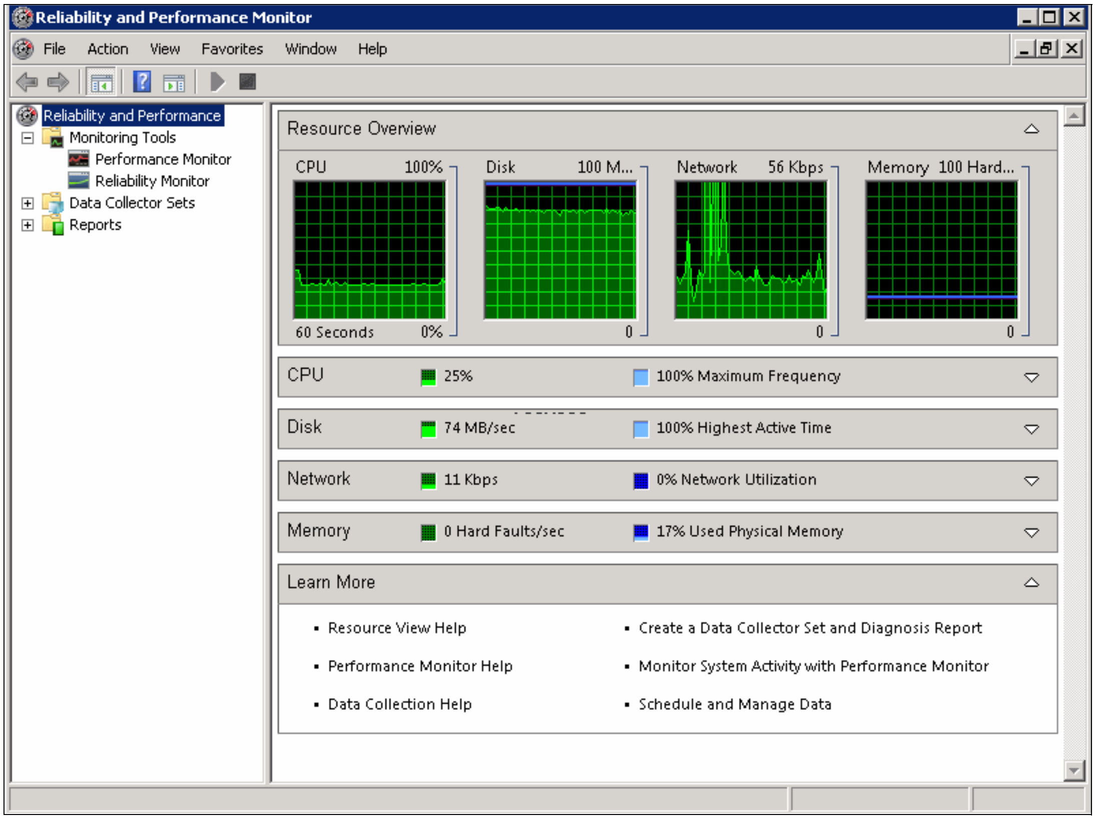


Figure 10-4 Windows Server 2008 perfmon console

With Windows Server 2003, you can open the Performance console by clicking **Start** → **Programs** → **Administrative Tools** → **Performance** or by typing `perfmon` on the command line.

10.8.3 Performance log configuration and data export

With large numbers of physical disks and long collection periods required to identify certain disk bottlenecks, it is impractical to use real-time monitoring. In these cases, disk performance data can be logged for analysis over extended periods. Instructions for collecting the necessary log data and exporting the data to a spreadsheet-readable format for both Windows Server 2003 and Windows Server 2008 are in Appendix D, "Microsoft Windows server performance log collection" on page 659. See these instructions for collecting the necessary performance data to perform analysis. The remaining sections assume that you collected performance data.

10.8.4 Collecting configuration data

You need the correct configuration data first so that you can easily and clearly understand which disk in the system correlates to which logical volume in the DS8000. We guide you step-by-step through the process of getting configuration data for one disk. You need the following tools:

- Microsoft Management Console with Computer Management snap-in

  Click **Start** → **Run** and type `mmc`. A Microsoft Management Console window opens. Open **%SystemRoot%\System32\compmngmt.msc** and you get the Computer Management snap-in.

- IBM SDDDSM Console

  Click **Start** → **Program** → **Subsystem Device Driver** → **Subsystem Device Driver Management.** A command-line window opens.

- IBM DS8000 command-line interface (DSCLI) console

  Open `%SystemRoot%\Program Files\IBM\dscli\dscli.exe`.

- Identify the disks that you have in the Disk Management of the Computer Management snap-in. See Figure 10-5.

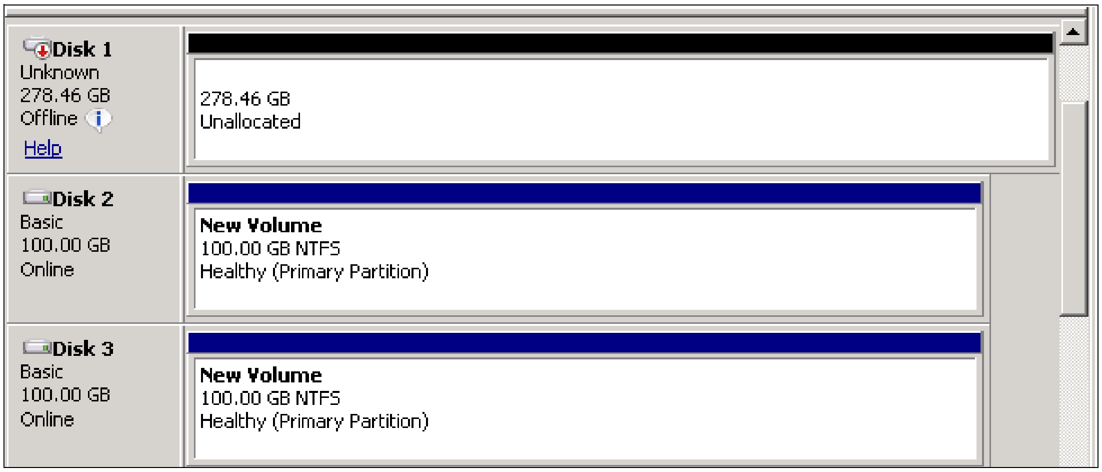


Figure 10-5 List of the volumes in MMC

Figure 10-5 shows several disks. To identify them, right-click the name and click the **Properties** option on the right-click menu on each of them. Disks from the DS8000 show IBM 2107 in the properties, which is the definition of the DS8000 machine-type. So, in this example, our disk from the DS8000 is Disk 2. See Figure 10-6 on page 415.

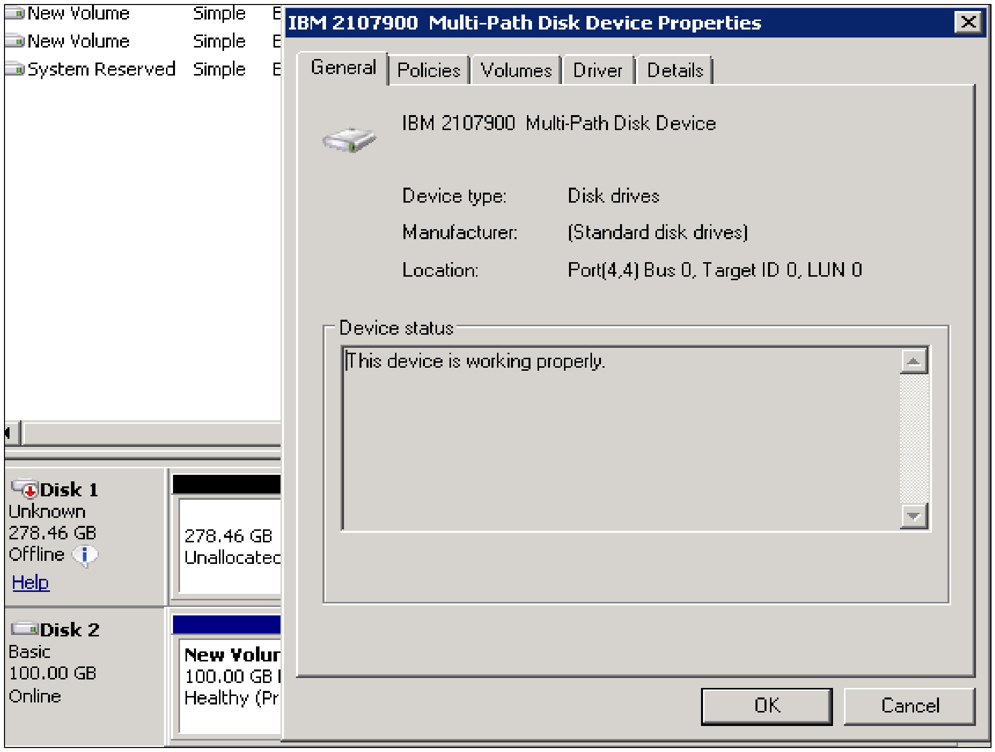


Figure 10-6 Properties of the DS8800 logical volume

Multi-Path Disk Device means that you are running SDD. You can also check for SDD from the Device Manager option of the Computer Management snap-in. See Figure 10-7.

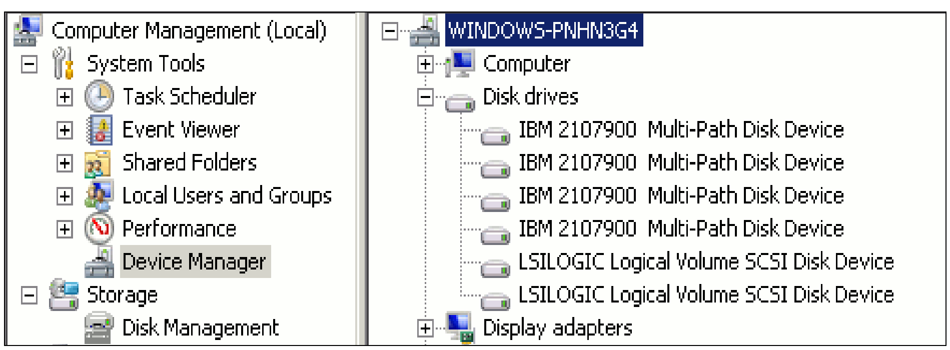


Figure 10-7 SDD multipathing is running

In Figure 10-7, you see several devices and one SDD that is running.

Use the **datapath query device** command to show the disk information in the SDDDSM console (Example 10-1).

Example 10-1 The datapath query device command

```
c:\Program Files\IBM\SDDDSM>datapath query device
Total Devices : 4
DEV#:   0  DEVICE NAME: Disk2 Part0  TYPE: 2107900    POLICY: OPTIMIZED
SERIAL: 75V1818601
============================================================================
Path#             Adapter/Hard Disk          State  Mode       Select     Errors
    0       Scsi Port4 Bus0/Disk2 Part0      OPEN   NORMAL       1520          0
```

```
   1     Scsi Port4 Bus0/Disk2 Part0     OPEN   NORMAL       1507          0
   2     Scsi Port2 Bus0/Disk2 Part0     OPEN   NORMAL         52         24
   3     Scsi Port2 Bus0/Disk2 Part0     OPEN   NORMAL        165         78
```

Figure 10-1 on page 399 shows the disk information. Disk 2 has serial number 75V1818601. The last four digits are the volume ID, which is 8601 in this example. This disk is connected with two FC ports of the FC adapter.

List the worldwide port names (WWPNs) of the ports with the **datapath query wwpn** command (Figure 10-2 on page 408).

Example 10-2 Listing the FC port WWPNs

```
c:\Program Files\IBM\SDDDSM>datapath query wwpn
        Adapter Name    PortWWN
        Scsi Port2:     2100001B32937DAE
        Scsi Port3:     2101001B32B37DAE
        Scsi Port4:     2100001B3293D5AD
        Scsi Port5:     2101001B32B3D5AD
```

In Figure 10-2 on page 408, there are two WWPNs for Port 2 and Port 4.

Identify the disk in the DS8000 with the DSCLI console (Figure 10-3 on page 412).

Example 10-3 Listing the volumes in the DS8800 DSCLI console (output truncated)

```
dscli> lsfbvol
Name ID   accstate datastate configstate deviceMTM datatype extpool cap (2^30B) cap (10^9B) cap (blocks)
===========================================================================================================
-                8600 Online   Normal    Normal      2107-900  FB 512   P4 100.0          -    209715200
-                8601 Online   Normal    Normal      2107-900  FB 512   P4 100.0          -    209715200
-                8603 Online   Normal    Normal      2107-900  FB 512   P4 100.0          -    209715200
-                8604 Online   Normal    Normal      2107-900  FB 512   P4 100.0          -    209715200
-                8605 Online   Normal    Normal      2107-900  FB 512   P4 100.0          -    209715200
-                8606 Online   Normal    Normal      2107-900  FB 512   P4 100.0          -    209715200
-                8607 Online   Normal    Normal      2107-900  FB 512   P4 100.0          -    209715200
```

Figure 10-3 on page 412 shows the output of the DSCLI command **lsfbvol** that lists all the fixed block (FB) volumes in the system. Our volume has ID 8601. It is shown in bold. It is created on extent pool number P4.

Next, list the ranks allocated with this volume with command **showfbvol -rank *Vol_ID*** (Example 10-4).

Example 10-4 Listing the ranks allocated with this volume

```
dscli> showfbvol -rank 8601
Name            -
ID              8601
accstate        Online
datastate       Normal
configstate     Normal
deviceMTM       2107-900
datatype        FB 512
addrgrp         8
extpool         P4
exts            100
captype         DS
cap (2^30B)     100.0
```

```
cap (10^9B)      -
cap (blocks)     209715200
volgrp           V8
ranks            2
dbexts           0
sam              ESE
repcapalloc      -
eam              managed
reqcap (blocks) 209715200
realextents      4
virtualextents  100
migrating        0
perfgrp          PG0
migratingfrom    -
resgrp           RG0
==============Rank extents==============
rank extents
============
R17        2
R18        2
```

Figure 10-4 on page 413 shows the output of the command. Volume 8601 is the extent space-efficient (ESE) volume, with virtual capacity of 100 GB, that occupies four extents (two from each) from ranks R17 and R18. We have 4 GB of occupied space. Extent pool P4 is under the Easy Tier automatic management.

Check for the arrays, RAID type, and the DA pair allocation (Figure 10-5 on page 414).

Example 10-5 Listing the array and DA pair

```
dscli> lsrank
ID  Group State    datastate Array RAIDtype extpoolID stgtype
=============================================================
R16     1 Normal   Normal    A13          5 P1        ckd
R17     0 Normal   Normal    A2           5 P4        fb
R18     0 Normal   Normal    A3           5 P4        fb
R20     1 Reserved Normal    A5           5 P5        fb

dscli> lsarray
Array State      Data   RAIDtype    arsite Rank DA Pair DDMcap (10^9B)
======================================================================
A0    Assigned   Normal 5 (6+P+S)   S1     R0   0               300.0
A1    Assigned   Normal 5 (6+P+S)   S2     R1   0               300.0
A2    Assigned   Normal 5 (6+P+S)   S3     R17  2               600.0
A3    Assigned   Normal 5 (6+P+S)   S4     R18  2               600.0
A4    Unassigned Normal 5 (6+P+S)   S5     -    2               600.0
```

Figure 10-5 on page 414 shows the array and DA pair allocation with the **lsarray** and **lsrank** commands. Ranks R17 and R18 relate to arrays A2 and A3 and the Enterprise drives of 600 GB on the DA Pair 2.

From Example 10-4 on page 416, we know that Volume 8601 is in Volume Group V8.

We can list the host connection properties (Figure 10-6 on page 415).

Example 10-6 Listing the host connection properties

```
dscli> lshostconnect -volgrp v8
```

```
Name ID   WWPN             HostType Profile              portgrp volgrpID ESSIOport
==========================================================================================
x3850_lab_5_s12p20 0008 2100001B32937DAE -       Intel - Windows 2008       0 V8       all
x3850_lab_5_s12p22 0009 2100001B3293D5AD -       Intel - Windows 2008       0 V8       all
```

With the command **lshostconnect -volgrp *VolumeGroup_ID***, we can list the ports to which this volume group is connected. See Example 10-6 on page 417. This volume group uses host connections with IDs 0008 and 0009 and the worldwide port names (WWPNs) that are shown in bold. These WWPNs are the same as the WWNs in Example 10-2 on page 416.

List the ports that are used in the disk system for the host connections (Example 10-7).

Example 10-7 Listing the host port connections (outputs truncated)

```
dscli> showhostconnect 0008
Name          x3850_lab_5_s12p20
ID            0008
WWPN          2100001B32937DAE
HostType      -
LBS           512
addrDiscovery LUNPolling
Profile       Intel - Windows 2008
portgrp       0
volgrpID      V8
atchtopo      -
ESSIOport     I0002

dscli> showhostconnect 0009
Name          x3850_lab_5_s12p22
ID            0009
WWPN          2100001B3293D5AD
HostType      -
LBS           512
addrDiscovery LUNPolling
Profile       Intel - Windows 2008
portgrp       0
volgrpID      V8
atchtopo      -
ESSIOport     I0003

dscli> lsioport
ID    WWPN             State  Type             topo     portgrp
===============================================================
I0000 500507630A00029F Online Fibre Channel-SW SCSI-FCP 0
I0001 500507630A00429F Online Fibre Channel-SW FC-AL    0
I0002 500507630A00829F Online Fibre Channel-SW SCSI-FCP 0
I0003 500507630A00C29F Online Fibre Channel-SW SCSI-FCP 0
I0004 500507630A40029F Online Fibre Channel-SW FICON    0
I0005 500507630A40429F Online Fibre Channel-SW SCSI-FCP 0
I0006 500507630A40829F Online Fibre Channel-SW SCSI-FCP 0
I0007 500507630A40C29F Online Fibre Channel-SW SCSI-FCP 0
```

Example 10-7 shows how to obtain the WWPNs and port IDs with the **showhostconnect** and **lsioport** commands. All of the information that we need is in bold.

After these steps, we have all the configuration information for a single disk in the system.

10.8.5 Correlating performance and configuration data

We showed how to get configuration information for one disk manually. Because most of the information was obtained with the CLI, the process can be organized in scripts. This process is easier if you have a large number of disks. It is also possible to correlate the performance collected data and the configuration data. A selection of basic post-processing scripts is in Appendix E, “Post-processing scripts” on page 677. The scripts can be adopted to your needs or modified. The scripts perform several functions:

- Correlates the DS8000 LUNs with the PhysicalDisks seen on the server
- Reformats the perfmon headers so that they are easier to place on charts
- Reformats the data so that it is easier to use Excel Pivot tables
- Reformats the data so that it is less likely to encounter maximum Excel column limitations
- Converts bytes to KBs
- Converts seconds to milliseconds

10.8.6 Analyzing performance data

Performance analysis can be complicated. Because there are many methods to analyze performance, this work cannot be done randomly. The methodology for analyzing disk I/O issues is provided in 7.7, “End-to-end analysis of I/O performance problems” on page 292. In this section, we go deeper into detail and provide suggestions that are specific for the Windows environment and applications.

After the performance data is correlated to the DS8000 LUNs and reformatted, open the performance data file in Microsoft Excel. It looks similar to Figure 10-8.

DATE	TIME	Subsystem Serial	LUN	Disk	Disk Reads/sec	Avg Read RT(ms)	Avg Total Time	Avg Read Queue Length	Read KB/sec
11/3/2008	13:44:48	75GB192	75GB1924	Disk6	1,035.77	0.612	633.59	0.63	66,289.14
11/3/2008	13:44:48	75GB192	75GB1924	Disk2	1,035.75	0.613	634.49	0.63	66,288.07
11/3/2008	13:44:48	75GB192	75GB1924	Disk3	1,035.77	0.612	633.87	0.63	66,289.14
11/3/2008	13:44:48	75GB192	75GB1924	Disk5	1,035.77	0.615	637.11	0.64	66,289.14
11/3/2008	13:44:48	75GB192	75GB1924	Disk4	1,035.75	0.612	634.38	0.63	66,288.07
11/3/2008	13:44:48	75GB192	75GB1924	Disk1	1,035.77	0.612	633.88	0.63	66,289.14
11/3/2008	14:29:48	75GB192	75GB1924	Disk6	1,047.24	5.076	5,315.42	5.32	67,023.08
11/3/2008	14:29:48	75GB192	75GB1924	Disk2	1,047.27	5.058	5,296.86	5.30	67,025.21
11/3/2008	14:29:48	75GB192	75GB1924	Disk3	1,047.29	5.036	5,274.30	5.27	67,026.28
11/3/2008	14:29:48	75GB192	75GB1924	Disk5	1,047.25	5.052	5,291.01	5.29	67,024.14
11/3/2008	14:29:48	75GB192	75GB1924	Disk4	1,047.29	5.064	5,303.36	5.30	67,026.28
11/3/2008	14:29:48	75GB192	75GB1924	Disk1	1,047.29	5.052	5,290.89	5.29	67,026.28
11/3/2008	13:43:48	75GB192	75GB1924	Disk6	1,035.61	0.612	634.16	0.63	66,279.00
11/3/2008	13:43:48	75GB192	75GB1924	Disk2	1,035.61	0.612	633.88	0.63	66,279.00
11/3/2008	13:43:48	75GB192	75GB1924	Disk3	1,035.61	0.615	636.72	0.64	66,279.00

Figure 10-8 The perfmon-essmap.pl script output

A quick look at the compiled data in Figure 10-8 gives us a high increase of the response time without an appropriate increase in the number of IOPS (Disk Reads/sec). This counter shows that a problem occurred, which we confirm with the increased Queue Length value. We must look at the drives that show the response time increase and collect additional data for the drives.

However, even with this data, we can assume possible reasons for what happened and propose further steps:

- Read response time has a value near zero before it increased. The reads were serviced from the cache mostly. Even distribution through all the drives indicates that the workload is mostly random and balanced through all the paths, but it is one of the points to check more.
- We assume that the increase in response time is caused by the increase in the number of read operations, but this assumption is not confirmed. The reason for the assumption is that the response time value increase is about 10 times, but the number of read operations increased by less than 1%. So, this reason cannot be the reason for this response time increase.
- Another assumption might be that the blocksize of the I/O increased and caused a longer time to respond. There is no confirmation for that situation. If you look at the number of megabytes per second, you do not see any dramatic increase. The blocksize remains about 64 KB. This assumption is also not a reason for the increase in response time.
- The only and most likely reason for the behavior of the system is the write activity on the front end or at the back end of the system. That activity typically occurs on the same drives on which the read activity occurs. That write activity is not as cache friendly as expected, which caused the increase of the response time and the increase in queue length. Probably, it was a batch activity or a large block write activity.

Because we have a possible reason for the read response time increase, we can specify the further steps to confirm it:

- Gather additional performance data for the volumes from the Windows server, including write activity.
- Gather performance data from the back end of the disk system on those volumes for any background activity or secondary operations.
- Examine the balancing policy on the disk path, but it looks good.
- Examine the periodic processes initiated in the application. There might be activity on the log files.
- For database applications, separate the log files from the main data and indexes.
- Check for any other activity on that drive that can cause an increase of the write I/Os.

You can see how even a small and uncertain amount of the collected performance data can help you in the early detection of performance problems and help you quickly identify further steps.

Removing disk bottlenecks

Disk bottlenecks occur in two potential places:

- Applications and operating system
- Disk system

At the disk subsystem level, there can be bottlenecks on the rank level, extent pool level, device adapter pair level, and cache level that lead to problems on the volume level. Table 10-3 on page 421 describes the reasons for the problems on different levels.

Table 10-3 Bottlenecks on the disk system levels (the numbers correspond)

Disk system level	Possible reasons	Actions
Rank level	1. Rank IOPS capability exceeded: Rank bandwidth capability exceeded 2. RAID type does not fit the workload type 3. Disk type is wrong 4. Physical problems with the disks in the rank	1. Split workload between several ranks organized into one extent pool with rotate extents feature. If already organized, manually rebalance the ranks. 2. Change the RAID level to a better performing RAID level (RAID 5 to RAID 10, for example) or migrate extents to another extent pool with better conditions. 3. Migrate extents to the better performing disks. 4. Fix the problems with the disks.
Extent pool level	1. Extent pool capability reached its maximum 2. Conflicting workloads mixed in the same extent pool 3. No Easy Tier management for this pool 4. One of the ranks in the pool is overloaded 5. Physical problems with the disk in the rank	1. Add more ranks to the pool; examine the STAT reports for the recommendations; add more tiers in the pool; or benefit from hot promotion and cold demotion. 2. Split workloads to the separate pools; split one pool into two dedicated to both processor complexes; examine the STAT data and add required tier; or set priorities for the workloads and enable IOPM. 3. Start Easy Tier for this pool by following the recommendations from the STAT tool. 4. Perform the extent redistribution for the pool; start Easy Tier and follow recommendations from the STAT tool; or use rotate extents method. 5. Fix the problems with the disks or remove the rank from the pool.
DA pair level	1. DA pair is overloaded 2. Hardware malfunction on DA pair 3. Hardware problem with one or more ranks on the DA pair	1. Mind the limits of the DA pair; decrease the number of solid-state drives (SSDs) on the DA pair; or use ranks from as many different DA pairs as possible. 2. Fix the problem with the hardware. 3. Fix the problem with the rank or exclude it from the extent pools.

Disk system level	Possible reasons	Actions
Cache level	1. Cache-memory limits are reached 2. Workload is not “cache-friendly” 3. Large number of write requests to single volume (rank or extent pool)	1. Upgrade the cache memory; add more ranks to the extent pool; enable Easy Tier; or split extent pools evenly between CECs. 2. Add more disks; benefit from the micro-tiering to unload the 15K rpm drives; or tune application if possible. 3. Split the pools and ranks evenly between CEC to be able to use all the cache memory.
CEC level	1. CECs overloaded 2. Uneven volume, extent pool, or rank assignment on CECs 3. CEC hardware problems	1. Split the workload between two CECs evenly; stop the unnecessary Copy Services; or upgrade the system to higher model. 2. Split the pools and ranks evenly between CEC to be able to use all the cache memory and processor power. 3. Fix the problems.
Host adapter level	1. Ports overloaded 2. Host I/Os mixed with Copy Services I/Os 3. Faulty SFPs 4. Incorrect cabling 5. Other hardware problems	1. Add more host adapters; change host adapters to the better performing kind (4 Gbps - 8 Gbps); use other multipathing balancing approach; or remember the recommended number of the logical volumes per port. 2. Use dedicated adapters for the Copy Services; split host ports for the different operating system; or separate backup workloads and host I/O workloads. 3. Replace SFPs. 4. Change cabling or check cables with tools. 5. Fix the problems with SAN hardware.

At the application level, there can be bottlenecks in the application, multipathing drivers, device drivers, zoning misconfiguration, or adapter settings. However, a Microsoft environment is self-tuning and many problems might be fixed without any indication. Windows can cache many I/Os and serve them from cache. It is important to maintain a large amount of free Windows Server memory for peak usage. Also, the paging file must be set up based on our suggestions in 10.4.3, “Paging file” on page 401.

To avoid bottlenecks in the application, follow these guidelines:

- MS SQL Servers do not require separate placement of the log files and data files, because many transactions are cached in the operating system and SQL Server. However, it is still a database, and you need to place log files and data files separately. Never put log files with the paging files. Never place log files on the main system disks (that is, SystemRoot%). Use three-tier technology whenever possible, because the SQL Server varies in its I\O and access pattern. Use IOPM to keep SQL Server volumes at the highest priority all the time. For additional MS SQL Server tuning suggestions, see this link:

  http://technet.microsoft.com/en-us/library/bb545450.aspx

- MS Exchange Server is a unique application that behaves as the OLTP database and the data warehousing database. Serial Advanced Technology Attachment (SATA) or nearline drives are not advised for the MS Exchange Server, but with Easy Tier and Enterprise+Nearline tiers in one extent pool, you can use most of the capacity on the nearline drives. This approach is true from the workload pattern view, which is up to 50 - 60% write. Also, the micro-tiering option works well. So, you might consider putting 90% of the capacity on 10K rpm drives. Remember to not use RAID 6 or RAID 5 for high-performing MS Exchange Servers due to the high random write ratio. The MS Exchange database and indexing must be physically separated from the data. At least two logical disks on separate extent pools are advised for the MS Exchange setup. For additional MS Exchange Server tuning suggestions, see this link:

  http://technet.microsoft.com/en-us/library/aa996058%28EXCHG.80%29.aspx

For the other Microsoft applications, follow the suggestions in Table 10-1 on page 404:

- To avoid bottlenecks on the SDDDSM side, maintain a balanced use of all the paths and keep them active always. See Example 10-8. You can see the numbers for reads and writes on each adapter, which are nearly the same.

Example 10-8 Finding out the FC adapter statistics

```
c:\Program Files\IBM\SDDDSM>datapath query adaptstats
Adapter #:  0
=============
                 Total Read  Total Write  Active Read  Active Write   Maximum
I/O:                   3003         2522            0             0         2
SECTOR:              306812       367257            0             0      2048
Adapter #:  1
=============
                 Total Read  Total Write  Active Read  Active Write   Maximum
I/O:                   3103         2422            0             0         2
SECTOR:              306595       306094            0             0      2049
```

- SAN zoning, cabling, and FC-adapter settings must be done according to the *IBM System Storage DS8700 and DS8800 Introduction and Planning Guide,* GC27-2297-07, but remember not to have more than four paths per logical volume.

After you detect a disk bottleneck, you might perform several of these actions:

- If the disk bottleneck is a result of another application in the shared environment that causes disk contention, request a LUN on a less utilized rank and migrate the data from the current rank to the new rank. Start by using Priority Groups.
- If the disk bottleneck is caused by too much load that is generated from the Windows Server to a single DS8000 LUN, spread the I/O activity across more DS8000 ranks, which might require the allocation of additional LUNs. Start Easy Tier for the volumes and migrate to hybrid pools.

- On Windows Server 2008 for sequential workloads with large transfer sizes (256 KB), consider the micro-tiering option or Enterprise+nearline drive hybrid pools.
- Move processing to another system in the network (either users, applications, or services).
- Add more RAM. Adding memory increases system memory disk cache, which might reduce the number of required physical I/Os and indirectly reduce disk response times.
- If the problem is a result of a lack of bandwidth on the HBAs, install additional HBAs to provide more bandwidth to the DS8000.

For more information about Windows Server disk subsystem tuning, see the following document:

http://www.microsoft.com/whdc/archive/subsys_perf.mspx

11

Performance considerations for VMware

This chapter describes the monitoring and tuning tools and techniques that can be used with VMware ESX/ESXi to optimize throughput and performance when attaching to the DS8000.

This chapter addresses the following topics:

- I/O architecture from a VMware perspective
- Initial planning considerations for optimum performance of VMware host systems that use the DS8000 in a storage area network (SAN)
- Specific VMware performance measuring tools and tuning options
- SAN multipathing considerations
- Testing and verifying the DS8000 storage attached to VMware host systems
- Configuring VMware logical storage for optimum performance
- VMware operating system tuning considerations for maximum storage performance

11.1 Disk I/O architecture overview

The DS8000 currently supports the VMware high-end virtualization solution. Other VMware products, such as VMware Server and Workstation, are not intended for the data center-class environments where the DS8000 is typically used.

This chapter introduces the relevant logical configuration concepts needed to attach VMware ESX/ESXi Server to a DS8800/DS8700 and focuses on performance-relevant configuration and measuring options. For further information about how to set up ESX/ESXi Server with the DS8000, see *IBM System Storage DS8000 Architecture and Implementation*, SG24-8886. You can obtain general suggestions about how to set up VMware with IBM hardware from *Tuning IBM System x Servers for Performance*, SG24-5287.

The main difference between ESX and ESXi is that ESXi is an embedded platform of ESX on a hardware component and instead of a service console with a traditional platform, it acts like firmware. Among other differences, ESXi does not support SAN boot. You can obtain more information about a comparison of the versions at this website:

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1015000

VMware ESX/ESXi Server supports the use of external storage that can reside on a DS8000 system. The DS8000 storage is typically connected by Fibre Channel (FC) and accessed over a SAN. Each logical volume that is accessed by a ESX/ESXi Server is configured in a specific way, and this storage can be presented to the virtual machines (VMs) as virtual disks.

To understand how storage is configured in ESX Server, you must understand the layers of abstraction that are shown in Figure 11-1.

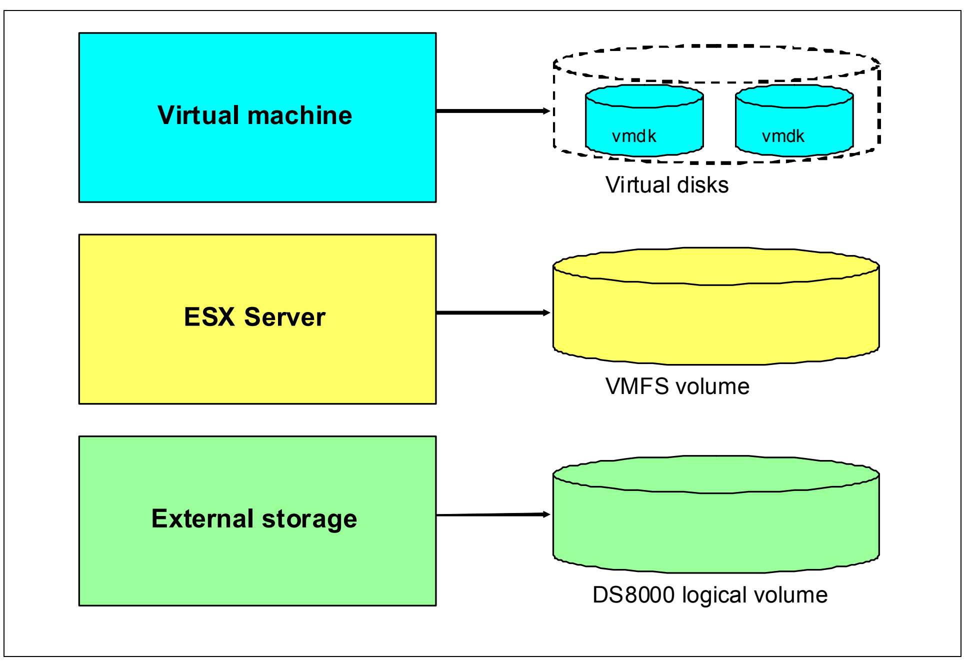


Figure 11-1 Storage stack for ESX Server

For VMware to use external storage, VMware needs to be configured with logical volumes that are defined in accordance with the expectations of the users, which might include the use of RAID or striping at a storage hardware level. Striping at a storage hardware level is preferred, because DS8800/DS8700 can combine EasyTier and IOPM mechanisms. These logical volumes must be presented to ESX/ESXi Server. For the DS8000, host access to the volumes includes the correct configuration of logical volumes, host mapping, correct logical unit number (LUN) masking, and zoning of the involved SAN fabric.

At the ESX Server layer, these logical volumes can be addressed as a VMware ESX/ESXi Server File System (VMFS) volume or as a raw disk that uses Raw Device Mapping (RDM). A VMFS volume is a storage resource that can serve several VMs as well as several ESX Servers as *consolidated storage*. An RDM volume, however, is intended for usage as *isolated storage* by a single VM.

Two options exist to use these logical drives within vSphere Server:

- Formatting these disks with the VMFS: This option is the most common option, because a number of features require that the virtual disks are stored on VMFS volumes.
- Passing the disk through to the guest OS as a raw disk. No further virtualization occurs. Instead, the OS writes its own filesystem onto that disk directly as though it is in a stand-alone environment without an underlying VMFS structure.

The VMFS volumes house the virtual disks that the guest OS sees as its real disks. These virtual disks are in the form of a file with the extension.vmdk. The guest OS either read/writes to the virtual disk file (.vmdk) or writes through the VMware ESX/ESXi Server abstraction layer to a raw disk. In either case, the guest OS considers the disk to be real.

Next, in Figure 11-2, we compare VMware VMFS volumes to logical volumes, so that you can understand the logical volumes for a DS8800/DS8700 as references to volume IDs, for example, 1000, 1001, and 1002.

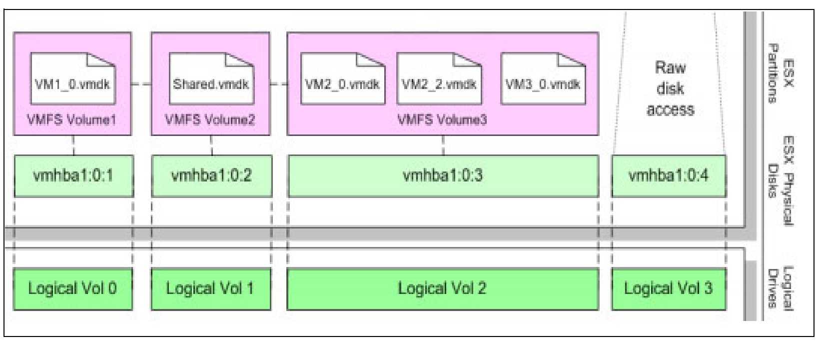


Figure 11-2 Logical drives compared to VMware VMFS volumes

On the Virtual Machine layer, you can configure one or several Virtual Disks (VMDKs) out of a single VMFS volume. These Virtual Disks can be configured for use by several VMs.

VMware datastore concept

ESX/ESXi Server uses specially formatted logical containers called *datastores*. These datastores can reside on various types of physical storage devices, local disks inside ESX/ESXi Server, and FC-attached disks, as well as iSCSI disks and Network File System (NFS) disks.

The virtual machine disks are stored as files within a VMFS. When a guest operating system issues a Small Computer System Interface (SCSI) command to its virtual disks, the VMware virtualization layer converts this command to VMFS file operations. From the standpoint of the virtual machine operating system, each virtual disk is recognized as a direct-attached SCSI drive connected to a SCSI adapter. Device drivers in the virtual machine operating system communicate with the VMware virtual SCSI controllers. Figure 11-3 on page 428 illustrates the virtual disk mapping within VMFS.

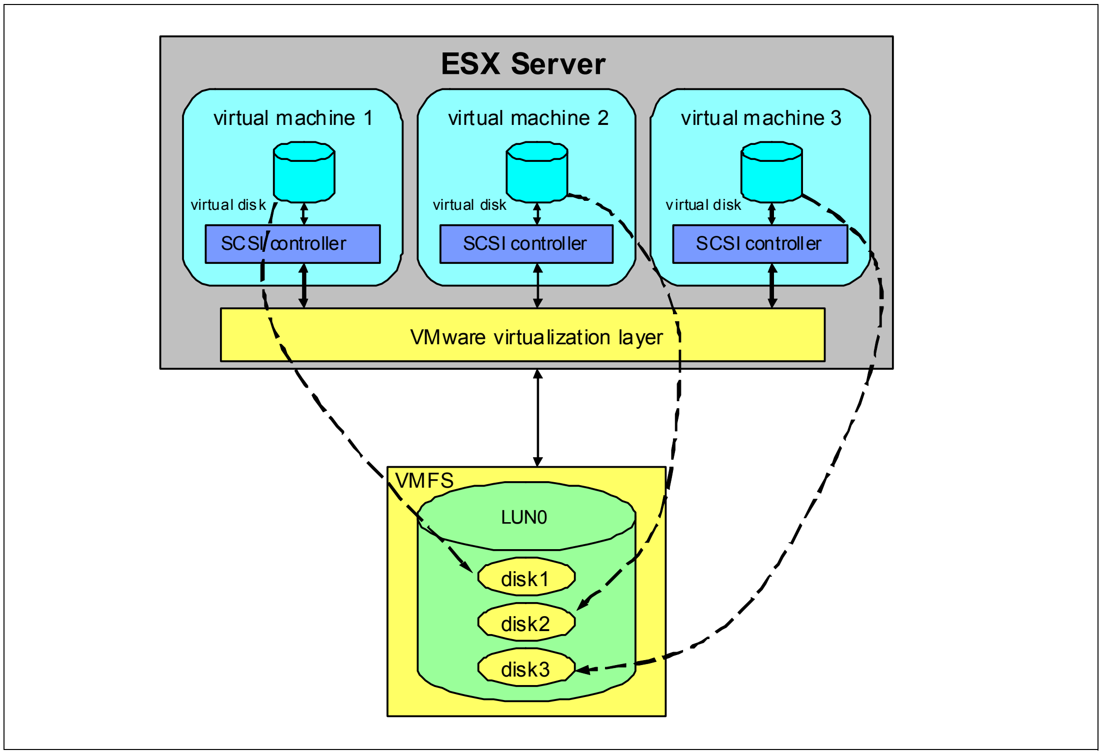


Figure 11-3 Mapping virtual disks to LUNs within VMFS

VMFS is optimized to run multiple virtual machines as one workload to minimize disk I/O overhead. A VMFS volume can be spanned across several logical volumes, but there is no striping available to improve disk throughput in these configurations. Each VMFS volume can be extended by adding additional logical volumes while the virtual machines use this volume.

The VMFS volumes store this information:

- Virtual machine .vmdk files
- The memory images from virtual machines that are suspended
- Snapshot files for the .vmdk files that are set to a disk mode of non-persistent, undoable, or append

> **Important:** A VMFS volume can be spanned across several logical volumes, but there is no striping available to improve disk throughput in these configurations. With Easy Tier, hot/cold extents can be promoted or demoted, and you can achieve superior performance versus economics on VMware ESX/ESXi hosts as well.

An RDM is implemented as a special file in a VMFS volume that acts as a proxy for a raw device. An RDM combines the advantages of direct access to physical devices with the advantages of virtual disks in the VMFS. In special configurations, you must use RDM raw devices, such as in Microsoft Cluster Services (MSCS) clustering, by using virtual machine snapshots, or by using VMotion, which enables the migration of virtual machines from one datastore to another with zero downtime.

With RDM volumes, ESX/ESXi Server supports the use of N_Port ID Virtualization (NPIV). This host bus adapter (HBA) virtualization technology allows a single physical HBA port to function as multiple logical ports, each with its own worldwide port name (WWPN). This function can be helpful when you migrate virtual machines between ESX/ESXi Servers by using VMotion, as well as to separate workloads of multiple VMs configured to the same paths on the HBA level for performance measurement purposes.

The VMware ESX/ESXi virtualization of datastores is shown in Figure 11-4.

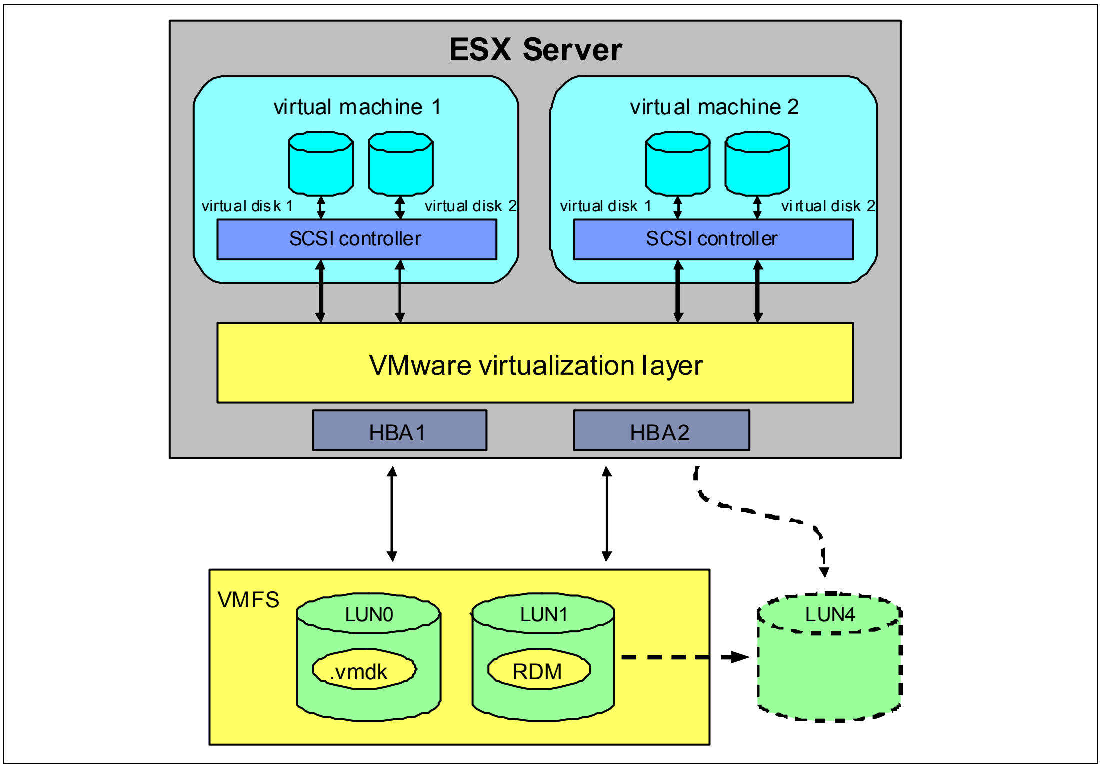


Figure 11-4 VMware virtualization of datastores

11.2 Host type for the DS8000

To create your volume groups on a DS8000 to support VMware ESX/ESXi hosts, you must create them using **type scsimap256**. And for host connections, you must reference each host by using the host type **VMware** (with **LUNPolling** as the SCSI address discovery method) as seen in Example 11-1. A list of available SCSI host types on DS8800/DS8700 can be obtained with the command **lshosttype** by specifying the option **-type scsiall**.

Example 11-1 Creating volume groups and host connections for VMware hosts

```
dscli> mkvolgrp -type scsimap256 VMware_Host_1_volgrp_1
CMUC00030I mkvolgrp: Volume group V19 successfully created.

dscli> mkhostconnect -wwpn 21000024FF2D0F8D -hosttype VMware -volgrp V19 -desc "Vmware host1 hba1"
Vmware_host_1_hba_1
CMUC00012I mkhostconnect: Host connection 0036 successfully created.

dscli> lshostconnect -l -volgrp V19
Name                ID   WWPN             HostType LBS addrDiscovery Profile portgrp volgrpID atchtopo ESSIOport speed   desc
===============================================================================================================================
Vmware_host_1_hba_1 0036 21000024FF2D0F8D VMWare   512 LUNPolling    VMWare        0 V19      -        all       Unknown Vmware host1 hba1
```

11.3 Multipathing considerations

In ESX/ESXi Server, the name of the storage device is displayed as a sequence of three to four numbers separated by colons, for example, `vmhba2:0:1:1`. This naming has the following meaning:

<HBA>:<SCSI target>:<SCSI LUN>:<disk partition>

The abbreviation vmhba refers to the physical HBA types: either a Fibre Channel HBA, a SCSI adapter, or even an iSCSI initiator. The SCSI target number and SCSI LUN are assigned during the scanning of the HBAs for available storage and usually do not change later. The fourth number indicates a partition on a disk that a VMFS datastore occupies and must never change for a selected disk. Thus, this example, `vmhba2:0:1:1`, refers to the first partition on SCSI LUN1, SCSI target 1, and is accessed through HBA 2.

In multipathing environments, which are the standard configuration in correctly implemented SAN environments, the same LUN can be accessed through several paths. The same LUN has two or more storage device names. After a rescan or reboot, the path information displayed by ESX/ESXi Server might change; however, the name still refers to the same physical device. In Figure 11-5, the identical LUN can be addressed as `vmhba2:0:1` or `vmhba2:1:1`, which can easily by verified in the Canonical Path column. If multiple HBAs are used to connect to the SAN-attached storage for redundancy reasons, this LUN can also be addressed through a different HBA, in this configuration, for example, `vmhba5:0:1` and `vmhba5:1:2`.

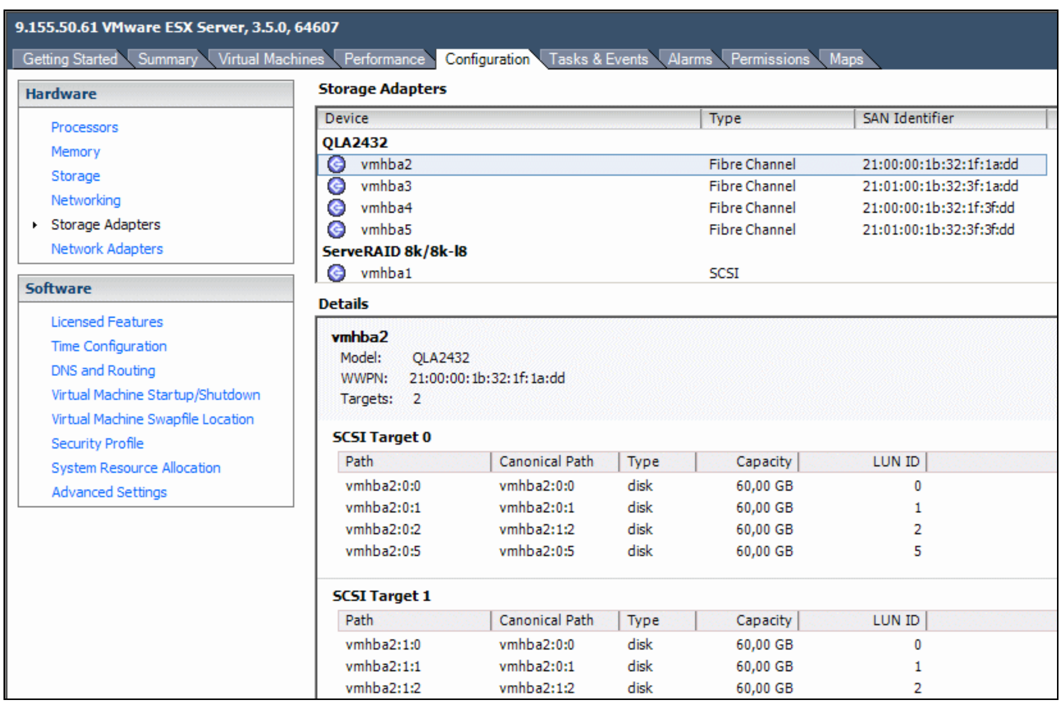


Figure 11-5 Storage Adapters properties view in the Virtual Infrastructure Client (VI Client)

ESX/ESXi Server provides built-in multipathing support, which means that it is not necessary to install any additional failover driver. Any external failover drivers, such as subsystem device drivers (SDD), are not supported for VMware ESX/ESXi. Since ESX/ESXi 4.0, it supports path failover and the round-robin algorithm.

ESX/ESXi Server 4.0 provides three major multipathing policies for use in production environments: *Most Recently Used (MRU), Fixed,* and *Round Robin (RR)*:

- MRU policy is designed for usage by active/passive storage devices, such as DS4000s, with only one active controller available per LUN.
- The Fixed policy ensures that the designated preferred path to the storage is used whenever available. During a path failure, an alternate path is used, and as soon as the preferred path is available again, the multipathing module switches back to it as the active path.
- The Round Robin policy with a DS8000 system uses all available paths to rotate through the available paths. It is possible to switch from MRU and Fixed to RR without interruptions. With RR, you can change the number of bytes and number of I/O operations sent along one path before you switch to the other path. RR is a good approach for various systems; however, it is not supported for use with virtual machines that are part of Microsoft Cluster Services (MSCS).

The default multipath policy for ALUA devices since ESX/ESXi 5 is MRU.

The multipathing policy and the preferred path can be configured from the VI Client or by using the command-line tool **`esxcfg-mpath`** `or` **`esxcli`** (newer versions). See Table 11-1 for command differences among the ESX/ESXi versions.

Table 11-1 Commands for changing multipath policy on ESX/ESXi hosts

VMware version	Change Multipath Policy Cmd	Path Policy available
ESX/ESXi 5	esxcli storage nmp device set --device <*naa_id*> --psp <*path_policy*>	VMW_PSP_FIXED, VMW_PSP_MRU, and VMW_PSP_RR
ESX/ESXi 4	esxcli nmp -d <*naa_id*> <*path_policy*>	VMW_PSP_FIXED, VMW_PSP_MRU, and VMW_PSP_RR
ESX/ESXi 3	esxcfg-mpath --policy=<*path_policy*> --lun=<*device*>	fixed, mru, or rr

Figure 11-6 shows how the preferred path changed from the VI Client.

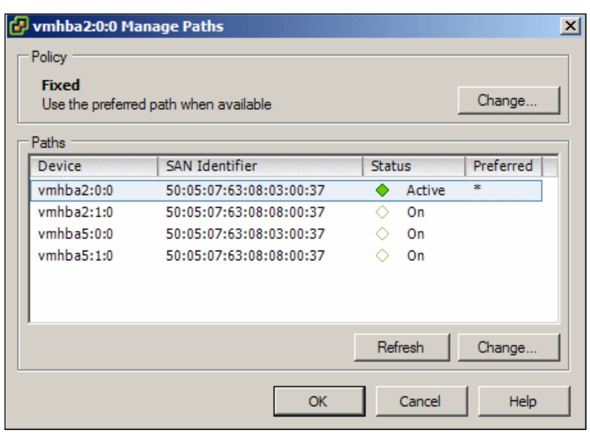


Figure 11-6 Manage Paths window in the VI Client

By using the Fixed multipathing policy, you can implement *static load balancing* if several LUNs are attached to the VMware ESX/ESXi Server. The multipathing policy is set on a per

LUN basis, as well as the preferred path is chosen for each LUN. If ESX/ESXi Server is connected over four paths to its DS8000 storage, we suggest that you spread the preferred paths over all four available physical paths.

> **Important:** Remember that before zoning your VMware host to a DS8000, you must ensure that the number of available paths for each LUN must meet a minimum of two paths (for redundancy), but due to limitations on VMware a maximum of all paths on a host is 1024. The number of paths to a LUN is limited to 32, and the maximum available LUNS per host is 256. Also, the maximum size of a LUN is (2 TB - 256 bytes). So, plan the LUN size and the number of paths available to each VMware host carefully to avoid future problems with provisioning to your VMware hosts.

For example, when you want to configure four LUNs, assign the preferred path of LUN0 through the first path, the one for LUN1 through the second path, the preferred path for LUN2 through the third path, and the one for LUN3 through the fourth path. With this method, you can spread the throughput over all physical paths in the SAN fabric. Thus, this method results in optimized performance for the physical connections between the ESX/ESXi Server and the DS8000.

If the workload varies greatly between the accessed LUNs, it might be a good approach to monitor the performance on the paths and adjust the configuration according to the workload. It might be necessary to assign one path as preferred to only one LUN with a high workload but to share another path as preferred between five separate LUNs that show moderate workloads. This *static load balancing* works only if all paths are available. As soon as one path fails, all LUNs that selected this failing path as preferred fail over to another path and put additional workload onto those paths. Furthermore, there is no capability to influence the failover algorithm to which path the failover occurs.

When the active path fails, for example, due to a physical path failure, I/O might pause for about 30 - 60 seconds until the FC driver determines that the link is down and fails over to one of the remaining paths. This behavior can cause the virtual disks used by the operating systems of the virtual machines to appear unresponsive. After failover is complete, I/O resumes normally. The timeout value for detecting a failed link can be adjusted; it is set in the HBA BIOS or driver and the way to set this option depends on the HBA hardware and vendor. The typical failover timeout value is 30 seconds. With VMware ESX/ESXi, you can adjust this value by editing the device driver options for the installed HBAs in `/etc/vmware/esx.conf`.

Additionally, you can increase the standard disk timeout value in the virtual machine operating system to ensure that the operating system is not extensively disrupted and to ensure that the system logs permanent errors during the failover phase. Adjusting this timeout again depends on the operating system that is used and the amount of queue that is expected by one path once it fails; see the appropriate technical documentation for details.

11.4 Performance monitoring tools

This section reviews the performance monitoring tools that are available with VMware ESX/ESXi.

11.4.1 Virtual Center performance statistics

Virtual Center (VC) is the entry point for virtual platform management in VMware ESX/ESXi. It also includes a module to view and analyze performance statistics and counters. The Virtual

Center performance counters collection is reduced by default to a minimum level, but you can modify the settings to allow a detailed analysis.

VC includes real-time performance counters that display the past hour (which is not archived), as well as archived statistics that are stored in a database. The real-time statistics are collected every 20 seconds and presented in the VI Client for the past 60 minutes (Figure 11-7).

These real-time counters are also the basis for the archived statistics, but to avoid too much performance database expansion, the granularity is recalculated according to the age of the performance counters. Virtual Center collects those real-time counters, aggregates them for a data point every 5 minutes and stores them as past-day statistics in the database. After one day, these counters are aggregated once more to a 30-minute interval for the past week statistics. For the past month, a data point is available every 2 hours, and for the last year, one datapoint is stored per day.

In general, the Virtual Center statistics are a good basis to get an overview about the actual performance statistics and to further analyze performance counters over a longer period, for example, several days or weeks. If a granularity of 20-second intervals is sufficient for your individual performance monitoring perspective, VC can be a good data source after configuration. You can obtain more information about how to use the Virtual Center Performance Statistics at this website:

http://communities.vmware.com/docs/DOC-5230

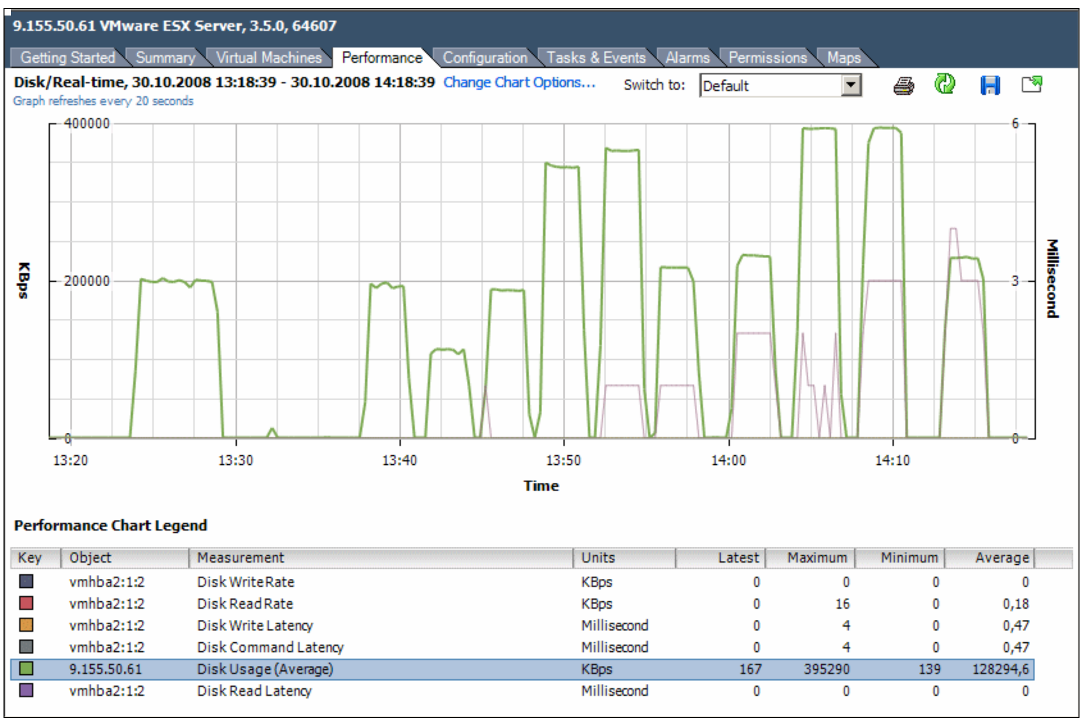


Figure 11-7 Real-time performance statistics in VI Client

11.4.2 Performance monitoring with esxtop

The **esxtop** command-line tool provides the finest granularity among the performance counters available within VMware ESX/ESXi Server. The tool is available on the ESX/ESXi Server service console, and you must have root privileges to use the tool. The **esxtop** command is available for usage either in Interactive Mode or in Batch Mode. When using Interactive Mode, the performance statistics are displayed inside the command-line console.

With Batch Mode, you collect and save the performance counters in a file. The `esxtop` utility reads its default configuration from a file called `~/.esxtop3rc`. The best way to configure this default configuration to fit your needs is to change and adjust it for a running **esxtop** process and then save this configuration by using the W interactive command. Example 11-2 illustrates the basic adjustments required to monitor disk performance on a SAN-attached storage device.

Example 11-2 Esxtop basic adjustment for monitoring disk performance

```
esxtop                      #starts esxtop in Interactive Mode

PCPU(%):   3.27,    3.13,    2.71,    2.66 ;   used total:    2.94
       7      7 drivers            16     0.01     0.01     0.00 1571.25     0.00     0.00     0.00     0.00     0.00
       8      8 vmotion             1     0.00     0.00     0.00   98.20     0.00     0.00     0.00     0.00     0.00
       9      9 console             1     0.85     0.84     0.01   97.04     0.32    97.03     0.06     0.00     0.00
      21     21 vmware-vmkauthd     1     0.00     0.00     0.00   98.20     0.00     0.00     0.00     0.00     0.00
      22     22 Virtual Center     12     6.79     6.81     0.00 1170.05     1.60   383.57     0.20     0.00     0.00
      31     31 Windows1            6     0.86     0.86     0.00  588.24     0.14    97.37     0.13     0.00     0.00
      32     32 Windows2            6     0.58     0.58     0.00  588.54     0.09    97.84     0.05     0.00     0.00
      33     33 Windows3            6     0.60     0.61     0.00  588.52     0.09    97.95     0.08     0.00     0.00
      34     34 Linux 1             5     1.40     1.41     0.00  489.17     0.46    96.56     0.22     0.00     0.00

d                           #changes to disk storage utilization panels
e vmhba2                    #selects expanded display of vmhba2
a vmhba2:0                  #selects expanded display of SCSI channel 0
t vmhba2:0:0                #selects expanded display mode of SCSI target 0
W                           #writes the current configuration into ~/.esctop3rc file
```

After this initial configuration, the performance counters are displayed as shown in Example 11-3.

Example 11-3 Disk performance metrics in esxtop

```
1:25:39pm up 12 days 23:37, 86 worlds; CPU load average: 0.36, 0.14, 0.17

 ADAPTR CID TID LID  NCHNS NTGTS NLUNS NVMS AQLEN LQLEN WQLEN ACTV %USD  LOAD   CMDS/s  READS/s WRITES/s MBREAD/s
 vmhba1   -   -   -      2     1     1   32   238     0     0    -    -     -     4.11     0.20     3.91     0.00
 vmhba2   0   0   0      1     1     1   10  4096    32     0    8   25  0.25 25369.69 25369.30     0.39   198.19
 vmhba2   0   0   1      1     1     1   10  4096    32     0    0    0  0.00     0.00     0.00     0.00     0.00
 vmhba2   0   0   2      1     1     1   10  4096    32     0    0    0  0.00     0.39     0.00     0.39     0.00
 vmhba2   0   0   3      1     1     1    9  4096    32     0    0    0  0.00     0.39     0.00     0.39     0.00
 vmhba2   0   0   4      1     1     1    4  4096    32     0    0    0  0.00     0.00     0.00     0.00     0.00
 vmhba2   0   0   5      1     1     1   17  4096    32     0    0    0  0.00     0.00     0.00     0.00     0.00
 vmhba2   0   0   6      1     1     1    4  4096    32     0    0    0  0.00     0.00     0.00     0.00     0.00
 vmhba2   0   0   7      1     1     1    4  4096    32     0    0    0  0.00     0.00     0.00     0.00     0.00
 vmhba2   0   0   8      1     1     1    4  4096    32     0    0    0  0.00     0.00     0.00     0.00     0.00
 vmhba2   0   0   9      1     1     1    4  4096    32     0    0    0  0.00     0.00     0.00     0.00     0.00
 vmhba2   0   1   -      1     1    10   76  4096     0     0    -    -     -     0.00     0.00     0.00     0.00
 vmhba3   -   -   -      1     2     4   16  4096     0     0    -    -     -     0.00     0.00     0.00     0.00
 vmhba4   -   -   -      1     2     4   16  4096     0     0    -    -     -     0.00     0.00     0.00     0.00
 vmhba5   -   -   -      1     2    20  152  4096     0     0    -    -     -     0.78     0.00     0.78     0.00
```

Additionally, you can change the field order and select or clear various performance counters in the view. The minimum refresh rate is 2 seconds, and the default setting is 5 seconds.

When you use esxtop in Batch Mode, always include all of the counters by using the option **-a**. To collect the performance counters every 10 seconds for 100 iterations and save them to a file, run **esxtop** this way:

```
esxtop -b -a -d 10 -n 100 > perf_counters.csv
```

For additional information about how to use **esxtop** and other tools, see *vSphere Resource Management Guide*:

http://www.vmware.com/pdf/vsphere4/r40/vsp_40_resource_mgmt.pdf

Each VMware version has its own particularities and some might have performance analysis tools that are not part of the older versions.

11.4.3 Guest-based performance monitoring

Because the operating systems that run in the virtual machines host the applications that perform the host workload, it makes sense to use performance monitoring in these operating systems as well. We describe the tools that you use in Chapter 10, “Performance considerations for Microsoft Windows servers” on page 397 and Chapter 12, “Performance considerations for Linux” on page 445.

The guest operating system is unaware of the underlying VMware ESX/ESXi virtualization layer, so any performance data captured inside the VMs can be misleading and must be analyzed and interpreted only in conjunction with the actual configuration and performance data gathered in ESX/ESXi Server or on a disk or SAN layer.

There is one additional benefit of the Windows Performance Monitor perfmon (see 10.8.2, “Windows Performance console (perfmon)” on page 411). When you use **esxtop** in Batch Mode with option **-a**, it collects all available performance counters and thus the collected comma-separated values (csv) data gets large and cannot be easily parsed. Perfmon can help you to quickly analyze results or to reduce the amount of csv data to a subset of counters that can be analyzed more easily by using other utilities. You can obtain more information about importing the **esxtop** csv output into perfmon:

http://communities.vmware.com/docs/DOC-5100

VMware specific tuning for maximum performance

Due to the special VMware ESX/ESXi Server setup and the additional virtualization layer implemented in ESX/ESXi Server, it is necessary to focus on additional topics and configuration options when you tune VMware ESX/ESXi. This section focuses on important points about tuning the VMware ESX/ESXi Server with attached DS8000 storage to achieve maximum performance.

11.4.4 Workload spreading

We suggest that you spread the I/O workload across the available hardware. This method is the most effective way to avoid any hardware limitations of either the HBA, processor complex, device adapter, or disk drives that negatively affect the potential performance of your ESX/ESXi Server.

It is also important to identify and separate specific workloads, because they can negatively influence other workloads that might be more business critical.

Within ESX/ESXi Server, it is not possible to configure striping over several LUNs for one datastore. It is possible to add more than one LUN to a datastore, but adding more than one LUN to a datastore only extends the available amount of storage by concatenating one or more additional LUNs without balancing the data over the available logical volumes.

The easiest way to implement striping over several hardware resources is to use storage pool striping in extent pools (see 4.8, “Planning extent pools” on page 115 for further information) of the attached DS8000.

The only other possibility to achieve striping at the Virtual Machine level is to configure several virtual disks for a VM that are on different hardware resources, such as different HBAs, device adapters, or servers, and then configure striping of those virtual disks within the host operating system layer.

We suggest that you use storage pool striping, because the striping only has to be implemented one time when configuring the DS8000. Implementing the striping on the host operating system level requires you to configure it for each of the VMs separately. Furthermore, according to the VMware documentation, host-based striping is currently only supported using striping within Windows dynamic disks.

For performance monitoring purposes, be careful with spanned volumes or even avoid these configurations. When configuring more than one LUN to a VMFS datastore, the volume space is spanned across multiple LUNs, which can cause an imbalance in the utilization of those LUNs. If several virtual disks are initially configured within a datastore and the disks are mapped to different virtual machines, it is no longer possible to identify in which area of the configured LUNs the data of each VM is allocated. Thus, it is no longer possible to pinpoint which host workload causes a possible performance problem.

In summary, avoid using spanned volumes and configure your systems with only one LUN per datastore.

11.4.5 Virtual machines sharing the LUN

The SCSI protocol allows multiple commands to be active for the same LUN at one time. A configurable parameter called *LUN queue depth* determines how many commands can be active at one time for a certain LUN. This queue depth parameter is handled by the SCSI driver for a specific HBA. Depending on the HBA type, you can configure up to 255 outstanding commands for a QLogic HBA, and Emulex supports up to 128. The default value for both vendors is 32.

If a virtual machine (VM) generates more commands to a LUN than the LUN queue depth, these additional commands are queued in the ESX/ESXi kernel, which increases the latency. The queue depth is defined on a per LUN basis, not per initiator. An HBA (SCSI initiator) supports many more outstanding commands.

For ESX/ESXi Server, if two virtual machines access their virtual disks on two different LUNs, each VM can generate as many active commands as the LUN queue depth. But if those two virtual machines have their virtual disks on the same LUN (within the same VMFS volume), the total number of active commands that the two VMs combined can generate without queuing I/Os in the ESX/ESXi kernel is equal to the LUN queue depth. Therefore, when several virtual machines share a LUN, the maximum number of outstanding commands to that LUN from all those VMs together must not exceed the LUN queue depth.

Within ESX/ESXi Server, there is a configuration parameter *Disk.SchedNumReqOutstanding,* which can be configured from the Virtual Center. If the total number of outstanding commands from all virtual machines for a specific LUN exceeds this parameter, the remaining commands are queued to the ESX/ESXi kernel. This parameter must always be set at the same value as the queue depth for the HBA.

To reduce latency, it is important to ensure that the sum of active commands from all virtual machines of an ESX/ESXi Server does not frequently exceed the LUN queue depth. If the LUN queue depth is exceeded regularly, you might either increase the queue depth or move the virtual disks of a few virtual machines to different VMFS volumes. Therefore, you lower the number of virtual machines that access a single LUN. The maximum LUN queue depth per ESX/ESXi Server must not exceed 64. The maximum LUN queue depth per ESX/ESXi Server can be up to 128 *only* when a server has exclusive access to a LUN.

VMFS is a filesystem for clustered environments, and it uses SCSI reservations during administrative operations, such as creating or deleting virtual disks or extending VMFS volumes. A reservation ensures that at a specific time, a LUN is only available to one

ESX/ESXi Server exclusively. These SCSI reservations usually are used for administrative tasks that require a metadata update only.

To avoid SCSI reservation conflicts in a productive environment with several ESX/ESXi Servers that access shared LUNs, it might be helpful to perform those administrative tasks at off-peak hours. If this approach is not possible, perform the administrative tasks from an ESX/ESXi Server that also hosts I/O-intensive virtual machines, which are less affected because the SCSI reservation is set on the SCSI initiator level, which means for the complete ESX/ESXi Server.

The maximum number of virtual machines that can share the LUN depends on several conditions. In general, virtual machines with heavy I/O activity result in a smaller number of possible VMs per LUN. Additionally, you must consider the already discussed LUN queue depth limits per ESX/ESXi Server and the storage system-specific limits.

11.4.6 ESX/ESXi filesystem considerations

VMware ESX/ESXi Server offers two possibilities to manage Virtual Disks: VMware ESX Server File System (VMFS) and Raw Device Mapping (RDM). VMFS is a clustered filesystem that allows concurrent access by multiple hosts. RDM is implemented as a proxy for a raw physical device. It uses a mapping file that contains metadata, and all disk traffic is redirected to the physical device. RDM can only be accessed by one virtual machine exclusively.

RDM offers two configuration modes: virtual compatibility mode and physical compatibility mode. When you use physical compatibility mode, all SCSI commands toward the virtual disk are passed directly to the device, which means that all physical characteristics of the underlying hardware become apparent. Within virtual compatibility mode, the virtual disk is mapped as a file within a VMFS volume, which allows advanced file locking support and the use of snapshots. Figure 11-8 compares both possible RDM configuration modes and VMFS.

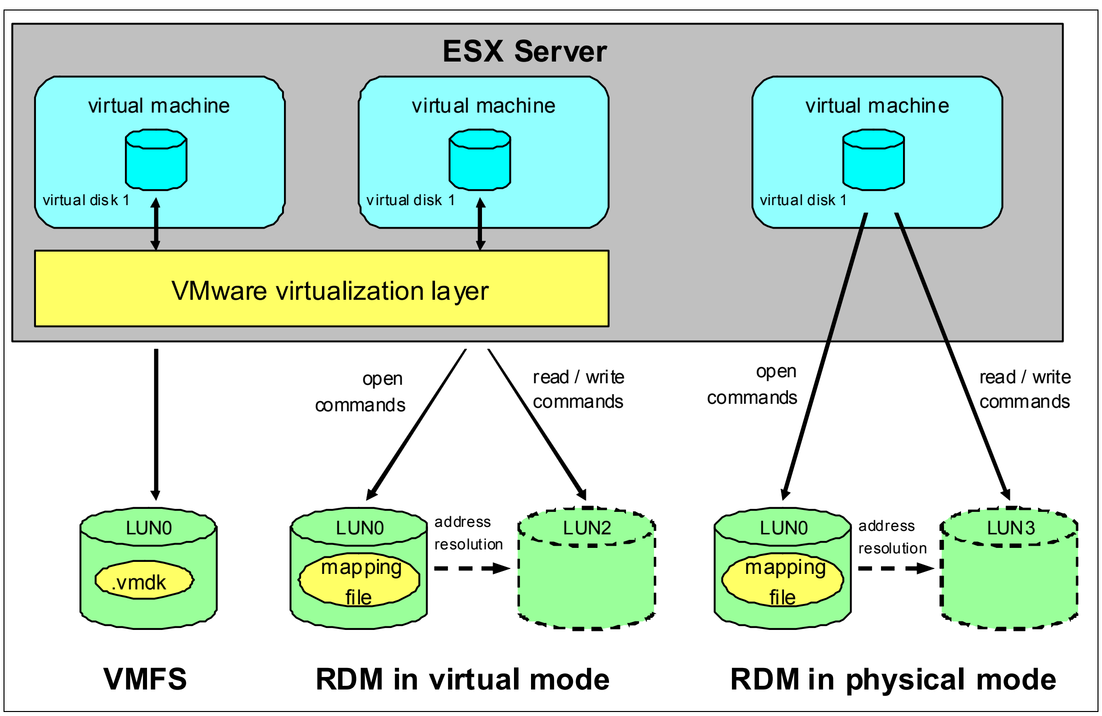


Figure 11-8 Comparison of RDM virtual and physical modes with VMFS

The implementations of VMFS and RDM imply a possible impact on the performance of the virtual disks; therefore, all three possible implementations are tested together with the DS8000. This section summarizes the outcome of those performance tests.

In general, the filesystem selection affects the performance in a limited manner:

- For random workloads, the measured throughput is almost equal between VMFS, RDM physical, and RDM virtual. Only for read requests of 32 KB, 64 KB, and 128 KB transfer sizes, both RDM implementations show a slight performance advantage (Figure 11-10 on page 439).
- For sequential workloads for all transfer sizes, we verified a slight performance advantage for both RDM implementations against VMFS. For all sequential write and certain read requests, the measured throughput for RDM virtual was slightly higher than for RDM physical mode. This difference might be caused by an additional caching of data within the virtualization layer, which is not used in RDM physical mode (Figure 11-9).

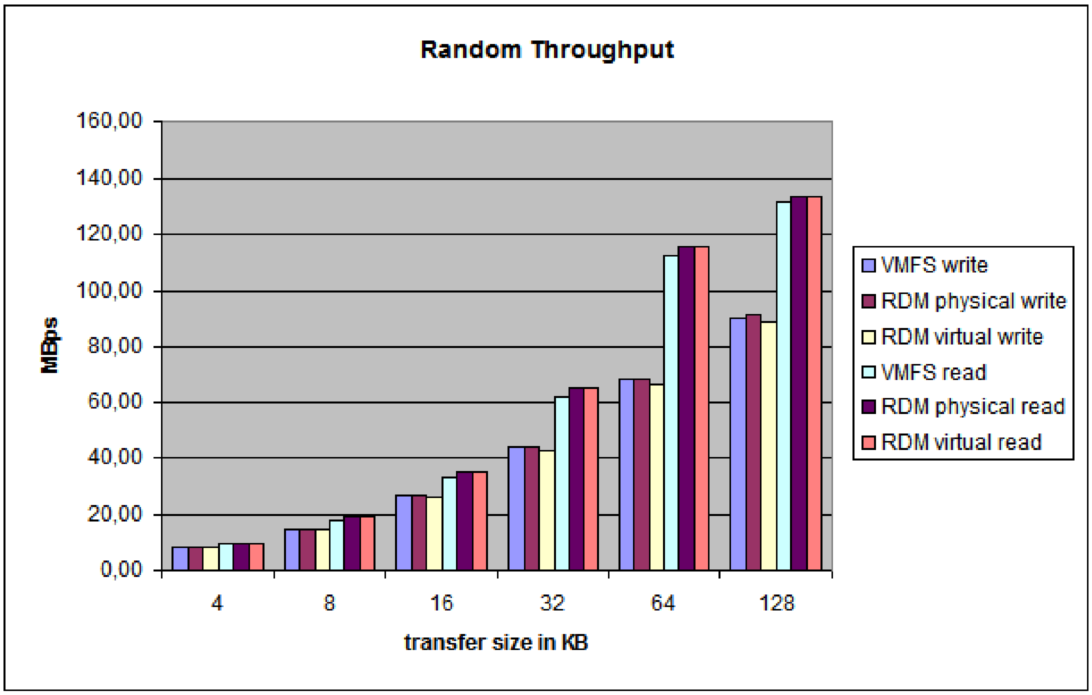


Figure 11-9 Result of random workload test for VMFS, RDM physical, and RDM virtual

Performance data varies: The performance data contained in Figure 11-9 and Figure 11-10 was obtained in a controlled, isolated environment at a specific point in time by using the configurations, hardware, and software levels available at that time. Actual results that might be obtained in other operating environments can vary. There is no guarantee that the same or similar results can be obtained elsewhere. The data is intended to help illustrate only how different technologies behave in relation to each other.

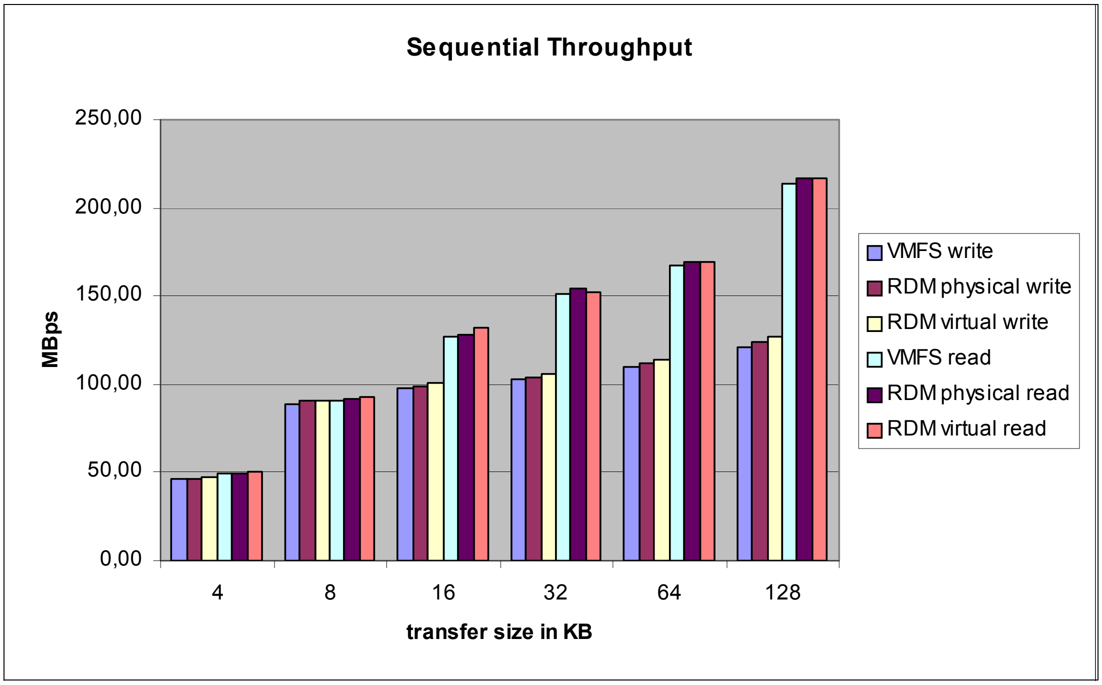


Figure 11-10 Result of sequential workload test for VMFS, RDM physical, and RDM virtual

The choice between the available filesystems, VMFS and RDM, has a limited influence on the data performance of the virtual machines. These tests verified a possible performance increase of about 2 - 3%.

11.4.7 Aligning partitions

In a RAID array, the smallest hardware unit used to build a logical volume or LUN is called a *stripe*. These stripes are distributed onto several physical drives in the array according to the RAID algorithm that is used. Usually, stripe sizes are much larger than sectors. For the DS8000, we use a 256 KB stripe size for RAID 5 and RAID 10 and 192 KB for RAID 6 in an Open Systems attachment. Thus, a SCSI request that intends to read a single sector in reality reads one stripe from disk.

When using VMware ESX/ESXi, each VMFS datastore segments the allocated LUN into blocks, which can be between 1 - 8 MB in size. The filesystem used by the virtual machine operating system optimizes I/O by grouping several sectors into one cluster. The cluster size usually is in the range of several KB.

If the VM operating system reads a single cluster from its virtual disk, at least one block (within VMFS) and all the corresponding stripes on physical disk need to be read. Depending on the sizes and the starting sector of the clusters, blocks, and stripes, reading one cluster might require reading two blocks and all of the corresponding stripes. Figure 11-11 on page 440 illustrates that in an *unaligned structure,* a single I/O request can cause additional I/O operations. Thus, an unaligned partition setup results in additional I/O that incurs a penalty on throughput and latency and leads to lower performance for the host data traffic.

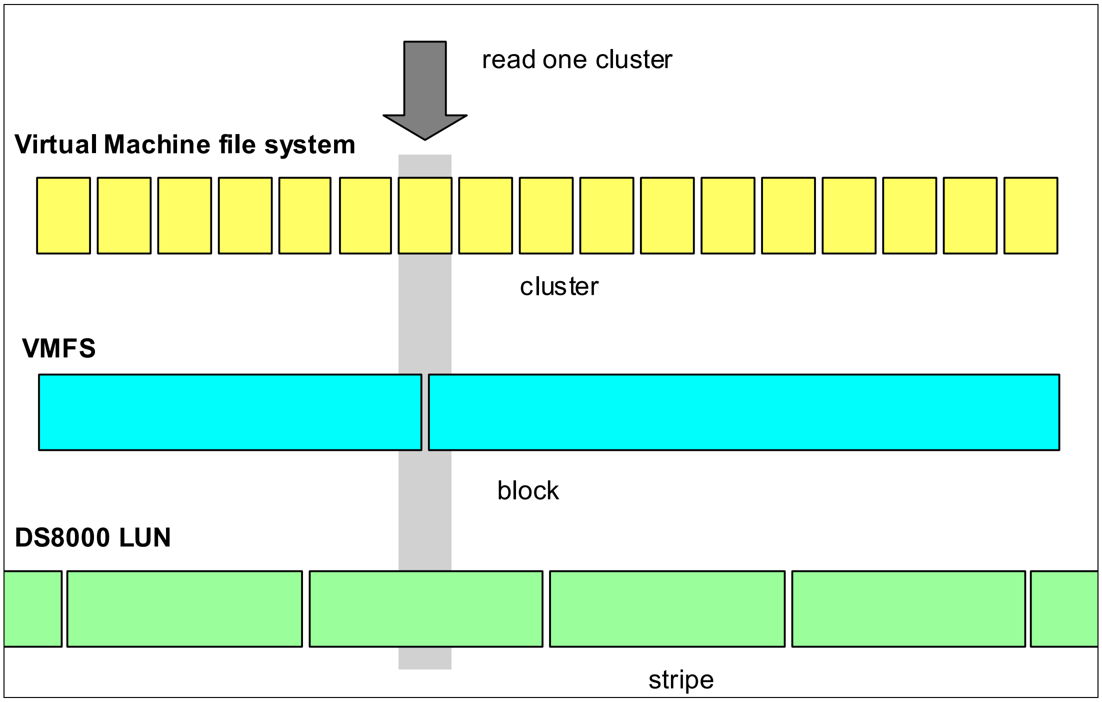


Figure 11-11 Processing of a data request in an unaligned structure

An aligned partition setup ensures that a single I/O request results in a minimum number of physical disk I/Os, eliminating the additional disk operations, which, in fact, results in an overall performance improvement.

Operating systems using the x86 architecture create partitions with a master boot record (MBR) of 63 sectors. This design is a relief from older BIOS code from personal computers that used cylinder, head, and sector addressing instead of Logical Block Addressing (LBA). The first track is always reserved for the MBR, and the first partition starts at the second track (cylinder 0, head 1, and sector 1), which is sector 63 in LBA. Also, in current operating systems, the first 63 sectors cannot be used for data partitions. The first possible start sector for a partition is 63.

In a VMware ESX/ESXi environment, because of the additional virtualization layer implemented by ESX/ESXi, this partition alignment must be performed for both layers: VMFS and the host filesystems. Because of that additional layer, the use of correctly aligned partitions is considered to have even a higher performance effect than in the usual host setups without an additional virtualization layer. Figure 11-12 on page 441 illustrates how a single I/O request is fulfilled within an aligned setup without causing additional physical disk I/O.

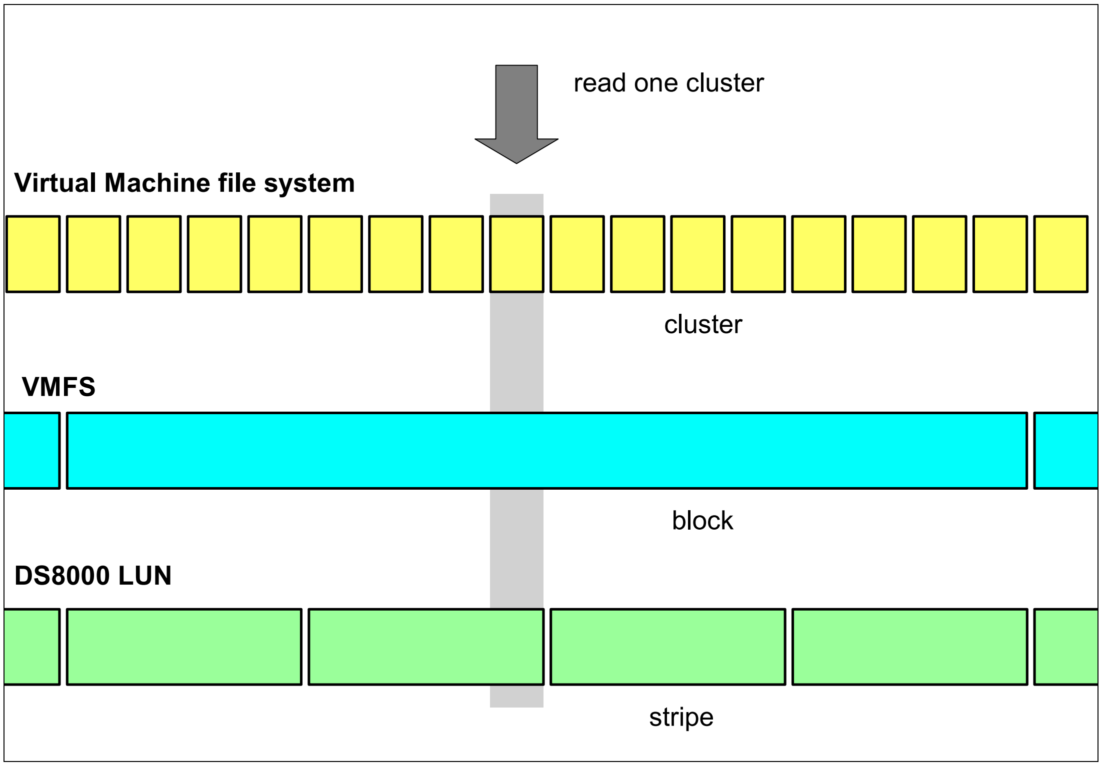


Figure 11-12 Processing a data request in an aligned structure

Partition alignment is a known issue in filesystems, but its effect on performance is somehow controversial. In performance lab tests, it turned out that in general all workloads show a slight increase in throughput when the partitions are aligned. A significant effect can only be verified on sequential workloads. Starting with transfer sizes of 32 KB and larger, we recognized performance improvements of up to 15%.

In general, aligning partitions can improve the overall performance. For random workloads, we only identified a slight effect. For sequential workloads, a possible performance gain of about 10% seems to be realistic. So, we can suggest that you align partitions especially for sequential workload characteristics.

Aligning partitions within an ESX/ESXi Server environment requires two steps. First, the VMFS partition needs to be aligned. And then, the partitions within the VMware guest system filesystems must be aligned as well for maximum effectiveness.

You can align the VMFS partition only when configuring a new datastore. When using the VI Client, the new partition is automatically configured to an offset of 128 sectors = 64 KB. But, in fact, this configuration is not ideal when using the DS8000 disk storage. Because the DS8000 uses larger stripe sizes, the offset must be configured to at least the stripe size. For RAID 5 and RAID 10 in Open Systems attachments, the stripe size is 256 KB, and it is a good approach to set the offset to 256 KB (or 512 sectors). You can configure an individual offset only from the ESX/ESXi Server command line.

Example 11-4 shows how to create an aligned partition with an offset of 512 using **fdisk**.

Example 11-4 Creating an aligned VMFS partition using fdisk

```
fdisk /dev/sdf                                        #invoke fdisk for /dev/sdf
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
```

```
until you decide to write them. After that, of course, the previous
content won't be recoverable.


The number of cylinders for this disk is set to 61440.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help):
n                                                             #create a new partition
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-61440, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-61440, default 61440):
Using default value 61440

Command (m for help): t                                       #set partitions system id
Selected partition 1
Hex code (type L to list codes): fb                           #fb = VWware VMFS volume
Changed system type of partition 1 to fb (Unknown)

Command (m for help): x                                       #enter expert mode

Expert command (m for help): b                                #set starting block number
Partition number (1-4): 1
New beginning of data (32-125829119, default 32): 512         #partition offset set to 512

Expert command (m for help): w                                #save changes
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

fdisk -lu /dev/sdf                                            #check the partition config

Disk /dev/sdf: 64.4 GB, 64424509440 bytes
64 heads, 32 sectors/track, 61440 cylinders, total 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sdf1           512 125829119  62914304   fb  Unknown
```

Then, you must create a VMFS filesystem within the aligned partition by using the **vmkfstools** command as shown in Example 11-5.

Example 11-5 Creating a VMFS volume by using vmkfstools

```
vmkfstools -C vmfs3 -b 1m -S LUNO vmhba2:0:0:1
Creating vmfs3 file system on "vmhba2:0:0:1" with blockSize 1048576 and volume label
"LUNO".
Successfully created new volume: 490a0a3b-cabf436e-bf22-001a646677d8
```

As the last step, all the partitions at the virtual machine level must be aligned as well. This task needs to be performed from the operating system of each VM by using the available tools. For example, for Windows, use the **diskpart** utility as shown in Example 11-6. Windows allows you to align basic partitions only, and the offset size is set in KB (not in sectors).

Example 11-6 Creating an aligned NTFS partition by using diskpart

```
DISKPART> create partition primary align=256

DiskPart succeeded in creating the specified partition.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
* Partition 1    Primary             59 GB   256 KB

DISKPART>
```

You can obtain additional information about aligning VMFS partitions and the performance effects from the document *VMware Infrastructure 3: Recommendations for aligning VMFS partitions*:

http://www.vmware.com/pdf/esx3_partition_align.pdf

Also, see *Performance Best Practices for VMware vSphere 4.0*:

http://www.vmware.com/pdf/Perf_Best_Practices_vSphere4.0.pdf

11.5 Tuning virtual machines

After the ESX/ESXi Server environment is tuned to optimal performance, the individual virtual machines and their operating systems must be examined more closely. In general, all the information described in the chapters for the Windows and Linux operating systems also applies to those environments where these operating systems run in a virtual machine in an ESX/ESXi environment.

Due to the additional virtualization layer implemented by ESX/ESXi Server, it is necessary to pay attention to the following information:

- ESX/ESXi Server emulates either a BusLogic or an LSI Logic SCSI adapter. The specifics of the implementation of the SCSI driver that is used inside the virtual machine operating system can affect disk I/O performance. For the BusLogic adapters, VMware provides a driver customized for Windows that is suggested for high performance. The BusLogic driver is part of the VMware tools and is installed automatically when the VMware tools are installed.
- In virtual machines that run Windows using the LSI Logic driver, I/Os larger than 64 KB might be split into multiple I/Os of a maximum size of 64 KB, which might negatively affect I/O performance. You can improve I/O performance by editing the registry setting: `HKLM\SYSTEM\CurrentControlSet\Services\Symmpi\Parameters\Device\MaximumSGList`.

  For further information, see "Large Block Size Support" at this website:

  http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&docType=kc&externalId=9645697&sliceId=1

12

Performance considerations for Linux

This chapter describes the monitoring and tuning tools and techniques that can be used with Linux systems to optimize throughput and performance when attaching the DS8000.

We also describe the supported distributions of Linux when you use the DS8000, as well as the tools that can be helpful for the monitoring and tuning activities:

- Linux disk I/O architecture
- Host bus adapter (HBA) considerations
- Multipathing
- Software RAID functions
- Logical Volume Manager (LVM)
- Disk I/O schedulers
- Filesystem considerations

12.1 Supported platforms and distributions

Linux tends to be the only operating system that is available for almost all hardware platforms. For DS8000 attachment, IBM currently supports Linux on following platforms:

- On x86-based servers in 32-bit and 64-bit mode
- On System p® servers in 32-bit and 64-bit mode
- On System z servers in 31-bit and 64-bit mode
- As a guest that runs in VMware ESX
- As a guest that runs in z/VM

IBM currently supports the following major Linux distributions:

- Red Hat Enterprise Linux (RHEL) 4, 5, and 6
- Novell SuSE Linux Enterprise (SLES) 9, 10, and 11

For further clarification and the *most current* information about supported Linux distributions and hardware prerequisites, see the System Storage Interoperation Center (SSIC) website:

http://www.ibm.com/systems/support/storage/config/ssic

Further information about supported kernel versions and additional restrictions can be obtained from the IBM Subsystem Device Driver for Linux website:

http://www-01.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S4000107

This chapter introduces the relevant logical configuration concepts needed to attach Linux operating systems to a DS8000 and focuses on performance relevant configuration and measuring options. For further information about hardware-specific Linux implementation and general performance considerations about the hardware setup, see the following documentation:

- For a general Linux implementation overview:

  Linux Handbook A Guide to IBM Linux Solutions and Resources, SG24-7000

- For x86-based architectures:
  - *Tuning IBM System x Servers for Performance*, SG24-5287
  - *Tuning Red Hat Enterprise Linux on IBM eServer xSeries Servers*, REDP-3861
  - *Tuning SUSE LINUX Enterprise Server on IBM eServer xSeries Servers*, REDP-3862
- For System p hardware:
  - *Virtualizing an Infrastructure with System p and Linux*, SG24-7499
  - *Tuning Linux OS on System p The POWER Of Innovation*, SG24-7338
- For System z hardware:
  - *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
  - *Linux for IBM System z9 and IBM zSeries*, SG24-6694
  - *z/VM and Linux on IBM System z*, SG24-7492

12.2 Linux disk I/O architecture

Before describing relevant disk I/O-related performance topics, this section briefly introduces the Linux disk I/O architecture. We look at the Linux disk I/O subsystem to better understand the components that have a major effect on system performance.

The architecture that we describe applies to Open Systems servers attached to the DS8000 by using the Fibre Channel Protocol (FCP). For Linux on System z with extended count key data (ECKD) (Fibre Channel connection (FICON) attached) volumes, a different disk I/O setup applies. For further information about disk I/O setup and configuration for System z, see the IBM Redbooks publication *Linux for IBM System z9 and IBM zSeries*, SG24-6694.

12.2.1 I/O subsystem architecture

Figure 12-1 illustrates the I/O subsystem architecture.

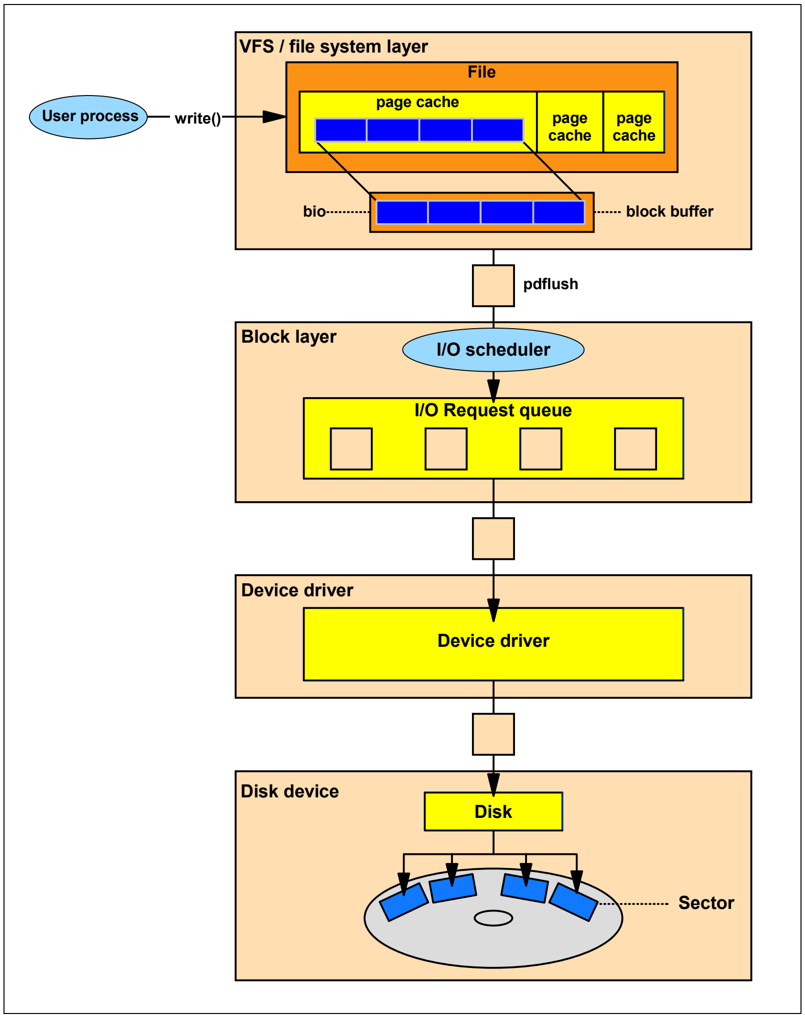


Figure 12-1 I/O subsystem architecture

For a quick overview of overall I/O subsystem operations, we use an example of writing data to a disk. The following sequence outlines the fundamental operations that occur when a disk-write operation is performed, assuming that the file data is on sectors on disk platters, that it was read, and is on the page cache:

1. A process requests to write a file through the **`write()`** system call.
2. The kernel updates the *page cache* mapped to the file.
3. A *pdflush* kernel thread takes care of flushing the page cache to disk.
4. The filesystem layer puts each block buffer together to a *bio* struct (see 12.2.3, “Block layer” on page 449) and submits a write request to the block device layer.
5. The block device layer gets requests from upper layers and performs an *I/O elevator* operation and puts the requests into the I/O request queue.

6. A device driver, such as Small Computer System Interface (SCSI) or other device-specific drivers, takes care of the write operation. The SCSI subsystem contains three layers: upper layer (block, tape, and generic), midlayer (common SCSI operations), and device driver layer. See 12.2.4, “I/O device driver” on page 450.
7. A disk device firmware performs hardware operations, such as seek head, rotation, and data transfer to the sector on the platter.

This sequence is simplified, because it reflects only I/Os to local physical disks (a SCSI disk attached via a native SCSI adapter). Storage configurations that use additional virtualization layers and SAN attachment require additional operations and layers, such as in the DS8000 storage system.

12.2.2 Cache and locality of reference

Achieving a high cache hit rate is the key for performance improvement. In Linux, the technique called “*locality of reference*” is used. This technique is based on the following principles:

- The data most recently used has a high probability of being used in the near future (*temporal locality*).
- The data that resides close to the data, which was used, has a high probability of being used (*spatial locality*).

Figure 12-2 illustrates this principle.

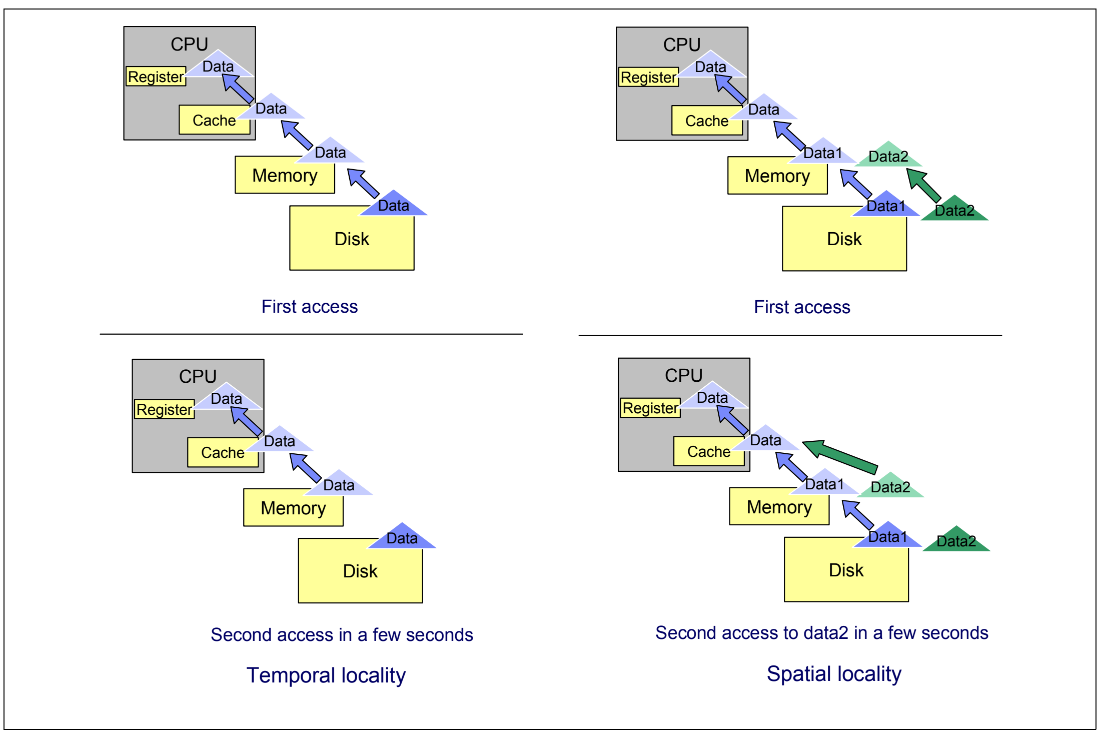


Figure 12-2 Locality of reference

Linux uses this principle in many components, such as page cache, file object cache (i-node cache and directory entry cache), and read ahead buffer.

Flushing a dirty buffer

When a process reads data from disk, the data is copied to memory. The process and other processes can retrieve the same data from the copy of the data cached in memory. When a process tries to change the data, it is changed in memory first. At this time, the data on disk and the data in memory are not identical, and the data in memory is referred to as a *dirty buffer*. The dirty buffer must be synchronized to the data on disk as soon as possible, or the data in memory can be lost if a sudden crash occurs.

The synchronization process for a dirty buffer is called *flush*. In the Linux kernel 2.6 implementation, the *pdflush* kernel thread is responsible for flushing data to the disk. The flush occurs on a regular basis (kupdate) and when the proportion of dirty buffers in memory exceeds a certain threshold (bdflush). The threshold is configurable in the `/proc/sys/vm/dirty_background_ratio` file.

The operating system synchronizes the data regularly. But with large amounts of system memory, it might keep updated data for several days. This delay is unsafe for the data in a failure. To avoid this situation, we advise that you use the **sync** command. When invoking the **sync** command, all changes and records are updated on the disks and all buffers are cleared. Periodic usage of **sync** is necessary in transaction processing environments that frequently update the same dataset file, which is intended to stay in memory. Data synchronization can be set to frequent updates automatically, but the **sync** command is useful in situations when large data movements are required and copy functions are involved.

> **Important:** Remember to invoke the **sync** command after the data synchronization in an application before issuing a FlashCopy and starting remote mirror initial data synchronization.

12.2.3 Block layer

The block layer handles all the activity related to block device operation (see Figure 12-1 on page 447). The key data structure in the block layer is the *bio* structure. The bio structure is an interface between the filesystem layer and the block layer. When the kernel, in the form of a filesystem, the virtual memory subsystem, or a system call, decides that a set of blocks must be transferred to or from a block I/O device, it puts together a bio structure to describe that operation. That structure is then handed to the block I/O code, which merges it into an existing request structure or, if needed, creates a new one. The bio structure contains everything that a block driver needs to carry out the request without reference to the user-space process that caused that request to be initiated.

When a write is performed, the filesystem layer tries to write to the page cache, which is made up of block buffers. It makes up a bio structure by putting the contiguous blocks together and then sends the bio to the block layer (see Figure 12-1 on page 447).

The block layer handles the bio request and links these requests into a queue called the *I/O request queue*. This linking operation is called *I/O elevator* or *I/O scheduler*. In Linux kernel 2.6 implementations, four types of I/O elevator algorithms are available.

The Linux kernel 2.6 employs a new I/O elevator model. The Linux kernel 2.4 used a single, general-purpose I/O elevator, but the Linux kernel 2.6 offers a choice of four elevators. Because the Linux operating system can be used for a wide range of tasks, both I/O devices and workload characteristics change significantly. A notebook computer probably has different I/O requirements than a 10000 user database system. To accommodate these differences, four I/O elevators are available. Further discussion about I/O elevator implementation and tuning is discussed in 12.3.4, “Tuning the disk I/O scheduler” on page 456.

12.2.4 I/O device driver

The Linux kernel takes control of devices using a *device driver*. The device driver is a separate kernel module and is provided for each device (or group of devices) to make the device available for the Linux operating system. After the device driver is loaded, it runs as a part of the Linux kernel and takes full control of the device. We describe SCSI device drivers.

SCSI

The Small Computer System Interface (SCSI) is the most commonly used I/O device technology, especially in the enterprise server environment. In Linux kernel implementations, SCSI devices are controlled by device driver modules. They consist of the following types of modules (Figure 12-3):

- Upper level drivers: sd_mod, sr_mod (SCSI-CDROM), st (SCSI Tape), and sq (SCSI generic device)

  They provide functionality to support several types of SCSI devices, such as SCSI CD-ROM, and SCSI tape.

- Middle level driver: scsi_mod

  It implements SCSI protocol and common SCSI functionality.

- Low-level drivers

  They provide lower-level access to each device. A low-level driver is specific to a hardware device and is provided for each device, for example, ips for the IBM ServeRAID controller, qla2300 for the QLogic HBA, and mptscsih for the LSI Logic SCSI controller.

- Pseudo driver: ide-scsi

  It is used for IDE-SCSI emulation.

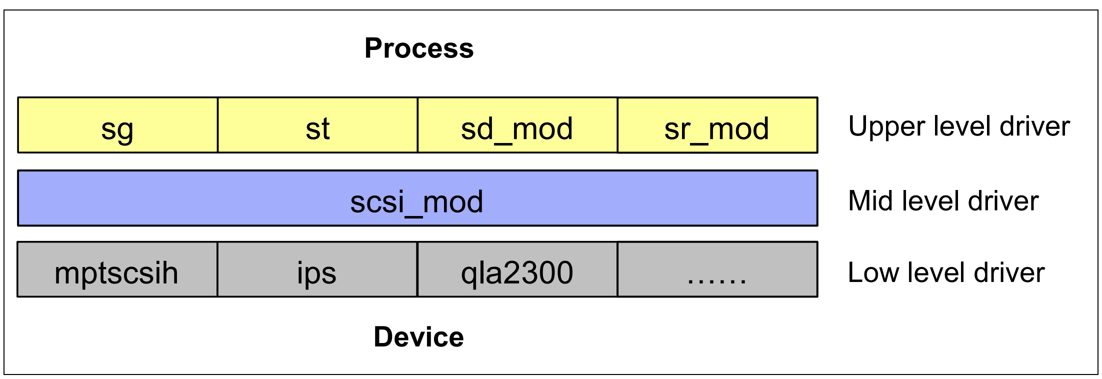


Figure 12-3 Structure of SCSI drivers

If specific functionality is implemented for a device, it must be implemented in device firmware and the low-level device driver. The supported functionality depends on which hardware you use and which version of device driver you use. The device must also support the desired functionality. Specific functions are usually tuned by a device driver parameter.

12.3 Specific configuration for storage performance

Many specific parameters influence the whole system performance, as well as the performance for a specific application, which also applies to Linux systems. The focus of this chapter is disk I/O performance only, thus, we do not discuss the influence of cpu usage, memory usage, and paging, as well as specific performance tuning possibilities for these areas.

For further general performance and tuning recommendations, see *Linux Performance and Tuning Guidelines*, REDP-4285.

12.3.1 Host bus adapter for Linux

IBM supports several host bus adapters (HBAs) in many possible configurations. To confirm that a specific HBA is supported by IBM, check the latest information in the System Storage Interoperation Center (SSIC):

http://www.ibm.com/systems/support/storage/config/ssic/index.jsp

For each HBA, there are BIOS levels and driver versions available. The supported versions for each Linux kernel level, distribution, and related information are available from the following link:

http://www.ibm.com/systems/support/storage/config/hba/index.wss

To configure the HBA correctly, see the *IBM TotalStorage DS8000: Host Systems Attachment Guide*, SC26-7625, which includes detailed procedures and suggested settings. Also, read the readme files and manuals of the driver, BIOS, and HBA.

With each HBA driver, you can configure several parameters. The list of available parameters depends on the specific HBA type and driver implementation. If these settings are not configured correctly, it might affect performance or the system might not work correctly.

You can configure each parameter as either temporary or persistent. For temporary configurations, you can use the **modprobe** command. Persistent configuration is performed by editing the following file (based on distribution):

- `/etc/modprobe.conf` for RHEL
- `/etc/modprobe.conf.local` for SLES

To set the queue depth of an Emulex HBA to 20, add the following line to `modprobe.conf(.local)`:

```
options lpfc lpfc_lun_queue_depth=20
```

Specific HBA types support a failover on the HBA level, for example, QLogic HBAs. HBA level multipathing is generally not supported for the DS8000. When using Device Mapper - Multipath I/O (DM-MP) or Subsystem Device Driver (SDD) for multipathing, this failover on the HBA level needs to be disabled. HBA level path failover is disabled by default in new Linux distributions, although it is enabled in SLES9. To disable failover on a QLogic qla2xxx adapter, add the following line to `modeprobe.conf(.local)`:

```
options qla2xxx ql2xfailover=0
```

For performance reasons, the parameters *queue depth* and several timeout and retry parameters in path errors can be interesting.

By changing the queue depth, you can queue more outstanding I/Os on the adapter level, which can in certain configurations have a positive effect on throughput. However, increasing the queue depth cannot be generally advised, because it can slow performance or cause delays, depending on the actual configuration. Thus, the complete setup needs to be checked carefully before adjusting the queue depth. Increasing queue depth is good enough for the sequential large block write workloads as well as for some sequential read workloads. Random workloads do not benefit much from the increased queue depth values. Indeed, you can only notice a slight improvement in performance after the queue depth is increased. However, the improvement might b greater if other methods of optimization are used. You can monitor the necessity of the queue depth increase using the **iostat -kx** command (Example 12-1). We review the **iostat** tool in "The iostat command" on page 470.

Example 12-1 Output of the iostat -kx command (output truncated)

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,00    0,00    3,12    4,75    0,00   92,13
Device:         rrqm/s   wrqm/s     r/s     w/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
dm-0              0,00 201428,00    0,00 1582,00     0,00 809984,00  1024,00   137,88   87,54   0,63 100,00
dm-1              0,00     0,00    0,00    0,00     0,00     0,00     0,00     0,00    0,00   0,00   0,00
```

Example 12-1shows the output of the **iostat -kx** command. You can see that the average queue size value might look high enough that the queue depth parameter might be increased. If you look closer at the statistics, you can see that the service time is low enough and the counter for write requests merged is high. Many write requests can be merged into fewer write requests before they are sent to the adapter. Service times for write requests less than 1 ms indicate that writes are cached. Taking all these observations into consideration, everything is fine with queue depth setting in this example.

For queue depth settings, follow these guidelines:

- The average queue size cannot be zero if any workload is present.
- Large block workloads always mean high average queue size values. So, queue size is not the only consideration for overall performance. Also, consider other values, such as response time.
- Read workloads must always have lower values of the average queue size than write workloads.
- Random small block workloads, read or write, do not benefit much from increased queue depth parameters. It is better to keep the queue depth parameter at the suggested default described in the manual for the HBA or multipathing driver.

High queue depth parameter values might lead to adapter overload situations, which can cause adapter resets and loss of paths. In turn, this situation might cause adapter failover and overload the rest of the paths. It might lead to situations where I/O is stuck for a period of time. We suggested that you decrease those values to allow the faster reaction of the multipathing module in path or adapter problems to avoid potential failures when you use DM-MP.

12.3.2 Multipathing in Linux

In a Linux environment, IBM supports two multipathing solutions for the DS8000:

- Subsystem Device Driver (SDD)
- Device Mapper - Multipath I/O (DM-MP)

Subsystem Device Driver (SDD) is a generic device driver designed to support the multipath configuration environments in the DS8000. SDD is provided and maintained by IBM for several operating systems, including Linux. SDD is the old approach of a multipathing solution. Starting with kernel Version 2.6, new, smarter multipathing support is available for Linux.

The Multipath I/O support included in Linux 2.6 kernel versions is based on *Device Mapper* (DM), a layer for block device virtualization that supports logical volume management, multipathing, and software RAID.

With Device Mapper, a virtual block device is presented where blocks can be mapped to any existing physical block device. By using the multipath module, the virtual block device can be mapped to several paths toward the same physical target block device. Device Mapper balances the workload of I/O operations across all available paths and detects defective links and fails over to the remaining links.

In general, IBM SDD is supported on older releases, such as SLES 8 and 9 and RHEL 3 and 4 versions only. For the new distribution releases SLES 10 and 11 and RHEL 5 and 6, DM-MP is the only supported multipathing solution. Only for SLES9 greater than and equal to SP3 and RHEL 4 greater than and equal to U6, there is both SDD and DM-MP support, but DM-MP is generally the suggested solution.

We advise that you use DM-MP if possible for your system configuration. DM-MP already is the preferred multipathing solution for most Linux 2.6 kernels. It is also available for 2.4 kernels, but it needs to be manually included and configured during kernel compilation. DM-MP is the required multipathing setup for LVM2.

Further information about supported distribution releases, kernel versions, and multipathing software is documented at the IBM Subsystem Device Driver for Linux website:

http://www.ibm.com/support/docview.wss?rs=540&context=ST52G7&uid=ssg1S4000107

DM-MP provides round-robin load balancing for multiple paths per LUN. The userspace component is responsible for automated path discovery and grouping, as well as path handling and retesting of previously failed paths. The framework is extensible for hardware-specific functions and additional load balancing or failover algorithms. For more information about DM-MP, see *IBM System Storage DS8000 Host Attachment and Interoperability,* SG24-8887.

IBM provides a device-specific configuration file for the DS8000 for the supported levels of RHEL and SLES. You must copy the device-specific section of the file to `/etc/multipath.conf` before the multipath driver and multipath tools are started. Example 12-2 sets default parameters for the scanned logical unit numbers (LUNs) and creates user friendly names for the multipath devices that are managed by DM-MP. Further configuration, adding aliases for certain LUNs, or blacklisting specific devices can be manually configured by editing this file.

Example 12-2 Contents of the multipath.conf file (truncated)

```
# These are the default settings for 2107 (IBM DS8000)
# Uncomment them if needed on this system
    device {
        vendor                  "IBM"
        product                 "2107900"
        path_grouping_policy    group_by_serial
    }
```

To get the device-specific configuration file, see these links:

https://www.ibm.com/support/docview.wss?uid=ssg1S4000107

ftp://ftp.software.ibm.com/storage/subsystem/linux/dm-multipath/3.01/SLES11/multipath.conf

Using DM-MP, you can configure various path failover policies, path priorities, and failover priorities. This type of configuration can be done for each device in the `/etc/multipath.conf` setup.

When using DM-MP, consider changing the default HBA timeout settings. In case a path fails, the default HBA timeout setting must be reported to the multipath module as fast as possible to avoid delay, because of I/O retries on the HBA level. Multipath I/O (MPIO) is then able to react quickly and fail over to one of the remaining healthy paths. This setting needs to be configured at the HBA level. Edit the file `/etc/modprobe.conf` or `/etc/modprobe.conf.local` (depending on the distribution). For example, you can use the following settings for an QLogic qla2xxx HBA (Example 12-3).

Example 12-3 HBA settings in /etc/modeprobe.conf.local

```
cat /etc/modprobe.conf.local
# please add local extensions to this file
options qla2xxx qlport_down_retry=1 ql2xfailover=0 ql2xretrycount=5
```

> **Timeout value:** Do not set low timeout and retry count settings on the HBA. They might lead to the unnecessary failover of the paths. For shortwave distances, it is sufficient to use the default timeout setting, which is about 70 seconds. However, heavy online transaction processing (OLTP) environments might require a lower timeout setting because of data protection, so setting a lower value of about 30 seconds might be required.

For further configuration and setup information, see the following publications:

- For SLES:
  http://www.suse.com/documentation/sles11/pdfdoc/stor_admin/stor_admin.pdf
- For RHEL:
  http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6-Beta/html/DM_Multipath/index.html
- Considerations and comparisons between IBM SDD for Linux and DM-MP:
  http://www.ibm.com/support/docview.wss?uid=ssg1S7001664&rs=555

12.3.3 Logical Volume Management

Logical Volume Management (LVM) is necessary in any modern enterprise operating system. With LVM, you can extend the functionality of the disk subsystem and manage data and logical volumes better. Also, the LVM functionality might be required for HA and DR solutions. These LMVs are available in Linux:

- Logical Volume Manager 2 (LVM2)
- Enterprise Volume Management System (EVMS)
- Software RAID functions

Logical Volume Manager (LVM2)

Starting with kernel version 2.6, Logical Volume Manager Version 2 (LVM2) is available and downward-compatible with the previous LVM. LVM 2 does not require any kernel patches. It uses the DM integrated in kernel 2.6. With kernel 2.6, only LVM2 is supported. Therefore, this section always refers to LVM Version 2.

With the use of LVM, you can configure *logical extents* on multiple physical drives or LUNs. Each LUN mapped from the DS8000 is divided into one or more *physical volumes* (PVs). Several of those PVs can be added to a logical *volume group* (VG), and later on, *logical volumes* (LVs) are configured out of a volume group. Each physical volume (PV) consists of a number of fixed-size *physical extents* (PEs). Similarly, each logical volume (LV) consists of a number of fixed-size logical extents (LEs). A logical volume (LV) is created by mapping logical extents (LEs) to physical extents (PEs) within a volume group. An LV can be created with a size from one extent to all available extents in a VG.

With LVM2, you can influence the way that LEs (for a logical volume) are mapped to the available PEs. With LVM linear mapping, the extents of several PVs are concatenated to build a larger logical volume. Figure 12-4 illustrates a logical volume across several physical volumes. With striped mapping, groups of contiguous physical extents, which are called *stripes*, are mapped to a single physical volume. With this functionality, it is possible to configure striping between several LUNs within LVM, which provides approximately the same performance benefits as the software RAID functions.

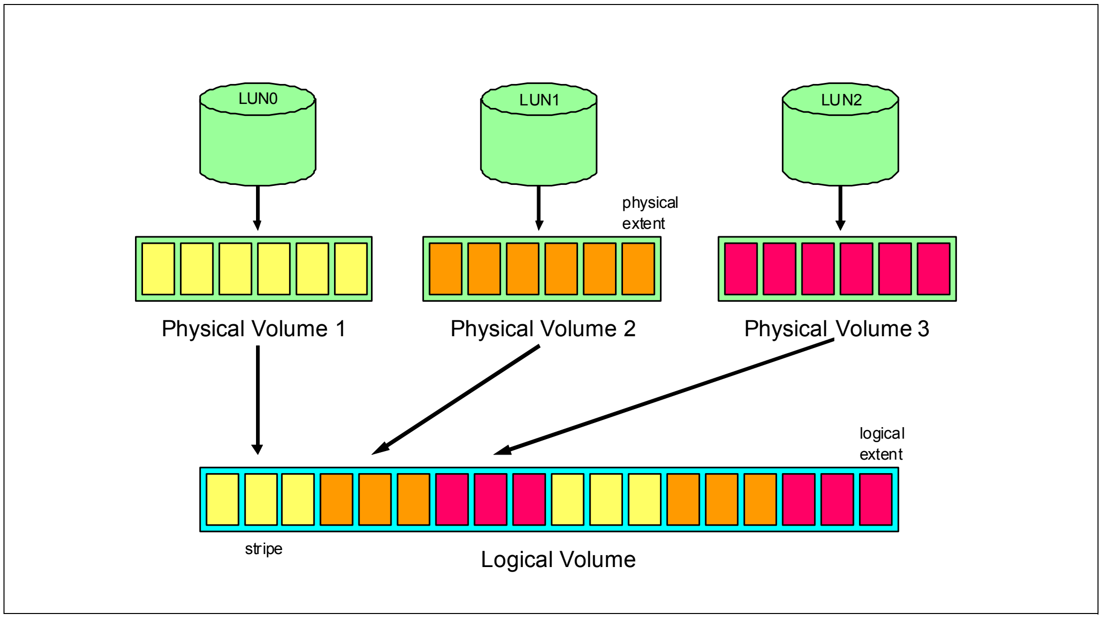


Figure 12-4 LVM striped mapping of three LUNs to a single logical volume

Furthermore, LVM2 offers additional functions and flexibility:

- Logical volumes can be resized during operation.
- Data from one physical volume can be relocated during operations, for example, in data migration scenarios.
- Logical volumes can be mirrored between several physical volumes for redundancy.
- Logical volume snapshots can be created for backup purposes.

Further documentation about LVM2 can be obtained from these websites:

http://sources.redhat.com/lvm2/

http://www.tldp.org/HOWTO/LVM-HOWTO/index.html

Enterprise Volume Management System (EVMS)

The Enterprise Volume Management System is an open source logical volume manager that integrates all aspects of storage management within an extensible framework. EVMS integrates disk partitioning, logical volume management (LVM2), Device Mapper multipath (DM-MP), software RAID management, and filesystem operations within a single management tool. EVMS provides a common management interface for the plugged-in functions of all included management layers.

EVMS offers a graphical user interface (EVMS GUI), a menu-driven interface (EVMS Ncurses), and a command-line interpreter (EVMS CLI). Depending on your needs, all three available user interfaces offer about the same functionality.

Using EVMS does not influence the performance of a Linux system directly, but with EVMS and the LVM, filesystem and Software RAID functions can be configured to influence the storage performance of a Linux system.

You can obtain more information about using EVMS from this website:

http://evms.sourceforge.net/user_guide/

EVMS used to be included with older SLES releases but was never officially supported for RHEL. It is not shipped with SLES11 anymore. The latest stable version of EVMS is 2.5.5, which was released on February 26, 2006. Development on EVMS was discontinued and LVM is the standard logical volume management solution for Linux.

Software RAID functions

Software RAID in the Linux 2.6 kernel distributions is implemented through the *md device driver*. This driver implementation is device-independent and therefore is flexible and allows many types of disk storage to be configured as a RAID array. Supported software RAID levels are RAID 0 (striping), RAID 1 (mirroring), RAID 5 (striping with parity), RAID 6 (striping with double parity), and RAID 10 (combination of mirroring and striping).

You can obtain further documentation about how to use the command-line RAID tools in Linux from this website:

http://tldp.org/HOWTO/Software-RAID-HOWTO-5.html

Modern practice

The role of LVM changed with the DS8000 microcode Release 6.2, Easy Tier, and I/O Priority Manager (IOPM). Earlier approaches with rank-based volume separation are not necessarily reasonable configurations anymore. With the DS8000 features, such as Easy Tier and IOPM, the use of hybrid or multi-tier pools with automated cross-tier and intra-tier management, as well as micro-tiering capabilities for optimum data relocation might offer excellent performance. These methods require less storage management effort than the use of many single-tier volumes striped with LVM. The use of LVM in these configurations might even result in decreasing real skew factors and inefficient Easy Tier optimization due to diluted heat distributions.

The preferred way to use LVM, Easy Tier, and IOPM is to use LVM concatenated logical volumes. This method might be useful when it is not possible to use volumes larger than 2 TB in DS8700/DS8800 (there are still some copy function limitations) or when implementing disaster recovery solutions that require LVM involvement. In other cases, follow the preferred practices described in Chapter 4, “Logical configuration performance considerations” on page 87 and Chapter 3, “Logical configuration concepts and terminology” on page 51. General LVM configurations must be with one logical volume per one DS8000 volume. With this approach, you can fully use the latest DS8000 Easy Tier and IOPM capabilities.

12.3.4 Tuning the disk I/O scheduler

The Linux kernel 2.6 employs a new I/O elevator model. The Linux kernel 2.4 used a single, general-purpose I/O scheduler. The Linux kernel 2.6 offers the choice of four schedulers or elevators.

The I/O scheduler forms the interface between the generic block layer and the low-level device drivers. Functions provided by the block layer can be utilized by the filesystems and the virtual memory manager to submit I/O requests to the block devices. These requests are transformed by the I/O scheduler to the low-level device driver. Red Hat Enterprise Linux AS 4 and SUSE Linux Enterprise Server 11 support four types of I/O schedulers.

You can obtain additional details about configuring and setting up I/O schedulers in *Tuning Linux OS on System p The POWER Of Innovation*, SG24-7338.

Descriptions of the available I/O schedulers

The following list shows the four available I/O schedulers:

- Deadline I/O scheduler

  The deadline I/O scheduler incorporates a per-request expiration-based approach and operates on five I/O queues. The idea behind I/O scheduler is that all read requests are satisfied within a specified time period. However, write requests do not have any specific deadlines. Web servers perform better when configured with deadline I/O scheduler and ext3.

- Noop I/O scheduler

  The Noop I/O scheduler is an I/O scheduler that performs and provides basic merging and sorting functions. In large I/O subsystems that incorporate RAID controllers and a vast number of contemporary physical disk drives called *TCQ drives*, the Noop I/O scheduler potentially can outperform the other three I/O schedulers as the workload increases.

- Completely Fair Queuing I/O scheduler

  The Completely Fair Queuing (CFQ) I/O scheduler is implemented on the concept of fair allocation of I/O bandwidth among all the initiators of I/O requests. The CFQ I/O scheduler strives to manage per-process I/O bandwidth and provide fairness at the level of process granularity. The sequential writes perform better when CFQ I/O scheduler is configured with eXtended Filesystem (XFS). The number of requests fetched from each queue is controlled by the cfq_quantum tunable parameter.

- Anticipatory I/O scheduler

  The anticipatory I/O scheduler attempts to reduce the per-thread read response time. It introduces a controlled delay component into the dispatching equation. The delay is invoked on any new read request to the device driver. File servers are found to perform better when the anticipatory I/O scheduler is configured with ext3. Sequential reads perform better when the anticipatory I/O scheduler is configured with XFS or ext3.

Selecting the correct I/O elevator for a selected type of workload

For most server workloads, either the Complete Fair Queuing (CFQ) elevator or the Deadline elevator is an adequate choice, because both of them are optimized for the multiuser and multi-process environment in which a typical server operates. Enterprise distributions typically default to the CFQ elevator. However, on Linux for IBM System z, the Deadline scheduler is favored as the default elevator. Certain environments can benefit from selecting a different I/O elevator. With Red Hat Enterprise Linux 5.0 and Novell SUSE Linux Enterprise Server 10, the I/O schedulers can now be selected on a per disk subsystem basis as opposed to the global setting in Red Hat Enterprise Linux 4.0 and Novell SUSE Linux Enterprise Server 9.

With the capability to have different I/O elevators per disk subsystem, the administrator now can isolate a specific I/O pattern on a disk subsystem (such as write intensive workloads) and select the appropriate elevator algorithm:

- Synchronous filesystem access

  Certain types of applications need to perform filesystem operations synchronously, which can be true for databases that might even use a raw filesystem or for large disk subsystems where caching asynchronous disk accesses is not an option. In those cases, the performance of the anticipatory elevator usually has the least throughput and the highest latency. The three other schedulers perform equally up to an I/O size of roughly 16 KB where the CFQ and the Noop elevators begin to outperform the deadline elevator (unless disk access is seek-intense).

- Complex disk subsystems

  Benchmarks show that the Noop elevator is an interesting alternative in high-end server environments. When using configurations with enterprise-class disk subsystems, such as the DS8000, the lack of ordering capability of the Noop elevator becomes its strength. It becomes difficult for an I/O elevator to anticipate the I/O characteristics of such complex subsystems correctly, so you might often observe at least equal performance at less overhead when using the Noop I/O elevator. Most large-scale benchmarks that use hundreds of disks most likely use the Noop elevator.

- Database systems

  Due to the seek-oriented nature of most database workloads, some performance gain can be achieved when selecting the deadline elevator for these workloads.

- Virtual machines

  Virtual machines, regardless of whether in VMware or VM for System z, usually communicate through a virtualization layer with the underlying hardware. So, a virtual machine is not aware of whether the assigned disk device consists of a single SCSI device or an array of Fibre Channel disks on a DS8000. The virtualization layer takes care of necessary I/O reordering and the communication with the physical block devices.

- CPU-bound applications

  While certain I/O schedulers can offer superior throughput, they can at the same time create more system overhead. The overhead that for instance the CFQ or deadline elevators cause comes from aggressively merging and reordering the I/O queue. Sometimes, the workload is not so much limited by the performance of the disk subsystem as by the performance of the CPU, for example, with a scientific workload or a data warehouse that processes complex queries. In these scenarios, the Noop elevator offers an advantage over the other elevators, because it causes less CPU overhead as shown on Figure 12-5 on page 459. However, when you compare CPU overhead to throughput, the deadline and CFQ elevators are still the best choices for most access patterns to asynchronous filesystems.

- Single Advanced Technology Attachment (ATA) or Serial Advanced Technology Attachment (SATA) disk subsystems

  If you choose to use a single physical ATA or SATA disk, for example, for the boot partition of your Linux system, consider using the anticipatory I/O elevator, which reorders disk writes to accommodate the single disk head in these devices.

12.3.5 Filesystems

The filesystems that are available for Linux are designed with different workload and availability characteristics. If your Linux distribution and the application allow the selection of a different filesystem, it might be worthwhile to investigate if Ext, Journal File System (JFS), ReiserFS, or eXtended File System (XFS) is the optimal choice for the planned workload. ReiserFS is more suited to accommodate small I/O requests. XFS and JFS are tailored toward large filesystems and large I/O sizes. Ext3 fits the gap between ReiserFS and JFS and XFS, because it can accommodate small I/O requests while offering good multiprocessor scalability.

The workload patterns JFS and XFS are best suited for high-end data warehouses, scientific workloads, large symmetric multiprocessor (SMP) servers, or streaming media servers. ReiserFS and Ext3 are typically used for file, web, or mail serving. For write-intense workloads that create smaller I/Os up to 64 KB, ReiserFS might have an edge over Ext3 with default journaling mode as seen in Figure 12-5. However, this advantage is only true for synchronous file operations.

An option to consider is the Ext2 filesystem. Due to its lack of journaling abilities, Ext2 outperforms ReiserFS and Ext3 for synchronous filesystem access regardless of the access pattern and I/O size. So, Ext2 might be an option when performance is more important than data integrity. But, the lack of journaling can result in an unrecoverable FS in certain circumstances. The Ext3 filesystem can also be configured without journaling in which case it is equal to Ext2 from a performance point of view.

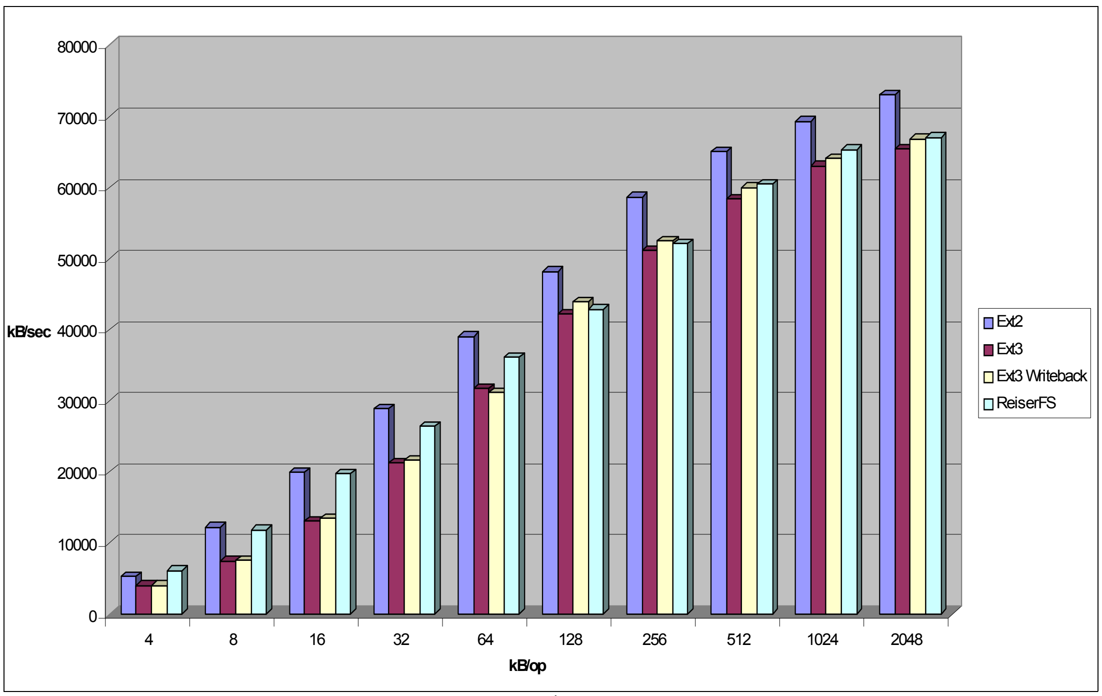


Figure 12-5 Random write throughput comparison[1] between Ext and ReiserFS (synchronous)

In the most common scenario of an asynchronous filesystem, ReiserFS most often delivers solid performance and outperforms Ext3 with the default journaling mode (data=ordered). However, Ext3 is equal to ReiserFS as soon as the default journaling mode is switched to writeback.

[1] The performance data contained in this figure was obtained in a controlled, isolated environment at a specific point in time by using the configurations, hardware, and software levels available at that time. Actual results that might be obtained in other operating environments can vary. There is no guarantee that the same or similar results can be obtained elsewhere. The data is intended only to help illustrate how different technologies behave in relation to each other.

Using ionice to assign I/O priority

A feature of the CFQ I/O elevator is the option to assign priorities on a process level. By using the **ionice** utility, you can restrict the disk subsystem utilization of a specific process:

- Idle: A process with the assigned I/O priority *idle* is granted access to the disk subsystems if no other processes with a priority of *best-effort* or higher request access to the data. This setting is useful for tasks that run when the system has free resources, such as the **updatedb** task.
- Best-effort: As a default, all processes, which do not request a specific I/O priority, are assigned to this class. Processes inherit eight levels of the priority of their CPU nice level to the I/O priority class.
- Real time: The highest available I/O priority is real time, which means that the respective process is always given priority access to the disk subsystem. The real time priority setting can also accept eight priority levels. Use caution when assigning a thread a priority level of real time, because this process can cause the other tasks to be unable to access the disk subsystem.

Ionice only has an effect when several processes compete for I/Os. If you use ionice to favor certain processes, other, maybe even essential, I/Os of other processes can suffer.

With the DS8000 I/O Priority Manager (IOPM) feature on the DS8700/DS8800 storage system, you must decide where to use I/O priority management. IOPM has following advantages:

- Provides the flexibility of many levels of priorities to be set
- Does not consume the resources on the server
- Sets real priority at the disk level
- Manages internal bandwidth access contention between several servers

With ionice, you can control the priorities for the use of server resources, that is, to manage access to the HBA, which is not possible with the DS8000 IOPM. However, this capability is limited to a single server only, and you cannot manage priorities at the disk system back end.

We suggest that you use IBM DS8000 IOPM in most of the cases for priority management. The operating system priority management can be used combined with IOPM. This combination provides the highest level of flexibility.

Access time updates

The Linux filesystem records when files are created, updated, and accessed. Default operations include updating the last-time-read attribute for files during reads and writes to files. Because writing is an expensive operation, eliminating unnecessary I/O can lead to overall improved performance. However, under most conditions, disabling file access time updates only yields a small performance improvement.

Mounting filesystems with the **noatime** option prevents inode access times from being updated. If file and directory update times are not critical to your implementation, as in a web serving environment, an administrator might choose to mount filesystems with the **noatime** flag in the `/etc/fstab` file as shown in Example 12-4. The performance benefit of disabling access time updates to be written to the filesystem ranges from 0 - 10% with an average of 3% for file server workloads.

Example 12-4 Update /etc/fstab file with noatime option set on mounted filesystems

```
/dev/sdb1 /mountlocation ext3 defaults,noatime 1 2
```

Selecting the journaling mode of the filesystem

Three journaling options for most filesystems can be set with the **data** option in the **mount** command. However, the journaling mode has the greatest effect on performance for Ext3 filesystems, so we suggest that you use this tuning option for the Red Hat default filesystem:

- data=journal

  This journaling option provides the highest form of data consistency by causing both file data and metadata to be journaled. It also has the highest performance overhead.

- data=ordered (default)

  In this mode, only metadata is written. However, file data is guaranteed to be written first. This setting is the default setting.

- data=writeback

  This journaling option provides the fastest access to the data at the expense of data consistency. The data is guaranteed to be consistent as the metadata is still being logged. However, no special handling of actual file data is done, which can lead to old data appearing in files after a system crash. The type of metadata journaling implemented when using the writeback mode is comparable to the defaults of ReiserFS, JFS, or XFS. The writeback journaling mode improves Ext3 performance especially for small I/O sizes as shown in Figure 12-6. The benefit of using writeback journaling declines as I/O sizes grow. Also, the journaling mode of your filesystem only affects write performance. Therefore, a workload that performs mostly reads (for example, a web server) does not benefit from changing the journaling mode.

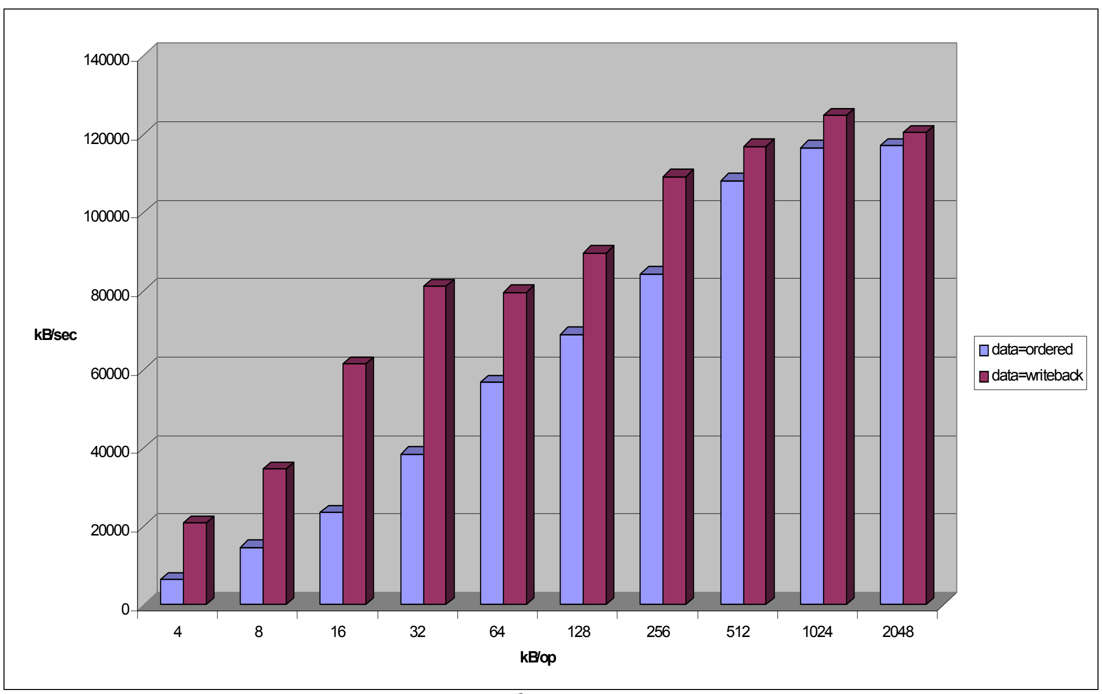


Figure 12-6 Random write performance impact[2] of data=writeback

[2] The performance data contained in this figure is obtained in a controlled, isolated environment at a specific point in time by using the configurations, hardware, and software levels available at that time. Actual results that might be obtained in other operating environments can vary. There is no guarantee that the same or similar results can be obtained elsewhere. The data is intended to help illustrate how different technologies behave in relation to each other.

There are three ways to change the journaling mode on a filesystem:

- When executing the **mount** command:

  `mount -o data=writeback /dev/sdb1 /mnt/mountpoint`

  /dev/sdb1 is the filesystem being mounted.

- Including it in the options section of the /etc/fstab file:

  `/dev/sdb1 /testfs ext3 defaults,data=writeback 0 0`

- If you want to modify the default **data=ordered** option on the root partition, change the /etc/fstab file. Then, execute the **mkinitrd** command to scan the changes in the /etc/fstab file and create an image. Update grub or lilo to point to the new image.

Blocksizes

The *blocksize*, the smallest amount of data that can be read or written to a drive, can have a direct impact on server performance. As a guideline, if your server handles many small files, a smaller blocksize is more efficient. If your server is dedicated to handling large files, a larger blocksize might improve performance. Blocksizes cannot be changed dynamically on existing filesystems, and only a reformat modifies the current blocksize. Most Linux distributions allow blocksizes between 1 K, 2 K, and 4 K. As benchmarks demonstrate, there is hardly any performance improvement to gain from changing the blocksize of a filesystem, so it is better to leave it at the default of 4 K. You can use the suggestions of the application vendor in addition.

12.4 Linux performance monitoring tools

This section introduces the commonly used performance measurement tools. Additionally, Linux offers several other performance measurement tools. Most of them are equal to those tools that are available in the UNIX operating systems.

12.4.1 Gathering configuration data

Prior to collecting the performance statistics, you must have the disk and path configuration data to be able to identify disks shown in the statistics and find the problems easily. The following suggestions are based on the use of DM-MP multipathing. For IBM SDD multipathing, refer to Example 10-1 on page 415 because the procedures are the same.

Discovering the disks in the system

First, locate all of the disks of the system. There are many ways to locate all the disks, but the simplest way is to use the **dmesg|grep** disk command. This command lists all the disks discovered during the boot process or during the disk discovery by an HBA driver. You can get the following output, as shown in Example 12-5.

Example 12-5 Command dmesg|grep disk output (output truncated)

```
[    1.839013] sd 0:2:0:0: [sda] Attached SCSI disk
[976053.023172] sd 15:0:1:0: [sdd] Attached SCSI disk
[976053.023178] sd 15:0:0:0: [sdb] Attached SCSI disk
[976053.030194] sd 15:0:0:1: [sdc] Attached SCSI disk
[976053.030300] sd 15:0:1:1: [sde] Attached SCSI disk
[976054.949359] sd 17:0:0:0: [sdf] Attached SCSI disk
[976054.951185] sd 17:0:0:1: [sdg] Attached SCSI disk
[976055.945078] sd 17:0:1:0: [sdh] Attached SCSI disk
[976055.946964] sd 17:0:1:1: [sdi] Attached SCSI disk
```

Nine disks are displayed. In this case, disk *sda* is the root disk of the system. You can verify with the **dmesg|grep sda** and **mount** commands as shown in Example 12-6.

Example 12-6 Checking the sda disk with the dmesg|grep sda command

```
#dmesg|grep sda
[    1.836698]  sda: sda1 sda2
[    1.839013] sd 0:2:0:0: [sda] Attached SCSI disk
[   12.169482] EXT3 FS on sda2, internal journal
[   29.832183] Adding 2104472k swap on /dev/sda1.  Priority:-1 extents:1 across:2104472k

# mount
/dev/sda2 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
```

In Example 12-6, you see that disk sda contains several partitions and the swap file. This disk is the root disk, which is proven by the output of the **mount** command.

The other disks in the system seem to be the DS8000 disks. You can verify with the **multipath -ll** command as shown in Example 12-7. The **multipath -ll** is deprecated. In current Linux versions, the **multipathd -k** interactive prompt is used to communicate with DM-MP. For more information, see *IBM System Storage DS8000 Host Attachment and Interoperability,* SG24-8887.

Example 12-7 Output of the multipath -ll command

```
# multipath -ll
mpathc (3600507630affc29f0000000000008607) dm-0 IBM,2107900
size=100G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 15:0:0:1 sdc 8:32  active ready running
  |- 15:0:1:1 sde 8:64  active ready running
  |- 17:0:0:1 sdg 8:96  active ready running
  `- 17:0:1:1 sdi 8:128 active ready running
mpathb (3600507630affc29f0000000000008603) dm-1 IBM,2107900
size=100G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 15:0:0:0 sdb 8:16  active ready running
  |- 15:0:1:0 sdd 8:48  active ready running
  |- 17:0:0:0 sdf 8:80  active ready running
  `- 17:0:1:0 sdh 8:112 active ready running
```

Example 12-7 shows that the device is a DS8000 volume with an active-active configuration. The LUNs have user-friendly names of `mpathb` and `mpathc` and device names of `dm-0` and `dm-1`, which appear in the performance statistics. The LUN IDs in the parentheses, 3600507630affc29f000000000000**8603** and 3600507630affc29f000000000000**8607**, contain the ID of the logical volume in the DS8000 in the last four digits of the whole LUN ID: 8603 and 8607. The output also indicates that the size of each LUN is 100 GB. There is no hardware handler assigned to this device, and that I/O is supposed to be queued forever in the event that no paths are available. All paths group are in the `active` state, which means that all paths in this group carry all the I/Os to the storage. All paths to the device (LUN) are in `active ready` mode. There are four paths per LUN presented in the system as `sdX` devices where *X* is the index number of the disk.

Discovering the HBA information

All information about the devices is under `/sys/class`. Information about the HBAs and the ports is in the `/sys/class/fc_host` folder (Example 12-8 on page 464).

Example 12-8 Contents of the /sys/class/fc_host folder

```
# ls /sys/class/fc_host
host13  host14  host15  host16  host17  host18  host3  host4  host5  host6
```

Example 12-8 shows the contents of the `/sys/class/fc_host` folder. This system has 10 FC ports. To get the information about each port, use the following script, as shown in Example 12-9.

Example 12-9 Getting HBA port information with the script

```
for i in `seq 0 18` ;
    do echo #####Host $i ######;
    cat /sys/class/fc_host/host$i/port_name;
    cat /sys/class/fc_host/host$i/port_state;
    cat /sys/class/fc_host/host$i/port_type;
    cat /sys/class/fc_host/host$i/speed;
    cat /sys/class/fc_host/host$i/supported_speeds;
    done
```

This script simplifies the information gathering. It uses 18 as the maximum index for the FC port, which is true for Example 12-8. You might have your own indexing. See Example 12-10 for the script output.

Example 12-10 Script output

```
0x100000051eb2887b
Linkdown
Unknown
unknown
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit

0x100000051eb2887c
Linkdown
Unknown
unknown
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit

0x100000051e8c23b0
Linkdown
Unknown
unknown
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit

0x100000051e8c23b1
Linkdown
Unknown
unknown
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit

0x210000c0dd17f1a9
Online
Unknown
unknown
10 Gbit

0x210000c0dd17f1ab
Online
Unknown
unknown
```

```
10 Gbit

0x21000024ff2d0f8c
Online
NPort (fabric via point-to-point)
8 Gbit
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit

0x21000024ff2d0f8d
Online
NPort (fabric via point-to-point)
8 Gbit
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit

0x21000024ff2d0ed4
Online
NPort (fabric via point-to-point)
8 Gbit
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit

0x21000024ff2d0ed5
Online
NPort (fabric via point-to-point)
8 Gbit
1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit
```

In Example 12-10 on page 464, there are four HBA FC ports that are offline and four FC ports that are online. There are also two 10 Gb ports on the system.

Another way to discover the connection configuration is to use the **systool -av -c fc_host** command as shown in Example 12-11. This command displays extended output and information about the FC ports. However, this command might not be available in all Linux distributions.

Example 12-11 Output of the port information with systool -av -c fc_host (only one port shown)

```
Class Device = "host18"
  Class Device path =
"/sys/devices/pci0000:05/0000:05:00.0/0000:06:00.1/host18/fc_host/host18"
    fabric_name         = "0x100000051e470807"
    issue_lip           = <store method only>
    max_npiv_vports     = "254"
    node_name           = "0x20000024ff2d0ed5"
    npiv_vports_inuse   = "0"
    port_id             = "0x35e080"
    port_name           = "0x21000024ff2d0ed5"
    port_state          = "Online"
    port_type           = "NPort (fabric via point-to-point)"
    speed               = "8 Gbit"
    supported_classes   = "Class 3"
    supported_speeds    = "1 Gbit, 2 Gbit, 4 Gbit, 8 Gbit"
    symbolic_name       = "QLE2562 FW:v5.03.02 DVR:v8.03.01.06.11.1-k8"
    system_hostname     = ""
    tgtid_bind_type     = "wwpn (world wide port name)"
    uevent              =
    vport_create        = <store method only>
    vport_delete        = <store method only>
   Device = "host18"
    Device path = "/sys/devices/pci0000:05/0000:05:00.0/0000:06:00.1/host18"
      edc                 = <store method only>
```

```
    fw_dump             =
    nvram               = "ISP "
    optrom              =
    optrom_ctl          = <store method only>
    reset               = <store method only>
    sfp                 = ""
    uevent              = "DEVTYPE=scsi_host"
    vpd                 = "?("
```

Example 12-11 on page 465 shows the output for one FC port:

- The device file for this port is `host18`.
- This port has a worldwide port name (WWPN) of 21000024ff2d0ed5, which appears in the fabric.
- This port is connected at 8 Gb/sec.
- It is a QLogic card with firmware version 5.03.02.

Example 12-10 on page 464 shows all the information for all the FC ports on the system that are connected to the SAN with four QLogic HBA FC ports and their WWPNs shown in bold.

Discovering the configuration from the disk system

After you collect all the necessary information on the server side, collect the disk layout on the disk system side:

- Rank and extent pool allocation
- Physical disk and array information
- Volume group and host connection information

Locate volume ranks

Our example shows two disks with IDs 8607 and 8603. We show you how to get the configuration data for them. We use only disk 8607 in this example, because the same steps apply for disk 8603.

The rank configuration for the disk can be shown with the **`showfbvol -rank`** ***`VOL_ID`*** command, where *VOL_ID* is 8607 in this example (Example 12-12).

Example 12-12 Locating volume ranks

```
> showfbvol -rank 8607
Name            -
ID              8607
accstate        Online
datastate       Normal
configstate     Normal
deviceMTM       2107-900
datatype        FB 512
addrgrp         8
extpool         P4
exts            100
captype         DS
cap (2^30B)     100.0
cap (10^9B)     -
cap (blocks)    209715200
volgrp          V17
ranks           3
dbexts          0
sam             Standard
repcapalloc     -
```

```
eam            managed
reqcap (blocks) 209715200
realextents    100
virtualextents 0
migrating      0
perfgrp        PG0
migratingfrom  -
resgrp         RG0
==============Rank extents==============
rank extents
============
R1        85
R17        7
R18        8
```

Example 12-12 on page 466 shows the properties of the logical volume. The following information can be discovered for the volume:

- Occupies three ranks (R1, R17, and R18)
- Belongs to volume group V17
- Is 100 GB in size
- Uses an extent allocation method (EAM) that is *managed* by Easy Tier
- Uses a standard storage allocation method (SAM)
- Is a regular, non-thin provisioned volume

Collecting physical disk information

With the information about used ranks of the volume collected, we need to collect the information about the used physical disk arrays and RAID types. We use the following **showrank** and **showarray** commands.

Example 12-13 Gathering disk information

```
dscli> showrank r1
IBM.2107-75TV181
ID             R1
SN             -
Group          0
State          Normal
datastate      Normal
Array          A1
RAIDtype       5
extpoolID      P4
stgtype        fb
exts           1574
usedexts       119
widearrays     0
nararrays      1
trksize        128
strpsize       512
strpesize      0
extsize        16384
encryptgrp     -
migrating(in)  0
migrating(out) 0

dscli> showarray a1
Array          A1
SN             ATV1814D95C074K
State          Assigned
```

```
datastate      Normal
RAIDtype       5 (6+P+S)
arsite         S2
Rank           R1
DA Pair        0
DDMcap (10^9B) 300.0
DDMRPM         65000
Interface Type SAS
interrate      6.0 Gb/sec
diskclass      SSD
encrypt        unsupported
```

Example 12-13 on page 467 shows how to reveal the physical disk and array information. The properties of the rank provide the array number, and the array properties provide the disk information and the RAID type. In this case, rank R1 is located on array A1, which consists of 300 GB solid-state drives (SSDs) in a RAID 5 configuration. The same procedure can be used for the other ranks of the volume.

Collecting volume group and host connection information

From Example 12-12 on page 466, we know that volume 8607 participates in volume group V17. Example 12-14 shows how to get the volume group properties and host connection information.

Example 12-14 Gathering host connection information

```
dscli> lshostconnect
Name ID   WWPN             HostType  Profile               portgrp volgrpID ESSIOport
=========================================================================================
vfc_0 0032 C0507602916D0014 iSeries   IBM iSeries - OS/400        0 V14      all
vfc_1 0033 C0507602916D0016 iSeries   IBM iSeries - OS/400        0 V14      all
x3850_m2_01_1 0034 21000024FF2D0ED4 LinuxSuse Intel - Linux Suse 0 V17      I0035,I0002
x3850_m2_01_2 0035 21000024FF2D0F8C LinuxSuse Intel - Linux Suse 0 V17      I0336,I0105

dscli> showvolgrp v17
Name LinuxSuse
ID   V17
Type SCSI Map 256
Vols 8603 8607

dscli> showhostconnect 0034
Name          x3850_m2_01_1
ID            0034
WWPN          21000024FF2D0ED4
HostType      LinuxSuse
LBS           512
addrDiscovery LUNPolling
Profile       Intel - Linux Suse
portgrp       0
volgrpID      V17
atchtopo      -
ESSIOport     I0035,I0002
speed         Unknown
```

Example 12-14 shows that volume group V17 participates in two host connections for two WWPNs: 21000024FF2D0ED4 and 21000024FF2D0F8C. These WWPNs are the same WWPNs in Example 12-10 on page 464. Now, we have all the information for a specific volume.

12.4.2 Disk I/O performance indicators

The disk subsystem is often the most important aspect of server performance and is usually the most common bottleneck. Applications are considered to be I/O-bound when CPU cycles are wasted simply waiting for I/O tasks to finish. However, problems can be hidden by other factors, such as lack of memory.

The symptoms that show that the server might be suffering from a disk bottleneck (or a hidden memory problem) are shown in Table 12-1.

Table 12-1 Disk I/O performance indicators

Disk I/O indicators	Analysis
Disk I/O numbers and wait time	Analyze the number of I/Os to the LUN. This data can be used to discover if reads or writes are the cause of problem. Use **`iostat`** to get the disk I/Os. Use **`stap ioblock.stp`** to get read/write blocks. Also, use **`scsi.stp`** to get the scsi wait times, requested submitted, and completed. Also, long wait times might mean the I/O is to specific disks and not spread out.
Disk I/O size	The memory buffer available for the block I/O request might not be sufficient, and the page cache size can be smaller than the maximum number of Disk I/O size. Use **`stap ioblock.stp`** to get request sizes. Use **`iostat`** to get the blocksizes.
Disk I/O scheduler	An inappropriate I/O scheduler might cause performance bottlenecks. A certain I/O scheduler performs better if configured with the appropriate filesystem.
Disk filesystem	An inappropriate filesystem might cause performance bottlenecks. The appropriate filesystem must be chosen based on the requirements.
Disk I/O to physical device	If all the disk I/Os are directed to the same physical disk, it might cause a disk I/O bottleneck. Directing the disk I/O to different physical disks increases the performance.
Filesystem blocksize	If the filesystem is created with small-sized blocks, creating files larger than the blocksize might cause a performance bottleneck. Creating a filesystem with a proper blocksize improves the performance.
Swap device/area	If a single swap device/area is used, it might cause performance problems. To improve the performance, create multiple swap devices or areas.

12.4.3 Identifying disk bottlenecks

A server that exhibits the following symptoms might suffer from a disk bottleneck (or a hidden memory problem):

- Slow disks result in these issues:
  - Memory buffers that fill with write data (or wait for read data), which delays all requests, because free memory buffers are unavailable for write requests (or the response is waiting for read data in the disk queue).
  - Insufficient memory, as in the case of not enough memory buffers for network requests, causes synchronous disk I/O.
- Disk utilization, controller utilization, or both types are typically high.
- Most local area network (LAN) transfers happen only after disk I/O has completed, causing long response times and low network utilization.
- Disk I/O can take a relatively long time and disk queues become full, so the CPUs are idle or have low utilization, because they wait long periods of time before processing the next request.

Linux offers command-line tools to monitor performance-relevant information. Several of these tools are extremely helpful to get performance metrics for disk I/O-relevant areas.

The vmstat command

One way to track disk usage on a Linux system is by using the **vmstat** tool (Example 12-15). The important columns in **vmstat** with respect to I/O are the bi and bo fields. These fields monitor the movement of blocks in and out of the disk subsystem. Having a baseline is key to being able to identify any changes over time.

Example 12-15 The vmstat tool output

```
[root@x232 root]# vmstat 2
r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 2  1      0   9004  47196 1141672    0    0     0   950  149    74 87 13  0  0
 0  2      0   9672  47224 1140924    0    0    12 42392  189    65 88 10  0  1
 0  2      0   9276  47224 1141308    0    0   448     0  144    28  0  0  0 100
 0  2      0   9160  47224 1141424    0    0   448  1764  149    66  0  1  0 99
 0  2      0   9272  47224 1141280    0    0   448    60  155    46  0  1  0 99
 0  2      0   9180  47228 1141360    0    0  6208 10730  425   413  0  3  0 97
 1  0      0   9200  47228 1141340    0    0 11200     6  631   737  0  6  0 94
 1  0      0   9756  47228 1140784    0    0 12224  3632  684   763  0 11  0 89
 0  2      0   9448  47228 1141092    0    0  5824 25328  403   373  0  3  0 97
 0  2      0   9740  47228 1140832    0    0   640     0  159    31  0  0  0 100
```

The iostat command

Performance problems can be encountered when too many files are opened, read, and written to, and then closed repeatedly. Problems can become apparent as *seek times* (the time that it takes to move to the exact track where the data is stored) start to increase. By using the **iostat** tool, which is part of the sysstat package, you can monitor the I/O device loading in real time. Various options enable you to drill down even deeper to gather the necessary data.

In general, pay attention to the following metrics of the **iostat** data:

- %user is the percentage of CPU utilization that occurred while executing at the user level (application).

- %System is the percentage of CPU utilization that occurred while executing at the system level (kernel).
- %iowait is the percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request.
- %idle is the percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.
- tps is the number of transfers per second that were issued to the device. A transfer is an I/O request to the device. Multiple logical requests can be combined into a single I/O request to the device. A transfer is of indeterminate size.
- kB_read/s is the amount of data read from the device expressed in kilobytes per second.
- kB_wrtn/s is the amount of data written to the device expressed in kilobytes per second.
- r/s is the number of read requests that were issued to the device per second.
- w/s is the number of write requests that were issued to the device per second.
- avgrq-sz is the average size (in sectors) of the requests that were issued to the device.
- avgqu-sz is the average queue length of the requests that were issued to the device.
- await is the average time (in milliseconds) for I/O requests issued to the device to be served. This metric includes the time that is spent by the requests in queue and the time spent servicing them.
- svctm is the average service time (in milliseconds) for I/O requests that were issued to the device.

Remember to gather statistics in extended mode with timestamp and with kilobytes or megabytes values. This information is easier to understand, and you capture all necessary information at a time. Use the **iostat -kxt** or **iostat -mxt** command.

Look at the statistics from the **iostat** tool to help you understand the situation. You can use the following suggestions as shown in the examples.

Good situations

Good situations have the following characteristics:

- High tps value, high %user value, low %iowait, and low svctm: Very good condition, as expected
- High tps value, high %user value, medium %iowait, and medium svctm: Situation is still good, but requires attention. Probably write activity is a little higher than expected. Check write block size and queue size.
- Low tps value, low %user value, medium-high %iowait value, low %idle value, high MB\sec value, and high avgrq-sz value: System performs well with large block write or read activity.

Bad situations

- Low tps value, low %user value, low %iowait, and low svctm: System is not handling disk I/O. If the application still suffers from the disk I/O, look first in the application, not in the disk subsystem.
- Low tps value, low %user value, high %system value, high or low svctime, and 0 %idle value: System is stuck with disk I/O. This situation can happen when a path failed, a device adapter problem, and application errors.
- High tps value, medium %user value, medium %system value, high %iowait value, and high svctim: System consumed the disk resources, and you need to consider an upgrade. Increase the number of disks first.

- Low tps value, low %user value, high %iowait value, high service time, high read queue, and high write queue: High write large block activity exists on the same disks that are also intended for read activity. Split the data for writes and reads on the separate disks.

These situations are examples for your understanding. Plenty of similar situations might occur. Remember to analyze not one or two values of the collected data, but try to obtain a full picture by combining all the available data.

Monitoring workload distribution on the paths

As shown in Example 12-7 on page 463, disks dm-0 and dm-1 have four paths each. It is important to monitor whether the I/O workload is spread evenly across the paths (Example 12-16). However, DM-MP does not support true load balancing, as IBM SDD does. Therefore, you cannot expect a true even distribution, but nevertheless, several paths must be active while servicing the I/O workload.

Example 12-16 Monitoring workload distribution

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,00    0,00    0,62   25,78    0,09   73,51

Device:         rrqm/s   wrqm/s     r/s     w/s     rkB/s     wkB/s avgrq-sz avgqu-sz   await  svctm  %util
sda               0,00     0,00    0,00    0,00      0,00      0,00     0,00     0,00    0,00   0,00   0,00
sdb               0,00     0,00    0,00    0,00      0,00      0,00     0,00     0,00    0,00   0,00   0,00
sdc               0,00     0,00  978,22   11,88 124198,02   6083,17   263,17     1,75    1,76   0,51  50,30
sdd               0,00     0,00    0,00    0,00      0,00      0,00     0,00     0,00    0,00   0,00   0,00
sde               0,00     0,00   20,79    0,00   2538,61      0,00   244,19     0,03    0,95   0,76   1,58
dm-0          59369,31  1309,90 1967,33   15,84 244356,44   8110,89   254,61   154,74   44,90   0,50  99,41
dm-1              0,00     0,00    0,00    0,00      0,00      0,00     0,00     0,00    0,00   0,00   0,00
sdf               0,00     0,00    0,00    0,00      0,00      0,00     0,00     0,00    0,00   0,00   0,00
sdg               0,00     0,00    0,00    0,00      0,00      0,00     0,00     0,00    0,00   0,00   0,00
sdh               0,00     0,00    0,00    0,00      0,00      0,00     0,00     0,00    0,00   0,00   0,00
sdi               0,00     0,00  968,32    3,96 117619,80   2027,72   246,12     1,67    1,71   0,49  47,52
```

Example 12-16 shows that disk dm-0 is running the workload. It has four paths: sdc, sde, sdg, and sdi. Workload is distributed now to three paths for reading (sdc, sde, and sdi) and to two paths for writing (sdc and sdi).

Example 12-17 shows a potential I/O bottleneck on the device /dev/sdb1. This output shows average wait times (await) of about 2.7 seconds and service times (svctm) of 270 ms.

Example 12-17 Sample of an I/O bottleneck as shown with iostat 2 -x /dev/sdb1

```
[root@x232 root]# iostat 2 -x /dev/sdb1

avg-cpu:  %user   %nice    %sys   %idle
          11.50    0.00    2.00   86.50

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
/dev/sdb1  441.00 3030.00  7.00 30.50 3584.00 24480.00  1792.00 12240.00   748.37   101.70 2717.33 266.67 100.00

avg-cpu:  %user   %nice    %sys   %idle
          10.50    0.00    1.00   88.50

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
/dev/sdb1  441.00 3030.00  7.00 30.00 3584.00 24480.00  1792.00 12240.00   758.49   101.65 2739.19 270.27 100.00

avg-cpu:  %user   %nice    %sys   %idle
          10.95    0.00    1.00   88.06

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
/dev/sdb1  438.81 3165.67  6.97 30.35 3566.17 25576.12  1783.08 12788.06   781.01   101.69 2728.00 268.00 100.00
```

This output in Example 12-17 is a typical example of how a large block write workload can affect the response time of the small block read workload. There are about 30 write IOPS with 12240 KB/s bandwidth and 7 read IOPS with 1792 KB/s. This output shows a large I/O size, which is why it shows big queue sizes and high service times. However, the I/O bottleneck is not necessarily the disk system in this case. The application level needs changes to avoid large workloads in the future. To be able to satisfy these workloads, the disk system must have much more cache memory.

Example 12-18 shows the output of the **iostat** command on a logical partition (LPAR) configured with 1.2 CPUs running Red Hat Enterprise Linux AS 4 while issuing server writes to the disks sda and dm-2. The disk transfers per second are 130 for sda and 692 for dm-2. The %iowait is 6.37%, which might seem high for this workload, but it is not. It is normal for a mix of write and read workloads. However, it might grow rapidly in the future, so pay attention to it.

Example 12-18 Outputs from iostat command

```
#iostat -k
avg-cpu:  %user   %nice    %sys %iowait   %idle
           2.70    0.11    6.50    6.37   84.32
Device:            tps   kB_read/s   kB_wrtn/s   kB_read   kB_wrtn
sda             130.69     1732.56     5827.90    265688    893708
sda1              1.24        2.53        0.00       388         0
sda2              4.80        5.32        0.03       816         4
sda3            790.73     1717.40     5827.87    263364    893704
dm-0             96.19     1704.71      292.40    261418     44840
dm-1              0.29        2.35        0.00       360         0
dm-2            692.66        6.38     5535.47       978    848864
```

Example 12-19 shows the output of the **iostat -k** command on an LPAR configured with a 1.2 CPU running Red Hat Enterprise Linux AS 4 that issues server writes to the sda and dm-2 disks. The disk transfers per second are 428 for sda and 4024 for dm-2. The %iowait increased to 12.42%. Our prediction from the previous example is now true. The workload became higher and the %iowait value grew, but the %user value remained the same. The disk system now can hardly manage the workload and requires some tuning or an upgrade. Although the workload grew, the performance of the user processes did not improve. The application might issue more requests, but they must wait in the queue instead of being serviced. Gather the extended iostat statistics.

Example 12-19 Output of iostat to illustrate disk I/O bottleneck

```
# iostat -k
avg-cpu:  %user   %nice    %sys %iowait   %idle
           2.37    0.20   27.22   12.42   57.80
Device:            tps   kB_read/s   kB_wrtn/s   kB_read   kB_wrtn
sda             428.14      235.64    32248.23    269840  36928420
sda1              0.17        0.34        0.00       388         0
sda2              0.64        0.71        0.00       816         4
sda3           4039.46      233.61    32248.17    267516  36928352
dm-0             14.63      231.80       52.47    265442     60080
dm-1              0.04        0.31        0.00       360         0
dm-2           4024.58        0.97    32195.76      1106  36868336
```

Changes made to the elevator algorithm as described in 12.3.4, “Tuning the disk I/O scheduler” on page 456 are displayed in avgrq-sz (average size of request) and avgqu-sz (average queue length), as illustrated in Example 12-16 on page 472. As the latencies are lowered by manipulating the elevator settings, avgrq-sz decreases. You can also monitor the rrqm/s and wrqm/s to see the effect on the number of merged reads and writes that the disk can manage.

sar command

The **sar** command, which is included in the sysstat package, uses the standard system activity daily data file to generate a report.

The system must be configured to collect the information and log it; therefore, a **cron** job must be set up. Add the following lines as shown in Example 12-20 to the /etc/crontab for automatic log reporting with **cron**.

Example 12-20 Example of automatic log reporting with cron

```
....
#8am-7pm activity reports every 10 minutes during weekdays.
0 8-18 **1-5 /usr/lib/sa/sa1 600 6 &
#7pm-8am activity reports every hour during weekdays.
0 19-7 **1-5 /usr/lib/sa/sa1 &
#Activity reports every hour on Saturday and Sunday.
0 ***0,6 /usr/lib/sa/sa1 &
#Daily summary prepared at 19:05
5 19 ***/usr/lib/sa/sa2 -A &
....
```

You get a detailed overview of your CPU utilization (%user, %nice, %system, and %idle), memory paging, network I/O and transfer statistics, process creation activity, activity for block devices, and interrupts/second over time.

The **sar -A** command (the **-A** is equivalent to **-bBcdqrRuvwWy -I SUM -I PROC -n FULL -U ALL**, which selects the most relevant counters of the system) is the most effective way to grep all relevant performance counters. We suggest that you use the **sar** command to analyze whether a system is disk I/O-bound and waits too much, which results in filled-up memory buffers and low CPU usage. Furthermore, this method is useful to monitor the overall system performance over a longer time period, for example, days or weeks, to further understand which times a claimed performance bottleneck is seen.

A variety of additional performance data collection utilities are available for Linux. Most of them are transferred from UNIX systems. You can obtain more details about those additional tools in Chapter 9, "Performance considerations for UNIX servers" on page 327.

For a precise performance analysis, you must have statistics from the operating system and the disk subsystem. For a complete picture to understand the problem, you must analyze both of them. Statistics from the disk system can be gathered with Tivoli Productivity Center for Disk. For more information, see Chapter 7, "Practical performance management" on page 235.

13

Performance considerations for IBM i

In this chapter, we describe the topics related to the DS8000 performance with an IBM i host. The performance of IBM i database applications and batch jobs is sensitive to the disk response time. Therefore, it is important to understand how to plan, implement, and analyze the DS8000 performance with IBM i.

13.1 IBM i storage architecture

To understand the performance of the DS8000 with IBM i, you need insight into IBM i storage architecture. In this section, we explain this part of IBM i architecture and how it works with the DS8000.

The following IBM i specific features are important for the performance of external storage:

- Single-level storage
- Object-based architecture
- Storage management
- Types of disk pools

We describe these features and explain how they relate to the performance of a connected DS8000.

13.1.1 Single-level storage

The IBM i uses the same architectural component that is used by the iSeries and AS/400® platform: single-level storage. It treats the main memory and the disk space as one storage area. It uses the same set of 64-bit virtual addresses to cover both main memory and disk space. Paging in this virtual address space is performed in 4 KB memory pages.

13.1.2 Object-based architecture

One of the differences between the IBM i and other operating systems is the concept of objects. Anything that you can change in the operating system is a type of object. For example, data files, programs, libraries, queues, user profiles, and device descriptions are all types of objects. Every object on the IBM i is packaged with the set of rules for how it can be used, enhancing integrity, security, and virus-resistance.

13.1.3 Storage management

Storage management is a part of the IBM i licensed internal code that manages the I/O operations to store and place data on storage. Storage management handles the I/O operations the following way. When the application performs an I/O operation, the portion of the program that contains read or write instructions is first brought into main memory where the instructions are then executed.

With the read request, the virtual addresses of the needed record are resolved, and for each needed page, storage management first looks to see whether it is in main memory. If the page is there, it is used to resolve the read request. However, if the corresponding page is not in main memory, a page fault is encountered and it must be retrieved from disk. When a page is retrieved, it replaces another page in memory that recently was not used; the replaced page is paged out to disk.

Similarly writing a new record or updating an existing record is done in main memory, and the affected pages are marked as changed. A changed page normally remains in main memory until it is written to disk as a result of a page fault. Pages are also written to disk when a file is closed or when write-to-disk is forced by a user through commands and parameters.
The handling of I/O operations is shown in Figure 13-1 on page 477.

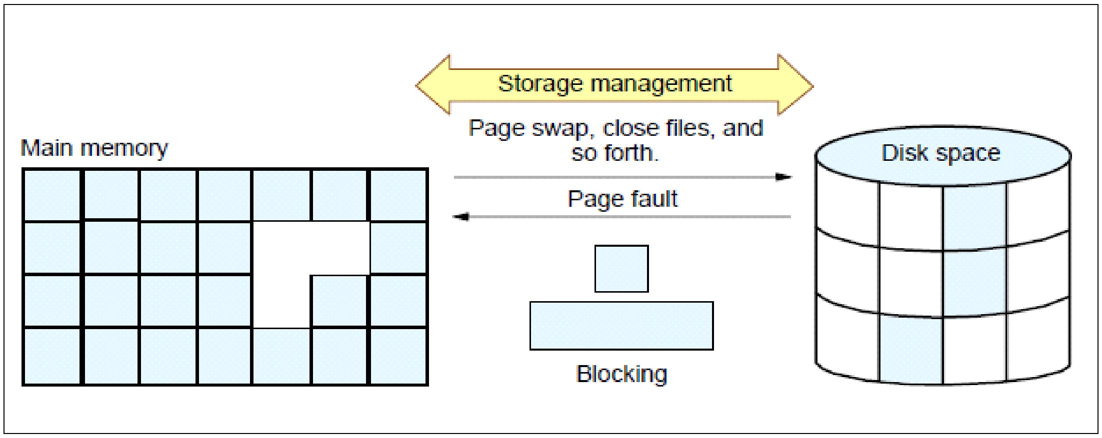


Figure 13-1 Storage management handling I/O operations

When resolving virtual addresses for I/O operations, storage management directories map the disk and sector to a virtual address. For a read operation, a directory lookup is performed to get the needed information for mapping. For a write operation, the information is retrieved from the page tables. Figure 13-2 illustrates resolving addresses to sectors in storage volumes.

An I/O operation is done for a block of 8 sectors, which equals one 4 KB page. The exception is when the storage is connected through Virtual I/O Server (VIOS). In this case, the 8 × 520 byte sectors from IBM i are mapped into the 9 × 512 byte sectors of the storage system.

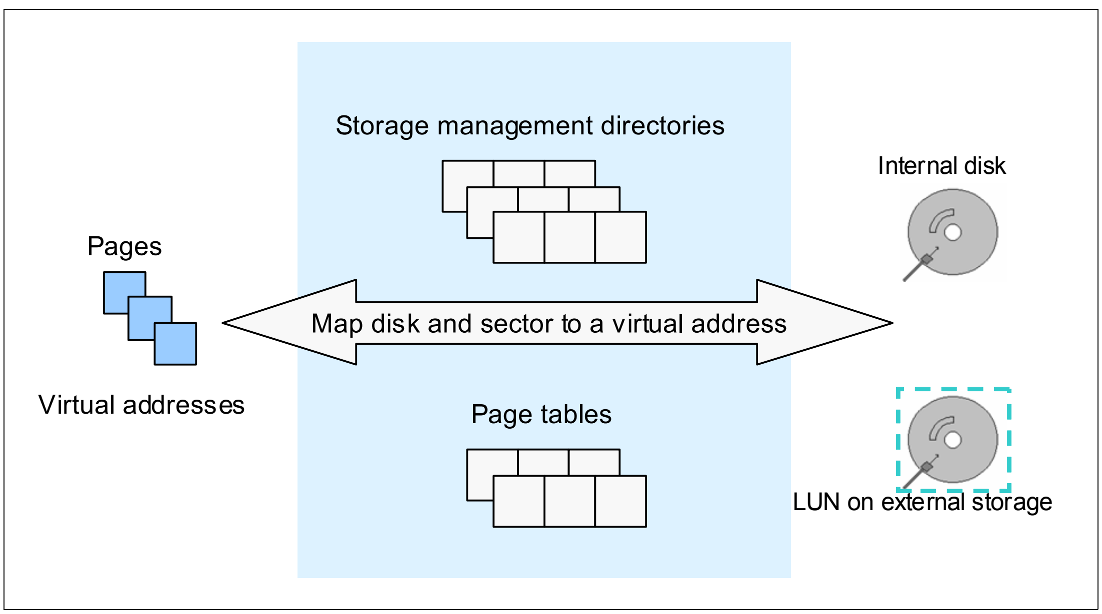


Figure 13-2 Storage management resolving virtual addresses

13.1.4 Disk pools in IBM i

The disk pools in IBM i are referred to as auxiliary storage pools (ASPs). The following types of disk pools exist in IBM i:

- System ASP
- User ASP
- Independent ASP

System ASP

The system ASP is the basic disk pool for IBM i. This ASP contains the IBM i boot disk (load source), system libraries, indexes, user profiles, and other system objects. The system ASP is always present in IBM i and is needed for IBM i. IBM i does not IPL (boot) if the system ASP is inaccessible.

User ASP

A user ASP separates the storage for different objects for easier management. For example, the libraries and database objects that belong to one application are in one user ASP, and the objects of another application are in a different user ASP. If user ASPs are defined in the IBM i system, they are needed for IBM i to IPL.

Independent ASP (IASP)

The independent ASP is a disk pool that can switch among two or more IBM i systems that reside in a cluster. An IBM i system can boot without accessing the IASP. Typically, the objects that belong to a particular application are in this disk pool. If the IBM i system with IASP fails, the independent disk pool can be switched to another system in a cluster. Or, if the IASP is on the DS8000, the copy of IASP (FlashCopy, Metro Mirror, or Global Mirror copy) is made available to another IBM i and the cluster, and the application continues to work from another IBM i.

13.2 Fibre Channel adapters and Multipath

In this section, we explain the usage of Fibre Channel (FC) adapters to connect the DS8000 to an IBM i system, with multiple ways to connect. We describe the performance capabilities of the adapters and the performance enhancement of using Multipath.

The DS8000 can connect to an IBM i system in one of the following ways:

- Native: FC adapters in IBM i are connected through a storage area network (SAN) to the host bus adapters (HBAs) in the DS8000.
- With Virtual I/O Server Node Port ID Virtualization (VIOS NPIV): FC adapters in the VIOS are connected through a SAN to the HBAs in the DS8000. IBM i is a client of the VIOS and uses virtual FC adapters; each virtual FC adapter is mapped to a port in an FC adapter in the VIOS.

  For more information about connecting the DS8000 to IBM i with VIOS_NPIV, see *DS8000 Copy Services for IBM i with VIOS*, REDP-4584, and *IBM System Storage DS8000: Host Attachment and Interoperability*, SG24-8887.
- With Virtual I/O Server (VIOS): FC adapters in the VIOS are connected through a SAN to the HBAs in the DS8000. IBM i is a client of the VIOS, and virtual SCSI adapters in VIOS are connected to the virtual SCSI adapters in IBM i.

  For more information about connecting storage systems to IBM i with the VIOS, see *IBM i and Midrange External Storage*, SG24-7668, and *DS8000 Copy Services for IBM i with VIOS*, REDP-4584.

Most installations use the native connection of the DS8000 to IBM i or the connection with VIOS_NPIV.

IOPs: The information provided in this section refers to connection with IBM i I/O processor (IOP)-less adapters. For similar information regarding older IOP-based adapters, see *IBM i and IBM System Storage: A Guide to Implementing External Disks on IBM i,* SG24-7120-01.

13.2.1 FC adapters for native connection

The following FC adapters are used to connect the DS8000 natively to an IBM i partition in a POWER® server:

- 4-Port 8 Gb PCIe Generation-2 Fibre Channel Adapter, Feature Code 5729
- 2-Port 8 Gb PCIe Fibre Channel Adapter, Feature Code 5735
- 2-Port 4 Gb PCIe Fibre Channel Adapter, Feature Code 5774
- 2-Port 4 Gb PCI-X Fibre Channel Adapter, Feature Code 5749

All listed adapters are IOP-less adapters. They do not require an I/O processor card to offload the data management. Instead, the CPU manages the I/O and communicates directly with the FC adapter. Thus, the IOP-less FC technology takes full advantage of the performance potential in IBM i.

Before the availability of IOP-less adapters, the DS8000 connected to IOP-based FC adapters that require the I/O processor card.

IOP-less FC architecture enables two technology functions that are important for the performance of the DS8000 with IBM i: Tag Command Queuing and Header Strip Merge.

Tagged Command Queuing

Tagged Command Queuing allows the IBM i to issue multiple commands to the DS8000 on the same path to a logical volume. In the past, the IBM i sent one command only per LUN path. Up to six I/O operations to the same LUN through one path are possible with Tag Command Queuing in the natively connected DS8000. With the natively connected DS8000, the queue depth on a LUN is 6.

Header Strip Merge

Header Strip Merge allows the IBM i to bundle data into 4 KB chunks. By merging the data together, it reduces the amount of overhead required for the management of smaller data chunks.

13.2.2 FC adapters in VIOS

The following FC adapters are used to connect the DS8000 to VIOS in a POWER server:

- 4-Port 8 Gb PCIe Generation-2 Fibre Channel Adapter, Feature Code 5729
- 2-Port 8 Gb PCIe Fibre Channel Adapter, Feature Code 5735
- 2-Port 4 Gb PCIe Fibre Channel Adapter, Feature Code 5774
- 1-Port 4 Gb PCIe Fibre Channel Adapter, Feature Code 5773

Queue depth and the number of command elements in the VIOS

When you connect the DS8000 to an IBM i through the VIOS, consider the following types of queue depths:

- The queue depth per LUN: SCSI command tag queuing in the IBM i operating system enables up to 32 I/O operations to one LUN at the same time. This queue depth is valid for either connection with VIOS vscsi or connection with VIOS_NPIV.
- If the DS8000 is connected in VIOS vscsi, consider the queue depth 32 per physical disk (hdisk) in the VIOS. This queue depth indicates the maximum number of I/O requests that can be outstanding on a physical disk in the VIOS at one time.
- The number of command elements per port in a physical adapter in the VIOS is 500 by default; we can increase it to 2048. This queue depth indicates the maximum number of I/O requests that can be outstanding on a port in a physical adapter in the VIOS at one time, either in VIOS vscsi connection or in VIOS_NPIV. The IBM i operating system has a fixed queue depth of 32, which is not changeable. However, the queue depth for a physical disk in the VIOS can be set up by a user. If needed, set the queue depth per physical disk in the VIOS to 32 by using the command **`chdev -dev hdiskxx -attr queue_depth=32`**.

13.2.3 Multipath

IBM i allows multiple connections from different ports on a single IBM i partition to the same logical volumes in the DS8000. This multiple connections support provides an extra level of availability and error recovery between the IBM i and the DS8000. If one IBM i adapter fails, or one connection to the DS8000 is lost, we can continue using the other connections and continue communicating with the disk unit. IBM i supports up to eight active connections (paths) to a single LUN in the DS8000.

In addition to high availability, multiple paths to the same LUN provide load balancing. A Round-Robin algorithm is used to select the path for sending the I/O requests. This algorithm enhances the performance of IBM i with the DS8000 LUNs connected in Multipath.

Multipath is part of the IBM i operating system. This Multipath differs from other platforms that have a specific software component to support multipathing, such as the Subsystem Device Driver (SDD).

When the DS8000 connects to IBM i through the VIOS, the Multipath in IBM i is implemented so that each path to a LUN uses a different VIOS. Therefore, at least two VIOSs are required to implement Multipath for an IBM i client. This way of multipathing provides additional resiliency in case one VIOS fails. In addition to the IBM i Multipath with two or more VIOS, the FC adapters in each VIOS can multipath to the connected DS8000 to provide additional resiliency and enhance performance.

13.3 Performance guidelines for DS8000 with IBM i

In this section, we describe the guidelines to use when planning and implementing the DS8000 for IBM i to achieve the desired performance.

13.3.1 RAID level

Most IBM i clients use RAID 10 or RAID 5 for their workloads. RAID 10 provides better resiliency than RAID 5, and in many cases, it enables better performance. The difference in performance is due to the lower RAID penalty that is experienced with RAID 10 compared to

RAID 5. The workloads with a low read/write ratio and with many random writes benefit the most from RAID 10 as far as performance.

We suggest that you use RAID 10 for IBM i systems, especially for the following types of workloads:

- Workloads with large I/O rates
- Workloads with many write operations (low read/write ratio)
- Workloads with many random writes
- Workloads with low write-cache efficiency

13.3.2 Number of ranks

To better understand why the number of disk drives or the number of ranks is important for an IBM i workload, we provide a short explanation of how IBM i spreads the I/O over disk drives.

When an IBM i page or a block of data is written to disk space, storage management spreads it over multiple disks. By spreading data over multiple disks, multiple disk arms work in parallel for any request to this piece of data, so writes and reads are faster.

When using external storage with IBM i, Storage management sees a logical volume (LUN) in the DS8000 as a "physical" disk unit. If a LUN is created with the *rotate volumes* extent allocation method (EAM), it occupies multiple stripes of a rank. If a LUN is created with the *rotate extents* EAM, it is composed of multiple stripes of different ranks. Figure 13-3 shows the use of the DS8000 disk with IBM i LUNs created with the rotate extents EAM.

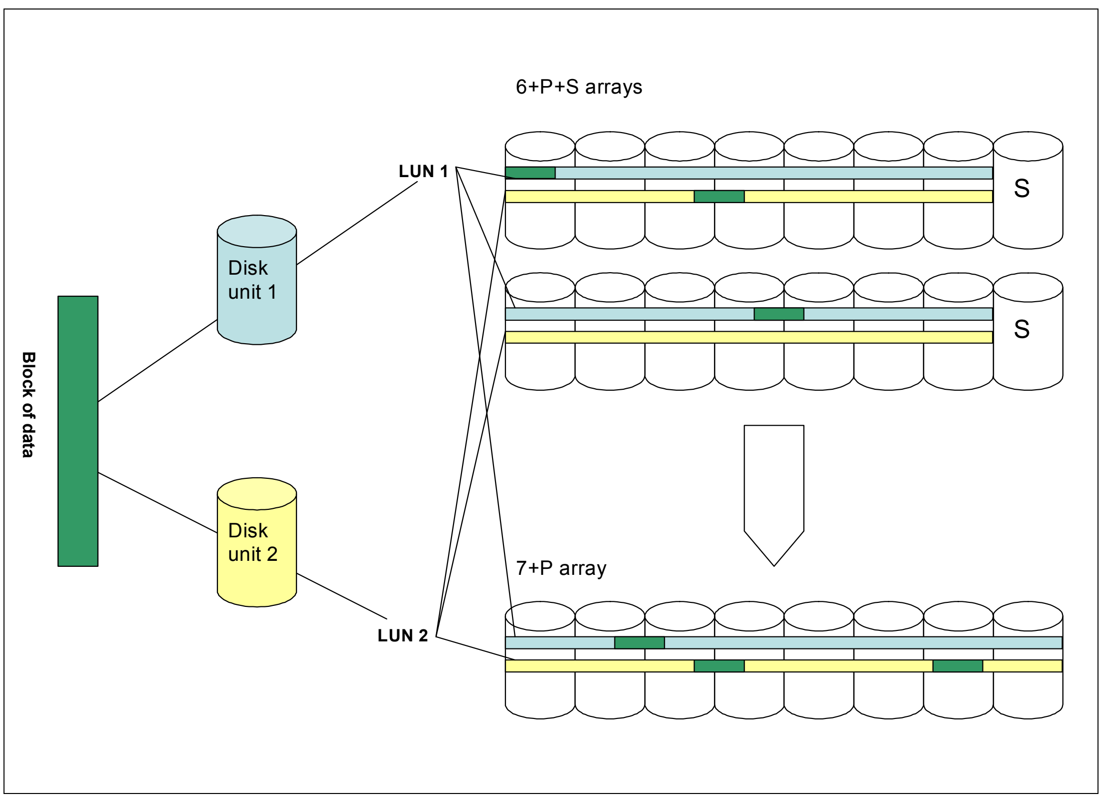


Figure 13-3 Use of disk arms with LUNs created in rotate extents method

Therefore, a LUN uses multiple DS8000 disk arms in parallel. The same DS8000 disk arms are used by multiple LUNs that belong to the same IBM i workload, or even to different IBM i workloads. To efficiently support this structure of I/O and data spreading across LUNs and disk drives, it is important to provide enough disk arms to an IBM i workload.

We suggest that you use the Disk Magic tool when you plan the number of ranks in the DS8000 for an IBM i workload. To provide a good starting point for Disk Magic modeling, consider the number of ranks that is needed to keep disk utilization under 60% for your IBM i workload. Table 13-1 shows the maximal number of IBM i I/O/sec for one rank to keep the disk utilization under 60%, for the workloads with read/write ratios 70/30 and 50/50.

Table 13-1 Suggested host IO/sec for an array[1]

RAID array, disk drive	Host I/O per second at 70% reads	Host I/O per second at 50% reads
RAID 5, 15K rpm	940	731
RAID 10, 15K rpm	1253	1116
RAID 6, 15K rpm	723	526
RAID 5, 10K rpm	749	583
RAID 10, 10K rpm	999	890
RAID 6, 10K rpm	576	420

Use the following steps to calculate the necessary number of ranks for your workload by using Table 13-1:

1. Decide which read/write ratio (70/30 or 50/50) is appropriate for your workload.
2. Decide which RAID level to use for the workload.
3. Look for the corresponding number in Table 13-1.
4. Divide the I/O/sec of your workload by the number from the table to get the number of ranks.

For example, we show a calculation for a medium IBM i workload with a read/write ratio of 50/50. The IBM i workload experiences 8500 I/O per second at a read/write ratio of 50/50. The 15K RPM disk drives in RAID 10 are used for the workload. The number of needed ranks is:

`8500 / 1116 = 7.6` or approximately 8

Therefore, we suggest that you use 8 ranks of 15K RPM disk drives in RAID 10 for the workload.

13.3.3 Number and size of LUNs

For the performance of the IBM i, ensure that the IBM i uses many disk units. The more disk units that are available to an IBM i system, the more server tasks are available for the IBM i storage management to use for managing the I/O operations to the disk space. The result is improved I/O operation performance.

[1] The calculations for the values in Table 13-1 are based on the measurements of how many I/O operations one rank can handle in a certain RAID level, assuming 20% read cache hit and 30% write cache efficiency for the IBM i workload. We assume that half of the used ranks have a spare and half are without a spare.

With IBM i internal disks, a disk unit is a physical disk drive. With a connected DS8000 storage system, a disk unit is a LUN. Therefore, it is important to provide many LUNs to IBM i. With a certain disk capacity, define more smaller LUNs.

> **Number of disk drives in the DS8000:** In addition to the suggestion for many LUNs, we also suggest a sufficient number of disk drives in the DS8000 to achieve good IBM i performance, as described in 13.3.2, "Number of ranks" on page 481.

Another reason why we suggest that you define smaller LUNs for the IBM i is the queue depth in Tagged Command queuing. With a natively connected DS8000, IBM i manages the queue depth of 6 concurrent I/O operations to a LUN. With the DS8000 connected through VIOS, the queue depth for a LUN is 32 concurrent I/O operations. Both of these queue depths are modest numbers compared to other operating systems. Therefore, you need to define sufficiently small LUNs for IBM i not to exceed the queue depth with I/O operations.

Also, by considering the manageability and limitations of external storage and IBM i, we currently suggest that you define LUN sizes of about 70 - 140 GB.

13.3.4 DS8000 extent pools for IBM i workloads

In this section, we suggest how to create extent pools and LUNs for IBM i and dedicate or share ranks for IBM i workloads.

Number of extent pools

We suggest that you create two extent pools for an IBM i workload with each pool in one rank group.

Rotate volumes or rotate extents EAMs for defining IBM i LUNs

IBM i storage management spreads each block of data across multiple LUNs. Therefore, even if the LUNs are created with the rotate volumes EAM, performing an IBM i I/O operation uses the disk arms of multiple ranks.

You might think that the rotate volumes EAM for creating IBM i LUNs provides sufficient disk arms for I/O operations and that the use of the rotate extents EAM is "overvirtualizing". However, based on the performance measurements and preferred practices, the rotate extents EAM of defining LUNs for IBM i still provides the best performance, so we advise that you use it.

Dedicating or sharing the ranks for IBM i workloads

When multiple IBM i LPARs use disk space on the DS8000, there is always a question whether to dedicate ranks (extent pools) to each of them or to share ranks among the IBM i systems.

Sharing the ranks among the IBM i systems enables the efficient use of the DS8000 resources. However, the performance of each LPAR is influenced by the workloads in the other LPARs.

For example, two extent pools are shared among IBM i LPARs A, B, and C. LPAR A experiences a long peak with large blocksizes that causes a high I/O load on the DS8000 ranks. During that time, the performance of B and the performance of C decrease. But, when the workload in A is low, B and C experience good response times, because they can use most of the disk arms in the shared extent pool. In these periods, the response times in B and C are possibly better than if they use dedicated ranks.

We typically cannot predict when the peaks in each LPAR happen, so we cannot predict how the performance in the other LPARs is influenced.

Many IBM i data centers successfully share the ranks with little unpredictable performance, because the disk arms and cache in the DS8000 are used more efficiently this way.

Other IBM i data centers prefer the stable and predictable performance of each system even at the cost of more DS8000 resources. These data centers dedicate extent pools to each of the IBM i LPARs.

Many BM i installations have one or two LPARs with important workloads and several smaller, less important LPARs. These data centers dedicate ranks to the large systems and share the ranks among the smaller ones.

13.3.5 Disk Magic modeling for IBM i

We highly encourage you to use Disk Magic to model the IBM i workload on the DS8000 before deciding which DS8000 configuration to use. The modeling provides the expected IBM i disk response time, several utilization values on the DS8000, and the growth of response time and utilization with IBM i I/O growth.

For more information about the use of Disk Magic with IBM i, see 6.1, “Disk Magic” on page 176 and *IBM i and IBM System Storage: A Guide to Implementing External Disks on IBM i,* SG24-7120.

13.4 Analyzing performance data

For performance issues with IBM i workloads that run on the DS8000, you need to determine and use the most appropriate performance tools in IBM i and in the DS8000.

13.4.1 IBM i performance tools

In this section, we present the performance tools that are used for IBM i. We also indicate which of the tools we employ with the planning and implementation of the DS8000 for IBM i.

To help you better understand the tool functions, we divide them into two groups. We divide them into performance data collectors (the tools that collect performance data) and performance data investigators (the tools to analyze the collected data).

The following tools are the IBM i performance data collectors:

- Collection Services
- IBM i Job Watcher
- IBM i Disk Watcher
- Performance Explorer (PEX)

Collectors can be managed by IBM System Director Navigator for i, IBM i Operations Navigator, or IBM i commands.

The following tools are or contain the IBM i performance data investigators:

- IBM Performance Tools for i
- IBM System Director Navigator for i
- iDoctor

Most of these comprehensive planning tools address the entire spectrum of workload performance on System i, including CPU, system memory, disks, and adapters. To plan or analyze performance for the DS8000 with IBM i, we use the parts of the tools or their reports that show the disk performance.

Collection Services

The major IBM System i performance data collector is called *Collection Services*. It is designed to run all the time to provide data for performance health checks, for analysis of a sudden performance problem, or for planning new hardware and software upgrades. The tool is documented in detail in the IBM i Information Center:

http://publib.boulder.ibm.com/eserver/ibmi.html

Collection Services is a sample-based engine (usually 5 - 15 minute intervals) that looks at jobs, threads, CPU, disk, and communications. It also has a set of specific statistics for the DS8000. For example, it shows which disk units are the DS8000 LUNs, whether they are connected in a single path or multipath, the disk service time, and wait time.

The following tools can be used to manage the data collection and report creation of Collection Services:

- IBM i Operation Navigator
- IBM System Director navigator
- IBM Performance Tools for i

iDoctor Collection Service Investigator can be used to create graphs and reports based on Collection Services data. For more information about iDoctor, see the IBM i iDoctor online documentation at the following link:

https://www-912.ibm.com/i_dir/idoctor.nsf/documentation.html

With IBM i level V7R1, the Collection Services tool offers additional data collection categories, including a category for external storage. This category supports the collection of non-standard data that is associated with certain external storage subsystems that are attached to an IBM i partition. This data can be viewed within iDoctor, which is described in “iDoctor” on page 487.

Job Watcher

Job Watcher is an advanced tool for collecting and analyzing performance information to help you effectively monitor your system or to analyze a performance issue. It is job-centric and thread-centric and can collect data at intervals of seconds. The collection contains vital information, such as job CPU and wait statistics, call stacks, SQL statements, objects waited on, sockets, and TCP. For more information about Job Watcher, see the IBM i Information Center:

http://publib.boulder.ibm.com/eserver/ibmi.html

Or, see “Web Power - New browser-based Job Watcher tasks help manage your IBM i performance” in the IBM Systems Magazine (on the IBM Systems magazine page, search for the title of the article):

http://www.ibmsystemsmag.com/ibmi

Disk Watcher

Disk Watcher is a function of IBM i that provides disk data to help identify the source of disk-related performance problems on the IBM i platform. It can either collect information about every I/O in trace mode or collect information in buckets in statistics mode. In statistics mode, it can run more often than Collection Services to see more granular statistics. The command strings and file layouts are documented in the System i Information Center:

http://publib.boulder.ibm.com/eserver/ibmi.html

For more information about the use of Disk Watcher, see “A New Way to Look at Disk Performance” and “Analyzing Disk Watcher Data” in the IBM Systems Magazine (on the IBM Systems magazine page, search for the title of the article):

http://www.ibmsystemsmag.com/ibmi

Disk Watcher gathers detailed information associated with I/O operations to disk units, and provides data beyond the data that is available in other IBM i integrated tools, such as Work with Disk Status (WRKDSKSTS), Work with System Status (WRKSYSSTS), and Work with System Activity (WKSYSACT).

Performance Explorer

Performance Explorer (PEX) is a data collection tool in IBM i that collects information about a specific system process or resource to provide detailed insight. PEX complements IBM i Collection Services.

An example of PEX, IBM i, and connection with external storage is identifying the IBM i objects that are most suitable to relocate to solid-state drives (SSDs). We often use PEX to collect IBM i disk events, such as synchronous and asynchronous reads, synchronous and asynchronous writes, page faults, and page-outs. The collected data is then analyzed by the iDoctor tool PEX-Analyzer to observe the I/O rates and disk service times of different objects. The objects that experience the highest accumulated read service time, the highest read rate, and a modest write rate at the same time are good candidates to relocate to SSDs.

For a better understanding of IBM i architecture and I/O rates, see 13.1, “IBM i storage architecture” on page 476. For more details about using SSDs with IBM i, see 13.5, “Easy Tier with IBM i” on page 489.

IBM Performance Tools for i

IBM Performance Tools for i is a tool to manage data collection, analyze data, and print reports to help you identify and correct performance problems. Performance Tools helps you gain insight into IBM i performance features, such as dynamic tuning, expert cache, job priorities, activity levels, and pool sizes. You can also identify ways to use these services better. The tool also provides analysis of collected performance data and produces conclusions and recommendations to improve system performance.

The Job Watcher part of Performance Tools analyzes the Job Watcher data through the IBM Systems Director Navigator for i Performance Data Visualizer.

Collection Services reports about disk utilization and activity, which are created with IBM Performance Tools for i, are used for sizing and Disk Magic modeling of the DS8000 for IBM i:

- The Disk Utilization section of the System report
- The Disk Utilization section of the Resource report
- The Disk Activity section of the Component report

IBM Systems Director Navigator for i

The *IBM Systems Director Navigator for i* is a web-based console that provides a single, easy-to-use view of the IBM i system. IBM Systems Director Navigator provides a strategic tool for managing a specific IBM i partition.

The Performance section of IBM Systems Director Navigator for i provides tasks to manage the collection of performance data and view the collections to investigate potential performance issues. Figure 13-4 on page 487 shows the menu of performance functions in the BM Systems Director Navigator for i.

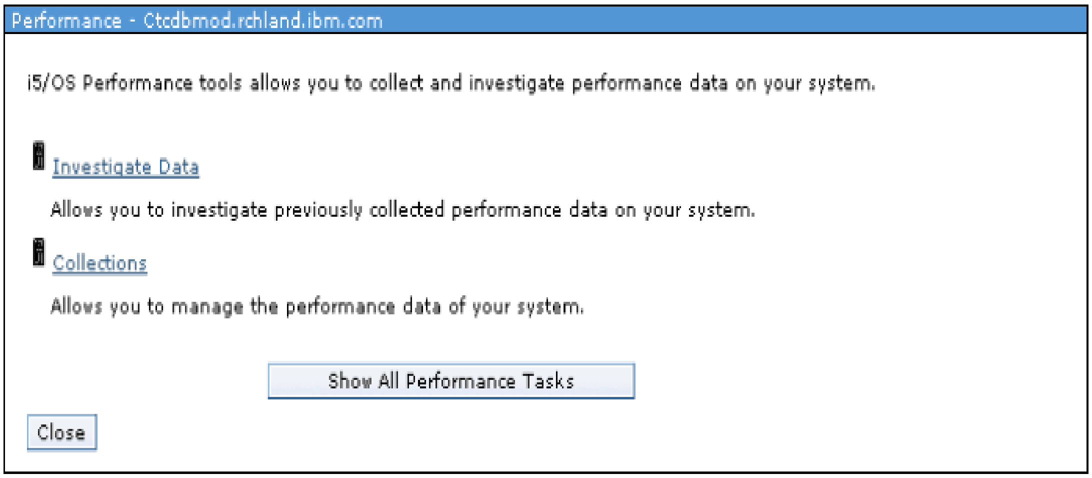


Figure 13-4 Performance tools of Systems Director Navigator

iDoctor

iDoctor is a suite of tools used to manage the collection of data, investigate performance data, and analyze performance data on IBM i. The goals of iDoctor are to broaden the user base for performance investigation, simplify and automate processes of collecting and investigating the performance data, provide immediate access to collected data, and offer more analysis options.

The iDoctor tools are used to monitor the overall system health at a high level or to drill down to the performance details within jobs, disk units, and programs. Use iDoctor to analyze data collected during performance situations. iDoctor is frequently used by IBM, clients, and consultants to help solve complex performance issues quickly.

One example of using iDoctor PEX-Analyzer is to determine the IBM i objects that are candidates to move to SSD. In iDoctor, you launch the tool PEX-Analyzer as shown in Figure 13-5 on page 488.

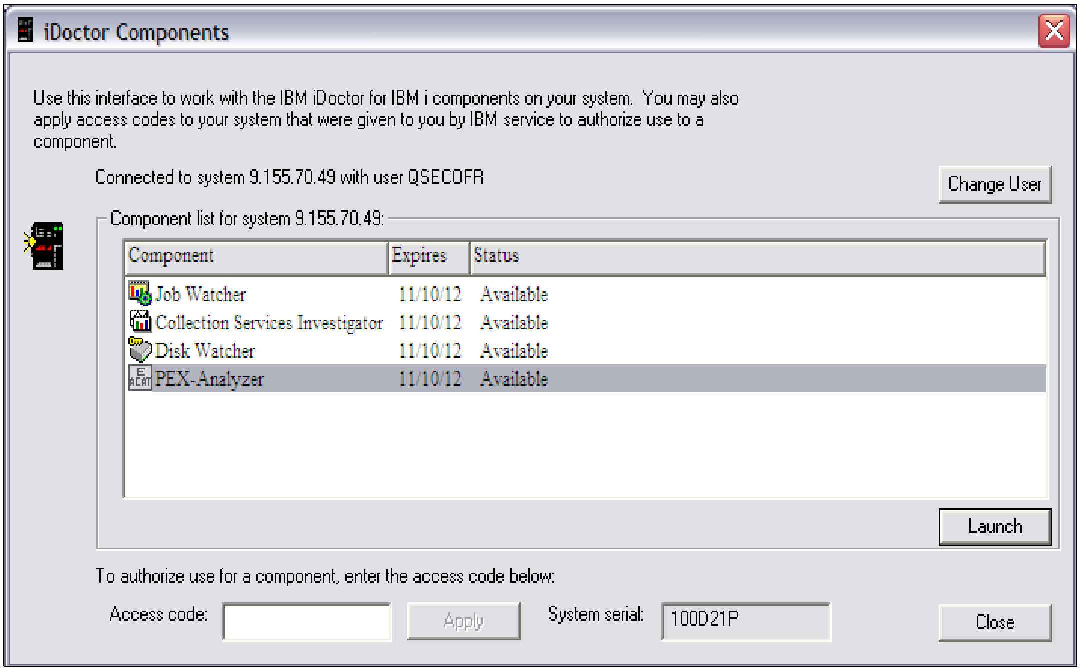


Figure 13-5 Launching the iDoctor PEX Analyzer

Select the PEX collection on which you want to work and select the type of graph that you want to create, as shown in Figure 13-6.

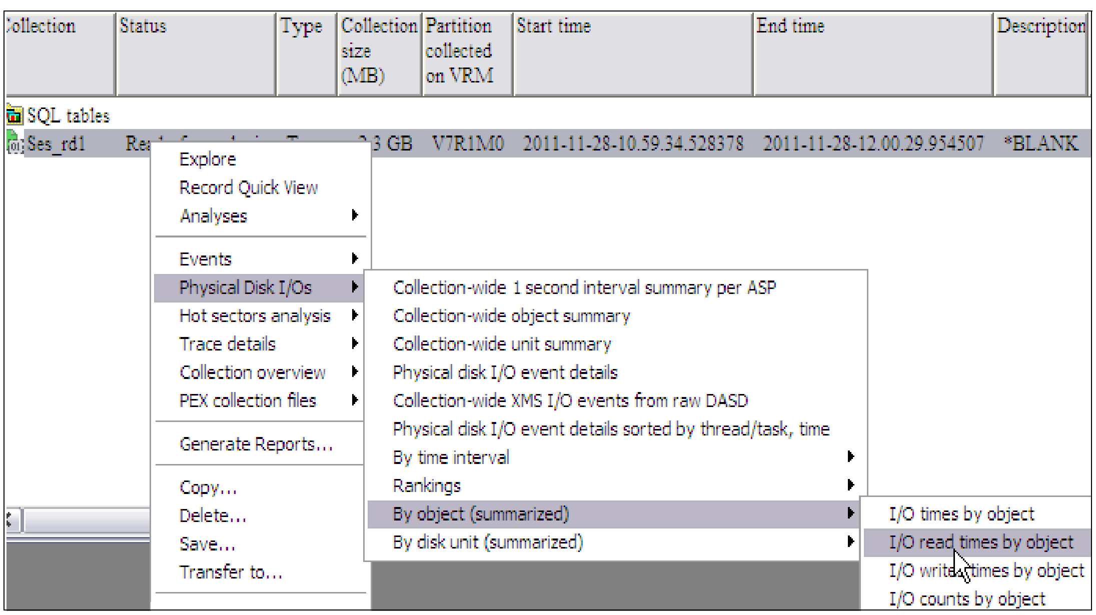


Figure 13-6 iDoctor selecting the query

Figure 13-7 on page 489 illustrates an example of the graph that shows the accumulated read disk service time on IBM i objects. The objects with the highest accumulated read service time are good candidates to relocate to SSD. For more information about relocating IBM i data to SSD, see 13.5.2, “IBM i methods for hot-spot management” on page 490.

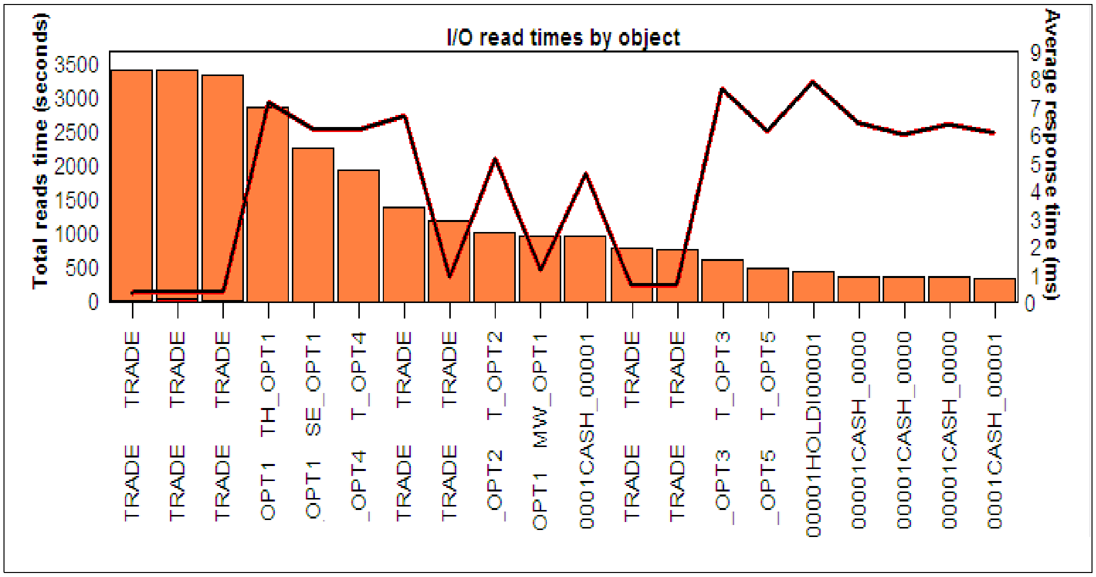


Figure 13-7 iDoctor query of I/O read times by object

13.4.2 DS8000 performance tools

We suggest that you use Tivoli Storage Productivity Center for Disk for analyzing the DS8000 performance data for an IBM i workload. For more information about this product, see Chapter 7, “Practical performance management” on page 235.

13.4.3 Periods and intervals of collecting data

For a successful performance analysis, ensure that the IBM i data and DS8000 data are collected during the same periods, and if possible, with the same collection intervals.

13.5 Easy Tier with IBM i

Next, we describe how to use Easy Tier with IBM i.

13.5.1 Hot data in an IBM i workload

An important feature of IBM i is object-based architecture. Everything on the system that can be worked with is considered an object. An IBM i library is an object. A database table, an index file, a temporary space, a job queue, and a user profile are objects. The intensity of I/Os is split by objects. The I/O rates are high on busy objects, such as application database files and index files. The I/Os rates are lower on user profiles.

IBM i Storage Manager spreads the IBM i data across the available disk units (LUNs) so that each disk drive is about equally occupied. The data is spread in extents that are from 4 KB - 1 MB or even 16 MB. The extents of each object usually span as many LUNs as possible to provide many volumes to serve the particular object. Therefore, if an object experiences a high I/O rate, this rate is evenly split among the LUNs. The extents that belong to the particular object on each LUN are I/O-intense.

Many of the IBM i performance tools work on the object level; they show different types of read and write rates on each object and disk service times on the objects. For more detailed information about the IBM i performance tools, see 13.4.1, “IBM i performance tools” on page 484. You can relocate hot data by objects by using the Media preference method, which is described in “IBM i Media preference” on page 490.

Also, the Easy Tier tool monitors and relocates data on the 1 GB extent level. And, IBM i ASP balancing, which is used to relocate data to SSDs, works on the 1 MB extent level. Monitoring extents and relocating extents do not depend on the object to which the extents belong; they occur on the sub-object level.

13.5.2 IBM i methods for hot-spot management

You can choose your tools for monitoring and relocating data to faster disk drives in the DS8000. Because IBM i recognizes the LUNs on SSDs in a natively connected DS8000, you can use the IBM i tools for monitoring and relocating hot data. The following IBM i methods are available:

- IBM i Media preference
- Auxiliary storage pool (ASP) balancing
- A process where you create a separate ASP with LUNs on SSDs and restore the application to run in the ASP

IBM i Media preference

This method provides monitoring capability and the relocation of the data on the object level. You are in control. You decide the criteria for which objects are hot and you control which objects to relocate to the SSDs. Some clients prefer Media preference to Easy Tier or ASP balancing. IBM i Media preference involves the following steps:

1. Use IBM i Performance Explorer (PEX) to collect disk events

   Carefully decide which disk events to trace. In certain occasions, you might collect read disk operations only, which might benefit the most from improved performance. Or, you might collect both disk read and write information for future decisions. Carefully select the peak period in which the PEX data is collected.

2. Examine the collected PEX data by using the PEX-Analyzer tool of iDoctor or by using the user-written queries to the PEX collection.

   It is a good idea to examine the accumulated read service time and the read I/O rate on specific objects. The objects with the highest accumulated read service time and the highest read I/O rate can be selected to relocate to SSDs. It is also helpful to analyze the write operations on particular objects. You might decide to relocate the objects with high read I/O and rather modest write I/O to SSD to benefit from lower disk service times and wait times. Sometimes, you must distinguish among the types of read and write operations on the objects. iDoctor queries provide the rates of asynchronous and synchronous database reads and page faults.

   Figure 13-8 on page 491 shows an example of a graph, which shows the accumulated read and write service times on IBM i objects. This graph was created by the iDoctor PEX-Analyzer query *I/O Times by Object* on PEX data collected during the IBM i workload described in 13.5.3, “Skew level of an IBM i workload” on page 492. The objects are ordered by the sum of accumulated read and write service times. The read service time on the objects is much higher than the write service time, so the objects might be good candidates to relocate to SSD.

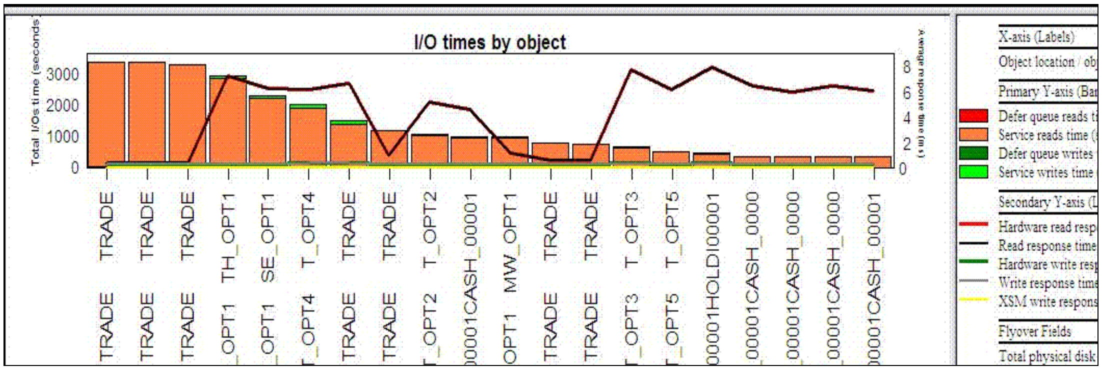


Figure 13-8 Accumulated read and write service time on objects

In certain cases, queries must be created to run on the PEX collection to provide specific information, for example, the query that provides information about which jobs and threads use the objects with the highest read service time. You might also need to run a query to provide the blocksizes of the read operations, because we expect that the reads with smaller blocksizes profit the most from SSDs. If these queries are needed, contact IBM Lab Services to create them.

3. Based on the PEX analysis, decide which database objects to relocate to the SSD in the DS8000. Then, use IBM i commands, such as Change Physical File (CHGPF) with parameter UNIT(*SSD). Or, use the SQL command ALTER TABLE UNIT SSD, which sets on the file a preferred media attribute that invokes dynamic data movement. The preferred media attribute can be set on database tables and indexes, as well as on User-Defined File Systems (UDFS).

   For more information about the UDFS, see the IBM i Information Center:

   http://publib.boulder.ibm.com/eserver/ibmi.html

ASP balancing

This IBM i method is similar to DS8000 Easy Tier, because it is based on the data movement within an ASP by IBM i ASP balancing. The ASP balancing function is designed to improve IBM i system performance by balancing disk utilization across all of the disk units (or LUNs) in an auxiliary storage pool. It provides four ways to balance an ASP. Two of these ways relate to data relocation to SSDs:

- Hierarchical Storage Management (HSM) balancing
- Media preference balancing

The *HSM balancer function*, which traditionally supports data migration between high-performance and low-performance internal disk drives, is extended for the support of data migration between SSDs and hard disk drives (HDDs). The disk drives can be internal or reside on the DS8000. The data movement is based on the weighted read I/O count statistics for each 1 MB extent of an ASP. Data monitoring and relocation is achieved by the following two steps:

1. Run the ASP balancer tracing function during the important period by using the TRCASPBAL command. This function collects the relevant data statistics.
2. By using the STRASPBAL TYPE(*HSM) command, you move the data to SSD and HDD based on the statistics that you collected in the previous step.

The *Media preference balancer function* is the ASP balancing function that helps to correct any issues with Media preference-flagged database objects or UDFS files not on their preferred media type, which is either SSD or HDD, based on the specified *subtype* parameter.

The function is invoked by the STRASPBAL TYPE(*MP) command with the parameter SUBTYPE equal to either *CALC (for data migration to both SSD and HDD), *SSD, or *HDD.

ASP balancer migration priority is an option in the ASP balancer so that you can specify the migration priority for certain balancing operations, including *HSM or *MP in levels of either *LOW, *MEDIUM, or *HIGH, thus influencing the speed of data migration.

Location: For data relocation with Media preference or ASP balancing, the LUNs defined on SSD and on HDD need to reside in the same IBM i ASP. It is not necessary that they are in the same extent pool in the DS8000.

A dedicated ASP for SSD LUNs

This method is suitable for the installations that use a library (or multiple libraries) with files that are all heavily used and critical for the workload performance. It is simple to implement and is suitable also for storage systems other than the DS8000 with which IBM i Media preference and ASP balancing cannot be used.

The method requires that you create a separate ASP that contains LUNs that reside on the DS8000 SSD and, then, save the relevant IBM i libraries and restore them to the ASP with SSD. All the files in the libraries then reside on SSDs, and the performance of the applications that use these files improves.

Additional information

For more information about the IBM i methods for SSD hot-spot management, including the information about IBM i prerequisites, see the following documents:

- *IBM i 7.1 Technical Overview*, SG24-7858
- "Performance Value of Solid State Drives using IBM i":

  http://www.ibm.com/systems/resources/ssd_ibmi.pdf
- *IBM i Two-Tiered Hybrid Storage with IBM System Storage DS8700 Solid State Drives Hot-Spot Data Analysis and Migration in an IBM i DS8700 Environment*:

  http://w3.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101868

13.5.3 Skew level of an IBM i workload

IBM i clients that are new to the DS8000 might consider the configuration of mixed SSDs and Enterprise drives of 15K rpm or 10K rpm, or they might select the configuration of mixed SSDs, Enterprise drives, and nearline drives. Clients that already run their workloads on the DS8000 with HDD might purchase additional SSDs to improve performance. Other clients might decide to add both SSD and nearline disks to get the most efficient configuration for both performance and capacity.

Before deciding on a mixed SSD and HDD environment or deciding to obtain additional SSDs, consider these questions:

- How many SSDs do you need to install to get the optimal balance between the performance improvement and the cost?
- What is the estimated performance improvement after you install the SSDs?

The clients that use IBM i Media preference get at least a partial answer to these questions from the collected PEX data by using queries and calculations. The clients that decide on DS8000 Easy Tier or even IBM i ASP balancing get the key information to answer these questions by the *skew level* of their workloads. The skew level describes how the I/O activity is distributed across the capacity for a specific workload. The workloads with the highest skew level (heavily skewed workloads) benefit most from the Easy Tier capabilities, because even when moving a small amount of data, the overall performance improves. For more information abut the skew level, see 6.5, “Disk Magic Easy Tier modeling” on page 221. We can obtain the skew level only for the workloads that run on the DS8000 with Easy Tier.

To provide an example of the skew level of a typical IBM i installation, we use the IBM i benchmark workload, which is based on the workload *TPC-E.* TPC-E is a new online transaction processing (OLTP) workload developed by the Tivoli Storage Productivity Center. It uses a database to model a brokerage firm with customers who generate transactions related to trades, account inquiries, and market research. The brokerage firm in turn interacts with financial markets to execute orders on behalf of the customers and updates relevant account information. The benchmark workload is scalable, which means that the number of customers defined for the brokerage firm can be varied to represent the workloads of different-sized businesses. The workload runs with a configurable number of job sets. Each job set executes independently, generating its own brokerage firm next transaction. By increasing the number of job sets, we increased the throughput and CPU utilization of the run.

We used the following configuration for Easy Tier monitoring for which we obtained the skew level:

- IBM i LPAR with 8 processing units and 60 GB memory in POWER7 model 770
- Disk space for the IBM i provided from an extent pool with 4 ranks of HDD in DS8800 code level 6.2
- 48 LUNs of the size 70 GB used for IBM i (the LUNs are defined in the rotate extents EAM from the extent pool)
- The LUNs are connected to IBM i in Multipath through 2 ports in separate 4 Gb FC adapters
- In the IBM i LPAR, we ran the following two workloads in turn:
  - The workload with 6 database instances and 6 job sets
  - The workload with 6 database instances and 3 job sets

The workload with 6 database instances was used to achieve 35% occupation of the disk space. During the run with 6 job sets, the access density was about 2.7 IO/sec/GB. During the run with 3 job sets, the access density was about 0.3 IO/sec/GB.

Figure 13-9 on page 494 and Figure 13-10 on page 495 show the skew level of the IBM i workload from the Easy Tier data collected during 24 hours. On Figure 13-9 on page 494, you can see the percentage of reads with small blocksizes and the percentage of transferred MB by the percentage of occupied disk space. The degree of skew is small due to the efficient spreading of data across available LUNs by IBM i storage management, which is described in 13.5.1, “Hot data in an IBM i workload” on page 489.

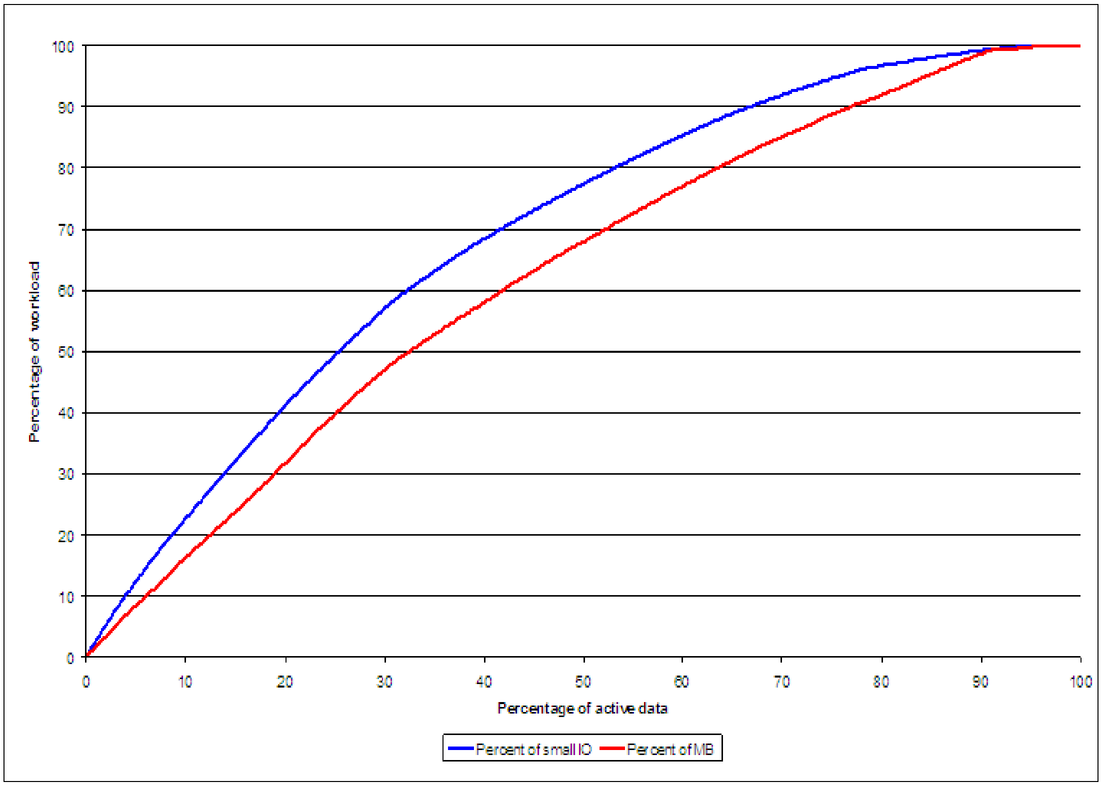


Figure 13-9   Skew level of the IBM i workload on active data

As shown in Figure 13-10 on page 495, the degree of skew for the same workload on all allocated extents is higher, because only 35% of the available disk space is occupied by IBM i data.

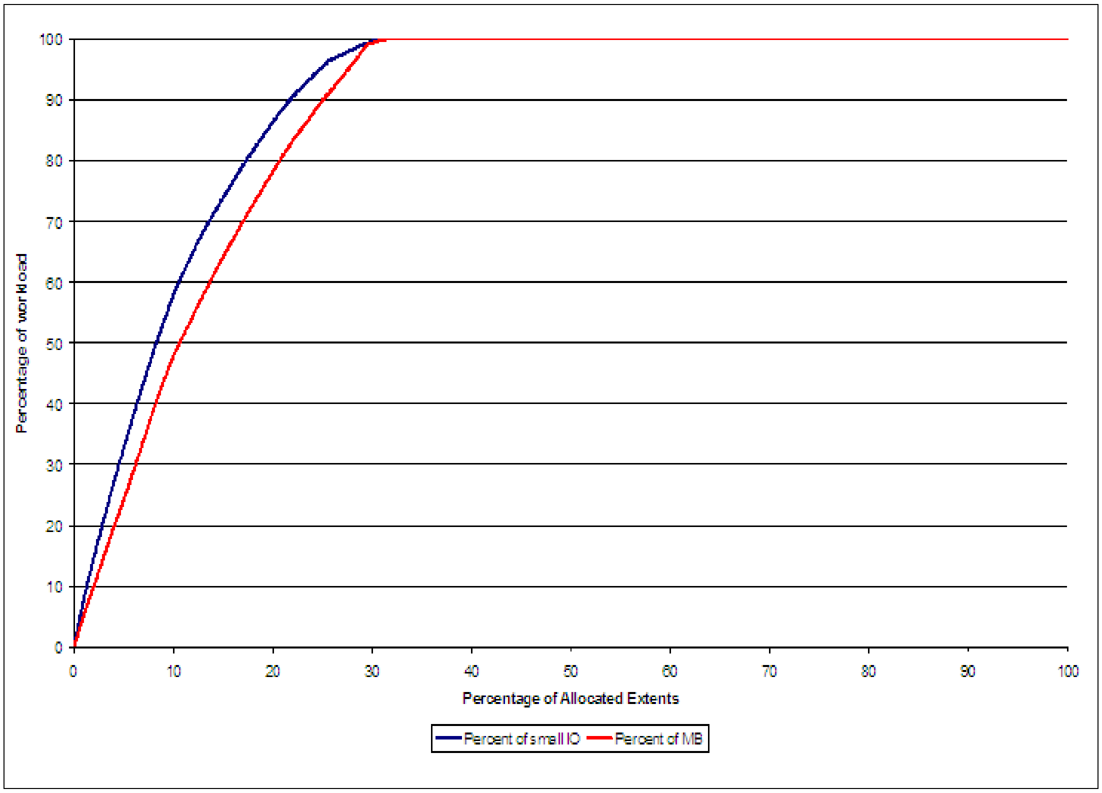


Figure 13-10 Skew level of the IBM i workload on allocated extents

13.5.4 Using Easy Tier with IBM i

IBM i installations that run their workloads in hybrid pools in the DS8000 benefit from Easy Tier in both planning for performance and actually improving the performance of the workload. In the planning phase, you can use *Storage Tier Advisory Tool (STAT)* to help you to determine the configuration and predict the performance improvement of the IBM i workload that runs the hybrid pool. Use this information with the skew level information described in 13.5.3, “Skew level of an IBM i workload” on page 492. In the production phase, Easy Tier relocates the
IBM i data to SSD and nearline disks to achieve the best performance and the most efficient spread of capacity.

Storage Tier Advisory Tool (STAT)

The STAT used on the IBM i data that is monitored by Easy Tier provides the heat map of the workload. The heat map shows the amount and distribution of the hot data across LUNs in the hybrid pool. For detailed information about the STAT, see 6.7, “Storage Tier Advisor Tool” on page 231.

Perform the following steps to use the STAT for an IBM i workload:

1. Enable the collection of the heat data I/O statistics by changing the Easy Tier monitor parameter to **`all`** or **`automode`**. Use the DS8000 command-line interface (DSCLI) command **`chsi -etmonitor all`** or **`chsi -etmonitor automode`**. The parameter **`-etmonitor all`** enables monitoring on all LUNs in the DS8000. The parameter **`-etmonitor automode`** monitors the volumes that are managed by Easy Tier automatic mode only.
2. Offload the collected data from the DS8000 clusters to the user workstation. Use either the DS8000 Storage Manager GUI or the DSCLI command **`offloadfile -etdata`**

`<directory>`, where <*directory*> is the directory where you want to store the files with the data on your workstation.

3. The offloaded data is stored in the following files in the specified directory:

   `SF75NT100ESS01_heat.data` and `SF75NT100ESS11_heat.data`

   The variable *75NT100* denotes the serial number of the DS8000 storage facility. The variables *ESS01* and *ESS11* are the processor complexes 0 and 1.

4. Use the STAT on your workstation to create the heat distribution report that can be presented in a web browser.

Figure 13-11 shows an example of the STAT heat distribution on IBM i LUNs after running the IBM i workload described in 13.5.3, “Skew level of an IBM i workload” on page 492. The hot and warm data is evenly spread across the volumes, which is typical for an IBM i workload distribution.

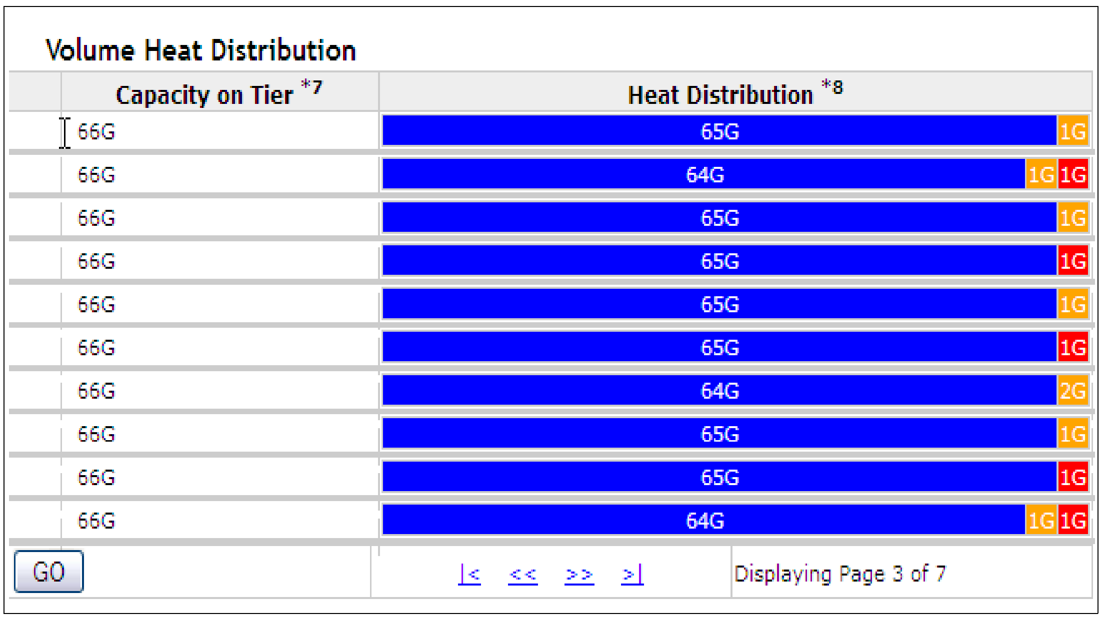


Figure 13-11 Heat distribution for IBM i workload

Relocating the IBM i data with Easy Tier

The Easy Tier relocation of IBM i data starts after a 24-hour learning period. The data is moved to SSD or nearline disks in 1 GB extents.

An IBM i client can also use the IBM i Media preference or ASP balancing method for hot-spot management. It is not our goal to compare the performance for the three relocation methods. However, we do not expect much difference in performance by using one or another method. Probably, factors such as ease of use, consolidation of the management method, or control over which data to move, are more important for an IBM i client to decide which method to use.

13.6 I/O Priority Manager with IBM i

It is common to use one storage system to serve many categories of workloads with different characteristics and requirements. I/O Priority Manager, which is a feature in IBM System Storage models DS8800 and DS8700, enables more effective storage performance management, by prioritizing access to Storage system resources for separate workloads. For more information about I/O Priority Manager, see 1.3.5, “I/O Priority Manager” on page 17.

Many IBM i clients run multiple IBM i workloads in different POWER partitions that share the disk space in the System Storage DS8000. The installations run important production systems and less important workloads for testing and developing. The other partitions can be used as disaster recovery targets of production systems in another location. We assume that IBM i centers with various workloads that share the DS8000 disk space use I/O Priority Manager to achieve a more efficient spread of storage resources.

Next, we show a simple example of using the I/O Priority Manager for two IBM i workloads.

The POWER partition ITSO_1 is configured with 4 processor units, 56 GB memory, and 48 × 70 GB LUNs in a DS8000 extent pool with Enterprise drives.

The partition ITSO-2 is configured with 1 processor unit, 16 GB memory, and 32 × 70 GB LUNs in a shared hybrid extent pool with all solid-state drives (SSDs), Enterprise drives, and nearline disk drives.

Both extent pools are managed by DS8000 Easy Tier.

We set up the I/O Priority Manager Performance Group 1 (PG1) for the volumes of ITSO_1 by using the DSCLI command:

```
chfbvol -perfgrp pg1 2000-203f
```

Performance Group 1 is defined for the 64 LUNs, but only 48 LUNs from these 60 LUNs are added to the ASP and used by the system ITSO_1; the other LUNs are in IBM i non-configured status.

We set up the I/O Priority Manager Performance Group 11 (PG11) for the volumes of ITSO_2 by using the DSCLI command:

```
chfbvol -perfgrp pg11 2200-221f
```

After we define the performance groups for the IBM i LUNs, we ran the IBM i benchmark workload described in 13.5.3, “Skew level of an IBM i workload” on page 492, with 40 job sets in each of the ITSO_1 and ITSO_2 partitions.

After the workload finished, we obtained the monitoring reports of each performance group, PG1 with the LUNs of ITSO_1 and PG11 with the LUNs of ITSO_2, during the 5-hour workload run with 15-minute monitoring intervals.

Figure 13-12 on page 498 and Figure 13-13 on page 498 show the DSCLI commands that we used to obtain the reports and the displayed performance values for each performance group. The workload in Performance Group PG11 shows different I/O characteristics than the workload in Performance Group 1. Performance Group PG11 also experiences relatively high response times compared to Performance Group 1. In our example, the workload characteristics and response times are influenced by the different priority groups, types of disk drives used, and Easy Tier management.

```
dscli> lsperfgrprpt -start 7h -stop 1h -interval 15m PG1
Date/Time: December 3, 2011 10:39:30 CET PM IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75TV181
time                grp                  resrc            avIO avMB   avresp pri avQ tgtQ %hlpT %dlyT %impt
===========================================================================================================
2011-12-03/15:45:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181    0 0.000  0.000    1 0   70   0     0     0
2011-12-03/16:00:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 5641 76.222 1.310    1 85  70   2     0     0
2011-12-03/16:15:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 3254 35.148 0.709    1 108 70   0     0     0
2011-12-03/16:30:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2167 19.292 1.026    1 102 70   0     0     0
2011-12-03/16:45:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 1879 15.958 0.885    1 107 70   0     0     0
2011-12-03/17:00:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 1913 16.794 0.896    1 104 70   0     0     0
2011-12-03/17:15:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 1925 17.063 0.884    1 108 70   0     0     0
2011-12-03/17:30:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 1998 17.684 1.084    1 106 70   0     0     0
2011-12-03/17:45:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 1997 17.648 1.101    1 106 70   0     0     0
2011-12-03/18:00:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 1991 17.639 1.110    1 106 70   0     0     0
2011-12-03/18:15:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2037 18.127 1.308    1 104 70   0     0     0
2011-12-03/18:30:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2069 18.473 1.345    1 104 70   0     0     0
2011-12-03/18:45:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2092 18.947 1.265    1 104 70   1     0     0
2011-12-03/19:00:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2109 19.286 1.253    1 105 70   0     0     0
2011-12-03/19:15:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2132 19.498 1.373    1 104 70   0     0     0
2011-12-03/19:30:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2150 19.734 1.391    1 103 70   0     0     0
2011-12-03/19:45:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2166 19.849 1.514    1 104 70   0     0     0
2011-12-03/20:00:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2202 20.330 1.326    1 104 70   0     0     0
2011-12-03/20:15:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2245 20.976 1.419    1 102 70   0     0     0
2011-12-03/20:30:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 2699 25.327 1.386    1 97  70   0     0     0
2011-12-03/20:45:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181 1552 17.402 5.481    1 34  70   4     0     0
2011-12-03/21:00:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181    1 0.024  0.250    1 100 70   0     0     0
2011-12-03/21:15:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181    1 0.025  0.280    1 100 70   0     0     0
2011-12-03/21:30:00 IBM.2107-75TV181/PG1 IBM.2107-75TV181    1 0.025  0.271    1 101 70   0     0     0
```

Figure 13-12 I/O Priority Management performance values for ITSO_1

```
dscli> lsperfgrprpt -start 7h -stop 1h -interval 15m PG11
Date/Time: December 3, 2011 10:40:10 CET PM IBM DSCLI Version: 7.6.10.511 DS: IBM.2107-75TV181
time                grp                   resrc             avIO  avMB    avresp pri avQ tgtQ %hlpT %dlyT %imp
===============================================================================================================
2011-12-03/15:45:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181    74 0.879   1.286   15 114 0    0     0     0
2011-12-03/16:00:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181  9040 101.327 5.499   15 25  0    1     0     101
2011-12-03/16:15:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 14933 158.168 2.153   15 30  0    0     0     100
2011-12-03/16:30:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 15288 154.254 1.431   15 37  0    0     0     0
2011-12-03/16:45:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 16016 159.503 1.364   15 40  0    0     0     100
2011-12-03/17:00:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 16236 158.533 1.193   15 43  0    0     0     100
2011-12-03/17:15:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 17980 169.486 1.302   15 34  0    0     0     0
2011-12-03/17:30:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 18711 170.260 1.061   15 43  0    0     0     0
2011-12-03/17:45:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 19190 171.624 1.061   15 45  0    0     0     0
2011-12-03/18:00:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 19852 173.986 1.051   15 44  0    0     0     0
2011-12-03/18:15:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 20576 177.721 0.952   15 39  0    0     0     0
2011-12-03/18:30:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 21310 180.309 0.827   15 50  0    0     0     0
2011-12-03/18:45:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 21712 184.416 1.008   15 43  0    0     0     100
2011-12-03/19:00:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 22406 188.075 0.972   15 38  0    0     0     0
2011-12-03/19:15:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 23120 190.645 0.717   15 44  0    0     0     0
2011-12-03/19:30:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 23723 193.182 0.718   15 46  0    0     0     0
2011-12-03/19:45:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 24607 198.093 0.734   15 35  0    0     0     0
2011-12-03/20:00:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 24714 198.473 0.754   15 42  0    0     0     0
2011-12-03/20:15:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 25256 201.734 0.778   15 49  0    0     0     0
2011-12-03/20:30:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181 25661 202.733 0.727   15 50  0    0     0     0
2011-12-03/20:45:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181  1729 15.483  6.485   15 41  0    2     3     164
2011-12-03/21:00:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181     4 0.019   0.707   15 94  0    0     0     0
2011-12-03/21:15:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181    11 0.049   0.431   15 99  0    0     0     0
2011-12-03/21:30:00 IBM.2107-75TV181/PG11 IBM.2107-75TV181     7 0.040   0.702   15 96  0    0     0     0
```

Figure 13-13 I/O Priority Management performance values for ITSO_2

For more information about IBM i performance with the DS8000 I/O Priority Manager, see *IBM i Shared Storage Performance using IBM System Storage DS8000 I/O Priority Manager*, WP101935, which is available on the IBM Techdocs library.

14

Performance considerations for System z servers

In this chapter, we describe the performance features and other enhancements that enable higher throughput and lower response time when connecting a DS8000 to your System z. We also review several of the monitoring tools and their usage with the DS8000.

14.1 Overview

The special synergy between the DS8000 features and the System z operating systems (mainly the z/OS Operating System) makes the DS8000 an outstanding performer in that environment.

The following specific DS8000 performance features relate to application I/O in a z/OS environment:

- Parallel Access Volumes (PAVs)
- Multiple allegiance
- I/O priority queuing
- I/O Priority Manager (IOPM)
- Logical volume sizes
- Fibre Channel connection (FICON)

In the following sections, we describe those DS8000 features and discuss how to best use them to boost performance.

Sample performance measurement data: Any sample performance measurement data provided in this chapter or the book is for comparative purposes only. Remember that the data was collected in controlled laboratory environments at a specific point in time by using the configurations, hardware, and firmware levels available at that time. Current performance in real-world environments varies. Actual throughput or performance that any user experiences also varies depending on considerations, such as the amount of multiprogramming in the user job stream, the I/O configuration, the storage configuration, and the workload processed. The data is intended only to help illustrate how different hardware technologies behave in relation to each other. Contact your IBM representative or IBM Business Partner if you have questions about the expected performance capability of IBM products in your environment.

14.2 Parallel Access Volumes

Parallel Access Volumes (PAVs) allow multiple concurrent I/Os to the same volume at the same time from applications that run on the same z/OS system image. This concurrency helps applications better share the logical volumes with reduced contention. The ability to send multiple concurrent I/O requests to the same volume nearly eliminates I/O queuing in the operating system, thus reducing I/O responses times.

Traditionally, access to highly active volumes involved manual tuning, splitting data across multiple volumes, and more actions to avoid those hot spots. With PAV and the z/OS Workload Manager, you can now almost forget about manual device level performance tuning or optimizing. The Workload Manager can automatically tune your PAV configuration and adjust it to workload changes. The DS8000 in conjunction with z/OS can meet the highest performance requirements.

PAV is implemented by defining alias addresses to the conventional base address. The alias address provides the mechanism for z/OS to initiate parallel I/O to a volume. An *alias* is another address/unit control block (UCB) that can be used to access the volume defined on the base address. An alias can be associated with a base address defined in the same logical control unit (LCU) only. The maximum number of addresses that you can define in an LCU is 256. Theoretically, you can define one base address, plus 255 aliases in an LCU.

14.2.1 Static PAV, Dynamic PAV, and HyperPAV

Aliases are initially defined to be associated to a certain base address. In a static PAV environment, the alias is always associated to the same base address. In a dynamic PAV environment and a HyperPAV environment, an alias can be reassigned to any base address as need dictates.

With dynamic PAV, you do not need to assign as many aliases in an LCU as compared to a static PAV environment, because the aliases are moved around to the base addresses that need an extra alias to satisfy an I/O request.

The z/OS Workload Manager (WLM) is used to implement dynamic PAVs. This function is called *dynamic alias management*. With dynamic alias management, WLM can automatically perform alias device reassignments from one base device to another base device to help meet its goals and to minimize I/Os queuing as workloads change.

WLM manages PAVs across all the members of a sysplex. When deciding on alias reassignment, WLM considers I/O from all systems in the sysplex. By default, the function is turned off, and must be explicitly activated for the sysplex through an option in the WLM service definition, and through a device-level option in the hardware configuration definition (HCD). Dynamic alias management requires your sysplex to run in WLM Goal mode.

In a HyperPAV environment, WLM is no longer involved in managing alias addresses. When the base address UCB is used by an I/O operation, each additional I/O against that address is assigned an alias that can be picked from a pool of alias addresses within the same LCU. This approach eliminates the latency caused by WLM having to manage the alias movement from one base address to another base address. Also, as soon as the I/O that uses the alias is finished, it drops that alias, which makes the alias available in the alias pool.

HyperPAV allows different hosts to use one alias to access different base addresses. This capability reduces the number of alias addresses required to support a set of base addresses in a System z environment.

14.2.2 HyperPAV compared to dynamic PAV test

The following simple test shows the advantage of HyperPAV compared to dynamic PAV. We use this test environment:

- The jobs that we run consist of six IEBGENER read jobs and five IEBDG write jobs.
- Every job uses its own dataset, and all datasets reside on one volume:
  - This way, all these jobs cause contention on the base address where the volume resides.
  - At the end of the job, the job resubmits itself, so that all the jobs keep running until they are canceled.
- On the dynamic PAV test, we reset the PAV to one, which means that the base address starts with no aliases assigned to it.

  With HyperPAV, the base address also starts with no aliases, because there is no I/O activity against that address.
- Every job is assigned the same WLM service class.

  The group of read and write jobs is assigned to a separate report class.

Figure 14-1 on page 500 shows the test result. The following observations can be concluded from this test result:

- Response time:
  - In the HyperPAV environment, the response time is about the same from the start of the test until the end of the test.
  - In the dynamic PAV environment, the response time is high initially, because the IOSQ is high due to the lack of aliases assigned to the base address. The aliases are added by WLM one at a time.
- I/O rate:
  - HyperPAV: The I/O rate reaches its maximum almost immediately. The I/O rate achieved is around 5780 IOPS.
  - Dynamic PAV: The I/O rate starts to increase from the beginning of the test and takes several minutes to reach its maximum of about 5630 IOPS.

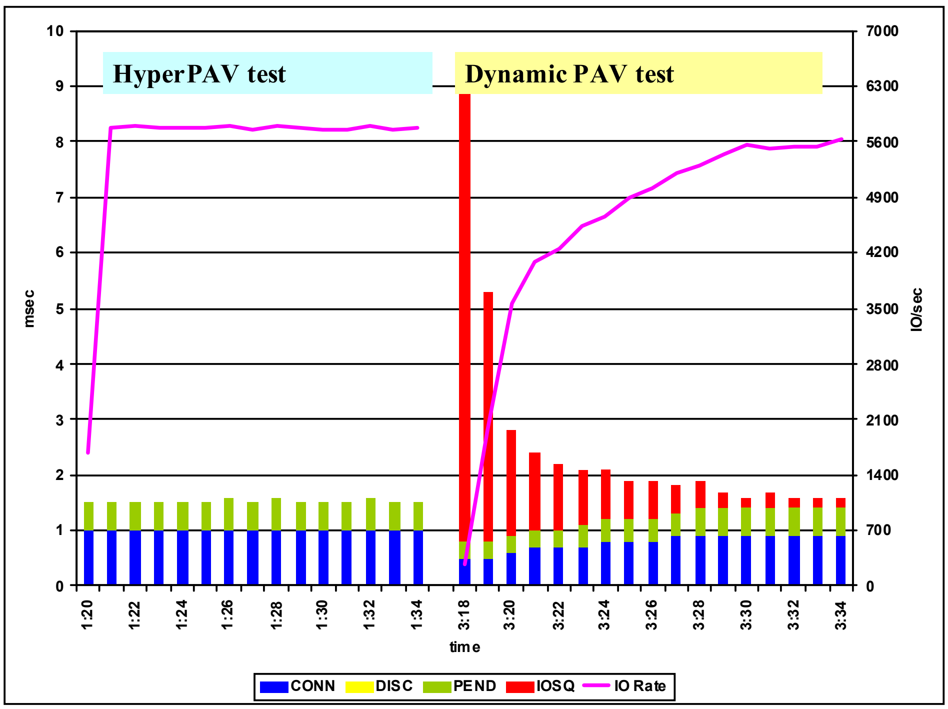


Figure 14-1 HyperPAV compared to dynamic PAV test result

Figure 14-2 on page 503 shows the number of PAVs assigned to the base address.

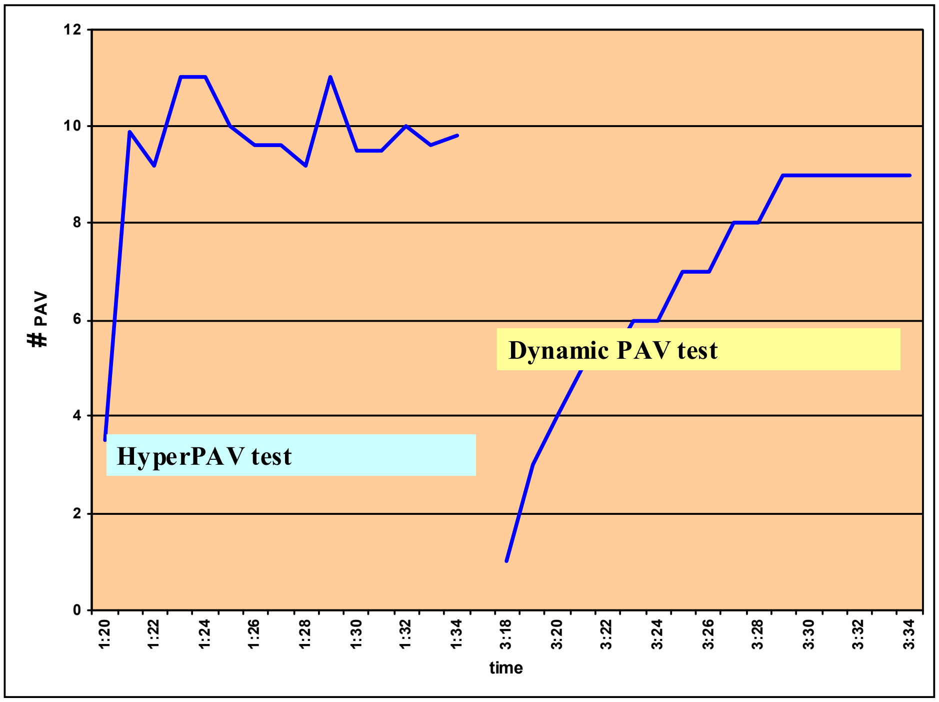


Figure 14-2 PAVs assigned by using HyperPAV and dynamic PAV

- HyperPAV: The number of PAVs almost immediately jumped to around 10 and fluctuates between 9 and 11.
- Dynamic PAV: The number of PAVs starts at one, and gradually WLM increases the PAV one at a time until it reaches a maximum number of nine.

In this test, we see that HyperPAV assigns more aliases compared to dynamic PAV. But, we also see that HyperPAV reaches a higher I/O rate compared to dynamic PAV. This test is an extreme test that tries to show how HyperPAV reacts to a high concurrent I/O rate to a single volume, as compared to how dynamic PAV responds to this condition.

The conclusion is that HyperPAV reacts immediately to a condition where there is a high concurrent demand on a volume. The other advantage of HyperPAV is that there is no overhead for assigning and releasing an alias for every I/O operation that needs an alias.

14.2.3 PAV and large volumes

By using queuing models, we can see the performance impact on the IOS queuing time when comparing a 3390-3 to larger volume sizes with various numbers of aliases. This modeling shows that one 3390-9 with three aliases (for a total of 4 UCBs) has less IOSQ time as compared to three 3390-3s with one alias each (for a total of 6 UCBs). Using larger volumes reduces the number of total UCBs required.

HyperPAV reduces the number of aliases required even further. *IBM Storage Advanced Technical Support* provides an analysis based on your Resource Measurement Facility (RMF). As a result of this study, they provide a recommendation of how many UCB aliases you need to define per LCU on your DS8000.

14.3 Multiple Allegiance

Normally, if a System z host image (server or logical partition (LPAR)) sends an I/O request to a device address for which the storage subsystem is already processing an I/O originating from another System z host image, the storage subsystem sends back a *device busy* indication and the I/O must be retried. This response delays the new request and adds to processor and channel overhead (this delay is reported in the RMF Device Activity Report PEND time column).

With older storage subsystems (before the DS8000 or ESS), a device has an implicit allegiance, that is, a relationship created in the disk control unit between the device and a channel path group, when an I/O operation was accepted by the device. The allegiance caused the control unit to guarantee access (no busy status presented) to the device for the remainder of the channel program over the set of paths associated with the allegiance.

The DS8000, because of Multiple Allegiance (MA), can accept multiple parallel I/O requests from different hosts to the same device address, increasing parallelism and reducing channel overhead. The requests are accepted by the DS8000 and all requests are processed in parallel, unless there is a conflict when writing data to the same extent of the count key data (CKD) logical volume. Still, good application access patterns can improve the global parallelism by avoiding reserves, limiting the extent scope to a minimum, and setting an appropriate file mask, for example, if no write is intended.

In systems without MA, all, except the first I/O request to a shared volume, are rejected, and the I/Os are queued in the System z channel subsystem, showing up as PEND time in the RMF reports.

MA provides significant benefits for environments running a sysplex or System z systems sharing access to volumes. MA and PAV can operate together to handle multiple requests from multiple hosts.

The DS8000 ability to run channel programs to the same device in parallel can dramatically reduce the IOSQ and the PEND time components in shared environments.

In particular, different workloads, for example, batch and online, running in parallel on different systems can unfavorably affect each other. In these cases, MA can dramatically improve the overall throughput.

14.4 How PAV and Multiple Allegiance work

These two functions allow multiple I/Os to execute concurrently against the same volume in a z/OS environment. In PAV, the I/Os come from the same LPAR or z/OS system. For MA, the I/Os come from different LPARs or z/OS systems.

First, we look at a disk subsystem that does not support both of these functions. If there is an outstanding I/O operation to a volume, all later I/Os must wait, as illustrated in Figure 14-3 on page 505. I/Os coming from the same LPAR wait in the LPAR, and this wait time is recorded in IOSQ Time. I/Os coming from different LPARs wait in the disk control unit and are recorded in Device Busy Delay Time, which is part of PEND Time.

In the DS8000, all these I/Os are executed concurrently by using PAV and MA, as shown in Figure 14-4 on page 505. I/O from the same LPAR is executed concurrently using UCB 1FF, which is an alias of base address 100. I/O from a different LPAR is accepted by the disk control unit and executed concurrently. All these I/O operations are satisfied from either the cache or one of the disk drive modules (DDMs) on a rank where the volume resides.

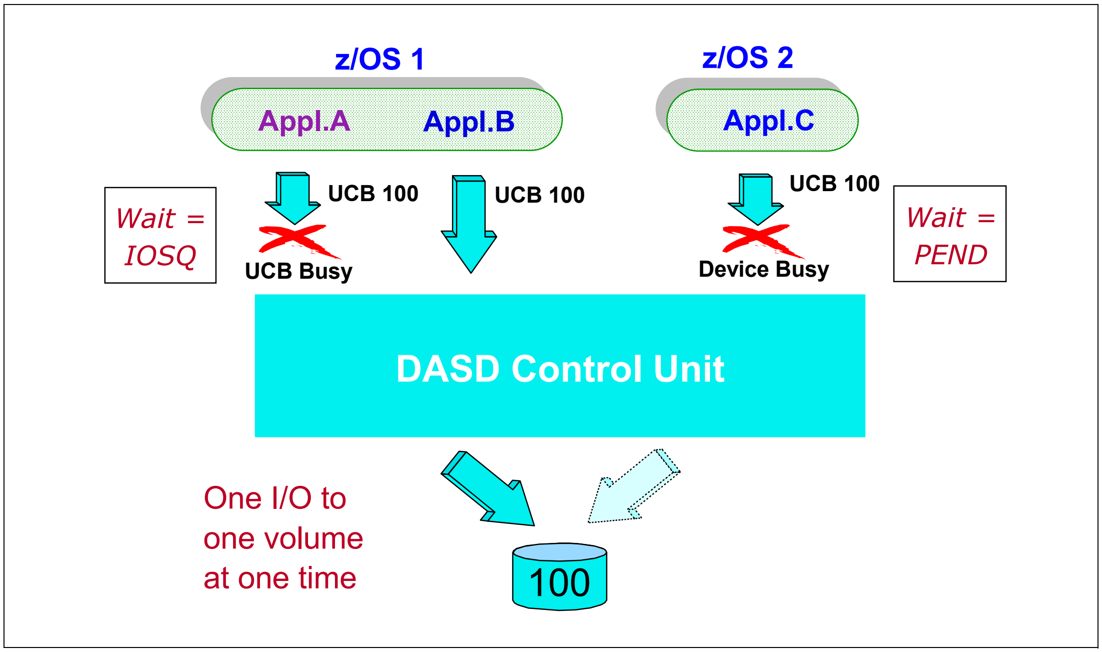


Figure 14-3 Concurrent I/O before PAV and Multiple Allegiance

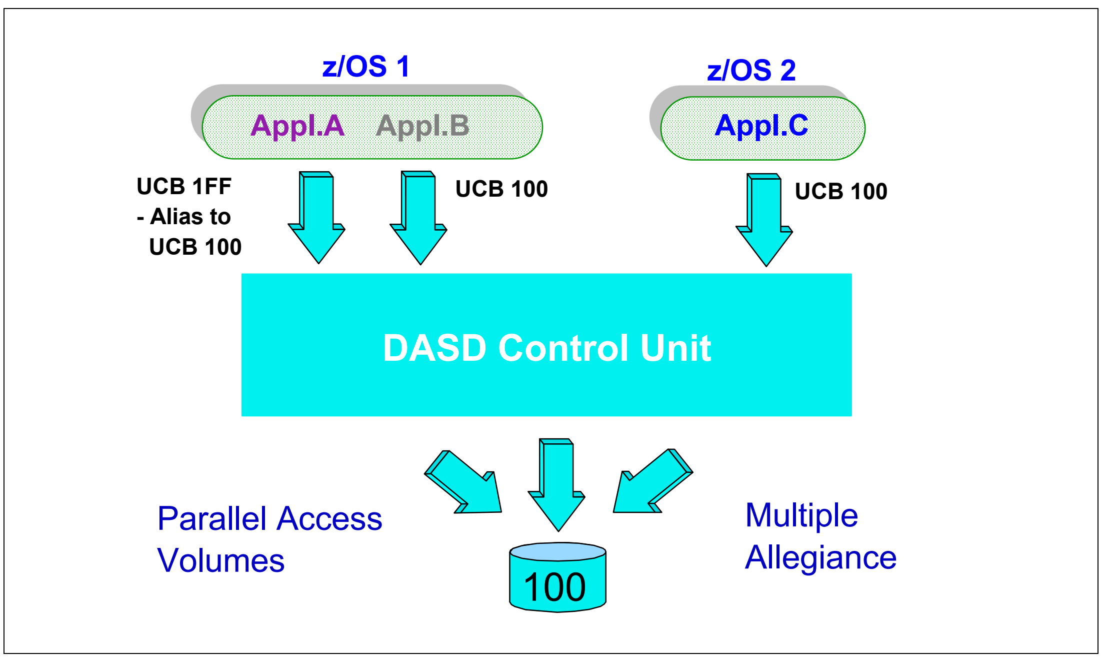


Figure 14-4 Concurrent I/O with PAV and Multiple Allegiance

14.4.1 Concurrent read operation

Figure 14-5 on page 506 shows that concurrent read operations from the same LPAR or different LPARs can be executed at the same time, even if they are accessing the same record on the same volume.

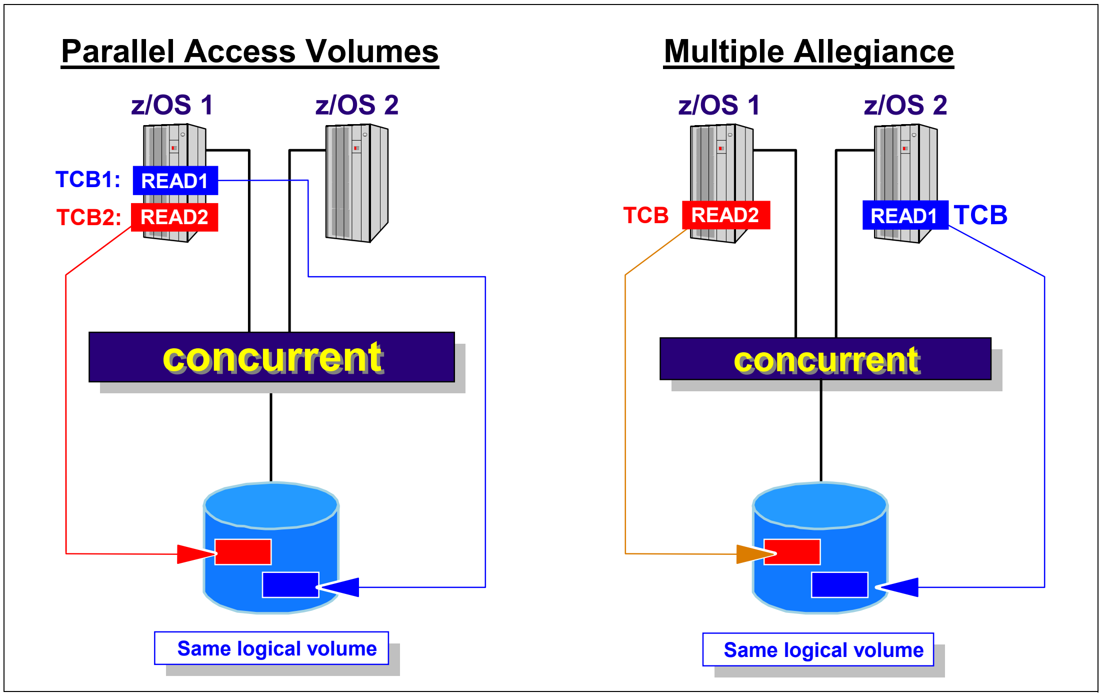


Figure 14-5 Concurrent read operations

14.4.2 Concurrent write operation

Figure 14-6 on page 507 shows the concurrent write operation. If the write I/Os are accessing different *domains* on the volume, all the write I/Os are executed concurrently. In case the write operations are directed to the same domain, the first write to that domain is executed and other writes to the same domain must wait until the first write finishes. This wait time is included in the Device Busy Delay time, which is part of PEND time.

Important: The domain of an I/O covers the specified extents to which the I/O operation applies. It is identified by the `Define extent` command in the channel program. The domain covered by the `Define extent` used to be much larger than the domain covered by the I/O operation. When concurrent I/Os to the same volume are not allowed, there is not an issue, because later I/Os waited anyway.

With the availability of PAV and MA, this extent conflict might prevent multiple I/Os from being executed concurrently. This extent conflict can occur when multiple I/O operations try to execute against the same domain on the volume. The solution is to update the channel programs so that they minimize the domain that each channel program is covering. For a random I/O operation, the domain must be the one track where the data resides.

If a write operation is being executed, any read or write to the same domain must wait. The same case happens if a read to a domain starts, later I/Os that want to write to the same domain must wait until the read operation is finished.

To summarize, all reads can be executed concurrently, even if they are going to the same domain on the same volume. A write operation cannot be executed concurrently with any other read or write operations that access the same domain on the same volume. The purpose of serializing a write operation to the same domain is to maintain data integrity.

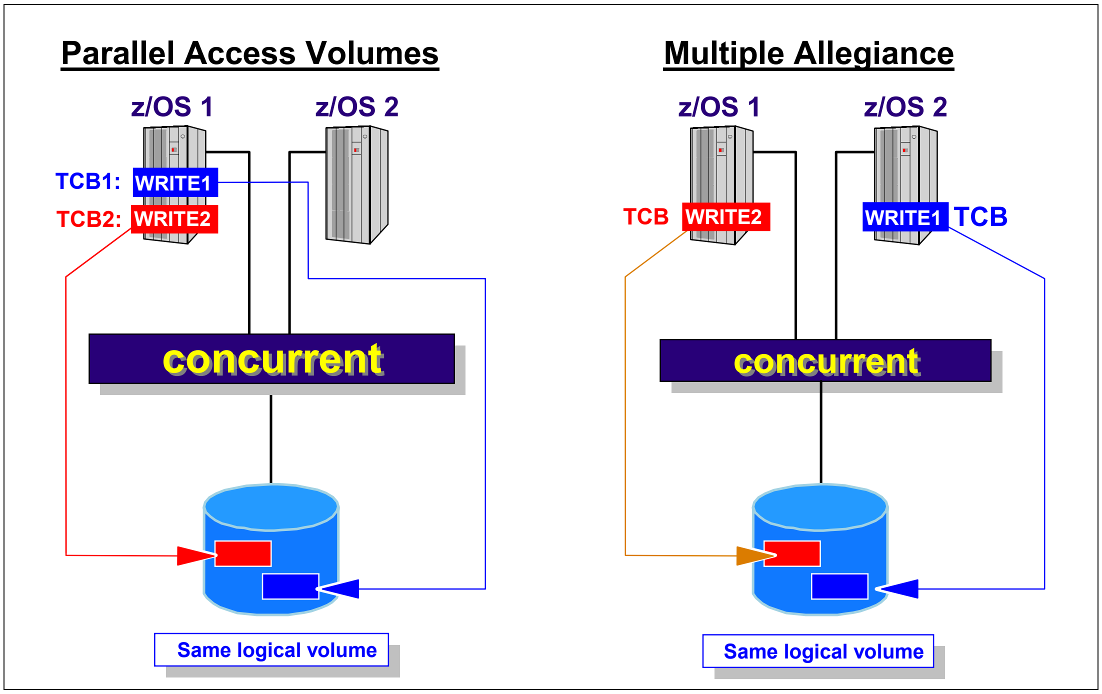


Figure 14-6 Concurrent write operation

14.5 I/O Priority Queuing

The DS8000 can manage multiple channel programs concurrently, as long as the data accessed by one channel program is not altered by another channel program. If I/Os cannot run in parallel, for example, due to extent conflicts, and must be serialized to ensure data consistency, the DS8000 internally queues I/Os.

Channel programs that cannot execute in parallel are processed in the order that they are queued. A fast system cannot monopolize access to a device also accessed from a slower system. Each system gets a fair share.

The DS8000 can also queue I/Os from different z/OS system images in a priority order. z/OS Workload Manager can use this prioritization to prioritize I/Os from one system against the others. You can activate I/O Priority Queuing in WLM Goal mode with the I/O priority management option in the WLM Service Definition settings.

When a channel program with a higher priority comes in and is put ahead of the queue of channel programs with lower priorities, the priorities of the lower priority programs are increased. This priority increase prevents high priority channel programs from dominating lower priority ones and gives each system a fair share, based on the priority assigned by WLM.

14.6 I/O Priority Manager

The I/O Priority Manager (IOPM) helps you to manage quality of service (QoS) levels for each application that runs on the system. This feature aligns distinct service levels to separate workloads in the system to help maintain the efficient performance of the DS8000 volume. The I/O Priority Manager together with Workload Manager (WLM) detects when a higher priority application is hindered by a lower priority application that competes for the same system resources. This contention can occur when multiple applications request data from the same drives. When IOPM encounters this situation, it delays lower priority I/O data to assist the more critical I/O data in meeting its performance targets.

The WLM integration for the I/O Priority Manager is available with z/OS V1.11 and higher, with the following necessary APARs: OA32298, OA34063, and OA34662.

With z/OS and zWLM software support, the user assigns application priorities through the Workload Manager. z/OS then assigns an *importance* value to each I/O, based on the zWLM inputs. Then, based on the prior history of I/O response times for I/Os with the same importance value, and based on the zWLM expectations for this response time, z/OS assigns an *achievement* value to each I/O. Importance and achievement values for each I/O are then compared, and the I/O becomes associated with a performance policy, independently of the volume performance group or policy. When a rank is overloaded, I/O is then managed according to the preassigned zWLM performance policy, that is, I/O Priority Manager begins throttling the IO with the lower priority.

For a more detailed explanation of I/O Priority Manager, see *DS8000 I/O Priority Manager,* REDP-4760.

14.7 Logical volume sizes

The DS8800 now supports CKD logical volumes of any size from one cylinder up to 1182006 cylinders. The term *custom volume* denotes that a user has the flexibility to select the size of a volume and does not need to match the size of the standard real devices, such as the 3339 cylinders of the 3390-3, or the 10017 cylinders of the 3390-9.

In addition to these standard models, the 3390-27 supports up to 32760 cylinders and the 3390-54 supports up to 65520 cylinders. With the availability of the Extended Address Volume (EAV), we now can support large volumes of up to 1182006 cylinders (1062 times the capacity of a 3390-1).

Maximum size increase: Originally, the maximum size for EAV volumes was 262668 cylinders. With R6.2 of the DS8000 and the introduction of EAV II volumes, the maximum size is increased to 1182006 cylinders.

14.7.1 Selecting the volume size

A key factor to consider when planning the CKD volumes configuration and sizes is the 256 device limit per logical subsystem (LSS). You need to define volumes with enough capacity, so that you can use all your installed capacity with at most 256 devices.

When planning the configuration, also consider future growth. You might want to define more alias addresses than needed, so that in the future you can add a rank on this LCU, if needed.

14.7.2 Larger volume compared to smaller volume performance

The performance of configurations that use larger custom volumes as compared to an equal total capacity configuration of smaller volumes was measured by using various online and batch workloads. In this section, we include measurement examples that can help you evaluate the performance implications of using larger volumes.

Benchmarks: Even though the benchmarks were performed on an ESS-F20, the comparative results are similar on the DS8000.

Random workload

The measurements for DB2 and IMS online transaction workloads in our measurements showed that there was only a slight difference in device response time between a six 3390-27 volume configuration compared to a 60 3390-3 volume configuration of equal capacity on the ESS-F20 using FICON channels.

The measurements for DB2 are shown in Figure 14-7. Even when the device response time for a large volume configuration is higher, the online transaction response time can sometimes be lower due to the reduced system overhead of managing fewer volumes.

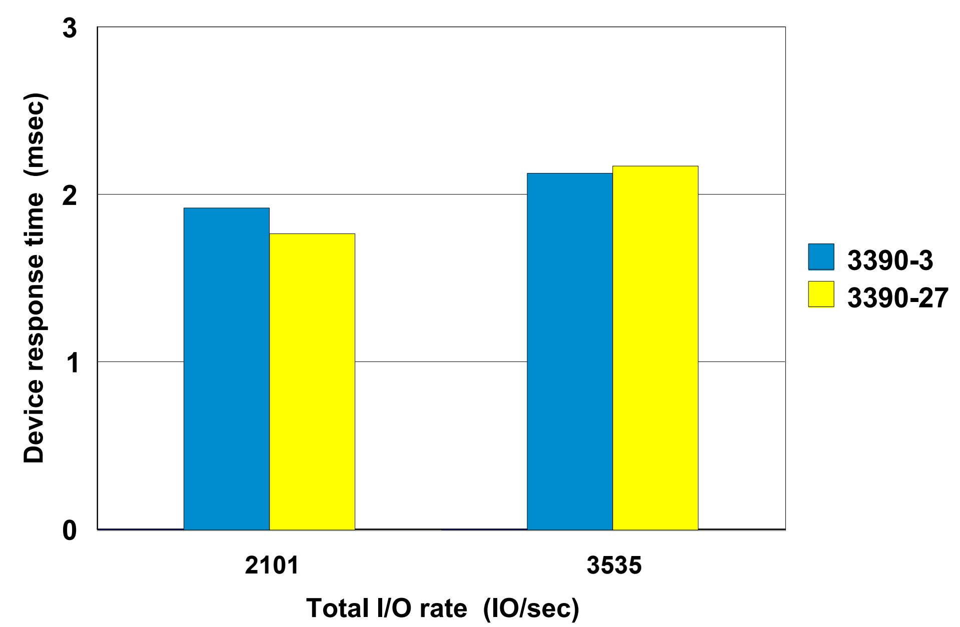


Figure 14-7 DB2 large volume performance

The measurements were carried out so that all volumes were initially assigned with zero or one alias. WLM dynamic alias management then assigned additional aliases, as needed. The number of aliases at the end of the test run reflects the number that was adequate to keep IOSQ down. For this DB2 benchmark, the alias assignment done by WLM resulted in an approximately 4:1 reduction in the total number of UCBs used.

Sequential workload

Figure 14-8 on page 510 shows elapsed time comparisons between nine 3390-3s compared to one 3390-27 when a DFSMSdss full volume physical dump and full volume physical restore are executed. The workloads were run on a 9672-XZ7 processor connected to an ESS-F20 with eight FICON channels. The volumes are dumped to or restored from a single 3590E tape with an A60 Control Unit with one FICON channel. No PAV aliases were assigned to any volumes for this test, even though an alias might improve the performance.

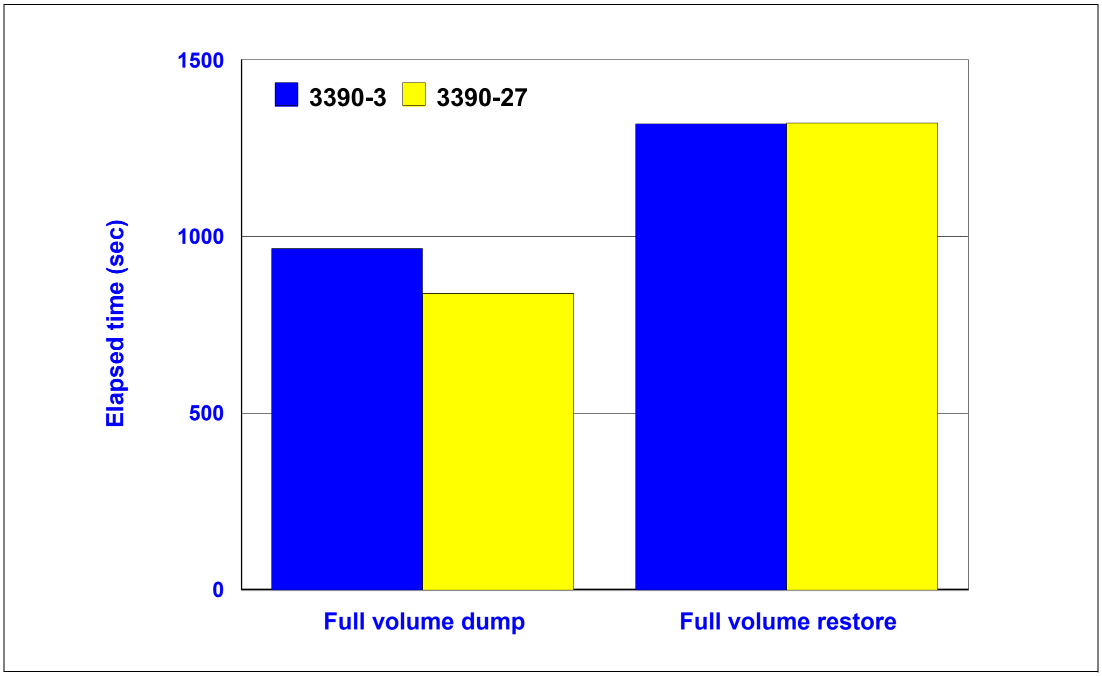


Figure 14-8 DSS dump large volume performance

14.7.3 Planning the volume sizes of your configuration

From a simplified storage management perspective, we suggest that you select and use a uniform volume size for most of your volumes. With a uniform volume size configuration in your DS8000, you do not have to track the size of each of your volumes. Several functions, such as FlashCopy, Peer-to-Peer Remote Copy (PPRC), and full volume restores, require that the target volume cannot be smaller than the source, which simplifies and avoids mistakes in your storage administration activities.

Larger volumes

To avoid potential I/O bottlenecks when using large volumes, you might also consider the following suggestions:

- Use PAVs to reduce IOS queuing.

  Parallel Access Volume (PAV) is of key importance when using large volumes. PAV enables one z/OS system image to initiate multiple I/Os to a device concurrently, which keeps IOSQ times down even with many active datasets on the same volume. PAV is a practical *must* with large volumes. In particular, we suggest using HyperPAV.

- Multiple Allegiance is a function that the DS8000 automatically provides.

  Multiple Allegiance automatically allows multiple I/Os from different z/OS systems to be executed concurrently, which reduces the Device Busy Delay time, which is part of PEND time.

- Eliminate unnecessary reserves.

  As the volume sizes grow larger, more data and datasets reside on a single CKD device address. Thus, having a few large volumes can reduce performance when there are significant and frequent activities on the volume that reserve the entire volume or the VTOC/VVDS.

- Certain applications might use poorly designed channel programs that define the whole volume or the whole extent of the dataset it is accessing as their extent range or *domain*, instead of just the actual track on which the I/O operates. This design prevents other I/Os from running simultaneously, if a write I/O is being executed against that volume or dataset, even when PAV is used. You need to identify such applications and allocate the datasets on volumes where they do not conflict with other applications. Custom volumes are an option. For an Independent software vendor (ISV) product, asking the vendor for an updated version might help solve the problem.

Consider these other benefits of using large volumes:

- Reduced number of required UCBs. We reduce the number of UCBs by consolidating smaller volumes to larger volumes, and we also reduce the number of total required aliases, as explained in 14.2.3, “PAV and large volumes” on page 503.
- Simplified storage administration.
- Reduced number of X37 abends and allocation failures due to larger pools of free space.
- Reduced number of multivolume datasets to manage.

14.8 FICON

The DS8800 storage system connects to System z hosts by using FICON channels, with the addition of Fibre Channel Protocol (FCP) connectivity for Linux on System z hosts.

FICON provides simplified system connectivity and high throughputs. The high data rate allows short data transfer times, improving overall batch window processing times and response times especially for data stored using large blocksizes. The pending time component of the response time, which is caused by director port busy, is eliminated, because collisions in the director are eliminated with the FICON architecture.

Another performance advantage delivered by FICON is that the DS8000 accepts multiple channel command words (CCWs) concurrently without waiting for completion of the previous CCW. Therefore, the setup and execution of multiple CCWs from a single channel happen concurrently. Contention among multiple I/Os that access the same data is now handled in the FICON host adapter and queued according to the I/O priority indicated by the Workload Manager.

FICON Express2 and FICON Express4 channels on the z9® EC and z9 BC systems introduced the support to the Modified Indirect Data Address Word (MIDAW) facility and a maximum of 64 open exchanges per channel. The previous maximum of 32 open exchanges was available on FICON, FICON Express, and IBM System z 990 (z990) and IBM System z 890 (z890) FICON Express2 channels.

Significant performance advantages can be realized by users that access the data remotely. FICON eliminates *data rate droop effect* for distances up to 100 km (62.1 miles) for both read and write operations by using enhanced data buffering and pacing schemes. FICON thus extends the DS8000 ability to deliver high bandwidth potential to the logical volumes that need it, when they need it.

For additional information about FICON, see 8.3.1, “FICON” on page 320.

14.8.1 Extended Distance FICON

For the DS8000, Extended Distance FICON, also known as *Simplified FICON Channel Extension*, is an enhancement to the FICON architecture. Extended Distance FICON is designed to help enable greater throughput over distance for IBM z/OS Global Mirror (XRC) by eliminating handshakes between the channel and the control unit. It can eliminate performance degradation at extended distances, having no channel extender installed, by implementing a new information unit (IU) pacing protocol. You can use a less complex and cheaper channel extender, which only performs frame forwarding, rather than a channel extender that dissects every channel control word (CCW) to optimize the transfer through the channel extender to get the best performance. These channel extenders typically have extended remote copy (XRC) emulation running on them.

An enhancement is implemented on the standard FICON architecture, the Fibre Channel single-byte command set 3 (FC-SB-3), which is a layer with a new protocol for persistent IU pacing. This enhancement is achieved for the DS8000 z/OS Global Mirror XRC, System Data Mover (SDM) read record set (RRS) data transfer from a DS8000 to the SDM host address space. The control units that support the extended distance FICON feature can increase the pacing count. The pacing count is the number of IUs that are *inflight* from a channel to a control unit.

Standard FICON supports IU pacing of 16 IUs in flight. Extended Distance FICON now extends the IU pacing for the RRS CCW chain to permit 255 IUs inflight without waiting for an acknowledgement from the control unit, eliminating engagement between the channel and control unit. This support allows the channel to remember the last pacing information and use this information for later operations to avoid performance degradation at the start of a new I/O operation.

Improved IU pacing with 255 IU instead of 16 improves the utilization of the link. For a 4 Gbps link, the channel remains used at a distance of 50 km (31.06 miles) and improves the distance between servers and control units.

Extended Distance FICON reduces the need for channel extenders in the DS8000 series 2-site and 3-site z/OS Global Mirror configurations because of the increased number of read commands simultaneously inflight. This capability provides greater throughput over distance for IBM z/OS Global Mirror (XRC) using the same distance.

Extended Distance FICON does not extend the achievable physical FICON distances, or offer any performance enhancements in a non-z/OS GM (XRC) environment.

Extended Distance FICON is, at this time, only supported on z10, z Enterprise 196, and z144 channels. FICON Express8S, FICON Express8, FICON Express4, and FICON Express2 are supported.

For more information about Extended Distance FICON, see *IBM System Storage DS8000 Host Attachment and Interoperability*, SG24-8887.

14.8.2 High Performance FICON

High Performance FICON (zHPF) is a protocol extension of FICON, which communicates in a single packet a set of commands that need to be executed (it looks like a Small Computer System Interface (SCSI) command descriptor block from an interface standpoint). It allows the control unit to stream the data for multiple commands back in a single data transfer section for I/Os initiated by various access methods, which improves the channel throughput on small block transfers. Since the early implementation in 2009, zHPF is improved to cover a wide range of applications.

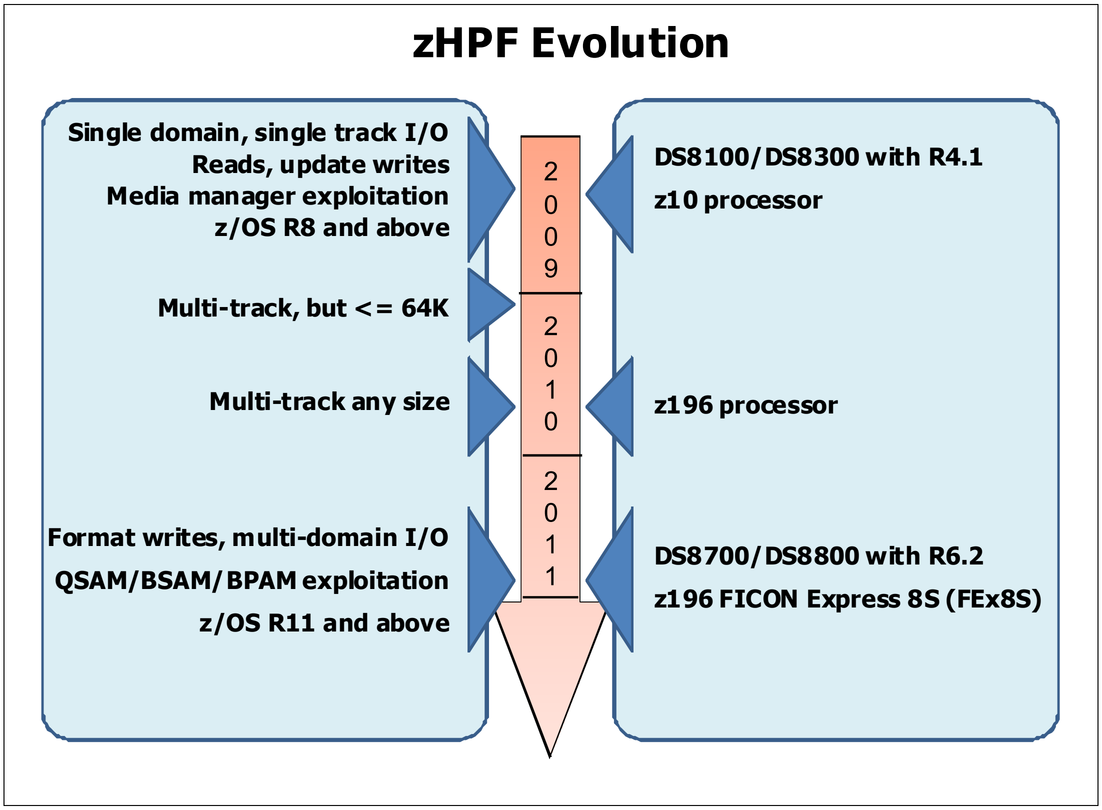


Figure 14-9 zHPF evolution

Figure 14-9 shows the major steps in the zHPF evolution. The latest generation of zHPF architecture introduced important enhancements:

- QSAM/BSAM/BPAM support:
  - Allows DSNTYPE=BASIC/LARGE datasets to use zHPF, and to achieve equal or better than Extended Format datasets
  - Partitioned datasets (but not “search key” I/Os for the directory)
  - VTOC reads that use BSAM
- Format Writes support:
  - Especially important when using 4K or 8K page sizes
  - Important to DB2 utility loads and restores, as well as Copy and Rebuild Index
  - Important to batch jobs in general
- DB2 list prefetch support:
  - Important for DB2 queries when index and data are disorganized
  - Enables new caching/disk algorithm, to be named later

Currently zHPF supports two major families of data:

- Media Manager data:
  - DB2
  - VSAM
  - Non-VSAM Extended Format (EF) datasets
  - IMS Fast Path
  - PDSE
  - HFS

  – z/FS
  – MVS™ system logger
- QSAM, BPAM, and BSAM data

We summarize the benefits of the zHPF architecture compared to the standard FICON architecture:

- Improve execution of small block I/O requests.
- Improve the efficiency of channel processors, host bus interfaces, and adapters by reducing the number of Information Units that need to be processed, increasing the average bytes per frame, and reducing the number of frames per I/O that need to be transferred.
- Improve reliability, availability, and serviceability (RAS) characteristics by enabling CU to provide additional information for fault isolate with an interrogate mechanism to query CU for a missing interrupt.
- zHPF multi-track data transfers support up to 256 tracks in a single transfer operation. Eliminating the 64 K byte limit and allowing a full exploitation of FICON Express8S and FICON Express8 available bandwidth.
- Increased performance for the zHPF protocol:
  – FICON Express8S uses a hardware data router that increases the performance for zHPF.
  – The FICON Express8, FICON Express4, and FICON Express2 channels use firmware-only zHPF implementation.

Enhancements made to both the z/Architecture and the FICON interface architecture deliver optimizations for online transaction processing (OLTP) workloads by using channel Fibre ports to transfer commands and data. zHPF helps to reduce the overhead that relates to the supported commands and improve performance.

From the z/OS point of view, the existing FICON architecture is called *command mode*, and zHPF architecture is called *transport mode*. Both modes are supported by the FICON Express4 and FICON Express2 features. A parameter in the Operation Request Block (ORB) is used to determine whether the FICON channel is running in command or transport mode.

There is a microcode requirement on both the CU side and the channel side to enable zHPF dynamically at the links level. The links basically initialize themselves to allow zHPF as long as both sides support zHPF. zHPF is exclusive to Systems zEnterprise 196, z114, z10 EC, and z10 BC.

LMC levels: The zHPF support for QSAM/BSAM/BPAM, Format Writes, and DB2 list prefetch needs the LMC 7.60.2.*xx.xx* or higher for DS8800 or LMC 6.60.2.*xx.xx* or higher for DS8700. Furthermore, all these enhancements are supported only on Systems z Enterprise 196 or System z114.

zHPF is available as a licensed feature (7092 and 0709) on the DS8800. Currently, the following releases of z/OS for zHPF are supported:

- z/OS V1.11 and higher with PTFs
- z/OS V1.8, V1.9, and V.10 with the IBM Lifecycle Extension with PTFs

zHPF settings

To enable or disable the zHPF, the following new statement is introduced to the IECIOSxx member of z/OS parmlib:

```
ZHPF=YES | NO
```

This setting can be dynamically changed using the SETIOS MVS command. Furthermore, to use the zHPF support for QSAM/BSAM/BPAM, the following statement is introduced to the IGDSMSxx member of z/OS parmlib:

```
SAM_USE_HPF(YES | NO)
```

Also, this setting can be dynamically changed by using the SETSMS MVS command.

Finally, to verify whether zHPF is enabled on your system, use the following command:

```
/D IOS,zHPF

RESPONSE=SYSX

 IOS630I 13.55.00 ZHPF FACILITY 011

 FICON CHANNEL EXTENSIONS FACILITY IS ENABLED


/D IOS,zHPF

RESPONSE=SYSY

 IOS630I 13.57.00 ZHPF FACILITY 063

 FICON CHANNEL EXTENSIONS FACILITY IS DISABLED


/D IOS,zHPF

RESPONSE=SYSZ

 IOS630I 13.59.00 ZHPF FACILITY 118

 HIGH PERFORMANCE FICON FACILITY NOT SUPPORTED BY PROCESSOR
```

zHPF performance

Figure 14-10 on page 516 and Figure 14-11 on page 516 show a performance comparison among the various generations of FICON channels with and without zHPF.

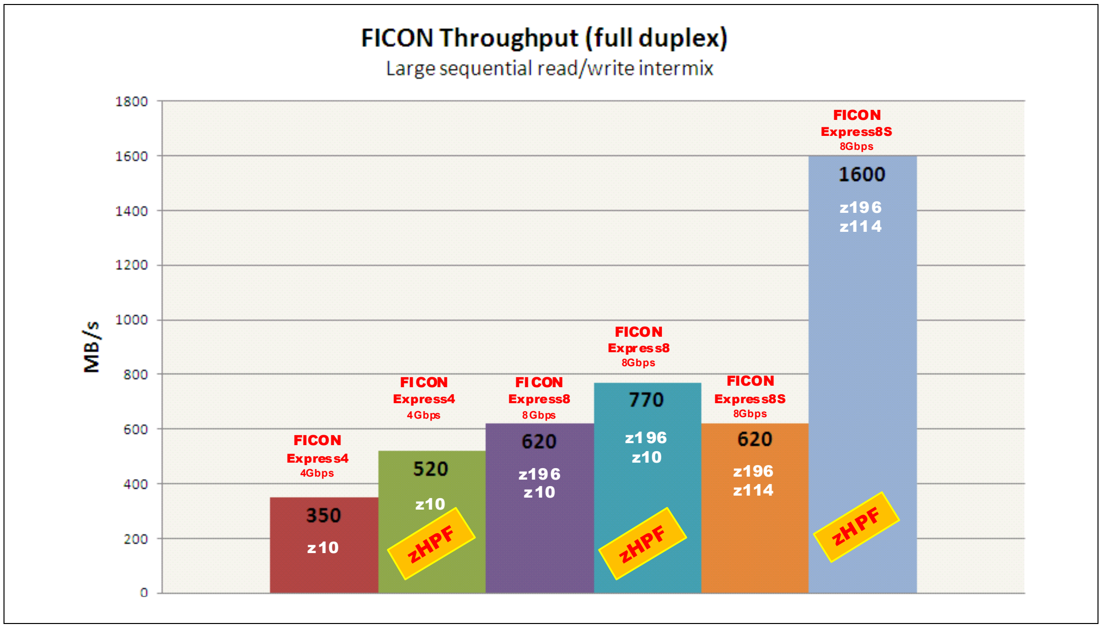


Figure 14-10 FICON throughput comparison

Figure 14-10 shows that zHPF brought significative improvement to the throughput since the introduction of the new protocol with FICON Express4 channels. The most impressive improvement is with the FICON Express8S channels running zHPF where the throughput increases 158% and 108% compared to FICON Express8S without zHPF and FICON Express8 with zHPF.

Figure 14-11 shows the improvements to the I/O per second. In this case, the improvement with the FICON Express8S channels running zHPF is 300% and 77% compared to FICON Express8S without zHPF and FICON Express8 with zHPF.

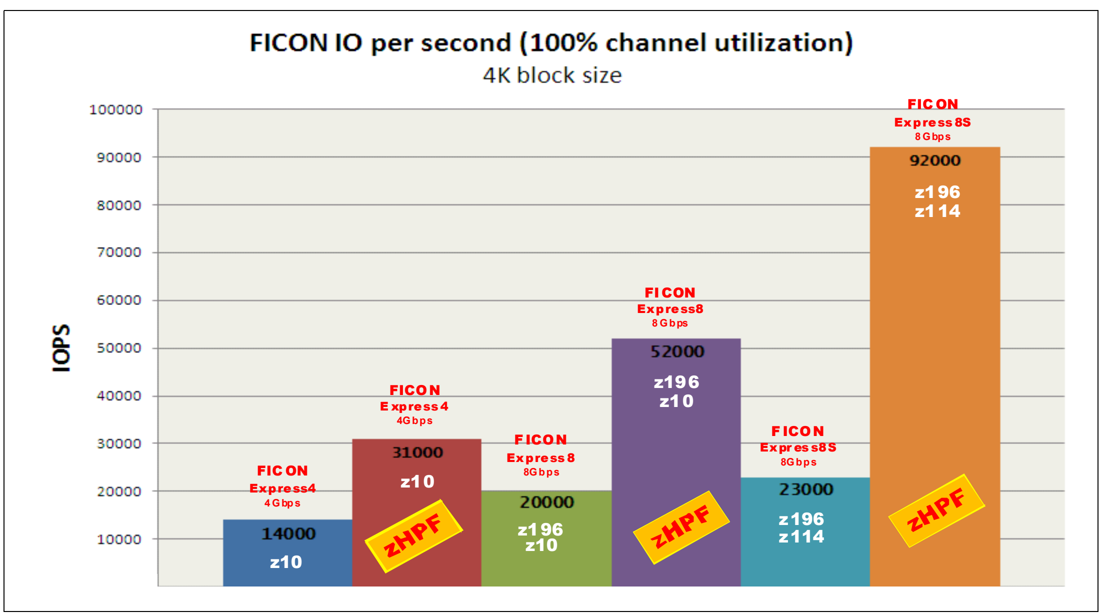


Figure 14-11 FICON I/O per second comparison

With the latest generation of zHPF and DS8000 microcode Version R6.2, the xSAM access methods (QSAM/BSAM/BPAM) benefit from this enhanced architecture. Figure 14-12 on page 517 and Figure 14-13 report comparisons between FICON and zHPF in a BSAM/QSAM test case.

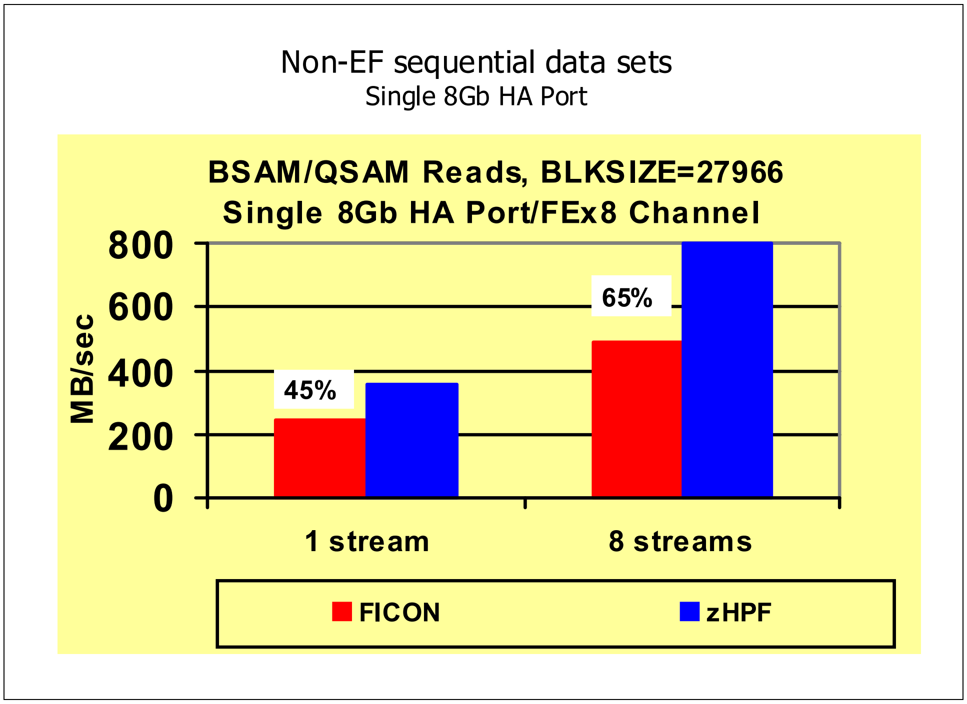


Figure 14-12 BSAM/QSAM reads test results

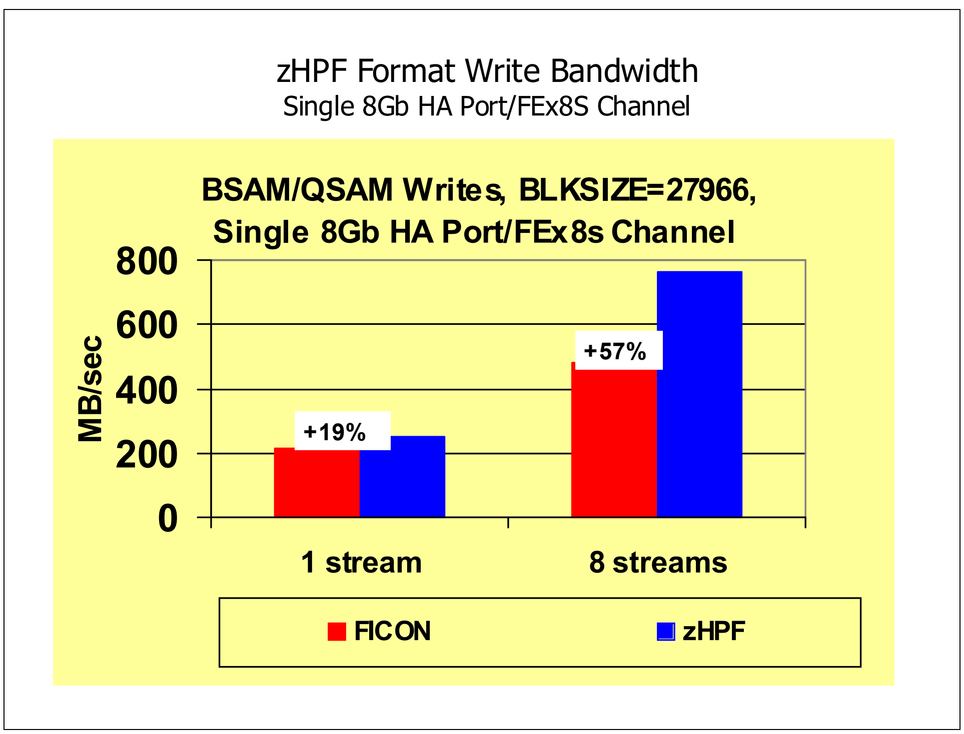


Figure 14-13 BSAM/QSAM writes (with Format Writes) test results

14.8.3 MIDAW

The IBM System z9® server introduced a Modified Indirect Data Address Word (MIDAW) facility, which in conjunction with the DS8000 and the FICON technology channels, delivers enhanced I/O performance for Media Manager applications.

The MIDAW facility is a modification to a channel programming technique from S/360 days. MIDAW is a method of gathering and scattering data into and from noncontiguous storage locations during an I/O operation, thus decreasing channel, fabric, and control unit overhead by reducing the number of channel command words (CCWs) and frames processed. There is no tuning needed to use this MIDAW facility. The following requirements are the minimum requirements to take advantage of this MIDAW facility:

- z9 server.
- Applications that use Media Manager.
- Applications that use long chains of small blocks.
- The biggest performance benefit comes with FICON Express4 channels running on 4 Gbps links, especially when processing extended format datasets.

If chains of small records are processed, MIDAW can improve FICON Express4 performance if the I/Os use Media Manager. Figure 14-14 shows improved FICON Express4 performance for a 32x4k READ channel program.

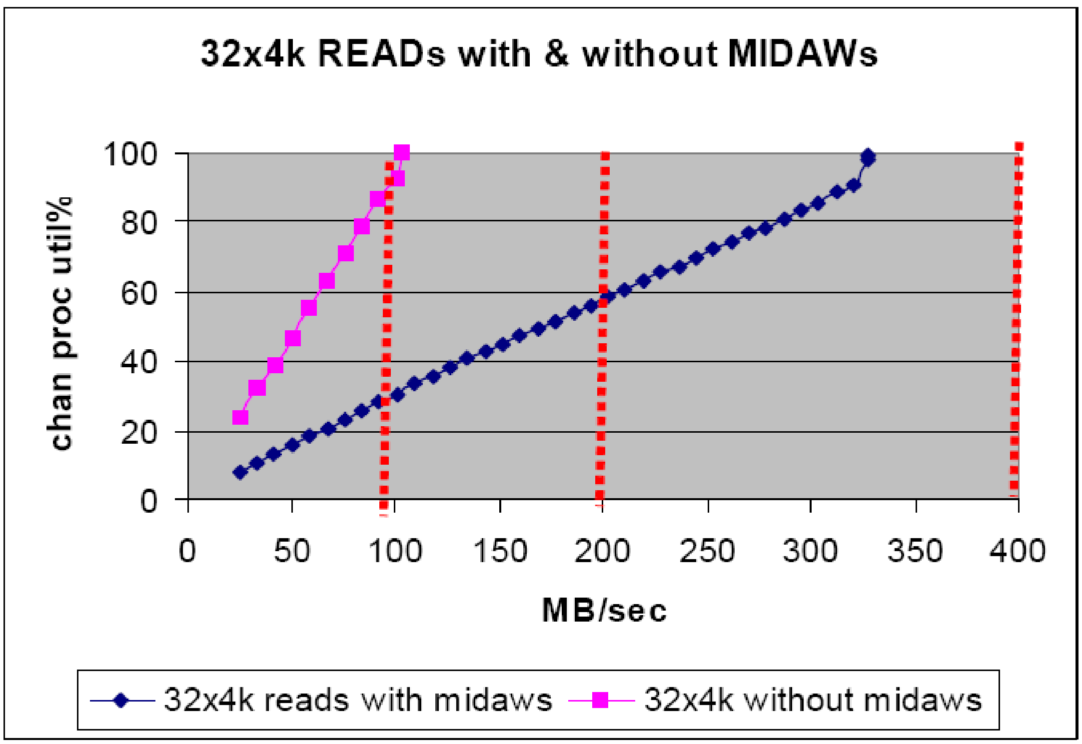


Figure 14-14 Channel utilization limits for hypothetical workloads

Without MIDAWs, a FICON Express4 channel is pushed to 100% channel processor utilization at just over 100 MBps of throughput, which is about the limit of a 1 Gigabit/s FICON link. Two FICON Express4 channels are needed to get to 200 MBps with a 32x4k channel program.

With MIDAWs, 100 MBps is achieved at only 30% channel utilization, and 200 MBps, which is about the limit of a 2 Gigabit/s FICON link, is achieved at about 60% channel utilization. A FICON Express4 channel operating at 4 Gigabit/s link speeds can achieve over 325 MBps.

We measured this performance with a single FICON Express4 channel connected to a FICON director to two 4 Gigabit/s Control Unit (CU) ports.

14.9 z/OS planning and configuration guidelines

This section discusses general configuration guidelines and suggestions for planning the DS8000 configuration. For a less generic and more detailed analysis that takes into account your particular environment, the Disk Magic modeling tools are available to IBM personnel and IBM Business Partners who can help you in the planning activities. Disk Magic can be used to help understand the performance effects of various configuration options, such as the number of ports and host adapters, disk drive capacity, and number of disks. See 6.1, "Disk Magic" on page 176.

14.9.1 Channel configuration

The following generic guidelines can be complemented with the information in 8.3, "Attaching IBM System z hosts" on page 320 and 4.10.1, "I/O port planning considerations" on page 147:

- If you use eight FICON channels, define all eight FICON channels as an 8-path-group to all the volumes on the DS8000.
- If you decide to use more than eight FICON channels, divide the FICON channels evenly between the LCUs, so that LCUs on the same processor complex (server) are assigned to the same channel-path-group.

The maximum throughput of a FICON 2 Gb port on the DS8000 with zHPF is 200 MB/s. The maximum throughput of a FICON 4 Gb port on the DS8000 with zHPF is 400 MB/s. The maximum throughput of a FICON 8 Gb port on the DS8000 with zHPF is 800 MB/s. The maximum throughput of FICON Express channels on the System z servers is shown in Figure 14-10 on page 516.

Considering, for instance, that the maximum throughput of a DS8000 FICON 8 Gb port is not that much higher than the maximum throughput of a FICON Express8 channel using zHPF, in general, we do not advise daisy chaining several FICON channels from multiple CECs onto the same DS8000 I/O port.

Daisy chaining can go either way:

- Connecting more than one FICON channel to one DS8000 port
- Connecting one FICON channel to more than one DS8000 port

However, if you have multiple DS8000s installed, it might be a good option to balance the channel load on the System z server. You can double the number of required FICON ports on the DS8000s and daisy chain these FICON ports to the same channels on the System z server. This design provides the advantage of being able to balance the load on the FICON channels, because the load on the DS8000 fluctuates during the day.

Figure 14-15 on page 520 shows configuration A with no daisy chaining. In this configuration, each DS8000 uses 8 FICON ports and each port is connected to a separate FICON channel on the host. In this case, we have two sets of 8 FICON ports connected to 16 FICON channels on the System z host.

In configuration B, we double the number of FICON ports on both DS8000s and keep the same number of FICON channels on the System z server. We can now connect each FICON channel to two FICON ports, one on each DS8000. Configuration B offers this advantage:

- Workload from each DS8000 is now spread across more FICON ports, which lowers the load on the FICON ports and FICON host adapters.
- Any imbalance in the load that is going to the two DS8000s is now spread more evenly across the 16 FICON channels.

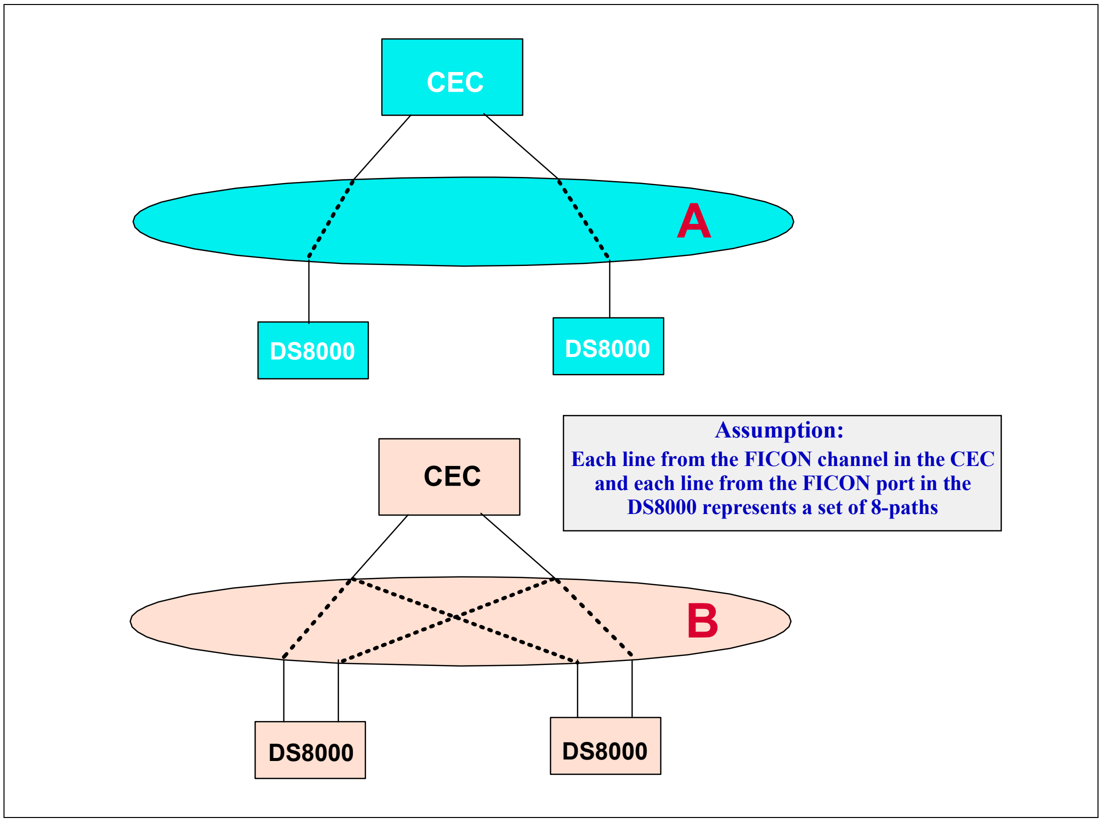


Figure 14-15 Daisy chaining DS8000s

14.9.2 Considerations for mixed workloads

With the larger capacity of the DS8000, you can combine data and workloads from several different types of independent servers into a single DS8000. The following examples are mixed workloads:

- z/OS and Open Systems
- Mission-critical production and test

Sharing resources in a DS8000 has advantages from a storage administration and resource-sharing perspective, but it has implications for workload planning. Resource-sharing has the benefit that a larger resource pool (for example, disk drives or cache) is available for critical applications. However, be careful to ensure that uncontrolled or unpredictable applications do not interfere with mission-critical work.

If you have a workload that is truly mission-critical, consider isolating it from other workloads, particularly if those other workloads are unpredictable in their demands.

There are several ways to isolate the workloads:

- Place the data on separate DS8000s. This option provides maximum isolation between the different workloads.
- Place the data on separate DS8000 servers. This option isolates the use of memory buses, microprocessors, and cache resource. However, before doing that, make sure that a *half DS8000* provides sufficient performance to meet the needs of your important application. Disk Magic provides a way to model the performance of a *half DS8000* by specifying the *Failover Mode*. Consult your IBM representative for a Disk Magic analysis.
- Use separate DS8000 host adapters (HA). This option offers isolation of different workloads or different servers on the host adapter level. However, using separate host adapters on DS8800 might be less of a concern because of the high throughput of the DS8800 8 Gbps host adapters. For more guidelines, see 4.10.1, “I/O port planning considerations” on page 147.
- Place the data behind separate device adapters. This option offers isolation of different workloads on a device adapter (DA) level. However, using separate device adapters on DS8800 might be less of a concern because of the higher throughput of the DS8800 8 Gbps DAs compared to DS8700 DAs.
- Place the data on separate ranks. This option offers additional isolation for different CKD workloads on the rank and DDM level, reducing contention for the use of DDMs.

Important: z/OS (CKD) data and Open Systems (fixed block (FB)) data are always on different extent pools and therefore can never share the ranks or DDMs.

14.10 DS8000 performance monitoring tools

There are several tools available that can help with monitoring the performance of the DS8000:

- Resource Management Facility (RMF)
- OMEGAMON®:
  - OMEGAMON is an IBM Tivoli product.
  - For more information, see this link:

    http://www.ibm.com/software/tivoli/
- Tivoli Storage Productivity Center:
  - Tivoli Productivity Center is not a required performance monitoring tool for a DS8000 running in a z/OS environment.
  - In a Remote Mirror and Copy environment, you might need Tivoli Storage Productivity Center to monitor the performance of the remote disk subsystem if there is no z/OS system that accesses this subsystem.
  - More information about Tivoli Storage Productivity Center is in 7.2.1, “Tivoli Storage Productivity Center overview” on page 237.
- Products provided by other vendors that use RMF data to provide performance analysis capabilities. One example of these products is IntelliMagic Vision (formerly named RMF Magic). For more formation about this product, see http://www.intellimagic.net/.

In the following section, we explain RMF.

14.11 RMF

Resource Management Facility (RMF), which is part of the z/OS operating system, provides performance information for the DS8000 and other disk subsystems for the users. RMF can help with monitoring the following performance components:

- I/O response time
- IOP/SAP
- FICON host channel
- FICON director
- Symmetric multiprocessing (SMP)
- Cache and nonvolatile storage (NVS)
- FICON/Fibre port and host adapter
- Extent pool and rank/array

14.11.1 I/O response time

The RMF DIRECT ACCESS DEVICE ACTIVITY report (see Example 14-1) is probably the first report to use to monitor the disk subsystems performance. Also, if a service-level agreement (SLA) is not being met and the problem might be related to storage, use this report as a starting point in your performance analysis. If possible, rank volumes related to the application by I/O intensity, which is the I/O rate multiplied by Service Time (PEND + DISC + CONN time). Concentrate on the largest component of the response time. Try to identify the bottleneck that causes this problem. We provide more detailed explanations in the following sections.

The device activity report accounts for all activity to a base and all of its associated alias addresses. Activity on alias addresses is not reported separately, but the alias addresses are accumulated into the base address.

Starting with z/OS Release 1.10, this report also shows the number of cylinders allocated to the volume. Example 14-1 shows 3390-9 volumes that have either 10017 or 30051 cylinders.

Example 14-1 RMF Direct Access Device Activity (DASD) report

```
                                    D I R E C T   A C C E S S   D E V I C E   A C T I V I T Y

                                                    DEVICE   AVG  AVG   AVG  AVG   AVG  AVG  AVG    %      %     %    AVG    %     %
STORAGE  DEV  DEVICE    NUMBER  VOLUME PAV  LCU  ACTIVITY RESP IOSQ  CMR  DB    PEND DISC CONN   DEV    DEV   DEV   NUMBER ANY   MT
 GROUP   NUM  TYPE      OF CYL  SERIAL           RATE     TIME TIME  DLY  DLY   TIME TIME TIME   CONN   UTIL  RESV  ALLOC  ALLOC PEND
         A100 33903       3339  AISL00    4  007E   0.017 1.39 .000  .128 .000  .213 .836 .341   0.00   0.00   0.0   0.0   100.0  0.0
         A101 33903       3339  AISL01    4  007E   0.014 1.29 .000  .128 .000  .207 .788 .295   0.00   0.00   0.0   0.0   100.0  0.0
DBS0     A102 33909      30051  D26454    5  007E   0.198 5.39 .000  .152 .000  .252 3.21 1.92   0.01   0.02   0.0  10.0   100.0  0.0
DBS0     A103 33909      30051  D26455    5  007E  11.218 1.47 .000  .140 .000  .237 .846 .389   0.09   0.28   0.0   9.0   100.0  0.0
DBS0     A104 33909      30051  D26456    5  007E   0.990 2.73 .000  .144 .000  .242 1.89 .597   0.01   0.05   0.0   6.0   100.0  0.0
DBS0     A105 33909      30051  D26457    5  007E   0.903 12.9 .000  .140 .000  .237 12.0 .640   0.01   0.23   0.0  11.0   100.0  0.0
DBS1     A106 33909      30051  D26458    8* 007E   2.654 .803 .000  .139 .000  .238 .228 .337   0.01   0.02   0.0  18.0   100.0  0.0
PRD0     A107 33909      10017  P26531    5  007E   0.154 2.53 .000  .149 .000  .250 .492 1.78   0.01   0.01   0.1   0.0   100.0  0.0
PRD0     A108 33909      10017  P26532    5  007E   0.741 6.50 .000  .142 .000  .245 4.96 1.30   0.02   0.09   0.0  10.0   100.0  0.0
PRD0     A109 33909      10017  P26533    5  007E   0.186 3.31 .000  .136 .000  .232 2.58 .497   0.00   0.01   0.0   7.0   100.0  0.0
PRD0     A10A 33909      10017  P26534    5  007E   0.111 1.60 .000  .133 .000  .224 .914 .463   0.00   0.00   0.0   1.0   100.0  0.0
```

PAV

PAV is the number of addresses assigned to a UCB, which includes the base address plus the number of aliases assigned to that base address.

RMF reports the number of PAV addresses (or in RMF terms, *exposures*) that are used by a device. In a dynamic PAV environment, when the number of exposures changes during the reporting interval, there is an asterisk next to the PAV number. Example 14-1 shows that address A106 has a PAV of 8*. The asterisk indicates that the number of PAVs is either lower or higher than 8 during the previous RMF period.

For HyperPAV, the number of PAVs is shown in this format: *n.nH*. The *H* indicates that this volume is supported by HyperPAV. The *n.n* is a one decimal number that shows the average number of PAVs assigned to the address during the RMF report period. Example 14-2 shows that address 9505 has an average of 9.6 PAVs assigned to it during this RMF period. When a volume has no I/O activity, the PAV is always 1, which means that there is no alias assigned to this base address, because in HyperPAV, an alias is used or assigned to a base address only during the period required to execute an I/O. The alias is then released and put back into the alias pool after the I/O is completed.

Important: The number of PAVs includes the base address plus the number of aliases assigned to it. Thus, a PAV=1 means that the base address has no aliases assigned to it.

Example 14-2 RMF DASD report for HyperPAV volumes (report created on pre-z/OS 1.10)

```
                                                  D I R E C T   A C C E S S   D E V I C E   A C T I V I T Y

                                   DEVICE   AVG  AVG   AVG  AVG   AVG  AVG  AVG    %      %      %     AVG    %      %
STORAGE  DEV  DEVICE   VOLUME PAV  LCU  ACTIVITY RESP IOSQ  CMR  DB    PEND DISC CONN   DEV    DEV    DEV   NUMBER ANY    MT
 GROUP   NUM  TYPE     SERIAL           RATE     TIME TIME  DLY  DLY   TIME TIME TIME   CONN   UTIL   RESV  ALLOC  ALLOC  PEND

         9500 3390     HY9500 1.0H 0227   0.900   0.8  0.0   0.3  0.0   0.4  0.0  0.4   0.04   0.04   0.0   0.0   100.0   0.0
         9501 3390     HY9501 1.0H 0227   0.000   0.0  0.0   0.0  0.0   0.0  0.0  0.0   0.00   0.00   0.0   0.0   100.0   0.0
         9502 3390     HY9502 1.0H 0227   0.000   0.0  0.0   0.0  0.0   0.0  0.0  0.0   0.00   0.00   0.0   0.0   100.0   0.0
         9503 3390     HY9503 1.0H 0227   0.000   0.0  0.0   0.0  0.0   0.0  0.0  0.0   0.00   0.00   0.0   0.0   100.0   0.0
         9504 3390     HY9504 1.0H 0227   0.000   0.0  0.0   0.0  0.0   0.0  0.0  0.0   0.00   0.00   0.0   0.0   100.0   0.0
         9505 3390     HY9505 9.6H 0227 5747.73   1.6  0.0   0.4  0.0   0.5  0.0  1.0  60.44  60.83   0.0  10.9   100.0   0.0
```

14.11.2 I/O response time components

I/O response time consists of the following components:

- IOSQ time

  Queuing at the host/CEC
- PEND time

  Overhead
- DISC time

  Non-productive time
- CONN time

  Data transfer time

IOSQ time

IOSQ time is the time measured when an I/O request is being queued in the LPAR by z/OS.

The following situations can cause high IOSQ time:

- One of the other response time components is high.

  When you see a high IOSQ time, look at the other response time components to investigate where the problem exists.
- Sometimes, the IOSQ time is due to the unavailability of aliases to initiate an I/O request.
- There is also a slight possibility that the IOSQ is caused by a long busy condition during device error recovery.

To reduce the high IOSQ time:

- Reduce the other component of the response time. Lowering the other component's response time automatically lowers the IOSQ time.
- Lower the I/O load through data in memory or use faster storage devices.
- Provide more aliases. Using HyperPAV is the best option.

PEND time

PEND time represents the time that an I/O request waits in the hardware. This PEND time can be increased by the following conditions:

- High FICON Director port or DS8000 FICON port utilization:
  - High FICON Director port or DS8000 FICON port utilization can be caused by a high activity rate on those ports.
  - More commonly, high FICON Director port or DS8000 FICON port utilization is due to daisy chaining multiple FICON channels from different CECs to the same *port* on the FICON DIrector or the DS8000 FICON host adapter.

    In this case, the FICON channel utilization as seen from the host might be low, but the combination or sum of the utilization of these channels that share the port (either on the Director or the DS8000) can be significant.
  - For more information, see "FICON director" on page 527 and "DS8000 FICON/Fibre port and host adapter" on page 530.
- High FICON host adapter utilization. Using too many ports within a DS8000 host adapter can overload the host adapter. We suggest that only two out of the four ports in a host adapter are used.
- I/O Processor (IOP/SAP) contention at the System z host. More IOP might be needed. IOP is the processor in the CEC that is assigned to handle I/Os. For more information, see "IOP/SAP" on page 525.
- CMR Delay is a component of PEND time. See Example 14-1 on page 522. It is the initial selection time for the first I/O command in a chain for a FICON channel. It can be elongated by contention downstream from the channel, such as a busy control unit.
- Device Busy Delay is also a component of PEND time. See Example 14-1 on page 522. Device Busy Delay is caused by a domain conflict, because of a read or write operation against a domain that is in use for update. A high Device Busy Delay time can be caused by the domain of the I/O not being limited to the track that the I/O operation is accessing. If you use an Independent Software Vendor (ISV) product, ask the vendor for an updated version, which might help solve this problem.

DISC time

If the major cause of delay is the DISC time, you need to search more to find the cause. The most probable cause of high DISC time is having to wait while data is being staged from the DS8000 rank into cache, because of a read miss operation. This time can be elongated by the following conditions:

- Low read hit ratio. See "Cache and NVS" on page 528. The lower the read hit ratio, the more read operations must wait for the data to be staged from the DDMs to the cache. Adding cache to the DS8000 can increase the read hit ratio.

- High DDM utilization. You can verify high DDM utilization from the ESS Rank Statistics report. See Figure 14-17 on page 532. Look at the rank read response time. As a rule, this number must be less than 35 ms. If it is higher than 35 ms, it is an indication that this rank is too busy, because the DDMs are saturated. If the rank is too busy, consider spreading the busy volumes allocated on this rank to other ranks that are not as busy.
- *Persistent memory full condition* or nonvolatile storage (NVS) full condition can also elongate the DISC time. See “Cache and NVS” on page 528.
- In a Metro Mirror environment, a significant transmission delay between the primary and the secondary site also causes a higher DISC time.

CONN time

For each I/O operation, the channel subsystem measures the time that the DS8000, channel, and CEC are connected during the data transmission. When there is a high level of utilization of resources, significant time can be spent in contention, rather than transferring data. Several reasons exist for high CONN time:

- FICON channel saturation. If the channel or BUS utilization at the host exceeds 50%, it elongates the CONN time. See 14.11.4, “FICON host channel” on page 526. In FICON channels, the data transmitted is divided into frames. When the channel is busy with multiple I/O requests, the frames from an I/O are multiplexed with the frames from other I/Os, thus elongating the elapsed time that it takes to transfer all of the frames that belong to that I/O. The total of this time, including the transmission time of the other multiplexed frames, is counted as CONN time.
- Contention in the FICON Director, FICON port, and FICON host adapter elongate the PEND time, which also has the same effect on CONN time. See the PEND time discussion in “PEND time” on page 524.
- Rank saturation caused by high DDM utilization increases DISC time, which also increases CONN time. See the DISC time discussion in “DISC time” on page 524.

14.11.3 IOP/SAP

The IOP/SAP is the CEC processor that handles the I/O operation. We check the I/O QUEUING ACTIVITY report (Example 14-3) to determine whether the IOP is saturated. An *average queue length* greater than 1 indicates that the IOP is saturated, even though an average queue length greater than 0.5 is considered as a warning sign. A burst of I/O can also trigger a high average queue length.

If only certain IOPS are saturated, redistributing the channels assigned to the disk subsystems can help balance the load to the IOP, because an IOP is assigned to handle a certain set of channel paths. So, assigning all of the channels from one IOP to access a busy disk subsystem can cause a saturation on that particular IOP. See the appropriate hardware manual of the CEC that you use.

Example 14-3 I/O Queuing Activity report

I/O QUEUING ACTIVITY

	- INITIATIVE QUEUE -		------- IOP UTILIZATION -------			-- % I/O REQUESTS RETRIED --					-------- RETRIES / SSCH ---------				
IOP	ACTIVITY RATE	AVG Q LNGTH	% IOP BUSY	I/O START RATE	INTERRUPT RATE	ALL	CP BUSY	DP BUSY	CU BUSY	DV BUSY	ALL	CP BUSY	DP BUSY	CU BUSY	DV BUSY
00	259.349	0.12	0.84	259.339	300.523	31.1	31.1	0.0	0.0	0.0	0.45	0.45	0.00	0.00	0.00
01	127.068	0.14	100.0	126.618	130.871	50.1	50.1	0.0	0.0	0.0	1.01	1.01	0.00	0.00	0.00
02	45.967	0.10	98.33	45.967	54.555	52.0	52.0	0.0	0.0	0.0	1.08	1.08	0.00	0.00	0.00
03	262.093	1.72	0.62	262.093	279.294	32.9	32.9	0.0	0.0	0.0	0.49	0.49	0.00	0.00	0.00
SYS	694.477	0.73	49.95	694.017	765.243	37.8	37.8	0.0	0.0	0.0	0.61	0.61	0.00	0.00	0.00

14.11.4 FICON host channel

The FICON report, which is shown in Figure 14-16 on page 527, shows the FICON channel-related statistics. The *PART Utilization* is the FICON channel utilization related to the I/O activity on this LPAR, and the *Total Utilization* is the total utilization of the FICON channel from all LPARs that are defined on the CEC. Consider this rule for FICON analysis:

- For the FICON Express CHPID utilization, the suggested maximum utilization level is 50%.
- For the FICON BUS busy utilization, the suggested maximum utilization level is 40%.
- For the FICON Express Link utilization with an estimated throughput of 2, 4, or 8 Gbps, the recommended maximum utilization threshold is 70%.

If these numbers exceed the threshold, you observe an elongated CONN time.

For small block transfers, the BUS utilization is less than the FICON channel utilization. For large block transfers, the BUS utilization is greater than the FICON channel utilization.

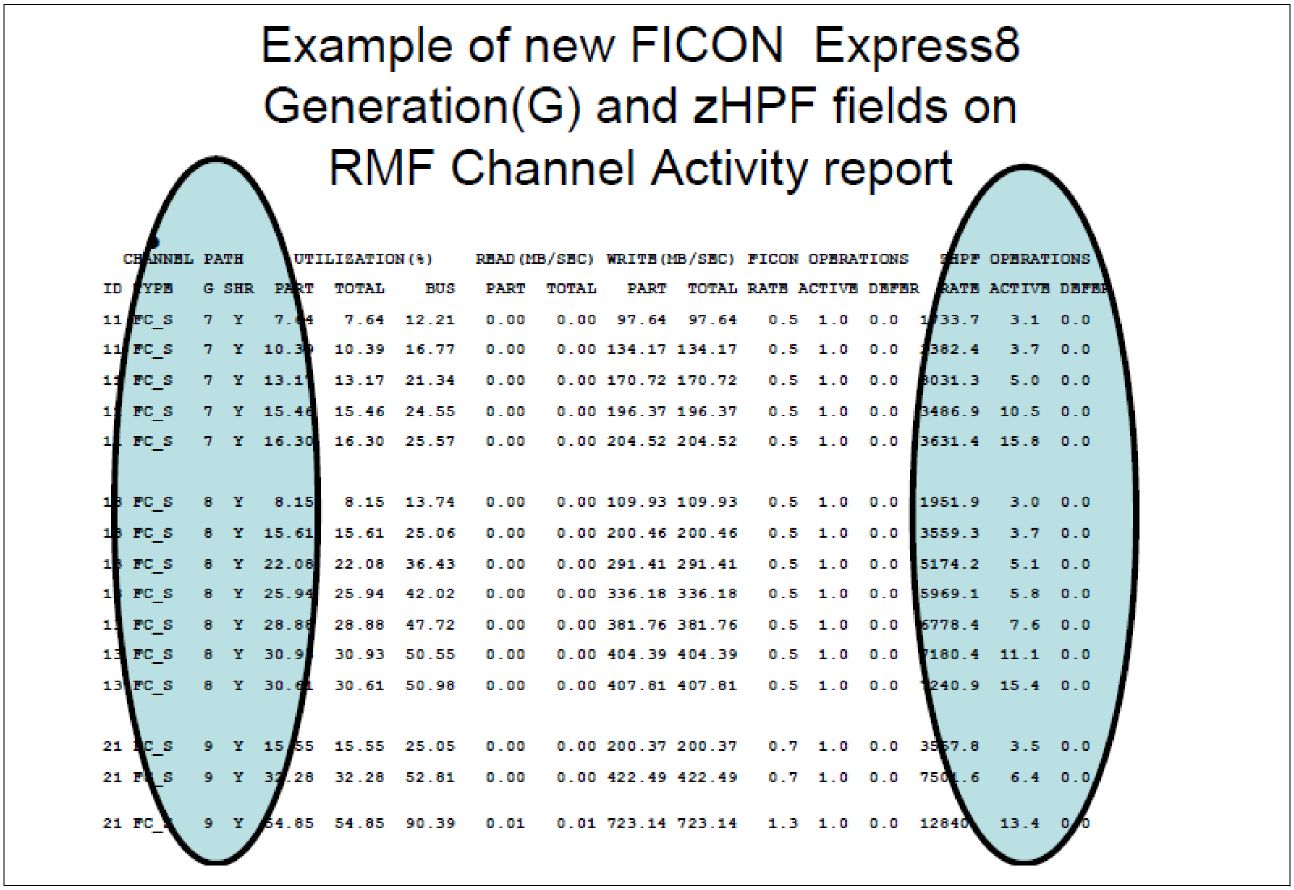
The Generation (G) field in the channel report shows the combination of the generation FICON channel that is being used and the speed of the FICON channel link for this CHPID at the time of the machine IPL. The G field does not include any information about the link between the director and the DS8000. See Table 14-1.

Table 14-1 G field in Channel Activity Report

Generation Field number (G)	Description
1	Link between the channel and the director is operating at 1 Gbps, which is applicable to a FICON Express channel
2	Link between the channel and the director is operating at 2 Gbps, which is applicable to a FICON Express channel
3	Link between the channel and the director is auto negotiating to 1 Gbps, which is applicable to a FICON Express2 or FICON Express4 channel
4	Link between the channel and the director is auto negotiating to 2 Gbps, which is applicable to a FICON Express2 or FICON Express4 channel
5	Link between the channel and the director is operating at 4 Gbps, which is applicable to a FICON Express4 channel
7	Link between the channel and the director is auto negotiating to 2 Gbps, which is applicable to a FICON Express8 channel
8	Link between the channel and the director is auto negotiating to 4 Gbps, which is applicable to a FICON Express8 channel
9	Link between the channel and the director is operating at 8 Gbps, which is applicable to a FICON Express8 channel
11	Link between the channel and the director is operating at 2 Gbps, which is applicable to a FICON Express8S channel
12	Link between the channel and the director is operating at 4 Gbps, which is applicable to a FICON Express8S channel
13	Link between the channel and the director is operating at 8 Gbps, which is applicable to a FICON Express8S channel

The link between the director and the DS8000 can run at 1, 2, 4, or 8 Gbps.

If the channel is point-to-point connected to the DS8000 FICON port, the G field indicates the speed that was negotiated between the FICON channel and the DS8000 port.



CHANNEL PATH				UTILIZATION(%)			READ(MB/SEC)		WRITE(MB/SEC)		FICON OPERATIONS			ZHPF OPERATIONS		
ID	TYPE	G	SHR	PART	TOTAL	BUS	PART	TOTAL	PART	TOTAL	RATE	ACTIVE	DEFER	RATE	ACTIVE	DEFER
11	FC_S	7	Y	[illegible]	7.64	12.21	0.00	0.00	97.64	97.64	0.5	1.0	0.0	[illegible]	3.1	0.0
11	FC_S	7	Y	[illegible]	10.39	16.77	0.00	0.00	134.17	134.17	0.5	1.0	0.0	[illegible]	3.7	0.0
[illegible]	FC_S	7	Y	[illegible]	13.17	21.34	0.00	0.00	170.72	170.72	0.5	1.0	0.0	[illegible]	5.0	0.0
[illegible]	FC_S	7	Y	15.46	15.46	24.55	0.00	0.00	196.37	196.37	0.5	1.0	0.0	3486.9	10.5	0.0
[illegible]	FC_S	7	Y	16.30	16.30	25.57	0.00	0.00	204.52	204.52	0.5	1.0	0.0	3631.4	15.8	0.0
[illegible]	FC_S	8	Y	8.15	8.15	13.74	0.00	0.00	109.93	109.93	0.5	1.0	0.0	1951.9	3.0	0.0
[illegible]	FC_S	8	Y	15.61	15.61	25.06	0.00	0.00	200.46	200.46	0.5	1.0	0.0	3559.3	3.7	0.0
[illegible]	FC_S	8	Y	22.08	22.08	36.43	0.00	0.00	291.41	291.41	0.5	1.0	0.0	5174.2	5.1	0.0
[illegible]	FC_S	8	Y	25.94	25.94	42.02	0.00	0.00	336.18	336.18	0.5	1.0	0.0	5969.1	5.8	0.0
[illegible]	FC_S	8	Y	[illegible]	28.88	47.72	0.00	0.00	381.76	381.76	0.5	1.0	0.0	[illegible]	7.6	0.0
13	FC_S	8	Y	[illegible]	30.93	50.55	0.00	0.00	404.39	404.39	0.5	1.0	0.0	[illegible]	11.1	0.0
13	FC_S	8	Y	[illegible]	30.61	50.98	0.00	0.00	407.81	407.81	0.5	1.0	0.0	[illegible]	15.4	0.0
21	[illegible]	9	Y	[illegible]	15.55	25.05	0.00	0.00	200.37	200.37	0.7	1.0	0.0	[illegible]	3.5	0.0
21	[illegible]	9	Y	[illegible]	32.28	52.81	0.00	0.00	422.49	422.49	0.7	1.0	0.0	[illegible]	6.4	0.0
21	[illegible]	9	Y	[illegible]	54.85	90.39	0.01	0.01	723.14	723.14	1.3	1.0	0.0	[illegible]	13.4	[illegible]

Figure 14-16 Channel Path Activity report

14.11.5 FICON director

The *FICON director* is the switch that is used to connect the host FICON channel to the DS8000 FICON port. FICON director performance statistics are collected in the System Management Facilities (SMF) record type 74 subtype 7. The FICON DIRECTOR ACTIVITY report (Example 14-4 on page 528) provides information about director and port activities. This report assists you in analyzing performance problems and in capacity planning.

The measurements provided for a port in this report include the I/O for the system on which the report is taken *and also* include all I/Os that are directed through this port, regardless of which LPAR requests the I/O.

The CONNECTION in this report shows where this port is connected:

- CHP: The port is connected to a FICON channel on the host.
- CHP-H: The port is connected to a FICON channel on the host that requested this report.
- CU: This port is connected to a port on a disk subsystem.
- SWITCH: This port is connected to another FICON director.

The important performance metric is *AVG FRAME PACING*. This metric shows the average time (in microseconds) that a frame waited before it was transmitted. The higher the contention on the director port, the higher the average frame pacing metric.

Example 14-4 FICON Director Activity report

```
                             F I C O N   D I R E C T O R   A C T I V I T Y

SWITCH DEVICE:0414   SWITCH ID:01   TYPE:005000   MODEL:001   MAN:MCD   PLANT:01   SERIAL:00000MK00109

PORT     -CONNECTION-     AVG FRAME       AVG FRAME SIZE            PORT BANDWIDTH (MB/SEC)          ERROR
ADDR     UNIT      ID     PACING          READ     WRITE            --READ     ----WRITE --          COUNT
05       CHP       FA          0          808        285             50.04            10.50              0
07       CHP       4A          0          149        964             20.55             5.01              0
09       CHP       FC          0          558       1424             50.07            10.53              0
0B       CHP-H     F4          0          872        896             50.00            10.56              0
12       CHP       D5          0           73        574             20.51             5.07              0
13       CHP       C8          0          868       1134             70.52             2.08              1
14       SWITCH  ----          0          962        287             50.03            10.59              0
15       CU      C800          0         1188        731             20.54             5.00              0
```

14.11.6 Processor complex

There is no SMF record that provides reports on the DS8000 processor complex utilization. A saturation of the processor complex usually occurs when running at a high I/O rate or running a heavy write remote copy workload. The general indications of a saturation are high PEND and CONN times that are not related to high channel utilization or high port utilization, either for the FICON director port or the DS8000 FICON port.

Sometimes, you see a saturation when you run a benchmark to test a new disk subsystem. Usually, in a benchmark, you try to run at the highest possible I/O rate on the disk subsystem.

14.11.7 Cache and NVS

The RMF CACHE SUBSYSTEM ACTIVITY report provides useful information for analyzing the reason for high DISC time. Example 14-5 on page 529 shows a sample cache report by LCU. Example 14-6 on page 529 is the continuation of this report, and it shows the cache statistics by volume.

The report shows the I/O requests by read and by write. It shows the rate, the hit rate, and the hit ratio of the read and the write activities. The read-to-write ratio is also calculated. The total I/O requests can be higher than the I/O rate shown in the DASD report. In the DASD report, one channel program is counted as one I/O. However, in the cache report, if there are multiple Locate Record commands in a channel program, each Locate Record command is counted as one I/O request.

In this report, we can check to see the value of the read hit ratio. Low read hit ratios contribute to higher DISC time. For a cache friendly workload, we see a read hit ratio of better than 90%. The write hit ratio is usually 100%.

High DFW BYPASS is an indication that persistent memory or nonvolatile storage (NVS) is overcommitted. DFW BYPASS means DASD Fast Write I/Os that are retried, because persistent memory is full. Calculate the quotient of DFW BYPASS divided by the total I/O rate. As a rule, if this number is higher than 1%, the write retry operations significantly affect the DISC time.

Check the *DISK ACTIVITY* part of the report. The Read response time must be less than 35 ms. If it is higher than 35 ms, it is an indication that the DDMs on the rank where this LCU resides are saturated.

Example 14-5 Cache Subsystem Activity summary

```
                                              C A C H E   S U B S Y S T E M   A C T I V I T Y

------------------------------------------------------------------------------------------------------------------------------------
                                                           CACHE SUBSYSTEM STATUS
------------------------------------------------------------------------------------------------------------------------------------
SUBSYSTEM STORAGE            NON-VOLATILE STORAGE        STATUS
CONFIGURED       31104M      CONFIGURED      1024.0M     CACHING                    - ACTIVE
AVAILABLE        26290M      PINNED              0.0     NON-VOLATILE STORAGE       - ACTIVE
PINNED              0.0                                  CACHE FAST WRITE           - ACTIVE
OFFLINE             0.0                                  IML DEVICE AVAILABLE       - YES
------------------------------------------------------------------------------------------------------------------------------------
                                                          CACHE SUBSYSTEM OVERVIEW
------------------------------------------------------------------------------------------------------------------------------------
TOTAL I/O     19976   CACHE I/O     19976   CACHE OFFLINE        0
TOTAL H/R     0.804   CACHE H/R     0.804
CACHE I/O     -------------READ I/O REQUESTS-------------    ----------------------WRITE I/O REQUESTS----------------------        %
REQUESTS        COUNT      RATE      HITS      RATE     H/R      COUNT      RATE      FAST      RATE      HITS      RATE     H/R      READ
NORMAL          14903     252.6     10984     186.2   0.737       5021      85.1      5021      85.1      5021      85.1   1.000      74.8
SEQUENTIAL          0       0.0         0       0.0     N/A         52       0.9        52       0.9        52       0.9   1.000       0.0
CFW DATA            0       0.0         0       0.0     N/A          0       0.0         0       0.0         0       0.0     N/A       N/A
TOTAL           14903     252.6     10984     186.2   0.737       5073      86.0      5073      86.0      5073      86.0   1.000      74.6
-----------------------CACHE MISSES-----------------------            ------------MISC------------          ------NON-CACHE I/O-----
REQUESTS       READ     RATE    WRITE     RATE   TRACKS     RATE                        COUNT    RATE                      COUNT   RATE
                                                                     DFW BYPASS             0     0.0         ICL              0    0.0
NORMAL         3919     66.4        0      0.0     3921     66.5     CFW BYPASS             0     0.0         BYPASS           0    0.0
SEQUENTIAL        0      0.0        0      0.0        0      0.0     DFW INHIBIT            0     0.0         TOTAL            0    0.0
CFW DATA          0      0.0        0      0.0                       ASYNC (TRKS)        3947    66.9
TOTAL          3919     RATE     66.4
---CKD STATISTICS---     ---RECORD CACHING---                        ----HOST ADAPTER ACTIVITY---      --------DISK ACTIVITY-------
                                                                                    BYTES   BYTES                 RESP   BYTES  BYTES
WRITE                0     READ MISSES        0                                      /REQ    /SEC                 TIME    /REQ   /SEC
WRITE HITS           0     WRITE PROM      3456                      READ            6.1K    1.5M      READ      6.772   53.8K   3.6M
                                                                     WRITE           5.7K  491.0K      WRITE    12.990    6.8K 455.4K
```

Following the report in Example 14-5 is the CACHE SUBSYSTEM ACTIVITY report by volume serial number, as shown in Example 14-6. You can see to which extent pool each volume belongs. In the case where we have the following setup, it is easier to perform the analysis if a performance problem happens on the LCU:

- One extent pool has one rank.
- All volumes on an LCU belong to the same extent pool.

If we look at the rank statistics in the report in Example 14-9 on page 532, we know that all the I/O activity on that rank comes from the same LCU. So, we can concentrate our analysis on the volumes on that LCU only.

> **Important:** Depending on the DDM size used and the 3390 model selected, you can put multiple LCUs on one rank, or you can also have an LCU that spans more than one rank.

Example 14-6 Cache Subsystem Activity by volume serial number

```
                                              C A C H E   S U B S Y S T E M   A C T I V I T Y

------------------------------------------------------------------------------------------------------------------------------------
                                                       CACHE SUBSYSTEM DEVICE OVERVIEW
------------------------------------------------------------------------------------------------------------------------------------
VOLUME      DEV  XTNT      %     I/O   ---CACHE HIT RATE--    ----------DASD I/O RATE----------    ASYNC   TOTAL    READ   WRITE       %
SERIAL      NUM  POOL    I/O    RATE    READ     DFW    CFW   STAGE  DFWBP     ICL    BYP  OTHER    RATE     H/R     H/R     H/R    READ
*ALL                   100.0   338.6   186.2    86.0    0.0    66.4    0.0     0.0    0.0    0.0    66.9   0.804   0.737   1.000    74.6
*CACHE-OFF               0.0     0.0
*CACHE                 100.0   338.6   186.2    86.0    0.0    66.4    0.0     0.0    0.0    0.0    66.9   0.804   0.737   1.000    74.6
PR7000     7000  0000   22.3    75.5    42.8    19.2    0.0    13.5    0.0     0.0    0.0    0.0    14.4   0.821   0.760   1.000    74.6
PR7001     7001  0000   11.5    38.8    20.9    10.5    0.0     7.5    0.0     0.0    0.0    0.0     7.6   0.807   0.736   1.000    73.1
PR7002     7002  0000   11.1    37.5    20.4     9.5    0.0     7.6    0.0     0.0    0.0    0.0     7.0   0.797   0.729   1.000    74.7
PR7003     7003  0000   11.3    38.3    22.0     8.9    0.0     7.4    0.0     0.0    0.0    0.0     6.8   0.806   0.747   1.000    76.8
PR7004     7004  0000    3.6    12.0     6.8     3.0    0.0     2.3    0.0     0.0    0.0    0.0     2.6   0.810   0.747   1.000    75.2
PR7005     7005  0000    3.7    12.4     6.8     3.2    0.0     2.4    0.0     0.0    0.0    0.0     2.7   0.808   0.741   1.000    74.1
PR7006     7006  0000    3.8    12.8     6.5     3.6    0.0     2.6    0.0     0.0    0.0    0.0     3.1   0.796   0.714   1.000    71.5
PR7007     7007  0000    3.6    12.3     6.9     3.1    0.0     2.4    0.0     0.0    0.0    0.0     2.5   0.806   0.742   1.000    75.2
PR7008     7008  0000    3.6    12.2     6.7     3.4    0.0     2.2    0.0     0.0    0.0    0.0     2.7   0.821   0.753   1.000    72.5
```

```
PR7009    7009  0000   3.6   12.2     6.8    2.9    0.0     2.5    0.0    0.0    0.0    0.0      2.3  0.796   0.732   1.000    76.4
```

If you specify REPORTS(CACHE(DEVICE)) when running the cache report, you get the detailed report by volume as in Example 14-7. This report gives you the detailed cache statistics of each volume. By specifying REPORTS(CACHE(SSID(*nnnn*))), you can limit this report to only certain LCUs.

The report shows the same performance statistics as in Example 14-5 on page 529, but at the level of each volume.

Example 14-7 Cache Device Activity report detail by volume

```
                                             C A C H E   D E V I C E   A C T I V I T Y

VOLSER     @9C02F      NUM    C02F     EXTENT POOL   0000
-------------------------------------------------------------------------------------------------------------------------
                                                        CACHE DEVICE STATUS
-------------------------------------------------------------------------------------------------------------------------
CACHE STATUS                                      DUPLEX PAIR STATUS
CACHING             - ACTIVE                      DUPLEX PAIR        - NOT ESTABLISHED
DASD FAST WRITE     - ACTIVE                      STATUS             - N/A
PINNED DATA         - NONE                        DUAL COPY VOLUME   - N/A
-------------------------------------------------------------------------------------------------------------------------
                                                       CACHE DEVICE ACTIVITY
-------------------------------------------------------------------------------------------------------------------------
TOTAL I/O     3115   CACHE I/O     3115   CACHE OFFLINE   N/A
TOTAL H/R    0.901   CACHE H/R    0.901
CACHE I/O    -------------READ I/O REQUESTS-------------    ----------------------WRITE I/O REQUESTS----------------------        %
REQUESTS       COUNT      RATE       HITS      RATE     H/R      COUNT      RATE      FAST      RATE      HITS      RATE     H/R      READ
NORMAL          2786      46.4       2477      41.3   0.889        329       5.5       329       5.5       329       5.5   1.000      89.4
SEQUENTIAL         0       0.0          0       0.0     N/A          0       0.0         0       0.0         0       0.0     N/A       N/A
CFW DATA           0       0.0          0       0.0     N/A          0       0.0         0       0.0         0       0.0     N/A       N/A
TOTAL           2786      46.4       2477      41.3   0.889        329       5.5       329       5.5       329       5.5   1.000      89.4
-----------------------CACHE MISSES-----------------------            ------------MISC------------          ------NON-CACHE I/O-----
REQUESTS      READ    RATE   WRITE    RATE  TRACKS    RATE                            COUNT   RATE                        COUNT   RATE
                                                                      DFW BYPASS          0    0.0          ICL               0    0.0
NORMAL         309     5.1       0     0.0     311     5.2            CFW BYPASS          0    0.0          BYPASS            0    0.0
SEQUENTIAL       0     0.0       0     0.0       0     0.0            DFW INHIBIT         0    0.0          TOTAL             0    0.0
CFW DATA         0     0.0       0     0.0                            ASYNC (TRKS)      173    2.9
TOTAL          309    RATE     5.1
---CKD STATISTICS---      ---RECORD CACHING---                        ----HOST ADAPTER ACTIVITY---      --------DISK ACTIVITY-------
                                                                                      BYTES  BYTES                  RESP   BYTES  BYTES
WRITE             0       READ MISSES        0                                         /REQ   /SEC                  TIME    /REQ   /SEC
WRITE HITS        0       WRITE PROM       111                        READ             4.1K 190.1K      READ   14.302   55.6K 288.4K
                                                                      WRITE            4.0K  21.8K      WRITE  43.472   18.1K  48.1K
```

14.11.8 DS8000 FICON/Fibre port and host adapter

The report in Example 14-8 on page 531 shows the port report on the DS8000, which includes the FICON ports and also the PPRC link ports.

The SAID is the port ID:

- The first two characters denote the enclosure number (see Figure 14-17 on page 532).
- The third character denotes the host adapter number within the enclosure:

  Numbered 0, 1, 3, and 4
- The last character denotes the port ID within that host adapter:

  Numbered 0, 1, 2, up to 7

The report shows that the ports are running at 2 Gbps. There are FICON ports, shown under the heading of *LINK TYPE* as *ECKD READ* and *ECKD WRITE*. There are also PPRC ports, shown as *PPRC SEND* and *PPRC RECEIVE*.

The *I/O INTENSITY* is the result of multiplication of the operations per second and the response time per operation. For FICON ports, it is calculated for both the read and write operations, while for PPRC ports, it is calculated for both the send and receive operations. The total I/O intensity is the sum of those two numbers on each port.

For FICON ports, if the total I/O intensity reaches 4000, the response time is affected, most probably, the PEND and CONN times. When this number already approaches 2000, proactive actions might be needed to prevent a further increase in the total I/O intensity. See the discussion on PEND and CONN times in "PEND time" on page 524 and "CONN time" on page 525. This rule does not apply for PPRC ports, especially if the distance between the primary site and the secondary site is significant.

If the DS8000 is shared between System z and Open Systems, the report in Example 14-8 also shows the port activity used by the Open Systems. It shows up as *SCSI READ* and *SCSI WRITE* on ports 0200 and 0201.

Example 14-8 DS8000 link statistics

```
                                                E S S   L I N K   S T A T I S T I C S

------ADAPTER------   --LINK TYPE--      BYTES      BYTES          OPERATIONS      RESP TIME        I/O
SAID  TYPE                               /SEC       /OPERATION     /SEC            /OPERATION       INTENSITY
0000  FIBRE 2Gb       ECKD READ           17.2M       9.9K         1735.2              0.1            131.6
                      ECKD WRITE           7.7M      14.5K          533.9              0.2            123.4
                                                                                                     ------
                                                                                                      255.0
0001  FIBRE 2Gb       ECKD READ            9.1M       8.4K         1087.2              0.1             79.9
                      ECKD WRITE           7.7M      17.0K          455.9              0.2            101.4
                                                                                                     ------
                                                                                                      181.2
0101  FIBRE 2Gb       PPRC SEND            6.0M      53.1K          112.2              9.1           1024.9
                      PPRC RECEIVE       0.0          0.0             0.0              0.0              0.0
                                                                                                     ------
                                                                                                     1024.9
0102  FIBRE 2Gb       PPRC SEND            6.2M      53.1K          115.9              8.6            998.0
                      PPRC RECEIVE       0.0          0.0             0.0              0.0              0.0
                                                                                                     ------
                                                                                                      998.0
0200  FIBRE 2Gb       SCSI READ           10.8M      30.7K          352.4              0.2             67.5
                      SCSI WRITE           1.9M      31.5K           60.9              1.4             83.3
                                                                                                     ------
                                                                                                      150.8
0201  FIBRE 2Gb       SCSI READ            9.0M      38.7K          232.0              0.2             53.3
                      SCSI WRITE         135.0K      10.7K           12.6              0.3              3.5
                                                                                                     ------
                                                                                                       56.8
```

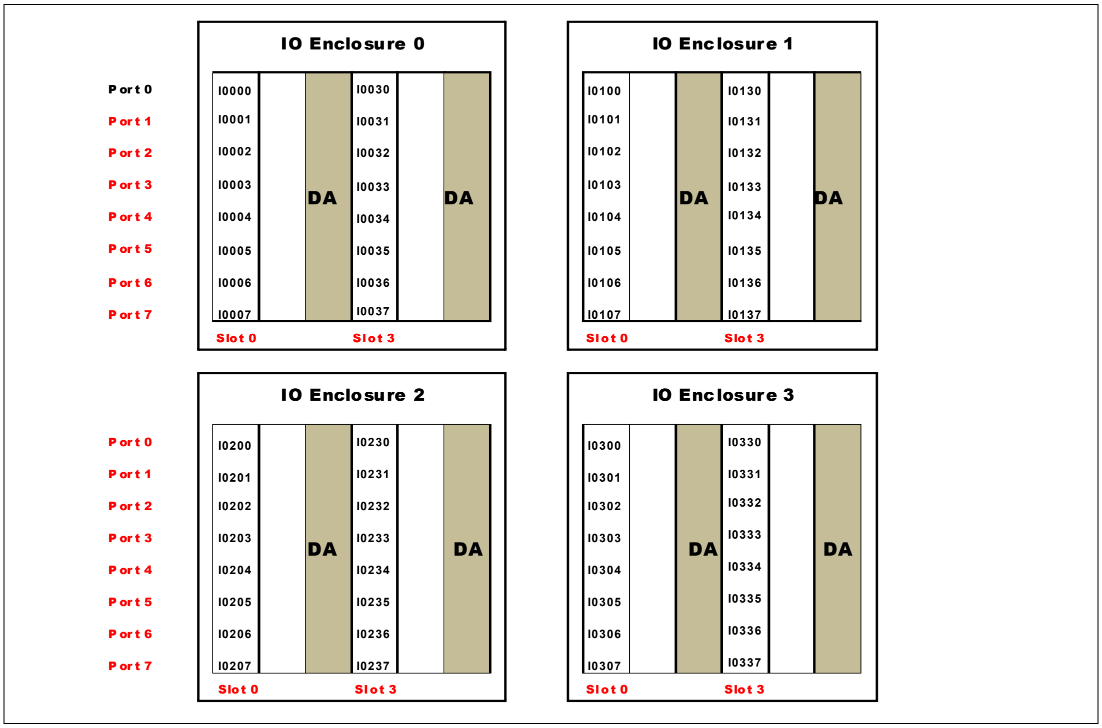


Figure 14-17 DS8800 port numbering

14.11.9 DS8000 extent pool and rank

Example 14-9 shows extent pool 0000, which contains RRID 0000, 0002, 0008, 000A, 0010, 0012, 0018, and 001A. This report shows a balanced load on these ranks. All of the performance metrics, such as *OPS/SEC*, *BYTES/OP*, *BYTES/SEC*, and *RTIME/OP*, are balanced across all ranks that reside on this extent pool for the read and also the write operations.

Example 14-9 Rank statistics for multiple ranks on one extent pool that uses storage pool striping

```
                                           E S S   R A N K   S T A T I S T I C S

------ READ OPERATIONS -------     ------ WRITE OPERATIONS ------
--EXTENT POOL--          OPS    BYTES   BYTES   RTIME       OPS    BYTES   BYTES    RTIME    --ARRAY--  MIN    RANK  RAID
 ID   TYPE       RRID   /SEC      /OP    /SEC     /OP      /SEC      /OP    /SEC      /OP    NUM  WDTH  RPM     CAP  TYPE
0000  CKD 1Gb    0000   40.9    49.1K    2.0M     6.4      11.3    13.1K  148.5K      7.1      1     6   15    438G  RAID 5
                 0002   39.2    47.6K    1.9M     6.8      11.9    15.8K  187.9K      7.5      1     6   15    438G  RAID 5
                 0008   40.9    48.2K    2.0M     6.4      10.9    16.5K  179.1K      7.2      1     6   15    438G  RAID 5
                 000A   37.7    48.4K    1.8M     6.5      10.3    13.8K  142.0K      7.2      1     6   15    438G  RAID 5
                 0010   40.9    48.9K    2.0M     6.2      11.4    14.4K  163.8K      7.3      1     6   15    438G  RAID 5
                 0012   40.0    47.8K    1.9M     6.2      10.5    12.3K  128.9K      7.1      1     6   15    438G  RAID 5
                 0018   42.1    47.8K    2.0M     6.6      12.1    12.4K  150.7K      7.2      1     6   15    438G  RAID 5
                 001A   45.9    48.6K    2.2M     6.7      12.7    15.1K  192.2K      7.3      1     6   15    438G  RAID 5
                 POOL  327.8    48.3K   15.8M     6.5      91.1    14.2K    1.3M      7.2      8    48   15   3504G  RAID 5
0001  CKD 1Gb    0004   40.8    47.8K    1.9M     6.7      11.2    14.0K  157.3K      9.2      1     7   15    511G  RAID 5
                 0006   42.0    48.8K    2.0M     6.9      11.9    16.1K  192.2K      9.2      1     7   15    511G  RAID 5
                 000C   38.4    49.4K    1.9M     6.5       9.5    16.5K  157.3K      8.8      1     7   15    511G  RAID 5
                 000E   43.0    49.0K    2.1M     6.6      12.1    16.6K  201.0K      8.5      1     7   15    511G  RAID 5
                 0014   40.4    49.2K    2.0M     6.6      10.8    15.2K  163.8K      8.4      1     7   15    511G  RAID 5
                 0016   40.5    48.8K    2.0M     6.5      11.5    16.0K  183.5K      8.5      1     7   15    511G  RAID 5
                 001C   37.9    47.0K    1.8M     6.2      10.5    14.5K  152.9K      8.2      1     7   15    511G  RAID 5
                 001E   39.6    48.6K    1.9M     7.2      10.9    16.5K  179.1K     10.0      1     7   15    511G  RAID 5
                 POOL  322.7    48.6K   15.7M     6.7      88.4    15.7K    1.4M      8.9      8    56   15   4088G  RAID 5
```

15

IBM System Storage SAN Volume Controller attachment

This chapter describes the guidelines and procedures to make the most of the performance available from your DS8000 storage subsystem when attached to the IBM SAN Volume Controller.

We describe the following topics:

- IBM System Storage SAN Volume Controller
- SAN Volume Controller performance considerations
- DS8000 performance considerations with SAN Volume Controller
- Performance monitoring
- TPC Reporter for Disk, and Storage Tiering Reports for SAN Volume Controller
- Advanced functions for the DS8000
- Configuration guidelines for optimizing performance
- Where to place the SSDs
- Where to place Easy Tier

15.1 IBM System Storage SAN Volume Controller

The IBM System Storage SAN Volume Controller is designed to increase the flexibility of your storage infrastructure by introducing an in-band virtualization layer between the servers and the storage systems. The SAN Volume Controller can enable a tiered storage environment to increase flexibility in storage management. The SAN Volume Controller combines the capacity from multiple disk storage systems into a single storage pool, which can be managed from a central point, which is simpler to manage and helps increase disk capacity utilization. With the SAN Volume Controller, you can apply SAN Volume Controller advanced Copy Services across storage systems from many vendors to help you simplify operations.

For more information about SAN Volume Controller, see *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933.

15.1.1 SAN Volume Controller concepts

The SAN Volume Controller is a storage area network (SAN) appliance that attaches storage devices to supported Open Systems servers. The SAN Volume Controller provides symmetric virtualization by creating a pool of managed disks from the attached storage subsystems, which are then mapped to a set of virtual disks for use by various attached host computer systems. System administrators can view and access a common pool of storage on the SAN, which enables them to use storage resources more efficiently and provides a common base for advanced functions.

The SAN Volume Controller solution is designed to reduce both the complexity and costs of managing your SAN-based storage. With the SAN Volume Controller, you can perform these tasks:

- Simplify management and increase administrator productivity by consolidating storage management intelligence from disparate storage controllers into a single view.
- Improve application availability by enabling data migration between disparate disk storage devices non-disruptively.
- Improve disaster recovery and business continuance needs by applying and managing Copy Services across disparate disk storage devices within the SAN.
- Provide advanced features and functions to the entire SAN:
  - Large scalable cache
  - Advanced Copy Services
  - Space management
  - Mapping based on desired performance characteristics
  - Quality of Service (QoS) metering and reporting

SAN Volume Controller clustering

The SAN Volume Controller is a collection of up to eight cluster nodes added in pairs. These eight nodes are managed as a set (cluster) and present a single point of control to the administrator for configuration and service activity.

For I/O purposes, SAN Volume Controller nodes within the cluster are grouped into pairs (called *I/O Groups*). A single pair is responsible for serving I/O on a specific virtual disk (VDisk) volume. One node within the I/O Group represents the preferred path for I/O to a specific VDisk volume, and the other node represents the non-preferred path. This preference alternates between nodes as each volume is created within an I/O Group to balance the workload evenly between the two nodes.

> **Preferred node:** The preferred node by no means signifies absolute ownership. The data is still accessed by the partner node in the I/O Group in the event of a failure or if the preferred node workload becomes too high.

Beyond automatic configuration and cluster administration, the data transmitted from attached application servers is also treated in the most reliable manner. When data is written by the server, the preferred node within the I/O Group stores the write data in its own write cache and the write cache of its partner (non-preferred) node before sending an *I/O complete* status back to the server application. To ensure that data is written in the event of a node failure, the surviving node empties its write cache and proceeds in write-through mode until the cluster is returned to a fully operational state.

> **Write-through mode:** *Write-through mode* is where the data is not cached in the nodes but is written directly to the disk system instead. While operating in this mode, performance is slightly degraded.

Furthermore, each node in the I/O Group is protected by its own dedicated uninterruptible power supply.

SAN Volume Controller virtualization

The SAN Volume Controller provides block aggregation into volumes and volume management for disk storage within the SAN. In simpler terms, the SAN Volume Controller manages a number of back-end storage controllers and maps the physical storage within those controllers to logical disk images that can be seen by application servers and workstations in the SAN.

The SAN must be zoned in such a way that the application servers cannot see the back-end storage, preventing the SAN Volume Controller and the application servers from both trying to manage the back-end storage. In the SAN fabric, distinct zones are defined:

- In the server zone, the server systems can identify and address the nodes. You can have more than one server zone. Generally, you create one server zone per server attachment.
- In the disk zone, the nodes can identify the disk storage subsystems. Generally, you create one zone for each distinct storage subsystem.
- In the SAN Volume Controller zone, all SAN Volume Controller node ports are permitted to communicate for cluster management.
- Where remote Copy Services are used, an inter-cluster zone must be created.

The SAN Volume Controller I/O Groups are connected to the SAN in such a way that all back-end storage and all application servers are visible to all of the I/O Groups. The SAN Volume Controller I/O Groups see the storage presented to the SAN by the back-end controllers as a number of disks, known as *Managed Disks (MDisks)*. Because the SAN Volume Controller does not attempt to provide recovery from physical disk failures within the back-end controllers, MDisks are usually, but not necessarily, part of a RAID array.

MDisks are collected into one or several groups, known as *Managed Disk Groups* (MDGs), or *storage pools*. When an MDisk is assigned to a storage pool, the MDisk is divided into a number of extents (extent minimum size is 16 MiB and extent maximum size is 8 GiB). The extents are numbered sequentially from the start to the end of each MDisk.

Using MDisks only: For performance considerations, we suggest that for a single-tiered pool, you create storage pools that use only MDisks, which have the same characteristics in performance or reliability. For a multi-tiered storage pool, this consideration applies to the respective tier.

The storage pool provides the capacity in the form of extents, which are used to create volumes, also known as *Virtual Disks* (VDisks).

When creating SAN Volume Controller volumes or VDisks, the default option of striped allocation is normally the best choice. This option helps to balance I/Os across all the managed disks in a storage pool, which optimizes overall performance and helps to reduce hot spots. Conceptually, this method is represented in Figure 15-1.

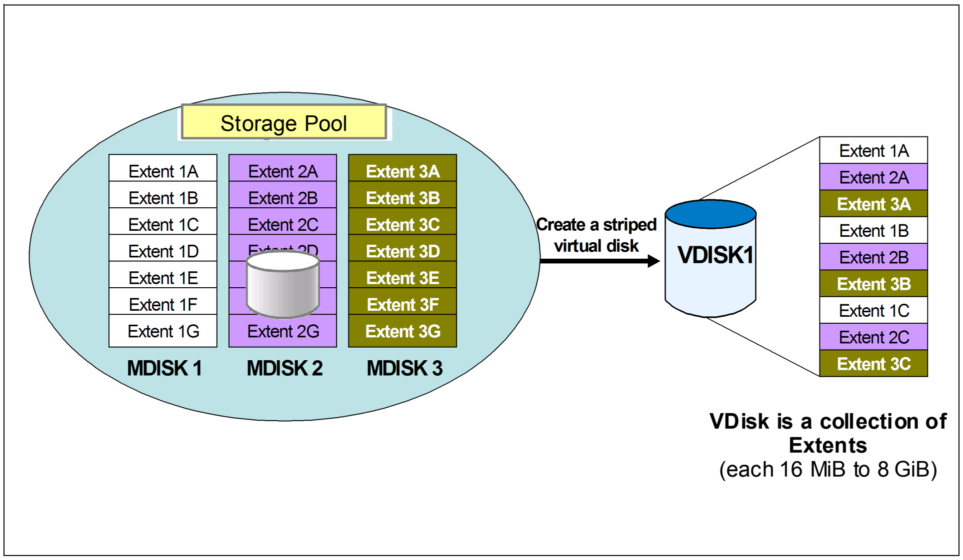


Figure 15-1 Extents being used to create Virtual Disks

The virtualization function in the SAN Volume Controller maps the volumes seen by the application servers onto the MDisks provided by the back-end controllers. I/O traffic for a particular volume is, at any one time, handled exclusively by the nodes in a single I/O Group. Thus, although a cluster can have many nodes within it, the nodes handle I/O in independent pairs, which means that the I/O capability of the SAN Volume Controller scales well (almost linearly), because additional throughput can be obtained by adding additional I/O Groups.

Figure 15-2 on page 537 summarizes the various relationships that bridge the physical disks through to the virtual disks within the SAN Volume Controller architecture.

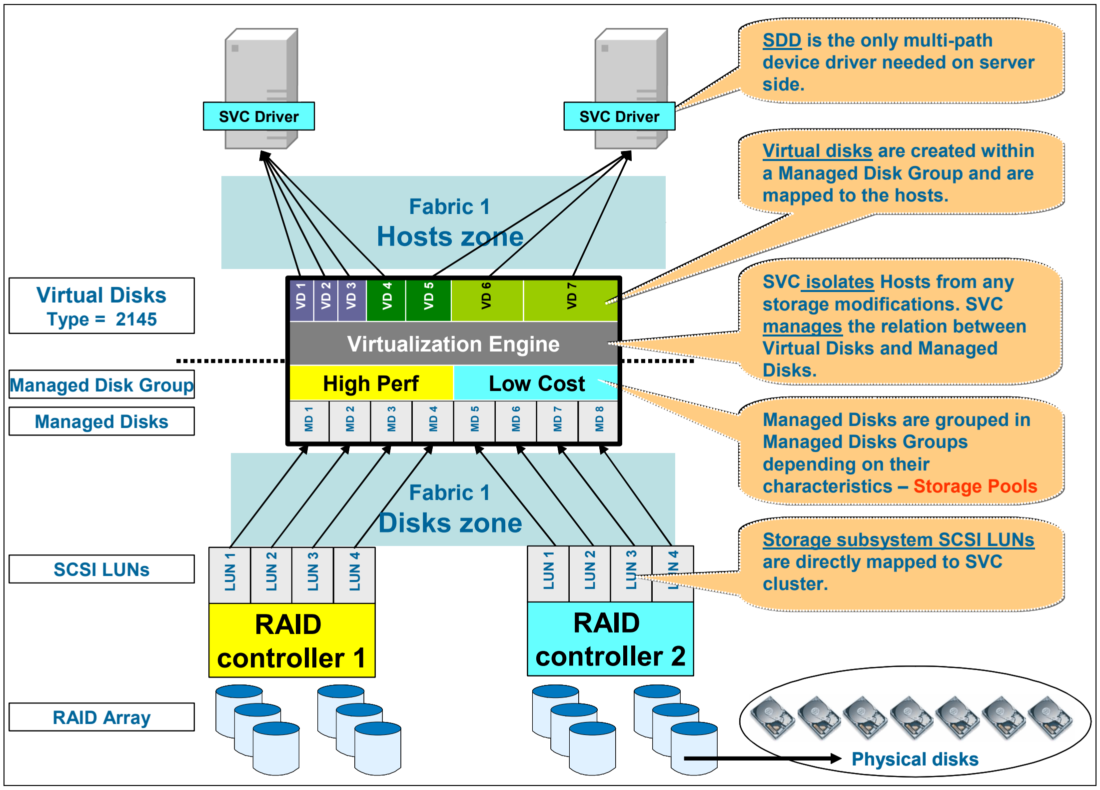


Figure 15-2 Relationship between physical and virtual disks

15.1.2 SAN Volume Controller multipathing

Each SAN Volume Controller node presents a VDisk to the SAN via multiple paths. We suggest that a VDisk can be seen in the SAN by *four* paths. In normal operation, two nodes provide redundant paths to the same storage, which means that, depending on zoning and SAN architecture, a single server might see *eight* paths to each LUN presented by the SAN Volume Controller. Each server host bus adapter (HBA) port needs to be zoned to a single port on each SAN Volume Controller node.

Because most operating systems cannot resolve multiple paths back to a single physical device, IBM provides a multipathing device driver. The multipathing driver supported by the SAN Volume Controller is the IBM Subsystem Device Driver (SDD). SDD groups all available paths to a virtual disk device and presents it to the operating system. SDD performs all the path handling and selects the active I/O paths.

SDD supports the concurrent attachment of various DS8000 models, DS6800, ESS, Storwize® V7000, and SAN Volume Controller storage systems to the same host system. Where one or more alternate storage systems are to be attached, you can identify the required version of SDD at this website:

http://www.ibm.com/support/docview.wss?uid=ssg1S7001350

You can use SDD with the native Multipath I/O (MPIO) device driver on AIX and on Microsoft Windows Server 2003 and Windows Server 2008. For AIX MPIO, a Path Control Module (SDDPCM) is provided to deliver I/O load balancing. The Subsystem Device Driver Device Specific Module (SDDDSM) provides multipath I/O support based on the MPIO technology of Microsoft. For newer Linux versions, a Device Mapper Multipath configuration file is available.

15.1.3 SAN Volume Controller Advanced Copy Services

The SAN Volume Controller provides Advanced Copy Services so that you can copy volumes (VDisks) by using FlashCopy and Remote Copy functions. These Copy Services are available for all supported servers that connect to the SAN Volume Controller cluster.

FlashCopy includes these functions:

- Single-Target FlashCopy (FC)
- Multiple-Target FlashCopy (MTFC)
- Cascaded FlashCopy (CFC)
- Incremental FlashCopy (IFC)

FlashCopy makes an instant, point-in-time copy from a source VDisk volume to a target volume. A FlashCopy can be made only to a volume within the same SAN Volume Controller.

Remote Copy includes these functions:

- Metro Mirror
- Global Mirror

Metro Mirror is a synchronous remote copy, which provides a consistent copy of a source volume to a target volume. Metro Mirror can copy between volumes (VDisks) on separate SAN Volume Controller clusters or between volumes within the same I/O Group on the same SAN Volume Controller.

Global Mirror is an asynchronous remote copy, which provides a remote copy over extended distances. Global Mirror can copy between volumes (VDisks) on separate SAN Volume Controller clusters or between volumes within the same I/O Group on the same SAN Volume Controller.

Important: SAN Volume Controller Copy Services functions are incompatible with the DS8000 Copy Services.

For details about the configuration and management of SAN Volume Controller Copy Services, see the Advanced Copy Services chapters of *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933, or *SAN Volume Controller V4.3.0 Advanced Copy Services*, SG24-7574.

A FlashCopy mapping can be created between any two VDisk volumes in a SAN Volume Controller cluster. It is not necessary for the volumes to be in the same I/O Group or storage pool. This functionality can optimize your storage allocation by using an auxiliary storage system (with, for example, lower performance) as the target of the FlashCopy. In this case, the resources of your high-performance storage system are dedicated for production. Your low-cost (lower performance) storage system is used for a secondary application (for example, backup or development).

An advantage of SAN Volume Controller remote copy is that we can implement these relationships between two SAN Volume Controller clusters with different back-end disk subsystems. In this case, you can reduce the overall cost of the disaster-recovery infrastructure. The production site can use high-performance back-end disk systems, and the recovery site can use low-cost back-end disk systems, even where the back-end disk subsystem Copy Services functions are not compatible (for example, different models or different manufacturers). This relationship is established at the volume level and does not depend on the back-end disk storage system Copy Services.

> **Important:** For Metro Mirror copies, the recovery site VDisk volumes must have performance characteristics similar to the production site volumes when a high write I/O rate is present to maintain the I/O response level for the host system.

15.2 SAN Volume Controller performance considerations

The SAN Volume Controller cluster is scalable up to eight nodes. The performance is almost linear when adding more I/O Groups to a SAN Volume Controller cluster, until it becomes limited by other components in the storage infrastructure. While virtualization with the SAN Volume Controller provides a great deal of flexibility, it does not diminish the necessity to have a SAN and disk systems that can deliver the desired performance.

In the following section, we present the SAN Volume Controller concepts and discuss the performance of the SAN Volume Controller. In this section, we assume that there are no bottlenecks in the SAN or on the disk system.

Determining the number of I/O Groups

Growing or adding new I/O Groups to a SAN Volume Controller cluster is a decision that you must make when either a configuration limit is reached or when the I/O load reaches a point where a new I/O Group is needed.

To determine the number of I/O Groups and to monitor the CPU performance of each node, you can use Tivoli Storage Productivity Center (TPC). The CPU performance is related to I/O performance and when the CPUs become consistently 70% busy, you must consider one of these actions:

- Adding more nodes to the cluster and moving part of the workload onto the new nodes
- Moving VDisk volumes to another I/O Group, if the other I/O Group is not busy

> **Important:** A VDisk volume can only be moved to another I/O Group if there is no I/O activity on that volume. Any data in cache on the server must be destaged to disk; it is an off-line operation from the host perspective. The SAN zoning and port masking might need to be updated to give access.

To see how busy your CPUs are, you can use the Tivoli Storage Productivity Center performance report, by selecting the CPU Utilization option.

The following activities affect CPU utilization:

- Volume activity: The preferred node is responsible for handling I/Os to the VDisk volumes and coordinates forwarding the I/Os to the alternate node. Although both systems exhibit similar CPU utilization, the preferred node is slightly busier. To be precise, a preferred node is always responsible for the destaging of writes for volumes that it owns.
- Cache management: The purpose of the cache component is to improve performance of read and write commands by holding some read or write data in SAN Volume Controller memory. Because the nodes in a caching pair have physically separate memories, the cache component must keep the caches on both nodes consistent.
- FlashCopy activity: Each node (of the FlashCopy source) maintains a copy of the bitmap; CPU utilization is similar.
- Mirror Copy activity: The preferred node is responsible for coordinating copy information to the target and also ensuring that the I/O Group is up-to-date with the copy progress information or change block information. Mirroring can increase CPU utilization. We

suggest that you start a Disk Magic sizing project to consider the correct number of node pairs.

With the newly added I/O Group, the SAN Volume Controller cluster can potentially double the I/O rate per second (IOPS) that it can sustain. A SAN Volume Controller cluster can be scaled up to an eight-node cluster with which we quadruple the total I/O rate.

Most common bottlenecks

Analyzing existing client environments, the most common bottleneck for a non-performing SAN Volume Controller installation is caused by insufficiencies in the storage system back end, specifically, too few disk drives. The second most common issue is high SAN Volume Controller port utilizations and documenting the need for adequate bandwidth sizing. SAN Volume Controller CPU utilization issues are the third most common bottleneck.

Number of ports in the SAN used by SAN Volume Controller

Each SAN Volume Controller node has four Fibre Channel ports so that you need eight ports in the SAN per I/O Group, or four ports in each fabric (two from each node).

The SAN Volume Controller ports are more heavily loaded than the ports of a “native” storage system, because the SAN Volume Controller nodes need to handle the I/O traffic of these other components:

- All the host I/O
- The read-cache miss I/O (the SAN Volume Controller cache-hit rate is less than the rate of a DS8000)
- All write destage I/Os (if VDisk mirroring doubled)
- All writes for cache mirroring
- Traffic for remote mirroring

You must carefully plan the SAN Volume Controller port bandwidth.

Number of paths from SAN Volume Controller to disk system

All SAN Volume Controller nodes in a cluster must be able to see the same set of storage system ports on each device. Any operation in this mode, in which two nodes do not see the same set of ports on the same device, is degraded, and the system logs errors that request a resolution.

For the DS8000, there is no controller affinity for the LUNs. So, a single zone for all SAN Volume Controller ports and up to eight DS8000 host adapter (HA) ports must be defined on each fabric. The DS8000 HA ports must be distributed over as many HA cards as available and dedicated to SAN Volume Controller use if possible. Using two or three ports on each DS8000 HA card provides the maximum bandwidth.

Configure a minimum of eight controller ports to the SAN Volume Controller per controller regardless of the number of nodes in the cluster. Configure 16 controller ports for large controller configurations where more than 48 DS8000 ranks are being presented to the SAN Volume Controller cluster.

Optimal storage pool configurations

A storage pool, or Managed Disk Group, provides the pool of storage from which virtual disks are created. It is therefore necessary to ensure that an entire tier in a storage pool provides the same performance and reliability characteristics.

For the DS8000, all LUNs in the same storage pool tier must have these characteristics:

- Use disk drive modules (DDMs) of the same capacity and speed.
- Arrays must be the same RAID type.
- Use LUNs that are the same size.

For the extent size, to maintain maximum flexibility, an extent size of 1 GiB (1024 MiB) is suggested.

For additional information, see the *SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521.

15.3 DS8000 performance considerations with SAN Volume Controller

We suggest the DS8000 configuration to optimize the performance of your virtualized environment.

15.3.1 DS8000 array

The DS8000 storage system provides protection against the failure of individual Disk Drive Modules (DDMs) by the use of RAID arrays. This protection is important, because the SAN Volume Controller provides no protection for the MDisks within a storage pool.

Array RAID configuration

A DS8000 array is a RAID 5, RAID 6, or RAID 10 array that is made up of 8 DDMs. A DS8000 array is created from one array site:

- RAID 5 arrays are either 6+P+S or 7+P.
- RAID 6 arrays are either 5+P+Q+S or 6+P+Q.
- RAID 10 arrays are either 3+3+2S or 4+4.

There are a number of workload attributes that influence the relative performance of RAID 5 compared to RAID 10, including the use of cache, the relative mix of read as opposed to write operations, and whether data is referenced randomly or sequentially.

Consider these RAID characteristics:

- For either sequential or random reads from disk, there is no significant difference in RAID 5 and RAID 10 performance, except at high I/O rates. RAID 6 can also provide acceptable results.
- For random writes to disk, RAID 10 performs better.
- For sequential writes to disk, RAID 5 performs better.
- RAID 6 performance is slightly inferior to RAID 5 but provides protection against two DDM failures.

SAN Volume Controller does not need to influence your choice of the RAID type used. For more details about the RAID 5 and RAID 10 differences, see 4.7, “Planning RAID arrays and ranks” on page 103.

15.3.2 DS8000 rank format

A *rank* is created for each array. There is a one-to-one relationship between arrays and ranks. The SAN Volume Controller requires that ranks are created in Fixed Block (FB) format, which divides each rank into 1 GiB extents (where 1 GiB = 2^{30} bytes). A rank must be assigned to an extent pool to be available for LUN creation.

The DS8000 processor complex (or server group) affinity is determined when the rank is assigned. Assign the same number of ranks in a DS8000 to each of the processor complexes. Additionally, if you do not need to use all of the arrays for your SAN Volume Controller storage pool, select the arrays so that you use arrays from as many device adapters (DAs) as possible, to best balance the load across the DAs also.

15.3.3 DS8000 extent pool implications

In the DS8000 architecture, extent pools are used to manage one or more ranks. An extent pool is visible to both processor complexes in the DS8000, but it is directly managed by only one of them. You must define a minimum of two extent pools with one extent pool created for each processor complex to fully use the resources.

Classical approach: One rank per extent pool configuration

For SAN Volume Controller attachments, many clients formatted the DS8000 ranks in 1:1 assignments between ranks and extent pools, therefore, disabling any DS8000 storage pool striping or auto-rebalancing activity. Then, they located one or two volumes (MDisks) in each extent pool exclusively on one rank only, and put all of those volumes into one SAN Volume Controller storage pool. The SAN Volume Controller controlled striping across all these volumes and balanced the load across the RAID ranks by that method. For example, with 600 GB disks with rank net capacities of up to 3.7 TiB, no more than 2 MDisks (DS8000 volumes) per rank were needed based on the traditional SAN Volume Controller limit on MDisk volume sizes of 2 TiB (SAN Volume Controller 5.*x*)[1]. So, the rank size determines the MDisk size. For example, if the rank is 3698 GiB, make two volumes of 1849 GiB each, and eventually put them in different storage pools to avoid double striping across one rank.

When adding more ranks, MDisks, into the same storage pool, start a manual re-striping by using a scripting tool, as shown in section 5.6 of *SAN Volume Controller Best Practices and Performance Guidelines,* SG24-7521.

Often, clients worked with at least two storage pools: one (or two) containing MDisks of all the 6+P RAID 5 ranks of the DS8000 and the other one (or more) containing the slightly larger 7+P RAID 5 ranks. This approach maintains equal load balancing across all ranks when the SAN Volume Controller striping occurs, because each MDisk in a storage pool is the same size then.

The SAN Volume Controller extent size is the stripe size used to stripe across all these single-rank MDisks.

This approach delivered good performance and has its justifications. However, it also has a few drawbacks:

- There can always be natural skew, for instance, a small file of a few hundred KiB that is heavily accessed. Even with a smaller SAN Volume Controller extent size, such as 256 MiB, this classical setup led in a few cases to ranks that are more loaded than other ranks.

[1] Starting with SAN Volume Controller 6.1, IBM introduced support for external managed disks (MDisks) larger than 2 TiB for certain types of storage systems. But it was not until SAN Volume Controller v6.2 that the DS8800 and DS8700 were identified in the SAN Volume Controller 6.2 Interoperability Matrix as having support for MDisks greater than 2 TiB. The SAN Volume Controller 6.3 restrictions document, which is available at http://www.ibm.com/support/docview.wss?rs=591&uid=ssg1S1003903#_Extents, provides a table with the maximum size of an MDisk dependent on the extent size of the storage pool.

- You need to manually adjust (rebalance) when adding ranks (MDisks) to an existing storage pool.
- With large disk sizes, such as 900 GB and larger, when having more MDisks on one rank, and not as many SAN Volume Controller storage pools, the SAN Volume Controller starts striping across many entities that are effectively in the same rank.
- Clients tend in DS8000 installations to go to larger (multi-rank) extent pools to use modern features, such as auto-rebalancing or tiering.

An advantage of this classical approach is that it delivers more options for fault isolation and control over where a certain volume and extent are located.

Modern approach: Multi-rank extent pool configuration

A more modern approach is to create a few DS8000 only extent pools, for instance, two DS8000 extent pools. Use either DS8000 storage pool striping or automated Easy Tier rebalancing to help prevent from overloading individual ranks.

You need only one MDisk volume size with this approach, because plenty of space is available in each large DS8000 extent pool. Often, clients choose 2 TiB (2048 GiB) MDisks for this approach. Create many 2-TiB volumes in each extent pool until the DS8000 extent pool is full, and provide these MDisks to the SAN Volume Controller to build the storage pools. Two extent pools are still needed so that each DS8000 processor complex (even/odd) is loaded.

You might think that with the classical approach, issues with overloaded ranks disappear due to large pools and DS8000 storage pool striping, even though Easy Tier auto-balancing was not used yet. When enlarging the extent pools, even without auto-rebalancing, re-striping is simple. You use only one DS8000 command. And with Easy Tier auto-rebalancing now available on DS8000 R6.2 and higher, we do not need to pay attention to load rebalancing when enlarging pools. The auto-rebalancing handles this task.

You can also introduce storage tiering with solid-state drives (SSDs), Enterprise drives, and nearline disks. DS8000 Easy Tier auto-tiering can help when creating a Tier 0 (SSDs) and a combined Tier 1^+ (Enterprise+nearline HDDs). SAN Volume Controller based Easy Tier can perform the overall tiering between Tier 0 (SSD) and Tier 1^+ (HDD). The DS8000 based Easy Tier can tier the storage within Tier 1^+, that is, between nearline and Enterprise drives. If you use DS8000 Easy Tier, do not use 100% of your extent pools, you must leave space for Easy Tier so that it can work.

To maintain the highest flexibility and for easier management, large DS8000 extent pools are beneficial. However if the SAN Volume Controller DS8000 installation is dedicated to shared-nothing environments, such as Oracle ASM, DB2 warehouses, or General Parallel File System (GPFS), use the single-rank extent pools.

15.3.4 DS8000 volume considerations with SAN Volume Controller

This section describes our suggestions about volume creation on the DS8000 when assigned to the SAN Volume Controller.

Number of volumes per extent pool and volume size considerations

In a classical (one rank = one extent pool) SAN Volume Controller environment, we suggest that you define one or two volumes per rank. Tests show a small response time advantage to the two LUNs per array configuration and a small IOPS advantage to the one LUN per array configuration for sequential workloads. Overall, the performance differences between these configurations are minimal.

With the DS8000 supporting volume sizes up to 16 TiB and SAN Volume Controller 6.2+ levels supporting these MDisk sizes, a classical approach is still possible when using large disks, such as with an array of the 3 TB nearline disks (RAID 6). The volume size in this case is determined by the rank capacity.

With the modern approach of using large multi-rank extent pools, more clients use a standard volume MDisk size, such as 2 TiB for all MDisks, with good results.

We suggest that you assign SAN Volume Controller DS8000 LUNs of all the same size for each storage pool. In this configuration, the workload applied to a Virtual Disk is equally balanced on the Managed Disks within the storage pool.

Dynamic Volume Expansion

Dynamic Logical Volume Expansion on the DS8000 is not supported for the SAN Volume Controller. All available capacity in the extent pool must be defined to the SAN Volume Controller MDisk to allow VDisk striping over the maximum number of extents. It is better to have the free space on the SAN Volume Controller from the start, because adding space to an existing storage pool does not redistribute the extents that are already defined.

A DS8000 LUN assigned as an MDisk can be expanded only if the MDisk is removed from the storage pool first, which automatically redistributes the defined VDisk extents to other MDisks in the pool, provided there is space available. The LUN can then be expanded, detected as a new MDisk, and reassigned to a storage pool.

Nearline drives

Nearline (7200 rpm) drives are in general unsuited to use as SAN Volume Controller MDisks for high-performance applications. However, if DS8000 Easy Tier is used, they might be added to pools that already consist of many Enterprise HDDs that take the main part of the load, or they can be part of a tiered concept.

15.3.5 Volume assignment to SAN Volume Controller

On the DS8000, we suggest creating one volume group in which to include all the volumes defined to be managed by SAN Volume Controller and all the host connections of the SAN Volume Controller node ports. The DS8000 offers the host type *SVC* for this function.

Volumes can be added dynamically to the SAN Volume Controller. When the volume is added to the volume group, run the command **`svctask detectmdisk`** on the SAN Volume Controller to add it as a new MDisk.

Before you delete or unmap a volume allocated to the SAN Volume Controller, remove the MDisk from the SAN Volume Controller storage pool, which automatically migrates any extents for defined volumes to other MDisks in the storage pool, provided there is space available. When it is unmapped on the DS8000, run the command **`svctask detectmdisk`** and then run the maintenance procedure on the SAN Volume Controller to confirm its removal.

15.4 Performance monitoring

You can use IBM Tivoli Storage Productivity Center (TPC) to manage the IBM SAN Volume Controller and monitor its performance. IBM Tivoli Storage Productivity Center and IBM Tivoli Storage Productivity Center for Disk are described in these books:

- *IBM Tivoli Storage Productivity Center V4.2 Release Guide*, SG24-7894

- *SAN Storage Performance Management Using Tivoli Storage Productivity Center*, SG24-7364
- *IBM TotalStorage Productivity Center Advanced Topics*, SG24-7348

New DS8000 shipments usually include the IBM System Storage Productivity Center (SSPC), which is a storage system management console that includes Tivoli Storage Productivity Center Basic Edition, which provides these functions:

- Storage Topology Viewer
- The ability to monitor capacity, alert, report, and provision storage
- Status dashboard
- IBM System Storage DS8000 GUI integration

The Tivoli Storage Productivity Center Basic Edition provided with the SSPC can be upgraded to the full Tivoli Storage Productivity Center Standard Edition license if required, which, like its module Tivoli Storage Productivity Center for Disk, enables performance monitoring.

15.4.1 Monitoring the SAN Volume Controller with TPC for Disk

To configure Tivoli Storage Productivity Center (TPC) for Disk to monitor IBM SAN Volume Controller, see *SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521.

Data collected from SAN Volume Controller

The two most important metrics when measuring I/O subsystem performance are response time in milliseconds and throughput in I/Os per second (IOPS):

- Response time in non-SAN Volume Controller environments is measured from when the server issues a command to when the storage controller reports the command as completed. With the SAN Volume Controller, we must consider response time from the server to the SAN Volume Controller nodes and also from the SAN Volume Controller nodes to the storage controllers. The VDisk volume response time is what the client sees, but if this number is high and there is no SAN Volume Controller bottleneck, it is determined by the MDisk response time.
- Throughput, however, can be measured at various points along the data path, and the SAN Volume Controller adds additional points where throughput is of interest and measurements can be obtained.

Tivoli Storage Productivity Center offers many disk performance reporting options that support the SAN Volume Controller environment and also the storage controller back end for various storage controller types. The following storage components are the most relevant for collecting performance metrics when monitoring storage controller performance:

- Subsystem
- Controller
- Array
- Managed Disk
- Managed Disk Group, or storage pool
- Port

With the SAN Volume Controller, you can monitor on the levels of the I/O Group and the SAN Volume Controller node.

SAN Volume Controller thresholds

Thresholds are used to determine watermarks for warning and error indicators for an assortment of storage metrics. SAN Volume Controller has the following thresholds within its default properties:

- Volume (VDisk) I/O rate

  Total number of virtual disk I/Os for each I/O Group
- Volume (VDisk) bytes per second

  Virtual disk bytes per second for each I/O Group
- MDisk I/O rate

  Total number of managed disk I/Os for each Managed Disk Group
- MDisk bytes per second

  Managed disk bytes per second for each Managed Disk Group

The default status for these properties is Disabled with the Warning and Error options set to None. Enable a particular threshold *only* after the minimum values for warning and error levels are defined.

> **Tip:** In Tivoli Storage Productivity Center for Disk, default threshold warning or error values of -1.0 are indicators that there is no suggested minimum value for the threshold and are therefore entirely user-defined. You can choose to provide any reasonable value for these thresholds based on the workload in your environment.

15.4.2 TPC Reporter for Disk

The TPC Reporter for Disk V2, which is mentioned in 7.5.8, "TPC Reporter for Disk" on page 279, is also monitoring and reporting the SAN Volume Controller performance counters. So when using TPC Reporter from the following website, you can gather performance PDF documents for both your SAN Volume Controller and for all the storage systems behind it:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS2618

15.4.3 The TPC Storage Tiering Reports and Storage Performance Optimizer

Starting with Tivoli Storage Productivity Center V4.2.2, Tivoli Storage Productivity Center provides capabilities for reporting on storage tiering activity to support data placement and to optimize resource utilization in a virtualized environment. The storage tiering reports use the estimated capability and actual performance data for the IBM SAN Volume Controller and offer storage administrators key insights:

- Are the back-end systems optimally utilized?
- Does moving a certain workload to low-cost storage affect service levels?
- How do I level out performance in a certain pool?
- Which data groups can be moved to an alternate tier of storage?

So, the Storage Tiering Reports combine storage virtualization and Tivoli Storage Productivity Center information to help users make smart decisions. The reports provide the capability to make "proactive" volume placement decisions. They take into account back-end storage configuration characteristics for analytic modeling. Predefined reports in Cognos® on storage pools (Managed Disk Group) or the virtual volumes provide details about the hottest and coolest resources and as well as detailed performance and historic capacity reports.

The Storage Tiering Reports can be used complementary and in addition to what the Tivoli Storage Productivity Center Storage Performance Optimizer can do. The Optimizer analyzes based on utilization levels, creates a graphical overview (heat map) and makes logical migration suggestions based on performance requirements and the heat map. Working together with the IBM SAN Volume Controller, clients can perform the non-disruptive movement of storage pools to an optimal configuration even cross-system without interrupting normal day-to-day operations.

The following paper describes the storage tiering reports of the SAN Volume Controller:

http://www.ibm.com/support/docview.wss?uid=swg27023263

15.5 Sharing the DS8000 between a server and the SAN Volume Controller

The DS8000 can be shared between servers and a SAN Volume Controller. This sharing can be useful if you want direct attachment for specific Open Systems servers or if you need to share your DS8000 between the SAN Volume Controller and an unsupported server, such System z. Also, this option might be appropriate for IBM i, which is only supported through VIOS with SAN Volume Controller.

For the latest list of hardware that is supported for attachment to the SAN Volume Controller, see this website:

http://www.ibm.com/systems/storage/software/virtualization/svc/interop.html

15.5.1 Sharing the DS8000 between Open Systems servers and the IBM SAN Volume Controller

If you have a mixed environment that includes IBM SAN Volume Controller and Open Systems servers, we suggest sharing the maximum of the DS8000 resources to both environments.

If an extent pool has multiple ranks, it is possible to create all SAN Volume Controller volumes by using the *rotate volumes* algorithm, which provides one larger MDisk per rank. Server volumes can use the *rotate extents* algorithm if desired. You can also use the *rotate extents* method with the SAN Volume Controller volumes for acceptable performance. If DS8000 Easy Tier is enabled and manages the extent pools, it auto-rebalances all DS8000 volumes, whether they are SAN Volume Controller MDisks or direct server volumes.

Most clients choose a DS8000 extent pool pair (or pairs) for their SAN Volume Controller volumes only, and other extent pool pairs for their directly attached servers. This approach is a preferred practice, but you can fully share on the drive level if preferred.

I/O Priority Manager works on the level of full DS8000 volumes. So, when you have large MDisks, which are the DS8000 volumes, I/O Priority Manager cannot prioritize between various smaller VDisk volumes that are cut out of these MDisks. I/O Priority Manager enables SAN Volume Controller volumes as a whole to be assigned different priorities compared to other direct server volumes. For instance, if IBM i mission-critical applications with directly attached volumes share extent pools with SAN Volume Controller MDisk volumes (uncommon), the I/O Priority Manager can throttle the complete SAN Volume Controller volumes in I/O contention to protect the IBM i application performance.

IBM supports sharing a DS8000 between a SAN Volume Controller and an Open Systems server. However, if a DS8000 port is in the same zone as a SAN Volume Controller port, that same DS8000 port must not be in the same zone as another server.

15.5.2 Sharing the DS8000 between System z servers and the IBM SAN Volume Controller

IBM SAN Volume Controller does not support System z server attachment. If you have a mixed server environment that includes IBM SAN Volume Controller and System z servers, you must share your DS8000 to provide a direct access to System z volumes and access to Open Systems server volumes through the IBM SAN Volume Controller.

In this case, you must split your DS8000 resources between two environments. You must create part of the ranks by using the count key data (CKD) format (used for System z access) and the other ranks in Fixed Block (FB) format (used for IBM SAN Volume Controller access). In this case, both environments get performance that is related to the allocated DS8000 resources.

A DS8000 port does not support a shared attachment between System z and IBM SAN Volume Controller. System z servers use the Fibre Channel connection (FICON), and IBM SAN Volume Controller supports Fibre Channel Protocol (FCP) connection only.

15.6 Advanced functions for the DS8000

The DS8000 provides Copy Services functions that are incompatible with the SAN Volume Controller Advanced Copy Services.

Because of the availability of cache-disabled VDisks in the SAN Volume Controller, you can enable Copy Services in the underlying RAID array controller for these LUNs.

15.6.1 Cache-disabled VDisks

Where Copy Services are used in the DS8000, the controller LUNs at both the source and destination must be mapped through the SAN Volume Controller as image mode *cache-disabled VDisks*. It is possible to access either the source or the target of the remote copy from a server directly, rather than through the SAN Volume Controller. The SAN Volume Controller Copy Services can be employed with the image mode VDisk representing the primary of the controller copy relationship. However, it does not make sense to use SAN Volume Controller Copy Services with the VDisk at the secondary site, because the SAN Volume Controller does not see the data that flows to this LUN through the controller.

Cache-disabled VDisks are primarily used when virtualizing an existing storage infrastructure, and you need to retain the existing storage system Copy Services. You might want to use cache-disabled VDisks where there is significant intellectual capital in existing Copy Services automation scripts. We suggest that you keep the use of cache-disabled VDisks to a minimum for normal workloads.

Another case where you might need to use cache-disabled VDisks is with servers, such as System z or non-VIOS IBM i, that are not supported by the SAN Volume Controller, but you need to maintain a single Global Mirror session for consistency between all servers. In this case, the DS8000 Global Mirror must be able to manage the LUNs for all server systems.

Important: When configuring cache-disabled VDisks, consider the guidelines for configuring the DS8000 for host system attachment, which has as great an impact on performance as the SAN Volume Controller.

Because the SAN Volume Controller does not stripe VDisks, in this case it is an advantage to use extent pools with multiple ranks to allow volumes to be created by using the rotate extents algorithm, or managed extent pools. The guidelines for the use of different DA pairs for FlashCopy source and target LUNs with affinity to the same DS8000 server also apply.

Cache-disabled VDisks can also be used to control the allocation of cache resources. By disabling the cache for certain VDisks, more cache resources are available to cache I/Os to other VDisks in the same I/O Group. This technique is effective where an I/O Group serves VDisks that can benefit from cache and other VDisks where the benefits of caching are small or non-existent.

The usage of the DS8000 Copy Services functions when having SAN Volume Controller in place is rare. Usually, in SAN Volume Controller attachments, the DS8000 is provided without any DS8000 Copy Services license features, only Oracle Enterprise Linux (OEL) and Easy Tier. The SAN Volume Controller Copy Services features and algorithms are used for replication to a secondary SAN Volume Controller site.

15.7 Configuration guidelines for optimizing performance

Guidelines: Many of these guidelines are not unique to configuring the DS8000 for SAN Volume Controller attachment. In general, any server can benefit from a balanced configuration that uses the maximum available bandwidth of the DS8000.

Follow the guidelines and procedures outlined in this section to make the most of the performance available from your DS8000 storage systems and to avoid potential I/O problems:

- Use multiple host adapters on the DS8000. Where possible, use no more than two ports on each card. Use a larger number of ports on the DS8000, usually 16 (SAN Volume Controller maximum).
- Unless you have special requirements, or if in doubt, build your MDisk volumes from large extent pools on the DS8000.
- If using a 1:1 mapping of ranks to DS8000 extent pools, use one, or a maximum of two volumes on this rank, and adjust the MDisk volume size for this 1:1 mapping.
- Create fewer and larger SAN Volume Controller storage pools and have multiple MDisks in each pool.
- For SAN Volume Controller releases before 6.3, do not create too few MDisk volumes. If you have too few MDisk volumes, the older SAN Volume Controller releases do not use all offered ports.
- Keep many DS8000 arrays active.
- Ensure that you have an equal number of extent pools and, as far as possible, spread the volumes equally across the device adapters and the two processor complexes of the DS8000 storage system.
- In a storage pool, ensure that for a certain tier, all MDisks have the same capacity and RAID/rpm characteristics.

- Do not mix HDD MDisks from different controllers in the same storage pool.
- For Metro Mirror configurations, always use DS8000 MDisks with similar characteristics for both the master VDisk volume and the auxiliary volume.
- Spread the VDisk volumes across all SAN Volume Controller nodes, and check for balanced preferred node assignments.
- In the SAN, use a dual fabric.
- Use multipathing software in the servers.
- Consider DS8000 Easy Tier auto-rebalancing for DS8000 homogeneous capacities.
- When using Easy Tier in the DS8000, consider a SAN Volume Controller extent size of 1 GiB (1024 MiB) minimum, not to put skew away from the DS8000 extents. You can only consider smaller SAN Volume Controller extent sizes, such as 256 or 512 MiB, to use this SSD space better when internal SSDs are present in the SAN Volume Controller nodes, because then the SAN Volume Controller Easy Tier is used.
- When using DS8000 Easy Tier, leave some small movement space empty in the extent pools to help it start working.
- Consider the right amount of cache, as explained in 2.2.2, "Determining the right amount of cache storage" on page 33. Usually, SAN Volume Controller installations have a DS8000 cache of not less than 128 GB.

15.8 Where to place the SSDs

Today, storage installations get deployed as tiered environments that contain some solid-state drives (SSDs). When implementing storage virtualization with the SAN Volume Controller, you can place SSDs in several places:

- SSDs in a particular server
- SSDs in the SAN Volume Controller nodes
- SSDs in a separate storage system (with the DS8000 handling the Enterprise drives)
- SSDs in the DS8000 (together with Enterprise and nearline disks

Each method has advantages and drawbacks.

SSDs in the server

Only one server benefits from the SSDs; all other servers are slower. Easy Tiering is not possible, so the server or database administrator manually must select those parts of the database to put on SSDs, or continually monitor and adjust data placement. RAID protection is still necessary. You need a server RAID adapter that can handle the amount of IOPS. In cases of backup automation or 2-site/3-site replication setups, these volumes are not part of the overall replication concept and offer less general protection. This concept is mostly discouraged.

SSDs in the SAN Volume Controller nodes

Up to four SSDs can be put into one SAN Volume Controller node. This design enables a low entry point to start with SSDs. Easy Tier works at the SAN Volume Controller level. Because many IOPS are written to and through these SSDs, RAID protection, configured by the storage administrator, is necessary. Never leave any of the disks without RAID protection, because in availability, the weakest point determines the overall availability. Ensure that you have an equal load on each node.

Each SSD MDisk goes into one pool, determining how many storage pools can benefit from SAN Volume Controller Easy Tier. The SSD size determines the granularity of the offered

SSD capacity. Scalability is limited compared to SSDs in a DS8000. With up to 8 SAN Volume Controller nodes in a cluster, we can work with up to 32 single SSDs only.

SSDs in a separate storage system

With a SAN Volume Controller, you can separate the SSD storage system from the HDD (spinning drive) storage system. The SAN Volume Controller Easy Tier automatically handles the optimal tiered configuration. For example, one system can be a DS8000, and the other system a DS3000, or vice versa. However, availability is an issue. If you consider your data or most of your data valuable enough that it needs to go on a System Storage DS8000 with the highest possible availability, you risk your data by putting another part of the data onto an entry-level machine. The SAN Volume Controller Easy Tier always creates mixed VDisk volumes that consist of DS3000 and DS8000 parts. Also, we cannot enable DS8000 Easy Tier across all tiers. Practically, we do not mix the DS8000 based and DS3000 based MDisks in one pool because of failure boundaries. The pool goes offline if one system fails.

Another argument against this concept is that the ports that handle the traffic to the SSD-only storage system experience exceptionally high workloads.

SSDs in the DS8000

This concept is disadvantageous because you need a minimum number of SSD drives of 8 (1 rank) with a minimum of 16 SSDs (2 ranks) suggested to balance the load across the two processor complexes. So, it is more expensive. But in general this approach offers many advantages from this concept. In this approach, we can enable DS8000 Easy Tier if we want, which also monitors the SSD device adapter usage and has more sophisticated algorithms for checking overloaded SSD (or HDD) ranks, or device adapters, and perform warm demotions if needed. Efficient RAID protection is performed by using the DS8000 device adapters. And scalability is higher, with hundreds of possible SSDs. Also, the DS8000 cache and advanced cache algorithms are used best.

15.9 Where to place Easy Tier

IBM Easy Tier is an algorithm developed by IBM Almaden Research and made available to storage systems, such as DS8800 and DS8700, as well as to the SAN Volume Controller. When using Easy Tier in the SAN Volume Controller with a mixed-tier storage pool, you flag the MDisks as *generic_ssd* when you introduce them to the SAN Volume Controller storage pool, instead of the default attribute *generic_hdd*. The generic_ssd attribute is not supported for disks that are not SSDs, for instance, to build SAN Volume Controller managed 2-tier pools with Enterprise and nearline disks.

When using the internal SSDs in the SAN Volume Controller nodes, only Easy Tier performed by the SAN Volume Controller is possible for the inter-tier movements between SSD and HDD tiers. The DS8000 intra-tier auto-rebalancing can be used and can monitor the usage of all the HDD ranks and move loads intra-tier by DS8000 if some ranks are more loaded. If you have Enterprise and nearline drives in the DS8000, you can also combine the DS8000 managed inter-tier Easy Tier (only for the 2-tier DS8000 HDD extent pool) with the SAN Volume Controller managed overall Easy Tier between the SAN Volume Controller SSDs and the 2-tier HDD pool, out of which you cut the HDD MDisks. Two tiers of HDDs justify DS8000 managed Easy Tier, at least for the HDD part.

When you have the SSDs in the DS8000, together with Enterprise HDDs and eventually also nearline HDDs, on which level do you perform the overall inter-tier Easy Tiering? It can be either done by the SAN Volume Controller, by setting the generic_ssd attribute for all the DS8000 SSD MDisks. Or, leave the generic_hdd attribute for all MDisks, but allow DS8000

Easy Tier to manage these MDisks, with 2-tier or 3-tier MDisks offered to the SAN Volume Controller, and contain some SSDs (which are invisible to the SAN Volume Controller).

There are differences between the Easy Tier algorithms in the DS8000 and in SAN Volume Controller. The DS8000 is in the third generation of Easy Tier, with additional functions available, such as Extended-Cold-Demote or Warm-Demote. The warm demote checking is reactive if certain SSD ranks or SSD device adapters suddenly are overloaded. The SAN Volume Controller needs to work with different vendors and varieties of SSD space offered, and use a more generic algorithm, which cannot learn easily whether the SSD rank of a certain vendor's disk system is approaching its limits.

As a rule when you use SSDs in the DS8000 and use many or even heterogeneous storage systems, consider implementing cross-tier Easy Tier on the highest level, that is, managed by the SAN Volume Controller. SAN Volume Controller Version 6.1 and higher can use larger blocksizes, such as 60K and over, which do not work well for DS8000 Easy Tier, so we have another reason to use SAN Volume Controller Easy Tier inter-tiering. However, observe the system by using the STAT for SAN Volume Controller. If the SSD space gets overloaded, consider either adding more SSDs as suggested by the STAT, or removing and reserving some of the SSD capacity so that it is not fully utilized by SAN Volume Controller, by creating smaller SSD MDisks and leaving empty space there.

If you have one main machine behind the SAN Volume Controller, leave the Easy Tier inter-tiering to the DS8000 logic. Use the more sophisticated DS8000 Easy Tier algorithms that consider the sudden overload conditions of solid-state drive (SSD) ranks or adapters. DS8000 Easy Tier algorithms have more insight into the DS8000 thresholds and what workload limit each component of the storage system can sustain. Choose a SAN Volume Controller extent size of 1 GiB to not eliminate the skew for the DS8000 tiering, which is on the 1 GiB extent level.

We advise that you use the most current level of SAN Volume Controller before you start SAN Volume Controller managed Easy Tier and that you use recent SAN Volume Controller node hardware.

16

IBM ProtecTIER deduplication

This chapter introduces you to attaching the DS8000 to an IBM ProtecTIER deduplication gateway.

16.1 IBM TS7600 ProtecTIER deduplication

Data deduplication is a technology that is used to reduce the amount of space required to store data on disk. Data deduplication is achieved by storing a single copy of data that is backed up repetitively. Data deduplication can provide greater data reduction than alternative technologies, such as compression and differencing. Differencing is used for differential backups.

Figure 16-1 illustrates the basic components of data deduplication for IBM System Storage TS7600 ProtecTIER servers.

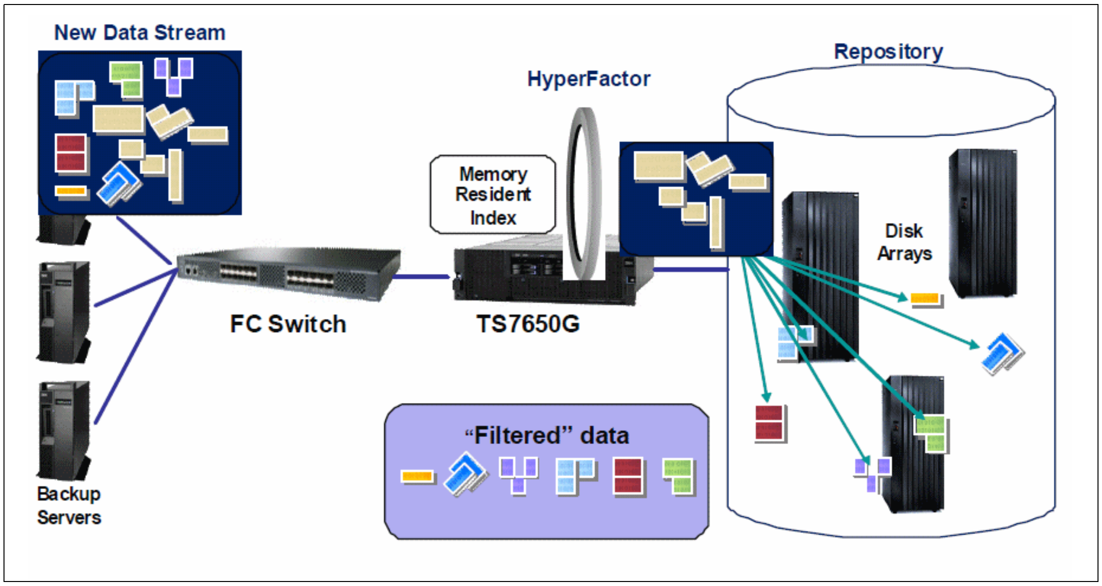


Figure 16-1 Basic concept of ProtecTIER deduplication

With data deduplication, data is read by the data deduplication product while it looks for duplicate data. Different data deduplication products use different methods of breaking up the data into elements, but each product uses a technique to create a signature or identifier for each data element. After the duplicate data is identified, one copy of each element is retained, pointers are created for the duplicate items, and the duplicate items are not stored.

The effectiveness of data deduplication depends on many variables, including the data rate of data change, the number of backups, and the data retention period. For example, if you back up the same incompressible data once a week for six months, you save the first copy and do not save the next 24. This example provides a 25:1 data deduplication ratio. If you back up an incompressible file on week one, back up the same file again on week two, and never back it up again, you have a 2:1 deduplication ratio.

The IBM System Storage TS7650G is a preconfigured Virtualization solution of IBM systems. The IBM ProtecTIER data deduplication software is designed to improve backup and recovery operations. The solution is available in a single-node or two-node cluster configurations designed to meet the disk-based data protection needs of a wide variety of IT environments and data centers. The TS7650G ProtecTIER Deduplication Gateway can scale to repositories in the petabyte (PB) range, and all DS8000 models are supported behind it. Your DS8000 can become a Virtual Tape Library (VTL). The multi-node concepts help achieve higher throughput and availability for the backup, and replication concepts are available.

For a detailed assessment of ProtecTIER and deduplication, see these publications:

- *IBM System Storage TS7600 with ProtecTIER Version 3.1*, SG24-7968
- *Implementing IBM Storage Data Deduplication Solutions*, SG24-7888
- *IBM System Storage TS7650, TS7650G, and TS7610*, SG24-7652
- *TS7680 Deduplication ProtecTIER Gateway for System z*, SG24-7796

16.2 DS8000 attachment considerations

A ProtecTIER gateway attachment with DS8000 resembles an IBM System Storage SAN Volume Controller attachment. Several of the SAN Volume Controller guidelines might also apply. For instance, you can choose to dedicate your full DS8000 to the SAN Volume Controller or ProtecTIER device. Or, you can share a DS8000, also with ProtecTIER, with direct server attachment of other volumes on the same DS8000, reducing the data center footprint. Backups and active data of a specific server must be physically separate for availability reasons. Therefore, in a multi-site concept, a DS8000 can incorporate the production data of some servers and backup data of other servers that are at another data center, if this type of sharing is used.

Extent pool isolation

For ProtecTIER, if a client chooses to share the DS8000 between active server volumes and ProtecTIER Gateway backup volumes, we suggest a separation within the DS8000 at the extent pool level. This way, we dedicate a certain extent pool pair (or pairs) fully to the ProtecTIER Gateway nodes only, and others for the production servers. The production server attachments go onto higher rpm hard disk drives (HDDs) or even solid-state drives (SSDs). The overall mix on the DS8000 system level uses the investment in DS8000 better.

Host bus adapters

We suggest that you use dedicated ports and application-specific integrated circuits (ASICs) for the I/O traffic to the ProtecTIER appliance, as opposed to server I/O attachment traffic on other DS8000 ports. It is a preferred practice to keep disk and tape I/O traffic on separate Small Computer System Interface (SCSI) buses.

Drive types used

Similar to planning backups, you might consider the least expensive drives for this environment, which are the nearline drives, such as 3 TB 7200 min^{-1} of a DS8800. Nearline drives usually come with RAID 6 data protection, which also protects data in a double-disk failure in the array, because rebuild times can get long on high capacities.

ProtecTIER access patterns though can have a high random-read content, with some random-write ratio for the metadata. Therefore, Enterprise drives with their higher rpm speeds and other RAID types outperform nearline drives also for ProtecTIER attachments. You can format the 2 TB 7200 min^{-1} Serial Advanced Technology Attachment (SATA) drives of a DS8700 in RAID 10. However, because Easy Tier automatically tiers storage between Enterprise and nearline drives, we suggest that you use a drive mix of Enterprise and nearline drives and let Easy Tier optimize between both tiers for you. Leave a small movement space empty so that the Easy Tier can start. The two tiers can have different RAID formats. With tiering in the DS8000, the ProtecTIER data that is more random-write-oriented automatically is promoted to the most suitable drive tier. For the overall sizing and choice of drives between Enterprise and nearline HDDs, consult your IBM representative.

Part 4

Advanced topics

This part provides additional performance considerations for advanced topics, such as databases and DS8000 Copy Services.

This part includes the followign topics:

- Database performance
- Copy Services performance

17

Database performance

This chapter reviews the major IBM database systems and the performance considerations when they are used with the DS8000 disk subsystem. We limit our discussion to the following databases:

- DB2 Universal Database™ (DB2 UDB) in a z/OS environment
- DB2 in an open environment
- IMS in a z/OS environment
- Oracle in an open environment

You can obtain additional information about IBM DB2 and IMS at these websites:

- http://www.ibm.com/software/data/db2/zos/family/
- http://www.ibm.com/software/data/db2/linux-unix-windows/
- http://www.ibm.com/software/data/ims/

17.1 DB2 in a z/OS environment

In this section, we provide a description of the characteristics of the various database workloads, as well as the types of data-related objects used by DB2 (in a z/OS environment). Also, we discuss the performance considerations and general guidelines for using DB2 with the DS8000, as well as a description of the tools and reports that can be used for monitoring DB2.

17.1.1 Understanding your database workload

To better understand and position the performance of your particular database system, it is helpful to first learn about the following common database profiles and their unique workload characteristics.

DB2 online transaction processing (OLTP)

OLTP databases are among the most mission-critical and widely deployed of all. The primary defining characteristic of OLTP systems is that the transactions are processed in real time or online and often require immediate response back to the user. The following examples are OLTP systems:

- A point of sale terminal in a retail business
- An automated teller machine (ATM) used for bank transactions
- A telemarketing site that processes sales orders and checks the inventories

From a workload perspective, OLTP databases typically have these characteristics:

- Process many concurrent user sessions
- Process many transactions by using simple SQL statements
- Process a single database row at a time
- Are expected to complete transactions in seconds, not minutes or hours

OLTP systems process the day-to-day operation of businesses and, therefore, have strict user response and availability requirements. They also have high throughput requirements and are characterized by large numbers of database inserts and updates. They typically serve hundreds, or even thousands, of concurrent users.

Decision support systems (DSSs)

DSSs differ from the typical transaction-oriented systems in that they often use data extracted from multiple sources to support user decision making. DSSs use these types of processing:

- Data analysis applications using predefined queries
- Application-generated queries
- Ad hoc user queries
- Reporting requirements

DSS systems typically deal with substantially larger volumes of data than OLTP systems due to their role in supplying users with large amounts of historical data. While 100 GB of data is considered large for an OLTP environment, a large DSS system might be 1 TB of data or more. The increased storage requirements of DSS systems can also be attributed to the fact that they often contain multiple, aggregated views of the same data.

While OLTP queries are mostly related to one specific business function, DSS queries are often substantially more complex. The need to process large amounts of data results in many CPU-intensive database sort and join operations. The complexity and variability of these types of queries must be given special consideration when estimating the performance of a DSS system.

17.1.2 DB2 overview

DB2 is a database management system based on the relational data model. Most users choose DB2 for applications that require good performance and high availability for large amounts of data. This data is stored in datasets mapped to DB2 tablespaces and distributed across DB2 databases. Data in tablespaces is often accessed by using indexes that are stored in index spaces.

Data tablespaces can be divided in two groups: *System tablespaces* and *user tablespaces*. Both of these tablespaces have identical data attributes. The difference is that system tablespaces are used to control and manage the DB2 subsystem and user data. System tablespaces require the highest availability and special considerations. User data cannot be accessed if the system data is not available.

In addition to data tablespaces, DB2 requires a group of traditional datasets not associated to tablespaces that are used by DB2 to provide data availability: The backup and recovery datasets.

In summary, there are three major dataset types in a DB2 subsystem:

- DB2 system tablespaces
- DB2 user tablespaces
- DB2 backup and recovery datasets

The following sections describe the objects and datasets that DB2 uses.

17.1.3 DB2 storage objects

DB2 manages data by associating it to a set of DB2 objects. These objects are logical entities, and several of them are kept in storage. The following list shows the DB2 data objects:

- TABLE
- TABLESPACE
- INDEX
- INDEXSPACE
- DATABASE
- STOGROUP

Here, we briefly describe each of them.

TABLE

All data managed by DB2 is associated to a table. The *table* is the main object used by DB2 applications.

TABLESPACE

A *tablespace* is used to store one or more tables. A tablespace is physically implemented with one or more datasets. Tablespaces are VSAM linear datasets (LDS). Because tablespaces can be larger than the largest possible VSAM dataset, a DB2 tablespace can require more than one VSAM dataset.

INDEX

A table can have one or more indexes (or can have no index). An *index* contains keys. Each key points to one or more data rows. The purpose of an index is to get direct and faster access to the data in a table.

INDEXSPACE

An *index space* is used to store an index. An index space is physically represented by one or more VSAM LDSs.

DATABASE

The *database* is a DB2 representation of a group of related objects. Each of the previously named objects must belong to a database. DB2 databases are used to organize and manage these objects.

STOGROUP

A DB2 *storage group* is a list of storage volumes. STOGROUPs are assigned to databases, tablespaces, or index spaces when using DB2 managed objects. DB2 uses STOGROUPs for disk allocation of the table and index spaces.

Installations that are storage management subsystem (SMS)-managed can define STOGROUP with VOLUME(*). This specification implies that the SMS assigns a volume to the table and index spaces in that STOGROUP. To assign a volume to the table and index spaces in the STOGROUP, SMS uses automatic class selection (ACS) routines to assign a storage class, a management class, and a storage group to the table or index space.

17.1.4 DB2 dataset types

DB2 uses system and user tablespaces for the data, as well as a group of datasets not associated with tablespaces that are used by DB2 to provide data availability; these datasets are backup and recovery datasets.

DB2 system tablespaces

DB2 uses databases to control and manage its own operation and the application data:

- The catalog and directory databases

  Both databases contain DB2 system tables. DB2 system tables hold data definitions, security information, data statistics, and recovery information for the DB2 system. The DB2 system tables reside in DB2 system tablespaces.

  The DB2 system tablespaces are allocated when a DB2 system is first created. DB2 provides the IDCAMS statements required to allocate these datasets as VSAM LDSs.

- The work database

  The work database is used by DB2 to resolve SQL queries that require temporary workspace. Multiple tablespaces can be created for the work database.

DB2 application tablespaces

All application data in DB2 is organized in the objects as described in 17.1.3, “DB2 storage objects” on page 561.

Application tablespaces and index spaces are VSAM LDSs with the same attributes as DB2 system tablespaces and index spaces. System and application data differ only because they have different performance and availability requirements.

DB2 recovery datasets

To provide data integrity, DB2 uses datasets for recovery purposes. We briefly describe these DB2 recovery datasets. These datasets are described in further detail in the *DB2 Version 9.1 for z/OS Administration Guide,* SC18-9840-07.

DB2 uses these recovery datasets:

- Bootstrap dataset

  DB2 uses the bootstrap dataset (BSDS) to manage recovery and other DB2 subsystem information. The BSDS contains information needed to restart and to recover DB2 from any abnormal circumstance. For example, all log datasets are automatically recorded with the BSDS. While DB2 is active, the BSDS is open and updated.

  DB2 always requires two copies of the BSDS, because they are critical for data integrity. For availability reasons, the two BSDS datasets must be put on separate servers on the DS8000 or in separate logical control units (LCUs).

- Active logs

  The active log datasets are used for data recovery and to ensure data integrity in software or hardware errors.

  DB2 uses active log datasets to record all updates to user and system data. The active log datasets are open as long as DB2 is active. Active log datasets are reused when the total active log space is used up, but only after the active log (to be overlaid) is copied to an archive log.

  DB2 supports dual active logs. We strongly suggest that you use dual active logs for all DB2 production environments. For availability reasons, the log datasets must be put on separate servers on the DS8000 or separate LCUs.

- Archive logs

  Archive log datasets are DB2 managed backups of the active log datasets. Archive log datasets are automatically created by DB2 whenever an active log is filled. DB2 supports dual archive logs, and we suggest that you use dual archive log datasets for all production environments.

  Archive log datasets are sequential datasets that can be defined on disk or on tape and migrated and deleted with standard procedures.

17.2 DS8000 considerations for DB2

By using the DS8000 in a DB2 environment, you can help realize the following benefits:

- DB2 takes advantage of the parallel access volume (PAV) function that allows multiple concurrent I/Os to the same volume at the same time from applications that run on a z/OS system image.
- Less disk contention when accessing the same volumes from different systems in a DB2 data sharing group that uses the Multiple Allegiance function.
- Higher bandwidth on the DS8000 allows higher I/O rates to be handled by the disk subsystem, thus allowing for higher application transaction rates.

17.3 DB2 with DS8000 performance recommendations

When using the DS8000, the following generic recommendations are useful when planning for good DB2 performance.

17.3.1 Know where your data resides

DB2 storage administration can be done by using SMS to simplify disk use and control, or also without using SMS. In both cases, it is important that you know where your data resides.

If you want optimal performance from the DS8000, do not treat the DS8000 like a “black box.” Understand how DB2 tables map to underlying volumes and how the volumes map to RAID arrays.

17.3.2 Balance workload across DS8000 resources

You can balance workload activity across the DS8000 resources by using the following methods:

- Spreading DB2 data across DS8000s if practical
- Spreading DB2 data across servers in each DS8000
- Spreading DB2 data across the DS8000 device adapters
- Spreading DB2 data across as many extent pools/ranks as practical

You can intermix tables and indexes and also system, application, and recovery datasets on the DS8000 ranks. The overall I/O activity is then more evenly spread, and I/O skews are avoided.

17.3.3 Solid-state drives

The access to data with a solid-state drive (SSD) is faster than access with a hard disk drive (HDD), because there is no read/write head to move and no magnetic platters need to spin (no latency). Datasets with high I/O rates and poor cache hit ratios are ideal candidates for SSDs. Datasets with good cache hit ratios or low I/O rates need to remain on spinning drives. SSDs are ideal for small-block/random workloads. Sequential workloads must stay on spinning disks.

The results in Figure 17-1 were measured by a DB2 I/O benchmark. They show random 4 KB read throughput and response times. The SSD response times are low across the curve. They are lower than the minimum HDD response time for all data points.

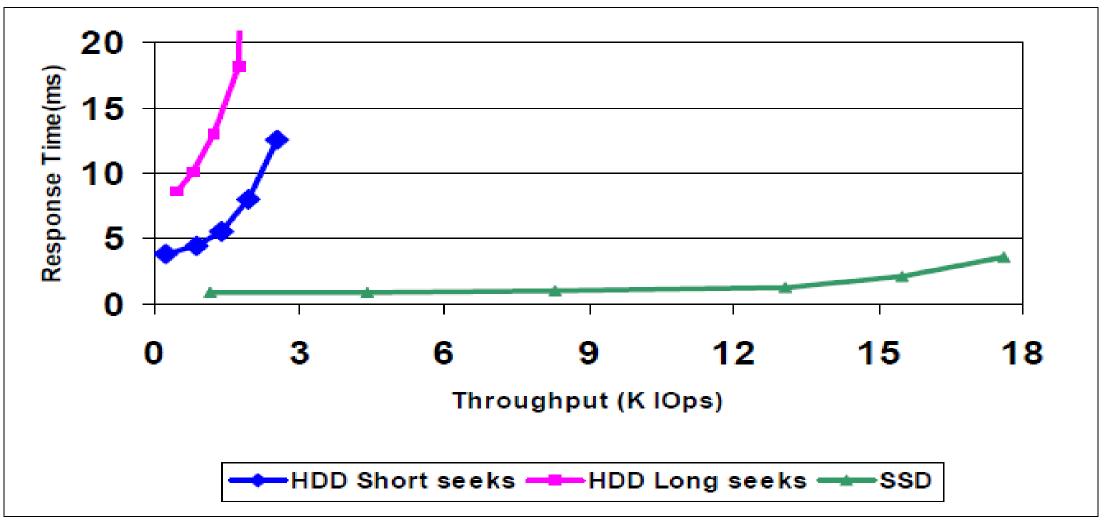


Figure 17-1 DB2 on count key data (CKD) random read throughput/response time curve

Identifying volumes or datasets to move to SSDs

The DS8000 can obtain cache statistics for every volume in the storage subsystem. These measurements include the count of the number of operations from DASD cache to the back-end storage, the number of random operations, the number of sequential reads and sequential writes, the time to execute those operations, and the number of bytes transferred. These statistics are placed in the SMF 74 subtype 5 record.

The FLASHDA is a tool based on SAS software to manage the transition to SSD. The tool provides volumes and dataset usage reports that use SAS code to analyze SMF 42 subtype 6 and SMF 74 subtype 5 records to help identify volumes and datasets that are good candidates to reside on SSDs.

This tool is available for download from the web page:

`http://www.ibm.com/systems/z/os/zos/downloads/flashda.html`

The FLASHDA user guide is available at this website:

`http://publibz.boulder.ibm.com/zoslib/pdf/flashda.pdf`

You can obtain reports to perform these tasks

- Identify page sets with high amounts of READ_ONLY_DISC time
- Analyze DASD cache statistics to identify volumes with high I/O rates
- Identify the datasets with the highest amount of write activity (number of write requests)

Based on the output report of these tools, you can select which hot volumes might benefit most when migrated to SSDs. The tool output also provides the hot data at the dataset level. Based on this data, the migration to the SSD ranks can be done by dataset by using the appropriate z/OS tools.

17.3.4 Take advantage of VSAM data striping

Before VSAM data striping was available, in a multi-extent, multi-volume VSAM dataset, sequential processing did not present any type of parallelism. When an I/O operation was executed for an extent in a volume, no other activity from the same task was scheduled to the other volumes.

VSAM data striping addresses this problem with two modifications to the traditional data organization:

- The records are not placed in key ranges along the volumes; instead, they are organized in stripes.
- Parallel I/O operations are scheduled to sequential stripes in different volumes.

By striping data, the VSAM control intervals (CIs) are spread across multiple devices. This format allows a single application request for records in multiple tracks and CIs to be satisfied by concurrent I/O requests to multiple volumes.

The result is improved data transfer to the application. The scheduling of I/O to multiple volumes to satisfy a single application request is referred as an *I/O path packet*.

We can stripe across ranks, device adapters, servers, and the DS8000s.

In a DS8000 with storage pool striping, the implementation of VSAM striping still provides a performance benefit. Because DB2 uses two engines for the list prefetch operation, VSAM striping increases the parallelism of DB2 list prefetch I/Os. This parallelism exists with respect to the channel operations as well as the disk access.

If you plan to enable VSAM I/O striping, see *DB2 9 for z/OS Performance Topics,* SG24-7473-00.

17.3.5 Large volumes

With Extended Address Volume (EAV), which supports up to 1,1182,006 cylinders per volume, System z users can allocate the larger capacity volumes in the DS8000. From the DS8000 perspective, the capacity of a volume does not determine its performance. From the z/OS perspective, PAVs reduce or eliminate any additional enqueues that might originate from the increased I/O on the larger volumes. From the storage administration perspective, configurations with larger volumes are simpler to manage.

Measurements that are oriented to determine how large volumes can affect DB2 performance show that similar response times can be obtained by using larger volumes compared to using the smaller 3390-3 standard-size volumes. See 14.7.2, “Larger volume compared to smaller volume performance” on page 508 for a discussion.

17.3.6 Modified Indirect Data Address Words (MIDAWs)

MIDAWs, as described in 14.8.3, “MIDAW” on page 518, help improve performance when you access large chains of small blocks of data. To get this benefit, the dataset must be accessed by Media Manager. MIDAWs cut Fibre Channel connection (FICON) channel utilization for DB2 sequential I/O streams by half or more and improve the sequential throughput of extended format datasets by about 30%. A larger benefit is realized for the following datasets:

- Extended Format (EF) datasets
- Datasets that have small blocksizes (4K)

Examples of DB2 applications that benefit from MIDAWs are DB2 prefetch and DB2 utilities.

17.3.7 Adaptive Multi-stream Prefetching (AMP)

As described in 2.2, “Processor memory and cache” on page 27, AMP works in conjunction with DB2 sequential and dynamic prefetch. It works even better for dynamic prefetch than it does for most other sequential applications, because dynamic prefetch uses two prefetch engines. DB2 can explicitly request that the DS8000 prefetch from the disks. AMP adjusts the prefetch quantity that the DS8000 uses to prefetch tracks from the disks into the cache. AMP achieves this improvement by increasing the prefetch quantity sufficiently to meet the needs of the application.

As more data is prefetched, more disks are employed in parallel. Therefore, high throughput is achieved by employing parallelism at the disk level. In addition to enabling one sequential stream to be faster, AMP also reduces disk thrashing when there is disk contention.

17.3.8 DB2 burst write

When DB2 updates a record, it first updates the record that resides in the buffer pool. If the percentage of changed records in the buffer pool reaches the threshold defined in the vertical deferred write threshold (VDWQT), DB2 starts to flush and to write these updated records. These write activities to the disk subsystem are a huge burst of write I/Os, especially if the buffer pool is large and the VDWQT is high. This burst can cause a nonvolatile storage (NVS) saturation, because it is being flooded with too many writes. It shows up in the Resource Measurement Facility (RMF) cache report as DASD Fast Write Bypass (DFWBP or DFW Bypass). The term *bypass* is misleading. In the 3990/3390 era, when the NVS is full, the write I/O bypasses the NVS and the data is written directly to the disk drive module (DDM). In the DS8000, when the NVS is full, the write I/O is retried from the host until NVS space becomes available. So DFW Bypass must be interpreted as DFW Retry for the DS8000.

If RMF shows that the DFW Bypass divided by the total I/O Rate is greater than 1%, that is an indication of NVS saturation. If this NVS saturation happens, we suggest that you set the VDWQT to 0 or 1. Setting the VDWQT to 0 does not mean that every record update causes a write I/O to be triggered. Because, despite the 0% threshold set, the DB2 buffer pool still has 40 buffers set aside. The last recently updated 32 buffers are scheduled for write, which allows multiple successive updates to the same record to be not written out every time that record is updated. This method prevents multiple write I/Os to the same record on the disk subsystem. Lowering the VDWQT has a cost. In this case, it increases the CPU utilization, which shows up as a higher DBM1 SRM CPU time.

17.3.9 DB2 Adaptive List Prefetch

Prefetch is a mechanism for reading a set of pages, usually 32x4K pages, into the buffer pool with only one asynchronous I/O operation. High-performance FICON (zHPF) provides improvements for DB2 list prefetch processing. The FICON Express 8S features allow DB2 for z/OS to read 32x4K pages in a single zHPF channel program, which results in fewer I/Os and CPU. The limitation before was 22, which caused z/OS to split the pages into two I/Os.

17.3.10 Monitoring DS8000 performance

You can use RMF to monitor the performance of the DS8000. For a detailed description, see 14.10, “DS8000 performance monitoring tools” on page 521.

17.4 DS8000 DB2 UDB in an Open Systems environment

This section describes the performance considerations when you use DB2 UDB with the DS8000 in an Open Systems environment.

The information presented in this section is further discussed in detail in (and liberally borrowed from) the book, *IBM ESS and IBM DB2 UDB Working Together*, SG24-6262. Many of the concepts presented are applicable to the DS8000. We highly suggest this book. However, based on client solution experiences using SG24-6262, there are two corrections that we want to point out:

- In *IBM ESS and IBM DB2 UDB Working Together*, SG24-6262, section 3.2.2, “Balance workload across ESS resources,” suggests that a data layout policy must be established that allows partitions and containers within partitions to be spread evenly across ESS resources. It further suggests that you can choose either a horizontal mapping, in which every partition has containers on every available ESS rank, or a vertical mapping in which DB2 partitions are isolated to specific arrays, with containers spread evenly across those ranks. We now suggest the vertical mapping approach. The vertical isolated storage approach is typically easier to configure, manage, and diagnose if problems arise in production.
- Another data placement consideration suggests that it is important to place frequently accessed files in the space allocated from the middle of an array. This suggestion was an error in the original publication. The intent of the section was to discuss how the placement considerations commonly used with non-RAID older disk technology have less significance in ESS environments.

> **Vertical data mapping approach:** Based on experience, we now suggest a vertical data mapping approach (shared nothing between data partitions). We also want to emphasize that you must not try to micromanage data placement on storage.

17.4.1 DB2 UDB storage concepts

DB2 Universal Database (DB2 UDB) is the IBM object-relational database for UNIX, Linux, and Microsoft Windows operating environments.

The database object that maps the physical storage is the *tablespace*. Figure 17-2 illustrates how DB2 UDB is logically structured and how the tablespace maps the physical object.

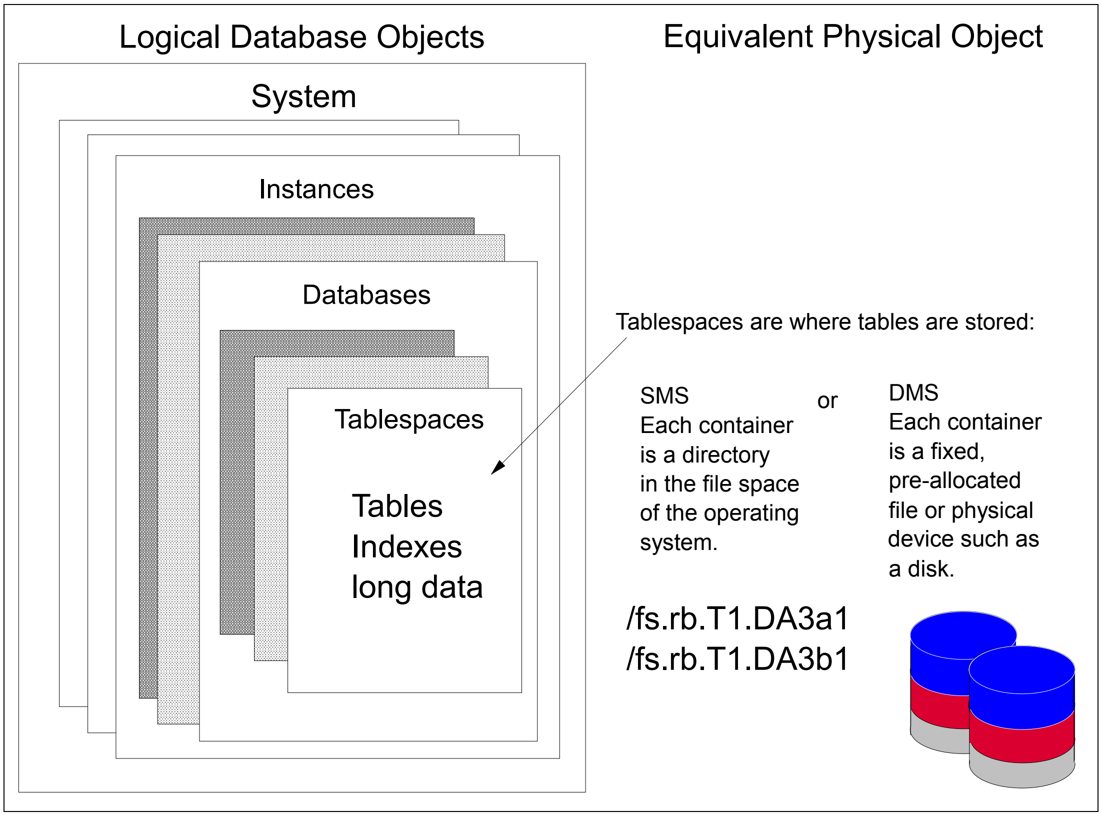


Figure 17-2 DB2 UDB logical structure

Instances

An *instance* is a logical database manager environment where databases are cataloged and configuration parameters are set. An instance is similar to an image of the actual database manager environment. You can have several instances of the database manager product on the same database server. You can use these instances to separate the development environment from the production environment, tune the database manager to a particular environment, and protect sensitive information from a particular group of users.

For database partitioning features (DPF) of the DB2 Enterprise Server Edition (ESE), all data partitions reside within a single instance.

Databases

A *relational database* structures data as a collection of database objects. The primary database object is the *table* (a defined number of columns and any number of rows). Each database includes a set of system catalog tables that describe the logical and physical structure of the data, configuration files that contain the parameter values allocated for the database, and recovery logs.

DB2 UDB allows multiple databases to be defined within a single database instance. Configuration parameters can also be set at the database level, so that you can tune, for example, memory usage and logging.

Database partitions

A *partition* number in DB2 UDB terminology is equivalent to a *data partition*. Databases with multiple data partitions and that reside on a symmetric multiprocessor (SMP) system are also called *multiple logical partition (MLN) databases*.

Partitions are identified by the physical system where they reside as well as by a logical port number with the physical system. The partition number, which can be 0 - 999, uniquely defines a partition. Partition numbers must be in ascending sequence (gaps in the sequence are allowed).

The configuration information of the database is stored in the *catalog partition*. The catalog partition is the partition from which you create the database.

Partitiongroups

A *partitiongroup* is a set of one or more database partitions. For non-partitioned implementations (all editions except for DPF), the partitiongroup is always made up of a single partition.

Partitioning map

When a partitiongroup is created, a *partitioning map* is associated to it. The partitioning map, in conjunction with the partitioning key and hashing algorithm, is used by the database manager to determine which database partition in the partitiongroup stores a specific row of data. Partitioning maps do not apply to non-partitioned databases.

Containers

A *container* is the way of defining where on the storage device the database objects are stored. Containers can be assigned from filesystems by specifying a directory. These containers are identified as PATH containers. Containers can also reference files that reside within a directory. These containers are identified as FILE containers, and a specific size must be identified. Containers can also reference raw devices. These containers are identified as DEVICE containers, and the device must exist on the system before the container can be used.

All containers must be unique across all databases; a container can belong to only one tablespace.

Tablespaces

A database is logically organized in *tablespaces*. A tablespace is a place to store tables. To spread a tablespace over one or more disk devices, you specify multiple containers.

For partitioned databases, the tablespaces reside in partitiongroups. In the create tablespace command execution, the containers themselves are assigned to a specific partition in the partitiongroup, thus maintaining the *shared nothing* character of DB2 UDB DPF.

Tablespaces can be either system-managed space (SMS) or data-managed space (DMS). For an SMS tablespace, each container is a directory in the filesystem, and the operating system file manager controls the storage space (Logical Volume Manager (LVM) for AIX). For a DMS tablespace, each container is either a fixed-size pre-allocated file, or a physical device, such as a disk (or in the case of the DS8000, a vpath), and the database manager controls the storage space.

There are three major types of user tablespaces: *regular* (index and data), *temporary*, and *long*. In addition to these user-defined tablespaces, DB2 requires that you define a system tablespace, the catalog tablespace. For partitioned database systems, this catalog tablespace resides on the catalog partition.

Tables, indexes, and LOBs

A *table* is a named data object that consists of a specific number of columns and unordered rows. Tables are uniquely identified units of storage maintained within a DB2 tablespace. They consist of a series of logically linked blocks of storage that have the same name. They also have a unique structure for storing information that relates to information about other tables.

When creating a table, you can choose to have certain objects, such as indexes and large object (LOB) data, stored separately from the rest of the table data, but you must define this table to a DMS tablespace.

Indexes are defined for a specific table and assist in the efficient retrieval of data to satisfy queries. They can also be used to assist in the clustering of data.

Large objects (LOBs) can be stored in columns of the table. These objects, although logically referenced as part of the table, can be stored in their own tablespace when the base table is defined to a DMS tablespace. This approach allows for more efficient access of both the LOB data and the related table data.

Pages

Data is transferred to and from devices in discrete blocks that are buffered in memory. These discrete blocks are called *pages*, and the memory reserved to buffer a page transfer is called an *I/O buffer*. DB2 UDB supports various page sizes, including 4 k, 8 k, 16 k, and 32 k.

When an application accesses data randomly, the page size determines the amount of data transferred. This size corresponds to the size of the data transfer request to the DS8000, which is sometimes referred to as the *physical record*.

Sequential read patterns can also influence the page size that is selected. Larger page sizes for workloads with sequential read patterns can enhance performance by reducing the number of I/Os.

Extents

An *extent* is a unit of space allocation within a container of a tablespace for a single tablespace object. This allocation consists of multiple pages. The extent size (number of pages) for an object is set when the tablespace is created:

- An extent is a group of consecutive pages defined to the database.
- The data in the tablespaces is striped by extent across all the containers in the system.

Buffer pools

A *buffer pool* is main memory allocated on the host processor to cache table and index data pages as they are read from disk or modified. The purpose of the buffer pool is to improve system performance. Data can be accessed much faster from memory than from disk; therefore, the fewer times that the database manager needs to read from or write to disk (I/O), the better the performance. Multiple buffer pools can be created.

DB2 prefetch (reads)

Prefetching is a technique for anticipating data needs and *reading ahead* from storage in large blocks. By transferring data in larger blocks, fewer system resources are used and less time is required.

Sequential prefetch reads consecutive pages into the buffer pool before they are needed by DB2. List prefetches are more complex. In this case, the DB2 optimizer optimizes the retrieval of randomly located data.

The amount of data that is prefetched determines the amount of parallel I/O activity. Ordinarily, the database administrator defines a prefetch value large enough to allow parallel use of all of the available containers.

Consider the following example:

- A tablespace is defined with a page size of 16 KB that uses raw DMS.
- The tablespace is defined across four containers. Each container resides on a separate logical device, and the logical devices are on different DS8000 ranks.
- The extent size is defined as 16 pages (or 256 KB).
- The prefetch value is specified as 64 pages (number of containers x extent size).
- A user submits a query that results in a tablespace scan, which then results in DB2 performing a prefetch operation.

The following actions happen:

- DB2, recognizing that this prefetch request for 64 pages (1 MB) evenly spans four containers, makes four parallel I/O requests, one against each of those containers. The request size to each container is 16 pages (or 256 KB).
- After receiving several of these requests, the DS8000 recognizes that these DB2 prefetch requests are arriving as sequential accesses, causing the DS8000 sequential prefetch to take effect. The sequential prefetch results in all of the disks in all four DS8000 ranks to operate concurrently, staging data to the DS8000 cache, to satisfy the DB2 prefetch operations.

Page cleaners

Page cleaners are present to make room in the buffer pool before prefetchers read pages on disk storage and move them into the buffer pool. For example, if a large amount of data is updated in a table, many data pages in the buffer pool might be updated but not written into disk storage (these pages are called *dirty pages*). Because prefetchers cannot place fetched data pages onto the dirty pages in the buffer pool, these dirty pages must be flushed to disk storage and become *clean* pages so that prefetchers can place fetched data pages from disk storage.

Logs

Changes to data pages in the buffer pool are logged. Agent processes, which are updating a data record in the database, update the associated page in the buffer pool and write a log record to a log buffer. The written log records in the log buffer are flushed into the log files asynchronously by the logger.

To optimize performance, the updated data pages in the buffer pool and the log records in the log buffer are *not* written to disk immediately. The updated data pages in the buffer pool are written to disk by page cleaners and the log records in the log buffer are written to disk by the logger.

The logger and the buffer pool manager cooperate and ensure that the updated data page is not written to disk storage before its associated log record is written to the log. This behavior ensures that the database manager can obtain enough information from the log to recover and protect a database from being left in an inconsistent state when the database crashes as a result of an event, such as a power failure.

Parallel operations

DB2 UDB extensively uses *parallelism* to optimize performance when accessing a database. DB2 supports several types of parallelism, including query and I/O parallelism.

Query parallelism

There are two dimensions of query parallelism: *inter-query parallelism* and *intra-query parallelism*. Inter-query parallelism refers to the ability of multiple applications to query a database at the same time. Each query executes independently of the other queries, but they are all executed at the same time. Intra-query parallelism refers to the simultaneous processing of parts of a single query, by using *intra-partition parallelism*, *inter-partition parallelism*, or both:

- Intra-partition parallelism subdivides what is considered a single database operation, such as index creation, database loading, or SQL queries, into multiple parts, many or all of which can be run in parallel within a single *database partition.*
- Inter-partition parallelism subdivides what is considered a single database operation, such as index creation, database loading, or SQL queries, into multiple parts, many or all of which can be run in parallel across multiple partitions of a partitioned database on one machine or on multiple machines. Inter-partition parallelism applies to DPF only.

I/O parallelism

When there are multiple *containers* for a tablespace, the database manager can use parallel I/O. *Parallel I/O* refers to the process of writing to, or reading from, two or more I/O devices simultaneously. Parallel I/O can result in significant improvements in throughput.

DB2 implements a form of data striping by spreading the data in a tablespace across multiple *containers.* In storage terminology, the part of a stripe that is on a single device is a *strip*. The DB2 term for strip is *extent.* If your tablespace has three containers, DB2 writes one extent to container 0, the next extent to container 1, the next extent to container 2, and then back to container 0. The stripe *width* (a generic term not often used in DB2 literature) is equal to the number of containers, or three in this case.

Extent sizes are normally measured in numbers of DB2 pages.

Containers for a tablespace are ordinarily placed on separate physical disks, allowing work to be spread across those disks, and allowing disks to operate in parallel. Because the DS8000 logical disks are striped across the rank, the database administrator can allocate DB2 containers on separate logical disks that reside on separate DS8000 arrays. This approach takes advantage of the parallelism both in DB2 and in the DS8000. For example, four DB2 containers that reside on four DS8000 logical disks on four different 7+P ranks have data spread across 32 physical disks.

17.5 DB2 UDB with DS8000 performance recommendations

When using a DS8000, the following suggestions are useful when planning for good DB2 UDB performance.

For a more detailed and accurate approach that considers the particularities of your DB2 UDB environment, contact your IBM representative, who can assist you with the DS8000 capacity and configuration planning.

17.5.1 Know where your data resides

Know where your data resides. Understand how DB2 containers map to the DS8000 logical disks, and how those logical disks are distributed across the DS8000 ranks. Spread DB2 data across as many DS8000 ranks as possible.

If you want optimal performance from the DS8000, do not treat it like a black box. Establish a storage allocation policy that allocates data by using several DS8000 ranks. Understand how DB2 tables map to underlying logical disks, and how the logical disks are allocated across the DS8000 ranks. One way of making this process easier to manage is to maintain a modest number of DS8000 logical disks.

17.5.2 Balance workload across DS8000 resources

Balance the workload across the DS8000 resources. Establish a storage allocation policy that allows balanced workload activity across RAID arrays. You can take advantage of the inherent balanced activity and parallelism within DB2, spreading the work for DB2 partitions and containers across the DS8000 arrays. This spreading applies to both OLTP and DSS workload types. If you spread the work and plan sufficient resource, many of the other decisions are secondary.

Consider the following general recommendations:

- DB2 query parallelism allows workload to be balanced across CPUs and, if DB2 Universal Database Partitioning Feature (DPF) is installed, across data partitions.
- DB2 I/O parallelism allows workload to be balanced across containers.

As a result, you can balance activity across the DS8000 resources by following these rules:

- Span the DS8000 storage units.
- Span ranks (RAID arrays) within a storage unit.
- Engage as many arrays as possible.

Figure 17-3 on page 574 illustrates this technique for a single tablespace that consists of eight containers.

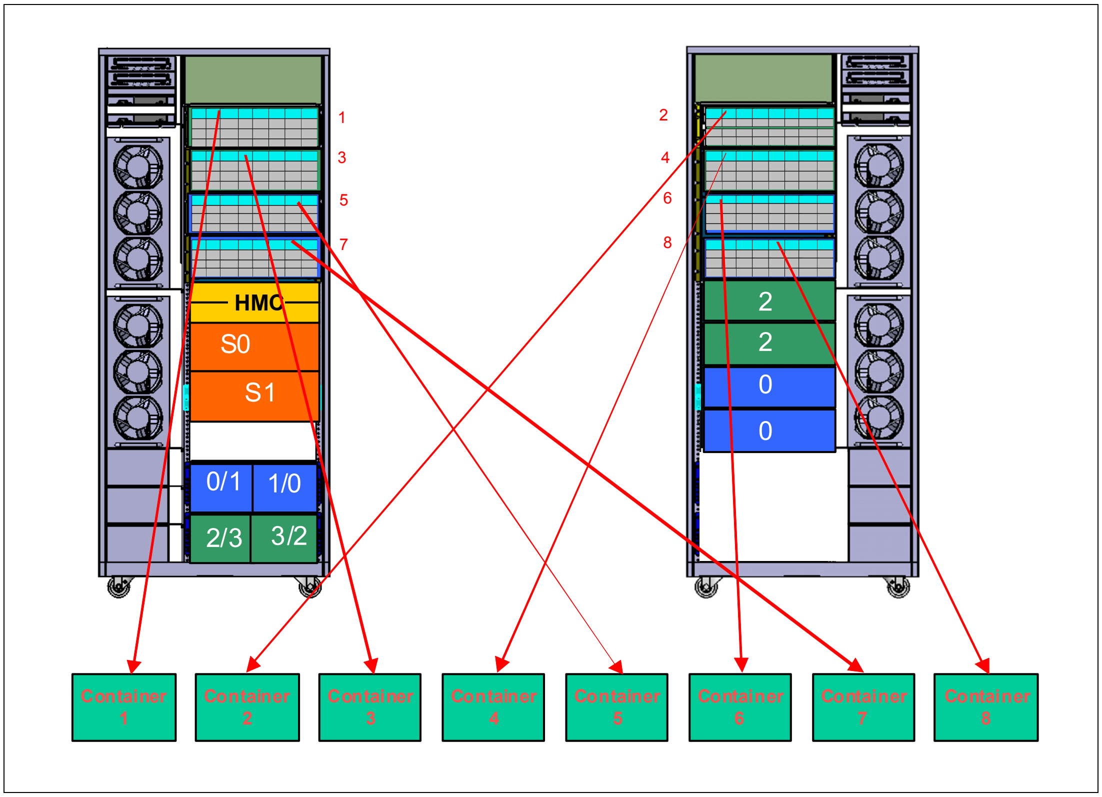


Figure 17-3 Allocating DB2 containers by using a "spread your data" approach

Also, remember these considerations:

- You can intermix data, indexes, and temp spaces on the DS8000 ranks. Your I/O activity is more evenly spread, and thus, you avoid the skew effect, which you otherwise see if the components are isolated.
- For DPF systems, establish a policy that allows partitions and containers within partitions to be spread evenly across the DS8000 resources. Choose a vertical mapping, in which DB2 partitions are isolated to specific arrays, with containers spread evenly across those arrays.

17.5.3 Use DB2 to stripe across containers

Use the inherent striping of DB2, placing containers for a tablespace on separate DS8000 logical disks on different DS8000 ranks. This striping eliminates the need for using underlying operating system or logical volume manager striping.

Look again at Figure 17-3. In this case, we stripe across arrays, disk adapters, clusters, and DS8000s, which can all be done by using the striping capabilities of the DB2 container and *shared nothing* concept. This approach eliminates the need to employ AIX logical volume striping.

17.5.4 Selecting DB2 logical sizes

The three settings in a DB2 system that primarily affect the movement of data to and from the disk subsystem work together:

- Page size
- Extent size

- Prefetch size

Page size

Page sizes are defined for each tablespace. There are four supported page sizes: 4 K, 8 K, 16 K, and 32 K. The following factors affect the choice of page size:

- The maximum number of records per page is 255. To avoid wasting space on a page, do not make page size greater than 255 times the row size plus the page overhead.
- The maximum size of a tablespace is proportional to the page size of its tablespace. In SMS, the data and index objects of a table have limits, as shown in Table 17-1. In DMS, these limits apply at the tablespace level.

Table 17-1 Page size relative to tablespace size

Page size	Maximum data/index object size
4 KB	64 GB
8 KB	128 GB
16 KB	256 GB
32 KB	512 GB

Select a page size that can accommodate the total expected growth requirements of the objects in the tablespace.

For OLTP applications that perform random row read and write operations, a smaller page size is preferable, because it wastes less buffer pool space with unwanted rows. For DSS applications that access large numbers of consecutive rows at a time, a larger page size is better, because it reduces the number of I/O requests that are required to read a specific number of rows.

Tip: Experience indicates that page size can be dictated to a certain degree by the type of workload. For pure OLTP workloads, we suggest a 4 KB page size. For a pure DSS workload, we suggest a 32 KB page size. For a mixture of OLTP and DSS workload characteristics, we suggest either an 8 KB page size or a 16 KB page size.

Extent size

If you want to stripe across multiple arrays in your DS8000, assign a LUN from each rank to be used as a DB2 container. During writes, DB2 writes one extent to the first container and the next extent to the second container until all eight containers are addressed before cycling back to the first container. DB2 stripes across containers at the tablespace level.

Because the DS8000 stripes at a fairly fine granularity (256 KB), selecting multiples of 256 KB for the extent size ensures that multiple DS8000 disks are used within a rank when a DB2 prefetch occurs. However, keep your extent size below 1 MB.

I/O performance is fairly insensitive to the selection of extent sizes, mostly because the DS8000 employs sequential detection and prefetch. For example, even if you select an extent size, such as 128 KB, which is smaller than the full array width (it accesses only four disks in the array), the DS8000 sequential prefetch keeps the other disks in the array busy.

Prefetch size

The tablespace prefetch size determines the degree to which separate containers can operate in parallel.

Although larger prefetch values might enhance throughput of individual queries, mixed applications generally operate best with moderate-sized prefetch and extent parameters. You want to engage as many arrays as possible in your prefetch to maximize throughput.

Prefetch size is tunable. We mean that prefetch size can be altered after the tablespace is defined and data loaded, which is not true for extents and page sizes that are set at tablespace creation time and cannot be altered without redefining the tablespace and reloading the data.

> **Tip:** The prefetch size must be set so that as many arrays as wanted can be working on behalf of the prefetch request. For other than the DS8000, the general suggestion is to calculate prefetch size to be equal to a multiple of the extent size times the number of containers in your tablespace. For the DS8000, you can work with a multiple of the extent size times the number of arrays underlying your tablespace.

17.5.5 Selecting the DS8000 logical disk sizes

The DS8000 gives you great flexibility when it comes to disk allocation. This flexibility is helpful, for example, when you need to attach multiple hosts. However, this flexibility can present a challenge as you plan for future requirements.

The DS8000 supports a high degree of parallelism and concurrency on a single logical disk. As a result, a single logical disk the size of an entire array achieves the same performance as many smaller logical disks. However, you must consider how logical disk size affects both the host I/O operations and the complexity of your systems administration.

Smaller logical disks provide more granularity, with their associated benefits. But smaller logical disks also increase the number of logical disks seen by the operating system. Select a DS8000 logical disk size that allows for granularity and growth without proliferating the number of logical disks.

Take into account your container size and how the containers map to AIX logical volumes and DS8000 logical disks. In the simplest situation, the container, the AIX logical volume, and the DS8000 logical disk are the same size.

> **Tip:** Try to strike a reasonable balance between flexibility and manageability for your needs. Our suggestion is that you create no fewer than two logical disks in an array, and the minimum logical disk size needs to be 16 GB. Unless you have a compelling reason, standardize a unique logical disk size throughout the DS8000.

Smaller logical disk sizes have the following advantages and disadvantages:

- Advantages of smaller size logical disks:
  - Easier to allocate storage for different applications and hosts.
  - Greater flexibility in performance reporting.
- Disadvantages of smaller size logical disks

  Small logical disk sizes can contribute to the proliferation of logical disks, particularly in SAN environments and large configurations. Administration gets complex and confusing.

Larger logical disk sizes have the following advantages and disadvantages:

- Advantages of larger size logical disks:
  - Simplifies understanding of how data maps to arrays.

  - Reduces the number of resources used by the operating system.
  - Storage administration is simpler, more efficient, and provides fewer chances for mistakes.
- Disadvantages of larger size logical disks

  Less granular storage administration, resulting in less flexibility in storage allocation.

Examples

Assume a 6+P array with 146 GB disk drives. You want to allocate disk space on your 16-array DS8000 as flexibly as possible. You can carve each of the 16 arrays into 32 GB logical disks or logical unit numbers (LUNs), resulting in 27 logical disks per array (with a little left over). This design yields a total of 16 x 27 = 432 LUNs. Then, you can implement 4-way multipathing, which in turn makes 4 x 432 = 1728 hdisks visible to the operating system.

This approach creates an administratively complex situation, and, at every reboot, the operating system queries each of those 1728 disks. Reboots might take a long time.

Alternatively, you create just 16 large logical disks. With multipathing and attachment of four Fibre Channel ports, you have 4 x 16 = 128 hdisks visible to the operating system. Although this number is large, it is more manageable, and reboots are much faster. After overcoming that problem, you can then use the operating system logical volume manager (LVM) to carve this space into smaller pieces for use.

There are problems with this large logical disk approach as well, however. If the DS8000 is connected to multiple hosts or it is on a SAN, disk allocation options are limited when you have so few logical disks. You must allocate entire arrays to a specific host, and if you want to add additional space, you must add it in array-size increments.

17.5.6 Multipathing

Use the DS8000 multipathing along with DB2 striping to ensure the balanced use of Fibre Channel paths.

Multipathing is the hardware and software support that provides multiple avenues of access to your data from the host computer. You need to provide at least two Fibre Channel paths from the host computer to the DS8000. Paths are defined by the number of host adapters on the DS8000 that service the LUNs of a certain host system, the number of Fibre Channel host bus adapters on the host system, and the SAN zoning configuration. The total number of paths includes consideration for the throughput requirements of the host system. If the host system requires more than (2 x 200) 400 MBps throughput, two host bus adapters are not adequate.

The DS8000 multipathing requires the installation of multipathing software. For AIX, you have two choices: Subsystem Device Driver Path Control Module (SDDPCM) or the IBM Subsystem Device Driver (SDD). For AIX, we suggest SDDPCM. We describe these products in Chapter 9, “Performance considerations for UNIX servers” on page 327 and Chapter 8, “Host attachment” on page 311.

There are several benefits you receive from using multipathing: higher availability, higher bandwidth, and easier management. A high availability implementation is one in which your application can still access data by using an alternate resource if a component fails. Easier performance management means that the multipathing software automatically balances the workload across the paths.

17.6 IMS in a z/OS environment

This section discusses IMS, its logging, and the performance considerations when IMS datasets are placed on the DS8000.

17.6.1 IMS overview

IMS consists of three components: the Transaction Manager (TM) component, the Database Manager (DB) component, and a set of system services that provides common services to the other two components.

IMS Transaction Manager

IMS Transaction Manager provides a network with access to the applications that run under IMS. The users can be people at terminals or workstations, or other application programs.

IMS Database Manager

IMS Database Manager provides a central point of control and access to the data that is processed by IMS applications. The Database Manager component of IMS supports databases that use the hierarchical database model of IMS. It provides access to the databases from the applications that run under the IMS Transaction Manager, the CICS® transaction monitor, and z/OS batch jobs.

IMS Database Manager provides functions for preserving the integrity of databases and maintaining the databases. It allows multiple tasks to access and update the data, while ensuring the integrity of the data. It also provides functions for reorganizing and restructuring the databases.

The IMS databases are organized internally by using a number of IMS internal database organization access methods. The database data is stored on disk storage by using the normal operating system access methods.

IMS system services

There are many functions that are common to the Database Manager and Transaction Manager:

- Restart and recovery of the IMS subsystem failures
- Security: Controlling access to IMS resources
- Managing the application programs: Dispatching work, loading application programs, and providing locking services
- Providing diagnostic and performance information
- Providing facilities for the operation of the IMS subsystem
- Providing an interface to other z/OS subsystems that interface with the IMS applications

17.6.2 IMS logging

IMS logging is one of the most write-intensive operations in a database environment.

During IMS execution, all information necessary to restart the system in the event of a failure is recorded on a system log dataset. The IMS logs are made up of the following information.

IMS log buffers

The log buffers are used to write the information that needs to be logged.

Online log datasets (OLDS)

The OLDS are datasets that contain all the log records required for restart and recovery. These datasets must be pre-allocated on DASD and hold the log records until they are archived.

The OLDS are made of multiple datasets that are used in a wraparound manner. At least three datasets must be allocated for the OLDS to allow IMS to start, while an upper limit of 100 datasets is supported.

Only complete log buffers are written to OLDS to enhance performance. If any incomplete buffers need to be written out, they are written to the write ahead datasets (WADS).

Write ahead datasets (WADS)

The WADS is a small direct-access dataset that contains a copy of committed log records that are in OLDS buffers, but they are not written to OLDS yet.

When IMS processing requires writing a partially filled OLDS buffer, a portion of the buffer is written to the WADS. If IMS or the system fails, the log data in the WADS is used to terminate the OLDS, which can be done as part of an emergency restart, or as an option on the IMS Log Recovery Utility.

The WADS space is continually reused after the appropriate log data is written to the OLDS. This dataset is required for all IMS systems, and must be pre-allocated and formatted at IMS startup when first used.

When using a DS8000 with storage pool striping, define the WADS volumes as 3390-Mod.1 and allocate them consecutively so that they are allocated to different ranks.

System log datasets (SLDS)

The SLDS is created by the IMS log archive utility, preferably after every OLDS switch. It is placed on tape, but it can reside on disk. The SLDS can contain the data from one or more OLDS datasets.

Recovery log datasets (RLDS)

When the IMS log archive utility is run, the user can request creation of an output dataset that contains all of the log records needed for database recovery. This dataset is the RLDS and also known to DBRC. The RLDS is optional.

17.7 DS8000 considerations for IMS

By using the DS8000 in an IMS environment, the following benefits are possible:

- IMS takes advantage of the PAV function that allows multiple concurrent I/Os to the same volume at the same time from applications that run on a z/OS system image.
- Less disk contention occurs when accessing the same volumes from different systems in an IMS data sharing group and using the Multiple Allegiance function.
- Higher bandwidth on the DS8000 allows higher I/O rates to be handled by the disk subsystem, thus allowing for higher application transaction rates.

17.8 IMS with DS8000 performance recommendations

When using the DS8000, the following generic recommendations are useful when planning for good IMS performance.

17.8.1 Know where your data resides

IMS storage administration can be done by using SMS to simplify disk use and control, or without using SMS. In both cases, it is important that you know where your data resides.

If you want optimal performance from the DS8000, do not treat it like a "black box." Understand how your IMS datasets map to underlying volumes, and how the volumes map to RAID arrays.

17.8.2 Balance workload across DS8000 resources

You can balance workload activity across the DS8000 resources by performing these tasks:

- Spreading IMS data across the DS8000s if practical
- Spreading IMS data across servers in each DS8000
- Spreading IMS data across the DS8000 device adapters
- Spreading IMS data across as many extent pools/ranks as practical

You can intermix IMS databases and log datasets on the DS8000 ranks. The overall I/O activity is more evenly spread, and I/O skews are avoided.

17.8.3 Large volumes

With large volume support, which supports up to 65520 cylinders per volume, System z users can allocate the larger capacity volumes in the DS8000. From the DS8000 perspective, the capacity of a volume does not determine its performance. From the z/OS perspective, PAVs reduce or eliminate any additional enqueues that might originate from the increased I/O on the larger volumes. From the storage administration perspective, configurations with larger volumes are simpler to manage.

Measurements to determine how large volumes can affect IMS performance show that similar response times can be obtained when using larger volumes as when using smaller 3390-3 standard-size volumes.

Figure 17-4 on page 581 illustrates the device response times when using thirty-two 3390-3 volumes compared to four large 3390-27 volumes on an ESS-F20 that uses FICON channels. Even though we performed the benchmark on an ESS-F20, the results are similar on the DS8000. The results show that with the larger volumes, the response times are similar to the standard size 3390-3 volumes.

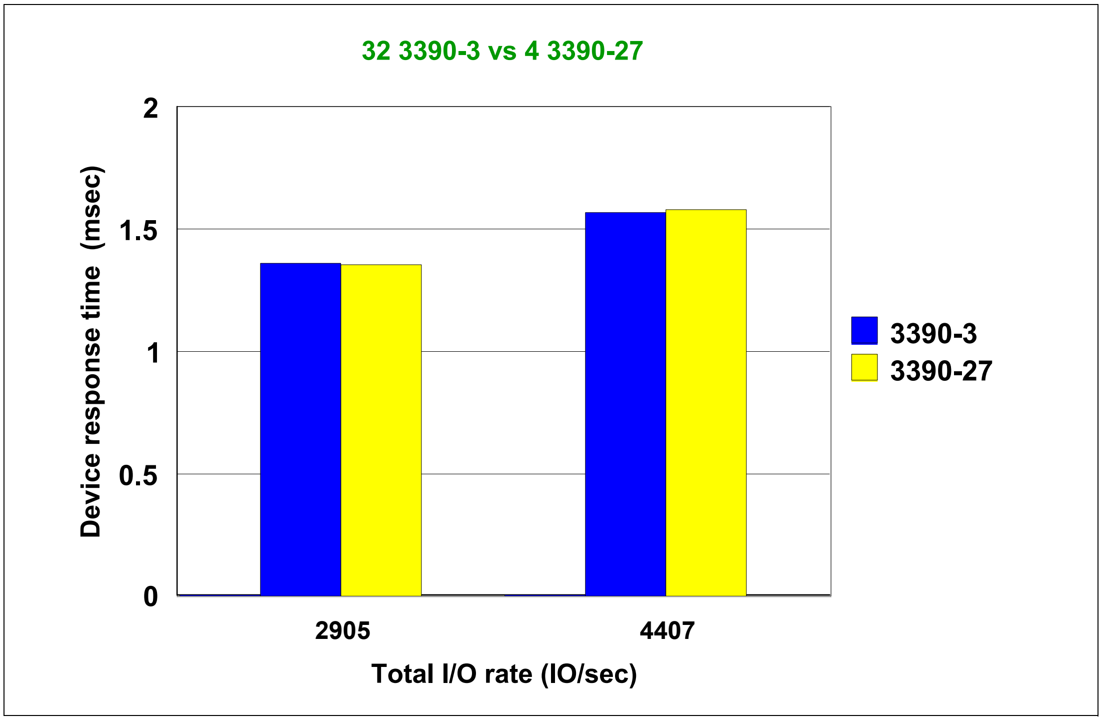


Figure 17-4 IMS large volume performance

17.8.4 Monitoring DS8000 performance

You can use RMF to monitor the performance of the DS8000. For a detailed description, see 14.10, “DS8000 performance monitoring tools” on page 521.

17.9 Oracle with DS8000 performance considerations

This section describes Oracle databases and some preferred practices with the DS8000 to achieve better performance results.

Also, this section is intended to focus on Oracle I/O characteristics. Some memory or CPU considerations are needed, but we understand that these considerations are done at the appropriate level according to your system specifications and planning.

Reviewing the following considerations can help you understand the Oracle I/O demand and your DS8800/DS8700 planning for its use.

17.9.1 Architecture overview

First, we review the components of an Oracle database for I/O considerations. The basic components of an Oracle server are a database and an instance. A database is composed of a set of datafiles that consists of data, redo logs, control files, and archive log files.

Although the instance is an important part of the Oracle components, our focus is on the datafiles. OLTP workloads can benefit from SSDs combined with Easy Tier automatic mode management to optimize performance. Furthermore, you also need to discuss segregation and resource-sharing aspects when performing separate levels of isolation on the storage for different components. Typically, in an Oracle database, you separate redo logs and archive logs from data and indexes. The redo logs and archives are known for performing intensive read/write workloads.

In a database, the disk part is considered the slowest component in the whole infrastructure. You must plan to avoid reconfigurations and time-consuming performance problem investigations when future problems, such as bottlenecks, might occur. However, as with all I/O subsystems, good planning and data layout can make the difference between having excellent I/O throughput and application performance, and having poor I/O throughput, high I/O response times, and correspondingly poor application performance.

In many cases, I/O performance problems can be traced directly to “hot” files that cause a bottleneck on some critical component, for example, a single physical disk. This problem can occur even when the overall I/O subsystem is fairly lightly loaded. When bottlenecks occur, storage or data base administrators might need to identify and manually relocate the high activity data files that contributed to the bottleneck condition. This problem solving tends to be a resource-intensive and often frustrating task. As the workload content changes with the daily operations of normal business cycles, for example, hour by hour through the business day or day by day through the accounting period, bottlenecks can mysteriously appear and disappear or migrate over time from one datafile or device to another.

17.9.2 DS8000 performance consideration

Generally, I/O (and therefore application) performance is best when the I/O activity is evenly spread across the entire I/O subsystem and, if available, the appropriate storage classes. Easy Tier automatic mode management, providing automatic intra-tier and cross-tier performance optimization, is an option and can make the management of datafiles easier. Even in homogeneous extent pools, you can benefit from Easy Tier automatic mode intra-tier rebalancing (auto-rebalance) to optimize the workload distribution across ranks, reducing workload skew and avoiding hot spots. The goal for balancing I/O activity across ranks, adapters, and different storage tiers can be easily achieved by using Easy Tier automatic mode even in shared environments. In addition, the prioritization of important database workloads to meet their quality of service (QoS) requirements when they share storage resources with less important workloads can be managed easily by using the DS8000 I/O Priority Manager.

In 4.7.1, “RAID-level performance considerations” on page 103, we reviewed the RAID levels and their performance aspects. It is important to discuss the RAID levels, because some datafiles can benefit from certain RAID levels, depending on their workload profile as shown in Figure 4-6 on page 109. However, advanced storage architectures, for example, cache and advanced cache algorithms, or even multi-tier configurations with Easy Tier automatic management can make RAID level considerations less important.

For instance, with 15K rpm Enterprise disks and a significant amount of cache available on the storage system, some environments might have similar performance on RAID 10 and RAID 5, although mostly workloads with a high percentage of random write activity and high I/O access densities generally benefit from RAID 10. RAID 10 benefits clients in single-tier pools. RAID 10 takes advantage of Easy Tier automatic intra-tier performance management (auto-rebalance) and constantly optimizes data placement across ranks based on rank utilization in the extent pool.

However, by using hybrid pools with SSDs and Easy Tier automode cross-tier performance management that promotes the hot extents to SSDs on a subvolume level, you can additionally boost database performance and take advantage of the capacity on the solid-state drive (SSD) tier and automatically adapt to changing workload conditions.

You might consider striping on one level only (storage system or host/application-level), depending on your needs. The use of host-level or application-level striping might be counterproductive when using Easy Tier in multi-tier extent pools, because striping dilutes the workload skew and can reduce the effectiveness of Easy Tier.

On previous DS8300/DS8100 systems, you benefited from using storage pool striping (rotate extents) and striping on the storage level. You can create your redo logs and spread them across as many extent pools and ranks as possible to avoid contention. On a DS8800/8700 system with Easy Tier, data placement and workload spreading in extent pools is automatic, even across different storage tiers.

You still can divide your workload across your planned extent pools (hybrid or homogeneous) and consider segregation on the storage level by using different storage classes or RAID levels or by separating tablespaces from logs across different extent pools with regard to failure boundary considerations.

17.9.3 Oracle for AIX

With Easy Tier, the storage system optimizes workload spreading and data placement across and within storage tiers, based on workload access patterns and by constantly adapting to workload changes over time. The use of host-level or application-level striping can dilute the workload skew and reduce the effectiveness of Easy Tier in multi-tier configurations.

However, if you consider striping on the AIX LVM level or the database level, for example, Oracle Automatic Storage Management (ASM), you need to consider the best approaches possible if you use it with Easy Tier and multi-tier configurations. Keep your physical partition (PP) size or stripe size at a high value to have enough skew with Easy Tier to efficiently promote hot extents.

AIX LVM also features different mount options if you consider logical filesystems instead of raw devices. For the AIX LVM options, see 9.2.4, “IBM Logical Volume Manager” on page 343. Next, we show you different mount options for logical filesystems as the preferred practices with Oracle databases. These mount options can be used for filesystems, as described in “Mount options” on page 340:

- Direct I/O (DIO):
  - Data is transferred directly from the disk to the application buffer. It bypasses the file buffer cache and avoids double caching (filesystem cache + Oracle System Global Area (SGA)).
  - Emulates a raw device implementation.
- Concurrent I/O (CIO):
  - Implicit use of DIO.
  - No inode locking: Multiple threads can perform reads and writes on the same file at the same time.
  - Performance achieved by using CIO is comparable to raw devices.
  - Avoid double caching: Some data is already cached in the Application layer (SGA).
  - Provides faster access to the back-end disk and reduces the CPU utilization.
  - Disables the inode-lock to allow several threads to read and write the same file (CIO only).
  - Because data transfer is bypassing the AIX buffer cache, Journaled File System 2 (JFS2) prefetching and write-behind cannot be used. These functions can be handled by Oracle.

When using DIO or CIO, IO requests made by Oracle must be aligned with the JFS2 blocksize to avoid a demoted I/O (returns to normal I/O after a DIO failure).

Comparing DIO, CIO, and raw devices, CIO is likely to perform in a similar manner to raw devices, and raw is likely to show the best results. Additionally, when using JFS2, consider using the INLINE log for filesystems so that it can have the log striped and not be just placed in a single AIX physical partition (PP).

Other options that are supported by Oracle include the Asynchronous I/O (AIO), General Parallel File System (GPFS), ASM, and raw device formats:

- AIO:
  - Allows multiple requests to be sent without having to wait until the disk subsystem completes the physical I/O.
  - Use of asynchronous I/O is advised no matter what type filesystem and mount option you implement (JFS, JFS2, CIO, or DIO).
- GPFS:
  - When implementing Oracle Real Application Clusters (RAC) environments, many clients prefer to use a clustered filesystem. GPFS is the IBM clustered filesystem offering for Oracle RAC on AIX. Other Oracle files, such as the ORACLE_HOME executable libraries, archive log directories that do not need to be shared between instances. These files can either be placed on local disk, for example, by using JFS2 filesystems, or a single copy can be shared across the RAC cluster by using GPFS. GPFS can provide administrative advantages, such as maintaining only one physical ORACLE_HOME, which ensures that archive logs are always available (even when nodes are down) when recoveries are required.
  - When used with Oracle, GPFS automatically stripes files across all of the available disks within a GPFS using a 1 MiB stripe size. Therefore, GPFS provides data and I/O distribution characteristics that are similar to PP spreading, LVM (large granularity) striping, and ASM course-grained striping techniques.
- ASM:
  - ASM is a database filesystem that provides cluster filesystem and volume manager capabilities. ASM is an alternative to conventional filesystem and LVM functions.
  - Integrated into the Oracle database at no additional cost for single or RAC databases.
  - With ASM, the management of Oracle data files is the same for the DBA on all platforms (UNIX, Linux, or Windows).
  - Datafiles are striped across all ASM disks, and I/O is spread evenly to prevent hot spots and maximize performance.
  - Online add/drop of disk devices with automatic online redistribution of data.
  - An ASM-managed database has approximately the same performance as a database that is implemented in raw devices.

18

Copy Services performance

In this chapter, we describe the performance-related considerations when implementing Copy Services for the DS8000. Copy Services is a collection of functions provided by the DS8000 that facilitate disaster recovery, data migration, and data duplication functions. Copy Services are optional licensed features that run on the DS8000 Storage Facility Image and support all attached host systems.

We review the Copy Services functions and give suggestions about preferred practices for configuration and performance:

- Copy Services introduction
- FlashCopy
- Metro Mirror
- Global Copy
- Global Mirror
- z/OS Global Mirror
- Metro/Global Mirror

18.1 Copy Services introduction

The DS8000 series offers an array of advanced functions for data backup, remote mirroring, and disaster recovery. The DS8000 advanced 2-site and 3-site business continuity capabilities provide synchronous and asynchronous data replication for mission-critical applications, giving availability when needed during both planned and unplanned system outages.

There are two primary types of Copy Services functions: *Point-in-Time Copy* and *Remote Mirror and Copy*. Generally, the Point-in-Time Copy functions are used for data duplication, and the Remote Mirror and Copy functions are used for data migration and disaster recovery. Table 18-1 is a reference chart for the Copy Services. The following copy operations are available for each function:

- Point-in-Time Copy:
  - FlashCopy
  - FlashCopy SE
- Remote Mirror and Copy:
  - Global Mirror
  - Metro Mirror
  - Global Copy
  - Three-site Metro/Global Mirror with Incremental Resync
- z/OS Global Mirror, previously known as Extended Remote Copy (XRC)
- z/OS Metro/Global Mirror across three sites with Incremental Resync

Table 18-1 Reference chart for DS Copy Services on DS8000

DS8000 function	ESS800 Version 2 function	Former name
FlashCopy	FlashCopy	FlashCopy
FlashCopy SE	N/A	N/A
Global Mirror	Global Mirror	Asynchronous PPRC
Metro Mirror	Metro Mirror	Synchronous PPRC
Global Copy	Global Copy	PPRC Extended Distance
z/OS Global Mirror	z/OS Global Mirror	Extended Remote Copy (XRC)
z/OS Metro/Global Mirror	z/OS Metro/Global Mirror	Three-site solution that uses Sync PPRC and XRC
Metro/Global Mirror	Metro/Global Mirror	Two or 3-site Asynchronous Cascading PPRC

See the Interoperability Matrixes for the DS8000 and ESS to confirm which products are supported on a particular disk subsystem.

Copy Services management interfaces

Copy Services functions can be managed through a number of network and in-band interfaces:

- DS command-line interface (CLI)
- DS Storage Manager (SM)
- DS Open API
- Tivoli Storage Productivity Center for Replication

In addition to these methods, there are several possible interfaces available specifically for System z users to manage the DS8000 Copy Services relationships. Table 18-2 lists these tools:

- TSO
- ICKDSF
- DFSMSdss
- The ANTRQST macro
- Native TPF commands (for z/TPF only)

Table 18-2 Copy Services management tools

	Runs on z/OS	Runs on Open Systems Server	Manages count key data (CKD)	Manages fixed block (FB) data
TSO	Yes	No	Yes	Yes[1]
ANTRQST	Yes	No	Yes	Yes[1]
ICKDSF	Yes	No	Yes	No
DSCLI	No	Yes	Yes	Yes
TPC for Replication	Yes	Yes	Yes	Yes
GDPS®	Yes	No	Yes	Yes[1]

1. A CKD unit address (and host unit control block (UCB)) must be defined in the same DS8000 server against which host I/O can be issued to manage Open Systems FB logical unit numbers (LUNs).

For detailed information about the DS8000 Copy Services, see the following IBM Redbooks publications:

- *DS8000 Copy Services for IBM System z*, SG24-6787
- *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788

18.2 FlashCopy

FlashCopy can help reduce or eliminate planned outages for critical applications. FlashCopy is designed to allow read and write access to the source data and the copy almost immediately following the FlashCopy volume pair establishment.

Standard FlashCopy uses a normal volume as target volume. This target volume must be the same size (or larger) than the source volume, and the space is allocated in the storage subsystem.

IBM FlashCopy SE uses track space-efficient (TSE) volumes as FlashCopy target volumes. A TSE target volume has a virtual size that is equal to or greater than the source volume size. However, space is not allocated for this volume when the volume is created and the FlashCopy is initiated. Only when updates are made to the source volume, any original tracks of the source volume to be modified are copied to the TSE volume. Space in the repository is allocated for just these tracks (or for any write to the target itself).

Additionally, thin provisioning support for FlashCopy was introduced with LMC 7.6.2.*xx.xx*. FlashCopy is supported to use extent space-efficient (ESE) volumes as source and target volumes. At the time of writing this book, this enhancement is valid for Open Systems (FB volumes) only.

Features: To enable the FlashCopy to use space-efficient volumes, specific licensed features are required:

- TSE volumes exploitation: FlashCopy SE Model SE function (SE license)
- ESE volumes exploitation: Point-in-time copy function (PTC license) and Thin provisioning LIC key feature

An ESE volume has a virtual size, and the size of the ESE target volume can be equal to or greater than the source volume size. When an ESE logical volume is created, the volume has no real capacity allocated on the extent pool but only metadata used to manage space allocation. An ESE volume extent is dynamically allocated when a write operation in the extent occurs. In a FlashCopy relationship, an ESE source volume extent is then allocated either when its corresponding extent in the source volume is allocated or when a write operation occurs directly to this extent in the target volume.

18.2.1 FlashCopy objectives

With the FlashCopy feature of the DS8000, you can make an immediate copy of a logical volume at a specific point in time, which we also refer to as a *point-in-time-copy*, *instantaneous copy*, or *time zero copy* (T0 copy), within a single DS8000 Storage Facility Image (SFI).

There are several points to consider when you plan to use FlashCopy that might help you minimize any impact that the FlashCopy operation can have on host I/O performance.

As Figure 18-1 illustrates, when FlashCopy is invoked, a relationship (or session) is established between the source and target volumes of the FlashCopy pair. This session includes the creation of the necessary bitmaps and metadata information needed to control the copy operation. This *FlashCopy establish* process completes quickly, and both the source volume and its *time zero* (T0) target volume are available for full read/write access. At this time, a background task within the DS8000 starts copying the tracks from the source to the target volume.

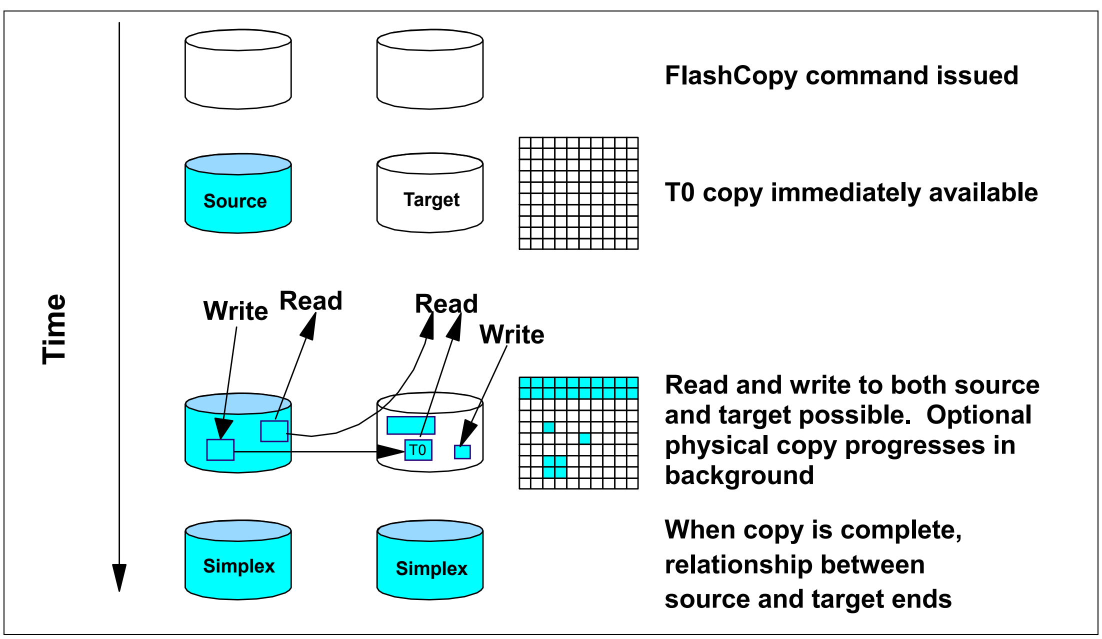


Figure 18-1 FlashCopy: Starting and ending the relationship

The DS8000 tracks which data was copied from source to target. As Figure 18-1 on page 588 shows, if an application wants to read data from the target that is not yet copied, the data is read from the source. Otherwise, the read can be satisfied from the target volume.

Optionally, you can suppress this background copy task by using the nocopy option. FlashCopy SE supports the nocopy option only. From LMC 7.6.2.*xx,* you can also use FlashCopy on a thin-provisioned (ESE) volume.

The nocopy option is efficient, for example, if you are making a temporary copy just to take a backup to tape. With the nocopy option, the full background copy does not take place and the actual copy of a track on the target volume occurs only following an update of that track on either the source or target volume. Furthermore, with the nocopy option, the FlashCopy relationship remains until explicitly withdrawn or until all the tracks are copied to the target volume.

FlashCopy SE is designed for temporary copies, such as this instance. Copy duration generally does not last longer than 24 hours unless the source and target volumes have little write activity. FlashCopy SE is optimized for use cases where a small percentage of the source volume is updated during the life of the relationship. If much more than 20% of the source is expected to change, there might be trade-offs in performance as opposed to space efficiency. In this case, standard FlashCopy might be considered as a good alternative.

FlashCopy on thin-provisioned (ESE) volumes is also efficient only if the ESE volume can be used as both source and target volume. Because only the data of allocated extents is copied. FlashCopy on thin-provisioned volume can also be used with the *nocopy* option.

FlashCopy has several options. Not all options are available to all user interfaces. It is important from the beginning to know the purpose for the target volume. Knowing this purpose, the FlashCopy options can be identified and the environment that supports the selected options can be chosen.

18.2.2 FlashCopy performance considerations

Many parameters can affect the performance of FlashCopy operations. It is important to review the data processing requirements and to select the correct FlashCopy options.

We examine when to use *copy* as opposed to *no copy* and where to place the FlashCopy *source* and *target* volumes/LUNs. We also describe when and how to use *incremental* FlashCopy, which you definitely need to evaluate for use in most applications.

IBM FlashCopy SE has special considerations. We discuss IBM FlashCopy SE in 18.2.3, “Performance planning for IBM FlashCopy SE” on page 594.

Important: This chapter is valid for System z *volumes* and Open Systems *LUNs*. In the following sections of the present chapter, we use only the terms *volume* or *volumes*, but the text is equally valid if the terms *LUN* and *LUNs* are used, *unless otherwise noted*.

Distribution of the workload: Location of source and target volumes

In general, you can achieve the best performance by distributing the load across all of the resources of the DS8000. Carefully plan your usage so that the load has these characteristics:

- Spread evenly across disk subsystems
- Within each disk subsystem, spread evenly across processor complexes
- Within each server, spread evenly across device adapters
- Within each device adapter, spread evenly across ranks

See Chapter 4, “Logical configuration performance considerations” on page 87.

It is always best to locate the FlashCopy target volume on the same DS8000 processor complex as the FlashCopy source volume, so that you can take advantage of code optimization to reduce overhead when source and target are on the same processor complex. It is also a preferred practice to locate the FlashCopy target volume on different ranks or even different device adapter (DA) pairs than the source volume, particularly when background copy is used.

Another available choice is whether to place the FlashCopy target volumes on the same ranks as the FlashCopy source volumes. In general, it is best not to place these two volumes on the same rank for the best performance. However, if source and target volumes need to be in the same non-managed, homogenous multi-rank extent pool and use rotate extents (storage pool striping) as an extent allocation method (EAM), consider consecutively created volumes as source and target volumes.

When creating a series of volumes in sequence in a homogeneous extent pool by using storage pool striping, then the extent allocation for consecutive volumes generally starts on different ranks in the pool with the following extents distributed across all available ranks in sequence. For example, when creating volumes 1000-10ff in sequence in a non-managed, homogeneous multi-rank extent pool that consists of ranks R1, R2, R3, and R4 and using storage pool striping, volume 1000 might allocate its first extent on rank R2. In this case, the successive volume 1001 starts its first extent allocation on rank R3 and following extents of both volumes are allocated in sequence across all ranks. Using volume 1000 as source and volume 1001 as target in this case ensures that each source extent of this FlashCopy pair is on a different rank than its target extent, provided that the extent allocation can successfully achieve a balanced extent distribution. Or, there are no constraints due to capacity limitations on selected ranks.

See Table 18-3 for a summary of the volume placement considerations.

Table 18-3 FlashCopy source and target volume location

	Processor complex	**Device adapter**	**Rank**
FlashCopy establish performance	Same server	Unimportant	Different ranks
Background copy performance	Same server	Different device adapter	Different ranks
FlashCopy impact to applications	Same server	Unimportant	Different ranks

Tip: To find the relative location of your volumes, you can use the following procedure:

1. Use the `lsfbvol` command to learn which extent pool contains the relevant volumes.
2. Use the `showfbvol -rank` command to learn in which rank the relevant volumes are allocated.
3. Use the `lsrank` command to display both the device adapter (DA) and the rank for each extent pool.
4. To determine which processor complex contains your volumes, look at the extent pool ID. Even-numbered extent pools are always from Server 0, and odd-numbered extent pools are always from Server 1.

Rank characteristics

Normal performance planning also includes the tasks to select the disk drives (capacity and rpms) and the RAID configurations that best match the performance needs of the applications.

With FlashCopy *nocopy* relationships, the DS8000 performs *copy-on-write* for each first change to a source volume track. If the disks of the target volume are slower than the disks of the source volume, *copy-on-write* might slow down production I/O. A *full copy* FlashCopy produces a high write activity on the disk drives of the target volume.

Therefore, it is always a preferred practice to use target volumes on ranks with the same characteristics as the source volumes.

Finally, you can achieve a small performance improvement by using identical rank geometries for both the source and target volumes. If the source volumes are on a rank with a 7+P configuration, the target volumes are also on a rank configured as 7+P.

FlashCopy establish performance

The FlashCopy of a volume has two distinct periods:

- The initial *logical* FlashCopy (also called *establish*)
- The *physical* FlashCopy (also called the *background* copy)

The FlashCopy establish phase is the period when the microcode is preparing the bitmaps that are necessary to create the FlashCopy relationship so that the microcode can correctly process later reads and writes to the related volumes. It takes only a few seconds to establish the FlashCopy relationships for tens to hundreds or more volume pairs. The copy is then immediately available for both read and write access. During this logical FlashCopy period, no writes are allowed to the source and target volume. However, this period is short. After the logical relationship is established, normal I/O activity is allowed to both source and target volumes according to the options selected.

There is a modest performance impact to logical FlashCopy establish performance when using incremental FlashCopy. In incremental FlashCopy, the DS8000 must create additional metadata (bitmaps). However, the impact is negligible in most cases.

Finally, the placement of the FlashCopy source and target volumes affects the establish performance. Table 18-3 on page 590 shows a summary of the recommendations.

Background copy performance

The *background copy* phase is the actual movement of the data from the source volume to the target volume. If the *copy* option is selected, upon completion of the logical FlashCopy establish phase, the source is copied to the target in an expedient manner.

If many volumes are established, do not expect to see all pairs actively copying data as soon as their logical FlashCopy relationship is completed. The DS8000 microcode has algorithms that limit the number of active pairs copying data. This algorithm tries to balance active copy pairs across the DS8000 DA resources. Microcode gives higher preference to application activity than copy activity.

> **Tip:** When creating many FlashCopy pairs, we suggest that all commands are submitted simultaneously, and you allow the DS8000 microcode to manage the internal resources. If using the DS8000 command-line interface (DSCLI), we suggest that you use single commands for many devices, rather than many commands with each device.

We described the suggested placement of the FlashCopy source and target volumes in the previous section. See Table 18-3 on page 590 for a summary of the conclusions. For the best background copy performance, always place the source and target volumes in different ranks. There are additional criteria to consider if the FlashCopy is a *full box copy* that involves all ranks.

Full box copy: The term *full box copy* implies that all rank resources are involved in the copy process. Either all or nearly all ranks have both source and target volumes, or half the ranks have source volumes and half the ranks have target volumes.

For full box copies, still place the source and target volumes in different ranks. When all ranks are participating in the FlashCopy, you can still place the source and target volumes in different ranks by performing a FlashCopy of volumes on rank R0 onto rank R1 and volumes on rank R1 onto rank R0, for example. Additionally, if there is heavy *application* activity in the source rank, performance is less affected if the background copy target was in another rank that has lighter *application* activity.

Important: If storage pool striping is used when allocating volumes, all ranks are more or less equally busy. Therefore, there is less need to be concerned about the data placement. But, ensure that you still keep the source and the target on the same processor complex.

If background copy performance is highly important in your environment, use incremental FlashCopy as much as possible. *Incremental* FlashCopy greatly reduces the amount of data that needs to be copied and, therefore, greatly reduces the background copy time.

If the FlashCopy relationship is established with the **`-nocp`** (*no copy*) option, only first write updates to the tracks in the source volume, or the target volume forces a copy from the source to the target. This forced copy is also called a *copy-on-write*.

Copy-on-write: The term *copy-on-write* describes a forced copy from the source to the target, because a write to the source occurred. This forced copy occurs on the *first* write to a track. Because the DS8000 writes to nonvolatile cache, there is typically no direct response time delay on host writes. A write to the source results in a copy of the track.

FlashCopy impact on applications

One of the most important considerations when implementing FlashCopy is to achieve an implementation with minimal impact on the performance of the user applications.

Consider all business requirements: The suggestions discussed in this chapter only consider the performance aspects of a FlashCopy implementation. But FlashCopy performance is only one aspect of an intelligent system design. You must consider all business requirements when designing a total solution. These additional requirements, together with the performance considerations, guide you when choosing FlashCopy options, such as *copy* or *no copy* and *incremental*, as well as when choosing source and target volume location.

The placement of the source and target volumes significantly affects the application performance. In addition to the placement of volumes, the selection of *copy* or *no copy* is also an important consideration about the effect on the application performance. Typically, the choice of *copy* or *no copy* depends primarily on how the FlashCopy is to be used and for what interval of time the FlashCopy relationship exists. From a purely performance point of view, the choice of whether to use *copy* or *no copy* depends on the type of workload. The general answer is to use *no copy*, but this choice is not always the best choice. For most workloads, including online transaction processing (OLTP) workloads, *no copy* typically is the preferred option. However, workloads that contain many random writes and that are not cache friendly might benefit from using the *copy* option.

FlashCopy nocopy

In a FlashCopy *nocopy* relationship, a copy-on-write is done whenever a write to a source track occurs for the first time after the FlashCopy is established. This type of FlashCopy is ideal when the target volumes are needed for a short time only, for example, to run the backup jobs. FlashCopy *nocopy* adds only a minimal workload on the back-end adapters and disk drives. However, it affects most of the writes to the source volumes as long as the relationship exists. When you plan to keep your target volumes for a long time, this choice might not be the best solution.

FlashCopy full copy

When you plan to use the target volumes for a longer time, or you plan to use them for production and you do not plan to repeat the FlashCopy often, then the *full copy* FlashCopy is the right choice. A full copy FlashCopy adds a high workload on the back-end DAs and disk drives. But this high additional workload lasts only for a few minutes or hours, depending on the capacity. After that, there is no additional overhead any more.

Incremental FlashCopy

Another important performance consideration is whether to use *incremental* FlashCopy. Use incremental FlashCopy when you perform FlashCopies always to the same target volumes on regular time intervals. Without the nocopy option, the first FlashCopy is a full copy, but later FlashCopy operations copy only the tracks of the source volume that are modified since the last FlashCopy.

Incremental FlashCopy has the least effect on applications. During normal operations, no copy-on-write is done (as in a nocopy relationship). And during a resync, the load on the back end is lower compared to a full copy. There is only a small overhead for the maintenance of out-of-sync bitmaps for the source and target volumes.

FlashCopy on thin-provisioned (ESE) volumes

The same considerations for standard FlashCopy also apply to the FlashCopy on thin-provisioned volumes. For the full background copy (copy option), there are more benefits compared with the standard FlashCopy because only the allocated extents are copied to the target volume for the ESE FlashCopy. Although ESE volumes have some overhead by allocating new extents due to the Quick Init process, the performance of ESE volumes is practically equal to that of standard volumes and, in general, better than FlashCopy SE.

> **Restriction:** At the time of writing this book (DS8000 LMC R6.2), thin-provisioned (ESE) volumes have these restrictions:
>
> - ESE volumes are only supported for Open Systems FB volumes.
> - ESE volume support for Copy Services is currently limited to FlashCopy.

Choosing the correct FlashCopy type

The type of copy required depends on the purpose for which the copy is made. FlashCopy *nocopy* is typically the best choice to minimize rank and DA activity within the physical DS8000. The choice depends on the purpose of the copy:

- Is the copy only to be used for creating a tape backup?

  If yes, use *nocopy* and the relationship is withdrawn after the tape backup is complete.

- Is the copy to be used for testing or development?

  If yes, *nocopy* again is typically the best choice.

- Do you need a copy of the copy?

  Use *background copy* so that the target is withdrawn from its relationship after all of the tracks are copied, allowing it to be a source in a new relationship. You might use the *nocopy* to *copy* option.

- Is the workload OLTP?

  If yes, *nocopy* typically is the best choice. Or, if there are many random writes and they are not cache friendly, *copy* might be the best choice.

18.2.3 Performance planning for IBM FlashCopy SE

FlashCopy SE has additional overhead compared to standard FlashCopy.

Data from source volumes is copied to space-efficient target volumes. The data is written to a repository, and there is a mapping mechanism to map the physical tracks to the logical tracks. See Figure 18-2. Each time that a track in the repository is accessed, it must go through this mapping process. The attributes of the volume that hosts the repository are important when planning a FlashCopy SE environment.

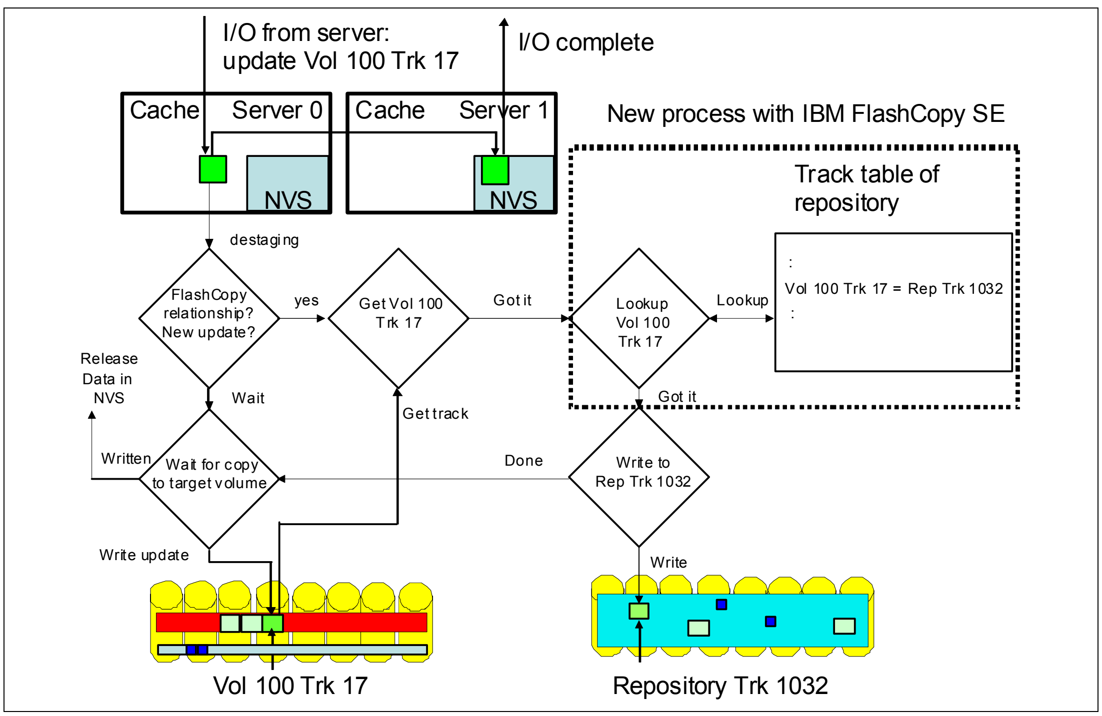


Figure 18-2 Updates to source volumes in an IBM FlashCopy SE relationship

Because of space efficiency, data is not physically ordered in the same sequence on the repository disks as it is on the source. Processes that might access the source data in a sequential manner might not benefit from sequential processing when accessing the target.

Another important consideration for FlashCopy SE is that we always have *nocopy* relationships. A full copy is not possible. If there are many source volumes that have targets in the same extent pool, all updates to these source volumes cause write activity to the repository of this one extent pool. Consider a repository as similar to a volume. So, we have writes to many source volumes being copied to just one *volume,* the repository.

Where a dedicated extent pool is defined specifically for use as a FlashCopy SE repository, there is less space in the repository than the total capacity (sum) of the source volumes. You might be tempted to use fewer disk spindles or disk drive modules (DDMs) for this extent pool. By definition, fewer spindles mean less performance, and so careful planning is needed to achieve the required throughput and response times from the space-efficient volumes. A good strategy is to keep the number of spindles roughly equivalent but to use smaller, faster drives (but do not use Serial Advanced Technology Attachment (SATA) drives). For example, if your source volumes are 300 GB 15K rpm disks, using 146 GB 15K rpm disks on the repository can provide both space efficiency and excellent repository performance.

Another possibility is to consider RAID 10 for the repository, although that goes somewhat against space efficiency (you might be better off using standard FlashCopy with RAID 5 than FlashCopy SE with RAID 10). However, there might be cases where trading off some of the space efficiency gains for a performance boost justifies RAID 10. If RAID 10 is used at the source, consider it also for the repository.

RAID: All destages to the repository volume are random destages. RAID 10 performs better than RAID 5, and RAID 5 performs better than RAID 6. There is no advantage in using RAID 6 for the repository other than resilience. Only consider RAID 6 on systems where RAID 6 is used as standard throughout the whole DS8000 and the write activity is low.

The repository always uses storage pool striping when in a multi-rank extent pool. With storage pool striping, the repository space is striped across multiple RAID arrays in an extent pool, which helps to balance the volume skew that might appear on the sources. It is generally best to use at least four RAID arrays in the multi-rank extent pool intended to hold the repository.

Finally, try to use at least the same number of disk spindles on the repository as the source volumes. Avoid severe “fan in” configurations, such as 32 ranks of source disk being mapped to an eight rank repository. This type of configuration likely has performance problems unless the update rate to the source is modest.

It is possible to share the repository with production volumes on the same extent pool, but use caution, because contention between the repository and the production volumes can affect performance. In this case, the repository for one extent pool can be placed in a different extent pool so that source and target volumes are on different ranks but on the same processor complex.

Expect a high random write workload for the repository. To prevent the repository from becoming overloaded, take the following precautions:

- Avoid placing standard source and repository target volumes in the same extent pool.
- Have the repository in an extent pool with several ranks (a repository is always striped) on the same rank group or storage server as the source volumes.
- Use fast 15K rpm disk drives for the repository ranks.
- Consider using RAID 10 instead of RAID 5, because RAID 10 can sustain a higher random write workload.
- Do not use RAID 6 for the repository unless the write activity for the source volumes is low.

Because FlashCopy SE does not need much capacity if your update rate is not too high, you might want to make several FlashCopies from the same source volume. For example, you might want to make a FlashCopy several times a day to set checkpoints to protect your data against viruses or for other reasons.

Creating more than one FlashCopy SE relationship for a source volume increases the overhead, because each first change to a source volume track must be copied several times for each FlashCopy SE relationship. Therefore, keep the number of concurrent FlashCopy SE relationships to a minimum, or test how many relationships you can have without affecting your application performance much. From a performance standpoint, avoiding multiple relationships also applies to normal FlashCopy.

There are no restrictions on the amount of virtual space or the number of SE volumes that can be defined for either z/OS or Open Systems storage.

18.3 Metro Mirror

Metro Mirror provides real-time mirroring of logical volumes between two DS8000s that can be located up to 300 km (186.4 miles) (more distance is supported via RPQ) from each other. It is a synchronous copy solution where write operations are completed on both copies before they are considered to be complete.

It is typically used for applications that cannot suffer any data loss in the event of a failure.

As data is transferred synchronously, the distance between primary and secondary disk subsystems determines the effect on application response time. Figure 18-3 illustrates the sequence of a write update with Metro Mirror.

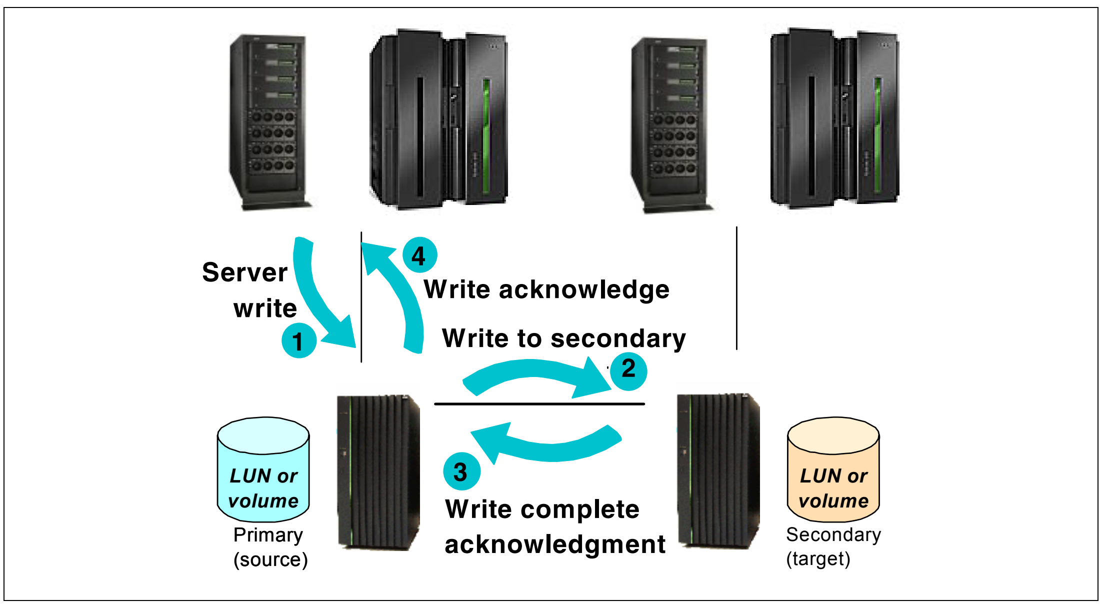


Figure 18-3 Metro Mirror sequence

When the application performs a write update operation to a primary volume, this process happens:

1. Write to primary volume (DS8000 cache)
2. Write to secondary (DS8000 cache)
3. Signal write complete on the secondary DS8000
4. Post I/O complete to host server

The Fibre Channel connection between primary and secondary subsystems can be direct, through a Fibre Channel SAN switch, via a SAN router using Fibre Channel over Internet

Protocol (FCIP), or through other supported distance solutions, such as Dense Wave Division Multiplexing (DWDM).

18.3.1 Metro Mirror configuration considerations

Metro Mirror pairs are set up between volumes, usually in different disk subsystems, which are normally in separate locations. To establish a Metro Mirror pair, there must be a Metro Mirror path between the logical subsystems (LSSs) in which the volumes reside. These paths can be shared by any Metro Mirror pairs in the same LSS to the secondary LSS in the same direction. A path must be explicitly defined in the reverse direction if required, but it can use the same Fibre Channel link. For bandwidth and redundancy, more than one path, up to a maximum of eight paths, can be created between the same LSSs. Metro Mirror balances the workload across the available paths between the primary and secondary.

The logical Metro Mirror paths are transported over physical links between the disk subsystems. The physical link includes the HA in the primary DS8000, the cabling, switches or directors, any wide band or long-distance transport devices (DWDM, channel extenders, or WAN), and the HAs in the secondary disk subsystem. Physical links can carry multiple logical Metro Mirror paths as shown in Figure 18-4 on page 598.

Metro Mirror Fibre Channel links

A DS8000 Fibre Channel port can simultaneously be the following objects:

- Sender for Metro Mirror primary
- Receiver for a Metro Mirror secondary
- Target for Fibre Channel Protocol (FCP) host I/O

Although one Fibre Channel (FC) link has sufficient bandwidth for most Metro Mirror environments, for redundancy reasons, we suggest that you configure at least two FC links between each primary and secondary disk subsystem. For better performance, use as many as the supported maximum of eight links. These links must take diverse routes between the DS8000 locations.

Metro Mirror Fibre Channel links can be direct-connection or connected by up to two switches.

Dedicating Fibre Channel ports for Metro Mirror use ensures no interference from host I/O activity, which we suggest with Metro Mirror, because it is time-critical and must not be affected by host I/O activity. The Metro Mirror ports that are used provide connectivity for all LSSs within the DS8000 and can carry multiple logical Metro Mirror paths.

Distance

The distance between your primary and secondary DS8000 subsystems affects the response time overhead of the Metro Mirror implementation. With the requirement of diverse connections for availability, it is common to have certain paths that are longer distance than others. Contact your IBM Field Technical Sales Specialist (FTSS) to assist you in assessing your configuration and the distance implications if necessary.

The maximum supported distance for Metro Mirror is 300 km (186.4 miles). There is approximately a 1 ms overhead per 100 km (62 miles) for write I/Os (this relationship between latency and physical distance might differ when you use a wide area network (WAN)). Distances of over 300 km (186.4 miles) are possible and supported by RPQ. The DS8000 *Interoperability Matrix* provides the details of SAN, network, and DWDM supported devices. Due to network configuration variability, the client must work with the channel extender vendor to determine the appropriate configuration to meet its requirements.

Logical paths

A Metro Mirror logical path is a logical connection between the sending LSS and the receiving LSS. An FC link can accommodate multiple Metro Mirror logical paths.

Figure 18-4 shows an example where we have a 1:1 mapping of source to target LSSs, and where the three logical paths are accommodated in one Metro Mirror link:

- LSS1 in DS8000 1 to LSS1 in DS8000 2
- LSS2 in DS8000 1 to LSS2 in DS8000 2
- LSS3 in DS8000 1 to LSS3 in DS8000 2

Alternatively, if the volumes in each of the LSSs of DS8000 1 map to volumes in all three secondary LSSs in DS8000 2, there are nine logical paths over the Metro Mirror link (not fully illustrated in Figure 18-4). We suggest a 1:1 LSS mapping.

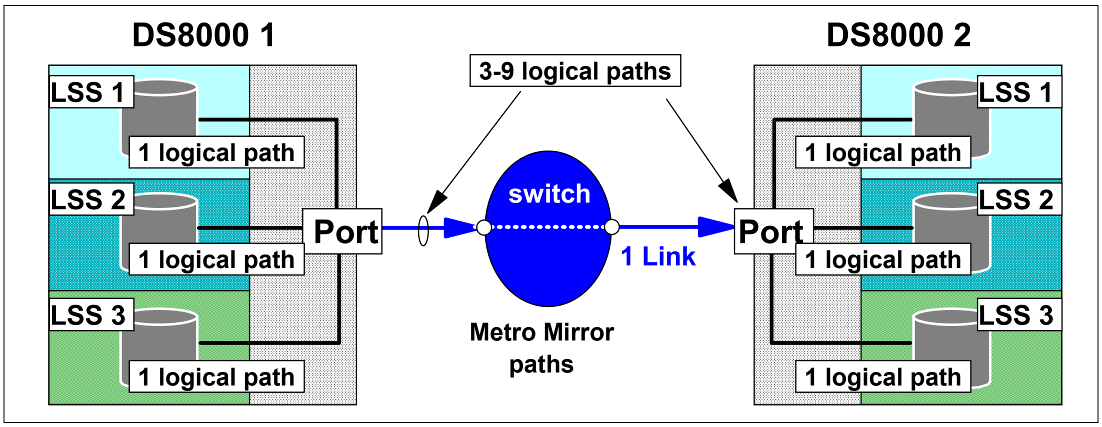


Figure 18-4 Logical paths for Metro Mirror

Metro Mirror links have the following architectural limits:

- A primary LSS can maintain paths to a maximum of four secondary LSSs. Each secondary LSS can reside in a separate DS8000.
- Up to eight logical paths per LSS to LSS relationship can be defined. Each Metro Mirror path requires a separate physical link.
- An FC port can host up to 2048 logical paths. These paths are the logical and directional paths that are made from LSS to LSS.
- An FC path (the physical path from one port to another port) can host up to 256 logical paths (Metro Mirror paths).
- An FC port can accommodate up to 126 different physical paths (DS8000 port to DS8000 port through the SAN).

For Metro Mirror, consistency requirements are managed through use of the *consistency group* or *Critical Mode* option when you define Metro Mirror paths between pairs of LSSs. Volumes or LUNs, which are paired between two LSSs whose paths are defined with the consistency group option, can be considered part of a consistency group.

Consistency is provided with the *extended long busy* (for z/OS) condition or *queue full* (for Open Systems) condition. These conditions are triggered when the DS8000 detects a condition where it cannot update the Metro Mirror secondary volume. The volume pair that first detects the error goes into the extended long busy or queue full condition, so that it does not perform any I/O. For z/OS, a system message is issued (`IEA494I state change message`); for Open Systems, a Simple Network Management Protocol (SNMP) trap message is issued. These messages can be used as triggers for automation purposes to provide data consistency by use of the `Freeze/Run` (or `Unfreeze`) commands. For more information about remote replication and data consistency, see *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788, and *DS8000 Copy Services for IBM System z*, SG24-6787.

Bandwidth

Before establishing your Metro Mirror solution, you must determine your peak bandwidth requirement. Determining your peak bandwidth requirement helps to ensure that you have enough Metro Mirror links in place to support that requirement.

To avoid any response time issues, establish the peak write rate for your systems and ensure that you have adequate bandwidth to cope with this load and to allow for growth. Remember that only writes are mirrored across to the target volumes after synchronization.

There are tools to assist you, such as Tivoli Storage Productivity Center (TPC) or the operating system-dependent tools, such as *iostat*. Another method, but not so exact, is to monitor the traffic over the FC switches by using FC switch tools and other management tools, and remember that only writes are mirrored by Metro Mirror. You can also understand the proportion of reads to writes by issuing the `datapath query devstats` command on Subsystem Device Driver (SDD)-attached servers.

A single 8 Gb Fibre Channel link of DS8800 can provide approximately 800 MBps throughput for the Metro Mirror establish. A single 4 Gb Fibre Channel link of DS8700 can provide approximately 400 MBps throughput for the Metro Mirror establish. This capability scales up linearly with additional links up to seven links. The maximum of eight links for an LSS pair provides a throughput of approximately 3,200 MBps with 4 Gbps Fibre Channel links.

> **Two-link minimum:** A minimum of two links is suggested between each DS8000 pair for resilience. The remaining capacity with a failed link can maintain synchronization.

LSS design

Because the DS8000 makes the LSS a topological construct, which is not tied to a physical array as in the ESS, the design of your LSS layout can be simplified. It is now possible to assign LSSs to applications, for example, without concern about the under-allocation or the over-allocation of physical disk subsystem resources. Assigning LSSs to applications can also simplify the Metro Mirror environment, because it is possible to reduce the number of commands that are required for data consistency.

Volume allocation

As an aid to planning and management of your Metro Mirror environment, we suggest that you maintain a *symmetrical configuration* in both physical and logical elements. As well as making the maintenance of the Metro Mirror configuration easier, maintaining a symmetrical configuration in both physical and logical elements helps you to balance the workload across the DS8000. Figure 18-5 on page 600 shows a logical configuration. This idea applies equally to the physical aspects of the DS8000. You need to attempt to balance the workload and apply symmetrical concepts to all aspects of your DS8000, which has the following benefits:

- Ensure even performance: The secondary site volumes must be created on ranks with DDMs of the same capacity and speed as the primary site.
- Simplify management: It is easy to see where volumes are mirrored and processes can be automated.
- Reduce administrator overhead: There is less administrator overhead due to automation and the simpler nature of the solution.

- Ease the addition of capacity into the environment: New arrays can be added in a modular fashion.
- Ease problem diagnosis: The simple structure of the solution helps in identifying where any problems might exist.

Figure 18-5 shows this idea in a graphical form. DS8000 #1 has Metro Mirror paths defined to DS8000 #2, which is in a remote location. On DS8000 #1, volumes defined in LSS 00 are mirrored to volumes in LSS 00 on DS8000 #2 (volume P1 is paired with volume S1, P2 with S2, and P3 with S3). Volumes in LSS 01 on DS8000 #1 are mirrored to volumes in LSS 01 on DS8000 #2. Requirements for additional capacity can be added in a symmetrical way also by the addition of volumes into existing LSSs, and by the addition of new LSSs when needed (for example, the addition of two volumes in LSS 03 and LSS 05 and one volume to LSS 04 make these LSSs have the same number of volumes as the other LSS. Additional volumes can then be distributed evenly across all LSSs, or LSSs can be added.

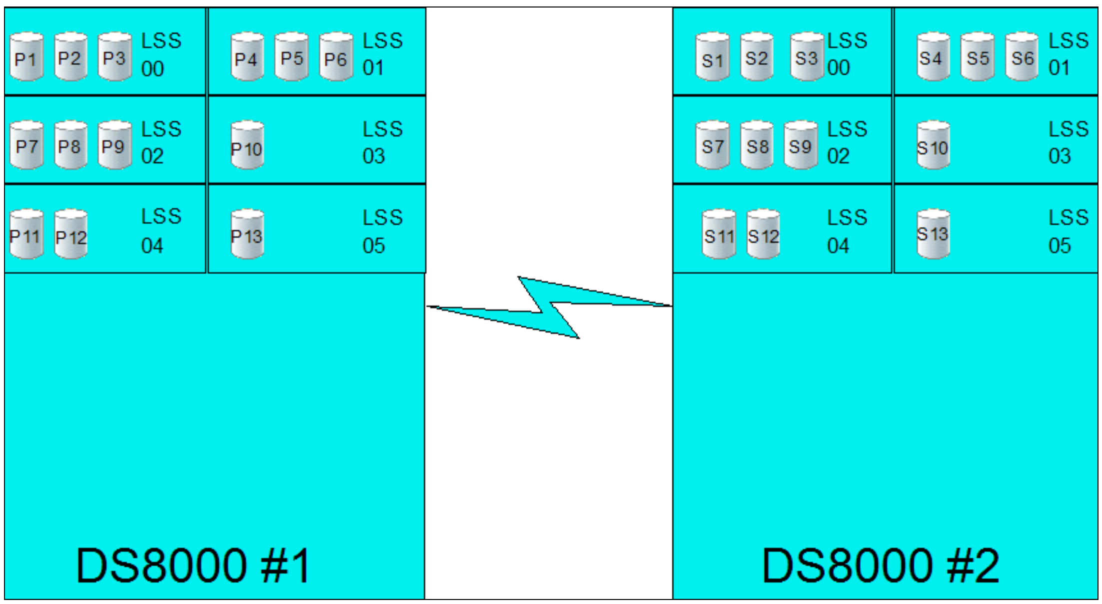


Figure 18-5 Symmetrical Metro Mirror configuration

Consider an asymmetrical configuration where the primary site has volumes defined on ranks comprised of 146 GB DDMs. The secondary site has ranks comprised of 300 GB DDMs. Because the capacity of the destination ranks is double that of the source ranks, it seems feasible to define twice as many LSSs per rank on the destination side. However, this situation, where four primary LSSs on four ranks feed into four secondary LSSs on two ranks, creates a performance bottleneck on the secondary rank and slows down the entire Metro Mirror process.

We also suggest that you maintain a symmetrical configuration in both physical and logical elements between primary and secondary storage systems in a Metro Mirror relationship when using Easy Tier automatic mode. This approach ensures that the same level of optimization and performance can be achieved on the secondary system after the production workload is switched to the secondary site and after Easy Tier successfully completes learning about the production workload and finishes data relocation on the secondary system, which requires additional time after the failover.

For more information, see 18.8, “Considerations for Easy Tier and remote replication” on page 626.

Volumes

You need to consider which volumes to mirror to the secondary site. One option is to mirror all volumes. This option is advantageous for the following reasons:

- You do not need to consider whether any required data was missed.
- Users do not need to remember which logical pool of volumes is mirrored and which is not.
- The addition of volumes to the environment is simplified. You do not have two processes for the addition of disks (one process for mirrored volumes and another process for non-mirrored volumes).
- You can move data around your disk environment easily without concern about whether the target volume is a mirrored volume.

> **Bandwidth:** Consider the bandwidth that you need to mirror all volumes. The amount of bandwidth might not be an issue if there are many volumes with a low write I/O rate. Review data from Tivoli Storage Productivity Center for Disk if it is available.

You can choose not to mirror all volumes (for example, swap devices for Open Systems or temporary work volumes for z/OS can be omitted). In this case, you must carefully control what data is placed on the mirrored volumes (to avoid any capacity issues) and what data is placed on the non-mirrored volumes (to avoid missing any required data). You can place all mirrored volumes in a particular set of LSSs, in which all volumes are Metro Mirror enabled, and direct all data that requires mirroring to these volumes.

For testing purposes, additional volumes can be configured at the remote site. These volumes can be used to take a FlashCopy of a consistent Metro Mirror image on the secondary volume and then allow the synchronous copy to restart while testing is performed.

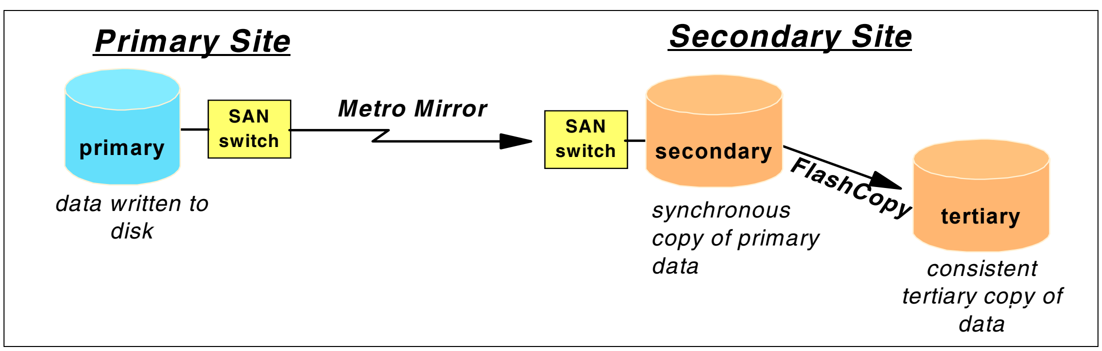


Figure 18-6 Metro Mirror environment for testing

To create a consistent copy for testing, the host I/O needs to be quiesced, or automation code, such as Geographically Dispersed Parallel Sysplex™ (GDPS) and Tivoli Storage Productivity Center for Replication, needs to be used to create a consistency group on the primary disks so that all dependent writes are copied to the secondary disks. You can also use consistent FlashCopy on the Metro Mirror secondary devices without suspending the pairs.

18.3.2 Metro Mirror performance considerations

Consider the following information when designing an infrastructure to support a Metro Mirror environment:

- The process of getting the primary and secondary Metro Mirror volumes into a synchronized state is called the *initial establish*. Each link I/O port provides a maximum throughput. Multiple LUNs in the initial establish quickly saturate the links, which is referred to as the *aggregate copy rate* and depends primarily on the number of links, or bandwidth between sites. It is important to understand this copy rate to have a realistic expectation about how long the initial establish takes to complete.

- Use Global Copy during the initial copy or for resynchronization to minimize the impact of performing both synchronous replication and the bulk copy at the same time.
- Production I/O is given priority over DS8000 replication I/O activity. High production I/O activity negatively affects both initial establish data rates and synchronous copy data rates.
- We suggest that you do not share the Metro Mirror link I/O ports with host attachment ports. Sharing them might cause unpredictable Metro Mirror performance and lead to a much more complicated root cause investigation in replication performance problems. Also, for additional considerations, see 4.10.1, “I/O port planning considerations” on page 147.
- Distance is an important value for the synchronous write performance. Data must go to the other site, and the acknowledgement goes back. Add possible latency times of certain active components on the way. We think it is a good rule to calculate 1 ms additional response time per 100 km (62 miles) of site separation for a write I/O.
- Know your workload characteristics. Factors, such as blocksize, read/write ratio, and random or sequential processing, are all key Metro Mirror performance considerations.
- Monitor link performance to determine whether links are becoming overutilized. Use tools, such as Tivoli Storage Productivity Center and Resource Measurement Facility (RMF), or review SAN switch statistics.

18.3.3 Scalability

The DS8000 Metro Mirror environment can be scaled up or down as required. If new volumes are added to the DS8000 that require mirroring, they can be dynamically added. If additional Metro Mirror paths are required, they also can be dynamically added.

> **Important:** The **`mkpprcpath`** command is used to add Metro Mirror paths. If paths are already established for the LSS pair, they must be included in the **`mkpprcpath`** command together with any additional path or the existing paths are removed.

Adding capacity to the same DS8000

If you are adding capacity to an existing DS8000, providing that your Metro Mirror link bandwidth is not close to or over capacity, it is possible that you only need to add volume pairs to your configuration. If you are adding more LSSs, you must define Metro Mirror paths before adding volume pairs.

Adding capacity in new DS8000s

If you are adding new DS8000s to your configuration, you must add physical Metro Mirror links before defining your Metro Mirror paths and volume pairs.

18.4 Global Copy

Global Copy is an asynchronous remote copy function for z/OS and Open Systems for greater distances than are possible with Metro Mirror. With Global Copy, write operations complete on the primary storage system before they are received by the secondary storage system. This capability is designed to prevent the performance of the primary system from being affected by wait time from writes on the secondary system. Therefore, the primary and secondary copies can be separated by any distance.

This function is appropriate for remote data migration, off-site backups, and the transmission of inactive database logs at virtually unlimited distances. See Figure 18-7.

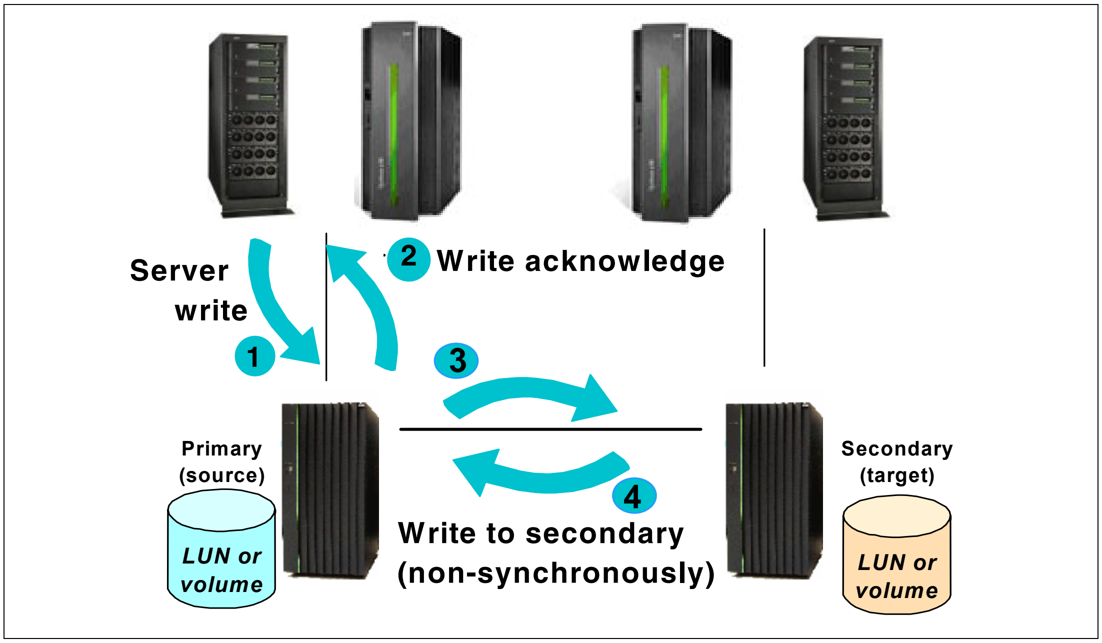


Figure 18-7 Global Copy

The following steps occur in Figure 18-7:

1. The host server requests a write I/O to the primary DS8000. The write is staged through cache and nonvolatile storage (NVS).
2. The write returns to the host server application.
3. A few moments later, in a nonsynchronous manner, the primary DS8000 sends the necessary data so that the updates are reflected on the secondary volumes. The updates are grouped in batches for efficient transmission. Note also that if the data is still in cache, only the changed sectors are sent. If the data is no longer in cache, the full track is read from disk.
4. The secondary DS8000 returns "write completed" to the primary DS8000 when the updates are secured in the secondary DS8000 cache and NVS. The primary DS8000 then resets its change recording information.

The primary volume remains in the *Copy Pending* state while the Global Copy session is active. This status only changes if a command is issued or the links between the storage subsystems are lost.

18.4.1 Global Copy configuration considerations

The requirements for establishing a Global Copy relationship are essentially the same as for Metro Mirror as described in 18.3.1, "Metro Mirror configuration considerations" on page 597.

A path must be established between the source LSS and target LSS over a Fibre Channel link. The major difference is the distance over which Global Copy can operate. Because it is a nonsynchronous copy, the distance is effectively unlimited. Consistency must be manually created by the user.

> **Global copy:** The consistency group is *not* specified on the `establish path` command. Data on Global Copy secondaries is not consistent so there is no need to maintain the order of dependent writes.

The decision about when to use Global Copy depends on a number of factors:

- The recovery of the system does not need to be current with the primary application system.
- There is a minor impact to application write I/O operations at the primary location.
- The recovery uses copies of data created by the user on tertiary volumes.
- Distances beyond FCP limits are required: 300 km (186 miles) for FCP links (RPQ for greater distances).
- You can use Global Copy as a tool to migrate data between data centers.

Distance

The maximum (supported) distance for a direct Fibre Channel connection is 10 km (6.2 miles). If you want to use Global Copy over longer distances, you can use the following connectivity technologies to extend this distance:

- Fibre Channel routers using Fibre Channel over Internet Protocol (FCIP)[1]
- Dense Wavelength Division Multiplexers (DWDM) on fiber

Global Copy Fibre Channel router support

Fibre Channel router (channel extender) vendors connect the DS8000 systems through various wide area network (WAN) connections, including Fibre Channel, Ethernet/IP, ATM-OC3, and T1/T3. When using such products with Global Copy, the vendor determines the maximum distance supported between the primary and secondary DS8000 systems. You must contact the vendor for its distance capability, line quality requirements, and WAN attachment capabilities. You also must contact the vendor about hardware and software prerequisites when you use that vendor's products in a DS8000 Global Copy configuration. Evaluation, qualification, approval, and support of Global Copy configurations using Fibre Channel router (channel extender) products is the sole responsibility of the vendor.

Global Copy Wave Division Multiplexer support

Wavelength Division Multiplexing (WDM) and Dense Wavelength Division Multiplexing (DWDM) are the basic technologies of fiber optic networking. It is a technique for carrying many separate and independent optical channels on a single fiber.

A simple way to envision DWDM is to consider that at the primary end, multiple fiber optic input channels, such as Fibre Channel, FICON, or Gbit Ethernet, are combined by the DWDM into a single fiber optic cable. Each channel is encoded as light of a different wavelength. You might think of each channel as an individual color; the DWDM system is transmitting a rainbow. At the receiving end, the DWDM fans out the different optical channels. DWDM, by the nature of its operation, provides the full bandwidth capability of the individual channel. Because the wavelength of light is from a practical perspective infinitely divisible, DWDM technology is only limited by the sensitivity of its receptors for the total possible aggregate bandwidth. You must contact the multiplexer vendor regarding hardware and software prerequisites when using the vendor's products in a DS8000 Global Copy configuration.

[1] Fibre Channel routers that use FCIP over wide area network (WAN) lines are also referred to as channel extenders.

Other planning considerations

When planning to use Global Copy for point-in-time backup solutions as shown in Figure 18-8, you must also consider the configuration of a second volume at the secondary site. If you plan to have tertiary copies, you must have available a set of volumes ready to become the FlashCopy target within the target Storage Facility Image. If your next step is to dump the tertiary volumes onto tapes, you must ensure that the tape resources can handle these dump operations in between the point-in-time checkpoints unless you have additional sets of volumes ready to become alternate FlashCopy targets within the secondary storage subsystems.

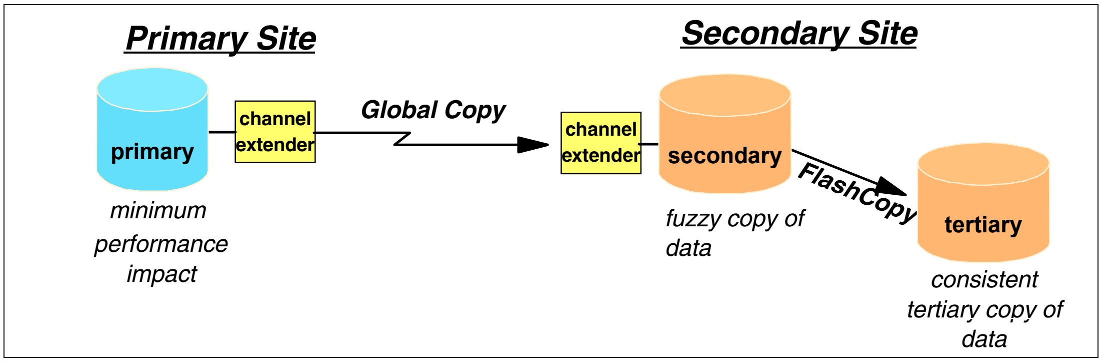


Figure 18-8 Global Copy environment for testing

Following these steps, the user creates consistent data:

1. Quiesce the I/O.
2. Suspend the pairs (*go-to-synch* and *suspend)*.

   FREEZE can be used and extended long busy is not returned to the server, because the consistency group was *not* specified on the establish path.
3. FlashCopy secondary to tertiary.

   The tertiary has consistent data.
4. Reestablish paths (if necessary).
5. RESYNC (`resumepprc`) Global Copy.

See "Creating a consistent point-in-time copy" in the "Global Copy options and configuration" chapter in *DS8000 Copy Services for IBM System z*, SG24-6787, and *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788.

18.4.2 Global Copy performance considerations

As the distance between DS8000s increases, Metro Mirror response time is proportionally affected, which negatively affects the application performance. When you need implementations over extended distances, Global Copy becomes an excellent trade-off solution.

You can estimate the Global Copy application impact to be similar to the impact of the application when working with Metro Mirror suspended volumes. For the DS8000, there is additional work to do with the Global Copy volumes compared to the suspended volumes, because with Global Copy, the changes must be sent to the remote DS8000. But this impact is negligible overhead for the application compared with the typical synchronous overhead.

There are no host system resources consumed by Global Copy volume pairs, excluding any management solution, because the Global Copy is managed by the DS8000 subsystem.

If you take a FlashCopy at the recovery site in your Global Copy implementation, consider the influence between Global Copy and the FlashCopy background copy. If you use the FlashCopy with the nocopy option at the recovery site, when the Global Copy target receives an update, the track on the FlashCopy source, which is also the Global Copy target, must be copied to the FlashCopy target before the data transfer operation completes. This copy operation to the FlashCopy target can complete by using the DS8000 cache and NVS without waiting for a physical write to the FlashCopy target. However, this data movement can influence the Global Copy activity. So, when considering the network bandwidth, consider that the FlashCopy effect over the Global Copy activity might in fact decrease the bandwidth utilization during certain intervals.

18.4.3 Scalability

The DS8000 Global Copy environment can be scaled up or down as required. If new volumes that require mirroring are added to the DS8000, they can be dynamically added. If additional Global Copy paths are required, they also can be dynamically added.

Addition of capacity

The logical nature of the LSS makes a Global Copy implementation on the DS8000 easier to plan, implement, and manage. However, if you need to add more LSSs to your Global Copy environment, your management and automation solutions must be set up to add this capacity.

Adding capacity to the same DS8000

If you add capacity into an existing DS8000, providing your Global Copy link bandwidth is not close to or over capacity, you might only need to add volume pairs into your configuration. If you add more LSSs, you need to define Metro Mirror paths before adding volume pairs. Remember that when you add capacity that you want to use for Global Copy, you might also have to purchase capacity upgrades to the appropriate feature code for Global Copy.

Adding capacity in new DS8000s

If you add DS8000s into your configuration, you need to add physical Global Copy links before defining your Global Copy paths and volume pairs. We suggest a minimum of two Global Copy paths per DS8000 pair for redundancy reasons. Your bandwidth analysis indicates whether you require more than two paths.

18.5 Global Mirror

Global Mirror copying provides a two-site extended distance remote mirroring function for z/OS and Open Systems servers by combining and coordinating Global Copy and FlashCopy operations (Figure 18-9 on page 607). With Global Mirror, the data that the host writes to the storage unit at the local site is asynchronously copied to the storage unit at the remote site. A consistent copy of the data is then periodically automatically maintained on the storage unit at the remote site by forming a consistency group at the local site, and later creating a tertiary copy of the data at the remote site with FlashCopy. This two-site data mirroring function is designed to provide a high performance, cost-effective, global distance data replication and disaster recovery solution.

Global Mirror objectives

The goal of Global Mirror is to provide these functions:

- Capability to achieve a recovery point objective (RPO) down to 1 - 2 seconds with sufficient bandwidth and resources

- No impact on production applications when insufficient bandwidth and resources are available
- Scalability, providing consistency across multiple primary and secondary disk subsystems
- Allowance for removal of duplicate writes within a consistency group before sending data to the remote site
- Allowance for less than peak bandwidth to be configured by allowing the RPO to increase without restriction at these times
- Consistency between System z and Open Systems data and between different platforms on Open Systems

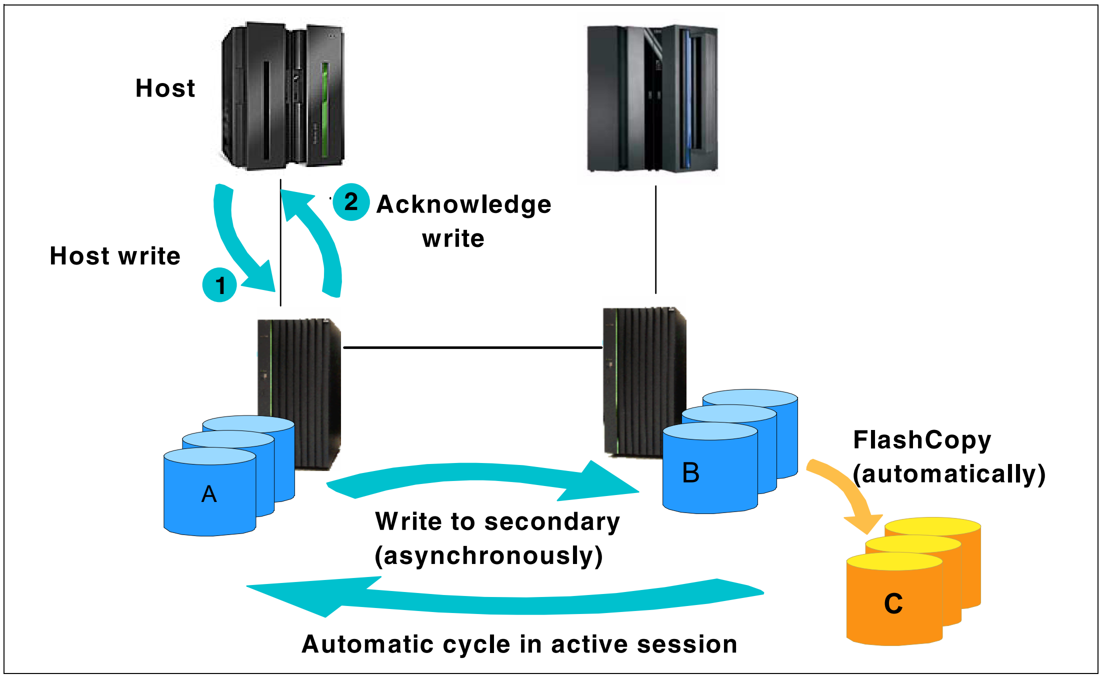


Figure 18-9 Global Mirror overview

The DS8000 manages the sequence to create a consistent copy at the remote site (Figure 18-10 on page 608):

- Asynchronous long-distance copy (Global Copy) with little to no impact to application writes.
- Momentarily pause for application writes (fraction of a millisecond to a few milliseconds).
- Create point-in-time consistency group across all primary subsystems in out-of-sync (OOS) bitmap. New updates are saved in the Change Recording bitmap.
- Restart application writes and complete the write (drain) of point-in-time consistent data to the remote site.
- Stop the drain of data from the primary after all consistent data is copied to the secondary.
- Logically FlashCopy all data to C volumes to preserve consistent data.
- Restart Global Copy writes from the primary.
- Automatic repeat of sequence from once per second to hours (this choice is selectable).

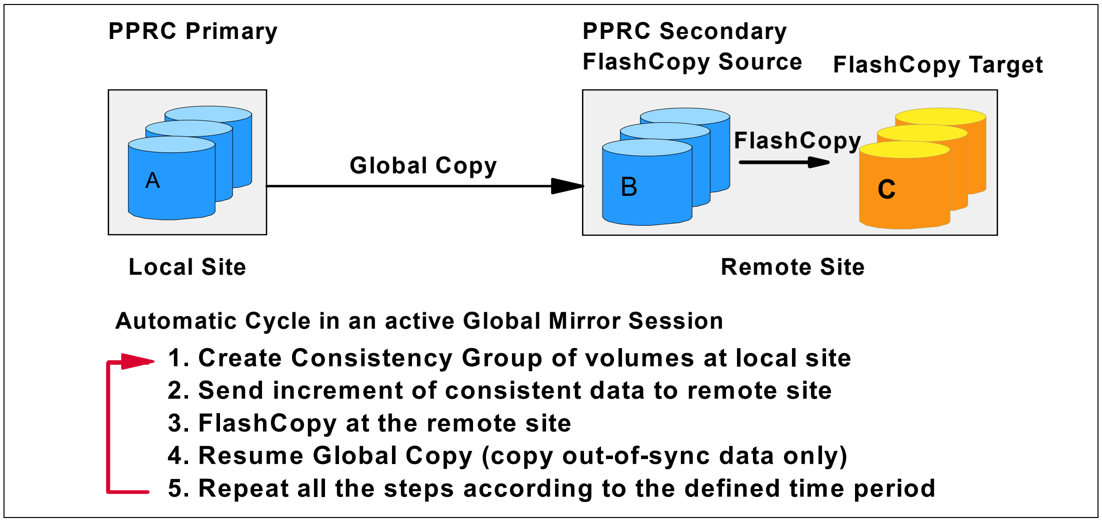


Figure 18-10 How Global Mirror works

The data at the remote site is current within 3 - 5 seconds, but this RPO depends on the workload and bandwidth available to the remote site.

Using this copy for recovery: The copy created with the consistency group is a power-fail consistent copy, not necessarily an application-based consistent copy. When you use this copy for recovery, you might need to perform additional recovery operations, such as the `fsck` command in an AIX filesystem.

This section explains performance aspects when planning and configuring for Global Mirror together with the potential impact to application write I/Os caused by the process used to form a consistency group.

We also consider distributing the target Global Copy and target FlashCopy volumes across various ranks to balance the load over the entire target storage server and minimize the I/O load for selected busy volumes.

18.5.1 Global Mirror performance considerations

Global Mirror consists of Global Copy and FlashCopy functions and combines both functions to create a distributed solution to provide consistent data at a remote site. We analyze the performance at the production site and at the recovery site, as well as between both sites, with the objective of providing a stable RPO without affecting production:

- At the production site, where production I/O always has a higher priority over DS8000 replication I/O activity, the storage server needs resources to handle both loads. If your primary storage server is already overloaded with production I/O, the potential delay before a consistency group can be formed might become unacceptable.
- The bandwidth between both sites needs to be sized for production load peaks. This sizing needs to allow for the loss of a link and still maintain the desired RPO.
- At the recovery site, even if there is no local production I/O workload, the recovery site hosts the target Global Copy volumes, handles the inherent FlashCopy processing, and needs the performance evaluated.

The performance of the Global Mirror session, the storage subsystems, and the links between subsystems must be monitored to collect data to allow performance issues to be investigated. There are a number of tools that can assist with these tasks:

- Tivoli Storage Productivity Center can collect data from storage subsystems and SAN switches. This data provides the details of the utilization of the individual components, such as I/O rates for individual volumes.
- RMF data can be collected for z/OS systems and analyzed using RMF Magic, which is available from IntelliMagic. For more information about this product, see this website:

  http://www.intellimagic.net
- Disk Magic, also available from IntelliMagic, can provide planning information about proposed workloads or additional workloads and provide estimates on the inter-site bandwidth required. For more information about Disk magic, see 6.1, “Disk Magic” on page 176.
- The Global Mirror Monitor is a tool available from IBM Field Technical Sales Specialists (FTSS). This tool provides information about the consistency group interval and details of the out-of synch (OOS) tracks.
- Tivoli Storage Productivity Center for Replication can monitor the status of the session and send SNMP alerts:
  - Session state change
  - Configuration change
  - Suspending-event notification
  - Communication failure
  - High-availability state change

Primary site DS8000 performance

Configure the primary DS8000 according to the recommendations made in Chapter 4, “Logical configuration performance considerations” on page 87. A balanced configuration that fully uses the internal DS8000 resources provides the most consistent performance.

If the primary DS8000 is already configured, you can measure the current performance by using tools, such as Tivoli Storage Productivity Center for Disk, to fix any performance bottlenecks caused by the configuration before the remote copy is established.

The PPRC links must use dedicated DS8000 HA ports to avoid any conflict with host I/O. If any subordinate storage subsystems are included in the Global Mirror session, the FC links to those subsystems must also use dedicated HA ports.

Global Mirror Fibre Channel links

The links between primary and secondary sites require bandwidth sufficient to maintain the desired RPO. The goal is typically to provide the synchronization of thousands of volumes on multiple primary and secondary storage subsystems with an RPO of 3 - 5 seconds.

The cost of providing these links can be high if they are sized to provide this RPO under all circumstances. If it is acceptable to allow the RPO to increase slightly during the highest workload times, you might be able to reduce the bandwidth and the cost of the link. In many instances, the highest write rate occurs overnight during backup processing, and increased RPO can be tolerated.

The difference in bandwidth and costs for maintaining an RPO of a few seconds might be double that of maintaining an RPO of a few minutes at peak times. Recovery to the latest consistency group is immediate after the peak passes; there is no catch-up time.

For more details and examples, see the *Global Mirror Technical Whitepaper*:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100642

FlashCopy in Global Mirror

This section looks at the aggregate impact of Global Copy and FlashCopy in the overall performance of Global Mirror. Remember that Global Copy has minimal or no significant impact on the response time of an application write I/O to a Global Copy primary volume.

The FlashCopy used as a part of this Global Copy operation is running in *nocopy* mode and causes additional internally triggered I/Os within the target storage server for each write I/O to the FlashCopy source volume, that is, the Global Copy target volume. This I/O is to preserve the last consistency group.

Each Global Copy write to its secondary volume during the time period between the formation of successive consistency groups causes an actual FlashCopy write I/O operation on the target DS8000 server. Figure 18-11 summarizes approximately what happens between two consistency group creation points when the application writes are received.

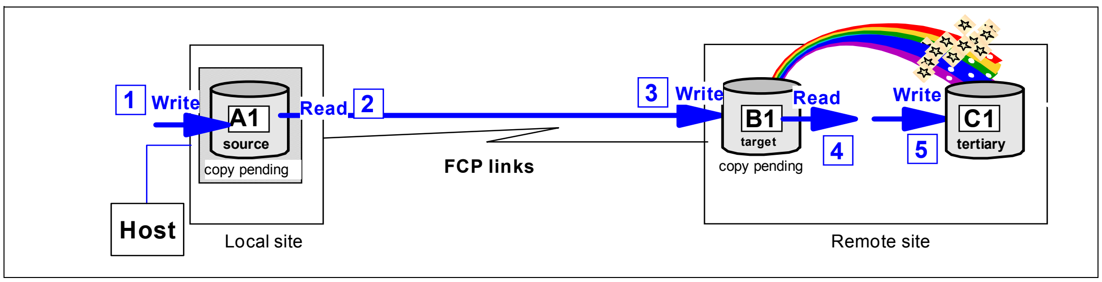


Figure 18-11 Global Copy with write hit at the remote site

The following steps show the FlashCopy write I/O operation on the target DS8000 server (follow the numbers in Figure 18-11):

1. The application write I/O completes immediately to volume A1 at the local site.
2. Global Copy nonsynchronously replicates the application I/O and reads the data at the local site to send to the remote site.
3. The modified track is written across the link to the remote B1 volume.
4. FlashCopy *nocopy* sees that the track is about to change.
5. The track is written to the C1 volume before the write to the B1 volume.

This process is an approximation of the sequence of internal I/O events. There are optimization and consolidation effects that make the entire process efficient.

Figure 18-11 showed the normal sequence of I/Os within a Global Mirror configuration. The critical path is between points (2) and (3). Usually (3) is simply a write hit in NVS in B1, and some time later and after (3) completes, the original FlashCopy source track is copied from B1 to C1.

If NVS is overcommitted in the secondary storage server, there is a potential impact on the Global Copy data replication operation performance. See Figure 18-12 on page 611.

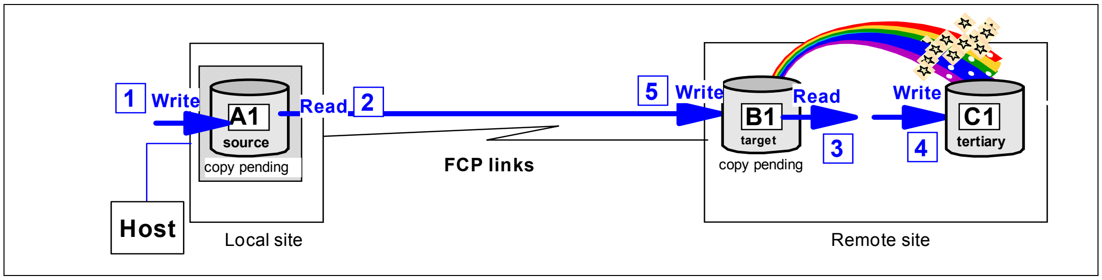


Figure 18-12 Application write I/O within two consistency group points

Figure 18-12 summarizes roughly what happens when NVS in the remote storage server is overcommitted. A read (3) and a write (4) to preserve the source track and write it to the C volume are required before the write (5) can complete. Eventually, the track gets updated on the B1 volume to complete the write (5). But usually, all writes are quick writes to cache and persistent memory and happen in the order as outlined in Figure 18-10 on page 608.

You can obtain a more detailed explanation of this processing in *DS8000 Copy Services for IBM System z*, SG24-6787, and *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788.

18.5.2 Global Mirror session parameters

Global Mirror has three tunable values to modify its behavior.

Default values: In most environments, use the default values. These default values are maximum intervals, and in practice, the actual interval is usually shorter.

Maximum coordination time

The *maximum coordination time* is the maximum time that Global Mirror allows for the determination of a consistent set of data before failing this consistency group. Having this cut-off ensures that even if there is an error recovery event or communications problem, the production applications do not experience significant impact from consistency group formation.

The default for the maximum coordination time is 50 ms, which is a small value compared to other I/O timeout values, such as the missing-interrupt handler (MIH) (30 seconds) or Small Computer System Interface (SCSI) I/O timeouts. Even in error situations where we might trigger this timeout, Global Mirror protects production performance rather than affecting production in an attempt to form consistency groups in a time where there might be error recovery or other problems occurring.

Performance considerations at coordination time

When looking at the three phases that Global Mirror goes through to create a set of data consistent volumes at the secondary site, determine whether the coordination window imposes an impact to the application write I/O. See Figure 18-13 on page 612.

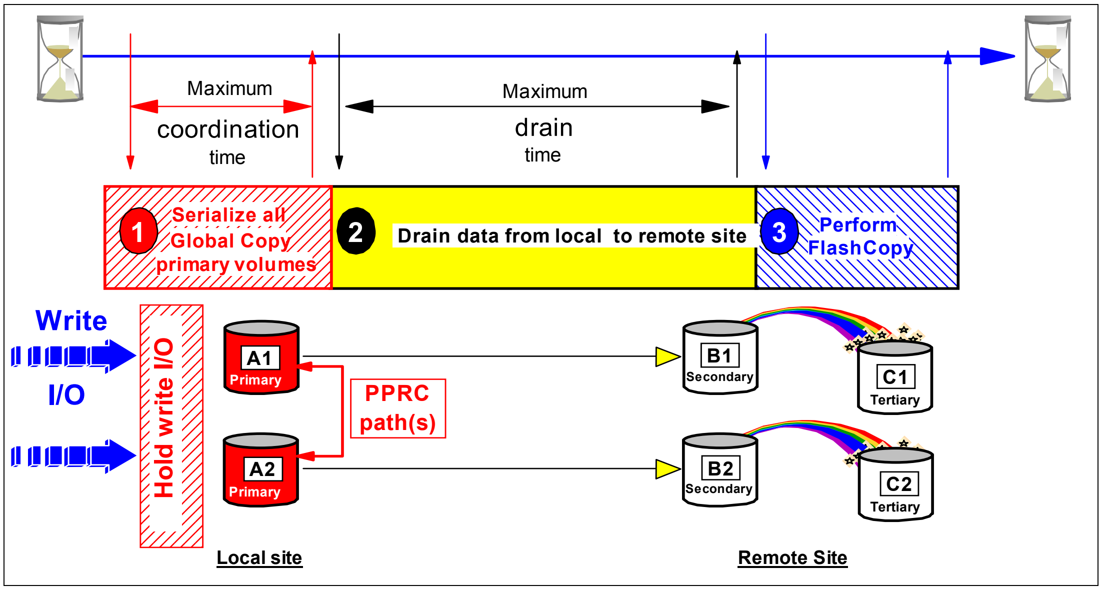


Figure 18-13 Coordination time and how it impacts application write I/Os

The *coordination time*, which you can limit by specifying a number of milliseconds, is the maximum impact to the application write I/Os that you allow when forming a consistency group. The intention is to keep the coordination time value as small as possible. The default of 50 ms might be high in a transaction processing environment. A valid number might also be in the single digit range. The required communication between the Master storage server and potential Subordinate storage servers is in-band over PPRC paths between the Master and Subordinates. This communication is highly optimized and you can minimize the potential application write I/O impact to 3 ms, for example. There must be at least one PPRC FC link between a Master storage server and each Subordinate storage server, although for redundancy, we suggest that you use two PPRC FC links.

One of the key design objectives for Global Mirror is to not affect the production applications. The consistency group formation process involves holding production write activity to create dependent write consistency across multiple devices and multiple disk subsystems. This process must be fast enough that the impact is small. With Global Mirror, the process of forming a consistency group is designed to take 1 - 3 ms. If we form consistency groups every 3 - 5 seconds, the percentage of production writes affected and the degree of impact is small.

The following example shows the type of impact that might be seen from consistency group formation in a Global Mirror environment.

We assume 24000 I/Os per second with a 3:1 R/W ratio. We perform 6000 write I/Os per second. Each write I/O takes 0.5 ms, and it takes 3 ms to create a consistent set of data.

Approximately 0.0035 x 6000 = 21 write I/Os are affected by the creation of consistency.

If each of these 21 I/Os experiences a 3 ms delay, and this delay happens every 3 seconds, we have an average response time (RT) delay of (21x 0.003)/18000 = 0.0035 ms.

A 0.0035 ms average impact to a 0.5 ms write is a 0.7% increase in response time, and normal performance reporting tools do not detect this level of impact.

Maximum drain time

The maximum drain time is the maximum amount of time that Global Mirror spends draining a consistency group *out-of-sync* bitmap before failing the consistency group. If the maximum drain time is exceeded, Global Mirror transitions to Global Copy mode for a period to catch up in the most efficient manner. While in Global Copy mode, the overhead is lower than continually trying and failing to create consistency groups.

The previous consistency group is still available on the C devices so the effect of this situation is that the RPO increases for a short period. The primary disk subsystem evaluates when it is possible to continue to form consistency groups and restarts consistency group formation at this time.

The default for the maximum drain time is 30 seconds, which allows a reasonable time to send a consistency group while ensuring that if there is a non-fatal network or communications issue that we do not wait too long before evaluating the situation and potentially dropping into Global Copy mode until the situation is resolved. In this way, we again protect the production performance rather than attempting (and possibly failing) to form consistency groups at a time when forming consistency groups might be inappropriate.

If we are unable to form consistency groups for 8 hours, by default, Global Mirror forms a consistency group without regard to the maximum drain time. It is possible to change this time if this behavior is undesirable in a particular environment.

Consistency group drain time

This *drain period* is the time required to replicate all remaining data for the consistency group from the primary to the secondary storage server. This drain period needs to fall within a time limit set by *maximum drain time,* which can also be limited. The default is 30 seconds, and this drain period might be too short in an environment with a write-intensive workload.

The actual replication process usually does not affect the application write I/O. There is a slight chance that the same track within a consistency group is updated before this track is replicated to the secondary site within the specified drain period. When this unlikely event happens, the affected track is immediately (synchronously) replicated to the secondary storage server before the application write I/O modifies the original track. In this exceptional case, the application write I/O is affected, because it must wait for the write to complete at the remote site as in a Metro Mirror synchronous configuration.

Later writes to this same track do not experience any delay, because the tracks are already replicated to the remote site.

Consistency group interval

The *consistency group interval* is the amount of time that Global Mirror spends in Global Copy mode between the formation of each consistency group. The effect of increasing this value is to increase the RPO and can increase the efficiency of the bandwidth utilization by increasing the number of duplicate updates that occur between consistency groups that then do not need to be sent from the primary to the secondary disk subsystems.

However, because it also increases the time between successive FlashCopies, increasing this value is not necessary and might be counterproductive in high-bandwidth environments, because frequent consistency group formation reduces the overhead of *Copy on Write* processing.

The default for the consistency group interval is 0 seconds, so Global Mirror continuously forms consistency groups as fast as the environment allows. In most situations, we suggest leaving this parameter at the default and allowing Global Mirror to form consistency groups as fast as possible given the workload, because Global Mirror automatically moves to Global Copy mode for a period if the drain time is exceeded.

18.5.3 Avoid unbalanced configurations

When the load distribution is unknown in your configuration, consider using Tivoli Storage Productivity Center for Disk or RMF to gather information about rank and volume utilization. There are two loads to consider: the production load on the production site and the Global Mirror load on both sites.

There are only production volumes on the production site. Configure the DS8000 for the best performance as we discussed in Chapter 4, “Logical configuration performance considerations” on page 87. At the same time, the storage server needs to be able to handle both production and replication workloads. In general, create a balanced configuration that uses all device adapters, ranks, and processor complexes.

At the recovery site, you have to consider Global Mirror volumes only, but there are two types: target Global Copy and target FlashCopy. Where the B volume is used for production in a failover situation, the DDM size can be double that of the production site DDM size. Global Mirror still gives an identical number of spindles and capacity, because the FlashCopy volume is not in use in this situation. Where a fourth volume is used to facilitate disaster recovery testing without removing the Global Mirror copy facility for the duration of the test, only 50% more drives are required if you use double-capacity DDMs.

With Easy Tier automatic mode controlling the extent distribution in managed extent pools, consider allocating the B and C volumes on the recovery site in the same multi-rank extent pool. Easy Tier spreads busy extents across ranks and optimizes overall extent distribution across ranks and tiers in the extent pool based on the workload profile. The workload is balanced across the resources in the extent pool and performance bottlenecks are avoided. This approach provides optimal performance.

You can still separate workloads by using different extent pools by using the principles of workload isolation as described in 4.8, “Planning extent pools” on page 115 or use a manual approach as described in “Remote DS8000 configuration” without Easy Tier automatic mode.

Remote DS8000 configuration

There are I/O skews and hot spots in storage servers for both the local and remote storage servers. In local storage servers, consider a horizontal pooling approach and spread each volume type across all ranks. Volume types in this context are, for example, DB2 database volumes, logging volumes, batch volumes, and temporary work volumes. Your goal can be to have the same number of each volume type within each rank.

Through a one-to-one mapping from local to remote storage server, you achieve the same configuration at the remote site for the B volumes and the C volumes. Figure 18-14 on page 615 proposes to spread the B and C volumes across different ranks at the remote storage server so that the FlashCopy target is on a different rank than the FlashCopy source.

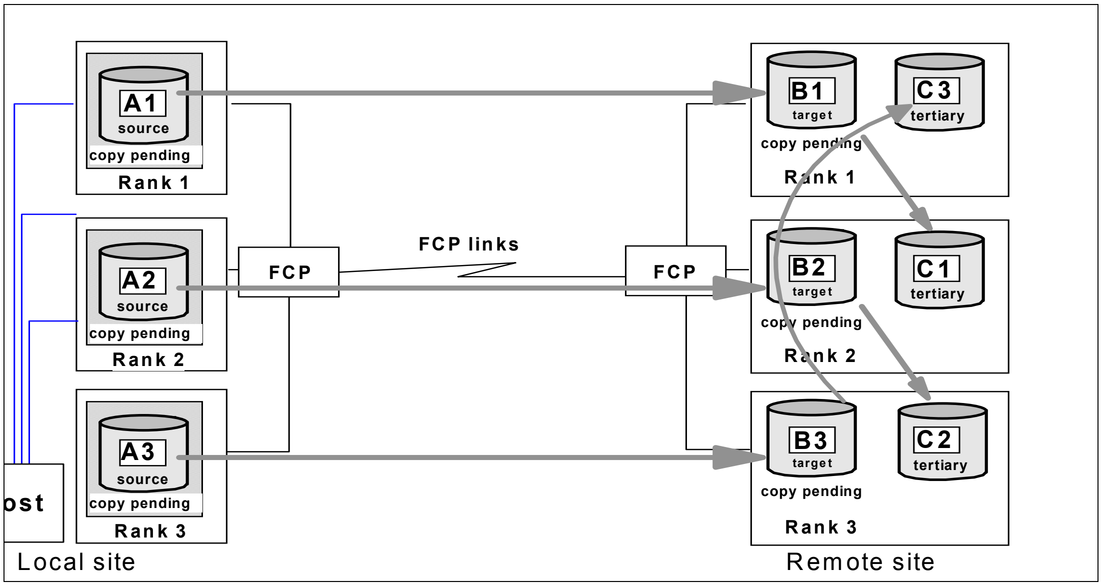


Figure 18-14 Remote storage server configuration: All ranks contain equal numbers of volumes

The goal is to put the same number of each volume type into each rank. The volume types that we describe refer to B volumes and C volumes within a Global Mirror configuration. To avoid performance bottlenecks, spread busy volumes over multiple ranks. Otherwise, hot spots can be concentrated on single ranks when you put the B and C volumes on the same rank. We suggest that you spread B and C volumes as Figure 18-14 suggests.

With mixed DDM capacities and different speeds at the remote storage server, consider spreading B volumes over the fast DDMs and over all ranks. Basically, follow a similar approach as Figure 18-14 suggests. You might keep busy B volumes and C volumes on the faster DDMs.

If the DDMs used at the remote site are double the capacity but the same speed as those DDMs used at the production site, an equal number of ranks can be formed. In a failover situation when the B volume is used for production, it provides the same performance as the production site, because the C volume is not then in use.

Important: Keep the FlashCopy target C volume on the same processor complex as the FlashCopy source B volume.

Figure 18-15 on page 616 introduces the D volumes.

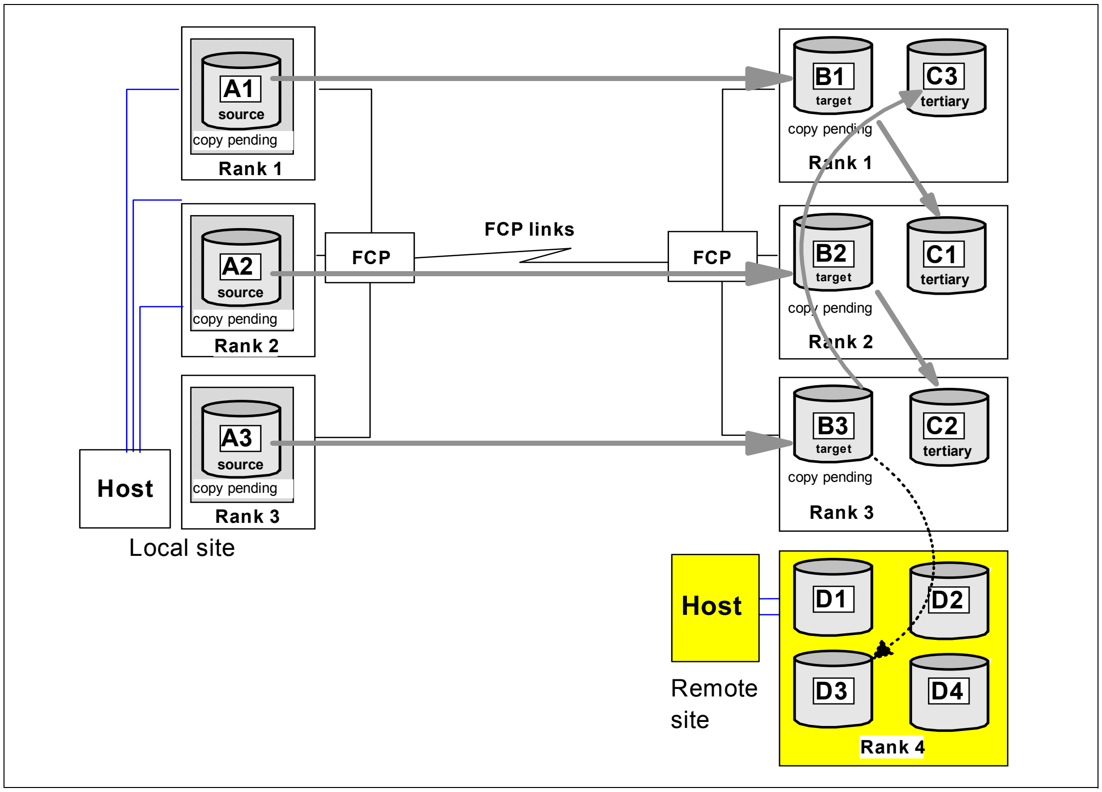


Figure 18-15 Remote storage server with D volumes

Figure 18-15 shows the three Global Mirror volumes and the addition of D volumes that you can create for test purposes. We suggest, as an alternative, a rank with larger and slower DDMs. The D volumes can be read from another host, and any other I/O to the D volumes does not affect the Global Mirror volumes in the other ranks. A *nocopy* relationship between B and D volumes reads the data from B when coming through the D volume. So, you might consider a physical COPY when you create D volumes on a different rank, which separates additional I/O to the D volumes from I/O to the ranks with the B volumes.

If you plan to use the D volumes as the production volumes at the remote site in a failover situation, the D volume ranks must be configured in the same way as the A volume ranks and use identical DDMs. You must make a full copy to the D volume for both testing and failover. When using Tivoli Storage Productivity Center for Replication, the *Copy Sets* for Global Mirror Failover/Failback w/ Practice are defined in this way. The Tivoli Storage Productivity Center for Replication volume definitions are listed:

- A volume defined as H1 volume (Host site 1)
- B volume defined as I2 volume (Intermediate site 2)
- C volume defined as J2 volume (Journal site 2)
- D volume defined as H2 volume (Host site 2)

Using FlashCopy SE with Global Mirror

The use of FlashCopy SE for the C volume requires a different configuration again. These volumes are physically allocated in a data repository. A repository volume per extent pool is used to provide physical storage for all space-efficient volumes in that extent pool. Figure 18-16 on page 617 shows an example of a Global Mirror setup with FlashCopy SE. In this example, the FlashCopy targets use a common repository.

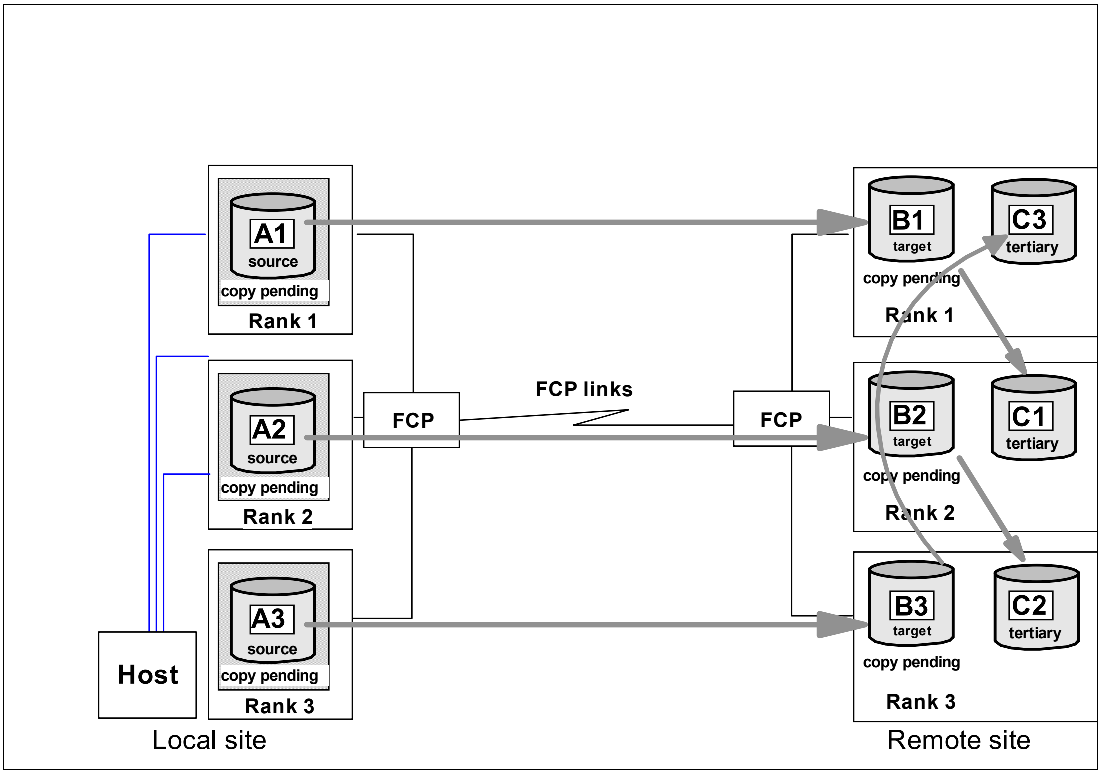


Figure 18-16 Remote disk subsystem with space-efficient FlashCopy target volumes

FlashCopy SE is optimized for use cases where less than 20% of the source volume is updated during the life of the relationship. In most cases, Global Mirror is configured to schedule consistency group creation at an interval of a few seconds, which means that a small amount of data is copied to the FlashCopy targets. From this point of view, Global Mirror is a suggested area of application for FlashCopy SE.

In contrast, Standard FlashCopy generally has superior performance to FlashCopy SE. The FlashCopy SE repository is critical regarding performance. When provisioning a repository, storage pool striping automatically is used with a multi-rank extent pool to balance the load across the available disks. In general, we suggest a minimum of four RAID arrays in the extent pool. Depending on the logical configuration of the DS8000, you might also consider the use of multiple space-efficient repositories for the FlashCopy target volume in a Global Mirror environment, at least one on each processor complex. The repository extent pool can also contain additional non-repository volumes.

Contention can arise if the extent pool is shared. After the repository is defined, you cannot expand it so it is important that you plan to ensure that it is large enough. If the repository fills, the FlashCopy SE relationship fails and the Global Mirror is not able to successfully create consistency groups.

18.5.4 Growth within Global Mirror configurations

When a session is active and running, you can alter the Global Mirror environment to add or remove volumes. You also can add storage disk subsystems to a Global Mirror session or you can change the interval between the formation of consistency groups.

When many volumes are used with Global Mirror, it is important that you configure sufficient cache memory to provide for the best possible overall function and performance. To accommodate additional data structures that are required to efficiently run a Global Mirror environment, Table 18-4 lists the maximum number of Global Mirror volume relationships that are currently suggested on a DS8000 as a function of the secondary subsystem cache size. This table also applies to the tertiary DS8000 in a Metro/Global Mirror (MGM) relationship.

Table 18-4 Recommended maximum number of Global Mirror volume relationships

Cache size	Recommended maximum number of Global Mirror volume relationships
16 GB	4500
32 GB	4500
64 GB	9000
128 GB	18000
256 GB	32640
384 GB	32640

Important: The suggestions are solely based on the number of Global Mirror volume relationships; the capacity of the volumes is irrelevant. One way to avoid exceeding these suggestions is to use fewer, larger volumes with Global Mirror.

With a maximum number of 65280 volumes on a single DS8000 storage system, it is not possible to have more than 32640 Global Mirror secondary devices (B volumes) on the secondary system due to the requirement for the additional FlashCopy journal devices (C volumes). If you have even an extra set of FlashCopy volumes (D volumes) for testing purposes, this limit is 21760.

Adding to or removing volumes from the Global Mirror session

Volumes can be added to the session at any time after the session number is defined to the LSS where the volumes reside. After the session is started, volumes can be added to the session, or they can be removed from the session also at any time.

Volumes can be added to a session in any state, for example, *simplex* or *pending*. Volumes that have not completed their initial copy phase stay in a *join pending* state until the first initial copy is complete. If a volume in a session is *suspended*, it causes consistency group formation to fail.

We suggest that you add only Global Copy source volumes that completed their initial copy or *first pass*, although the microcode stops volumes from joining the Global Mirror session until the first pass is complete. Also, we suggest that you wait until the initial copy is complete before you create the FlashCopy relationship between the B and the C volumes.

Important: You cannot add a Metro Mirror source volume to a Global Mirror session. Global Mirror supports only Global Copy pairs. When Global Mirror detects a volume that, for example, is converted from Global Copy to Metro Mirror, the following formation of a consistency group fails.

When you add a rather large number of volumes at one time to an existing Global Mirror session, then the available resources for Global Copy within the affected ranks can be used by the initial copy pass. To minimize the impact to the production servers when you add many volumes, consider adding the volumes to an existing Global Mirror session in stages.

If you use Tivoli Storage Productivity Center for Replication, the *copy sets* can be added using the GUI or Copy Services Manager CLI. Using the GUI or Copy Services Manager CLI manages the copy so that a volume is not added to the session before the Global Copy first pass completes.

Suspending a Global Copy pair that belongs to an active Global Mirror session affects the formation of consistency groups. When you intend to remove Global Copy volumes from an active Global Mirror session, follow these steps:

1. Remove the desired volumes from the Global Mirror session.
2. Withdraw the FlashCopy relationship between the B and C volumes.
3. Terminate the Global Copy pair to bring volume A and volume B into simplex mode.

Important: When you remove A volumes without pausing Global Mirror, you might see this situation reflected as an error condition with the **`showmigr -metrics`** command, indicating that the consistency group formation failed. However, this error condition does not mean that you lost a consistent copy at the remote site, because Global Mirror does not take the FlashCopy (B to C) for the failed consistency group data. This message indicates that just one consistency group formation failed, and Global Mirror retries the sequence.

Adding or removing storage disk subsystems or LSSs

When you plan to add a subordinate storage disk subsystem to an active session, you must stop the session first. Then, add the subordinate storage disk subsystem and start the session again. The session start command then contains the new subordinate storage disk subsystem. The same procedure applies when you remove a storage disk subsystem from a Global Mirror session, which can be a subordinate only. You cannot remove the master storage disk subsystem.

When you add an LSS to an active session and this LSS belongs to a storage disk subsystem that already has another LSS that belongs to this Global Mirror session, you can add the LSS to the session without stopping and starting the session again. This situation is true for either the master or for a subordinate storage disk subsystem.

If you use Tivoli Storage Productivity Center (TPC) for Replication, the new *subsystem* can be added by using the GUI or Copy Services Manager CLI. The *paths* must then be added for the new LSS pairs. Tivoli Storage Productivity Center for Replication adds only one path if the paths are not already defined. The *copy sets* can then be added to the session after the new subsystem is recognized by Tivoli Storage Productivity Center for Replication.

Important: When using Tivoli Storage Productivity Center for Replication to manage Copy Services, do not use the DSCLI to make any configuration changes. Make changes only with the Copy Services Manager CLI (CSM CLI) or the Tivoli Storage Productivity Center for Replication GUI.

18.6 z/OS Global Mirror

z/OS Global Mirror (zGM, formerly known as XRC) is a remote data mirroring function available for the z/OS and OS/390® operating systems. It involves a host-based System Data Mover (SDM) that is a component of OS/390 and z/OS. z/OS Global Mirror maintains a copy of the data asynchronously at a remote location and can be implemented over unlimited distances. It is a combined hardware and software solution offering data integrity and data availability that can be used as part of business continuance solutions, for workload movement, and for data migration. The z/OS Global Mirror function is an optional licensed function on the DS8000 known as Remote Mirror for z/OS with function code RMZ.

For a schematic overview of z/OS Global Mirror processing, see Figure 18-17, which illustrates a simplified view of the z/OS Global Mirror components and the data flow logic.

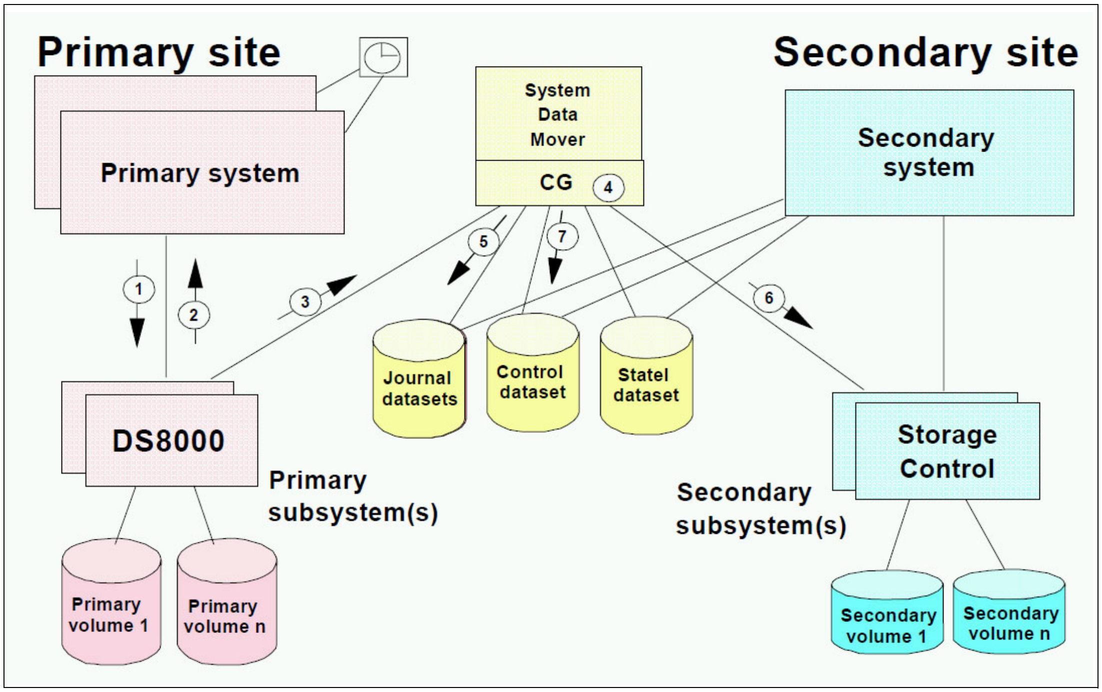


Figure 18-17 z/OS Global Mirror data flow

When a z/OS Global Mirror pair is established, the host system DFSMSdfp software starts to time stamp all later write I/Os to the primary volumes, which provides the basis for managing data consistency across multiple logical control units (LCUs). If these primary volumes are shared by systems running on different CECs, an IBM Sysplex Timer® is required to provide a common time reference for these timestamps. If all the primary systems are running in different logical partitions (LPARs) within the same CEC, the system time-of-day clock can be used.

z/OS Global Mirror is implemented in a cooperative way between the DS8000s on the primary site and the DFSMSdfp host system software component System Data Mover (SDM).

This logic explains the data flow in Figure 18-17:

1. The primary system writes to the primary volumes.
2. The application I/O operation is signaled completed when the data is written to primary DS8000 cache and NVS, which is when channel end and device end are returned to the primary system. The application write I/O operation is now complete, and the updated data is mirrored asynchronously according to the following steps.
3. The DS8000 groups the updates into record sets, which are asynchronously offloaded from the cache to the SDM system. Because z/OS Global Mirror uses this asynchronous copy technique, there is no performance impact on the primary application I/O operations.
4. The record sets, perhaps from multiple primary storage subsystems, are processed into consistency groups (CGs) by the SDM. The CG contains records that have their order of update preserved across multiple LCUs within a DS8000, across multiple DS8000s, and across other storage subsystems that participate in the same z/OS Global Mirror session.

This preservation of order is vital for dependent-write I/Os, such as databases and their logs. The creation of CGs guarantees that z/OS Global Mirror copies data to the secondary site with update sequence integrity.

5. When a CG is formed, it is written from the SDM real storage buffers to the Journal datasets.
6. Immediately after the CG is hardened on the Journal datasets, the records are written to their corresponding secondary volumes. Those records are also written from the SDM real storage buffers. Because of the data in transit between the primary and secondary sites, the currency of the data on secondary volumes lags slightly behind the currency of the data at the primary site.
7. The control dataset is updated to reflect that the records in the CG are written to the secondary volumes.

For a complete and detailed description of all z/OS Global Mirror performance and tuning options, see “z/OS Global Mirror performance options” in *DS8000 Copy Services for IBM System z*, SG24-6787. In the following sections, we present considerations for z/OS Global Mirror.

18.6.1 z/OS Global Mirror control dataset placement

The z/OS Global Mirror control datasets (the Journal, Control, and State datasets) are critical to the performance of your z/OS Global Mirror environment. The Journals are important, and you need to consider how to allocate them for the best performance.

You have two options, depending on your available space and your configuration:

- Dedicate ranks or even disk storage systems to the Journals to avoid any interference from other workloads with the Journal I/O.
- Share resources between both the Journal volumes and the secondary target volumes. Spread the updates to secondary target volumes and the updates to the Journal volumes across the maximum available resources in an ordered and balanced configuration to balance out the workload and avoid any potential hot spots.

If the control datasets are allocated on the secondary (or target) disk subsystem, you need to consider the impact of the I/O activity from the mirrored volumes on the same disk subsystem. Experience shows that placing the Journal datasets over many ranks and sharing those ranks with secondary targets works well. Also, placing the Control and State datasets with the secondary targets works well. Sharing the resources is the most common approach by current IBM z/OS Global Mirror clients today. We strongly suggest that you monitor the performance of all volumes to ensure that the environment is healthy.

18.6.2 z/OS Global Mirror tuning parameters

This section describes the tuning parameters available for z/OS Global Mirror. For additional information about the topics described in this section, see *z/OS DFSMS Advanced Copy Services,* SC35-0428.

z/OS Global Mirror provides the flexibility of allowing System Data Mover (SDM) operations to be tailored to installation requirements and also supports the modification of key parameters, either from the PARMLIB dataset or through the XSET command.

For a detailed description of z/OS Global Mirror tuning parameters, see “z/OS Global Mirror tuning parameters” in *DS8000 Copy Services for IBM System z*, SG24-6787.

18.6.3 z/OS Global Mirror enhanced multiple reader

In a zGM environment, you define a *logical session*. This logical session is made up of multiple physical sessions with at least one physical session for each logical subsystem (LSS). Each LSS contains volumes that participate in the zGM logical session. Based on the bandwidth study and the configuration of volumes in an LSS, a single LSS can have more than one physical zGM session. There is a limit of 64 physical sessions per LSS, which also includes other session types, such as Concurrent Copy.

With the zGM *single reader* implementation, you must carefully plan to balance the primary volumes' update rates for all zGM volumes in an LSS against the SDM update drain rate, because all updates for a physical session on an LSS are read by the SDM through a single SDM reader. If the updates occur at a faster rate than the rate which the SDM can offload those updates, record sets accumulate in the cache. When the cache fills up, the storage subsystem coupled with the SDM begins to execute the algorithms to start pacing or device-level blocking. Pacing or device-level blocking affects the performance of the host application. If the effect of pacing and device-level blocking is insufficient, zGM eventually suspends.

When you set up a zGM configuration during the bandwidth study, the MBps update rate for each volume is determined and the volumes are placed in an LSS based on their update rates and the associated SDM reader offload rate. Sometimes, more than one physical zGM session is required to be able to drain the updates for all the volumes that reside in an LSS. In this case, SDM must manage multiple physical sessions for the LSS.

With zGM *multiple reader* support, the SDM can now drain the record set updates off using multiple fixed utility base addresses or a single base address with aliases assigned to it. Through the single physical zGM session on an LSS, *multiple reader* paths can be used by the SDM to drain all the updates, which can reduce the number of base addresses required per LSS for zGM fixed utility addresses and can be even more dynamic in nature when HyperPAV is used. This support enables the SDM to offload the record updates on an LSS through multiple paths against the same *sidefile* while maintaining the record set Time Sequenced Order. zGM *multiple reader* support permits the SDM to balance the updates across multiple readers, enabling simpler planning for zGM.

SDM can manage a combination of physical sessions with single or multiple readers depending whether the *multiple reader* support is installed and active or not active for each subsystem involved in the zGM logical session. The DS8000 function is called z/OS Global Mirror Multiple Reader. *Multiple reader* support can also help to simplify the move to larger devices. And, it can reduce the sensitivity of zGM in draining updates as workload characteristics change or capacity growth occurs. Less manual effort is required to manage the SDM offload process.

For more information about *multiple reader,* see "Multiple Reader (enhanced readers)" in *DS8000 Copy Services for IBM System z*, SG24-6787.

18.6.4 zGM enhanced multiple reader performance improvement

The following charts show the performance improvement when you compare a zGM with a single reader to a zGM that is running with multiple readers. In this case, the zGM uses four readers. In general, the longer the distance, the better the multiple readers perform when compared to the single reader.

Figure 18-18 on page 623 shows the performance improvement when running a workload that is performing 4 KB sequential writes to a single volume. At a 3200 km (1988.3 miles) distance, the multiple reader I/O rate is three times better compared to the single reader I/O rate.

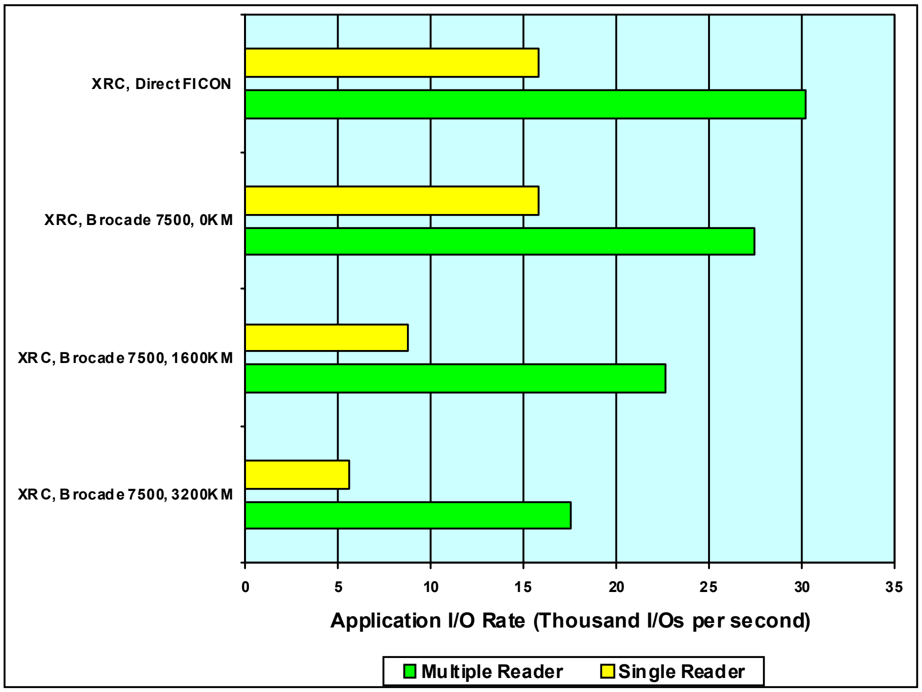


Figure 18-18 Test of 4 KB sequential write workload to one volume

Figure 18-19 shows the comparison when running a 27 KB sequential write workload to a single volume. We compare the MB per second throughput. Even though the improvement is not as dramatic as on the 4 KB sequential write workload, we still see that the multiple reader provides better performance compared to the single reader.

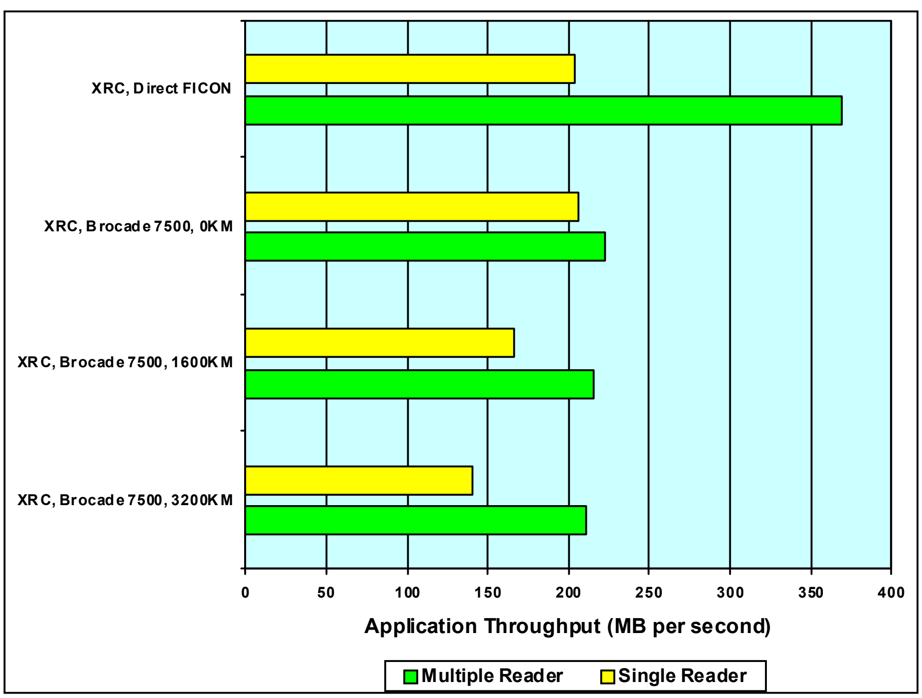


Figure 18-19 Test of 27 KB sequential write workload to one volume

Figure 18-20 on page 624 shows the benchmark result where we compare the application throughput measured by the total I/O rate when running a database random write to one LSS. We again see a significant improvement on the throughput when running with multiple readers.

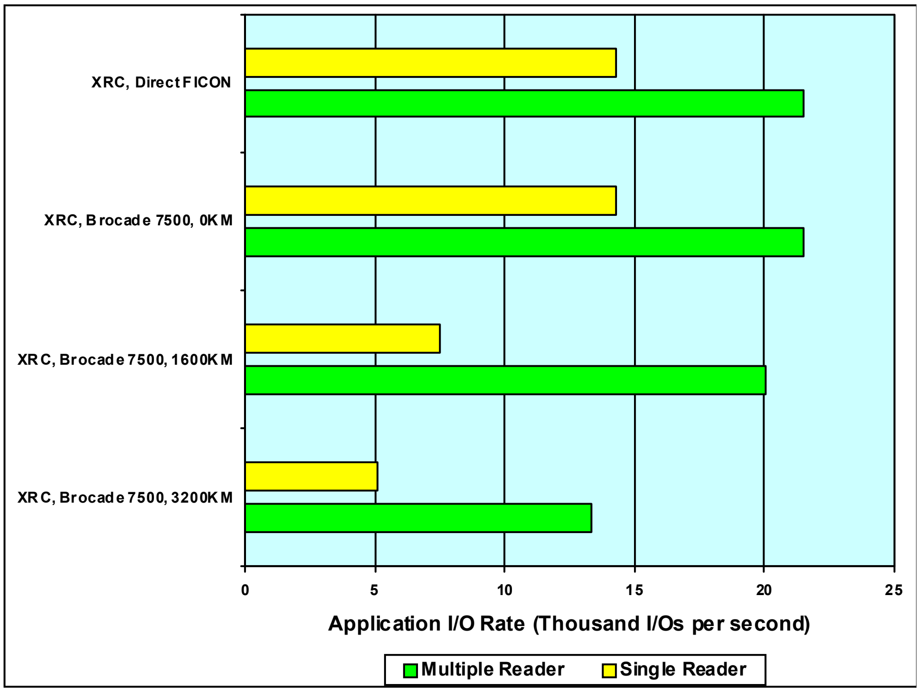


Figure 18-20 Database random write to one LSS

18.6.5 XRC Performance Monitor

The IBM XRC Performance Monitor is a licensed IBM licensed program to monitor and evaluate a z/OS Global Mirror system in tuning, system constraint determination, the evaluation of system growth, and capacity planning. For more details, see *IBM TotalStorage XRC Performance Monitor Installation and User's Guide,* GC26-7479.

18.7 Metro/Global Mirror

Metro/Global Mirror is a 3-site, multi-purpose, replication solution for both System z and Open Systems data. As shown in Figure 18-21 on page 625, Metro Mirror provides high availability replication from a local site (site A) to an intermediate site (site B). Global Mirror provides long-distance disaster recovery replication from an intermediate site (site B) to a remote site (site C).

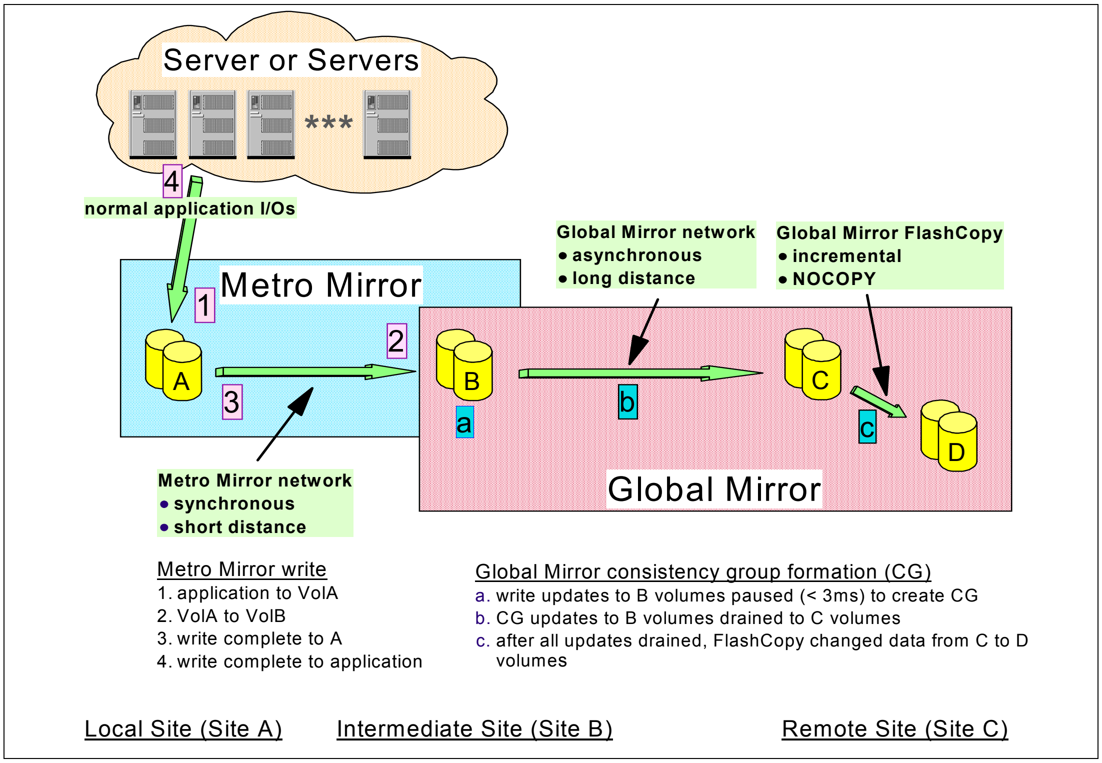


Figure 18-21 Metro/Global Mirror overview diagram

IBM offers services and solutions for the automation and management of the Metro Mirror environment, which include GDPS for System z and Tivoli Storage Productivity Center for Replication. You can obtain more details about GDPS at the following website:

http://www.ibm.com/systems/z/advantages/gdps

18.7.1 Metro/Global Mirror performance

When you configure the Metro/Global Mirror environment, you must consider both the Metro Mirror function plus the Global Mirror function. When you set up the configuration of the storage subsystems and links, consider the factors discussed earlier in this chapter. Metro/Global Mirror also has the added requirement of links from both Metro Mirror storage subsystems to the remote third site.

In a normal configuration, the synchronous copy is from A to B with the Global Mirror to C. If site B is lost, the links must already be in place from A to C to maintain the Global Mirror function. This link must provide the same bandwidth as the B to C link.

18.7.2 z/OS Metro/Global Mirror

z/OS Metro/Global Mirror (MzGM) uses z/OS Global Mirror to mirror primary site data to a remote location and also uses Metro Mirror for primary site data to a location within Metro Mirror distance limits. This approach (shown in Figure 18-22 on page 626) gives you a three-site high-availability and disaster recovery solution.

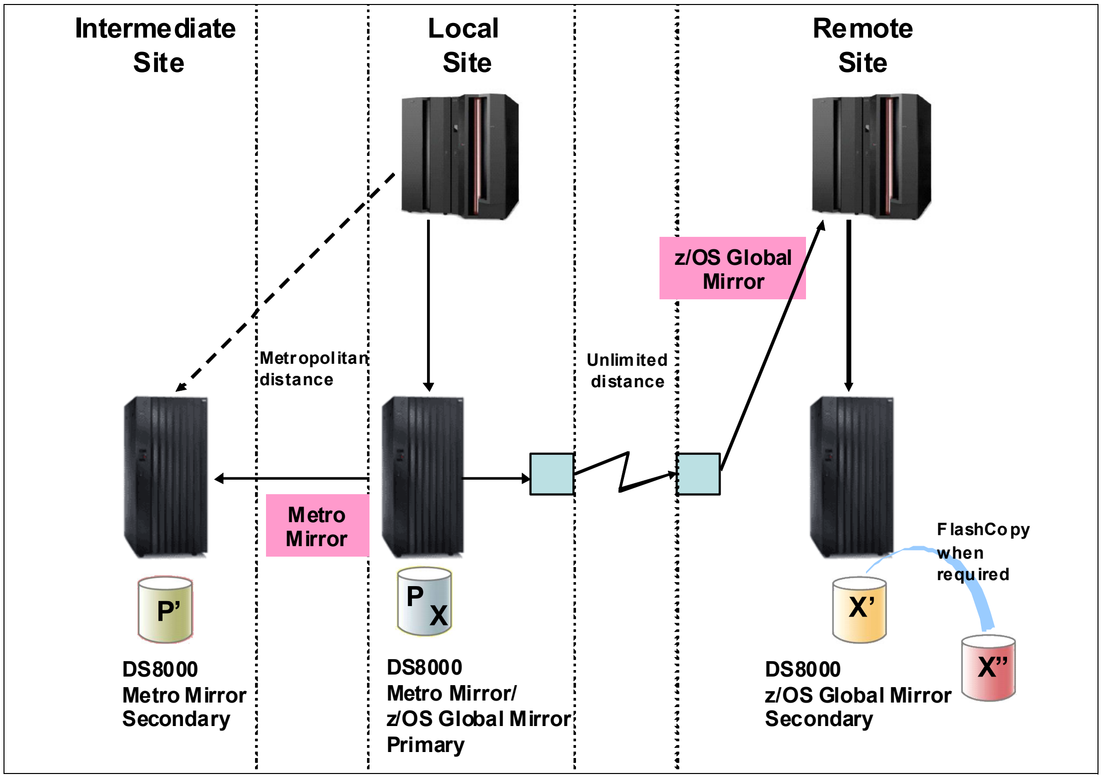


Figure 18-22 Example three-site configuration with z/OS Metro/Global Mirror

In the example that is shown in Figure 18-22, the System z environment in the Local Site is normally accessing the DS8000 disk in the Local Site. These disks are mirrored back to the Intermediate Site with Metro Mirror to another DS8000. At the same time, the Local Site disk has z/OS Global Mirror pairs established to the Remote Site to another DS8000, which can be at continental distances from the Local Site.

18.7.3 z/OS Metro/Global Mirror performance

Apply the performance considerations described in this chapter under Metro Mirror and zGM to the respective parts of the MzGM environment.

18.8 Considerations for Easy Tier and remote replication

With Easy Tier automatic mode capabilities enabled, the extent allocation within an extent pool is optimized across ranks of the same storage tier (auto rebalance) and across the available storage tiers (automatic data relocation) according to the workload access patterns.

In a remote replication setup, such as Metro Mirror, the workload differs considerably between the primary and secondary system during normal replication. Because Easy Tier monitors the reads and writes of the production workload on the primary storage system but only the write activity on the secondary storage system (because no reads occur there), it is likely that the extent distribution and performance optimization achieved by Easy Tier differ between the primary and secondary storage system.

At the time of writing this book, the learning and optimization done on the primary DS8000 system is not sent to the secondary DS8000 system to provide the same extent distribution and optimization across the ranks on both systems. In a disaster situation with a failover of the production workload from the primary to the secondary site, the data distribution across the ranks on the secondary storage system most likely are not optimized in the same way as on the primary system. Although the data distribution is already optimized for the writes during regular replication, Easy Tier must relearn about the reads of the production workload on the secondary system. Easy Tier must create and execute a new extent migration plan to optimize extent allocation across ranks and storage tiers before achieving the same level of performance as on the primary system. This task takes time for relearning and data relocation during which the performance is not as optimized as on the primary system.

However, we suggest that you maintain a symmetrical configuration in both physical and logical elements between primary and secondary storage systems in a Metro Mirror relationship when you use Easy Tier automatic mode. This approach ensures that the same level of optimization and performance can be achieved on the secondary system after the production workload is switched to the secondary site.

> **Important:** In a three-tier extent pool configuration, the cross-tier extent migration occurs only between two adjacent tiers. After a failover from the primary to a secondary, some extents that are considered hot and allocated in the Nearline tier might need more than two days to be migrated to an solid-state drive (SSD) tier.

With Global Mirror and Easy Tier automatic mode controlling the extent distribution in managed extent pools, consider allocating the B and C volumes on the recovery site in the same multi-rank extent pool. Easy Tier spreads busy extents across ranks and optimizes overall extent distribution across ranks and tiers in the extent pool based on the workload profile. The workload is balanced across the resources in the extent pool and performance bottlenecks can be avoided, which provides optimal performance. However, you can still separate workloads by using different extent pools by using the principles of workload isolation as described in 4.8, “Planning extent pools” on page 115 or a manual approach as described in ““Remote DS8000 configuration” on page 614” without Easy Tier automatic mode.

Part 5

Appendixes

This part includes the following topics:

- Performance management process
- Benchmarking
- Planning and documenting your logical configuration
- Microsoft Windows server performance log collection

A

Performance management process

This chapter describes the need for performance management and the processes and approaches that are available for managing performance on the DS8000:

- Introduction
- Purpose
- Operational performance subprocess
- Tactical performance subprocess
- Strategic performance subprocess

Introduction

The IBM System Storage DS8000 series is designed to support the most demanding business applications with its exceptional performance and superior data throughput. This strength, combined with its world-class resiliency features, makes it an ideal storage platform for supporting today's 24x7, global business environment. Moreover, with its tremendous scalability, broad server support, and flexible virtualization capabilities, the DS8000 can help simplify the storage environment and consolidate multiple storage systems onto a single DS8000 system.

This power is the potential of the DS8000 but careful planning and management are essential to realize that potential in a complex IT environment. Even a well-configured system is subject to the following changes over time that affect performance:

- Additional host systems
- Increasing workload
- Additional users
- Additional DS8000 capacity

A typical case

To demonstrate the performance management process, we look at a typical situation where DS8000 performance is an issue.

Users begin to open incident tickets to the IT Help Desk claiming that the system is slow and therefore is delaying the processing of orders from their clients and the submission of invoices. IT Support investigates and detects that there is contention in I/O to the host systems. The Performance and Capacity team is involved and analyzes performance reports together with the IT Support teams. Each IT Support team (operating system, storage, database, and application) issues its report defining the actions necessary to resolve the problem. Certain actions might have a marginal effect but are faster to implement; other actions might be more effective but need more time and resources to put in place. Among the actions, the Storage Team and Performance and Capacity Team report that additional storage capacity is required to support the I/O workload of the application and ultimately to resolve the problem. IT Support presents its findings and recommendations to the company's Business Unit, requesting application downtime to implement the changes that can be made immediately. The Business Unit accepts the report but says that it has no money for the purchase of new storage. They ask the IT department how they can ensure that the additional storage can resolve the performance issue. Additionally, the Business Unit asks the IT department why the need for additional storage capacity was not submitted as a draft proposal three months ago when the budget was finalized for next year, knowing that the system is one of the most critical systems of the company.

Incidents, such as this one, make us realize the distance that can exist between the IT department and the company's business strategy. In many cases, the IT department plays a key role in determining the company's strategy. Therefore, consider these questions:

- How can we avoid situations like those just described?
- How can we make performance management become more proactive and less reactive?
- What are best practices for performance management?
- What are the key performance indicators of the IT infrastructure and what do they mean from the business perspective?
- Are the defined performance thresholds adequate?
- How can we identify the risks in managing the performance of assets (servers, storage systems, and applications) and mitigate them?

In the following pages, we present a method to implement a performance management process. The goal is to give you ideas and insights with particular reference to the DS8000. We assume in this instance that data from IBM Tivoli Storage Productivity Center is available.

To better align the understanding between the business and the technology, we use as a guide the Information Technology Infrastructure Library (ITIL) to develop a process for performance management as applied to DS8000 performance and tuning.

Purpose

The purpose of performance management is to ensure that the performance of the IT infrastructure matches the demands of the business. The following activities are involved:

- Define and review performance baselines and thresholds
- Collect performance data from the DS8000
- Check whether the performance of the resources is within the defined thresholds
- Analyze performance using collected DS8000 performance data and tuning suggestions
- Define and review standards and IT architecture related to performance
- Analyze performance trends
- Size new storage capacity requirements

Certain activities relate to the operational activities, such as the analysis of performance of DS8000 components, and other activities relate to tactical activities, such as the performance analysis and tuning. Other activities relate to strategic activities, such as storage capacity sizing. We can split the process into three subprocesses:

- Operational performance subprocess

  Analyze the performance of DS8000 components (processor complexes, device adapters (DAs), host adapters (HAs), and ranks) and ensure that they are within the defined thresholds and service-level objectives (SLOs) and service-level agreements (SLAs).

- Tactical performance subprocess

  Analyze performance data and generate reports for tuning recommendations and the review of baselines and performance trends.

- Strategic performance subprocess

  Analyze performance data and generate reports for storage sizing and the review of standards and architectures that relate to performance.

Every process consists of the following elements:

- Inputs: Data and information required for analysis. The possible inputs are:
  - Performance data collected by Tivoli Storage Productivity Center
  - Historical performance reports
  - Product specifications (benchmark results, performance thresholds, and performance baselines)
  - User specifications (SLOs and SLAs)
- Outputs: The deliverables or results from the process. Possible types of output are:
  - Performance reports and tuning recommendations
  - Performance trends
  - Performance alerts
- Tasks: The activities that are the smallest unit of work of a process. These tasks can be:

   - Performance data collection
   - Performance report generation
   - Analysis and tuning recommendations
- Actors: A department or person in the organization that is specialized to perform a certain type of work. Actors can vary from organization to organization. In smaller organizations, a single person can own multiple actor responsibilities, for example:
   - Capacity and performance team
   - Storage team
   - Server teams
   - Database team
   - Application team
   - Operations team
   - IT Architect
   - IT Manager
   - Clients
- Roles: The tasks that need to be executed by an actor, but another actor might own the activity, and other actors might just be consulted. Knowing the roles is helpful when you define the steps of the process and who is going to do what. The roles can be:
   - Responsible: The person that executes that task but is not necessarily the owner of that task. Suppose that the capacity team is the owner for the generation of the performance report with the tuning recommendations, but the specialist with the skill to suggest tuning in the DS8000 is the Storage Administrator.
   - Accountable: The owner of that activity. There can only be one owner.
   - Consulted: The people that are consulted and whose opinions are considered. Suppose that the IT Architect proposes a new architecture for the storage. Normally, the opinion of the Storage Administrator is requested.
   - Informed: The people who are kept up-to-date on progress. The IT Manager normally wants to know the evolution of activities.

When assigning the tasks, you can use a Responsible, Accountable, Consulted, and Informed (RACI) matrix to list the actors and the roles that are necessary to define a process or subprocess. A RACI diagram, or RACI matrix, is used to describe the roles and responsibilities of various teams or people to deliver a project or perform an operation. It is useful in clarifying roles and responsibilities in cross-functional and cross-departmental projects and processes.

Operational performance subprocess

The operational performance subprocess relates to daily activities that can be executed every few minutes, hourly, or daily. For example, you can monitor the DS8000 storage system, and, if there is utilization above a defined threshold, automatically send an alert to the designated person.

With Tivoli Storage Productivity Center, you can set performance thresholds for two major categories:

- Status change alerts
- Configuration change alerts

Important: Tivoli Storage Productivity Center for Disk is not designed to monitor hardware nor to report hardware failures. You can configure the DS8000 Hardware Management Console (HMC) to send alerts via Simple Network Management Protocol (SNMP) or email when a hardware failure occurs.

You might also need to compare the DS8000 performance with the users' performance requirements. Often, these requirements are explicitly defined in formal agreements between IT management and user management. These agreements are referred to as service-level agreements (SLA) or service-level objectives (SLO). These agreements provide a framework for measuring IT resource performance requirements against IT resource fulfillment.

Performance SLA

A performance SLA is a formal agreement between IT Management and User representatives concerning the performance of the IT resources. Often, these SLAs provide goals for end-to-end transaction response times. For storage, these types of goals typically relate to average disk response times for different types of storage. Missing the technical goals described in the SLA results in financial penalties to the IT service management providers.

Performance SLO

Performance SLOs are similar to SLAs with the exception that misses do not carry financial penalties. While SLO misses do not carry financial penalties, misses are a breach of contract in many cases and can lead to serious consequences if not remedied.

Having reports that show you how many alerts and how many misses in SLOs/SLAs occurred over time is important. The reports tell how effective your storage strategy is (standards, architectures, and policy allocation) in the steady state. In fact, the numbers in those reports are inversely proportional to the effectiveness of your storage strategy. The more effective your storage strategy, the fewer performance threshold alerts are registered and the fewer SLO/SLA targets are missed.

It is not necessary to implement SLOs or SLAs for you to discover the effectiveness of your current storage strategy. The definition of SLO/SLA requires a deep and clear understanding of your storage strategy and how well your DS8000 is running. That is why, before implementing this process, we suggest that you start with the tactical performance process:

- Generate the performance reports
- Define tuning suggestions
- Review the baseline after implementing tuning recommendations
- Generate performance trends reports

Then, redefine the thresholds with fresh performance numbers. The failure to redefine the thresholds with fresh performance numbers causes you to spend time dealing with performance incident tickets with false-positive alerts and not spend the time analyzing the performance and suggesting tuning for your DS8000. Let us look at the characteristics of this process.

Inputs

The following inputs are necessary to make this process effective:

- Performance trends reports of DS8000 components: Many people ask for the IBM recommended thresholds. In our opinion, the best recommended thresholds are those thresholds that fit your environment. The best thresholds depend on the configuration of your DS8000 and the types of workloads. For example, you need to define thresholds for I/O per second (IOPS) if your application is a transactional system. If the application is a data warehouse, you need to define thresholds for throughput. Also, you must not expect the same performance from different ranks where one set of ranks has 300 GB, 15K revolutions per minute (rpm) disk drive modules (DDMs) and another set of ranks has 3 TB, 7200 rpm nearline DDMs. Check the outputs generated from the tactical performance subprocess for additional information.

- Performance SLO and performance SLA: You can define the SLO/SLA requirements in two ways:
  - By hardware (IOPS by rank or MB/s by port): This performance report is the easiest way to implement an SLO or SLA but the most difficult method for which to get client agreement. The client normally does not understand the technical aspects of a DS8000.
  - By host or application (IOPS by system or MB/s by host): Most probably this performance report is the only way that you are going to get an agreement from the client but this agreement is not certain. The client sometimes does not understand the technical aspects of IT infrastructure. The typical way to define a performance SLA is by the average execution time or response time of a transaction in the application. So, the performance SLA/SLO for the DS8000 is normally an internal agreement among the support teams, which creates additional work for you to generate those reports, and there is no *predefined* solution. It depends on your environment's configuration and the conditions that define those SLOs/SLAs. We suggest that when configuring the DS8000 with SLO/SLA requirements that you separate the applications or hosts by LSS (reserve two LSSs, one even and one odd, for each host, system, or instance). The benefit of generating performance reports using this method is that they are more meaningful to the other support teams and to the client. So, the level of communication increases and reduce chances for misunderstandings.

Important: When defining a DS8000 related SLA or SLO, ensure that the goals are based on empirical evidence of performance within the environment. Application architects with applications that are highly sensitive to changes in I/O throughput or response time need to consider the measurement of percentiles or standard deviations as opposed to average values over an extended period. IT management must ensure that the technical requirements are appropriate for the technology.

In cases where contractual penalties are associated with production performance SLA or SLO misses, be careful in the management and implementation of the DS8000. Even in the cases where no SLA or SLO exists, users have performance expectations that are not formally communicated. In these cases, they let IT management know when the performance of the IT resources is not meeting their expectations. Unfortunately, by the time they communicate their missed expectations, they are often frustrated, and their ability to manage their business is severely affected by performance issues.

Although there might not be any immediate financial penalties associated with missed user expectations, prolonged negative experiences with underperforming IT resources result in low user satisfaction.

Outputs

The following outputs are generated by this process:

- Documentation of defined DS8000 performance thresholds. It is important to document the agreed-to thresholds. Not just for you, but also for other members of your team or other teams that need to know.
- DS8000 alerts for performance utilization. These alerts are generated when a DS8000 component reaches a defined level of utilization. With Tivoli Storage Productivity Center for Disk, you can automate the performance data collection and also configure Tivoli Storage Productivity Center to send an alert when this type of an event occurs.
- Performance reports comparing the performance utilization of DS8000 with the performance SLO and SLA.

Tasks, actors, and roles

It is easier to visualize and understand the tasks, actors, and roles when they are combined using a Responsible, Accountable, Consulted, and kept Informed (RACI) matrix, as shown in Figure A-1.

Tasks / Actors	Capacity & Performance team	Storage Support team	Monitoring team
Provide the baseline performance of the systems or hosts	A	R	
Define what thresholds to monitor and their respective values, severity, queue to open the ticket and additional instructions		RA	
Implement the performance monitoring		RA	
Implement the alerts		RA	
Publish the documentation among the involved areas	I	RA	I

Figure A-1 Operational tasks, actors, and roles

Figure A-1 is an example of a RACI matrix for the operational performance subprocess, with all the tasks, actors, and roles identified and defined:

- Provide performance trends report: This report is an important input for the operational performance subprocess. With this data, you can identify and define the thresholds that best fit your DS8000. Consider how the workload is distributed between of the internal components of the DS8000: HAs, processor complexes, DAs, and ranks. This analysis avoids the definition of thresholds that generate false-positive performance alerts and ensure that you monitor only what is relevant to your environment.
- Define the thresholds to be monitored and their respective values, severity, queue to open the ticket, and additional instructions: In this task, using the baseline performance report, you can identify and set the relevant threshold values. You can use Tivoli Storage Productivity Center to create alerts when these thresholds are exceeded. For example, you can configure Tivoli Storage Productivity Center to send the alerts through Simple Network Management Protocol (SNMP) traps to Tivoli Enterprise Console (TEC) or through email. However, the opening of an incident ticket needs to be performed by the Monitoring team that needs to know the severity to set, on which queue to open the ticket,

and any additional information that is required in the ticket. Figure A-2 is an example of the required details.

Component	Metric	Threshold	Severity	Triggered action	Action	Additional information
Array	Disk Utilization Percentage	50	Warning Stress	TEC	Open a ticket with severity 4 to Storage team, queue XYZ.	N/A
Array	Disk Utilization Percentage	80	Critical Stress	TEC	Open a ticket with severity 3 to Storage team, queue XYZ.	N/A
Controller	Total I/O Rate (overall)	38000	Warning Stress	TEC	Open a ticket with severity 4 to Storage team, queue XYZ.	N/A
Port	Total Port Data Rate	200	Warning Stress	TEC	Open a ticket with severity 4 to Storage team, queue XYZ.	N/A

Figure A-2 Thresholds definitions table

- Implement performance monitoring and alerting: After you define the DS8000 components to monitor, set their corresponding threshold values. For detailed information about how to configure Tivoli Storage Productivity Center, see the IBM Tivoli Storage Productivity Center documentation:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/index.jsp

- Publish the documentation to the IT Management team: After you implement the monitoring, send the respective documentation to those people who need to know.

Performance troubleshooting

If an incident ticket is open for performance issues, you might be asked to investigate. The following tips can help during your problem determination.

Sample questions for an AIX host

The following questions and comments are examples of the types of questions to ask when engaged to analyze a performance issue. They might not all be appropriate for every environment:

- What was running during the sample period (backups, production batch, or online queries)?
- Describe your application (complex data warehouse, online order fulfillment, or DB2).
- Explain the types of delays experienced.
- What other factors might indicate a storage area network (SAN)-related or DS8000 related I/O issue?
- Describe any recent changes or upgrades to the application, operating system, microcode, or database.
- When did the issue start and is there a known frequency (specify the time zone)?
- Does the error report show any disk, logical volume (LV), HA, or other I/O-type errors?
- What is the interpolicy of the logical volumes (maximum or minimum)?
- Describe any striping, mirroring, or RAID configurations in the affected LVs.
- Is this workload a production or development workload, and is it associated with benchmarking or breakpoint testing?

In addition to the answers to these questions, the client must provide server performance and configuration data. See the relevant host chapters in this book for more detail.

Identify problems and recommend a fix

To identify problems in the environment, the storage and performance management team must have the tools to monitor, collect, and analyze performance data. While the tools might vary, these processes are worthless without storage resource management tools. The following sample process provides a way to correct a DS8000 disk bottleneck:

1. The performance management team identifies the hot RAID array and the logical unit number (LUN) to which you need to migrate to alleviate the disk array bottleneck.
2. The performance team identifies the target RAID array with low utilization.
3. The client using the hot LUN is contacted and requested to open a change request to allocate a new LUN on a lesser utilized array.
4. The client opens a ticket and requests a new LUN.
5. The storage management team defines and zones the new LUN.
6. The client migrates data from the old LUN to the new LUN. The specifics of this step are operating system (OS)-specific and application-specific and are not explained.
7. The client opens a change request to delete the old LUN.
8. Performance management evaluates the change and confirms success. If the disk array still has disk contention, further changes might be recommended.

Tactical performance subprocess

This process deals with activities that occur over a cycle of weeks or months. These activities relate to the collection of data and the generation of reports for performance and utilization trends. With these reports, you can produce tuning recommendations. With this process, you can gain a better understanding of the workload of each application or host system. You can also verify that you are getting the expected benefits of new tuning recommendations implemented on the DS8000. The performance reports generated by this process are also used as inputs for the operational and strategic performance subprocesses.

> **Tip:** We suggest the tactical performance subprocess as the starting point for the implementation of a performance management process.

Regardless of the environment, the implementation of proactive processes to identify potential performance issues before they become a crisis saves time and money. All the methods that we describe depend on storage management reporting tools. These tools must provide for the long-term gathering of performance metrics, allow thresholds to be set, and provide for alerts to be sent. These capabilities permit Performance Management to effectively analyze the I/O workload and establish proactive processes to manage potential problems. Key performance indicators provide information to identify consistent performance hot spots or workload imbalances.

Inputs

The following inputs are necessary to make this process effective:

- Product specifications: Documents that describe the characteristics and features of the DS8000, such as data sheets, Announcement Letters, and planning manuals
- Product documentation: Documents that provide information about the installation and use of the DS8000, such as user manuals, white papers, and IBM Redbooks publications
- Performance SLOs/performance SLAs: The documentation of performance SLO/SLAs to which the client agreed for DS8000.

Outputs

Performance reports with tuning recommendations and performance trends reports are the outputs that are generated by this process.

Performance reports with tuning recommendations

We recommend that you create your performance report with three chapters at least:

- Hardware view: This view provides information about the performance of each component of the DS8000. You can determine the health of the storage subsystem by analyzing key performance metrics for the HA, port, array, and volume. You can generate workload profiles at the DS8000 subsystem or component level by gathering key performance metrics for each component.
- Application or host system view: This view helps you identify the workload profile of each application or host system that accesses the DS8000. This view helps you understand which RAID configuration performs best or whether it is advisable to move a specific system from ranks with DDMs of 300 GB/15K rpm to ranks with DDMs of 900 GB/10K rpm, for example. This information also helps other support teams, such as Database Administrators, the IT Architect, the IT Manager, or even the clients. This type of report can assist you in meetings to become more interactive with more people asking questions.
- Conclusions and recommendations: This section is the information that the IT Manager, the clients, and the IT Architect read first. Based on the recommendations, agreement can be reached about actions to implement to mitigate any performance issue.

Performance trends reports

It is important for you to see how the performance of the DS8000 is changing over time. As in the performance report with tuning recommendations section, we recommend that you create three chapters:

- Hardware view: Analysis of the key performance indicators referenced in the 7.3, “Tivoli Storage Productivity Center data collection” on page 250.
- Application or host system view: One technique for creating workload profiles is to group volume performance data into logical categories based on their application or host system.
- Conclusions and recommendations: The reports might help you recommend changing the threshold values for your DS8000 performance monitoring or alerting. The report might show that a specific DS8000 component will soon reach its capacity or performance limit.

Tasks, actors, and roles

Figure A-3 on page 641 shows an example of a RACI matrix for the tactical performance subprocess with all the tasks, actors, and roles identified and defined.

Tasks \ Actors	Customer	IT Manager	IT Architect	Capacity & Performance team	Storage Support team	Windows team	UNIX team	DBA team	Middleware team	Basis team
Collect configuration and raw performance data				A	R					
Generate performance graphics				RA						
Generate performance reports with tuning recommendations				RA	R	R	R			
Generate performance reports for trend analysis				RA	R					
Schedule meetings with the involved areas	C	C	C	A	R	C	C	C	C	C

Figure A-3 Tactical tasks, actors, and roles

The RACI matrix includes the following tasks:

- Collect configuration and raw performance data: Use Tivoli Storage Productivity Center for Disk daily probes to collect configuration data. Set up the Subsystem Performance Monitor to run indefinitely and to collect data at 15-minute intervals for each DS8000.
- Generate performance graphics: Produce one key metric for each physical component in the DS8000 over time. For example, if your workload is evenly distributed across the entire day, show the average daily disk utilization for each disk array. Configure thresholds in the chart to identify when a performance constraint might occur. Use a spreadsheet to create a linear trend line based on the data previously collected and identify when a constraint might occur.
- Generate performance reports with tuning recommendations: You must collect and review host performance data on a regular basis. On discovery of a performance issue, Performance Management must work with the Storage team and the client to develop a plan to resolve it. This plan can involve some form of data migration on the DS8000. For key I/O-related performance metrics, see the chapters in this book for your specific operating system. Typically, the end-to-end I/O response time is measured, because it provides the most direct measurement of the health of the SAN and disk subsystem.
- Generate performance reports for trend analysis: Methodologies typically applied in capacity planning can be applied to the storage performance arena. These methodologies rely on workload characterization, historical trends, linear trending techniques, I/O workload modeling, and "What if" scenarios. You can obtain details about disk management in Chapter 6, "Performance planning tools" on page 175.
- Schedule meetings with the involved areas: The frequency depends on the dynamism of your IT environment. The greater the rate of change, such as the deployment of new systems, upgrades of software and hardware, fix management, allocation of new LUNs, and the implementation of Copy Services, determines the frequency of meetings. For the performance reports with tuning recommendations, we suggest weekly meetings. For performance trends reports, we suggest meetings on a monthly basis. You might want to change the frequency of these meetings after you gain more confidence and familiarity with the performance management process. At the end of these meetings, define with the other support teams and the IT Manager the actions to resolve any potential issues that are identified.

Strategic performance subprocess

The strategic performance subprocess relates to activities that occur over a cycle of six to 12 months. These activities define or review standards and architecture or size new or existing DS8000s.

It might be an obvious observation, but it is important to remember that the IT resources are finite and some day they will run out. In the same way, the money to invest in IT Infrastructure is limited, which is why this process is important. In each company, there is normally a time when the budget for the next year is decided. So, even if you present a list of requirements with performance reports to justify the investments, you might not be successful. The timing of the request and the benefit of the investment to the business are also important considerations.

Just keeping the IT systems up and running is not enough. The IT Manager and Chief Information Officer (CIO) need to show business benefit for the company. Usually, this benefit means providing the service at the lowest cost but also showing a financial advantage that the services provide. This benefit is how the IT industry grew over the years while it increased productivity, reduced costs, and enabled new opportunities.

You need to check with your IT Manager or Architect to learn when the budget is set and start three to four months before this date. You can then define the priorities for the IT infrastructure for the coming year to meet the business requirements.

Inputs

The following inputs are required to make this process effective:

- Performance reports with tuning recommendations
- Performance trends reports

Outputs

The following outputs are generated by this process:

- Standards and architectures: Documents that specify:
  - Naming convention for the DS8000 components: ranks, extent pools, volume groups, host connections, and LUNs.
  - Rules to format and configure the DS8000: Arrays, RAID, ranks, extent pools, volume groups, host connections, logical subsystems (LSSs), and LUNs.
  - Policy allocation: When to pool the applications or host systems on the same set of ranks. When to segment the applications or hosts systems in different ranks. Which type of workload must use RAID 5, RAID 6, or RAID 10? Which type of workload must use solid-state drives (SSDs) or DDMs of 146 GB/15K rpm, 450 GB/10K rpm, or 900 GB/10K rpm?
- Sizing of new or existing DS8000: According to the business demands, what are the recommended capacity, cache, and host ports for a new or existing DS8000?
- Plan configuration of new DS8000: What is the planned configuration of the new DS8000 based on your standards and architecture and according to the workload of the systems that will be deployed?

Tasks, actors, and roles

Figure A-4 is an example of a RACI matrix for the strategic performance subprocess.

Tasks \ Actors	IT Manager	IT Architect	Capacity & Performance team	Storage Support team
Define priorities of new investments	C	RA	R	R
Define/review standards and architectures		RA		R
Size new or existing DS8000		A		R
Plan configuration of new DS8000				RA

Figure A-4 Strategic tasks, actors, and roles

Figure A-4 is an example of the RACI matrix for the strategic performance subprocess with all the tasks, actors, and roles identified and defined:

- Define priorities of new investments: In defining the priorities of where to invest, you must consider these four objectives:
  - Reduce cost: The simplest example is storage consolidation. There might be several storage systems in your data center that are nearing the ends of their useful lives. The costs of maintenance are increasing, and the storage subsystems use more energy than new models. The IT Architect can create a case for storage consolidation but needs your help to specify and size the new storage.
  - Increase availability: There are production systems that need to be available 24x7. The IT Architect needs to submit a new solution for this case to provide data mirroring. The IT Architect requires your help to specify the new storage for the secondary site and to provide figures for the necessary performance.
  - Mitigate risks: Consider a case where a system is running on an old storage model without a support contract from the vendor. That system started as a pilot with no importance. Over time, that system presented great performance and is now a key application for the company. The IT Architect needs to submit a proposal to migrate to a new storage system. Again, the IT Architect needs your help to specify the new storage requirements.
  - Business units' demands: Depending on the target results that each business unit must meet, the business units might require additional IT resources. The IT Architect requires information about the additional capacity that is required.
- Define and review standards and architectures: After you define the priorities, you might need to review the standards and architecture. New technologies appear so you might need to specify new standards for new storage models. Or maybe, after a period analyzing the performance of your DS8000, you discover that for a certain workload, you might need to change a standard.
- Size new or existing DS8000: Modeling tools, such as Disk Magic, which is described in 6.1.6, "Disk Magic modeling" on page 179, can gather multiple workload profiles based on host performance data into one model and provide a method to assess the impact of one or more changes to the I/O workload or DS8000 configuration.

> **Tip:** For environments with multiple applications on the same physical servers or on logical partitions (LPARs) using the same Virtual I/O servers, defining new requirements can be challenging. We suggest building profiles at the DS8000 level first and eventually moving into more in-depth study and understanding of the other shared resources in the environment.

- Plan configuration of new DS8000: Configuring the DS8000 to meet the specific I/O performance requirements of an application reduces the probability of production performance issues. To produce a design to meet these requirements, Storage Management needs to know:
  - I/Os per second
  - Read to write ratios
  - I/O transfer size
  - Access type: Sequential or random

  For help in translating application profiles to I/O workload, see Chapter 5, “Understanding your workload” on page 157.

  After the I/O requirements are identified, documented, and agreed upon, the DS8000 layout and logical planning can begin. For additional detail and considerations for planning for performance, see Chapter 4, “Logical configuration performance considerations” on page 87.

> **Communication:** A lack of communication between the Application Architects and the Storage Management team regarding I/O requirements can likely result in production performance issues. It is essential that these requirements are clearly defined.

For existing applications, you can use Disk Magic to analyze an application I/O profile. Details about Disk Management are in Chapter 6, “Performance planning tools” on page 175.

B

Benchmarking

Benchmarking storage systems is complex due to all of the hardware and software that are used for storage systems. In this appendix, we discuss the goals and the ways to conduct an effective storage benchmark.

Goals of benchmarking

Today, clients face difficult choices about the number of storage vendors and their product portfolios. Performance information provided by storage vendors can be generic and often not representative of real client environments. Benchmarking can help you make decisions, because it is a way to get an accurate representation of the storage product performance in simulated application environments. The main objective of benchmarking is to identify performance capabilities of a specific production environment and compare the performance of two or more storage systems. Including the use of real production data in the benchmark can be ideal.

To conduct a benchmark, you need a solid understanding of all of the parts of your environment. This understanding includes the storage system requirements and also the storage area network (SAN) infrastructure, the server environments, and the applications. Emulating the actual environment, including actual applications and data, along with user simulation, provides efficient and accurate analysis of the performance of the storage system tested. The characteristic of a performance benchmark test is that results must be reproducible to validate the integrity of the test.

Benchmark key indicators

The popularity of a benchmark is based on how representative the workload is and whether the results are meaningful. Three key indicators can be used out of the benchmark results to evaluate the performance of the storage system:

- Performance results in a real application environment
- Reliability
- Total cost of ownership (TCO)

Performance is not the only component to consider in benchmark results. Reliability and cost-effectiveness must be considered. Balancing benchmark performance results with reliability, functionalities, and TCO of the storage system provides a global view of the storage product value.

To help client understanding of intrinsic storage product values in the marketplace, vendor-neutral independent organizations developed several generic benchmarks. The following organizations are the two most famous organizations:

- Storage Performance Council (SPC):

  `http://www.storageperformance.org`

- Transaction Processing Performance Council (TPC):

  `http://www.tpc.org`

The popularity of these benchmarks depends on how meaningful the workload is compared to the main and new workloads that companies deploy today. If the generic benchmark workloads are representative of your production, you can use the different benchmark results to identify the product you implement in your production environment. But, if the generic benchmark definition is not representative or does not include your requirements or restrictions, running a dedicated benchmark designed to be representative of your workload provides information to help you choose the right storage system.

Requirements for a benchmark

You need to carefully review your requirements before you set up a storage benchmark and use these requirements to develop a detailed but reasonable benchmark specification and time frame. Furthermore, you need to clearly identify the objective of the benchmark with all the participants and precisely define the success criteria of the results.

Defining the benchmark architecture

This process includes the specific storage equipment that you want to test, the servers that host your application, the servers used to generate the workload, and the SAN equipment used to interconnect the servers and the storage subsystem. The monitoring equipment and software are also part of the benchmark architecture.

Defining the benchmark workload

Your application environment can have different categories of data processing. In most cases, two data processing types can be identified: One is characterized as an online transaction processing (OLTP) type and the other type as batch processing.

The OLTP category typically has many users, who all access the same disk storage subsystem and a common set of files. The requests are typically spread across many files; therefore, the file sizes are typically small and randomly accessed. Typical applications consist of a network file server or disk subsystem that is accessed by a sales department that enters order information.

Batch workloads are frequently a mixture of random database accesses, skip-sequential, pure sequential, and sorting. They generate large data transfers and result in high path utilizations. Often constrained to operate within a particular window of time, during which time online operation is restricted or shut down. Poorer or better performance is often not recognized unless it affects this window.

To identify the specificity of your production workload, you can use monitoring tools that are available at the operating system level.

In a benchmark environment, there are two ways to generate the workload.

The first way to generate the workload, which is the most complex, is to set up the production environment, including the applications software and the application data. In this case, you must ensure that the application is well-configured and optimized on the server operating system. The data volume also must be representative of the production environment. Depending on your application, workload can be generated by using application scripts or an external transaction simulation tool. These tools provide a simulation of users accessing your application. You use workload tools to provide application stress from end-to-end. To configure an external simulation tool, you first record a standard request from a single user and then generate this request several times. This process can provide an emulation of hundreds or thousands of concurrent users to put the application through the rigors of real-life user loads and measure the response times of key business processes. Examples of available software include IBM Rational® Software and Mercury Load Runner TM.

The other way to generate the workload is to use a standard workload generator. These tools, specific to each operating system, produce different kinds of workloads. You can configure and tune these tools to match your application workload. The main tuning parameters include the type of workload (sequential or random), the read/write ratio, the I/O blocksize, the number of I/Os per second, and the test duration. With a minimum of setup, these simulation tools can help you to re-create your production environment workload without setting up all of the software components. Examples of available software include iozone, iometer, and others.

> **Important:** Each workload test must be defined with a minimum time duration in order to eliminate any side effects or warm-up period, such as populating cache, which can generate incorrect results.

Monitoring the performance

Monitoring is a critical component of benchmarking and must be fully integrated into the benchmark architecture. The more information about component activity at each level of a benchmark environment, the more we understand where the solution weaknesses are. With this critical source of information, we can precisely identify bottlenecks and can optimize component utilization and improve the configuration.

A minimum of monitoring tools are required at different levels in a storage benchmark architecture:

- Storage level: Monitoring at the storage level provides information of intrinsic performance of the storage equipment components. Most of these monitoring tools report storage server utilization, storage cache utilization, volume performance, RAID array performance, disk drive module (DDM) performance, and adapter utilization.
- SAN level: Monitoring at the SAN level provides information of the interconnect workloads between servers and storage subsystems. This monitoring helps to check that workload is balanced between the different paths used for production and to verify that the interconnection is not a bottleneck in performance.
- Server level: Monitoring at the server level provides information about server component utilization (processor, memory, storage adapter, and filesystem). This monitoring helps you also to understand the type of workload that the application hosted on the server is generating and evaluate the storage performance in response time and bandwidth from the application point of view.
- Application level: This monitoring is the most powerful tool in performance analysis, because these tools monitor the performance at the user point of view and highlight bottlenecks of the entire solution. Monitoring the application is not always possible; only a few applications provide a performance module for monitoring application processes.

Monitoring can have an impact on component performance. In that case, implement the monitoring tools in the first sequence of tests to understand your workload and then disable it to eliminate any impact, which can distort performance results.

Defining the benchmark time frame

Consider the following information when you plan and define the benchmark:

- Time to set up the environment (hardware and software), restore your data, and validate that the solution works.
- Time of execution of each scenario, considering that each scenario must be run several times.

- Time to analyze the monitoring data that is collected.
- After each run, benchmark data can be changed, inserted, deleted, or otherwise modified so that it must be restored before another test iteration. In that case, consider the time needed to restore the original data after each run.

During a benchmark, each scenario must be run several times to understand how the different components perform by using monitoring tools, to identify bottlenecks, and then, to test different ways to get an overall performance improvement by tuning each component.

Use caution when using benchmark results to design production

While using benchmark results as the foundation to build your production environment infrastructure, you must look closely at the benchmark performance information and watch out for the following points:

- Benchmark hardware resources (servers, storage subsystems, SAN equipment, and network) are dedicated only for this performance test.
- Benchmark infrastructure configuration must be fully documented. Ensure that there are no dissimilar configurations (for example, dissimilar cache sizes, or different disk capacities/speeds).
- Benchmarks are focused on the core application performance and do not often include interference with other applications that can occur in the real infrastructure.
- The benchmark configuration must be realistic (a technical choice about performance compared to usability or availability).
- The scenarios that are built must be representative in different ways:
  - Volume of data
  - Extreme or unrealistic workload
  - Timing execution
  - Workload ramp-up
  - Avoid *side effects,* such as populating cache, which can generate incorrect results
- Document the detailed optimization actions performed on each of the components of the infrastructure (including application, servers, and storage subsystems).
- Servers and storage performance reports must be fully detailed, including bandwidth, I/O per second, and also response time.
- Understand the availability level of the solution tested, considering the impact of each component failure (host bus adapter (HBA), switch, and RAID).
- Balance the performance results with your high availability requirements. First, consider the advanced Copy Services available on the specific storage subsystem to duplicate or mirror your production data within the subsystem or to a remote backup site. Then, watch out that performance is not equivalent if your data is mirrored or duplicated.

C

Planning and documenting your logical configuration

Large disk storage system configurations must be well-documented to be well-managed. The documentation process begins with the initial system installation and configuration process. It continues throughout the lifecycle of the system, because changes are often made to support requests for additional capacity to “attach” to existing or new servers.

We start with an example out of the Capacity Magic tool, to describe how we plan the extent pool and divide the machine between mainframe and Open Systems pools. We provide examples about how to structure the multi-tier Open Systems pools into either large 3-tier pools. We provide an example with more isolation into a higher-performance tier and a lower-performance tier that are separate from each other.

We also discuss the choice of the performance tools to track the configuration, and we present in more detail the DS8000 Query and Reporting tool (DSQTOOL).

Considering hardware resources for throughput

An important factor in logically configuring your DS8000 for optimal performance throughput is to assess your hardware resource availability. An adequate sizing, which is based on your current workloads and workload increase predictions, assisted by the IBM functions and by using the Disk Magic tool, is key. Take inventory of the amount of cache, number of device adapter (DA) pairs, ranks (arrays and array sites of the various tiers), their affinity to a specific server complex, the number of I/O enclosures, the number of host adapters (HAs), and the number of I/O ports available on each adapter.

A general idea is to evenly distribute the data across as many hardware resources as possible without oversaturating those hardware resources through an imbalanced configuration. Easy Tier performs a major part of this balancing automatically for you. Still, you might need to review your system and your choices after the sizing is complete and the machine is available.

You must size for both performance and for capacity needs. Most often, the System z load and the Open Systems load are on the new box. Clients often consolidate several older storage systems into one new storage system. You must plan the extent pools separately for System z and Open Systems, so for both environments, you need to consider SSD, Enterprise, and nearline ranks.

Figure C-1 shows a Capacity Magic view of a 2-frame 4-way DS8800, that contains a mix of SSD, Enterprise and nearline drives, for both count key data (CKD) and Open Systems.

Figure C-1 Capacity Magic view of a 2-frame DS8800

Capacity Magic is always a good start for any planning, because it shows in detail the number of spares used on the various RAID arrays, the net capacities that we can expect, and which DA pair is used for each rank.

For instance, we can see that the System z arrays (which show a small green “z” in their upper corner) are bound to DA pairs 0 and 3 only (enclosures 3 - 6 in base frame). The System z arrays are isolated from the Open Systems arrays and the System z DA pairs are isolated from the Open Systems DA pairs. Therefore, there is no interference between Open Systems and mainframe loads.

This view can be transferred and brought into a Lotus® Symphony™ or MS Excel spreadsheet, as shown in Figure C-2.

2-frame 4-way DS8800

FRAME A

DA							
2	RAID-6 (5+P+P+S)	RAID-6 (5+P+P+S)	RAID-6 (5+P+P+S)				Drive Types: 3 TB 3.5" 7.2k
1	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 450 GB 2.5" 10 k
3	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 450 GB 2.5" 10 k
0	RAID-5 (6+P+S)	RAID-5 (6+P+S)	Empty	Empty	Empty	Empty	Drive Types: 300 GB SSD
2	RAID-5 (6+P+S)	RAID-5 (6+P+S)	Empty	Empty	Empty	Empty	Drive Types: 300 GB SSD

FRAME B

DA							
7	RAID-6 (5+P+P+S)	RAID-6 (5+P+P+S)	RAID-6 (5+P+P+S)				Drive Types: 3 TB 3.5" 7.2k
4	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 300 GB 2.5" 15k
6	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 300 GB 2.5" 15k
5	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 300 GB 2.5" 15k
7	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 300 GB 2.5" 15k
4	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 300 GB 2.5" 15k
6	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (6+P+S)	RAID-5 (7+P)	RAID-5 (7+P)	Drive Types: 300 GB 2.5" 15k

Total Gross Capacity 283 TB
Total Net Capacity: 188 TB

Figure C-2 A spreadsheet to document the DS8800 configuration

This machine has a total net capacity of 188 TB, as shown by Capacity Magic. The detailed report folder shows us that this capacity is split into 35 TB net for mainframe and 153 TB for Open Systems.

Of the four SSD ranks, we use two ranks for the mainframe, balancing between the two processor complexes, and we make two 2-tier pools for CKD. Each pool has one 300 GB SSD rank, and six 450 GB/10K ranks. The SSD capacity ratio is about 10%. Our sizing shows previously (an assumption in this case) that this SSD ratio can be combined with this drive type 450 GB/10K, and total capacity. Of the 14 mainframe ranks, each extent pool gets seven ranks.

For Open Systems, which show the red “O” in the corner of each array in Figure C-1 on page 652), of the 153 TB net, we see in the Capacity Magic detail report that 83 TBs are nearline storage (7200 min^{-1}/RAID 6) and 70 TB consist of a mix of about 5% SSDs and about 95% Enterprise drives, which in this context are 300 GB/15K drives, in RAID 5. Again, this scenario assumes that our sizing showed that this ratio is an adequate ratio of SSDs to combine with the 300 GB/15K drives.

In total, there are 44 ranks for Open Systems. We have several options to create extent pools from these ranks.

One option is to create two large Open Systems extent pools. Each extent pool contains 22 ranks (one SSD rank, 18 Enterprise ranks, and three nearline ranks). This design provides two 3-tier extent pools of about 77 TBs each. Easy Tier has the maximum flexibility to promote and demote extents between the three tiers in each pool.

Using a spreadsheet, this configuration looks like Figure C-3 on page 654.

2-frame 4-way DS8800

	Frame	DAP	Enclosure	Drive Class	RAID Type	Disk Size/Speed	Extents
4	A	2	9 and 10	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
5	A	2	9 and 10	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
6	A	2	9 and 10	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
7	A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
8	A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
9	A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
10	A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
11	A	1	7 and 8	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
12	A	1	7 and 8	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
13	A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
14	A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
15	A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
16	A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
17	A	3	5 and 6	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
18	A	3	5 and 6	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
19	A	0	3 and 4	SSD	RAID-5 (6+P+S)	300 GB SSD	1763
20	A	0	3 and 4	SSD	RAID-5 (6+P+S)	300 GB SSD	1763
21	A	2	1 and 2	SSD	RAID-5 (6+P+S)	300 GB SSD	1574
22	A	2	1 and 2	SSD	RAID-5 (6+P+S)	300 GB SSD	1574
23	B	7	13 and 14	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
24	B	7	13 and 14	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
25	B	7	13 and 14	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
26	B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
27	B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
28	B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
29	B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
30	B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
31	B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
32	B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
33	B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
34	B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
35	B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
36	B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
37	B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
38	B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
39	B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
40	B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
41	B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
42	B	5	7 and 8	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
43	B	5	7 and 8	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
44	B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
45	B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
46	B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
47	B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
48	B	7	5 and 6	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
49	B	7	5 and 6	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
50	B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
51	B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
52	B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
53	B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
54	B	4	3 and 4	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
55	B	4	3 and 4	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
56	B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
57	B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
58	B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
59	B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
60	B	6	1 and 2	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
61	B	6	1 and 2	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835

Logical Configuration

Colour	DAPs	Extent Pool	Destination	Processor Complex	Storage Type	Tiering	Easy Tier - EAM	Capacity Extents
	0,1,3	0+1	System z	0+1	ckd	Multi-tier (two tier)	Yes - Managed	37262
	2,4,5,6,7	2	Open Systems	0	fb	Multi-tier (three tier)	Yes - Managed	71282
	2,4,5,6,7	3	Open Systems	1	fb	Multi-tier (three tier)	Yes - Managed	71282

Workload Definitions		Total Extents	= GiB net	= GB net
A	= System z, Isolated	37262	32829	35250
B	= Open Systems, Resource Sharing	142564	142564	153077

Figure C-3 Shared configuration example when using a spreadsheet

Another option is to leave the nearline space isolated. With 83 TB net, the nearline space is larger than the combined SSD+ENT space of 70 TB for Open Systems, so we use the nearline space as backup space. For this setup, we create 4 Open Systems extent pools (2 pairs). One extent pool pair has 19 ranks (1 SSD rank and 18 Enterprise ranks), and one extent pool pair with each pool containing three nearline ranks. The cross-tier Easy Tier is only active in the SSD+ENT "production" pool pair, but at the same time, the intra-tier Easy Tier auto-rebalancing is active in the nearline pools.

The third option to build two 2-tier pool pairs. One tier is the "Gold" tier, and one tier is the "Silver" tier. Only the Gold tier contains SSDs. All applications that we place in the Silver tier can only be on either Enterprise or nearline ranks. Assume that from our 66 TB net in 300 GB/15K drives, we split them into 22 TB together with the SSD space, which creates a Gold tier of around 25 TB net. The remaining 44 TB of these drives are with the nearline drives. Then, the result is four open extent pools/two pairs:

- One Gold pair: Each with 1 SSD rank plus 6 Enterprise ranks
- One Silver pair: Each with 12 Enterprise ranks and 3 nearline ranks

Figure C-4 shows the spreadsheet.

2-frame 4-way DS8800

Frame	DAP	Enclosure	Drive Class	RAID Type	Disk Size/Speed	Extents
A	2	9 and 10	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
A	2	9 and 10	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
A	2	9 and 10	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	1	7 and 8	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	1	7 and 8	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
A	1	7 and 8	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	3	5 and 6	Enterprise	RAID-5 (6+P+S)	450 GB 2.5" 10 k	2664
A	3	5 and 6	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
A	3	5 and 6	Enterprise	RAID-5 (7+P)	450 GB 2.5" 10 k	3106
A	0	3 and 4	SSD	RAID-5 (6+P+S)	300 GB SSD	1763
A	0	3 and 4	SSD	RAID-5 (6+P+S)	300 GB SSD	1763
A	2	1 and 2	SSD	RAID-5 (6+P+S)	300 GB SSD	1574
A	2	1 and 2	SSD	RAID-5 (6+P+S)	300 GB SSD	1574
B	7	13 and 14	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
B	7	13 and 14	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
B	7	13 and 14	Nearline	RAID-6 (5+P+P+S)	3 TB 3.5" 7.2k	12922
B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	4	11 and 12	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	9 and 10	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	5	7 and 8	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	5	7 and 8	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	5	7 and 8	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	7	5 and 6	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	7	5 and 6	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	7	5 and 6	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	4	3 and 4	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	4	3 and 4	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	4	3 and 4	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	6	1 and 2	Enterprise	RAID-5 (6+P+S)	300 GB 2.5" 15 k	1574
B	6	1 and 2	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835
B	6	1 and 2	Enterprise	RAID-5 (7+P)	300 GB 2.5" 15 k	1835

Logical Configuration

Colour	DAPs	Extent Pool	Destination	Processor Complex	Storage Type	Tiering	Easy Tier - EAM	Capacity Extents
	0,1,3	0+1	System z	0+1	ckd	Multi-tier (two tier)	Yes - Managed	37262
	5,6,7	2	Open Silver	0	fb	Multi-tier (two tier)	Yes - Managed	59220
	2,5,6,7	3	Open Silver	1	fb	Multi-tier (two tier)	Yes - Managed	59220
	2,4	4	Open Gold	0	fb	Multi-tier (two tier)	Yes - Managed	12062
	2,4	5	Open Gold	1	fb	Multi-tier (two tier)	Yes - Managed	12062

Workload Definitions		Total Extents	= GiB net	= GB net
A	= System z, Isolated	37262	32829	35250
B	= Open Systems, Silver	118440	118440	127174
C	= Open Systems, Gold	24124	24124	25903

Figure C-4 Gold and Silver configuration when using the spreadsheet

With the manual Easy Tier functions, we can move volumes around between Silver and Gold. If we see that a Silver volume does not perform well, we can move it to Gold online. We can also change our allocated capacities later in the operation if we see that the capacity for Gold was too small. If there are too many ranks in the Silver pool pair, we can unconfigure ranks online and move them out of the Silver pool pair. We can put them in the Gold pool pair to enable more applications to participate in the SSD tiering.

Also, in FlashCopy layouts, it might be advantageous to separate source ranks and target ranks into different extent pool pairs. This approach can also lead to creating more than one extent pool pair.

In Figure C-4, you can see that we achieved load separation by DA pair to a large extent. The mainframe loads its own DA pairs. On the Open Systems side, one DA pair (4) is exclusive to the Gold tier, and three DA pairs (5, 6, and 7) are exclusive to the Silver tier. Only DA pair 2 serves disks from both the Gold and Silver open tiers. Because the Silver tier disks of this pair are nearline with little load, we can see that DA pair 2 mainly serves the solid-state drives (SSDs) with their many IOPS. Therefore, we consider that it belongs to the Gold tier.

In both Figure C-3 on page 654 and Figure C-4 on page 655, we distributed the disks of a certain DA pair, DA pair 5, between an even extent pool and an odd extent pool always. Because we have two DAs in a pair, we want to put load on both DAs. If we give all ranks of DA pair 5 to an even extent pool and all ranks of DA pair 7 to an odd extent pool, we share the load across both processor complexes, but we do not use all 4 DAs of the two DA pairs, 5 and 7.

Performance tools

We described the use of the various performance tools, such as Tivoli Storage Productivity Center (TPC) for Disk, for performance monitoring and problem analysis in earlier chapters. Tivoli Storage Productivity Center also can store performance data, over a longer time, in a database, so that historical trending is possible.

You can use the TPC Reporter to take regular snapshots of your DS8000, for example, monthly snapshots. The PDF reports contain the performance data of this month, the configuration of the various arrays and the drive types for these arrays, their respective utilizations, and the number of HBAs. The PDF files are easy to store. Use the PDF files later as references to see how loads grow over time and how certain pools become overloaded slowly to facilitates upgrade planning.

The Storage Tier Advisor Tool (STAT) provides information about the amount of hot data and which rank types are best to add next and with what expected results. You can collect the raw data for the STAT in regular intervals for each machine and store it for reference.

You can use the DS8000 command-line interface (DSCLI) commands to see an overview of the configuration and performance. Use the **`lsperfgrprpt`** command of the I/O Priority Manager for simple performance tracking. The other DSCLI **`list`** commands either show the configuration (**`ls-`** commands), including the rank-to-DA mapping when using the **`lsarray -l`**, or use additional performance counters with the **`-metrics`** options of **`showfbvol`** and **`showrank`**. The results of DSCLI outputs can be saved at any time for later reference.

DS8000 Query and Reporting tool (DSQTOOL)

In this section, we introduce a storage system configuration DS8000 Query and Reporting tool (DSQTOOL).

The DS8000 products provide both DS8000 command-line interface (DSCLI) and GUI (DS Storage Manager) tools for configuration management. A combination of the tools allows both novice and highly skilled storage administrators to accomplish the routine tasks involved in day-to-day storage management and configuration reporting.

However, the task of maintaining up-to-date configuration documentation is often time consuming. The documentation might not be up-to-date due to the amount of time required to combine DSCLI and DS Storage Manager output display information into a consolidated document.

To help automate the process, VBScript and Excel macro programs were written to provide a quick and easy-to-use interface to DS8000 storage servers through DSCLI, which is passed to an Excel macro to generate a summary workbook with detailed configuration information.

The tool is launched from a desktop icon on a Microsoft Windows system (probably a storage administrator desktop computer). A VBScript program is included to create the icon with a link to the first Excel macro that displays a DS8000 Hardware Management Console (HMC) selection panel to start the query process.

The DSQTOOL is a combination of VBScript and Excel macro programs that interface with the DS8000 Command Line Interface (DSCLI) utility. The DSQTOOL uses **list** and **show** commands to query and report on system configurations. The design point of the programs is to automate a repeatable process of creating configuration documentation for a specific DS8000.

You can obtain this tool, along with other tools, such as DS8CAPGEN, from this IBM link:

http://congsa.ibm.com/~dlutz/

You can see two DSQTOOL samples of our machine in Figure C-5 and Figure C-6 on page 658. Look at the tabs at the bottom. This tool collects a significant amount of DS8000 configuration information to help you track the changes in your machine over time.

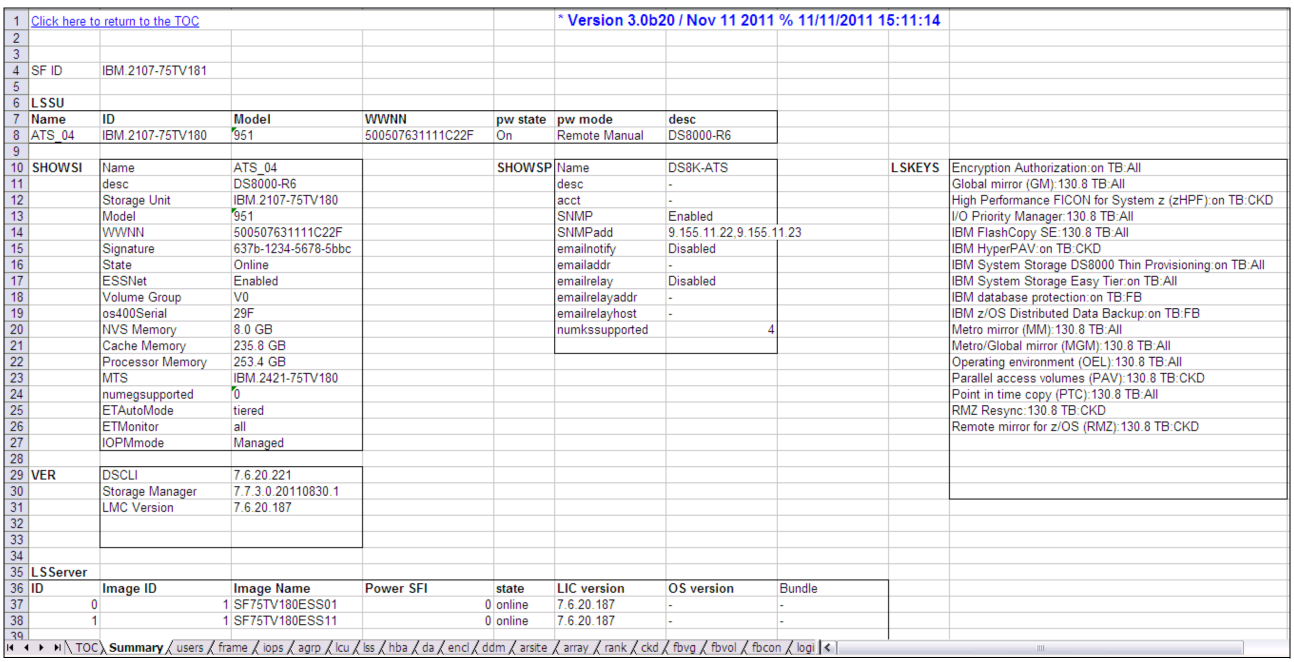
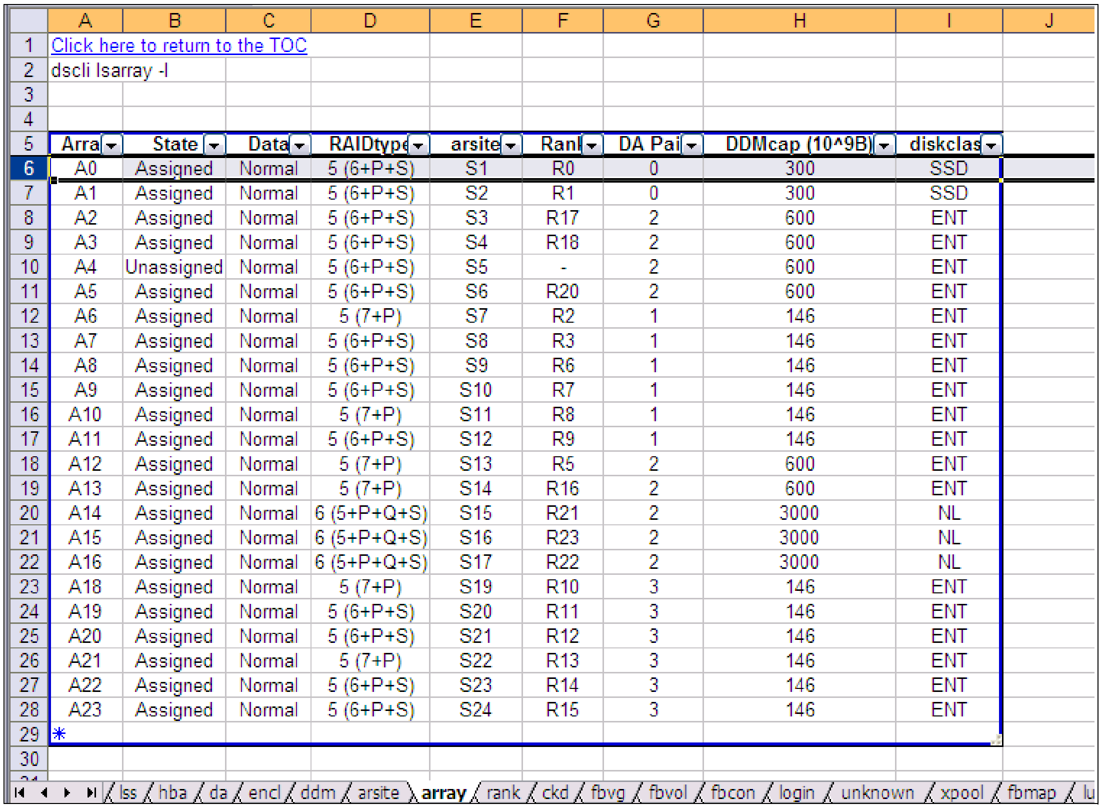


Figure C-5 DSQTOOL Summary tab



Click here to return to the TOC

dscli lsarray -l

Arra	State	Data	RAIDtype	arsite	Rank	DA Pai	DDMcap (10^9B)	diskclas
A0	Assigned	Normal	5 (6+P+S)	S1	R0	0	300	SSD
A1	Assigned	Normal	5 (6+P+S)	S2	R1	0	300	SSD
A2	Assigned	Normal	5 (6+P+S)	S3	R17	2	600	ENT
A3	Assigned	Normal	5 (6+P+S)	S4	R18	2	600	ENT
A4	Unassigned	Normal	5 (6+P+S)	S5	-	2	600	ENT
A5	Assigned	Normal	5 (6+P+S)	S6	R20	2	600	ENT
A6	Assigned	Normal	5 (7+P)	S7	R2	1	146	ENT
A7	Assigned	Normal	5 (6+P+S)	S8	R3	1	146	ENT
A8	Assigned	Normal	5 (6+P+S)	S9	R6	1	146	ENT
A9	Assigned	Normal	5 (6+P+S)	S10	R7	1	146	ENT
A10	Assigned	Normal	5 (7+P)	S11	R8	1	146	ENT
A11	Assigned	Normal	5 (6+P+S)	S12	R9	1	146	ENT
A12	Assigned	Normal	5 (7+P)	S13	R5	2	600	ENT
A13	Assigned	Normal	5 (7+P)	S14	R16	2	600	ENT
A14	Assigned	Normal	6 (5+P+Q+S)	S15	R21	2	3000	NL
A15	Assigned	Normal	6 (5+P+Q+S)	S16	R23	2	3000	NL
A16	Assigned	Normal	6 (5+P+Q+S)	S17	R22	2	3000	NL
A18	Assigned	Normal	5 (7+P)	S19	R10	3	146	ENT
A19	Assigned	Normal	5 (6+P+S)	S20	R11	3	146	ENT
A20	Assigned	Normal	5 (6+P+S)	S21	R12	3	146	ENT
A21	Assigned	Normal	5 (7+P)	S22	R13	3	146	ENT
A22	Assigned	Normal	5 (6+P+S)	S23	R14	3	146	ENT
A23	Assigned	Normal	5 (6+P+S)	S24	R15	3	146	ENT

Figure C-6 DSQTOOL array tab

D

Microsoft Windows server performance log collection

This appendix describes the necessary procedures to collect Microsoft Windows Server 2003 and Windows Server 2008 server disk performance data for later analysis. We describe the process of collecting data for performance tuning and searching for bottlenecks. This process differs from the data collection for alerts and monitoring by the shorter period of collection and shorter intervals. Collected values raise no alerts. In general, principles of data collection in Windows Server 2003 and Windows Server 2008 are the same. The difference is in the GUI mostly and the names of the menu options.

Windows Server 2003 log file configuration

The Performance console is a snap-in for Microsoft Management Console (MMC). You use the Performance console to configure the System Monitor and Performance Logs and Alerts tools.

You can open the Performance console by clicking **Start** → **Programs** → **Administrative Tools** → **Performance** or by typing **`perfmon`** on the command line.

Also, you can start the Performance console from the Microsoft Management Console. Click **File** → **Open**. Go to the `Windows\System32` folder and select **perfmon.msc**. See Figure D-1.

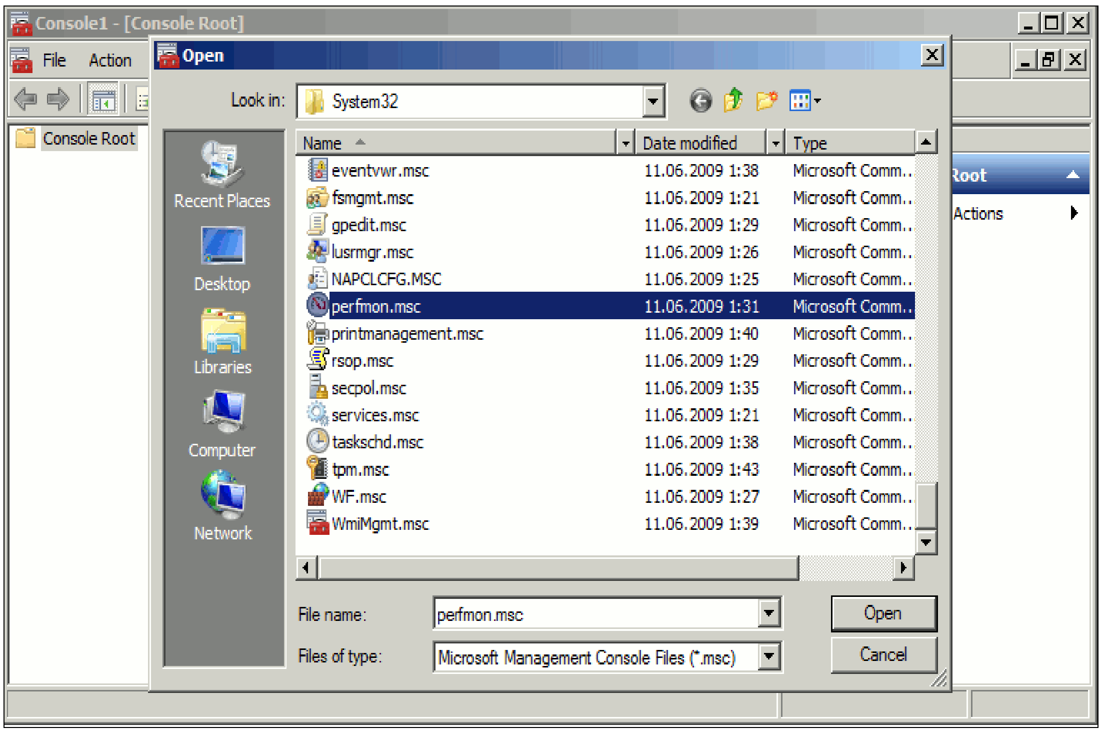


Figure D-1 Opening the performance console

On the Performance Logs and Alerts window, which is shown in Figure D-2, you can collect performance data manually or automatically from local or remote systems. Saved data can be displayed in the System Monitor or data can be exported to a spreadsheet or database.

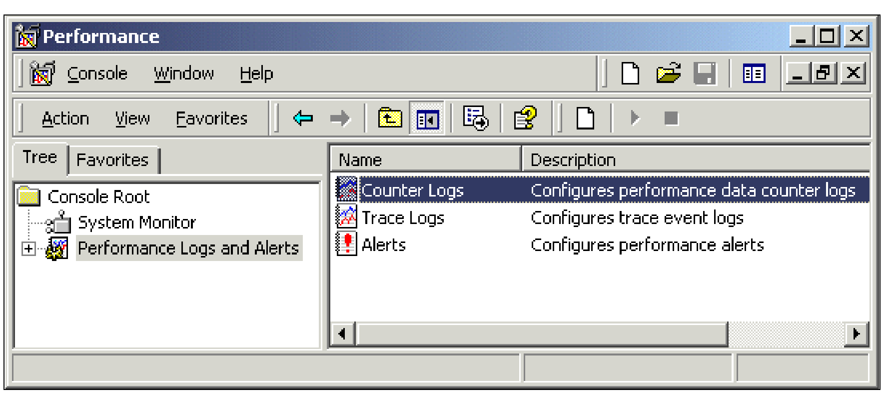


Figure D-2 Performance Logs and Alerts

Performance Logs and Alerts provides the following functions:

- Counter logs

  Use this function to create a log file with specific system objects, counters, and their instances. Log files can be saved in different formats (file name + file number, or file name + file creation date) for use in the System Monitor or for exporting to database or spreadsheet applications. You can schedule the logging of data, or the counter log can be started manually by using program shortcuts. Counter log settings can also be saved in HTML format for use in a browser either locally or remotely via TCP/IP.

- Trace logs

  Use this function to create trace logs that contain trace data provider objects. Trace logs differ from counter logs in that they measure data continuously rather than at specific intervals. You can log operating system or application activity by using event providers. There are two kinds of providers: system and non-system providers.

- Alerts

  Use this function to track objects and counters to ensure that they are within a specified range. If the counter value is under or over the specified value, an alert is issued.

Configuring logging of disk metrics in Windows Server 2003

Follow these steps to configure logging:

1. From the main Performance (perfmon) console, right-click **Counter Logs** and select **New Log Settings**.
2. Enter the New Log Settings name, for example: `Disk Performance`.
3. The Disk Performance General tab log settings configuration window opens, as shown in Figure D-3.

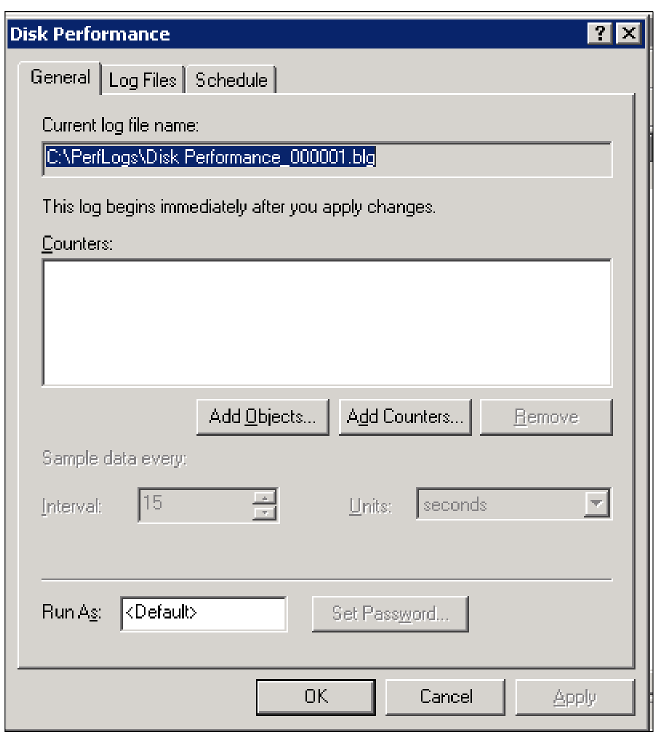


Figure D-3 Windows Server 2003 Disk Performance log General tab

4. Click **Add Objects**, select the **PhysicalDisk** and **LogicalDisk** objects, and select **Add**.
5. In the General tab, the “Sample data every:” field is used to set how frequently you capture the data. When configuring the interval, specify an interval that provides enough granularity so that you can identify the issue. Consider these rules:
   – Collect data for the required period only. Do not capture useless data.
   – If you want to size the system, collect the data for the typical period of work.
   – If you want to identify the performance issue, collect the data for the period in which that issue might appear.
   – If you want to run the benchmark, collect the data for that period only.
   – A shorter period of data collection requires smaller granularity. If you need to capture the data for 30 seconds, an interval of 15 seconds between each collection does not work. You have only two values in this case, which you cannot use for analysis and to capture the issue. Setting an interval for 1 second, for example, and collecting data for several hours makes your log file large and difficult to work with.

   > **Important:** Different spreadsheet software has different capabilities to work with large amounts of data. When configuring the interval, remember both the problem profile and the planned collection duration to not capture more data than can be reasonably analyzed. For data collection during several hours, you might want several files, such as one file for each hour, rather than one large file for several hours. Data collected for one hour with a 1 second interval gives you 3600 entries for each value. The total number of entries must be multiplied by the number of values collected.

6. In the Run As field, enter the account with sufficient rights to collect the information about the server to be monitored, and then click **Set Password** to enter the relevant password.
7. In the Log Files tab, shown in Figure D-4 on page 663, you set the type of the saved file, the suffix that is appended to the file name, and an optional comment. You can use two types of suffixes in a file name: numbers or dates. The log file types are listed in Table D-1 on page 663. If you click **Configure**, you can also set the location, file name, and file size for a log file. The Binary File format takes the least amount of space and is suggested for most logging.

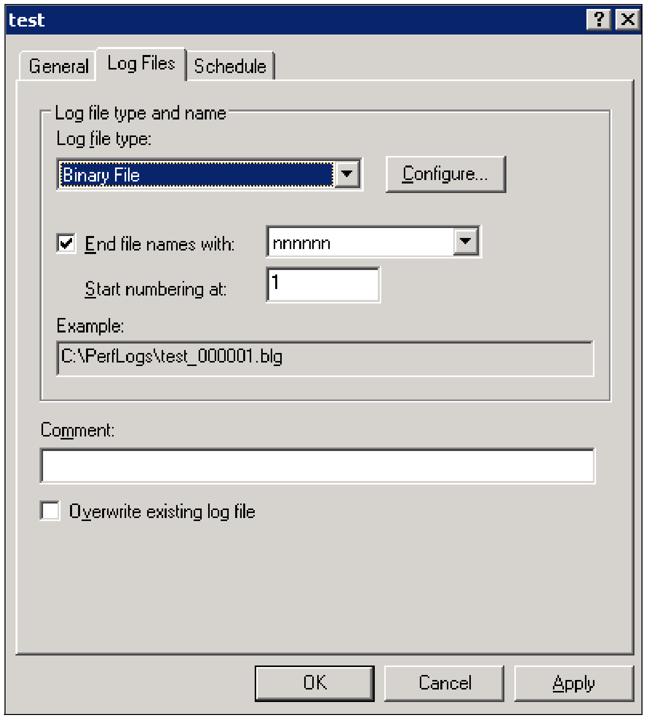


Figure D-4 Windows Server 2003 Log Files tab

Table D-1 Counter log file formats

Log file type	Description
Text file - csv	Comma-delimited values log file (csv extension). Use this format to export to a spreadsheet.
Text file - TSV	Tab-delimited log file (TSV extension). Use this format to export the data to a spreadsheet program.
Binary file	Sequential, binary log file (BLG extension). Use this format to capture data intermittently (stopping and resuming log operation).
Binary circular file	Circular, binary-format log file (BLG extension). Use this format to store log data to the same log file, overwriting old data.
SQL Database	Logs are output as an SQL database.

8. In the Schedule tab that is shown in Figure D-5 on page 664, you specify when this log is started and stopped. You can select the option box in the start log and stop log sections to manage this log manually by using the Performance console shortcut menu. You can configure to start a new log file or run a command when this log file closes. This capability is useful when collecting data for a long period and with many values. You can set the schedule for each hour and get one file per hour.

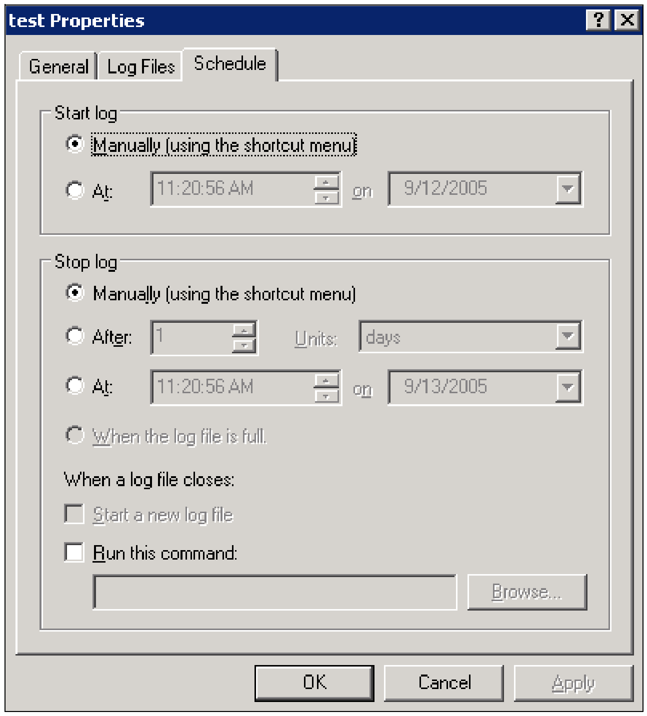


Figure D-5 Windows Server 2003 log file Schedule tab

9. After clicking **OK**, the logging starts automatically. If for any reason, it does not start, right-click the Log Settings file in the perfmon window and click **Start**. See Figure D-6 on page 665.
10. To stop the counter log, click the **Stop the selected log** icon on the toolbar or the **Stop** in the menu when you right-click.

Saving counter log settings

You can save the counter log settings to use them later. Follow these steps to save log settings:

1. Select the counter log file in which you want to save settings.
2. Right-click the log. The window shown in Figure D-6 on page 665 appears.
3. Click **Save Setting As**.
4. Select a location and enter a file name, and then click **Save** (saving to an HTML file is the only option).

This log settings file in HTML format can then be opened with any web browser. You can also use the pop-up menu to start, stop, and save the logs, as shown in Figure D-6 on page 665.

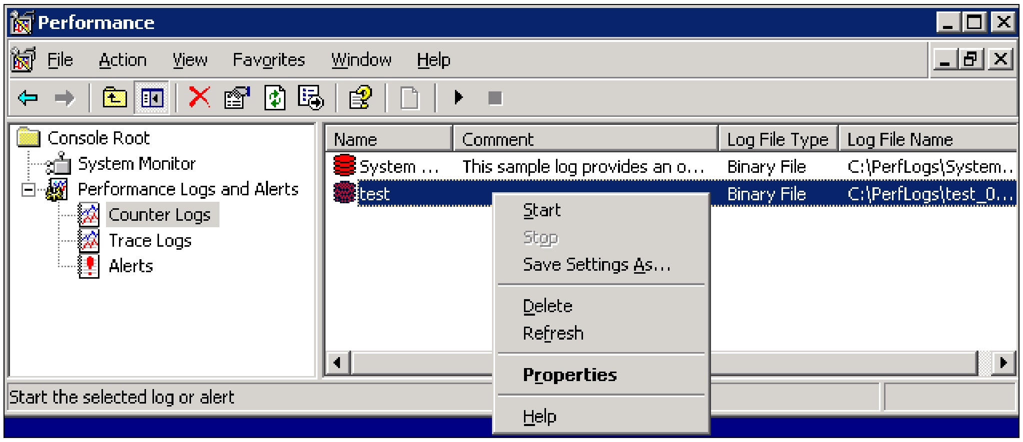


Figure D-6 Windows Server 2003 Counter log pop-up menu

Importing counter log properties

You can import counter log settings from saved files. Follow these steps to import settings:

1. Right-click the right window.
2. Select **New Log Settings From**.
3. The Open dialog window appears. Choose the location, select a file name, and then click **Open**.
4. The Name dialog box opens. If you want to change the log setting name, you can enter a name; otherwise, click **OK**.
5. A dialog window opens where you can add or remove counters and change log file settings and scheduling.
6. If you want to edit the settings, change the required fields; otherwise, click **OK**.

Analyzing disk performance from collected data

Performance and bottleneck analysis requires the data to be collected for some period. The current activity window of the Performance console shows you data for about 2 minutes only. It is definitely not enough for analysis and summary. So, it is better to use log files with collected data, because the log files show the disk usage over an extended period. The information in the log can be viewed and evaluated at a later time. The following section describes the process for viewing and exporting data that is collected in a log file.

Retrieving data from a counter log file

After you save data to a log file, you can retrieve that data and process it. By default, the System Monitor displays real-time data. Follow these steps to display previously logged data:

1. Click the **View log data file** icon on the System Monitor toolbar.
2. The System Monitor Properties dialog shown in Figure D-7 on page 666 opens at the Source tab. Select **Log Files**, and then click **Add**. The Log files dialog opens. Select the log file that you want and click **Open**.

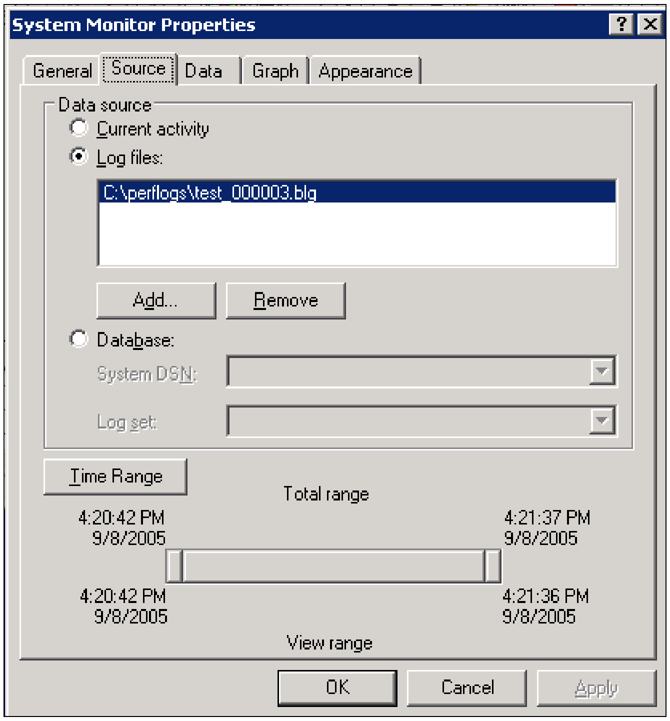


Figure D-7 Windows Server 2003 System Monitor Properties (Source tab)

3. At the System Monitor Properties dialog, select the **Data** tab. You now see any counters that you specified when setting up the Counter Log as shown in Figure D-8. If you only selected counter objects, the Counters section is empty. To add counters from an object, click **Add,** and then select the appropriate counters.

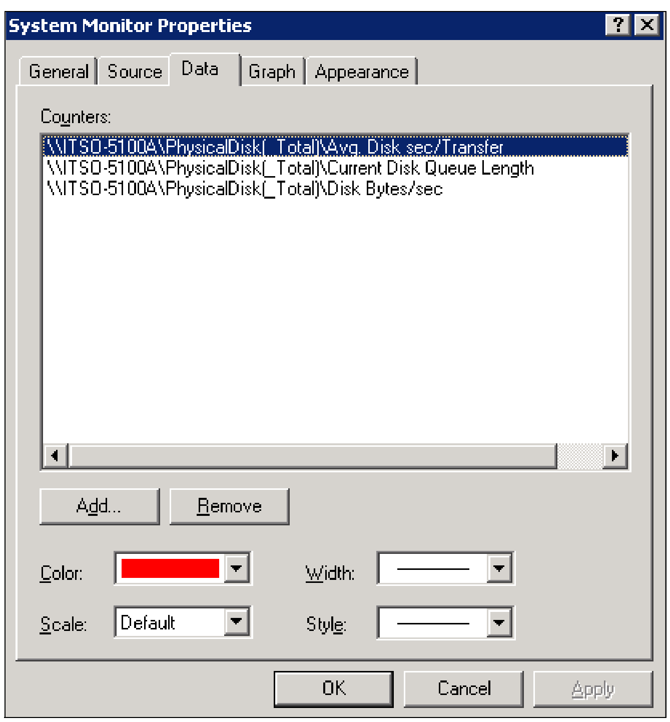


Figure D-8 System Monitor Properties (Data tab)

4. Select the period of time to analyze data. By default, data for the counter is shown for the entire period of time that data was collected. See Figure D-9.

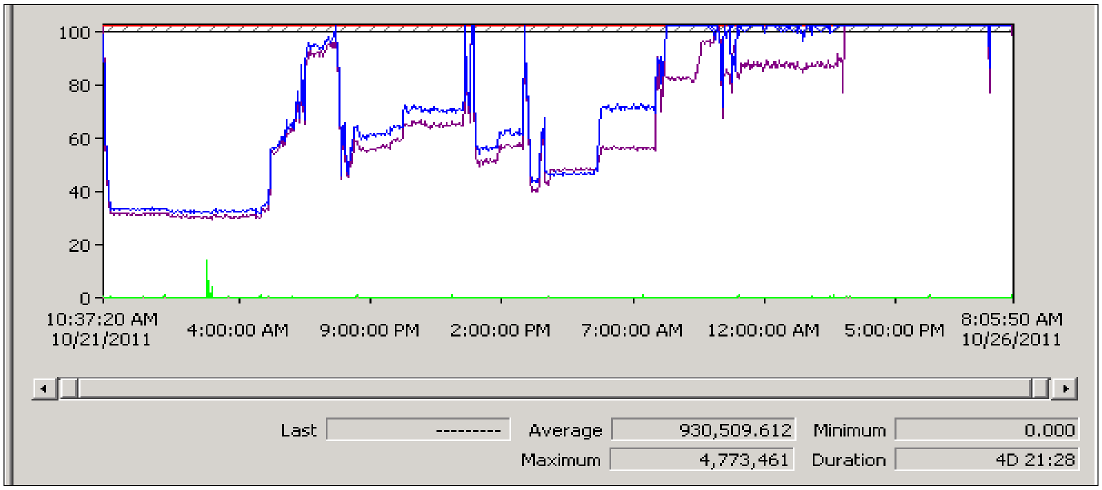


Figure D-9   Performance monitor window with the entire period of collected data

Figure D-9 shows the data for the entire period of collection. In this example, the data is for five days. So, you can see the trend, but not the details.

5. To be able to examine data for a smaller period (few minutes), you can use the Zoom function of the Performance Monitor console. See Figure D-10.

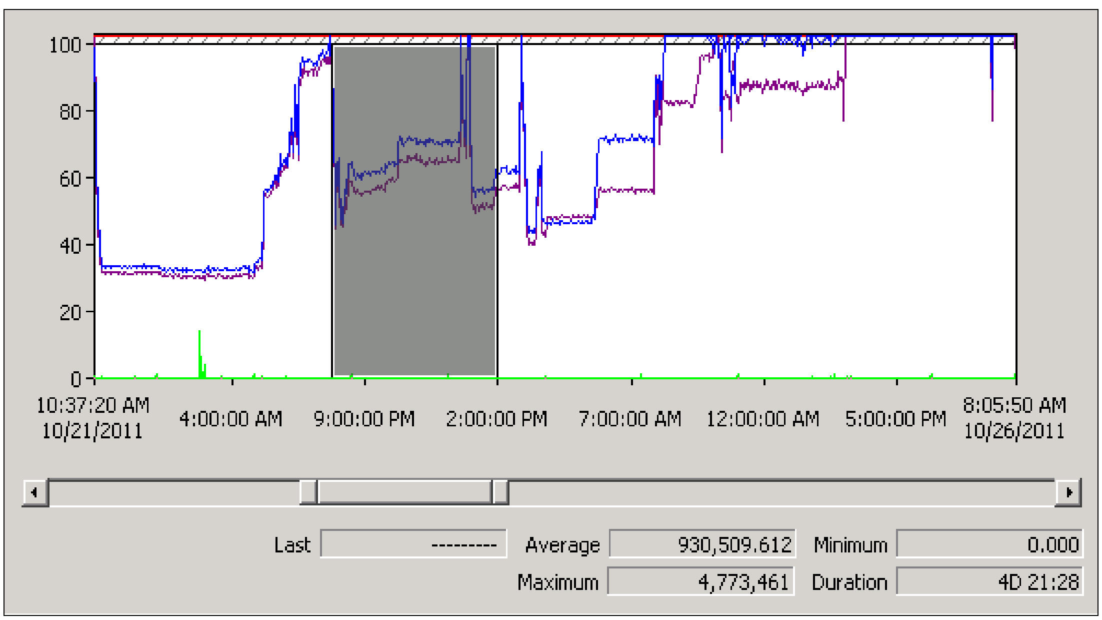


Figure D-10   Selecting the interval

Select the interval you want (see Figure D-10) to examine and click **Zoom** or press Ctrl+Z. The result is shown in Figure D-11 on page 668.

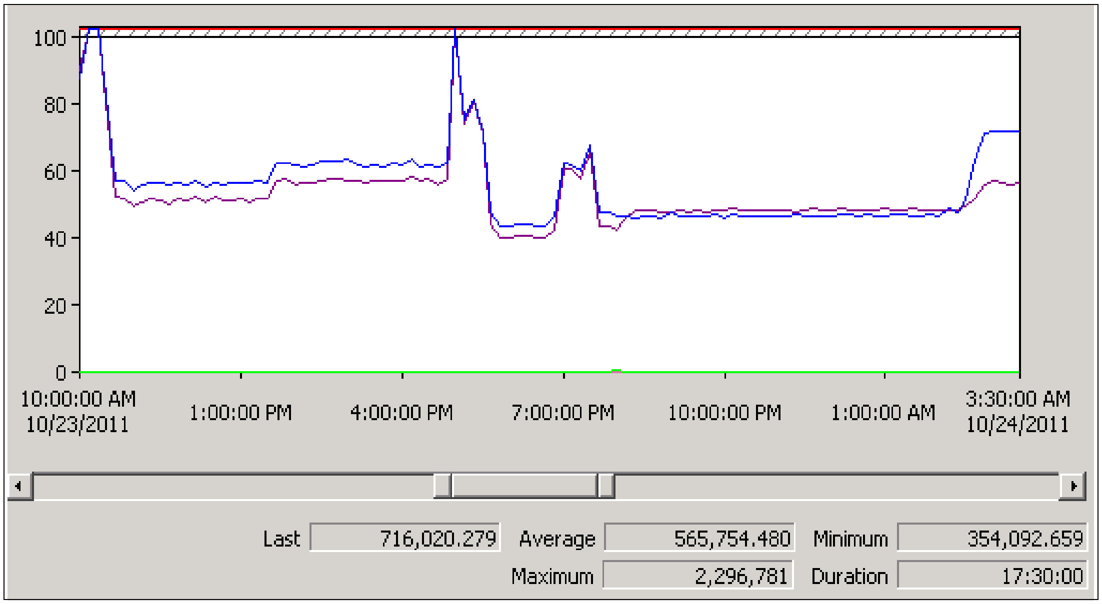


Figure D-11 Zoomed interval of performance data

To zoom out, you can move the slider ends to make it wider. Click **Zoom** or press Ctrl+Z, and the graph zooms out.

6. Highlight the counter that you want to see more clearly than the other counters. Click **Highlight** or press Ctrl+H (Figure D-12).

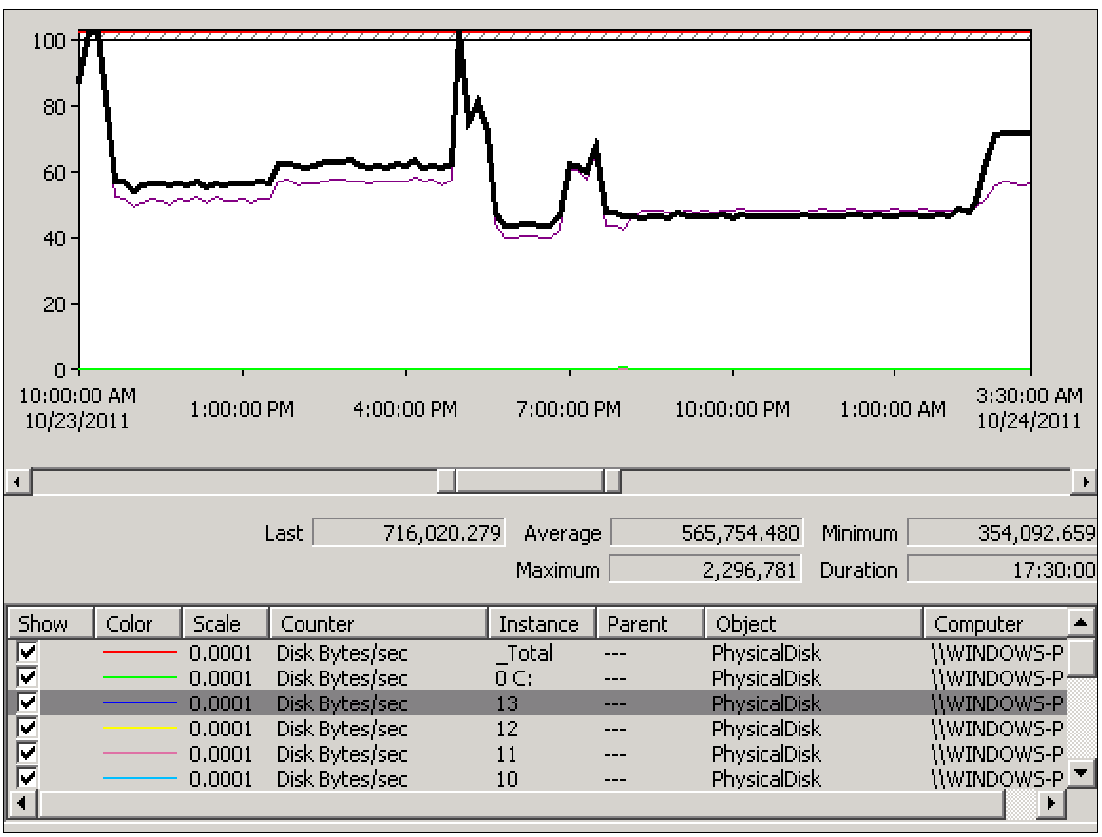


Figure D-12 Highlighting required value performance graph

Figure D-12 on page 668 shows you the highlighted graph of the value Disk Bytes/sec for disk number 13. It makes easy to analyze the data. For the selected performance counter, you can see the Last, Average, Minimum, and Maximum values. Duration shows the period of time that is selected now and it can be shortened or made longer.

7. Scale the value to make it fit to the graph area. Some values might be too high or too low to be drawn in the graph area. For example, response time counters are measured in seconds, when in reality they are in milliseconds. With the graph area showing 0 - 100, these values do not appear as graphs. The scaling option can be used to avoid this situation. See Figure D-13.

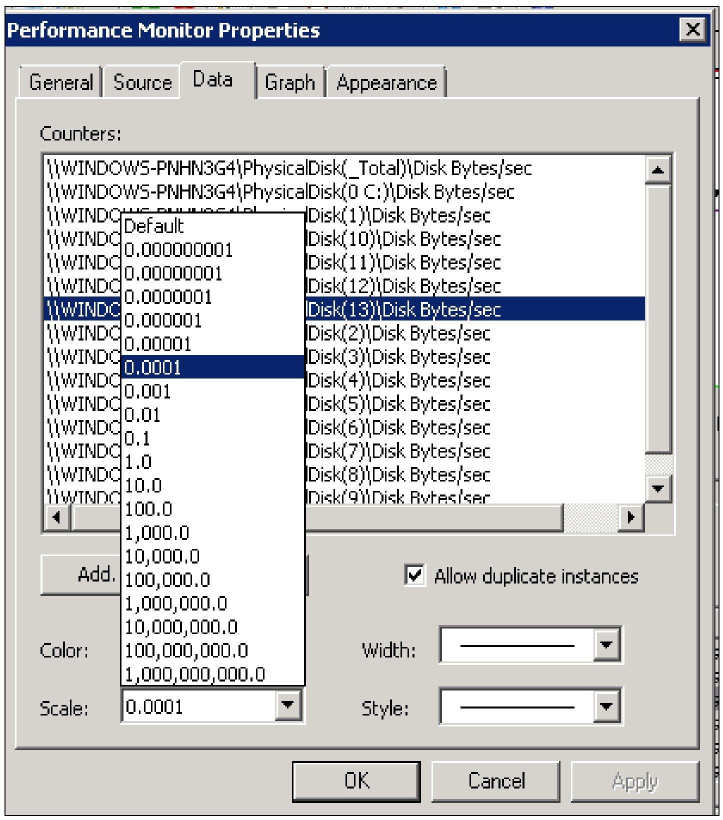


Figure D-13 Scaling the graph values

8. Right-click the value and then click **Properties**. You see the graph options for this value. You can select scaling options. For response times, set the scaling option to 1000, so it appears in the graph field. For the other values, for IOPS numbers, for example, you might select 0.0001 or 0.001 to make them appear in the graph area, because IOPS are measured in thousands and exceed the values of the graph area. You can also select the color, width, and the style of the line for each value.

Exporting logged data on Windows Server 2003

Follow these steps to export logged data:

1. Stop the data collection by right-clicking the Log Settings File and selecting **Stop**.
2. Open the **System Monitor** and remove any existing counters.
3. Select the **Log file** icon, select your Log file as shown in Figure D-14 on page 670, and click **OK**.

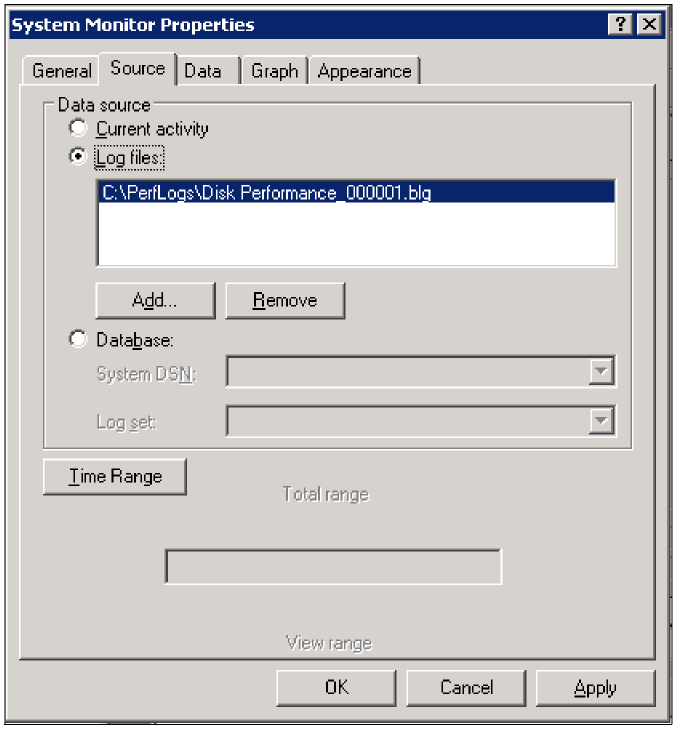


Figure D-14 Windows Server 2003 Log file selection

4. In the Monitor View, click **Add**, select all the key PhysicalDisk counters for all instances, and click **Add**.
5. You see a window similar to Figure D-15.

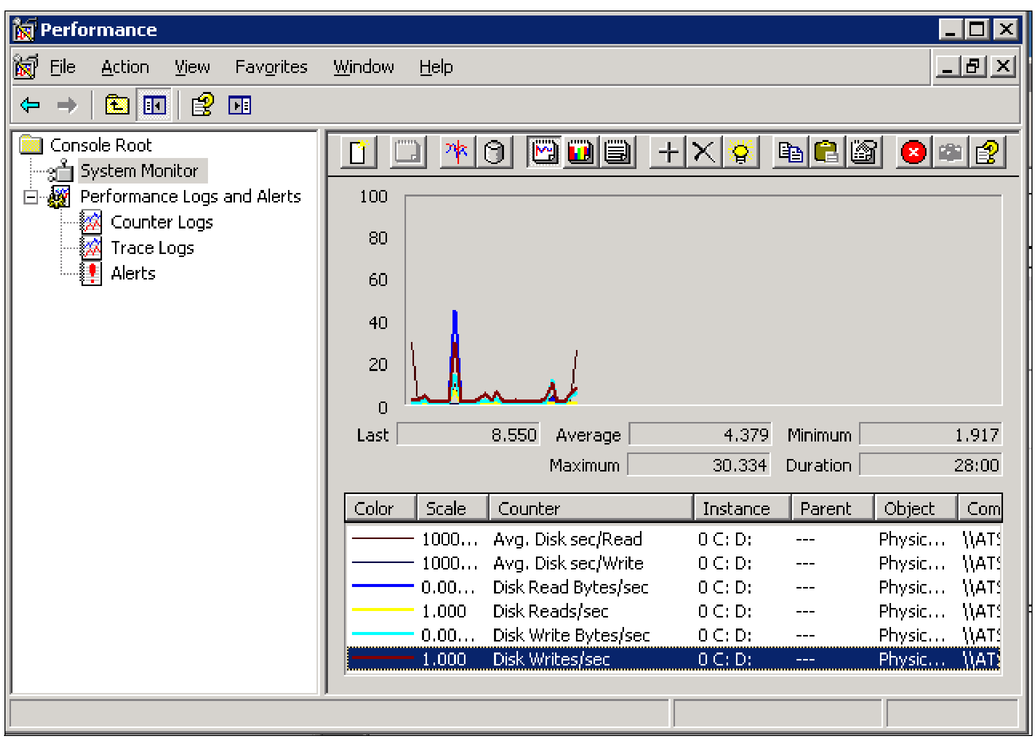


Figure D-15 Windows System Monitor using Performance Log

6. Right-click the chart side, and click **Save Data As**.
7. In the Save Data As window, select a file type of **Text File (Comma delimited)(*.csv)**.
8. Provide a file name, such as Disk_Performance.

Windows Server 2008 log file configuration

In Windows Server 2008, the performance monitor is initiated in the same manner as on Windows Server 2003. There are differences in the 2008 implementation of perfmon. Windows Server 2008 refers to log configuration files as *Data Collector Sets*. These sets are logical groupings of collection sets that, unlike Windows Server 2003, can contain both system and event trace counters in one collector set. Data Collector Sets can use a default collection set or can be manually configured. See Figure D-16. For symmetry with the techniques that we are using on Windows Server 2003, we use a manual configuration process.

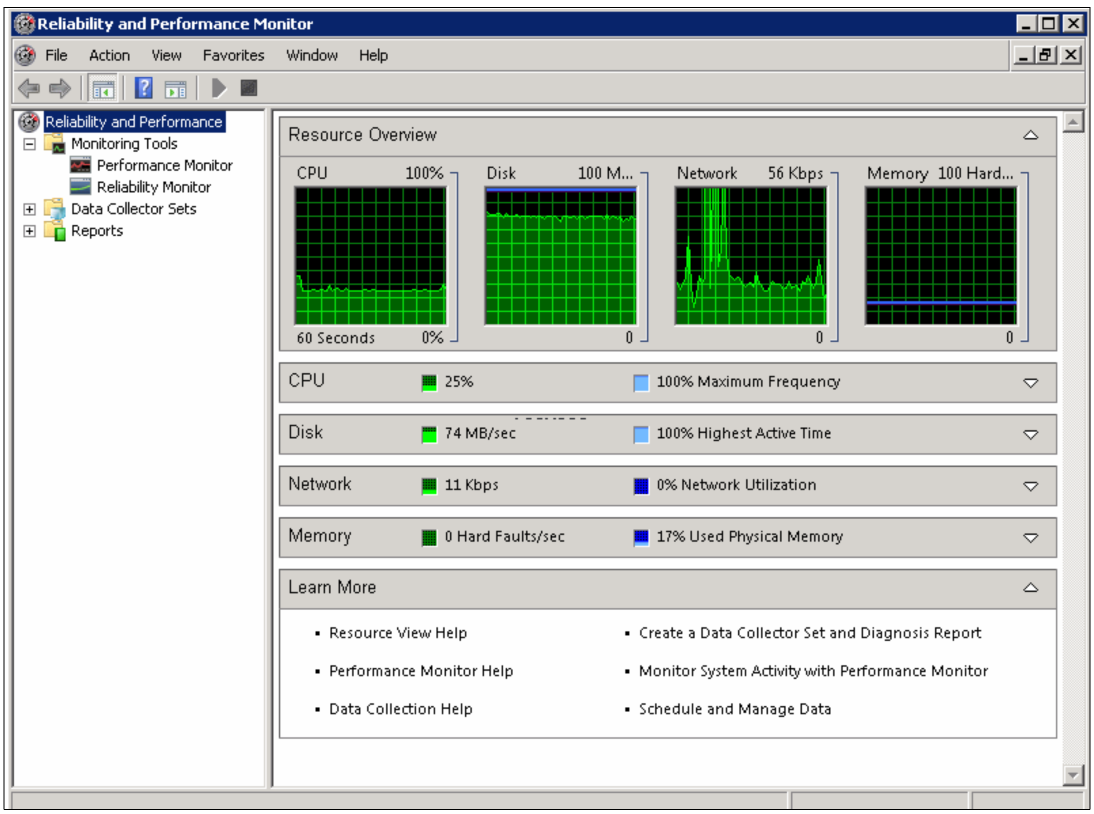


Figure D-16 Windows Server 2008 Reliability and Performance Monitor

Follow these steps to run the Performance Monitor:

1. Expand the **Data Collection Sets** element in the navigation tree as referenced in Figure D-17 on page 672. Right-click **User Defined**.

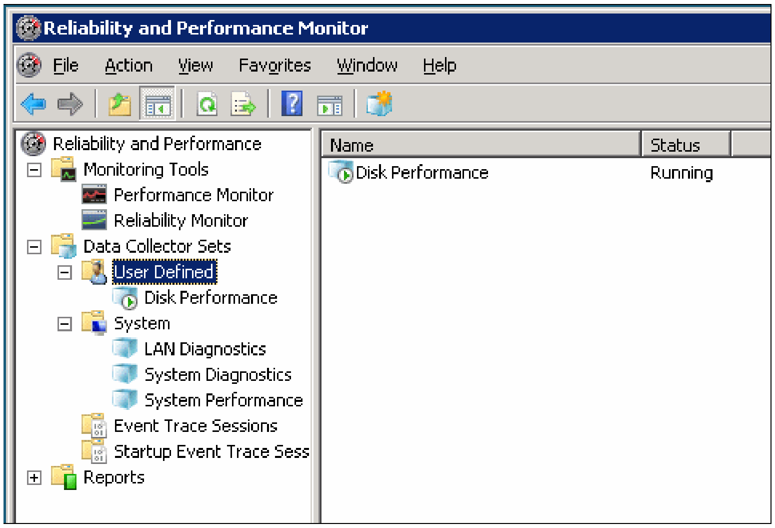


Figure D-17 Windows Server 2008 User Defined Data Collector Sets

2. Select **New** → **Data Collector Set**.
3. The Create new Data Collector Set wizard opens, as shown in Figure D-18. Click **Next**.

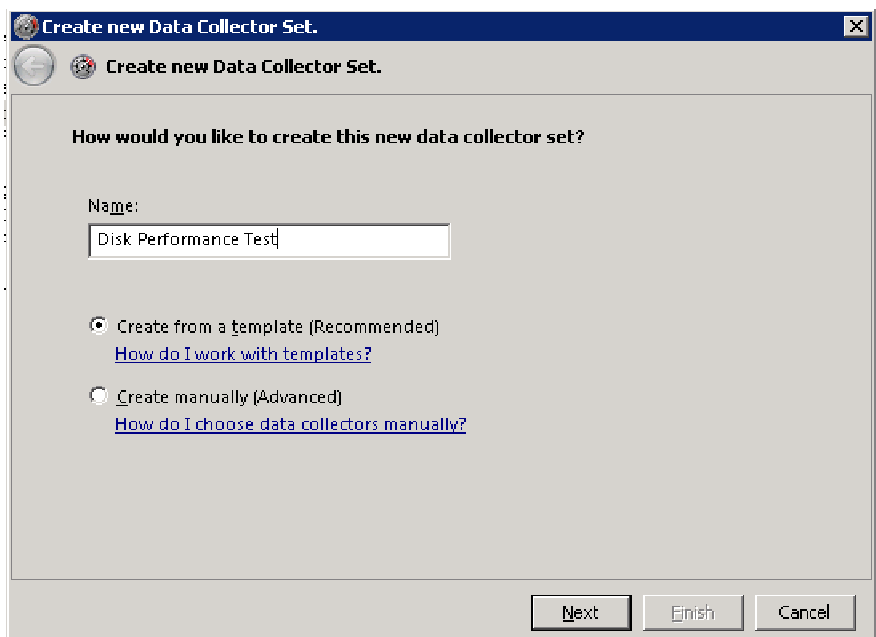


Figure D-18 Windows Server 2008 Data Collector Set wizard

4. Provide a file name, such as Disk_Performance.
5. Select **Create manually (Advanced)** and click **Next**.
6. As shown in Figure D-19 on page 673, check **Performance counter** and click **Next**.

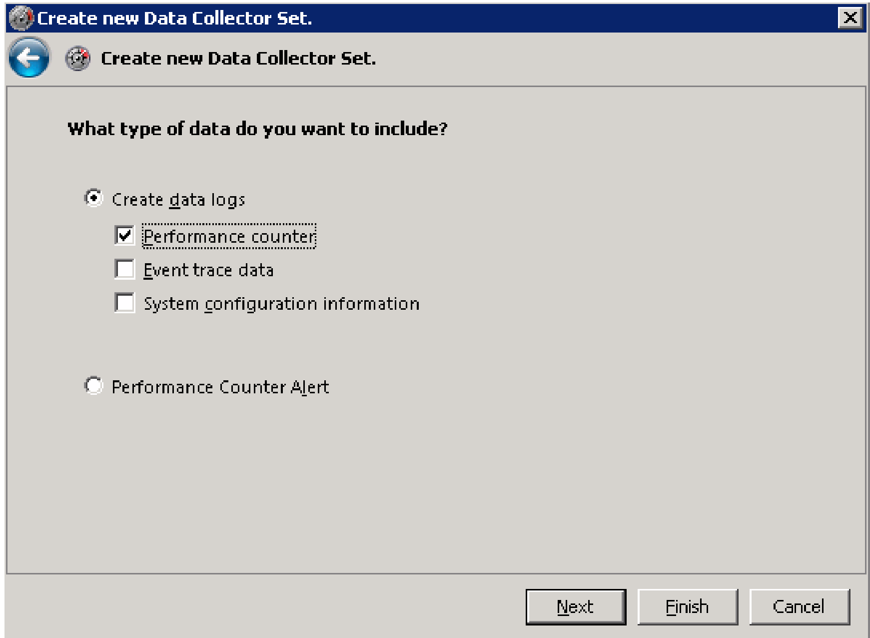


Figure D-19 Windows Server 2008 Advanced Data Collector Set configuration

7. As shown in Figure D-20, you are asked to select the performance counters that you want to log. Click **Add** to see a window similar to Figure D-21 on page 674.

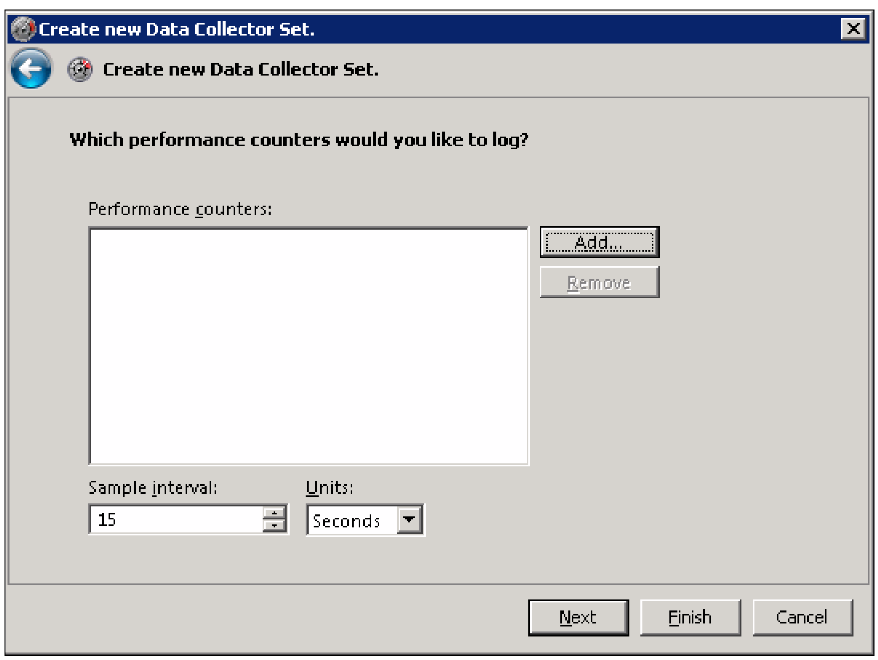


Figure D-20 Windows Server 2008 Add performance counters

8. Select **all instances** of the disks except the Total, and manually select all the individual counters identified in Table 10-2 on page 409. **Select** the computer that you want, as well. Click **OK**.

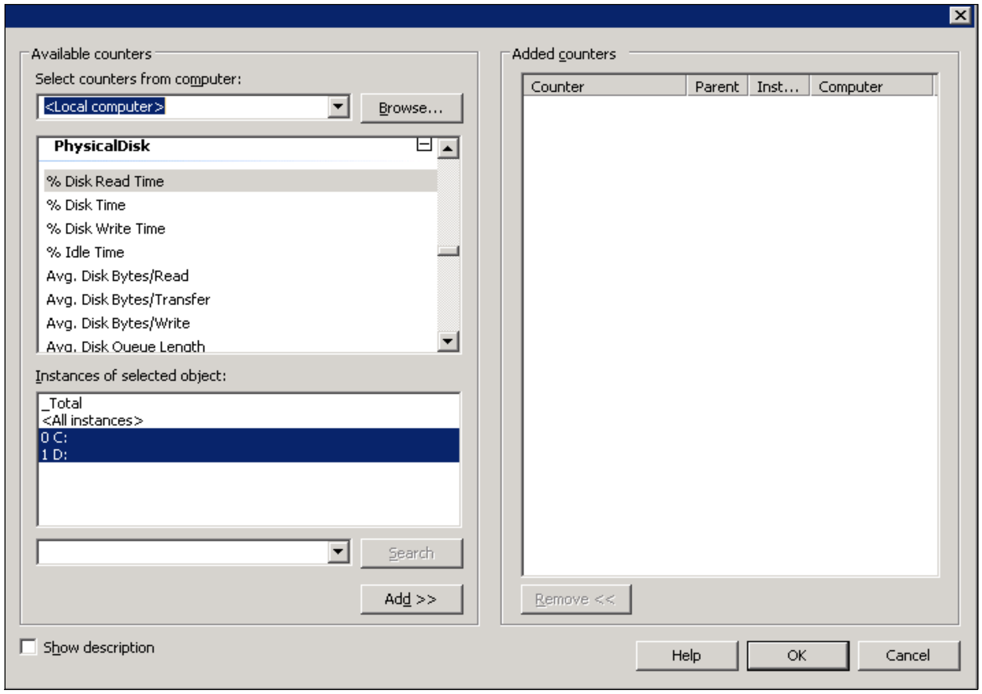


Figure D-21 Windows Server 2008 performance counters

9. The Create new Data Collector Set wizard window opens. In the Sample interval list box, set how frequently you capture the data. When configuring the interval, specify an interval that can provide enough granularity so that you can identify the issue. The rules are the same as for Windows Server 2003 data collection step 5 on page 662. Click **Next**.

> **Important:** Different spreadsheet software has different capabilities to work with large amounts of data. When configuring the interval, remember both the problem profile and the planned collection duration in order not to capture more data than can be reasonably analyzed. For data collection during several hours, you might want several files, one file for each hour, rather than one large file for several hours. Data collected for one hour with a 1 second interval gives you 3600 entries for each value. The total number of entries must be multiplied by the number of values collected.

10. You are prompted to enter the location of the log file and the file name as shown in Figure D-22 on page 675. In this example, the default file name is shown. After entering the file name and directory, click **Finish**.

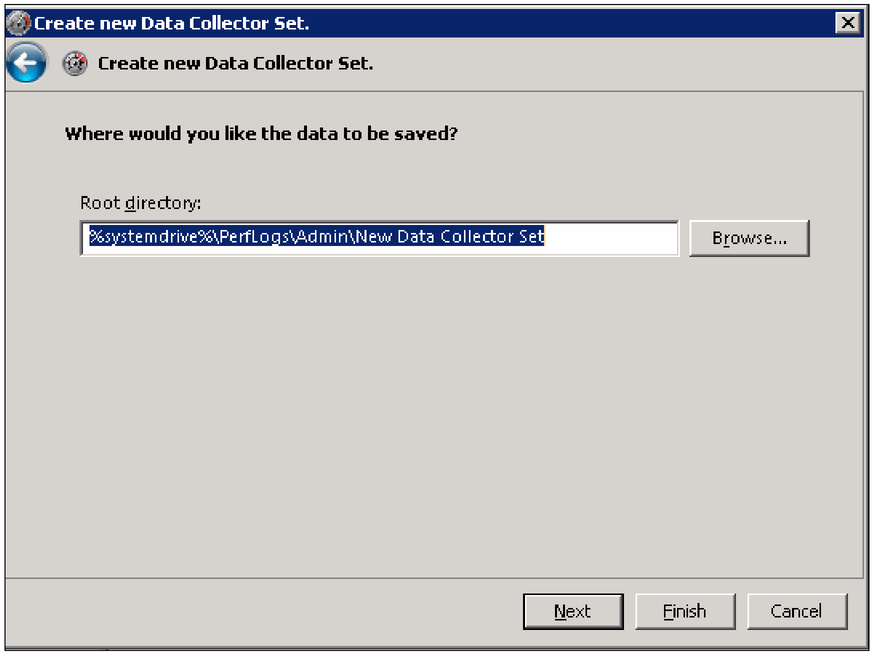


Figure D-22 Windows Server 2008 log file placement

11. You see a window similar to Figure D-23.

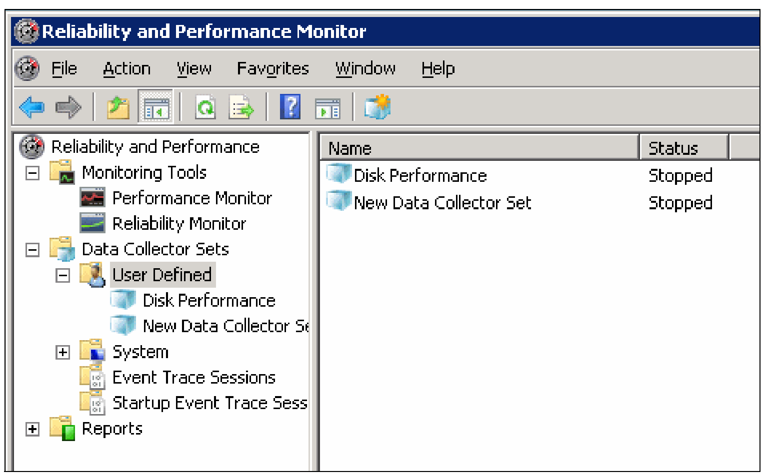


Figure D-23 Windows Server 2008 Performance Monitor with new Data Collector Set

12. Right-click **Data Collector Set** and select **Start**.
13. After you collect the data for the period that you want, you can select the **Stop** icon, or right-click **Data Collector Set** and select **Stop**.

Windows Server 2008 Export

Follow these steps to export the log file:

1. Under Monitoring Tools, select **Performance Monitor**.
2. Select the **log file** icon, select your log file as shown in Figure D-14 on page 670, and click **OK.** This step in the process is identical to Windows Server 2003.

3. In the Monitor View, click **Add**, select all the key PhysicalDisk counters for all instances, and click **Add**.
4. You see a window similar to Figure D-24.

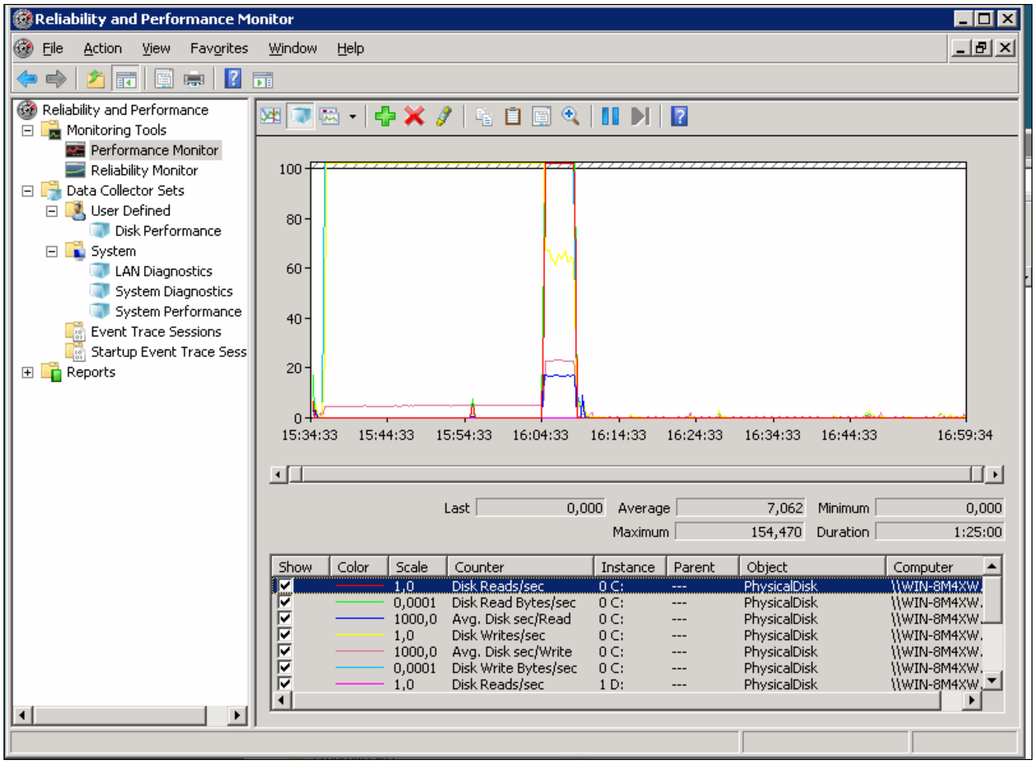


Figure D-24 Windows Server 2008 Performance Monitor

5. Right-click the Chart side and left-click **Save Data As**.
6. In the Save Data As window, select a file type of **Text File (Comma delimited)(*.csv)**.
7. Provide a file name, such as Disk_Performance.

For more information about Windows Server 2008 Performance Monitor, see this link:

http://technet.microsoft.com/en-us/library/cc771692%28WS.10%29.aspx

E

Post-processing scripts

This appendix provides several scripts for post-processing and correlating data from different sources.

Introduction

In our experience, it is often necessary to correlate data from different data sources. There are several methods to correlate data from different sources, including correlating the data manually, using Microsoft Excel macros, or using shell scripts. The information in this appendix demonstrates how to use PERL shell scripts to correlate and post-process data from multiple sources. Although these scripts can be converted to run on AIX fairly easily, we assume that you plan to run these scripts on a Microsoft Windows system. In this appendix, we describe the following information:

- Dependencies
- Sample scripts:
  - `perfmon-essmap.pl`
  - `iostat_aix53_essmap.pl`
  - `iostat_sun-mpio.pl`

By default, all the scripts print standard output. If you want to place the output in a file, redirect the output to a file.

> **Scripts:** The purpose of these scripts is not to provide a toolkit that addresses every possible configuration scenario; rather, it is to demonstrate several of the available possibilities.

Dependencies

To execute the scripts described in this section, you need to prepare your system.

Software

You need install Active Perl 5.6.*x* or later. At the time of writing this book, you can download PERL from the following website:

`http://www.activestate.com/activeperl/downloads`

Script location

We place the script in the same directory as the performance and capacity configuration data.

Script creation

To run these scripts, copy the entire contents of the script into a file and name the file with a "`.pl`" suffix.

Sample scripts

We explain the three scripts.

perfmon-essmap.pl

The purpose of this script (Example E-1 on page 679) is to correlate the disk performance data of Windows server with Subsystem Device Driver (SDD) **`datapath query essmap`** data. The rank column is invalid for DS8000s with multi-rank extent pools.

An example of the output of **`perfmon-essmap.pl`** is shown in Figure E-1 on page 679.

DATE	TIME	Subsystem Serial	Rank	LUN	Disk	Disk Reads/sec	Avg Read RT(ms)	Disk Writes/sec	Avg Write RT(ms)	Avg Total Time	Avg Read Queue Length
10/30/2008	11:59:43	75GB192	0	75GB1924805	Disk6	141.41	4.56	64.82	0.26	662.39	0.65
10/30/2008	11:59:43	75GB192	0	75GB1924801	Disk2	138.97	4.62	64.99	0.26	659.17	0.64
10/30/2008	11:59:43	75GB192	0	75GB1924802	Disk3	136.76	4.62	66.02	0.26	648.56	0.63
10/30/2008	11:59:43	75GB192	0	75GB1924804	Disk5	135.69	4.64	63.89	0.26	646.08	0.63
10/30/2008	11:59:43	75GB192	0	75GB1924803	Disk4	137.81	5.27	65.95	0.26	742.88	0.73
10/30/2008	11:59:43	75GB192	0	75GB1924800	Disk1	138.77	5.47	66.39	0.26	776.24	0.76
10/30/2008	12:00:43	75GB192	0	75GB1924805	Disk6	119.20	5.34	57.09	0.26	651.08	0.64

Figure E-1 Sample perfmon-sdd output

Several of the columns are hidden in order to fit the example output into the provided space.

To run the scripts, you need to collect performance and configuration data. This section describes the logical steps that are required for post-processing the perfmon and essmap data. The other scripts have a similar logical flow:

1. Collect Windows server performance statistics as described in Appendix D, “Microsoft Windows server performance log collection” on page 659.
2. Collect Windows server SDD or Subsystem Device Driver Device Specific Module (SDDDSM) output as described in 10.8.4, “Collecting configuration data” on page 413.
3. Place the output of the performance data and the configuration data in the same directory.
4. Open a shell.
5. Run the script. The script takes two input parameters. The first parameter is the file name for the Windows perfmon data and the second parameter is the file for the **`datapath query essmap`** output. Use the following syntax:

   `perfmon-essmap.pl` `Disk_perf.csv essmap.txt > DiskPerf_Essmap.csv`
6. Open the output file (`DiskPerf_Essmap.csv`) in Excel.

> **Important:** If your file is blank or contains only headers, you need to determine whether the issue is with the input files, or if you damaged the script when you created the script.

For the interpretation and analysis of the Windows data, see 10.8.6, “Analyzing performance data” on page 419.

Example: E-1 The perfmon-essmap.pl script

```
#!C:\Perl\bin\Perl.exe
################################################################
# Script:  perfmon-essmap.pl
# Purpose: Correlate perfmon data with datapath query essmap
# and create a normalized data for input into spreadsheet
# The following perfmon counters are needed:
# Avg Disk sec/Read
# Avg Disk sec/Write
# Disk Reads/sec
# Disk Writes/sec
# Disk Total Response Time - Calculated to sum avg rt * avg i/o rate
# Disk Read Queue
# Disk Write Queue
# Disk Read KB/sec - calc
# Disk Write KB/sec - calc
################################################################
$file = $ARGV[0];  # Set perfmon csv output to 1st arg
```

```
open(PERFMON,$file) or die "cannot open perfmon $file\n";

$datapath = $ARGV[1];# Set 'datapath query essmap' output to 2nd arg
open(DATAPATH,$datapath) or die "cannot open datapath $datapath\n";

#########################################################################
# Read in essmap and create hash with hdisk as key and LUN SN as value #
#########################################################################
while (<DATAPATH>) {
   if (/^$/) { next; }# Skip empty lines
   if (/^Disk /) { next; }# Skip empty lines
   if (/^--/) { next; }# Skip empty lines
   @line = split(/\s+|\t/,$_);# Build temp array of current line
   $lun = $line[4];# Set lun ID
   $path = $line[1];# Set path
   $disk = $line[0];# Set disk#
   $hba = $line[2];# Set hba port - use sdd gethba.exe to get wwpn
   $size = $line[7];# Set size in gb
   $lss = $line[8];# Set ds lss
   $vol = $line[9];# Set DS8K volume
   $rank = $line[10];# Set rank - DOES NOT WORK FOR ROTATE VOLUME OR ROTATE extent
   $c_a = $line[11]; # Set the Cluster and adapter accessing rank
   $dshba = $line[13];# Set shark hba - this is unusable with perfmon which isn't aware of paths
   $dsport = $line[14];# Set shark port physical location - this is unusable with perfmon which isn't aware
of paths
   $lun{$disk} = $lun;# Set the LUN in hash with disk as key for later lookup
   $disk{$lun} = $disk;# Set vpath in hash with lun as key
   $lss{$lun} = $lss;# Set lss in hash with lun as key
   $rank{$lun} = $rank;# Set rank in hash with lun as key
   $dshba{$lun} = $dshba;# Set dshba in hash with lun as key - this is unusable with perfmon which isn't
aware of paths
   $dsport{$lun} = $dsport;# Set dsport in hash with lun as key - this is unusable with perfmon which isn't
aware of paths
   if (length($lun) > 8) {
      $ds = substr($lun,0,7);# Set the DS8K serial
   } else {
      $ds = substr($lun,3,5);# Set the ESS serial
   }
   $ds{$lun} = $ds;     # set ds8k in hash with LUN as key
}
################
# Print Header #
################
print "DATE,TIME,Subsystem Serial,Rank,LUN,Disk,Disk Reads/sec, Avg Read RT(ms),Disk Writes/sec,Avg Write
RT(ms),Avg Total Time,Avg Read Queue Length,Avg Write Queue Length,Read KB/sec,Write KB/sec\n";

##################################################################################################
# Read in perfmon and create record for each hdisk and split the first column into date and time #
##################################################################################################
while (<PERFMON>) {
   if (/^$/) { next; }                         # Skip empty lines
   if (/^--/) { next; }                        # Skip empty lines
   if (/PDH-CSV/) {
      @header = split(/,/,$_);              # Build header array
      shift(@header);                                 # Remove the date element
      unshift(@header,"Date","Time"); # Add in date, time
      next;                                                        # Go to next line
   }
   @line = split(/\t|,/,$_);                 # Build temp array for current line
   @temp = split(/\s|\s+|\"/,$line[0]);# Split the first element into array
```

```
    $date = $temp[1];                                  # Set date to second element of array
    $time = $temp[2];                                  # Set time to third element of array
    shift(@line);                                        # Remove old date
    unshift(@line,"\"$date\"","\"$time\"");# Add in the new date and time
    chomp(@line);                                        # Remove carriage return at end
    for ($i=0; $i<=$#line; $i++) {     # Loop through each element in line
          $line[$i] =~ s/"//g;                       # Remove double quotes from input line
          $header[$i] =~ s/"//g;                 # Remove double quotes from header array
          @arr = split(/\\/,$header[$i]);# Split current header array element
          $hostname = $arr[2];                       # Extract hostname from header
          $disk = $arr[3];                              # Set disk to the 4th element
          $counter = $arr[4];                        # Set counter to 5th element
          if ($disk =~ /Physical/) {         # If we find Physical Object
             if ($disk =~ /Total/) { next; }# If disk instance is Total then skip
             @tmpdisk = split(/Physical|\(|\s|\)/,$disk);# Create temp array of disk name
             $newdisk = $tmpdisk[1] . $tmpdisk[2];# Create newly formatted disk name to match SDD output
             if ($counter =~ /Avg. Disk sec\/Read/) {# If counter is Avg. Disk sec/Read
                $diskrrt{$date}{$time}{$newdisk} = $line[$i]*1000;# Then set disk read response time hash
             }
             if ($counter =~ /Avg. Disk sec\/Write/) {# If counter is Avg. Disk sec/Write
                $diskwrt{$date}{$time}{$newdisk} = $line[$i]*1000;# Then set disk write response time has
             }
             if ($counter =~ /Disk Reads\/sec/) {# If counter is Disk Reads/sec
                $diskreads{$date}{$time}{$newdisk} = $line[$i];# Then set Disk Reads/sec hash
             }
             if ($counter =~ /Disk Writes\/sec/) {# If counter is Disk Writes/sec
                $diskwrites{$date}{$time}{$newdisk} = $line[$i];# Then set Disk Writes/sec hash
             }
             if ($counter =~ /Avg. Disk Read Queue Length/) {# If counter is Disk Read Queue Length
                $diskrql{$date}{$time}{$newdisk} = $line[$i];# Then set Disk Read Queue Length hash
             }
             if ($counter =~ /Avg. Disk Write Queue Length/) {# If counter is Disk Write Queue Length
                $diskwql{$date}{$time}{$newdisk} = $line[$i];# Then set Disk Write Queue Length hash
             }
             if ($counter =~ /Disk Read Bytes\/sec/) {# If counter is Disk Read Bytes/sec
                $diskrkbs{$date}{$time}{$newdisk} = $line[$i]/1024;# Then calc kb an set in hash
             }
             if ($counter =~ /Disk Write Bytes\/sec/) {# If counter is Disk Write Bytes/sec
                $diskwkbs{$date}{$time}{$newdisk} = $line[$i]/1024;# Then calc kb and set in hash
             }
          }
    }
}

### Print out the data here - key is date, time, disk
while (($date,$times) = each(%diskrrt)) {# Loop through each date-time hash
   while (($time,$disks) = each(%$times)) {# Nested Loop through each time-disk hash
      while (($disk,$value) = each(%$disks)) {# Nest loop through disk-value hash
         $diskrrt = $diskrrt{$date}{$time}{$disk};# Set shortnames for easier print
         $diskwrt = $diskwrt{$date}{$time}{$disk};
         $diskreads = $diskreads{$date}{$time}{$disk};
         $diskwrites = $diskwrites{$date}{$time}{$disk};
         $total_time = ($diskrrt*$diskreads)+($diskwrt*$diskwrites);
         $diskrql = $diskrql{$date}{$time}{$disk};
         $diskwql = $diskwql{$date}{$time}{$disk};
         $diskrkbs = $diskrkbs{$date}{$time}{$disk};
         $diskwkbs = $diskwkbs{$date}{$time}{$disk};
         $lun = $lun{$disk};                        # Lookup lun for current disk
```

```
        print
"$date,$time,$ds{$lun},$rank{$lun},$lun,$disk,$diskreads,$diskrrt,$diskwrites,$diskwrt,$total_time,$diskrql
,$diskwql,$diskrkbs,$diskwkbs\n";
      }
   }
}
```

iostat_aix53_essmap.pl

The purpose of this script (Example E-2) is to correlate AIX 5.3 **iostat -D** data with SDD **datapath query essmap** output for analysis. Beginning in AIX 5.3, the **iostat** command provides the ability to continuously collect read and write response times. Prior to AIX 5.3, **filemon** was required to collect disk read and write response times.

Use this syntax:

iostat_aix53_essmap.pl -i [iostatD.out] -e [essmap.out] > [output.csv]

Example output of the `iostat_aix53_essmap.pl` is shown in Figure E-2.

IME	STORAGE	DS HBA	DS PORT	LSS	RANK	LUN	VPATH	HDISK	#BUSY	KBPS	KB READ	KB WRITE
0:00:00	75P2831	R1-B1-H3-ZA	30	29	ffe3	75P28311D0A	vpath13	hdisk51	-	0.61	0.14	0.47
0:00:00	75P2831	R1-B1-H3-ZA	30	23	ffe9	75P2831170A	vpath12	hdisk50	0.80	8.10	8.10	-
0:00:00	75P2831	R1-B5-H3-ZB	431	23	ffe9	75P2831170A	vpath12	hdisk52	0.60	8.60	8.60	-
0:00:00	75P2831	R1-B5-H3-ZB	431	29	ffe3	75P28311D0A	vpath13	hdisk53	-	0.95	0.22	0.73
0:00:00	75P2831	R1-B6-H1-ZA	500	29	ffe3	75P28311D0A	vpath13	hdisk55	-	0.54	0.21	0.33
0:00:00	75P2831	R1-B6-H1-ZA	500	23	ffe9	75P2831170A	vpath12	hdisk54	0.80	8.90	8.90	-
0:00:00	75P2831	R1-B8-H1-ZA	700	29	ffe3	75P28311D0A	vpath13	hdisk57	0.10	0.52	0.12	0.40
0:00:00	75P2831	R1-B8-H1-ZA	700	23	ffe9	75P2831170A	vpath12	hdisk56	0.60	7.60	7.60	-
1:01:00	75P2831	R1-B1-H3-ZA	30	29	ffe3	75P28311D0A	vpath13	hdisk51	-	4.80	3.80	1.00

Figure E-2 The iostat_aix53_essmap.pl sample output

Several of the columns are hidden to fit the example output into the provided space.

Example: E-2 iostat_aix53_essmap.pl

```
#!C:\Perl\bin\Perl.exe
###############################################################
# Script:  iostat_aix53_essmap.pl
# Purpose: Process AIX 5.3 iostat disk and essmap
# normalize data for input into spreadsheet
###############################################################
use Getopt::Std;
#######################################################################################
main();                                                     # Start main logic
sub main{
   parseparms();                                     # Get input parameters
   readessmap($essmap);                        # Invoke routine to read datapath query essmap output
   readiostat($iostat);                        # Invoke routine to read iostat
}
############## Parse input parameters ##################################################
sub parseparms {
   $rc = getopts('i:e:');                  # Define inputs
   $iostat = $opt_i;                             # Set value for iostat
   $essmap = $opt_e;                             # Set value for dp query essmap
   (defined $iostat) or usage();   # If iostat is not set exit
   (defined $essmap) or usage();   # If essmap is not set exit
}
############ Usage #####################################################################
sub usage {
```

```
   print "\nUSAGE: iostat_aix53_essmap.pl [-ie] -h";
   print "\n -i  The file containing the iostat output";
   print "\n -e  The file containing the datapath query essmap output";
   print "\n  ALL ARGUMENTS ARE REQUIRED!\n";
   exit ;
}

### Read in pcmpath and create hash with hdisk as key and LUN SN as value ############
$file = $ARGV[0];                                 # Set iostat -D for 1st arg
$essmap = $ARGV[1];                            # Set 'datapath query essmap' output to 2nd arg

### Read in essmap and create hash with hdisk as key and LUN SN as value
sub readessmap($essmap) {
   open(ESSMAP,$essmap) or die "cannot open $essmap\n";
   while (<ESSMAP>) {
      if (/^$/) { next; }                      # Skip empty lines
      if (/^--/) { next; }                     # Skip empty lines
      if (/^Disk/) { next; }                # Skip header
      @line = split(/\s+|\t/,$_);     # Build temp array
      $lun = $line[4];                                # Set lun
      $hdisk = $line[1];                          # set hdisk
      $vpath = $line[0];                          # set vpath
      $hba = $line[3];                               # set hba
      $lss = $line[8];                               # set lss
      $rank = $line[10];                          # set rank
      $dshba = $line[13];                         # set shark hba
      $dsport = $line[14];                        # set shark port
      $vpath{$lun} = $vpath;                # Set vpath in hash
      $lss{$lun} = $lss;                          # Set lss in hash
      $rank{$lun} = $rank;                        # Set rank in hash
      $dshba{$hdisk} = $dshba;            # Set dshba in hash
      $dsport{$hdisk} = $dsport;          # Set dsport in hash
      $lun{$hdisk} = $lun;           # Hash with hdisk as key and lun as value
      if (length($lun) > 8) {
         $ds = substr($lun,0,7);             # Set ds serial to first 7 chars
      } else {
         $ds = substr($lun,3,5);             # or this is ESS and only 5 chars
      }
      $ds{$lun} = $ds;                            # set the ds serial in a hash
   }
}
### Read in iostat and create record for each hdisk
sub readiostat($iostat) {
   ### Print Header
   print "TIME,STORAGE SN,DS HBA,DS PORT,LSS,RANK,LUN,VPATH,HDISK,#BUSY,KBPS,KB READ PS,KB WRITE
PS,TPS,RPS,READ_AVG_SVC,READ_MIN_SVC,READ_MAX_SVC,READ_TO,WPS,WRITE_AVG_SVC,WRITE_MIN_SVC,WRITE_MAX_SVC,WRI
TE_TO,AVG QUE,MIN QUE, MAX QUE,QUE SIZE\n";
   $time = 0;                                         # Set time variable to 0
   $cnt = 0;                                          # Set count to zero
   $newtime = 'Time_1';                               # Set a relative time stamp
   open(IOSTAT,$iostat) or die "cannot open $iostat\n";# Open iostat file
   while (<IOSTAT>) {                           # Read in iostat file
      if ($time == 0) {
         if (/^Mon|^Tue|^Wed|^Thu|^Fri|^Sat|^Sun/) {# This only works if a time stamp was in file
            $date_found = 1;                        # Set flag for
            @line = split(/\s+|\s|\t|,|\//,$_);# build temp array
            $date = $line[1] . " " . $line[2] . " " . $line[5];# Create date
            $time = $line[3];                       # Set time
            $newtime = $time;                       # Set newtime
            $interval = 60;                         # Set interval to 60 seconds
```

```
            next;
        }  else {
            $time++;                                        # Set time counter to 1 if no time is in
file
        }
    }
    if (/^#|System/) { next; }        # Skip notes
    if (/^hdisk|^dac/) {                    # If line starts with hdisk
        @line = split(/\s+|\t|,|\//,$_);# build temp array
        $pv = $line[0];                         # Set physical disk to 1st element
        $xfer = 1;                                  # we are in the transfer stanza now
        next;
    }
    if (/-------/) {
        $cnt++;                                         # count the number of stanzas
        if ($date_found == 1) {           # If date flag is set
            $newtime = gettime($time,$interval,$cnt);# get time based on original time,cnt, and interval
        } else {
            $time++;
            $newtime = 'Time_' . $time;# Set a relative time stamp
        }
        next;                                           # Go to next line
    }
    if ($xfer == 1) {                           # If in transfer section
        @line = split(/\s+|\t|,|\//,$_);# build temp array
        $busy{$pv} = $line[1];            # Set busy time
        if ($line[2] =~ /[K]/) {        # If K is in value
            $line[2] =~ s/K//g;                 # remove K
            $kbps{$pv} = $line[2];          # Set value
        } elsif ($line[2] =~ /[M]/) {# If M
            $line[2] =~ s/M//g;                 # remove M
            $kbps{$pv} = $line[2]*1024;# Multiply value by 1024
        } else {
            $kbps{$pv} = $line[2]/1024;# Else its bytes so div by 1024
        }
        $tps{$pv} = $line[3];             # Set transfer per sec
        if ($line[4] =~ /[K]/) {       # If K then remove K
            $line[4] =~ s/K//g;
            $bread{$pv} = $line[4];
        } elsif ($line[4] =~ /[M]/) {# If Mbytes convert to K
            $line[4] =~ s/M//g;
            $bread{$pv} = $line[4]*1024;
        } else {
            $bread{$pv} = $line[4]/1024;# Else bytes and convert ot K
        }
        if ($line[5] =~ /[K]/) {        # Same logic as above
            $line[5] =~ s/K//g;
            $bwrtn{$pv} = $line[5];
        } elsif ($line[5] =~ /[M]/) {
            $line[5] =~ s/M//g;
            $bwrtn{$pv} = $line[5]*1024;
        } else {
            $bwrtn{$pv} = $line[5]/1024;
        }
        $xfer = 0;
        next;
    }
    if (/read:/) { $read = 1; next; }
    if ($read == 1) {                           # If read flag is set
        @line = split(/\s+|\t|,|\//,$_);# build temp array
```

```
         $rps{$pv} = $line[1];              # set reads per second
         $rrt{$pv} = $line[2];              # set read rt
         $rminrt{$pv} = $line[3];        # set read min rt
         if ($line[4] =~ /[S]/) {
            $line[4] =~ s/S//g;
            $rmaxrt{$pv} = $line[4]*1000;# set read max rt convert from seconds if necessary
         } else {
            $rmaxrt{$pv} = $line[4];
         }
         $rto{$pv} = $line[5];
         $read = 0;
         next;
      }
      if (/write:/) { $write = 1; next; }# If write flag is set
      if ($write == 1) {
         @line = split(/\s+|\t|,|\//,$_);# build temp array
         $wps{$pv} = $line[1];              # set writes per sec
         $wrt{$pv} = $line[2];              # set write rt
         $wminrt{$pv} = $line[3];        # set min write rt
         if ($line[4] =~ /[S]/) {
            $line[4] =~ s/S//g;
            $wmaxrt{$pv} = $line[4]*1000;# set max rt and convert from secs if necessary
         } else {
            $wmaxrt{$pv} = $line[4];
         }
         $wto{$pv} = $line[5];
         $write = 0;
         next;
      }
      if (/queue:/) { $queue = 1; next; }# If queue flag is set
      if ($queue == 1) {
         @line = split(/\s+|\t|,|\//,$_);# build temp array
         $qt{$pv} = $line[1];                  # set queue time
         $qmint{$pv} = $line[2];            # set queue min time
         $qmaxt{$pv} = $line[3];            # Set queue max time
         $qsize{$pv} = $line[4];
         $queue = 0;
         $time{$pv} = $newtime;
         $lun = $lun{$pv};
         print
"$time{$pv},$ds{$lun},$dshba{$pv},$dsport{$pv},$lss{$lun},$rank{$lun},$lun,$vpath{$lun},$pv,$busy{$pv},$kbp
s{$pv},$bread{$pv},$bwrtn{$pv},$tps{$pv},$rps{$pv},$rrt{$pv},$rminrt{$pv},$rmaxrt{$pv},$rto{$pv},$wps{$pv},
$wrt{$pv},$wminrt{$pv},$wmaxrt{$pv},$wto{$pv},$qt{$pv},$qmint{$pv},$qmaxt{$pv},$qsize{$pv}\n";
         next;
      }
      if (/^--/) { $time++; }
   }
}
############ Convert Time ####################################################################
sub gettime() {
   my $time = $_[0];
   my $interval = $_[1];
   my $cnt = $_[2];
   my $hr = substr($time,0,2);
   my $min = substr($time,3,2);
   my $sec = substr($time,6,2);
   $hrsecs =$hr * 3600;
   $minsecs = $min * 60;
   my $addsecs = $interval * $cnt;
   my $totsecs = $hrsecs + $minsecs + $sec + $addsecs;
```

```
   $newhr = int($totsecs/3600);
   $newsecs = $totsecs%3600;
   $newmin =  int($newsecs/60);
   $justsecs = $newsecs%60;

   $newtime = $newhr . ":" . $newmin . ":" . $justsecs;
   return $newtime;
}
```

iostat_sun-mpio.pl

The purpose of this script (Example E-3) is to reformat Solaris **iostat -xn** data so that it can be analyzed in a spreadsheet. The logical unit number (LUN) identification works properly with Solaris systems running MPxIO only. In the **iostat -xn** output with MPxIO, there is only one disk shown per LUN in the output. There are no flags required. Use this syntax:

iostat_sun-mpio.pl iostat.out > output.csv

Example: E-3 The iostat_sun-mpio.pl script

```
#!C:\Perl\bin\Perl.exe
################################################################
# Script:  iostat_sun-mpio.pl
# Purpose: Correlate disk# with Sun iostat and extract LUN
# normalize data for input into spreadsheet
################################################################
use Getopt::Std;
$file = $ARGV[0];                               # Set iostat -xcn to 1st arg
open(IOSTAT,$file) or die "cannot open $file\n";# Open file

### Print Header
print "DATE,TIME,LUN,HDISK,#READS,AVG_READ_KB,WRITES,AVG_WRITE_KB,AVG_WAIT,AVG_SVC\n";

### Read in iostat and create record for each hdisk
$cnt=0;
$time=0;
while (<IOSTAT>) {
   if ($time == 0) {
      if (/^Mon|^Tue|^Wed|^Thu|^Fri|^Sat|^Sun/) {
         @line = split(/\s+|\s|\t|,|\//,$_);# build temp array
         $date = $line[1] . " " . $line[2] . " " . $line[5];
         $time = $line[3];
         $interval = 60;
         next;
      }  else {
         $date = 'Date Not Available';
         $time++;                                           # Set time counter to 1 if no time is in file
      }
   }
   if (/tty|tin|md|r\/s|^$|#/) { next;}
   if (/extended/) {
      $cnt++;                                           # count the number of stanzas
      if ($date_found == 1) {            # If date flat is set
         $newtime = gettime($time,$interval,$cnt);# get time based on original time,cnt, and interval
      } else {
         $newtime = 'Time' . $time;# Set a relative time stamp
         $time++;
      }
```

```
      next;
   }
   @line = split(/\s+|\t|,|\//,$_);# Build temp array for each line
   $pv = $line[11];                              # Set pv to 11th element
   $lun = substr($pv,31,4);           # Set lun to substring of pv
   $date{$pv} = $date;                      # Set date to date hash
   $time{$pv} = $newtime;                # Set time to time hash
   $reads{$pv} = $line[1];               # Set read hash
   $writes{$pv} = $line[2];           # Set write hash
   $readkbps{$pv} = $line[3];         # Set read kbps
   $writekbps{$pv} = $line[4];     # Set write kbps
   $avg_wait{$pv} = $line[7];         # Set avg wait time
   $avg_svc{$pv} = $line[8];          # Set avg response time
   print "$date{$pv},$time{$pv}, $lun,$pv,
$reads{$pv},$readkbps{$pv},$writes{$pv},$writekbps{$pv},$avg_wait{$pv},$avg_svc{$pv}\n";
}


sub gettime() {
   my $time = $_[0];
   my $interval = $_[1];
   my $cnt = $_[2];
   my $hr = substr($time,0,2);
   my $min = substr($time,3,2);
   my $sec = substr($time,6,2);
   $hrsecs =$hr * 3600;
   $minsecs = $min * 60;
   my $addsecs = $interval * $cnt;
   my $totsecs = $hrsecs + $minsecs + $sec + $addsecs;

   $newhr = int($totsecs/3600);
   $newsecs = $totsecs%3600;
   $newmin =  int($newsecs/60);
   $justsecs = $newsecs%60;

   $newtime = $newhr . ":" . $newmin . ":" . $justsecs;
   return $newtime;
}
```

Index

A

AAL 8, 38
achievement value 23
address group 72
address groups 73
Advanced Copy Services 534, 538, 548
affinity 61
agile view 388
AIO 355–356
aioo 356
AIX
  filemon command 366–367
  lvmstat command 360
  nmon command 362
  topas command 361
aligned partition 440–442
allocation 61
anticipatory 458
anticipatory I/O Scheduler 457
application workload 163
arbitrated loop 313
array site 39, 58
array sites 58, 75
arrays 58, 61
arrays across loops see AAL
Assymetric Logical Unit Access (ALUA) 389
auto rebalance 626
automatic data relocation 626
auto-rebalance 63–64, 80, 84, 89, 99, 120, 123, 126
auto-reblance 126

B

balanced
  DB2 workload 564
  IMS workload 580
bandwidth 597, 599
benchmark
  cautions 649
  requirements 646–647
benchmarking 645–646, 648
  goals 646
bio structure 449
block layer 449
block size 462, 469
buffer pools 570
BusLogic 443

C

cache 27–29, 242, 246–449, 469
  as intermediate repository 27
  fast writes 28
cache and I/O operations
  read operations 28
  write operations 28
cache information 242
cache management 28–29
  performance of SARC 30
capacity 602, 606
Capacity Magic 50
CEC 72, 130
characterize performance 395
choosing
  CKD volume size 508
  disk size with DB2 UDB 576
CKD volumes 65, 70, 75
  allocation and deletion 70
CLOCK 32
commands
  cron 391
  datapath 358
  dd 173, 381
  filemon 366–367
  lvmstat 360
  nmon 362
  sar 389, 391, 474
  SDD datapath 319
  topas 361
compatability mode 437
Complete Fair Queuing 457–458, 460
Completely Fair Queuing I/O Scheduler 457
Concurrent I/O 341
concurrent write operation 506
configuration states 71
CONN time 522–523, 525
connectivity
  Global Copy 604
considerations
  DB2 performance 563
  DB2 UDB performance 567, 572
  logical disks in a SAN 314
  z/OS planning 519
  zones 313
Consistency Group
  drain time 613
constraints
  definition 260
containers 567, 569–571, 573
control data sets 621
Copy Pending 603
Copy Services 585
  FlashCopy 586–588
  FlashCopy objectives 588
  introduction 586
  Metro Mirror 596
  Metro Mirror configuration considerations 597
cron command 391
cross-tier data relocation 123
cross-tier management 84

CSCAN 32

D

DA 35
data mining 165
data rate droop effect 511
data warehousing 165
databases 559–562, 568
  buffer pools 570
  containers 567, 569, 571
  DB2 in a z/OS environment 560
  DB2 overview 561
  DB2 prefetch 565–566, 571
  extent size 570–571, 575
  extents 570
  IMS in a z/OS environment 578, 581
  logs 563, 568, 571, 578
  MIDAWs 566
  multi-pathing 577
  page cleaners 571
  page size 570–571, 575
  pages 570–572
  parallel operations 572
  partitiongroups 569
  partitioning map 569
  partitions 569
  prefetch size 575
  selecting DB2 logical sizes 574
  tables, indexes and LOBs 570
  tablespace 569–571
  workload 161, 560
dataset placement 621
datastore 427
DB2 559–561
  in a z/OS environment 560
  logging 161–162
  overview 561
  prefetch 565–566, 571
  query 162
  recovery data sets 562
  selecting logical sizes 574
  storage objects 561
  transaction environment 162
DB2 UDB 567–569
  container 567, 569–570, 573
  instance 568–569
  performance considerations 567, 572
  striping 574
  tablespace 561, 568–569
dd command 173, 381
DDM 52
DDMs 34–35, 38–39
deadline 457–458
deadline I/O scheduler 457
Dense Wavelength Division Multiplexing (DWDM) 604
description and characteristics of a SAN 313
device adapter see DA
device busy 504
Device Mapper 452
Device Mapper - Multipath I/O
  DM-MP 451
Device Mapper - Mutlipath I/O (DM-MPIO) 451–452
DFW Bypass 566
DFWBP 566
digital video editing 166
direct connect 313
Direct I/O 329, 340, 370
dirty buffer 449
  flushing 449
DISC time 523–524, 528
Discovered Direct I/O 370
disk bottlenecks
  iostat command 470, 473
  vmstat command 470
disk drive module 52
disk enclosure 38
disk I/O subsystem
  block layer 449
  cache ??–447447–449, 469
  I/O subsystem architecture 447
Disk Magic 176–177, 609
  modeling 177, 179
Disk Magic Control file 182
Disk Magic for open systems 198
Disk Magic for zSeries 181
  hardware configuration 197, 213
disk subsystem 35
  fragmentation 401
disk xfer 354–355
diskpart 443
disks
  configuring for SAN 314
  size with DB2 UDB 576
distance 596–597
distributions
  Linux 445–446
DMC 182
drain period 613
drain time 613–614
DS8000
  AAL
  cache management 28–29
  Copy Services 585
  DA
  DDMs 34–35, 38–39
  disk enclosure 38
  FICON 7–9
  FICON port concurrency 527
  hardware overview 5
  host adapter 7
  host attachment 312
  I/O priority queuing 10
  Multiple Allegiance 9
  PAV
  POWER5 7–8
  priority queuing 507
  SDD
  z/OS Global Mirror
DS8000 nI/O ports 603
DS8100

processor memory 27
DS8300 43
dual parity 53
DVR 71
dynamic alias management 501, 509
dynamic buffer cache 383
Dynamic extent pool merge 16
dynamic extent pool merge 62, 79
Dynamic Path Reconnect (DPR) 48
Dynamic Path Selection (DPS) 48
dynamic PAV 501–502
dynamic volume expansion 71
dynamic volume relocation 71, 79

E

EAM 15, 63, 65, 68, 71, 119
Easy Tier 10, 23, 78, 80, 89–91, 115, 120, 122, 456, 614, 627
Automatic mode 11
automatic mode) 23
Manual mode 15
Easy Tier Monitoring 17
engineering and scientific applications 165–166
ENT 138
Enterprise Volume Management System (EVMS) 455
enterprise-class drives 138
ESCON 48
ESE 66
ESX Server
multipathing 430
ESX server 426–427, 430
esxcfg-mpath 431
esxtop 433–434
examples
datapath query output 358
FICON connection 323
filemon command outputs 366
larger vs. smaller volumes - random workload 509
rank device spreading 344, 385
sar output 390
zoning in a SAN environment 314
Ext2 459
Ext3 459, 461
eXtended File System 459
Extended Remote Copy see z/OS Global Mirror
extent allocation method 15, 61, 63, 65, 68, 71, 76, 84, 99, 119
extent allocation methods 79
extent pool 61, 590, 595
extent pools 61, 65
extent size 570–571, 575
extent space efficient 66
extent type 59, 61
extents 570

F

fast write 28
fast writes 28
FATA 595
FC adapter 318
FCP
supported servers 312
fcstat 363, 365
Fibre Channel 312
distances 46
host adapters 46, 49, 147
topologies 313
Fibre Channel and FICON host adapters 46
FICON 7–9, 46, 48, 320, 500, 509, 511
host adapters 46, 49, 147
host attachment 322
FICON Express2 511
FICON Express4 320, 322, 511, 518
FICON Open Exchange 527
filemon 357, 366
filemon command 366–367
filemon measurements 366
fixed block 73
fixed block LUNs 64
Fixed policy 431
FlashCopy 586–588
performance 588–589
FlashCopy objectives 588
FlashCopy SE 587
fragmentation, disk 401
fsbuf 354

G

Global Copy 586, 602
capacity 606
connectivity 604
distance considerations 604
DS8000 I/O ports 603
performance 602, 605–606, 608, 610
planning considerations 605
scalability 606
Global Mirror
add or remove storage servers or LSSs 619
avoid unbalanced configurations 614
Consistency Group drain time 613
growth within configurations 617
performance 608
performance aspects 608
performance considerations at coordination time 611
remote storage server configuration 614
volumes 611, 614–615, 618
guidelines
z/OS planning 519

H

hardware configuration planning
allocating hardware components to workloads 5
array sites, RAID, and spares 39
cache as intermediate 27
Capacity Magic 50
disk subsystem 35
DS8000 overview 5
DS8300 43

Fibre Channel and FICON host adapters 46
host adapters 34, 46
I/O enclosures 34–35
modeling your workload 5
multiple paths to open systems servers 48
multiple paths to zSeries servers 48
performance numbers 4
processor complex 26–27
processor memory 27, 34
RIO-G interconnect and I/O enclosures 34
RIO-G loop 35
spreading host attachments 48
storage server challenge 4
switched disk architecture 36
tools to aid in hardware planning 49
whitepapers 49
HCD (hardware configuration definition) 48
High performance FICON (zHPF) 512
High Performance FileSystem (HFS) 382
homogeneous extent pools 84
host adapter (HA) 7
host adapters 34, 46
Fibre Channel 46, 49, 147
FICON 46, 49, 147
host attachment 74–75
description and characteristics of a SAN 313
FC adapter 318
Fibre Channel 312
Fibre Channel topologies 313
FICON 320
single path mode 318
supported Fibre Channel 312
host attachments 74
host connection 74
HP MirrorDisk/UX 385
hybrid 81
hybrid extent pools 84
HyperPAV 501, 503

I

I/O
priority queuing 507
I/O elevator 447, 449, 460
anticipatory 458
Complete Fair Queuing 457–458, 460
deadline 457–458
NOOP 458
I/O elevator model 449
I/O enclosure 34–35
I/O latency 27
I/O Priority Manager 17, 80, 90, 115, 456
I/O priority queuing 10
I/O scheduler 449, 469
I/O subsystem architecture 447
IBM FlashCopy SE 66
IBM TotalStorage Multipath Subsystem Device Driver see SDD
IBM TotalStorage XRC Performance Monitor 624
implement the logical configuration 151
importance value 23
IMS 559, 578
logging 578
performance considerations 578–579
WADS 579
IMS in a z/OS environment 578, 581
index 561–562, 570
indexspace 562
installation planning
host attachment 312
instance 568–569
Intelligent Write Caching (IWC) 32
IntelliMagic Direction 176
inter-tier management 84
intra-tier data relocation 123
intra-tier management 84
intra-tier performance management 120
IOCDS 48
ionice 460
IOP/SAP 522, 524–525
IOPM 456
IOSQ time 503, 523
iostat command 470, 473
IWC 32

J

Journal File System 458, 461
Journaled File System (JFS) 383
journaling
mode 459, 461–462
Journals 621

L

large form factor 107, 113
large volume support
planning volume size 510
LCU 72
legacy AIO 355–356
LFF 107, 113
links
Fibre Channel 597
Linux 445–446
load balancing
SDD 318
locality of reference 448
log buffers 578
logging 578
logical configuration
disks in a SAN 314
logical control unit 72
logical device
choosing the size 508
configuring in a SAN 314
planning volume size 510
logical paths 598
Logical Session 622
logical subsystem see LSS
Logical Volume Manager 454
logical volumes 57, 59, 64
logs 563, 568, 571, 578

LSI Logic 443
LSS 72, 597–598, 619
add or remove in Global Mirror 619
LSS design 599
LUN
masking 314, 316
LUN queue depth 436
LUNs
allocation and deletion 70
fixed block 64
LVM 454
LVM 2 454
LVM2 454
lvmstat 357, 360–361
lvmstat command 360

M
managed 69, 80–81, 84, 122
Managed Disk Group 540
Managed Disks Group 535
Manual volume rebalance 16
manual volume rebalance 62–63, 129
maximum drain time 613
metric 240, 262, 265
Metro Mirror 596
adding capacity in new DS6000s 602
addition of capacity 602
bandwidth 597, 599
distance 596–597
Fibre Channel links 597
logical paths 598
LSS design 599
performance 597, 600–601
scalability 602
symmetrical configuration 599
volumes 596, 598, 601
Metro Mirror configuration considerations 597
Metro/Global Mirror
performance 617, 625
Microsoft Cluster Services (MSCS) 428
micro-tiering 63, 80, 120, 123
MIDAW 566
MIDAWs 566
modeling your workload 5
monitor host workload 169
open systems servers 169
monitoring
DS8000 workload 168
monitoring tools
Disk Magic 176–177
Performance console 411–413
mount command 461–462
MPIO 297
MRU policy 431
multi-pathing 577
multipathing
DB2 UDB environment 577
SDD 407
Multiple Allegiance 9, 500, 504, 510
Multiple Allegiance (MA) 504
multiple reader 622
multi-rank 595, 617
multi-tier 81
multi-tier extent pool 84

N
N_Port ID Virtualization (NPIV) 428
Nearline-class drives 138
NL 138
nmon command 362
noatime 460
nocopy 591, 593–594
NOOP 458
Noop 457
Noop I/O Scheduler 457

O
OLDS 579
OLTP 560, 573, 575
open systems servers 169
characterize performance 395
dynamic buffer cache 383
filemon 357, 366
lvmstat 357, 360–361
performance logs 413
removing disk bottlenecks 420
topas 361–362
tuning I/O buffers 339
verify the storage subsystem 392
overhead 590, 593–594, 596

P
page cache 447
page cleaners 571
page size 570–571, 575
pages 570–572
Parallel 572
Parallel Access Volume (PAV) 510
Parallel Access Volumes see PAV
parallel operations 572
partitiongroups 569
partitioning map 569
path failover 319
PAV 9, 500–503, 522
PAV and Multiple Allegiance 504
pbuf 354
pdflush 447, 449
PEND time 504, 506, 524
performance 519, 585, 589, 595, 605, 608, 610
DB2 considerations 563
DB2 UDB considerations 567, 572
Disk Magic tool 176–177
FlashCopy overview 588–589
IMS considerations 578–579
planning for UNIX systems 328
size of cache 33
tuning Windows systems 397–398
Performance Accelerator feature 45

Performance console 411–413
performance data collection 253, 260, 275
performance data collection task 260
performance group 18
performance logs 413
performance monitor 260
performance numbers 4
performance policies 18
plan Address Groups and LSSs 140
plan RAID Arrays and Ranks 103
planning
  logical volume size 510
  UNIX servers for performance 328
planning and monitoring tools
  Disk Magic 176
  Disk Magic for open systems 198
  Disk Magic for zSeries 181
  Disk Magic modeling 177, 179
  Disk Magic output information 179
  workload growth projection 179
planning considerations 605
port group 74
ports 243, 261
posix AIO 355
POWER5 7–8
prefetch size 575
priority queuing 507
processor complex 26–27, 72, 130
processor memory 27, 34
pstat 355

Q
QoS 18, 22
Quality of Service 18, 22
quality of service 80
queue depth settings 452
QuickInit 66

R
RAID 39
RAID 10 595
  drive failure 54
  implementation 54
RAID 5 595
  drive failure 52
  implementation 52–53
RAID 6 53, 58
rank depopulation 90
rank group 63, 71–72
rank groups 62
ranks 59, 61, 65
RAS
  spare creation 56
Raw Device Mapping (RDM) 427, 437
RDM 427–428, 437
real-time sampling and display 390
recovery data sets 562
Red Hat 446, 461
Redbooks Web site 699
Redbooks website
  Contact us xx
ReiserFS 459, 461
repository 66
Resource Management Facility (RMF) 522
resource sharing 80
RIO-G loop 35
RMF 602, 609, 614
RMF Loader 182
RMF Magic 521, 609
RMFPack 182
RMZ 620
rotate extents 61, 68, 72, 79, 81, 84, 115, 125–126
rotate volumes 68, 79, 81, 126
Rotated volume 68
rotated volume 68

S
SAN 48
  cabling for availability 313
  zoning example 314
SAN Volume Controller (SVC) 536
SAR
  display previously captured data 391
  real-time sampling and display 390
  sar summary 392
sar command 389, 391, 474
sar summary 392
SARC 8
saturation 20
scalability 602, 606
SCSI
  See Small Computer System Interface
SCSI reservation 436–437
SDD 9, 297, 407, 452
  DB2 UDB environment 577
Sequential Prefetching in Adaptive Replacement Cache
see SARC
server affinity 61, 63, 71
setting I/O schedulers 457
SFF 108, 113
single path mode 318
single-rank extent pool 78–79, 118
single-tier extent pool 79, 84
Small Computer System Interface 450
small form factor 108, 113
Smart Rebuild 53
SMI-S standard
  IBM extensions 241
  ports 243
solid-state drive 7, 38, 138
solid-state drives 114
space efficient volumes 66
spares 39, 56
  floating 56
spatial ordering 32
spindle 595
spreading host attachments 48
SSD 7, 38, 114, 138
STAT 17, 85

static and dynamic PAVs 501
storage economics 89, 122
storage facility image 26
storage image 26
storage LPAR 26
Storage Optimizer 276
storage performance 89, 122
Storage Pool Striping 68, 592, 595, 617
Storage pool striping 68
storage pool striping 61–62, 68, 79, 84, 119, 125
storage server 72, 130
storage server challenge 4
storage servers
  add or remove in Global Mirror 619
Storage Tier Advisor Tool 17, 82, 85, 97, 123, 231
storage unit 26
striping
  DB2 UDB 574
  VSAM data striping 565
Subsystem Device Driver 452
Subsystem Device Driver (SDD) 452
supported Fibre Channel 312
supported servers
  FCP attachment 312
SuSE 446
SVC 297
switched fabric 313
symmetrical configuration 599
System Data Mover (SDM) 619
System Management Facilities (SMF) 178
System z servers
  concurrent write operation 506
  CONN time 522–523, 525
  DISC time 523–524, 528
  IOSQ time 503, 523
  PAV and Multiple Allegiance 504
  PEND time 504, 506, 524

T
table 561–562, 568
tables, indexes, and LOBs 570
tablespace 561, 568–571
  application 562
  system 562, 569
thin provisioned 66
thin provisioning 66
tiered storage 88
timestamps 250, 260
topas 361–362
topas command 361
topologies
  arbitrated loop 313
  direct connect 313
  switched fabric 313
tpctool CLI 272, 275
tpctool lstype command 273
track space efficient 13, 66
track space-efficient 587
TSE 13, 66, 587
tuning
  disk subsystem ??–456, ??–456
  Windows systems 397–398
tuning I/O buffers 339
tuning parameters 621

U
UNIX shell scripts
  introduction 660, 678

V
VDisk 536–537
Veritas Dynamic MultiPathing (DMP) 407
video on demand 165
Virtual Center (VC) 432
Virtual Machine File System (VMFS) 437
virtualization
  address groups 73
  array site 58
  array sites 58, 75
  arrays 58, 61
  extent pools 61, 65
  hierarchy 59, 75
  host attachment 74–75
  logical volumes 57, 59, 64
  ranks 59, 61, 65
  volume group 74–75
VMFS datastore 430, 436
VMotion 428–429
vmstat 353–355
vmstat command 470
VMware datastore 427
volume groups 74–75
volume heat distribution 85, 123
volume heat map 85, 123
volume migration 71
Volume migration 15
volumes 243, 247, 596, 598, 601
  add or remove 611, 614–615, 618
  CKD 65, 70, 75

W
WADS 579
Wavelength Division Multiplexing (WDM) 604
whitepapers 49
Windows
  tuning 397–398
Windows Server 2003
  fragmentation 401
workload 589, 592–593, 595
  databases 560
workload growth projection 179
workload isolation 80
Workload Manager 500, 507, 511
write ahead data sets, see WADS

X
XRC Performance Monitor 624
XRC see z/OS Global Mirror

xSeries servers
Linux 445–446

Z

z/ OS Workload Manager 22
z/OS
planning guidelines 519
z/OS Global Mirror 586, 619–620
dataset placement 621
IBM TotalStorage XRC Performance Monitor 624
tuning parameters 621
z/OS Global Mirror Multiple Reader 622
z/OS Workload Manager 500–501
zGM multiple reader 622
zSeries servers
overview 500
PAV 500–501, 503, 522
static and dynamic PAVs 501
zWLM 22

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 699. Most of the documents referenced here will be available in softcopy only:

- *IBM System Storage DS8000 Architecture and Implementation*, SG24-8886
- *IBM System Storage DS8000 Host Attachment and Interoperability,* SG24-8887
- *IBM System Storage Solutions Handbook*, SG24-5250
- *DS8000 Copy Services for IBM System z*, SG24-6787
- *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788
- *IBM System Storage DS8000 Easy Tier*, REDP-4667
- *DS8000 I/O Priority Manager*, REDP-4760
- *DS8000: Introducing Solid State Drives*, REDP-4522
- *DS8000 Performance Monitoring and Tuning*, SG24-7146
- *Tivoli Storage Productivity Center 4.2 Release Update*, SG24-7894
- *FICON Native Implementation and Reference Guide*, SG24-6266
- *Introduction to Storage Area Networks*, SG24-5470
- *Implementing the IBM System Storage SAN Volume Controller V6.1*, SG24-7933
- *SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521
- *SVC V4.3.0 Advanced Copy Services*, SG24-7574
- *SAN Storage Performance Management Using Tivoli Storage Productivity Center*, SG24-7364
- *Tuning IBM System x Servers for Performance*, SG24-5287
- *Tuning Linux OS on System p The POWER Of Innovation*, SG24-7338
- *Tuning Red Hat Enterprise Linux on IBM eServer xSeries Servers*, REDP-3861
- *Tuning SUSE LINUX Enterprise Server on IBM eServer xSeries Servers*, REDP-3862
- *Linux Performance and Tuning Guidelines*, REDP-4285
- *Tuning Windows Server 2003 on IBM System x Servers*, REDP-3943
- *Linux for IBM System z9 and IBM zSeries*, SG24-6694
- *Linux Handbook A Guide to IBM Linux Solutions and Resources,* SG24-7000
- *Linux on IBM System z: Performance Measurement and Tuning*, SG24-6926
- *Virtualizing an Infrastructure with System p and Linux*, SG24-7499
- *z/VM and Linux on IBM System z*, SG24-7492
- *Linux Performance and Tuning Guidelines*, REDP-4285

- *DB2 10 for z/OS Performance Topics*, SG24-7942
- *DB2 9 for z/OS Performance Topics*, SG24-7473
- *DB2 9 for z/OS: Backup and Recovery I/O Related Performance Considerations*, REDP-4452
- *Ready to Access DB2 for z/OS Data on Solid-State Drives*, REDP-4537
- *Effective zSeries Performance Monitoring Using Resource Measurement Facility*, SG24-6645
- *Database Performance Tuning on AIX*, SG24-5511
- *IBM i and IBM System Storage: A Guide to Implementing External Disks on IBM i*, SG24-7120
- *End to End Performance Management on IBM i*, SG24-7808
- *High Availability and Disaster Recovery Options for DB2 on Linux, UNIX, and Windows*, SG24-7363
- *AIX 5L Performance Tools Handbook*, SG24-6039
- *AIX 5L™ Practical Performance Tools and Tuning Guide*, SG24-6478
- *IBM AIX Version 7.1 Differences Guide*, SG24-7910
- *Best Practices for DB2 on AIX 6.1 for POWER Systems*, SG24-7821
- *Implementing IBM Storage Data Deduplication Solutions*, SG24-7888
- *IBM System Storage TS7650, TS7650G, and TS7610*, SG24-7652
- *TS7680 Deduplication ProtecTIER Gateway for System z*, SG24-7796
- *Performance Metrics in TotalStorage Productivity Center Performance Reports*, REDP-4347

Other publications

These publications are also relevant as further information sources:

- *IBM System Storage DS8800 and DS8700 Introduction and Planning Guide,* GC27-2297
- *IBM System Storage DS8000 Host Systems Attachment Guide,* GC27-2298
- *IBM System Storage DS Command-Line Interface User's Guide for the DS6000 series and DS8000 series*, GC53-1127
- *IBM System Storage DS Open Application Programming Interface Reference*, GC35-0516
- *IBM System Storage Multipath Subsystem Device Driver User's Guide,* GC52-1309
- *IBM System Storage DS8800 Performance Whitepaper*, WP102025
- *IBM System Storage DS8700 Performance Whitepaper*, WP101614
- *IBM System Storage DS8700 Performance with Easy Tier,* WP101675
- *IBM System Storage DS8700 and DS8800 Performance with Easy Tier 2nd Generation*, WP101961
- *IBM System Storage DS8800 and DS8700 Performance with Easy Tier 3rd Generation*, WP102024
- *IBM DS8000 Storage Virtualization Overview Including Storage Pool Striping, Thin Provisioning, Easy Tier,* WP101550

- *IBM Easy Tier on DS8000, SVC and Storwize V7000 Deployment Considerations Guide January 2011*, WP101844
- *IBM i Two-Tiered Hybrid Storage with IBM System Storage DS8700 Solid State Drives*, WP101868
- *Solid State Drive FLASHDA Analysis Case Study*, WP101948
- *Driving Business Value on Power Systems with Solid State Drives*, POW03025USEN
- *IBM i Shared Storage Performance using IBM System Storage DS8000 I/O Priority Manager*, WP101935
- *AIX disk queue depth tuning for performance*, TD105745
- *IBM DS8000 Performance Configuration Guidelines for Implementing Oracle Databases with ASM*, WP101375
- *Tuning SAP with Oracle on IBM AIX*, WP100377
- *Tuning SAP with DB2 on IBM AIX*, WP101601
- *Tuning SAP on DB2 for z/OS on System z*, WP100287
- *SAP BW on DB2 for z/OS System Check and Performance Recommendations*, WP101826
- *SAP on IBM i Solid State Disk (SSD) Usage Recommendations*, WP101795
- IBM i Information Center:

  http://publib.boulder.ibm.com/eserver/ibmi.html
- "Web Power – New browser-based Job Watcher tasks help manage your IBM i performance" in the IBM Systems Magazine:

  http://www.ibmsystemsmag.com/ibmi

Online resources

These websites and URLs are also relevant as further information sources:

- IBM Disk Storage Feature Activation (DSFA)

  http://www.ibm.com/storage/dsfa
- Documentation for the DS8000

  http://www.ibm.com/systems/storage/disk/ds8000/index.html
- IBM System Storage Interoperation Center (SSIC)

  http://www.ibm.com/systems/support/storage/config/ssic/index.jsp
- IBM Announcement letters (for example, search for R6.2)

  http://www.ibm.com/common/ssi/index.wss
- IBM Techdocs Library - The IBM Technical Sales Library

  http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy IBM Redbooks publications or CD-ROMs, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

DS8800 Performance Monitoring and Tuning

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

DS8800 Performance Monitoring and Tuning

Understand the performance and features of the DS8800 architecture

Configure the DS8800 to fully exploit its capabilities

Use planning and monitoring tools with the DS8800

This IBM Redbooks publication provides guidance about how to configure, monitor, and manage your IBM System Storage DS8800 and DS8700 storage systems to achieve optimum performance. It describes the DS8800 and DS8700 performance features and characteristics, including IBM System Storage Easy Tier and DS8000 I/O Priority Manager. It also describes how they can be used with the various server platforms that attach to the storage system. Then, in separate chapters, we detail specific performance recommendations and discussions that apply for each server environment, as well as for database and DS8000 Copy Services environments.

We also outline the various tools available for monitoring and measuring I/O performance for different server environments, as well as describe how to monitor the performance of the entire DS8000 storage system.

This book is intended for individuals who want to maximize the performance of their DS8800 and DS8700 storage systems and investigate the planning and monitoring tools that are available.

The IBM System Storage DS8800 and DS8700 storage system features, as described in this book, are available for the DS8700 with Licensed Machine Code (LMC) level 6.6.2x.xxx or higher and the DS8800 with Licensed Machine Code (LMC) level 7.6.2x.xxx or higher.

For information about optimizing performance with the previous DS8000 models, DS8100 and DS8300, see the following IBM Redbooks publication: *DS8000 Performance Monitoring and Tuning*, SG24-7146.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8013-00 ISBN 0738436836