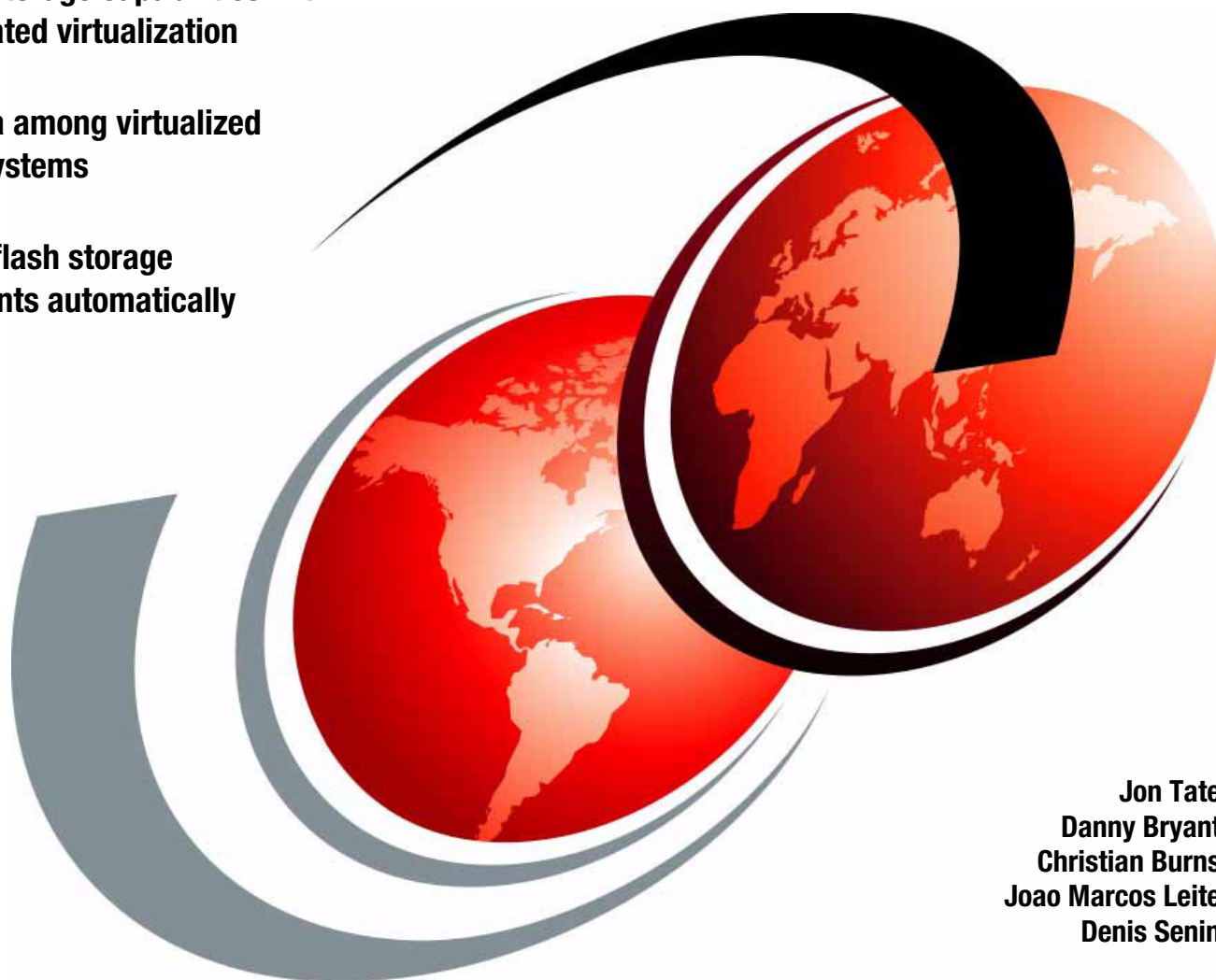


Implementing the IBM SAN Volume Controller and FlashSystem 820

Enhance storage capabilities with sophisticated virtualization

Move data among virtualized storage systems

Optimize flash storage deployments automatically



Jon Tate
Danny Bryant
Christian Burns
Joao Marcos Leite
Denis Senin

Redbooks



International Technical Support Organization

**Implementing the IBM SAN Volume Controller and
FlashSystem 820**

September 2013

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

First Edition (September 2013)

This edition applies to the hardware and software listed in Appendix B, “Example environment details” on page 139.

© Copyright International Business Machines Corporation 2013. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
 Preface	 ix
Authors	x
Now you can become a published author, too!	xii
Comments welcome	xii
Stay connected to IBM Redbooks	xii
 Chapter 1. Introduction to IBM SAN Volume Controller and IBM FlashSystem 820 . . .	 1
1.1 IBM System Storage SAN Volume Controller	2
1.1.1 IBM System Storage SAN Volume Controller overview	2
1.1.2 IBM System Storage SAN Volume Controller design overview	2
1.1.3 SAN Volume Controller architecture and components	5
1.1.4 IBM SmartCloud Virtual Storage Center	7
1.2 SAN Volume Controller hardware and software updates of interest to this book	8
1.2.1 Hardware updates for SVC	8
1.2.2 New features in SVC Storage Software version 7.1	9
1.3 Introduction to IBM FlashSystem storage systems	10
1.3.1 IBM FlashSystem storage system portfolio	11
1.3.2 Differences between IBM FlashSystem families and models	12
1.4 Introduction to flash solid-state technology	14
1.4.1 Types of flash memory used in IBM FlashSystems	15
1.4.2 Unique characteristics	16
1.4.3 Single-level cell memory	16
1.4.4 Multi-level cell memory	17
1.4.5 Solid-state drive architecture	18
1.4.6 Flash memory lifetime	21
1.5 IBM FlashSystem storage systems technology overview	22
1.5.1 IBM FlashSystem storage systems architecture	24
1.5.2 Data protection and redundancy in the IBM FlashSystem 820	32
1.5.3 RAID technologies	32
1.5.4 Flash memory protection	35
1.5.5 RAID rebuild process	35
1.5.6 Differences between IBM FlashSystem storage systems and SSD-based storage systems	36
1.5.7 Usage considerations for IBM FlashSystem storage systems	38
 Chapter 2. Usage considerations and scenarios	 41
2.1 Usage considerations	42
2.1.1 Port assignment scenarios and related considerations	42
2.1.2 SAN Volume Controller Stretched Cluster	46
2.1.3 Port masking on SVC with IBM FlashSystem 820	46
2.1.4 Host multipathing	47
2.1.5 Considerations regarding number of FlashSystem per I/O group	47
2.2 Usage scenarios	48
2.2.1 All FlashSystem usage scenario	48
2.2.2 FlashSystem with SAN Volume Controller volume mirroring	51
2.3 SAN Volume Controller volume mirroring use cases with FlashSystem	57

2.3.1	Volume mirroring between two FlashSystem storage systems.	57
2.3.2	Volume mirroring between a FlashSystem storage system and a non-flash storage system	58
2.4	Using FlashSystem with SVC Easy Tier	58
2.4.1	General considerations with Easy Tier	58
2.4.2	Separate FlashSystem storage pools and existing storage pools that are not using IBM Easy Tier	59
2.5	FlashSystem with SVC replication	60
2.6	Failure protection capabilities	61
2.6.1	FlashSystem 820 hardware failure protection.	61
2.6.2	SVC hardware failure protection	62
2.6.3	SAN fabric failure considerations	62
Chapter 3.	Planning and installation of the IBM FlashSystem 820.	63
3.1	Physical device accessibility	64
3.1.1	Physical layout	64
3.1.2	Rack placement and cabling considerations.	64
3.2	Physical cabling	66
3.2.1	FC port speed settings	66
3.2.2	Cabling methods	66
3.2.3	FC cable type	67
3.2.4	Ethernet management cabling	67
3.3	Power and cooling considerations	68
3.3.1	Power requirements	68
3.3.2	Cooling requirements	68
3.4	Performance Guidelines	69
3.4.1	VMware VAAI	69
3.4.2	Performance Redpaper.	69
3.4.3	Performance data and statistics collection	70
Chapter 4.	Configuration and administration	77
4.1	Setup and configuration of IBM FlashSystem for use with SAN Volume Controller	78
4.1.1	Configuring the FlashSystem management IP addresses	78
4.1.2	FlashSystem management tools.	80
4.1.3	IBM FlashSystem feature licenses	83
4.1.4	Configuring additional network settings	85
4.1.5	email notifications and call home	90
4.1.6	Fibre Channel port settings.	94
4.1.7	Logical unit creation and access policies	99
4.2	Port masking and SAN zoning configuration.	108
4.2.1	SAN Volume Controller Fibre Channel port masking setup	108
4.2.2	Host Fibre Channel port masking setup	110
4.2.3	SAN zoning setup	110
4.3	SAN Volume Controller MDisk configuration.	111
4.3.1	MDisk LUN configuration on IBM FlashSystem 820	111
4.3.2	Storage pool configuration guidelines.	113
4.3.3	Quorum disk allocation	113
4.3.4	Storage pool extent size	116
4.3.5	MDisk mapping and storage pool creation using the CLI	117
4.3.6	MDisk mapping and storage pool creation using the GUI.	118
Chapter 5.	Diagnostics, planned outages, and troubleshooting	123
5.1	I/O group volume migration for planned outages	124
5.2	FlashSystem firmware update non-disruptively with the use of volume mirroring.	124

5.3	Volume migration to another storage pool for planned outages	125
5.4	Call home features of SAN Volume Controller and FlashSystem storage systems . .	126
5.4.1	SVC call home	126
5.4.2	IBM FlashSystem call home and event notification.	127
5.5	Easy Tier and FlashSystem planned outages.	127
5.6	Hardware replacement guide for IBM FlashSystem storage systems.	128
Appendix A. FlashSystem CLI commands used in the example environment		129
	Configuring the example IBM FlashSystem storage systems using the CLI	130
	Setting the management control processor host name.	130
	Setting the Domain Name Service domain	130
	Configuring the call home feature	131
	Configuring the events notification feature	131
	Configuring Fibre Channel port settings using the CLI	132
	Configuring logical unit numbers and access policies using the CLI	134
Appendix B. Example environment details.		139
	Example environment hardware components	140
	List of hardware components	140
	Example environment firmware and software levels	141
	Example environment topology	141
Related publications		143
	IBM Redbooks	143
	Other publications	143
	Online resources	144
	Help from IBM	144

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 grant 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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:

Calibrated Vectors Cooling™	IBM SmartCloud®	Smarter Planet®
Cognos®	IBM®	Storwize®
Easy Tier®	Intelligent Cluster™	System Storage®
FlashCopy®	Real-time Compression™	System x®
FlashSystem™	Redbooks®	Tivoli®
Global Technology Services®	Redpaper™	XIV®
IBM FlashSystem™	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

VSR, and the Texas Memory Systems logo are trademarks or registered trademarks of Texas Memory Systems, an IBM Company.

Intel Xeon, 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.

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

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Preface

In today's 24 x 7 world, there is likely not a business on this planet, IBM® Smarter Planet® or not, that finds that their storage requirements are growing too fast and demand is starting to outpace supply. Not only this, but in this cost-conscious environment of today, the costs of managing this growth are likely to be eating into the IT budget.

One way to make better use of existing storage without adding more complexity to the infrastructure is the IBM System Storage® SAN Volume Controller (SVC). For many years now this has helped business become more flexible, agile, and introduced an extremely efficient storage environment. SAN Volume Controller is designed to deliver the benefits of storage virtualization in environments from large enterprises to small businesses and midmarket companies.

Virtualizing storage with SAN Volume Controller helps make new and existing storage more effective. SAN Volume Controller includes many functions that are traditionally deployed separately in disk systems. By including these in a virtualization system, SAN Volume Controller standardizes functions across virtualized storage for greater flexibility and potentially lower costs.

Now, with IBM FlashSystem™ storage, SAN Volume Controller is enabled to extend its reach and benefit all virtualized storage. For example, IBM Easy Tier® optimizes use of flash storage. And IBM Real-time Compression™ enhances efficiency even further by enabling the storage of up to five times as much active primary data in the same physical disk space.

In this IBM Redbooks® publication, we show how to integrate the IBM FlashSystem 820 to provide storage to the SAN Volume Controller, and show how they are designed to operate seamlessly together, reducing management effort.

In this book, which is aimed at pre- and post-sales support, storage administrators, and people that want to get an overview of this new and exciting technology, we show the steps required to implement the IBM FlashSystem 820 in an existing SAN Volume Controller environment. We also highlight some of the new features in SAN Volume Controller that increase performance.

If you are not already familiar with the SAN Volume Controller, it is beneficial to read the following IBM Redbooks publications:

- ▶ *Implementing the IBM System Storage SAN Volume Controller V6.3, SG24-7933*
- ▶ *Implementing the IBM Storwize V7000 V6.3, SG24-7938*
- ▶ *Real-time Compression in SAN Volume Controller and Storwize V7000, REDP-4859*
- ▶ *IBM SAN Volume Controller and IBM FlashSystem 820: Best Practices and Performance Capabilities, REDP-5027*
- ▶ *IBM FlashSystem 710 and IBM FlashSystem 810, TIPS1002*
- ▶ *IBM FlashSystem 720 and IBM FlashSystem 820, TIPS1003*
- ▶ *Flash or SSD: Why and When to Use IBM FlashSystem, REDP-5020*

Authors

This book was produced by a team of specialists from around the world working at the home of virtualization, IBM Hursley Labs, UK.



Jon Tate is a Project Manager for IBM System Storage SAN Solutions at the International Technical Support Organization (ITSO), San Jose Center. Before joining the ITSO in 1999, he worked in the IBM Technical Support Center, providing Level 2/3 support for IBM storage products. Jon has over 27 years of experience in storage software and management, services, and support, and is both an IBM Certified Consulting IT Specialist and an IBM SAN Certified Specialist. He is also the UK Chairman of the Storage Networking Industry Association.



Danny Bryant is a Client Technical Specialist (CTS) for IBM System Storage for Systems Technology Group (STG) in Melbourne, Australia. Before joining STG, he worked in IBM Global Technology Services® (GTS) as a design and implementation specialist for IBM System Storage within enterprise accounts. Danny has 10 years of experience in the IT industry and a strong infrastructure background. He also specializes in VMware.



Christian Burns is an IBM Storage Solution Architect based in New Jersey. As a member of the Storage Solutions Engineering team in Littleton, MA, he works with clients, IBM Business Partners, and IBMers worldwide, designing and implementing storage solutions that include various IBM products and technologies. Christian's areas of expertise include IBM Real-time Compression, SVC, XIV®, and IBM FlashSystem. Before joining IBM, Christian was the Director of Sales Engineering at IBM Storwize®, prior to its becoming IBM Storwize. He brings over a decade of industry experience in the areas of sales engineering, solution design, and software development. Christian holds a BA degree in Physics and Computer Science from Rutgers College.



Joao Marcos Leite is an IT Specialist who joined IBM Brazil in 2000. He has worked for the IBM Systems and Technology Group in the field of storage solutions design for clients, and is currently working as an Advanced Technical Skills (ATS) member for the IBM Growth Markets Unit (GMU) in Latin America, focused on Storage Software. His areas of expertise include storage virtualization and storage management, and he co-authored previous updates to the SAN Volume Controller Implementation, and to the IBM Tivoli® Storage Productivity Center V4.2 Release Guide Redbooks publications. Joao graduated as a Data Processing Technologist from the Universidade Federal do Parana in Curitiba, Brazil. He has 32 years of experience as an IT Specialist and is a member of the Technology Leadership Council Brazil (TLC-BR), an affiliate of the IBM Academy of Technology. Joao is a Thought Leader (Level 3) Certified IT Specialist by IBM and holds a title of Distinguished IT Specialist by The Open Group.



Denis Senin is an ATS CEE Storage Specialist in IBM Russia. He has 10 years of experience in IT industry and has worked at IBM for 8 years. Denis holds a Master's degree of design-engineer of computer systems from The Moscow State Institute of Radiotechnics, Electronics and Automatics, and has a background of systems design and development. His current areas of expertise include Open Systems, high-performance, disaster recovery, and flash memory-based storage solutions.

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

Ian Boden
Carlos Fuente
Evelyn Perez
Greg Shepherd
Vairavan Sockalingam
Barry Whyte
IBM Hursley

Matt Key
Adrian Flores-serafin
Kevin Powell
Bobby Sumners
Brad Duncan
David Drinnan
Travis Dockery
IBM US

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 that is 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>



Introduction to IBM SAN Volume Controller and IBM FlashSystem 820

This chapter provides an introduction to IBM FlashSystem storage systems. It describes the primary concepts concerning flash memory technology, including design architecture, internal processes, and the advantages and limitations of flash. The architecture of the IBM FlashSystem 820 is covered and we provide a brief introduction to the IBM SAN Volume Controller, including an overview of the latest software and hardware updates.

The following topics are covered:

- ▶ Introduction to the IBM SAN Volume Controller
- ▶ Introduction to the IBM FlashSystem storage system
- ▶ Architecture and internal processes of the IBM FlashSystem 820
- ▶ Introduction to flash memory technology
- ▶ Usage considerations of traditional and flash-based storage systems
- ▶ Unique benefits of the IBM FlashSystem storage system implementation

1.1 IBM System Storage SAN Volume Controller

The IBM System Storage SAN Volume Controller (SVC) is a storage virtualization solution that helps to increase the utilization of existing storage capacity and centralize the management of multiple controllers in an open-system storage area network (SAN) environment.

The SAN Volume Controller supports attachment to both IBM and non-IBM storage systems. It enables storage administrators to reallocate and scale storage capacity and make changes to underlying storage systems without disruption to applications.

In this section, we provide an overview of the SVC. Details about some of the new features and capabilities that were introduced with SAN Volume Controller 7.1 are described in section 1.2, “SAN Volume Controller hardware and software updates of interest to this book” on page 8.

1.1.1 IBM System Storage SAN Volume Controller overview

The IBM System Storage SAN Volume Controller, machine type 2145, and accompanying software, provides the ability to simplify storage infrastructure, utilize storage resources more efficiently, improve personnel productivity, and increase application availability.

SAN Volume Controller pools storage volumes from IBM and non-IBM disk arrays into a single reservoir of capacity, which can be managed from a central point. SAN Volume Controller also allows data to be migrated between heterogeneous disk arrays without disruption to applications. By moving copy services functionality into the network, SVC allows you to use a standardized suite of copy services tools that can be applied across the entire storage infrastructure, irrespective of storage vendor restrictions that normally apply for the individual disk controllers in use.

Additionally, SAN Volume Controller adds functions to the infrastructure that might not be present in each virtualized subsystem. Examples include thin provisioning, automated tiering, volume mirroring, and data compression.

Note: Some of the SAN Volume Controller functions mentioned above are included in the base virtualization license, although for others an additional license might need to be purchased.

1.1.2 IBM System Storage SAN Volume Controller design overview

The IBM System Storage SAN Volume Controller is designed to handle the following tasks:

- ▶ Combine storage capacity from multiple vendors into a single repository of capacity with a central management point
- ▶ Help increase storage utilization by providing host applications with more flexible access to capacity
- ▶ Help improve productivity of storage administrators by enabling management of combined storage volumes from a single, user-friendly interface
- ▶ Support improved application availability by insulating host applications from changes to the physical storage infrastructure

- Enable a tiered storage environment, in which the cost of storage can be better matched to the value of data
- Support advanced copy services, from higher-cost to lower-cost devices and across subsystems from multiple vendors

SAN Volume Controller combines hardware and software into a comprehensive, modular appliance. Using IBM System x® server technology in highly reliable clustered pairs, the SAN Volume Controller has no single points of failure. The SAN Volume Controller software is a highly available cluster that is optimized for performance and ease of use.

Storage utilization

The SAN Volume Controller is designed to help increase the amount of storage capacity that is available to host applications. By pooling the capacity from multiple disk arrays within the SAN, it enables host applications to access capacity beyond their island of SAN storage. The Storage Networking Industry Association (SNIA) estimates that open systems disk utilization in a non-virtualized environment is currently only between 30% - 50%. With storage virtualization, this utilization can grow up to 80%, on average.

Scalability

A SAN Volume Controller configuration can start with a single I/O group. An I/O group is a pair of high performance, redundant Intel processor-based servers, referred to as *nodes* or *storage engines*. Highly available I/O groups are the basic configuration of a cluster. Adding additional I/O groups can help increase cluster performance and bandwidth.

SAN Volume Controller can scale out to support up to four I/O groups. SAN Volume Controller Version 7.1 supports up to 2048 host servers when using CF8 and CG8 engines. For every cluster, the SVC supports up to 8192 volumes, each one with up to 256 TB in size, and a total virtualized capacity up to 32 PB.

Note: For the most up-to-date SVC configuration limits, refer to the Configuration Limits and Restrictions website for the latest SAN Volume Controller version:

<http://www.ibm.com/support/docview.wss?uid=ssg1S1003658>

This configuration flexibility means that SAN Volume Controller configurations can start small with an attractive price to suit smaller environments or pilot projects, and then can grow with your business to manage very large storage environments.

Management

The SAN Volume Controller is managed at the cluster level, providing a single point of control over all the managed storage. The SAN Volume Controller provides a comprehensive, easy-to-use graphical user interface (GUI) for central management. This simple interface incorporates the Storage Management Initiative Specification (SMI-S) application programming interface (API), and further demonstrates the commitment of IBM to open standards. The SAN Volume Controller cluster can also be managed and monitored through a comprehensive command-line interface (CLI) via Secure Shell (SSH), enabling the use of scripts and automate repeatable operations.

The SAN Volume Controller GUI is designed for ease of use and includes many built-in IBM guidelines that simplify storage provisioning and enables new clients to get started quickly with a rapid learning curve.

Clients using IBM Tivoli Storage Productivity Center, IBM Systems Director, and IBM Tivoli Storage FlashCopy® Manager can take further advantage of integration points with the SVC.

As with managing the SVC under Tivoli Storage Productivity Center, IBM Systems Director will be enabled to perform the most common day-to-day activities for SVC without ever needing to leave the IBM Systems Director user interface.

For historic performance and capacity management from both the host and the virtualized storage devices' perspectives, Tivoli Storage Productivity Center helps clients with an end-to-end view and control of the virtualized storage infrastructure. Regarding data protection, Tivoli Storage FlashCopy Manager helps integrate the SVC FlashCopy function with major applications for consistent backups and restores.

Linking infrastructure performance to business goals

By pooling storage into a single reservoir, the SAN Volume Controller helps insulate host applications from physical changes to the storage pool, minimizing disruption. The SVC simplifies storage infrastructure by including a dynamic data-migration function, allowing for online volume migration from one device to another. By using this function, administrators can reallocate, scale storage capacity, and apply maintenance to storage subsystems without disrupting applications, increasing application availability.

With the SVC, your business can build an infrastructure from existing assets that is simpler to manage, easier to provision, and can be changed without impact to application availability. Businesses can use their assets more efficiently and actually measure the improvements. They can allocate and provision storage to applications from a single view and know the effect on their overall capacity instantaneously. They can also quantify improvements in their application availability to enable better quality of service goals. These benefits help businesses manage their costs and capabilities more closely, linking the performance of their infrastructure to their individual business goals.

Tiered storage

In most IT environments, inactive data makes up the bulk of stored data. The SAN Volume Controller helps administrators control storage growth more effectively by moving low-activity or inactive data into a hierarchy of lower-cost storage. Administrators can free disk space on higher-value storage for more important, active data. It is achieved by easily creating various groups of storage, or *storage pools*, corresponding to underlying storage with various characteristics. Examples are speed and reliability. With the SAN Volume Controller, you can better match the cost of the storage used to the value of data placed on it.

Copy services

With many conventional SAN disk arrays, copy services can only be performed within the array, or between identical arrays. The SAN Volume Controller enables administrators to apply a single set of advanced copy services, such as IBM FlashCopy, Metro Mirror, and Global Mirror services, across multiple storage subsystems from various vendors.

A volume can also be mirrored locally between two different storage subsystems for high availability. This volume mirroring function is the basis for stretched cluster configurations where the I/O group nodes are spread over two different locations to build an even more resilient solution.

Technology for an on demand environment

Businesses are facing growth in critical application data that is supported by complex heterogeneous storage environments, while their staffs are overburdened. The SAN Volume Controller is one of many offerings in the IBM System Storage portfolio that is essential for an on demand storage environment. These offerings can help you simplify your IT infrastructure, manage information throughout its lifecycle, and maintain business continuity.

1.1.3 SAN Volume Controller architecture and components

SAN-based storage is managed by the SAN Volume Controller in one or more *I/O groups (pairs)* of SVC hardware nodes, referred to as a *clustered system*. These nodes are attached to the SAN fabric, along with storage controllers and host systems. The SAN fabric is zoned to allow the SVC to “see” the storage controllers, and for the hosts to “see” the SVC.

The hosts are not allowed to “see” or operate on the same physical storage (logical unit number (LUNs)) from the storage controllers that have been assigned to the SVC, and all data transfer happens through the SVC nodes. This design is commonly described as *symmetric virtualization*.

Storage controllers can be shared between the SVC and direct host access as long as the same LUNs are not shared, and both types of access use compatible multi-pathing drives in the same host or operating system instance. The zoning capabilities of the SAN switch must be used to create distinct zones to ensure that this rule is enforced.

Figure 1-1 shows a conceptual diagram of a storage environment using the SVC. It shows a number of hosts that are connected to a SAN fabric or LAN along with the SVC storage nodes and the storage subsystems that provide capacity to be virtualized. In practical implementations that have high-availability requirements (most of the target clients for SVC), the SAN fabric “cloud” represents a redundant SAN. A redundant SAN consists of a fault-tolerant arrangement of two or more counterpart SANs, providing alternate paths for each SAN-attached device. The SVC can be connected to up to four fabrics.

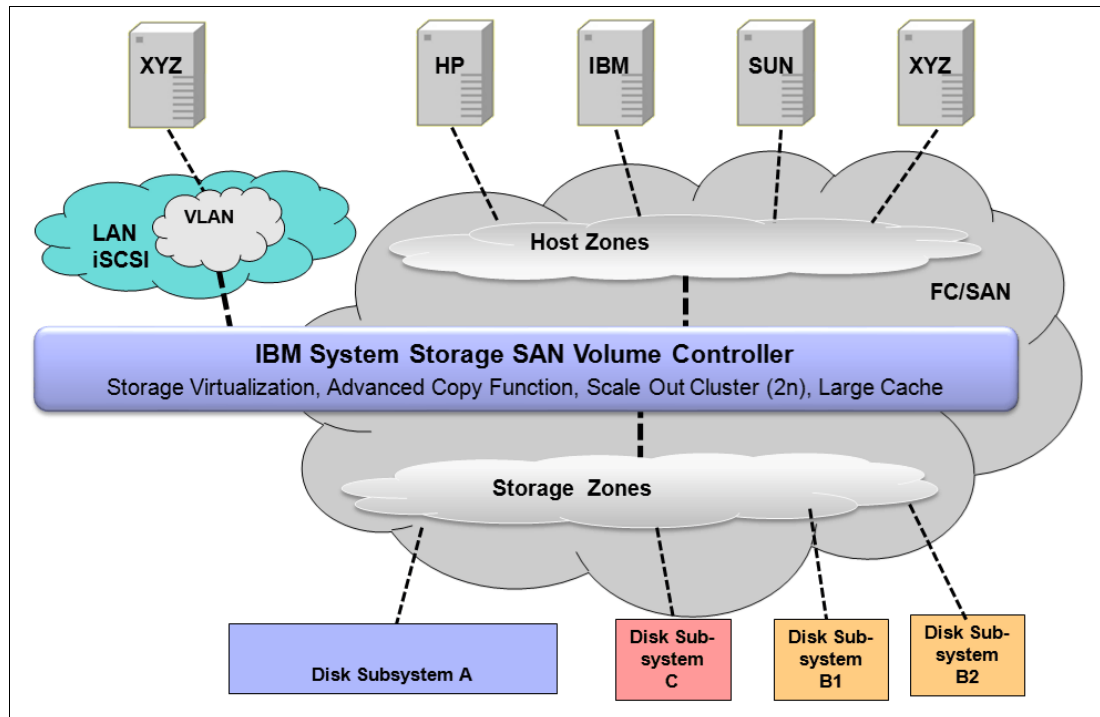


Figure 1-1 Conceptual diagram of SVC and the SAN infrastructure

Both scenarios (using a single network and using two physically separate networks) are supported for Internet Small Computer System Interface (iSCSI)-based and LAN-based access networks to the SVC. Redundant paths to volumes can be provided in both scenarios. For iSCSI-based access, using two networks and separating iSCSI traffic within the networks by using a dedicated virtual local area network (VLAN) for storage traffic prevents any IP

interface, switch, or target port failure from compromising the host servers' access to the volumes.

A *clustered system* of SVC nodes that are connected to the same fabric presents logical disks, or *volumes* to the hosts. These volumes are created from managed LUNs or *managed disks* (MDisks) that are presented to SVC by the storage subsystems and grouped in *storage pools*. Two distinct zones shown in the fabric:

- ▶ A host zone, in which the hosts can see and address the SVC nodes and access volumes
- ▶ A storage zone, in which the SVC nodes can see and address the MDisks/logical units (LUNs) that are presented by the storage subsystems

Figure 1-2 shows the logical architecture of SVC, illustrating how different storage pools are built grouping MDisks, and how the volumes are created from those storage pools and presented to the hosts through I/O groups (pairs of SVC nodes). In this diagram, Vol2, Vol7, and Vol8 are mirrored volumes, or volumes with two copies, with each copy residing in a different storage pool.

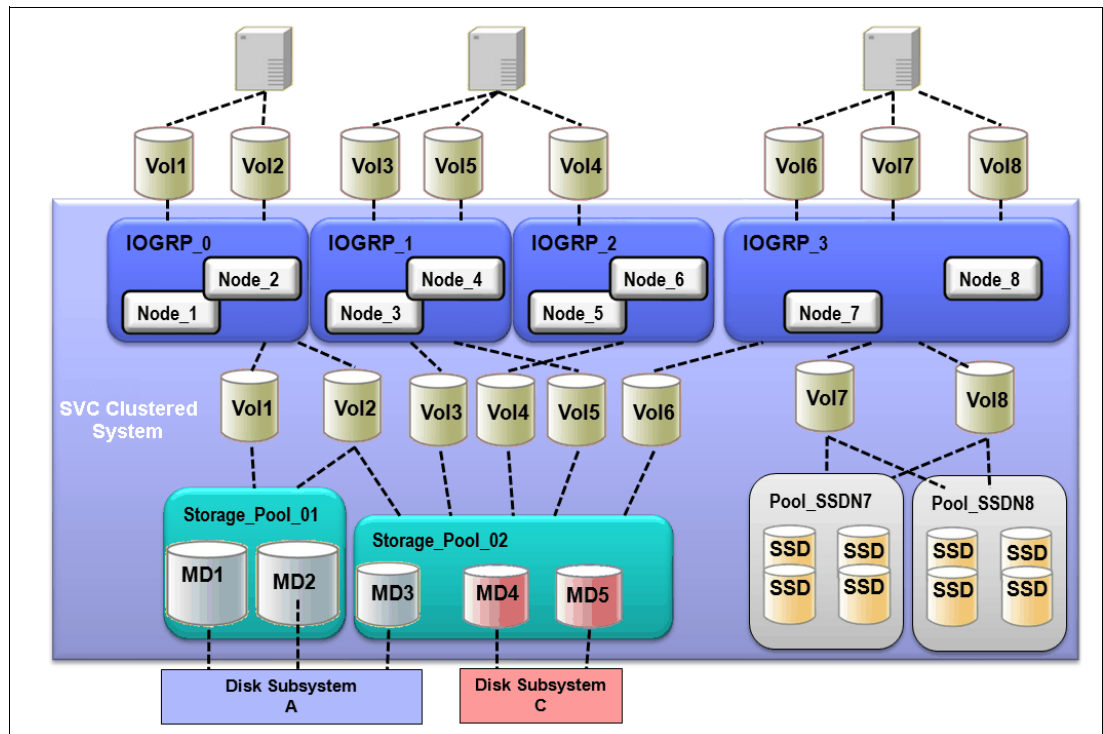


Figure 1-2 Overview of SVC clustered system with relations to the hosts and storage subsystems

Each MDisk in the storage pool is divided into a number of *extents*. The size of the extent is selected by the administrator at the creation time of the storage pool and cannot be changed later. The size of the extent ranges from 16 MB up to 8192 MB. Figure 1-3 on page 7 gives an outline on how a volume is built using the extents that come from a storage pool, depending on the volume type (striped or sequential).

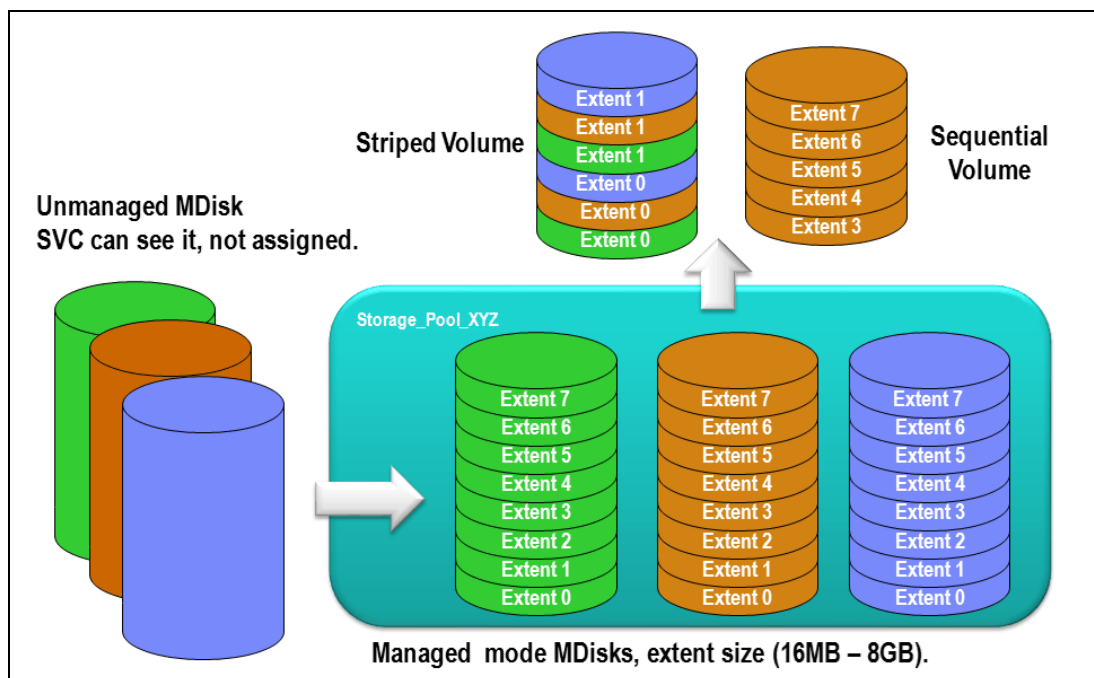


Figure 1-3 Volume creation from storage pools

1.1.4 IBM SmartCloud Virtual Storage Center

IBM SmartCloud® Virtual Storage Center (VSC) can help you to achieve enhanced storage efficiency, greater mobility, and stronger control over storage performance and management. The following capabilities are key value enhancers:

- ▶ Virtualized physical storage resources for improved asset utilization
- ▶ Easy data mobility across arrays and tiers
- ▶ Centralized management that offers visibility, control, and automation

SmartCloud Virtual Storage Center helps accelerate time-to-value and reduce significant total cost of ownership by providing:

- ▶ Insights that offer advanced topology views, metrics for storage configurations, performance, tiered capacity, and customizable IBM Cognos® based reporting
- ▶ Best practice recommendations that offer guidance for configuration, provisioning, and SAN planning to help ensure optimal setup
- ▶ Monitoring and reporting that offer performance heat maps, and threshold and fault alerting for high availability
- ▶ Storage optimization that offers guidance for optimal configuration and enhanced utilization, and data migration recommendations through tiered storage capacity optimization tools

SmartCloud Virtual Storage Center offers both a storage virtualization platform and capabilities for storage virtualization management. SmartCloud VSC V5.1 delivers to clients, under one licensed software product, the complete set of advanced functions available in IBM Tivoli Storage Productivity Center, all the functions available with the virtualization, remote-mirroring, FlashCopy capabilities of IBM System Storage SVC, and all the capabilities of IBM Tivoli Storage FlashCopy Manager.

This way, SmartCloud VSC is an excellent alternative to have storage virtualization with SVC nodes, complemented by a management platform and data protection, all under a single license. SmartCloud Virtual Storage Center is also referred to as a *Storage Hypervisor* because of its combination of a virtualization platform and a management platform.

1.2 SAN Volume Controller hardware and software updates of interest to this book

Throughout its lifecycle, SAN Volume Controller has leveraged IBM System x server technology to offer a modular, flexible platform for storage virtualization that can be rapidly adapted in response to changing market demands and evolving client requirements. This “flexible hardware” design allows us to quickly incorporate differentiating features that allow our clients to succeed. As of this writing, there are two significant hardware updates available for the SVC that are of interest:

- ▶ Feature code AHA7, offering an additional 4-port 8 Gb host bus adapter (HBA)
- ▶ 2145 - SVC Compression Accelerator, RPQ 8S1296, offering an additional 6-core CPU and an additional 24 GB of memory

Additionally, SVC Storage Software version 7.1 offers several feature updates of interest to us in this book:

- ▶ Second Fibre Channel HBA support
- ▶ Port masking
- ▶ Support for Easy Tier with compressed volumes

1.2.1 Hardware updates for SVC

There are two new hardware updates that we mention in this book.

Additional 4-port 8 Gb HBA

This feature adds an additional 4-port 8 Gbps Fibre Channel HBA to improve connectivity options on the SAN Volume Controller engine. The SVC engine comes standard with a 4-port 8 Gbps Fibre Channel HBA. This feature adds a second HBA.

The following example scenarios describe where these additional ports can provide benefits:

- ▶ Isolation of node-to-node communication, potentially boosting write performance
- ▶ Isolation of node to IBM FlashSystem communication, allowing for maximum performance

For more information about practical usage scenarios for this additional card, see 2.1, “Usage considerations” on page 42.

This feature code is only supported on 2145-CG8 nodes (both 4-core or 6-core CPU) and requires SVC Storage Software version 7.1 or higher. For more information, refer to *IBM System Storage SAN Volume Controller Storage Engine Fibre Channel host bus adapter feature*, *IBM United States Hardware Announcement 113-096*:

<http://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS113-096&appname=USN>

Additional CPU and memory

This request for price quotation (RPQ) allows you to add a second, 6-core CPU and an additional 24 GB of memory to a CG8 engine. The standard CG8 engine has a single, 6-core

CPU (type E5645) and 24 GB of memory.¹ As discussed in section 4.3.1, “MDisk LUN configuration on IBM FlashSystem 820” on page 111, four of the six standard CPU cores are used for general system processing, while the remaining two are reserved for use by IBM Real-time Compression (RtC).

When RtC is used on an SVC I/O group, the system surrenders an additional two cores per CPU and 2 GB of memory from each node in the I/O group for use by Real-time Compression.

This effectively reduces the amount of system CPU resources that are available. To avoid the potential for any impact that this may have, this RPQ has been designed to address this by:

- ▶ Closely preserving the amount of resources that are available for general system processing, regardless of whether Real-time Compression is in use or not
- ▶ Increasing the amount of resources available for Real-time Compression

Table 1-1 details the resource allocation differences for CG8 nodes with and without this RPQ, with Real-time Compression in use and not in use.

Table 1-1 Resource allocation for CG8 nodes with and without RPQ 8S1296

	Standard 2145-CG8 Node		2145-CG8 Node with RPQ 8S1296	
	No RtC	RtC in Use	No RtC	RtC in Use
System CPU Cores	4	2	4	4
System Memory	24 GB	22 GB	24 GB	22 GB
RtC CPU Cores	2	4	8	8
RtC Memory	none	2 GB	none	26 GB

For more information about this RPQ, contact your local IBM Storage Sales Specialist.

To submit a Solution for Compliance in a Regulated Environment (SCORE) request for this RPQ, see the following site:

<http://iprod.tucson.ibm.com/systems/support/storage/ssic/interoperability.wss>

1.2.2 New features in SVC Storage Software version 7.1

Three new software updates are mentioned in this book.

Second Fibre Channel HBA support

This new software feature adds support for the additional 4-port 8 Gbps Fibre Channel HBA that is available in feature code AHA7. SVC Storage Software version 7.1 or higher is required to use this new hardware feature code. The software includes GUI and CLI support for configuring and managing this additional card.

Port masking

The addition of more Fibre Channel HBA ports that are introduced with feature code AHA7 allow clients to optimize their SVC configuration by using dedicated ports for certain system functions. However, the addition of these ports necessitates the ability to ensure traffic isolation.

¹ Early CG8 nodes shipped with a single quad-core CPU (type E5630). RPQ 8S1296 is only available for six-core CG8 nodes. You can use the `lsnodevpd` command to determine the type of processor in your node.

Following are two examples of traffic types that you might want to isolate using port masking:

- ▶ Local node-to-node communication
- ▶ Replication traffic

For more information about port masking, see section 2.1, “Usage considerations” on page 42, and 4.2, “Port masking and SAN zoning configuration” on page 108.

Support for Easy Tier with compressed volumes

Easy Tier is a no-charge performance optimization function that automatically migrates “hot” extents belonging to a volume to MDisks that better meet the performance requirements of that extent. The Easy Tier function can be turned on or off at the storage pool level and at the volume.

Real-time Compression is a feature of SVC that addresses all the requirements of primary storage data reduction, including performance, using a purpose-built compression technology, allowing for data reduction of up to 80%.

In practice, clients have found that their target workloads for these two features have a significant overlap. Before SVC Storage Software version 7.1, the use of these two features was mutually exclusive at the volume level. In version 7.1, the concurrent use of Easy Tier and Real-time Compression is supported on the same volume.

1.3 Introduction to IBM FlashSystem storage systems

IBM FlashSystem storage systems deliver high performance, efficiency, and reliability to various storage environments, helping to address performance issues with the most important applications and infrastructure. These storage systems can either complement or replace traditional hard disk arrays for many business-critical applications that require high performance or low latency. Such applications include online transaction processing (OLTP), business intelligence (BI), online analytical processing (OLAP), virtual desktop infrastructures (VDIs), high-performance computing (HPC), and content delivery solutions (such as cloud storage and video-on-demand).

Known existing flash-based technologies, such as PCIe flash cards, serial-attached SCSI (SAS), or Serial Advanced Technology Attachment (SATA) solid-state drives (SSDs) are traditionally located inside individual servers. Such drives are limited in that they deliver additional performance capability only to the dedicated applications running on the server, and are typically limited in capacity. Hybrid shared storage systems, using both flash and spinning disk technology at the same time, offer the potential to improve performance for a wide range of tasks. However, in products of this type, the internal resources of the system (that is, bus, PCI adapters, and so on) are shared between SSD drives and spinning disks, limiting the performance that can be achieved using flash technology.

As shared data storage devices that are designed around flash technology, IBM FlashSystem storage systems deliver performance beyond that of most traditional arrays, even those that incorporate SSDs or other flash technology. FlashSystem storage systems can also be used as the top tier of storage, alongside traditional arrays in tiered storage architectures, such as SVC or Storwize V7000 storage virtualization platforms using IBM Easy Tier functionality. Additionally, IBM FlashSystem storage systems have sophisticated reliability features such as Variable Stripe Redundant Array of Independent Disks (RAID) that are typically not present on locally attached flash devices.

The IBM FlashSystem portfolio includes shared flash storage systems, SSD devices that are provided in disk storage systems, and server-based flash devices.

For more information, see the IBM FlashSystem storage systems home page:

<http://www.ibm.com/systems/storage/flash>

1.3.1 IBM FlashSystem storage system portfolio

IBM recently introduced the FlashSystem portfolio of flash-based storage systems. By using flash solid-state storage technology, FlashSystem devices are both cost-effective and high performance, and can be used to accelerate critical business applications.

Two families of the IBM FlashSystem storage systems exist:

- ▶ IBM FlashSystem 710 and IBM FlashSystem 810 family
- ▶ IBM FlashSystem 720 and IBM FlashSystem 820 family

Note: IBM has a rich portfolio of flash-based systems and products. However, for the purposes of this book, we use the term *IBM FlashSystem storage systems* to refer to the external flash-based systems with Fibre Channel host connectivity only.

IBM FlashSystem 710 and IBM FlashSystem 810

IBM FlashSystem 710 and IBM FlashSystem 810 devices feature Variable Stripe RAID, Active Spare support, and other unique reliability technologies. Connectivity options include four 8 Gbps Fibre Channel (FC) or four 40 Gbps quadruple data rate (QDR) InfiniBand interface ports. FlashSystem 710 and FlashSystem 810 storage systems occupy 1U of standard 19-inch rack space and are available with the following features:

- ▶ Four 8 Gbps FC or 40 Gbps QDR InfiniBand interface ports
- ▶ Up to 5 TB of usable single-level cell (SLC) flash storage (6.9 TB raw capacity), or 10 TB of usable enterprise multi-level cell (eMLC) flash storage (13.6 TB raw capacity)
- ▶ Dual power supplies with batteries to shut down safely in power loss events

For more information about the specifications and features of the IBM FlashSystem 710 and IBM FlashSystem 810 storage systems, refer to the following link:

<http://www.ibm.com/systems/storage/flash/710-810/index.html>

IBM FlashSystem 720 and IBM FlashSystem 820

IBM FlashSystem 720 and FlashSystem 820 storage systems are external, shared flash solid-state storage devices that provide high performance, density, and efficiency in small integrated rackmount footprints. The FlashSystem 720 and FlashSystem 820 have a unique combination of extremely low latency and high performance that offers clients scalable usable capacity points 5 - 20 TB (fully protected) using either SLC or eMLC flash storage media.

FlashSystem 720 and FlashSystem 820 products also incorporate advanced reliability technology, including 2D Flash RAID and Variable Stripe RAID self-healing data protection:

- ▶ Four 8 Gbps FC or 40 Gbps QDR InfiniBand interface ports
- ▶ Up to 10 TB of usable RAID 5 protected SLC flash storage capacity (12.4 TB usable RAID 0, 16.5 TB raw capacity), or 20 TB of usable RAID 5 protected eMLC flash (24.7 TB usable RAID 0, 33.0 TB raw capacity) storage
- ▶ Dual power supplies with batteries to shut down safely in power loss events

For more information about the specifications and features of the IBM FlashSystem 720 and IBM FlashSystem 820 storage systems, see the following site:

<http://www.ibm.com/systems/storage/flash/720-820/index.html>

1.3.2 Differences between IBM FlashSystem families and models

The basic technology and design approach is similar in all FlashSystem storage systems in the portfolio. However, there are some particular differences that should be taken into account when planning and sizing the solution.

Differences between FlashSystem x10 and x20

Table 1-2 briefly describes the key differences between IBM FlashSystem models.

Table 1-2 Key differences between IBM FlashSystem 710/810 and 720/820 storage systems

IBM FlashSystem 710/810	IBM FlashSystem 720/820
1D RAID across flash chips	2D RAID across flash chips and flash modules
Incremental capacities	Incremental capacities
No flash hot-swap	Flash module hot-swap
5 TB/10 TB maximum capacity	10 TB/20 TB maximum capacity

Differences between FlashSystem 710/720 and 810/820

The main difference between IBM FlashSystem models 710/720 and 810/820 is the flash memory technology that is used to provide storage capacity. For the differences between these two technologies, refer to 1.4.3, “Single-level cell memory” on page 16, and 1.4.4, “Multi-level cell memory” on page 17.

Table 1-3 briefly describes the key differences between IBM FlashSystem 710/720 and 810/820 models.

Table 1-3 Key differences between IBM FlashSystem 710/720 and 810/820 storage systems

IBM FlashSystem 710/720	IBM FlashSystem 810/820
Single-level cell (SLC) flash memory	Enterprise multi-level cell (eMLC) flash memory

Usage considerations for IBM FlashSystem storage systems

Selection of the appropriate storage system from any vendor always involves a compromise between the price and the performance. For flash-based systems, reliability must also be considered.

There are two groups of flash storage systems in the IBM portfolio:

- ▶ FlashSystem 720/820 with 2D Flash RAID data protection
- ▶ FlashSystem 710/810 without 2D Flash RAID data protection

Selection of the appropriate system depends on consideration of the following factors:

- ▶ Availability
- ▶ Flash memory technology
- ▶ Initial and maximum storage capacity
- ▶ Price per TB

Price per TB and flash memory technology should be considered during the initial solution design or after a detailed examination of the workload characteristics. Selection between performance and availability is not an obvious choice.

Figure 1-4 provides comparative performance and availability information for IBM FlashSystems to assist in this process.

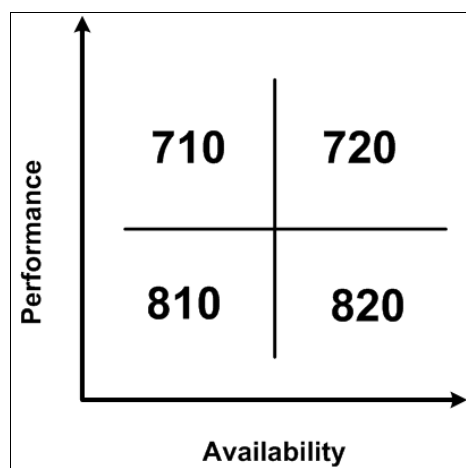


Figure 1-4 Quadrant chart detailing comparative performance and availability for IBM FlashSystems

Figure 1-4 shows the positioning of the flash systems in terms of availability and performance, allowing you to select between the most reliable and performing systems.

In addition, we offer the following criteria to be considered for selecting IBM FlashSystem 720 and 820 storage systems:

- ▶ If the environment is not known completely, or the workload amount and access patterns are not clearly identified, consider using IBM FlashSystem 720/820 first. This provides a balance between availability and performance.
- ▶ 2D Flash RAID data protection makes these systems capable of providing long-term data storage for frequently accessed and changing data.
- ▶ IBM SVC Easy Tier environments are ideal candidates for these systems.
- ▶ Implementation of IBM FlashSystems 720/820 into an existing IBM SVC with HDD-based disk systems environments will be more beneficial from an extent migration and data relocation point of view.

Likewise, we offer the following criteria to be considered for selecting IBM FlashSystem 710 and 810 storage systems:

- ▶ IBM FlashSystem 710/810 should be used in environments where mirroring or protection is provided by another layer in the environment.
- ▶ These systems can be used also to provide low latency for the following applications:
 - HPC data processing environments require the lowest latency possible. This is often because the tasks are single-threaded and meant to be processed sequentially. Data protection can be provided by traditional methods of making regular backups to keep the intermediate results of the processing. Data sources can be kept separately on Tier 1 and Tier 2 storage systems.
 - OLTP database environments, which require requests to specific areas of the data to be serviced with the lowest latency. FlashSystems can be used to store and service that data during peak periods of activity, or to provide the Tier 0 storage level to store the hottest data. Additional data protection can be provided by the Volume Mirroring option of SVC, or the mechanisms of the database software.
 - Video transcoding, processing, and editing tasks can also benefit from the ability of the IBM FlashSystem 710/810 to provide low latency I/O. This can reduce the processing

and transcoding times and reduce the time of the delivery of the content. Data protection may be provided using various methods. However, FlashSystems typically play the role of the temporary storage in this use case. If any error occurs, the data will be replaced with the original after the systems are fixed.

- Virtual Desktop Infrastructures, utilizing multiple access to single images, can experience significant benefit from IBM FlashSystems 710/810, which can allow them to serve more requests with lower latency during peak times of the workloads.

1.4 Introduction to flash solid-state technology

SSDs have evolved from the small storage capacity in mobile devices and selected computer systems, where they are used to provide low-power consumption and resist rough handling, to an enterprise class server and external storage, where SSDs provide high data transfer rates and extremely low access latency. Because flash technology is the key component of SSDs, it is important to understand it and know its advantages and limitations.

Flash memory is an electronic non-volatile storage device, which can be electronically programmed and reprogrammed. Two predominant types of flash memory are currently available in the market of storage technologies for portable devices and high-speed low latency access for servers and storage systems: *NOR² flash memory* and *NAND flash memory*. Because NAND flash memory provides higher density of storage, it has been wider used for storage devices like SSDs or consumer USB storage drives.

NOR flash has been used mostly for storing firmware and microcode in devices that require rapid booting and faster waking from the power-off state. The internal architecture of NOR flash enables short read times, which are critical for the random access nature of microprocessor instructions. This type of flash memory is ideal for lower-density, high-speed, mostly read-only applications, often referred to as *code-storage applications*.

NAND flash is an alternative to NOR flash. Its architecture's high storage density and smaller cell size enable faster write and erase performance by programming blocks of data. NAND flash is ideal for low-cost, high-density, high-speed program/erase applications, often referred to as *data-storage applications*. With the rapid growth in technology and the increasing demand to store and manage data in more devices. NAND memory, with its simple erase and write capability, has become more popular than NOR memory. Technology advancements have significantly decreased the cost of NAND memory and made it the popular solution for the flash-based storage devices.

The most significant technological difference between these two types of memory is the way that the data is addressed. NOR flash is addressed byte by byte, whereas NAND is addressed by page number. Pages are a power of two, commonly 512 or 2048 bytes, and optimized for reading and writing a page at a time. Erases also take place at the byte level for NOR memory, where NAND has to erase a whole block at one time, typically of 64 pages in size.

All future references to flash memory-based devices and systems in this book should be considered as NAND-based.

² Names NOR and NAND stand for the “Not-OR” and “Not-AND” logical operations and the appropriate types of the logical gates that this type of memory is made of. For more information about the technology used, refer to the following article: Pavan, Paolo; Bez, Roberto; Olivo, Piero; Zononi, Enrico (1997). “Flash Memory Cells – An Overview”. Proceedings of the IEEE 85 (8) (1997). pp. 1248 – 1271. doi:10.1109/5.622505.

1.4.1 Types of flash memory used in IBM FlashSystems

The two most commonly available types of flash memory that is used in most storage systems on the market today are: *SLC* and *MLC flash memory*. Before explaining each technology, we first detail how the flash cell operates.

Cell operation

Each cell consists of a transistor with an additional “floating” gate, which can store the electrons.³ The charge of the electrons applied to the gate allows the cell to be programmed to state 1 or 0, and the charge kept in the floating gate layer maintains that state. Because the “floating” gate is electrically isolated by an oxide layer, any electrons placed on it are trapped there, making flash memory non-volatile. Flash memory works by adding (charging) or removing (uncharging) electrons to or from a floating gate. For the internal structure of the transistor, refer to Figure 1-5.

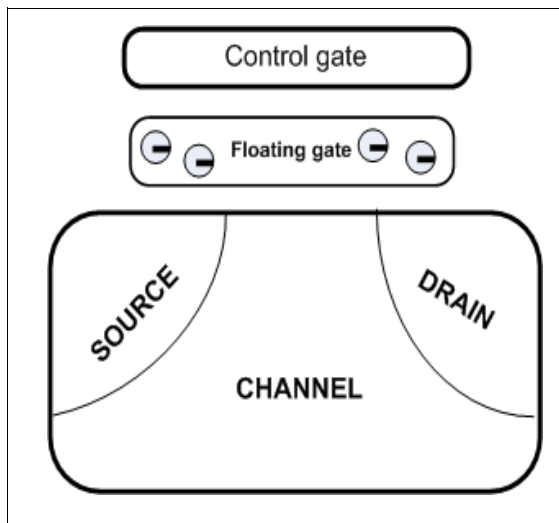


Figure 1-5 Memory cell transistor (simplified schematics)

Reading operation

For reading, the gate is electronically disconnected to measure the voltage levels between the drain and the source of the transistor. When electrons are present on the floating gate, current cannot flow through the transistor, and the bit state is 0. This is the normal state for a floating gate. When electrons are removed from the floating gate, current is allowed to flow, and the bit state becomes 1. A bit's 0 or 1 state depends upon whether or not the floating gate is charged or uncharged.

Write and erase operations

Writing is changing the state of the cell from one state to another. In terms of the transistor, that means applying or removing the charge of the gate. This is done by applying a programming voltage (which is quite high) to the gate and grounding the channel, setting up an electric field such that electrons are attracted to the surface of the channel. Some of these electrons have enough energy to tunnel through the insulating layer. These electrons are captured by the floating gate. Erasing is the opposite operation, where the gate is grounded and with programming voltage applied to the channel to create an electric field with the opposite polarity. This attracts electrons back to the channel, many of which will have enough energy to cross the insulating barrier.

³ There are two transistors in a single cell, but to keep things simple, we treat the two as one.

1.4.2 Unique characteristics

Two important characteristics of flash technology present challenges regarding designing storage products.

Disturb errors

Erasing (writing) the cell cannot be done on a one-by-one basis, due to the power of the electric fields applied to the gate. The size of the elements used in the cell is comparable to the length of the wave in the electrical field, so it may likely disturb the nearest cells and make them become reprogrammed also. To overcome this limitation, memory is written in terms of pages, typically 1 KB - 4 KB size. Erasing occurs at the block level, which is typically 32 - 128 pages.

Limited write/erase cycles

Due to the nature of the materials used and the methods of programming and erasing the cell, electrons must have enough energy to be able to cross the insulating oxide. Some of them have enough energy to cross the barrier that is insulating the gate and the transistor channel, but not enough energy to return. This forces them to reside in the insulating oxide. Over time, as more write/erase cycles pass through the cell, more electrons become trapped, increasing the charge of the gate itself. Eventually, the state between charged and uncharged becomes less easily detectable, rendering the cell useless. This process is commonly called *memory aging* or *wearing-out*. See Figure 1-7 on page 17 for reference.

1.4.3 Single-level cell memory

Single-level cell (SLC) memory operates similarly to the process described above. It is capable of storing one bit at a time, and the cell can be in only one of two possible states: programmed or erased. The voltage value measured on the cell depends entirely on the amount of the charge applied to the floating gate. To determine the state of the cell, the measured voltage value is compared to a reference voltage value. If the measured voltage is within a specific range that is higher than the reference voltage, the cell state is *programmed*. If the measured voltage is within a specific range that is lower than the reference voltage, the cell state is *erased*. Figure 1-6 shows the voltage reference for SLC.

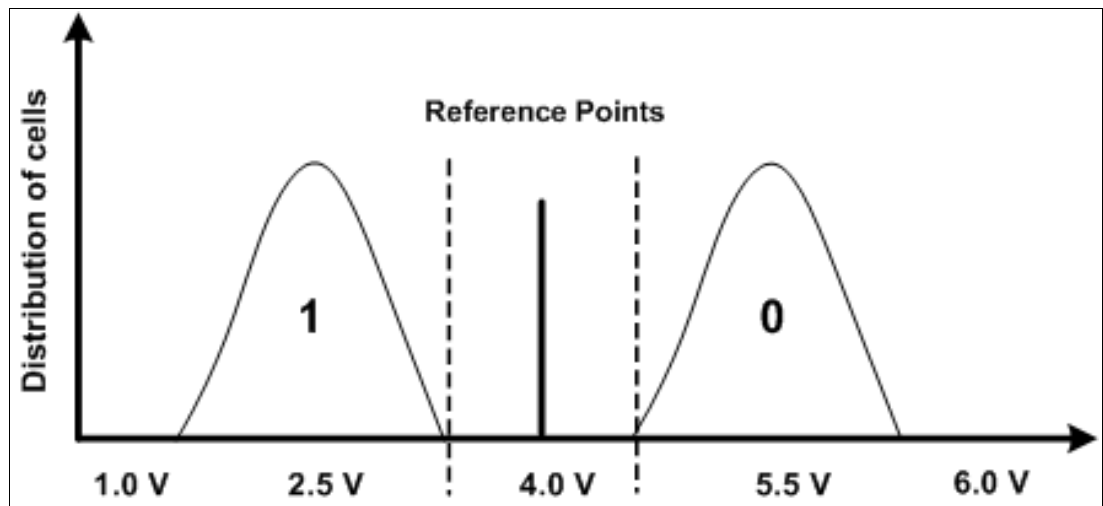


Figure 1-6 Voltage reference for SLC

Table 1-4 shows the possible states of the memory cell.

Table 1-4 SLC levels

Value	State
0	Programmed
1	Erased

As the number of the write/erase operations performed on the cell increases, the difference between *programmed* and *erased* measured voltages gets smaller, as shown in Figure 1-7. As the values approach each other, it becomes difficult to determine the state of the cell. This happens because accumulated charge at the floating gate prevents the complete cut-off of the current in the cell.

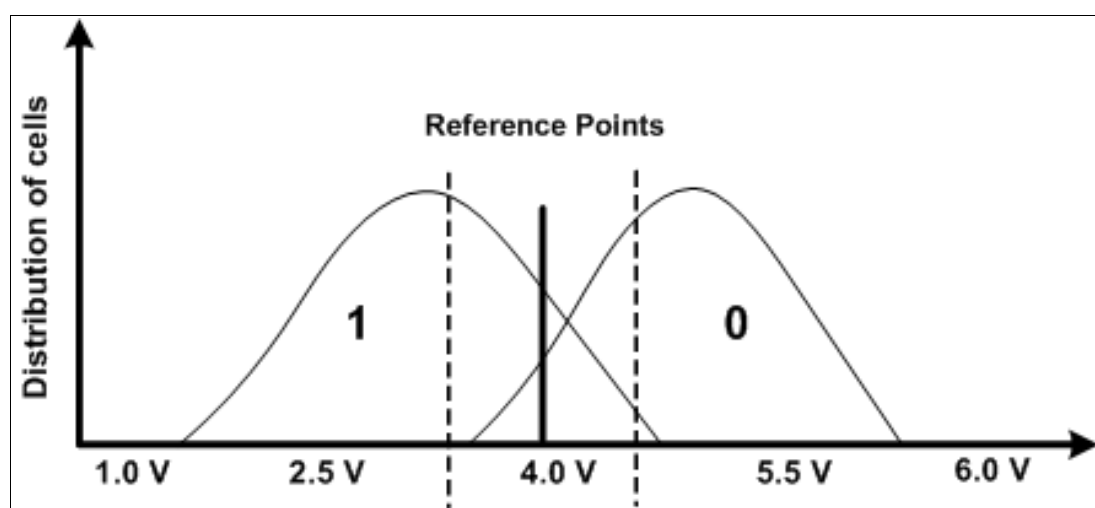


Figure 1-7 Potential overlapped voltage levels of SLC

1.4.4 Multi-level cell memory

Multi-level cell (MLC) memory⁴ can have more than two threshold levels, and as a result, store more than one bit at a time. Typically, they store two or three bits. After erasure, the cell is in one of two erased states. By changing the charge at the floating gate, the cell can be programmed from fully erased to partially erased, to partially programmed, and, finally, to fully programmed. This is done in the same manner as described earlier for gradually programming the SLC cell, by applying write pulses, then sensing the amount of charge to ensure that the cell was properly programmed. See Table 1-5 for the MLC levels.

Table 1-5 MLC levels

Value	State
00	Fully programmed
01	Partially programmed
10	Partially erased
11	Fully erased

⁴ In this book, references to multi-level cell memory refer to both multi-level cell and enterprise multi-level cell memory.

As shown in Figure 1-8, the gaps between the different states are closer than as in SLC. As a result, the measured voltages can begin to overlap sooner, reducing the usable life of the cell. In other terms, this can be thought of as a lower “signal-to-noise ratio” for MLC than for SLC, requiring better error correction codes and mechanisms.

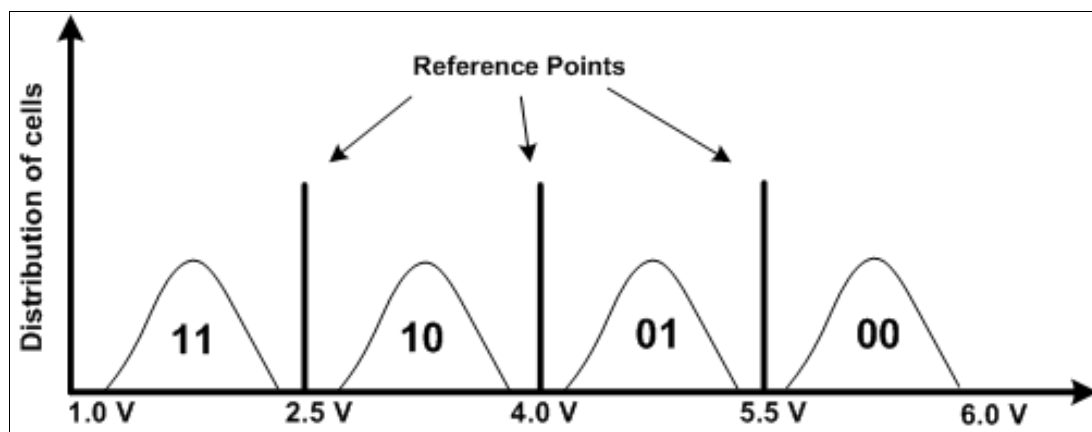


Figure 1-8 Voltage reference for MLC

The ability of MLC to have multiple states allows it to store more data per cell, but the amount of write/erase operations increases accordingly. This leads to shorter usable life of MLC flash as compared to SLC flash. For information about estimating the lifetime of flash, see 1.4.6, “Flash memory lifetime” on page 21.

Note: There are MLC flash chips that can store three or four bits per cell. For the purposes of this book, we refer to MLC flash chips storing two bits per cell only.

1.4.5 Solid-state drive architecture

SLC and MLC memory is a complex technology that should be properly managed to ensure that it provides all its potential benefits. Solid-state drives are the most popular implementation format of this technology in the enterprise storage space. Figure 1-9 on page 19 demonstrates the common architecture of the SSD.

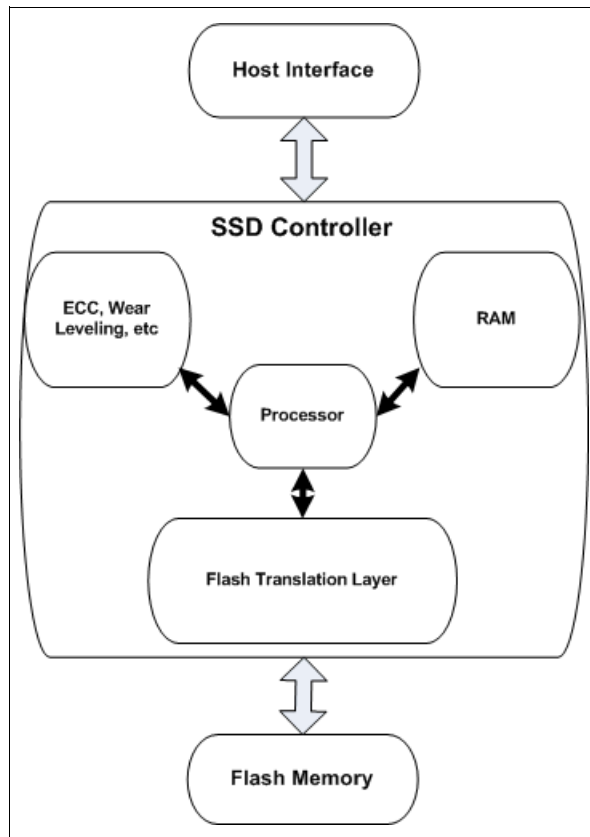


Figure 1-9 Architecture of the solid-state drive

Following are basic components of the SSD:

- ▶ Flash memory
- ▶ Disk controller
- ▶ Interface controller

Depending on the manufacturer, there might be other components included in the SSD, such as additional flash memory controllers, error correction code (ECC) controllers, and additional blocks in the disk controller.

Flash memory

Flash memory is the basic component of the SSD that provides the data storage. Depending on the disk purpose, the memory can be SLC or MLC. The amount of raw memory capacity in the SSD is usually several percent more than the usable capacity. This additional memory capacity is used for data placement, one of the key techniques used to address wear-leveling and increase the lifetime of the SSD. Flash memory components also include all necessary communication and ECC controllers.

Interface controller

The interface controller is responsible for presenting the SSD as a SATA, SAS, or FC drive in the operating system or external disk system. It also includes all necessary communications, connectors, and controllers.

Disk controller

The disk controller is a key component of the SSD architecture, and commonly the technology that differentiates the SSD drives of various manufactures. In general, the disk controller is the System-on-Chip (or several chips) representing the processor, firmware, random access memory (RAM), flash translation layer (FTL), wear-leveling algorithms, ECC controllers, and possibly, other components for encryption, data deduplication, additional aging prevention, and so on. The disk controller provides the following functionality:

- ▶ Flash translation layer function
- ▶ Garbage collection function
- ▶ Wear-leveling function

Flash translation layer

The flash translation layer emulates a standard block device by exposing only read/write operations to the upper software layers. It performs the virtual-to-physical address translations and hides the erase-before-write characteristics of flash memory. The FTL also emulates the over-write operation with out-of-place updates. In out-of-place updates, the physical location of data is changed on every write request, and the FTL maintains a mapping table between the logical sector and physical location. The address mapping table is usually stored in a small piece of RAM. According to the size of the mapping unit used, FTL schemes are classified as block mapping, page mapping, and hybrid mapping⁵. The flash translation layer also provides garbage collection and wear-leveling capabilities that are vital to the performance and reliability of flash SSD.

Garbage collection

Garbage collection (GC) is a process that erases dirty blocks and recycles their obsolete pages. If a block that is selected for erasure has some valid pages, those pages are migrated to other blocks before erasing the block. Because garbage collection involves time-consuming erase operations and numerous internal reads and writes, an ongoing GC process can stall incoming user requests until it completes. As a consequence of the queuing delay, the performance of flash SSD can be significantly degraded by 20%⁶. Various GC mechanisms have been proposed to minimize the garbage collection overhead. In particular, much effort has been focused on reducing the total amount of copied data from the erased blocks, because moving valid data from erased blocks to new blocks represents a large portion of the total execution time of a garbage collection process. The most common way to achieve this goal is to separate data based on update frequency, so that the number of obsolete blocks (that is, blocks that have no valid data) and almost-obsolete blocks (that is, blocks that have very little valid data) can be increased. Recycling obsolete or almost-obsolete blocks can substantially reduce overhead.⁷

Wear-leveling

The purpose of wear-leveling algorithms is to evenly distribute block erasures over the flash memory, and thus enhance its endurance. Wear-leveling algorithms can be classified into dynamic wear-leveling and static wear-leveling.

Dynamic wear-leveling

The performance of dynamic wear-leveling depends on hot and cold data identification. Cold data always stays in the same blocks, regardless of whether updates to hot data wear out

⁵ Covering these techniques is out of the scope of this book. For more information, refer to the following paper: Hot/Cold Clustering for Page Mapping in NAND Flash Memory, Ilhoon Shin, IEEE Transactions on Consumer Electronics, Vol. 57, No. 4, November 2011.

⁶ Estimated value, might vary.

⁷ For more information about garbage collection techniques, refer to this paper: Making Garbage Collection Wear Conscious for Flash SSD, Jonathan Tjioe, Andrés Blanco, Tao Xie, Yiming Ouyang; 2012 IEEE 7th International Conference on Networking, Architecture, and Storage.

other blocks. In a conditional threshold wear-leveling algorithm, when a block is erased, the cold data in the block with the minimum erase count is moved to the newly erased block, effectively swapping hot data and cold data.

Static wear-leveling

In static wear-leveling, when the difference between the maximum block erase count and minimum block erase count exceeds a specified threshold, static data is moved to the hot block to balance the erase count in each block. Accordingly, a block erasure table (BET) is created to identify which blocks have been erased during a given time period.

The wear-leveling threshold for various system environments can be very different. Using inadequately tuned parameters can cause unexpectedly high wear-leveling overhead and poor wear-leveling performance. The implementation complexity of the wear-leveling algorithm determines the applicability of the algorithm. Existing wear-leveling algorithms require prior knowledge of the system environment for threshold tuning, and thus increase their design complexity.⁸

1.4.6 Flash memory lifetime

The endurance of the flash memory is an important characteristic to consider when planning a flash storage implementation. A flash memory page cannot be programmed unless its blocks are erased first. A program (write) operation followed by an erase operation constitutes a program/erase cycle. If we consider that a flash device controller attempts to evenly distribute write activity across all available memory cells, we can infer that a single P/E cycle corresponds to writing once to the entire available capacity of the device.

Example: A program/erase cycle of a single 200 GB SSD equates to 200 GB of data.

The amount of program/erase cycles depends on the specific flash memory technology. In general, industry accepted numbers are:

- ▶ MLC: 10,000 program/erase cycles
- ▶ eMLC: 30,000 program/erase cycles
- ▶ SLC: 100,000 program/erase cycles

IBM FlashSystem storage systems incorporate the following proprietary technologies to increase the lifetime of the flash memory:

- ▶ Variable Stripe RAID
- ▶ Two-Dimensional (2D) RAID
- ▶ Wear-leveling algorithms
- ▶ Over-provisioning technology

The lifetime of a specific flash-based device can be estimated in the following ways:

- ▶ If the memory technology that is used is known, the write activity of the applications should be taken into account. Calculate out how much time is required to perform a full program/erase cycle under the average write activity. Multiply this amount of time by the number of program/erase cycles estimated to this type of the technology, as shown in Example 1-1 on page 22.

⁸ For more information, see: A Low-Complexity High-Performance Wear-Leveling Algorithm for Flash Memory System Design, Ching-Che Chung and Ning-Mi Hsueh, 978-1-4577-1729-1/12, ©2012 IEEE

Example 1-1 Calculating flash device lifetime using P/E cycles and average write activity

Drive size: 200 Gbytes SSD
Flash type: eMLC
Average write activity: 50 Mbytes/sec
P/E cycles: 30,000

1 P/E cycle = (200 Gbytes/50 Mbytes per sec)
1 P/E cycle = 4000 seconds
1 P/E cycle = ~1.1 hour

Lifetime = 30,000 P/E cycles * (~1.1 hours per PE cycle)
Lifetime = ~33,000 hours
Lifetime = ~3.7 years.

Thus, after 3.7 years of being under this average write workload the SSD will start degrading.

-
- ▶ Alternatively, the *endurance* specification of the flash device (measured in capacity) can be used. Endurance refers to the amount of randomly accessed data that can be written to the device before it starts to degrade. Dividing the endurance value by the average write activity value will yield the potential lifetime (measure in time), as shown in Example 1-2. Dividing by the capacity of the drive will yield the number of potential program/erase cycles.

Example 1-2 Calculating flash device lifetime using endurance specification-device capacity

Drive size: 200 Gbytes SSD
Endurance: 3.7 PBytes
Average write activity: 50 Mbytes/sec
Lifetime = (3.7 Pbytes / 50 Mbytes/sec / 31536000 sec/year)
Lifetime = ~ 2.3 years.
P/E cycles = (3.7 Pbytes / 200 Gbytes)
P/E cycles = 18500
Flash type: eMLC

Thus, after 2.3 years of being under this average write workload the SSD will start degrading.

1.5 IBM FlashSystem storage systems technology overview

IBM FlashSystem storage systems, with an all-hardware data path, are engineered to deliver the lowest possible latency. They incorporate proprietary flash controllers and leverage numerous patented technologies. FlashSystem controllers have proprietary logic design, firmware, and system software. There are no commodity 2.5-inch SSDs, PCIe cards, or any other significant non-IBM assemblies within the system. The flash chips, FPGA⁹ chips, CPUs, and other semiconductors in the system are carefully selected to be consistent with the “purpose-built” design, which is designed from the ground up for high performance, reliability, and efficiency.

⁹ FPGA: Field-programmable Gate Array. For more information, refer to the following link:
http://www.eecg.toronto.edu/~vaughn/challenge/fpga_arch.html

Notable architectural concepts of the IBM FlashSystem storage systems are:

- ▶ Hardware-only data path
 - Leverages FPGA's extensively
 - Field-upgradable hardware logic
 - Less expensive design cycle
- ▶ Extremely high degree of parallelism
- ▶ Intelligent flash modules
 - Series 7 FPGA-based flash controller
- ▶ Distributed computing model
 - 16 low-power PPC processors
 - Interface and flash processors run thin RTOS¹⁰
 - Not in active in data transfer
 - Responsible for garbage collection, monitoring
- ▶ Management processor communicates with the interface and flash processors via internal network
 - Minimal communication

The hardware-only data path design of IBM FlashSystem storage systems eliminates the software layer latency impact that is found in other vendor products. In order to achieve such extremely low latencies, IBM FlashSystem advanced software functions are limited. For this reason, implementing IBM FlashSystem storage systems with IBM SAN Volume Controller can offer an unmatched combination of performance, low latency, and rich software functionality.

It is important to note that IBM FlashSystem storage systems are different from traditional disk arrays regarding code upgrades. Unlike software-heavy storage arrays, IBM FlashSystem storage systems do not need frequent code upgrades. In fact, many clients perform only a handful of code upgrades throughout the lifetime of the system.

In order to simplify the understanding of the particularities of the flash technology that is used in the IBM FlashSystems, we provide the explanation of the basic terms used in this book. See Table 1-6 for reference.

Table 1-6 Technology terms explanation

Technology term	Explanation
Flash memory die	Simply, it is a crystal that incorporates semiconductor structures, which represent flash memory itself. Those structures are: memory cell transistors, bit lines, word lines, control structures, and so on. Each die has a certain number of memory pages and provides certain level of I/O characteristics. Dies are stacked in the package in various ways for parallel access and aggregated performance.

¹⁰ Real-Time Operating System

Technology term	Explanation
Flash memory page, block	A flash memory page is a minimal addressable amount of memory. Typically, the size of the memory page is 4 KB. Flash memory is accessed by groups of memory pages called <i>memory blocks</i> for write and erase procedures. Typically, each block is 32 - 128 pages in size. See 1.4, "Introduction to flash solid-state technology" on page 14 for reference.
Flash memory lane	Simply, it is a logical structure that is used to represent the addressing scheme inside the IBM FlashSystems. For example, Variable Stripe RAID used the ten pages lane to provide the data protection capabilities. Nine pages are used to store data; the tenth page is used to store XOR. For more information, see "Variable Stripe RAID" on page 33.
Flash memory chip	This physical device incorporates several flash memory dies, necessary control structures, and conductors that are required to provide access to the chip's outside connectors. Flash memory chip is the smallest physical memory structure that you can see in the IBM FlashSystem. For more information, see "Flash chips" on page 27, and Figure 1-13 on page 27.
Flash memory plane	Plane in flash-memory technology is both the physical and logical entity. Logically, it means the amount of the flash memory chips that are controlled by the same controller (or several controllers) and are following the same access rules. Physically, it means a number of flash memory chips that are located on the same flash memory board or module and are following the same electrical and signaling requirements. The plane can also incorporate flash memory controllers, connection lines, and other components that are needed for flash memory functioning.

1.5.1 IBM FlashSystem storage systems architecture

IBM FlashSystem storage systems, with an all-hardware data path design, have an internal architecture that is different from the other hybrid (SSD + HDD) or SSD-only based disk systems.

The basic elements of the IBM FlashSystem storage system architecture are:

- ▶ Management control processors
- ▶ Flash modules
- ▶ RAID controllers
- ▶ Dual-ported interface cards
- ▶ Power modules, batteries, and fans

Figure 1-10 on page 25 shows the internal components of the IBM FlashSystem 820 that are organized in a 1U chassis.

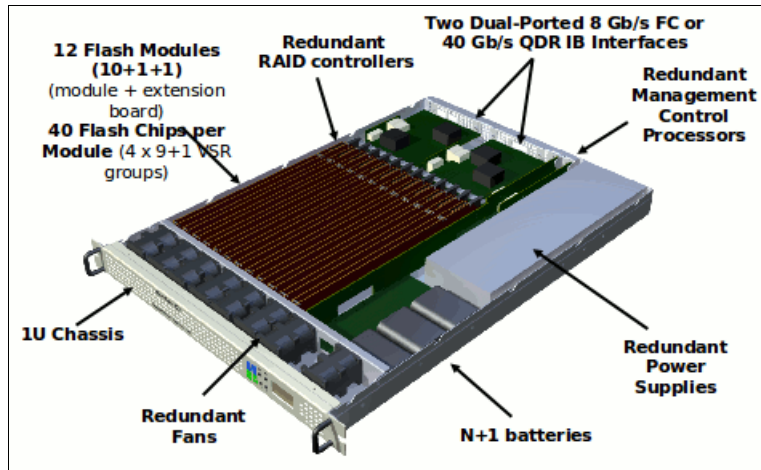


Figure 1-10 Components of the IBM FlashSystem 820 storage system

The basic architecture of the IBM FlashSystem 820 is shown in Figure 1-11.

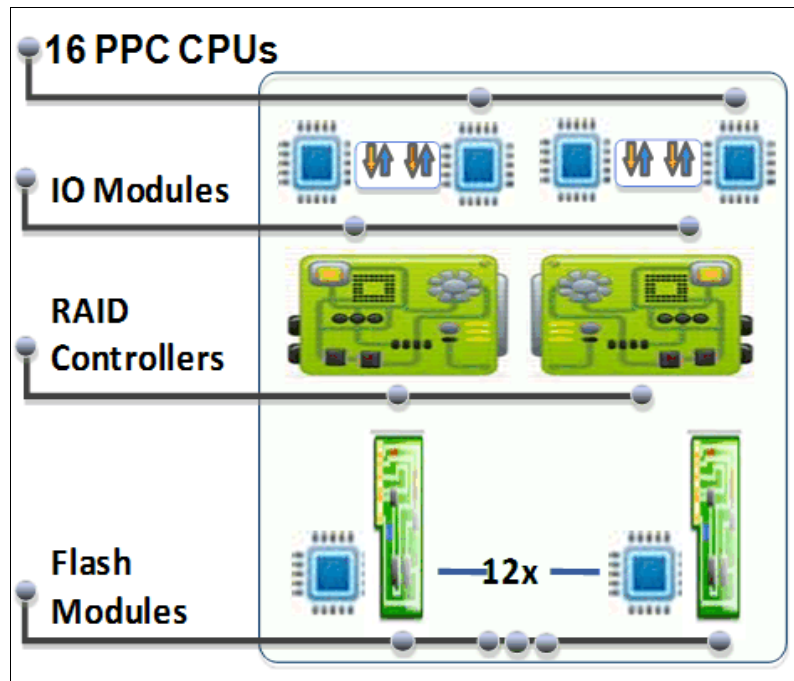


Figure 1-11 Architecture of the IBM FlashSystem 820 storage system

We now describe the main components of the architecture.

Flash modules

Flash modules are the main storage components of the IBM FlashSystem 820. They provide the storage capacity, addressing of the flash memory chips, RAID capability, write ordering, and layout. Each flash module consists of the following components:

- ▶ Flash controller
- ▶ Flash chips
- ▶ Gateway interface
- ▶ Expansion board

Note: The flash modules inside the IBM FlashSystem 820 are also referred to as *flashcards*.

Flash modules are hot-swappable in IBM FlashSystem 820 storage system.

Primary board

The primary board provides capacity to the flash module. It contains 40 flash chips and delivers up to 1 TB¹¹ of capacity. The primary board connects directly to the main board of the IBM FlashSystem 820.

Expansion board

The expansion board provides additional capacity to the flash module. It contains 40 flash chips and delivers up to 1 TB of capacity. The expansion board mounts to the primary board and provides access to the modules through the special connector.

Both boards utilize the same physical connection of the primary board to the main board of the IBM FlashSystem 820.

Figure 1-12 shows the primary and expansion board. Flash chips and highlighted in blue, flash controllers are highlighted in red, and the gateway interface is highlighted in purple.

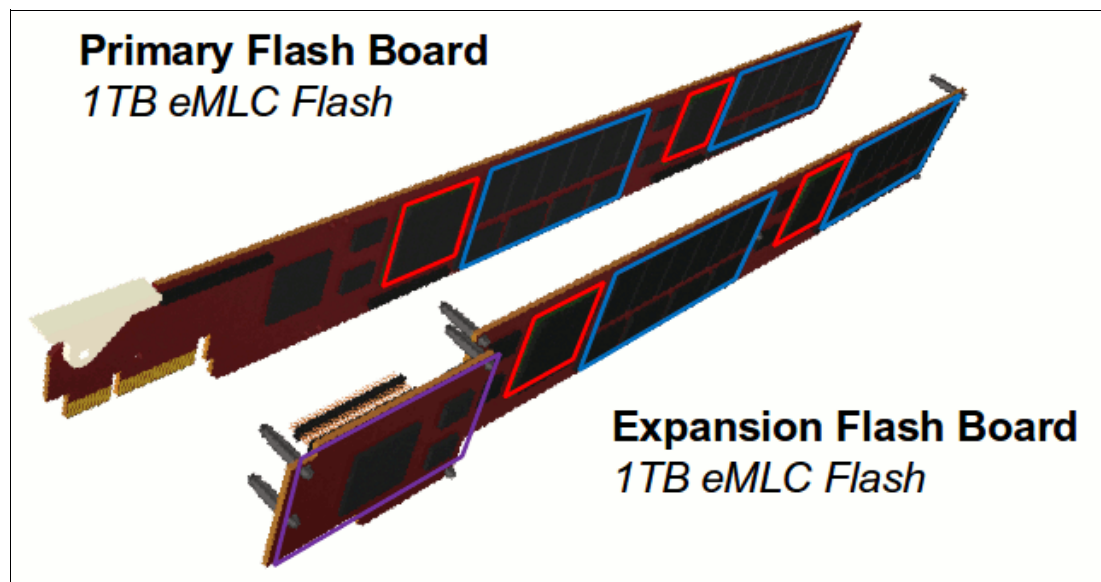


Figure 1-12 Flash module primary and expansion boards

Figure 1-13 on page 27 shows the architecture and the logical layout of the flash module.

¹¹ At the time of writing.

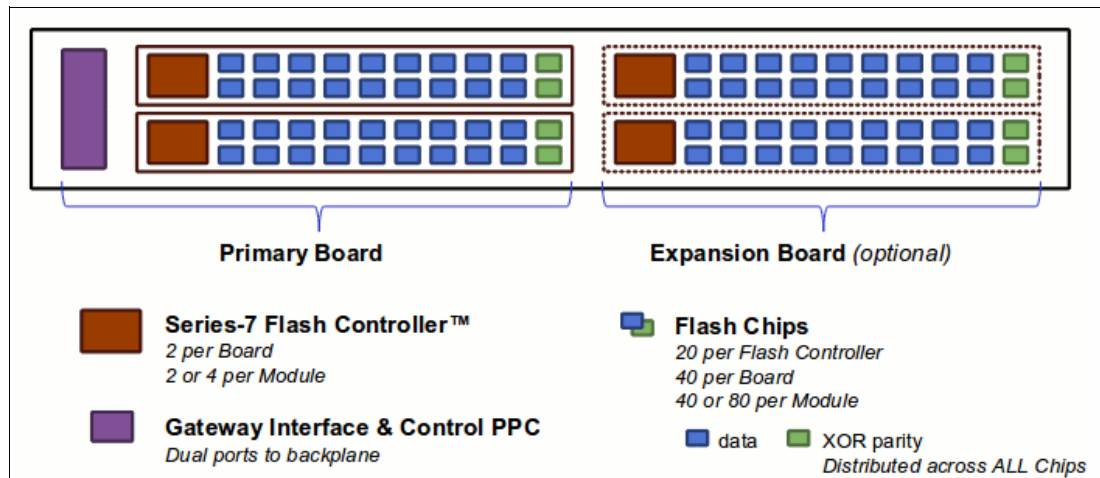


Figure 1-13 Flash module logical layout

Flash chips

The flash chip is the basic storage component of the flash module. There can be a maximum of 80 flash chips per flash module — 40 in the primary board and 40 in the expansion board. They can be of either SCL or eMLC technology, but not both in a single storage system. Combining flash chips of different flash technologies is not supported in the same flash board, module, or storage system.¹²

Flash Controller

The Series-7 flash controller FGPA of the flash module is used to provide access to the flash chips and is responsible for the following functions:

- ▶ Provide data path, hardware I/O logic
- ▶ Look up tables and write buffer
- ▶ Control 20 flash chips
- ▶ Operate independently of other controllers
- ▶ Maintain write ordering and layout
- ▶ Provide write setup
- ▶ Maintain garbage collection
- ▶ Provide error handling

The basic components of the flash controller are shown in Figure 1-14 on page 28.

¹² At the time of writing.

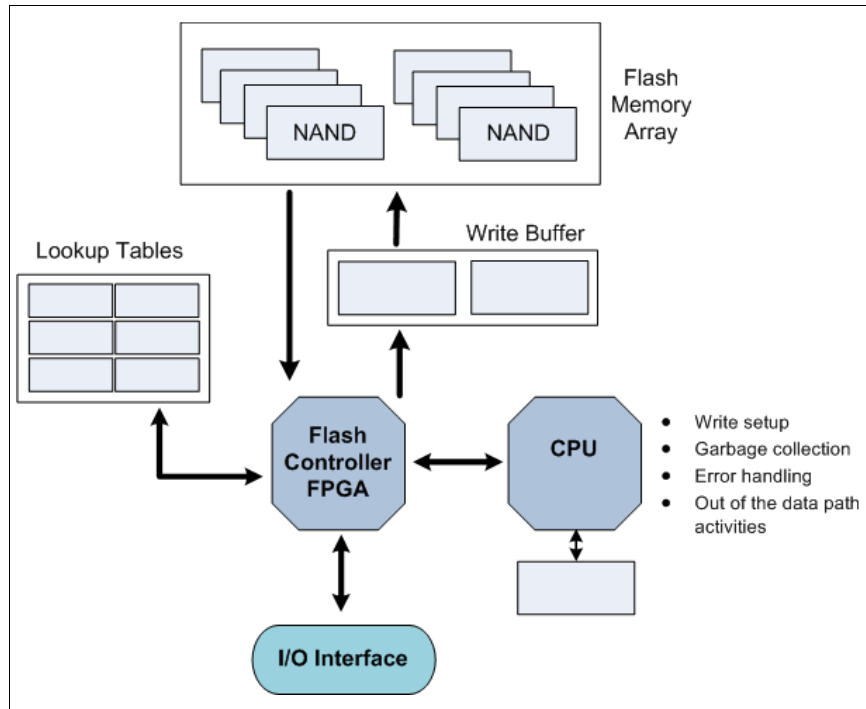


Figure 1-14 Flash controller design

The concurrent operations performed on the flash chips include moving data in and out of the chip via direct memory access (DMA), internally moving data and performing erases. This means that while actively transferring user-data in the service of host initiated I/O, the system can be simultaneously running garbage collection activities without impacting the I/O. The ratio of transparent background commands running concurrent to active data transfer commands is 7 to 1.

There are a maximum of four flash controllers per flash module: two per primary board and two per expansion board (Figure 1-13 on page 27). Each flash controller can perform 40 x 4 KB DMA operations in parallel, including IBM Variable Stripe RAID (VSR™) operations. For more information about VSR, see “Variable Stripe RAID” on page 33.

There are 44 flash controllers total in the IBM FlashSystem 820, supporting 1760 4 KB DMA operations in parallel. The IBM FlashSystem 820 can also issue eight concurrent operations per flash chip. With up to 80 chips per flash module, and 12 flash modules per system, an IBM FlashSystem 820 could be servicing up to 7680 simultaneous operations on the flash medium. This makes queue depth settings important when attaching hosts to the IBM FlashSystem 820.

Gateway interface

The gateway interface FPGA is responsible for providing I/O to the flash module and DMA path. It is located on the flash module and has two connections to the backplane.

RAID controllers

RAID controllers provide RAID data protection functionality for the flash modules in the IBM FlashSystem 820 storage system.

RAID controllers work in active/active load balanced mode, utilizing a round-robin balancing mechanism.

Figure 1-15 depicts the data flow inside the IBM FlashSystem 820 and the RAID controllers. The process flow is as follows:

- ▶ Data is split into parallel 4 KB transfers
- ▶ 4 KB transfers are rotated round-robin across logical lanes
- ▶ 4 KB transfers are rotated across RAID controllers for load balancing
- ▶ If a RAID controller fails, all transfers are routed through the non-failed controller

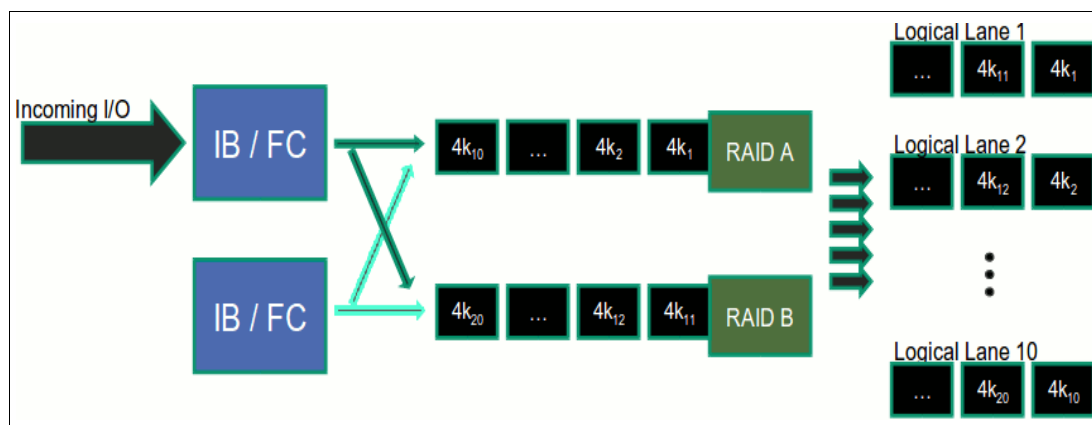


Figure 1-15 Incoming I/O data flow through the IBM FlashSystem 820 RAID controllers

Figure 1-16 shows the data flow through a single RAID controller to the flash modules of an IBM FlashSystem 820. Here you can see how the data is spread across the flash modules. For more information, see “2D Flash RAID” on page 34. The process flow is as follows:

- ▶ RAID stripes consist of 40 KB of data (4 KB blocks across 10 modules) + 4 KB parity
- ▶ Parity is round-robin rotated across flash modules upon each RAID stripe
- ▶ Optimized RAID stripe is updated on write
 - Old data and old parity read are read (if not already loaded)
 - Old data is backed out of parity
 - New data is added to parity
 - New data and parity are written

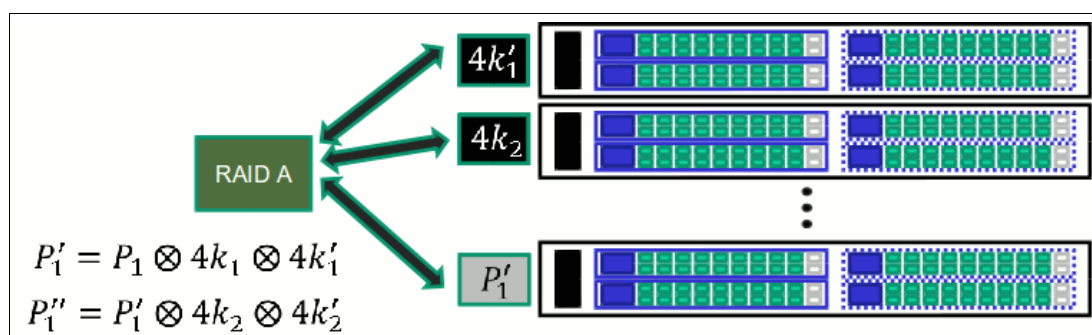


Figure 1-16 Data layout from the RAID controller to IBM FlashSystem 820 flash modules

Interface modules

IBM FlashSystem storage systems offer the choice of 8 Gb Fibre Channel Protocol (FCP) or 40 Gb Quad Data Rate (QDR) InfiniBand connectivity to match the performance of the most demanding I/O intensive applications¹³.

¹³ At the time of writing.

Note: InfiniBand connections are out of the scope of this book.

All available interface modules have the following common points:

- ▶ Common developed USIC code base:
 - PPC for instruction decode (direction) and FPGA for data path (flow)
 - uCOS real-time OS
 - Fast-path operations handled completely on the card
 - Active/active across ports
 - Active/active across cards

8 Gb Fibre Channel with FCP interface modules has the following characteristics:

- ▶ Autosensing 8 Gb/4 Gb/2 Gb
- ▶ Supports Arbitrated Loop (AL) and Point-to-Point (PP) connections
- ▶ Evolutionary upgrades on components (FPGAs, transceivers, uCode)
- ▶ Hot-swappable SFP modules

40 Gb QDR InfiniBand with SCSI Remote Direct Memory Access (RDMA) protocol (SRP)

- ▶ Dual-ported adapters
- ▶ Mellanox protocol driver
- ▶ Supports SRP over RDMA

At the time of writing, Ethernet-based storage interfaces like iSCSI and Fibre Channel over Ethernet (FCoE) are not available on IBM FlashSystem storage systems. File and object-based storage protocols are also not supported, but can be provided if the system is configured behind various gateway appliances.

Interface modules provide internal redundancy, but are not hot-swappable.

Management control processor modules

Management control processor modules (management modules) are responsible for providing Ethernet connectivity for system management by both CLI and GUI. They run a proprietary Linux based operating system that coordinates and monitors all significant functions in the system. Each management module has one RJ-45 Gigabit Ethernet port for remote management. Management modules are configured in an active/passive redundant pair. The following remote management methods are supported:

- ▶ Java based GUI through web browser
- ▶ CLI through Telnet and Secure Shell (SSH)
- ▶ Simple network management protocol (SNMP)
- ▶ Call home functionality

The following tasks can be performed with the management module:

- ▶ Defining the system user accounts and passwords
- ▶ Configuring email notifications
- ▶ Monitoring the status of the system:
 - Event log
 - Fans
 - Temperatures
 - Power
- ▶ Managing other FlashSystem units in the network from a single web console
- ▶ Controlling the FlashSystem unit:
 - Powering on/off
 - Updating firmware

- Configuring network settings
- Changing storage mode (just a bunch of Flash (JBOF) or RAID 5)
- Controlling rebuild process
- Creating logical volumes (LUNs)
- Configuring LUN masking
- Configuring host interfaces
- Configuring user security
- Setting the date and time

Management control processor modules are run in failover mode, but are not intended to be hot-swappable.

For more information about managing the IBM FlashSystem 820 using the CLI and GUI, see Chapter 4, “Configuration and administration” on page 77.

Power supply modules

Power supply modules are run in redundant mode and are hot-swappable. There are two power modules in IBM FlashSystem 820 storage system. The system can sustain single power module failure. Under normal operating conditions, power consumption is equally divided by the power modules.

Cooling fans

Cooling fans are designed to fail in place with internal redundancy, but are not hot-swappable. The IBM FlashSystem 820 storage system has one fan cage in front of the flash modules. These fans cool the components in the system:

- ▶ If a fan degrades in performance, the front panel display will show the speed of the fan. In this situation, the fan is still working, but it is not efficient.
- ▶ If a fan completely stops working, an error is reported on the front panel display.

For more information, see Chapter 5, “Diagnostics, planned outages, and troubleshooting” on page 123.

Batteries

Two batteries are included to provide emergency power to the system in case of a sudden power loss. The IBM FlashSystem 820 is shipped with the batteries connected. These batteries should stay connected at all times, even when the FlashSystem is not in use.

Internal sensors report on the battery voltage level, and a monthly test ensures that the electrical current that is supplied from the batteries will be enough to handle a sudden power loss. If the battery voltage is out of specification or the monthly battery test fails, warnings will be reported.

If they degrade and are unusable, errors are reported. The batteries are redundant; therefore, data is not at risk in the event of a power failure. However, quick replacement of the failed battery is recommended.

See Chapter 5, “Diagnostics, planned outages, and troubleshooting” on page 123.

1.5.2 Data protection and redundancy in the IBM FlashSystem 820

Storage systems of any kind are typically designed to perform two main functions: to store and protect data. IBM FlashSystem storage systems include the following options for data protection:

- ▶ RAID data protection
 - Variable Stripe RAID
 - 2D Flash RAID
- ▶ Flash memory protection methods
- ▶ Optimized RAID rebuild times

Table 1-7 details the data protection layers of the IBM FlashSystem 820 storage system.

Table 1-7 IBM FlashSystem 820 layers of data protection

Layer	Managed by	Protection
System-level RAID 5	Centralized RAID controllers	Module failure
Module-level RAID 5	Each module across the chips	Chip failure, page failure
Module-level Variable Stripe RAID	Each module across the chips	Subchip, chip, or multi-chip failure
Chip-level ECC	Each module using the chips	Bit and block errors

The proprietary 2D Flash RAID data protection scheme of the IBM FlashSystem 820 storage system combines system-level RAID 5 and module-level RAID 5. For more information, see “2D Flash RAID” on page 34.

1.5.3 RAID technologies

Data protection in traditional HDD, SSD, or hybrid (HDD+SSD)-based storage systems is implemented at the disk level. RAID protection defines the single disk as the lowest granularity level of data protection. This means that if something is wrong inside the disk itself, it is treated as failed, and the parity information stored in the RAID array is used to reconstruct the data on the failed drive after it is replaced.

With generic flash-based storage systems, the lowest granularity in the flash module can be the flash memory chip, the data block, or the memory page. Although failure can be identified at these low levels, none of these components can be practically replaced. The memory page and memory block represent logical entities, and the flash chip cannot be replaced because of the technology limitations and the difficulties of field maintenance. Therefore, if a single page, block or chip fails, the whole flash module that incorporates flash chips should be replaced. These single small part failures (kilobytes, megabytes, and so on) can lead to increased maintenance costs and the necessity to back up and restore much larger amounts of data (gigabytes) than what is actually contained in the failed component. The IBM FlashSystem 820 storage system overcomes this limitation through its patented VSR method of data protection.

Variable Stripe RAID¹⁴

VSR is a unique IBM technology that provides RAID-capable data protection of the memory page, block, or whole chip, which eliminates the necessity to replace a whole flash module in a case of a single memory chip failure.

The following problems exist with a traditional RAID-5 approach:

- ▶ If a chip fails, the flash controller uses the parity bit to rebuild lost data.
- ▶ The entire RAID stripe must be relocated. All dies touched by the stripe can no longer be used.
- ▶ If the stripe runs across ten dies, a failure of one die means that nine good dies go to waste.
- ▶ If you map out the full chip, you throw out the remaining good dies in the chip.

In traditional systems, the length of all of the data stripes used in the system is the same. Thus, in such systems, all of the data stored in the system is divided into data stripes of the same length, with each data stripe consisting of the same number of pages, and with each data stripe being stored in the same number of memory locations. In such systems, each data stripe conventionally utilizes the same form of data protection and the data protection information for each data stripe is determined in the same way. It requires the availability of a reserve or back-up storage location to take the place of the failed storage location. Such reserve or back-up locations can be costly and inefficient to provide and maintain, and are not always available.

The IBM FlashSystem storage system controllers are configured to store data such that each page stripe comprises a plurality of pages of data, with each page of data in the page stripe being stored in a flash memory device that is different from the flash memory devices in which the other pages of data in the page stripe are stored. See Figure 1-16 on page 29 for reference.

The system controller is also configured to maintain one or more buffers that contain information reflecting blocks of memory within the flash memory devices that have been erased and are available for storage of information, and to dynamically determine the number of pages to be included in a page stripe. This page stripe is based on the information that is contained in one or more buffers such that a first page stripe can have a first number of pages of data and a second page stripe can have a second number of pages of data, where the first number is different from the second number. See Figure 1-17 for a Variable Stripe RAID layout.

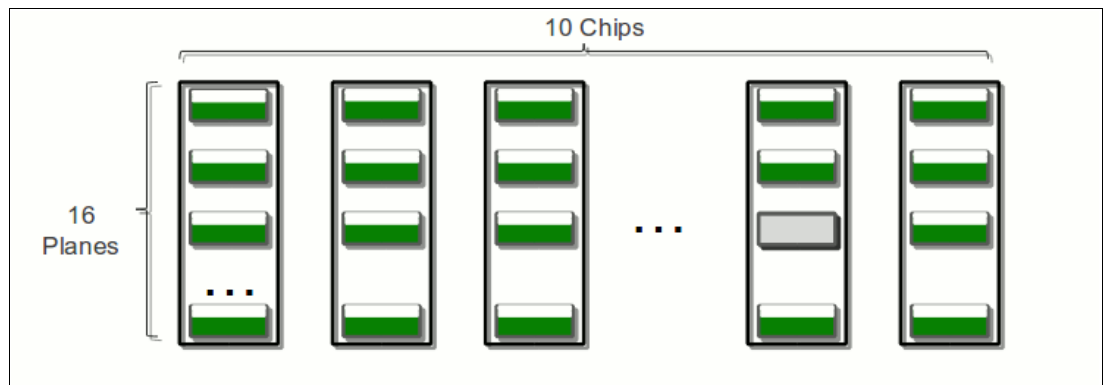


Figure 1-17 Variable Stripe RAID layout

¹⁴ The technology itself is described in the IBM Patent US 7856528 B1 "Method and apparatus for protecting data using variable size page stripes in a FLASH-based storage system". We are using short summarized descriptions for the purposes of this book.

To illustrate how VSR functions, assume that a memory chip associated with a given lane fails and is no longer available to store data. This might occur as a result of a physical failure within the chip, or some damage being inflicted on the address or power lines to the chip in the lane. The failure of the chips in the lane would be detected and the system could change the format of the page stripes that are used so that, as the system reads, writes, and moves data, the data that was previously stored in physical locations across chips in all *ten* lanes using a page stripe format with *ten* pages, is now stored across chips in only *nine* lanes using a page stripe format with *nine* pages as reflected. Thus, no data stored in the memory system was lost, and the memory system can self-adapt to the failure and continue to perform and operate by processing read and write requests from host devices.

This ability of the system to self-adapt automatically, when needed, to chip and intra-chip failures makes the memory system extremely rugged and robust, and capable of operating despite the failure of one or more chips or intra-chip regions. It also makes the system very user-friendly in that the failure of one, two, or even more individual memory chips or devices does not require the removal and potential disposal of previously used memory storage components. The reconfiguration or reformatting of the data to change the page stripe formatting to account for chip or intra-chip failures might reduce the amount of physical memory space that is held in reserve by the system and available for the system for background operation.

Following is a summary of the capabilities of VSR:

- ▶ Patented Variable Stripe RAID allows RAID stripe sizes to vary.
- ▶ If one die fails in a ten-chip stripe, only the failed die is bypassed, and then data is restriped across the remaining nine chips. No system rebuild is needed.
- ▶ VSR reduces maintenance intervals caused by flash failures.

2D Flash RAID

If the flash memory chip is not the only component that can fail in the flash module, data protection at the flash module level is required. 2D Flash RAID, available in the IBM FlashSystem 720 and 820 storage systems, provides flash module level protection, organizing the 12 available flash modules in RAID5-like structure consisting of 10 data modules, one parity module, and one spare module. This provides two very important enhancements to data protection:

- ▶ Two flash modules in the whole system can fail and data can still be accessible and consistent
- ▶ Hot swap option is available

Note: Two flash modules' failure means in total out of twelve modules, not the failure of the two modules at the same time.

2D Flash RAID incorporates the advantages of VSR and RAID 5 across flash modules, providing two-dimensions of protection for the modules. See Figure 1-18 on page 35 for the architecture.

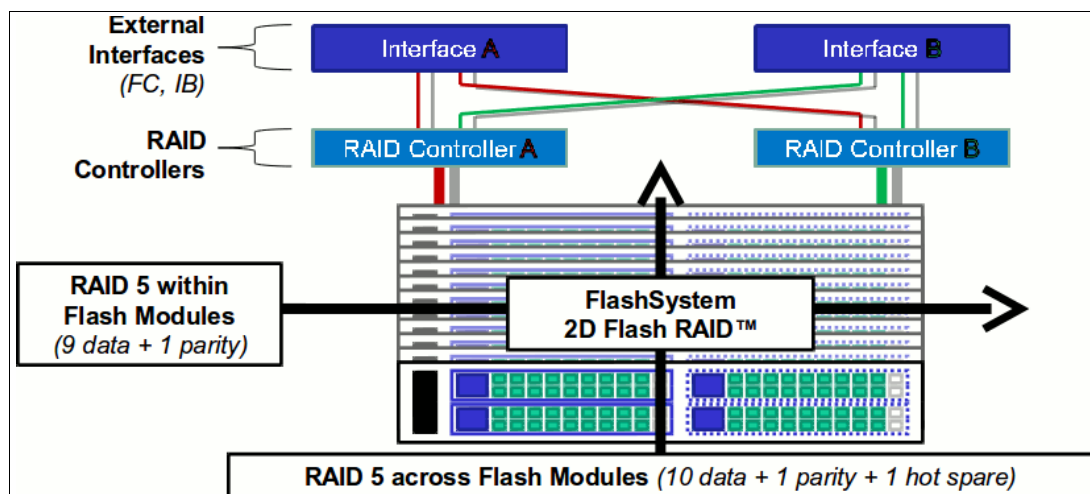


Figure 1-18 Architecture of IBM FlashSystem 820 storage system 2D Flash RAID

RAID 5-like protection for the flash memory within the flash modules, combined with RAID 5 protection for the flash modules themselves, provides almost complete two-layer data protection for the IBM FlashSystem 720/820, making it more reliable than the disk-based storage systems.

1.5.4 Flash memory protection

As described in 1.4.2, “Unique characteristics” on page 16, flash memory technology has unique limitations that require protection. The IBM FlashSystem 820 storage system utilizes the techniques shown in Table 1-8 to address these.

Table 1-8 IBM FlashSystem 820 flash memory protection

Problem	Resolution
Limited write-erase cycles	Wear-leveling
Erasing timing and block sizes	Over-provisioning
Bit errors	ECC
Disturb errors (read/write/erase)	Voltage and timing adjustments
Block/plane/device failures	Block remapping RAID/2D Flash RAID Variable Stripe RAID

1.5.5 RAID rebuild process

All data protection techniques are designed to restore the storage system to a stable and consistent state. The main purpose of those techniques is to get the system into that state as fast as possible. Fast rebuild of the failed RAID component is one of the key methods to provide data storage reliability. The RAID rebuild process of traditional, HDD-based disk systems takes time and consumes the resources of the system (CPU, disk, and internal bandwidth). The IBM FlashSystem 820 is optimized for faster rebuild times of the failed flash modules, but the duration of the rebuild still depends on the state of the system and the overall level of the load. In general, the process of the FlashSystem RAID rebuild is:

- Rebuild process controlled by Active Management Controller (AMC)

- AMC issues rebuild commands to interface in 1 GB chunks
- Each interface issues multiple outstanding 4 KB rebuild commands to the RAID controllers
- Interface balances fairness with external I/O
- ▶ Once the RAID is rebuilt, the failed flash module is off-lined and ready for replacement
- ▶ Once the failed flashcard is replaced, the new flashcard is marked as the new spare

In our lab environment, the following flash module rebuild times were observed:

- ▶ 39 minutes for a 2 TB flash module while idle
- ▶ 1 hour 17 minutes for a 2 TB flash module while under heavy load (240 K input/output operations per second (IOPS) @ average response time < 0.5 ms)

Note: The rebuild times that we achieved are for reference only. The rebuild times that you experience might differ significantly.

Following are the top factors that affect FlashSystem RAID rebuild times:

- ▶ Overall load of the system (CPU and bandwidth)
- ▶ Pattern of the workload (reading is easier than writing)
- ▶ I/O skew factor (evenly distributed I/Os decrease the additional load to the module being rebuilt)
- ▶ Utilized capacity does not affect rebuild times in general

For more information, see Chapter 5, “Diagnostics, planned outages, and troubleshooting” on page 123.

1.5.6 Differences between IBM FlashSystem storage systems and SSD-based storage systems

Flash memory technologies appeared in the traditional storage systems some time ago. As such, they have a history of usage and implementation best practices. These technologies help to successfully address the challenge of increasing I/Os per second and the demand for lower response times in particular tasks. An implementation example is the IBM Easy Tier technology.

However, these technologies typically rely on flash technology in the format of FC, SAS, or SATA disks, placed in the same storage system as traditional spinning disks, and utilizing the same resources and data paths. This approach can abstract the advantages of flash technology behind the limitations of traditional disk storage systems.

IBM FlashSystem storage systems provide a hardware-only data path and realize all the potential of flash memory technology. These systems are different from traditional storage systems, both in the technology and usage approaches.

Differences in technology

An SSD device with a hard disk form factor has flash memory that is put into a carrier or tray. This carrier is inserted into an array like a hard disk drive. The speed of storage access is limited by the following technology because it adds latency and cannot keep pace with flash technology:

- ▶ Array controllers and software layers

- ▶ SAS controllers and shared bus
- ▶ Tiering and shared data path
- ▶ Form factor enclosure

IBM FlashSystem products are fast and efficient. The hardware-only data path has a minimum number of the software layers, which are firmware components mostly, as well as management software separated from the data processing. The only other family of products that have hardware-only access to flash technology are PCIe flash products intended to be installed into a dedicated server. With the appearance of the IBM FlashSystem storage systems, the benefits of the PCIe flash products to a single server can now be shared by many servers.

The benefits of the IBM FlashSystem storage systems technology are clearly evidenced in performance testing and comparison of the rebuild times of SSDs and flash modules.

For more information, see *IBM SAN Volume Controller and IBM FlashSystem 820: Best Practices and Performance Capabilities*, REDP-5027.

Differences in usage

Existing hybrid (SSD+HDD) storage systems are well suited for storing the data and providing advanced functions such as snapshots and remote mirroring. IBM FlashSystem storage systems are intended to be implemented to provide the lowest I/O latency in most critical business applications.

Extreme performance

FlashSystem products increase application performance as much as 10x faster than other storage solutions. When compared to equivalent disk systems, IBM FlashStorage storage systems can deliver capacity in a single 1U rack, and are 19 times more cost efficient. These solutions include the latest in industry-standard, solid-state flash memory technology, including eMLC flash technology and SLC flash technology. Data is moved through the system as quickly as possible, with no bottlenecks.

IBM MicroLatency

FlashSystem products deliver extremely fast response time to accelerate critical applications. MicroLatency — that is, roughly 100-microsecond access time, enables faster decision making by facilitating an extreme-performance data path to accelerate critical applications and help users achieve a true competitive advantage. Dynamic random access memory (DRAM) on each module helps enable fast writes at 25 microseconds. Purpose-driven, highly parallel design maximizes host CPU efficiency and productivity.

MacroEfficiency

FlashSystem products can help consolidate hardware and software and deployment speed, and provide power and cooling savings. Benefits include:

- ▶ A 1U form factor, which has a minimal footprint for optimum ROI
- ▶ Two dual-port 8 Gb Fibre Channel controllers or dual-port 40 GB QDR InfiniBand controllers
- ▶ 350 watt or less power draw
- ▶ Hot swap flash modules to enable uninterrupted operations
- ▶ Up to a petabyte of FlashSystem storage can be placed in a single rack, on a single floor tile
- ▶ IBM FlashSystem storage systems offer one of industry's best IOPS per watt ratio to maximize energy savings

- ▶ IBM FlashSystem storage systems use hexagonal ventilation holes, a part of IBM Calibrated Vectors Cooling™ technology. Hexagonal holes can be grouped more densely than round holes, providing more efficient airflow through the system.

Enterprise reliability

IBM FlashSystem products have durable and reliable designs that use enterprise class flash and patented data protection technology. IBM FlashSystem devices are designed for cost-effective, high storage performance that is used to accelerate critical business applications. IBM FlashSystem devices feature Variable Stripe RAID, 2D Flash RAID, Active Spare support, ECC at the chip level, and other reliability technologies.

Two-dimensional Flash RAID eliminates single points of failure and provides enhanced system-level reliability. Patented Variable Stripe RAID technology helps reduce business interruptions and prevent chip failures to enhance the two-dimensional protection mechanism. It also maintains performance capacity levels. Hot-swappable flash modules and redundant components with built-in battery backup help boost data availability and IT productivity. An available integrated spare flashcard limits downtime.

1.5.7 Usage considerations for IBM FlashSystem storage systems

The history of the storage systems has demonstrated the following considerations regarding the industry's requirements for storage:

- ▶ Persistence
- ▶ Capacity
- ▶ Bandwidth
- ▶ IOPS

Although IOPS and bandwidth are important characteristics, more and more low latency is becoming the driving factor influencing storage acquisition. Increasing CPU speed at the server level allows the server to process data faster, and server CPUs spend more time idly waiting for the data from the storage systems. Thus, the service time of each request to storage plays a key role in keeping CPUs busy and making the most of investments in CPU power.

Figure 1-19 illustrates how overall request processing time depends now mostly on the disk latency.

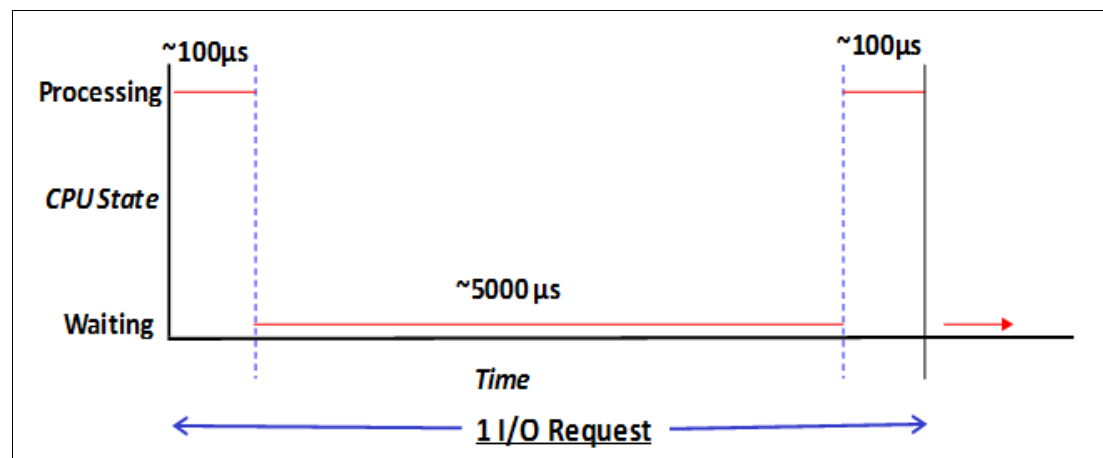


Figure 1-19 Overall I/O request time

Decreasing the I/O latency time increases the speed of the request processing and the overall speed of the applications used. This example, typical for the traditional storage systems, includes three stages of disk I/O request servicing, with the following durations:

- ▶ I/O request issue ~ 100 usec¹⁵
- ▶ Wait for I/O to be serviced ~ 5000 usec
- ▶ Process the I/O ~100 usec

In the example, the average total time of I/O processing is roughly 5200 microseconds. Such I/O wait time results in CPU utilization of approximately 4% (200 microseconds/5200 microseconds). Hybrid (SSD+HDD) storage systems can provide marginal improvement, decreasing the I/O wait time to 1000 - 1500 usec.

IBM FlashSystem storage systems proved to provide latencies closer to 200 - 400 microseconds, decreasing overall I/O processing time to 400 - 600 usec and increasing CPU utilization up to 50%.

There are many areas where IBM FlashSystem storage systems can provide benefits and improvements:

- ▶ Online analytical processing (OLAP)
- ▶ OLTP databases
- ▶ VDI infrastructures
- ▶ High-performance computing (HPC) and computational applications
- ▶ Video production/streaming media
- ▶ Using IBM SAN Volume Controller with IBM FlashSystem storage systems

Online analytical processing

Online analytical processing (OLAP) queries are typically complex and process large volumes of data from multiple sources. Many servers have adequate RAM and processor power to process massive amounts of data (frequently referred to as *big data*). However, the I/O that is required for reading data from storage for processing in the OLAP database server can frequently reduce performance. Delays come primarily from batch data loads and performance issues due to handling heavy complex queries that use I/O resources.

IBM FlashSystem storage solutions can help to address these challenges in the following ways:

- ▶ Dramatically boosting the performance of OLAP workloads with distributed scale-out architecture, providing almost linear and virtually unlimited performance and capacity scalability
- ▶ Significantly improving response time for better and timely decision making

VDI infrastructures

FlashSystems can significantly improve the performance of VDI infrastructures by providing an ultra-fast storage platform to perform such tasks as connection brokering and rapid cloning of desktops:

- ▶ Connection brokering is a simple transaction-based process that does not have a high amount of data but requires a rapid response time.
- ▶ The rapid cloning of virtual desktops for a new requirement is a heavily I/O intensive task, which requires a high amount of data transfer over ultra-fast response time.
- ▶ When using VDI, there are usually peak times when the system is pressed to perform. An example is the start of day log-on. FlashSystems' low latency response times allow VDI

¹⁵ Microseconds, 1 msec=1000 usec

environments to process large amounts of data, such as roaming profile uploads from multiple Windows based virtual desktops.

- ▶ If multiple virtual desktops require rebooting (as in recovering from a VDI pool outage), FlashSystem throughput and low latency can improve this process.

IBM FlashSystem storage systems for HPC and computational applications

High-performance flash storage systems for Intelligent Clusters were announced 04/30/2013 and include rack mount FlashSystem storage products. IBM FlashSystem 720 and FlashSystem 820 storage can be used in this solution for enterprise deployment. Variable Stripe RAID technology included in the IBM flash storage capability enhances system resiliency without sacrificing performance or usable capacity.

For more information, see the IBM FlashSystem storage systems at the IBM Intelligent Cluster™ website:

<http://www.ibm.com/systems/x/hardware/largescale/cluster/index.html>

Using IBM SAN Volume Controller with IBM FlashSystem storage systems

You can use SVC to add advanced storage functionality and high availability to the extreme performance of FlashSystem storage. SVC products have a minimal affect on latency and add the following features:

- ▶ Thin provisioning to allocate storage “just-in-time”
- ▶ Improved utilization to harvest all SAN capacity
- ▶ Disaster avoidance with location-proof data availability
- ▶ Easy Tier for storage efficiency
- ▶ IBM FlashCopy for point-in-time copies
- ▶ Mirroring/copy services for data replication and protection
- ▶ Real-time Compression to place up to five times more data in the same physical space

For more information, see Chapter 2, “Usage considerations and scenarios” on page 41, and Chapter 3, “Planning and installation of the IBM FlashSystem 820” on page 63.



Usage considerations and scenarios

This chapter covers design and configuration considerations for usage scenarios when using IBM FlashSystem storage systems with SAN Volume Controller (SVC). The chapter begins with detailed usage considerations, including port assignment, port masking, and host connectivity. Additionally, common usage scenarios are described. Throughout, recommendations, and best practices are identified, where applicable. The chapter concludes with a brief description of hardware failure protection capabilities and considerations.

2.1 Usage considerations

The IBM FlashSystem 820 storage system is a high performance storage device capable of sustaining extremely high throughput and low latency across its four 8 Gbps Fibre Channel adapter ports¹. In order to maximize the performance that you can achieve when deploying the FlashSystem 820 with SAN Volume Controller, careful consideration should be given to the assignment and usage of the Fibre Channel host bus adapter (HBA) ports on the SVC.

Specifically, storage area network (SAN) switch zoning, coupled with port masking (a new feature introduced in SVC Storage Software version 7.1) can be used for traffic isolation for various SVC functions, reducing congestion and improving latency.

This section provides guidance and best practices recommendations about using these capabilities. For detailed instructions on configuring port masking in SVC, see 4.2, “Port masking and SAN zoning configuration” on page 108.

2.1.1 Port assignment scenarios and related considerations

When using IBM FlashSystem 820 behind SVC, there are a number of factors to consider regarding the use of the Fibre Channel ports on the SVC. Such considerations include:

- ▶ The number of SVC HBAs and ports
- ▶ The number of FlashSystem storage systems
- ▶ The number, if any, of other virtualized storage
- ▶ The number of hosts
- ▶ Whether or not replication will be used
- ▶ Whether or not stretched cluster will be used

Suggested port assignment for a single SVC node with dual 4-port HBA cards

The primary goals of a port assignment scheme with SVC and IBM FlashSystem 820 are to maintain the lowest latency for the FlashSystem and to reduce congestion of the SVC Fibre Channel ports. As discussed in 1.2.1, “Hardware updates for SVC” on page 8, feature code AHA7 offers the ability to add a second HBA card to the 2145-CG8 SVC node. As such, the suggested port assignment for this hardware configuration is shown in Figure 2-1 on page 43.

¹ The IBM FlashSystem 820 storage system can be configured with either 8 Gbps Fibre Channel ports or 40 Gbps quadruple data rate (QDR) InfiniBand ports. Our environment is configured with Fibre Channel ports.

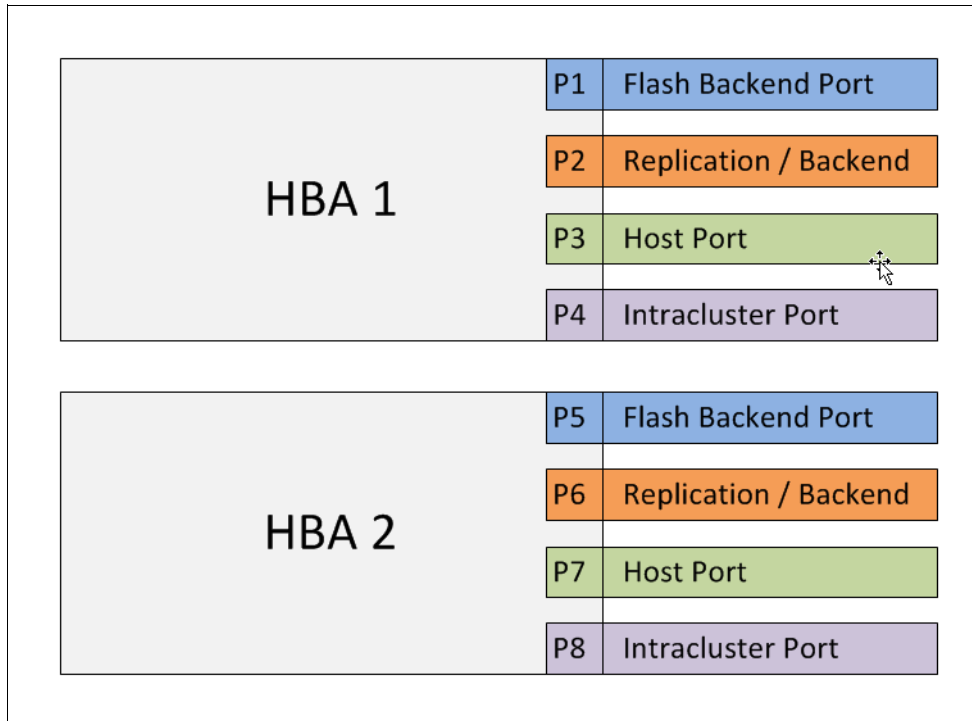


Figure 2-1 Suggested port assignment for a single SVC node with two HBA cards

Deploying the FlashSystem 820 with SVC nodes containing dual, quad-port HBAs allows you to completely realize the performance potential of the FlashSystem by providing the ability to completely segregate traffic types, isolating intracuster communications, back-end array communications, host communications, and replication traffic.

For this reason, deploying FlashSystem storage systems with SVC nodes containing dual HBAs is the recommended configuration. For more information about port assignment and port masking with such deployments, see “Related considerations” on page 44.

Note: At the time of writing, for SVC nodes with dual, quad-port HBA cards, ports 7 and 8 cannot be used for *any* back-end traffic.

Suggested port assignment for a single SVC node with single 4-port HBA cards

For deployments where each SVC node has a single, four-port HBA card, the suggested port assignment is shown in Figure 2-2 on page 44.

Note: Any SVC node requires at least two Fibre Channel ports for intracuster communication.

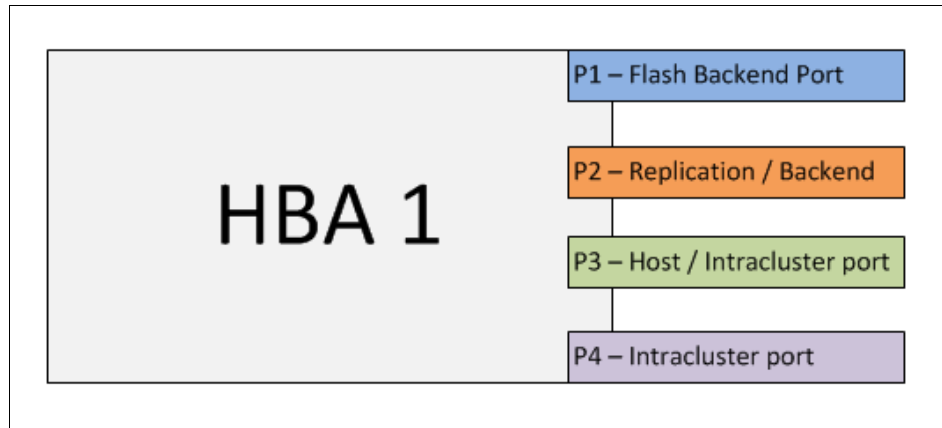


Figure 2-2 Suggested port assignment for a single SVC node with one HBA card

Deploying the FlashSystem 820 with SVC nodes containing single, quad-port HBAs, while allowing you to achieve traffic separation similar to dual HBA configurations, does have some considerations, as described in “Related considerations”.

Related considerations

Whether deploying your FlashSystem with single or dual card SVC nodes, the following points should be considered:

- ▶ Intracluster communication isolation, allowing you to improve active/active node communications, including caching between nodes in the cluster. A recommendation is to use port 4 (along with port 8 in dual HBA scenarios) and port 3 shared with host communication for intracluster (local) traffic. Sharing port 3 with host communication is only required for a single HBA node because there is a minimum requirement for two intracluster communication ports per node.
- ▶ Adequate and balanced port and bandwidth allocation for FlashSystem storage systems, creating a balanced and symmetric design. The best practice recommendation is to dedicate SVC Fibre Channel ports to FlashSystem connectivity. Specifically, dedicate port 1 for single HBA SVC node deployments, and ports 1 and 5 for dual HBA SVC node deployments.
- ▶ Host traffic isolation and balancing, guaranteeing dedicated host bandwidth, ensuring low latency, and isolating host maintenance activities from storage maintenance activities. A recommendation is to dedicate SVC Fibre Channel ports to host connectivity. Specifically, use port 3 (which is also shared for intracluster communication) for single HBA SVC node deployments, and ports 3 and 7 for dual HBA SVC node deployments.
- ▶ Replication traffic isolation, where possible, guaranteeing dedicated bandwidth for data replication, isolating back-end repetitive traffic from front-end and back-end host traffic, and ensuring that host I/O latency is unaffected by replication communication link latency. Depending on whether your environment will include replication traffic, traffic to other back-end storage, or both, the following recommendations might apply:
 - Dedicate ports 2 and 6 for replication traffic when there are no other back-end storage systems
 - Dedicate ports 2 and 6 for traffic to other back-end storage when there is no replication
 - If both conditions exist, combine replication traffic and traffic to other back-end storage on ports 2 and 6

Note: In scenarios where both replication traffic and traffic to other back-end storage exists, extreme port utilization could lead to port saturation or increased latency. In this scenario, consider dedicating port 2 for replication and port 6 for other back-end traffic.

Note: Because the IBM FlashSystem 820 has only four Fibre Channel ports, there is no need to assign more than four dedicated SVC Fibre Channel ports per I/O group.

As shown in Figure 2-3, the recommended port assignment creates symmetry and balance across the physical ports on the SVC two node cluster because there are four dedicated ports for use between the SVC and the IBM FlashSystem 820.

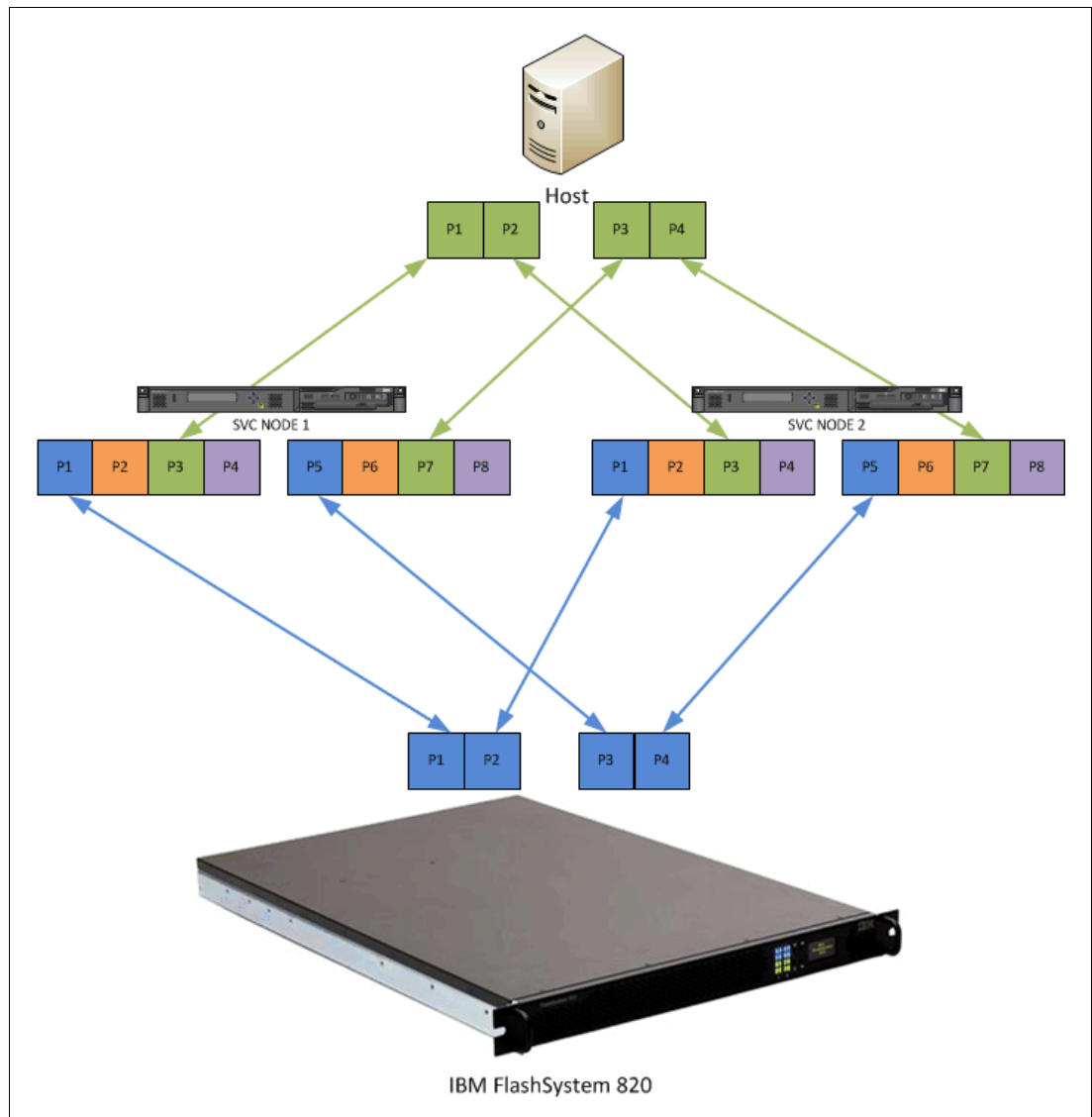


Figure 2-3 An example of a balanced dedicated port solution

Further, this provides dedicated host bandwidth using two ports per node for a total of four 8 GB Fibre Channel ports, equaling 32 GB of aggregate bandwidth from host to SVC. This also has a physical port ratio of 1:1, matching the IBM FlashSystem to SVC back-end port

allocation, and ensuring that all traffic has the same number of connections and is symmetrical and balanced.

Special considerations when using FlashSystem storage systems with single HBA SVC nodes

When using four-port SVC nodes with a FlashSystem storage system and additional back-end storage, port assignment can become more challenging because there are five types of traffic that must be distributed across four ports. Combining replication traffic and traffic to the other back-end array might saturate port 2. In this scenario, ensure that FlashSystem traffic isolation is maintained on port 1, and consider the following modifications to port assignments:

- ▶ Combine replication traffic with back-end traffic on port 2 *and* combine replication traffic with host and intracluster traffic on port 3.
- ▶ Further, if host, intracluster, and replication traffic saturate port 3, combine host and intracluster traffic with replication traffic on port 3 *and* combine host traffic with intracluster traffic on port 4.

2.1.2 SAN Volume Controller Stretched Cluster

SVC Stretched Cluster configurations are supported by IBM FlashSystem storage systems, whether you are using nodes with single or dual HBAs. When using FlashSystem storage systems in SVC Stretched Cluster environments, follow the guidance in the following Redbooks publications:

- ▶ *IBM SAN and SVC Stretched Cluster and VMware Solution Implementation*, SG24-8072
- ▶ *IBM SAN Volume Controller Stretched Cluster with PowerVM and PowerHA*, SG24-8142

2.1.3 Port masking on SVC with IBM FlashSystem 820

SVC Storage Software version 7.1 introduces port masking, a new feature that enables better control of the SVC node ports. Host port masking is supported in earlier SVC software versions. In those versions, host port masking provides the ability to define which SVC node ports were used to communicate with hosts.

The enhanced port masking in version 7.1 provides the ability to restrict intracluster communication and replication communication to specific ports, ensuring that these traffic types only occur on the desired ports. This eliminates the possibility of host or back-end port congestion due to intracluster communication or replication communication.

Note: An SVC node will attempt to communicate with other SVC nodes over any available path. Port masking, when enabled, ensures that this will not occur on that port.

Note: To utilize the port masking feature, use the `chsystem -localfcportmask` or `-partnerfcportmask` commands.

These features that are in use with storage area network (SAN) zoning and the physical port assignment that is detailed above, provide greater control and enable less congestion and better usage of the SVC ports.

Note: When using port masking with SVC, follow this configuration order:

1. Configure intracluster port masking.
2. Configure replication port masking (if using replication).
3. Configure SAN zones for intracluster communication.
4. Configure SAN zones for replication communication (if using replication).
5. Configure SAN zones for all back-end storage communication.
6. Configure SAN zones for host communication.

For more information about how to configure SVC port masking, see 4.2, “Port masking and SAN zoning configuration” on page 108.

2.1.4 Host multipathing

When using the IBM FlashSystem 820 with SVC, follow the existing recommended best practices regarding the attachment of specific host operating systems, as described in *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933, or later.

It is recommended that all hosts use round robin with the SAN Volume Controller, where supported. It is also recommended that all hosts have the latest version of the multipathing driver (SDD is recommended whenever possible) installed because this provides the best performance, most stable platform, and a more balanced pathing solution.

Important: If there are hosts using automated round robin *and* hosts using fixed path or most recently used multipathing, the fixed path and most recently used component must be manually and carefully managed to ensure that selected target storage ports are not overloaded and that the I/O workload is balanced across the target storage ports.

The SVC multipathing software (SDD) provides the ability to use round robin, and also utilizes the path with the smallest I/O queue over other paths with higher queue lengths.

2.1.5 Considerations regarding number of FlashSystem per I/O group

This section addresses considerations regarding the number of physical FlashSystem storage systems to use in specific usage scenarios with SVC.

- When FlashSystem storage systems are the only storage being used with SVC, the recommended number of FlashSystem storage systems per SVC I/O group is one.

Note: If volume mirroring is being utilized, at least one pair of FlashSystem storage systems are required per I/O group. For more information, see 2.2.2, “FlashSystem with SAN Volume Controller volume mirroring” on page 51.

- When FlashSystem storage systems are being used with existing spindle-based SVC storage pools as an Easy Tier storage device to improve the performance, the number of FlashSystem storage systems required is independent of the number of I/O groups. Rather, it depends on the amount of Easy Tier capacity required.

Note: The number of FlashSystem storage systems to use for SVC Easy Tier capacity depends on how many existing storage pools are in the environment. The maximum recommended number of MDisk to be allocated from a single FlashSystem is 16. As such, if you have more than 16 storage pools that you want to use with Easy Tier, you will require more than one FlashSystem device. For more information about using Easy Tier, see *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933, or later.

- ▶ When FlashSystem storage systems are being used with SVC to provide both Easy Tier capacity *and* to provide *all flash* capacity for specific volumes, the number of FlashSystem storage systems that are required depends on the specific capacity and performance requirements of the environment. This number can vary from one FlashSystem storage system per cluster up to multiple FlashSystem storage systems per I/O group. When determining your specific requirements, ensure that you follow the recommendations outlined in *IBM SAN Volume Controller and IBM FlashSystem 820: Best Practice and Performance Capabilities*, REDP-5027.

Extent size consideration

The general recommendation is to use an extent size of 1024 MB where possible. However, if you have existing pools that are not this size, and you need to be able to migrate volumes seamlessly between these storage pools and the FlashSystem storage pools, configure the extent size of your FlashSystem storage pools to match the existing pools, as required.

See 4.3.4, “Storage pool extent size” on page 116 for more information.

2.2 Usage scenarios

In this section, we describe the four most common scenarios for using IBM FlashSystem storage systems with SAN Volume Controller. For each scenario, we provide a description of the general design purpose and an overview of the architecture. Further, we describe for each any unique planning or implementation recommendations or best practices. Specific port assignment, masking, or MDisk configuration recommendations, if any, are included. For general port assignment and masking recommendations, refer to 2.1, “Usage considerations” on page 42.

For general MDisk and storage pool configuration information, refer to 4.3, “SAN Volume Controller MDisk configuration” on page 111.

2.2.1 All FlashSystem usage scenario

An all FlashSystem SAN Volume Controller solution refers to an SVC and FlashSystem deployment in which the FlashSystem storage system is the *only* storage device being virtualized by SVC. This is the type of deployment that is configured for the examples referenced in Chapter 4, “Configuration and administration” on page 77.

Our example environment consisted of a two-node SVC cluster and two IBM FlashSystem 820 devices. The SVC nodes were model CG8 with feature code AHA7 (additional HBA card) and RPQ 8S1296 (additional six-core CPU and 24 GB of RAM). This specific solution is well suited to provide low latency and high throughput to applications such as transactional databases, analytical databases, virtual desktop environments, cloud environments, and high-performance computing (HPC) environments.

This solution is tailored to deal with the I/O bandwidth and low latency that is required by these types of applications, and provides advanced features such as FlashCopy, Metro and Global Mirror, and volume mirroring. A high-level layout is shown in Figure 2-4.

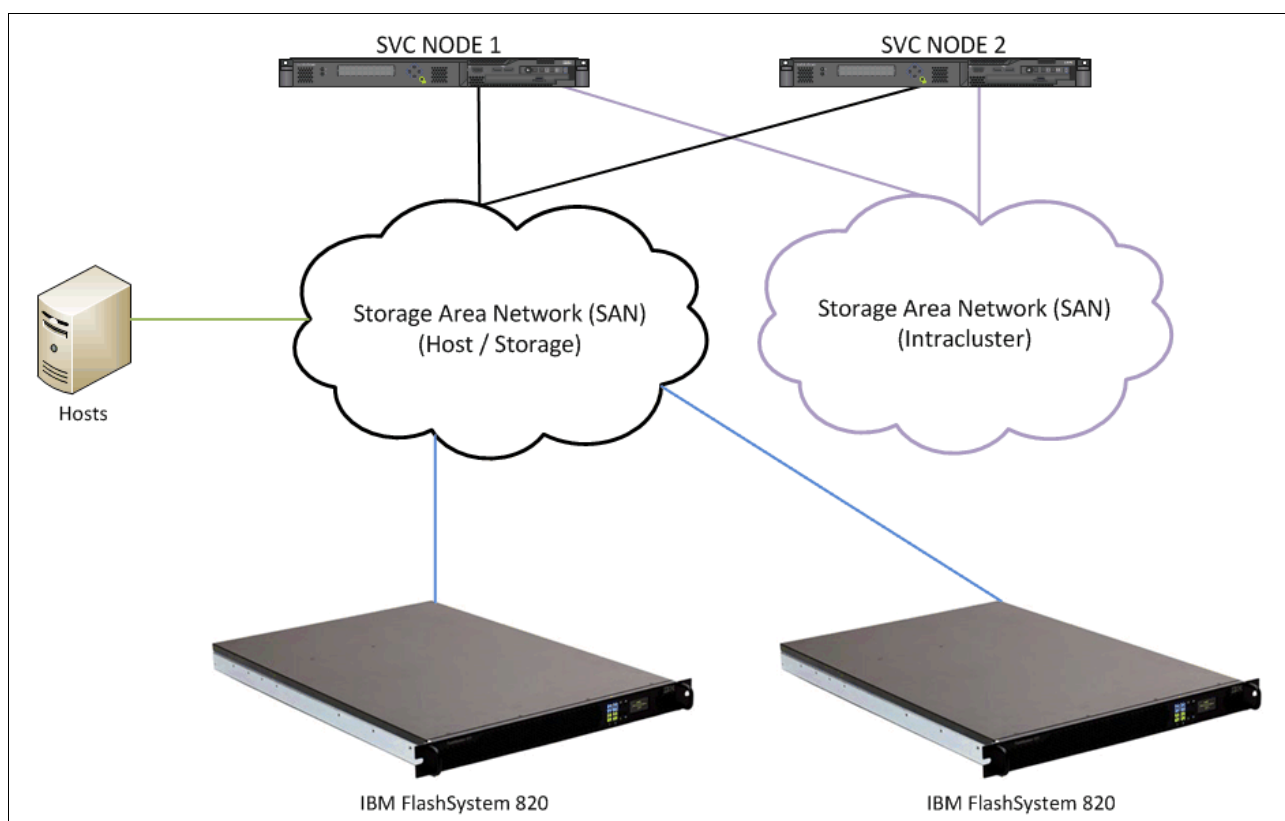


Figure 2-4 A high-level overview of the all FlashSystem and SVC solution

General considerations

This section details some design and configuration considerations for an all IBM FlashSystem 820 and SVC solution. This configuration is designed for simplicity and performance.

In this scenario, SVC nodes are dedicated for the IBM FlashSystem storage systems, allowing more flexibility for port assignment. For details about port assignment, see 2.1.1, “Port assignment scenarios and related considerations” on page 42. Specifically, this allows ports 2 and 6 to be flexibly assigned as:

- ▶ Additional flash back-end ports
- ▶ Additional host ports
- ▶ Dedicated replication ports

In this scenario, ports 4 and 8 should be used for dedicated intracuster communication, consistent with the port assignment recommendations described earlier in 2.1.1, “Port assignment scenarios and related considerations” on page 42.

Note: A best practice is to use redundant, dedicated physical fabrics for intracuster connectivity. This is to ensure that there is no congestion of available throughput on either the IBM FlashSystem back-end Fibre Channel ports or the intracuster ports. Alternatively, virtualized fabrics or VSANs can be used for traffic isolation.

Figure 2-5 shows an example of using SVC ports 2, 6, 3, and 7 for host traffic.

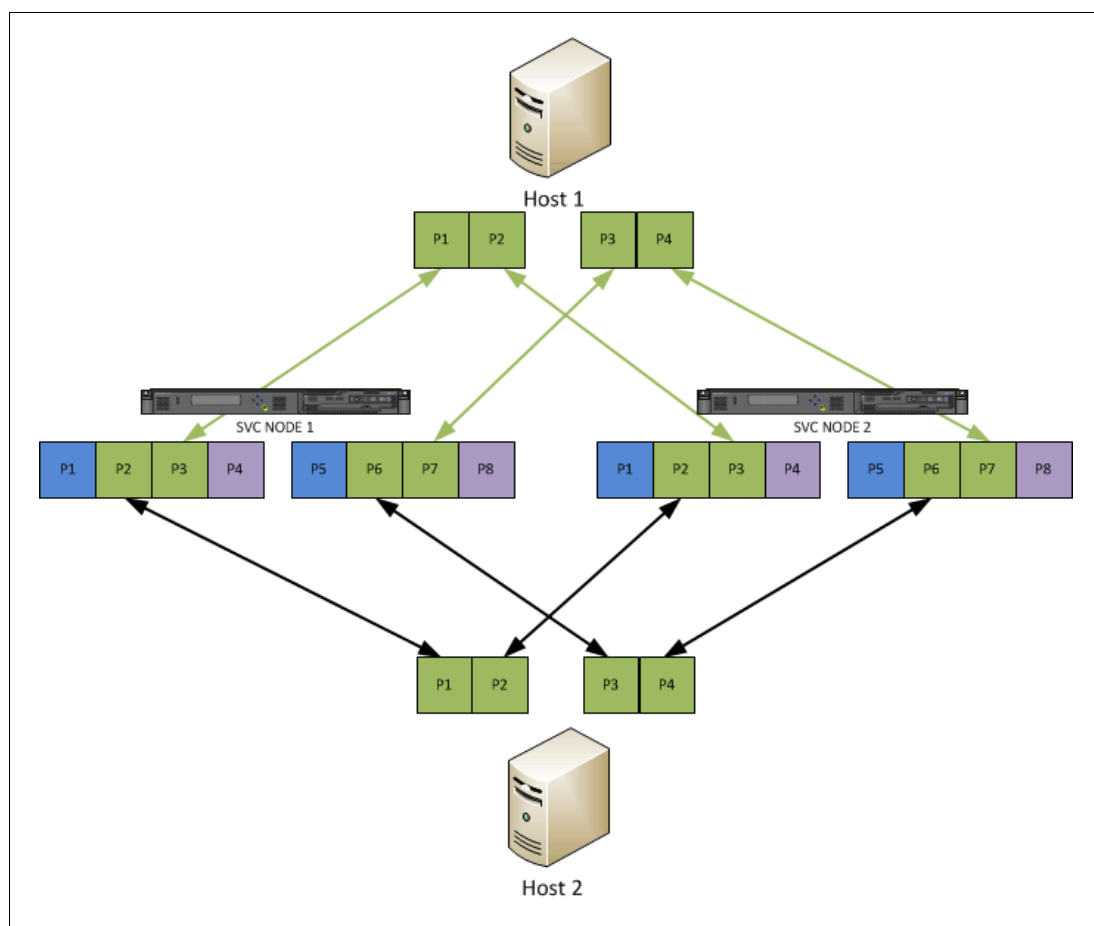


Figure 2-5 Using SVC ports 2, 6, 3, and 7 for host traffic

Ensure that port masking is correctly configured to isolate intracluster traffic from all other traffic. Table 2-1 details the port assignment to use when configuring port masking.

Table 2-1 Port allocation for port masking without replication

HBA	Intracluster communication port	Host communication port	FlashSystem back-end ports
HBA 1	4	3	1, 2
HBA 2	8	7	5, 6

In this sort of implementation, the nature and importance of the data to be stored on the system should be considered. If the data is business critical, the solution should include a design for disaster recovery. Examples are using Metro Mirror or Global Mirror to a second site.

While the FlashSystem 820 storage system has built-in hardware redundancy, it should be treated as any other disk subsystem regarding the protection of critical data. An alternative solution that can address data protection or high availability requirements with FlashSystem

storage systems and SVC is the use of volume mirroring, explained in 2.2.2, “FlashSystem with SAN Volume Controller volume mirroring” on page 51.

Note: The IBM FlashSystem 820 currently has four ports. Assigning four back-end ports per SVC node has the benefit of the IBM FlashSystem 820 not losing any performance during a node failure or code upgrade.

More information about how to set up and configure replication can be found in *IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services*, SG24-7574.

For information about the expected performance of this scenario, see the following IBM Redpaper™ publication:

IBM SAN Volume Controller and IBM FlashSystem 820: Best Practice and Performance Capabilities, REDP-5027

2.2.2 FlashSystem with SAN Volume Controller volume mirroring

In this section, we describe IBM FlashSystem with SVC volume mirroring.

Volume mirroring basics

SVC volume mirroring is a feature that allows the creation of one volume with two copies of its extents. The two data copies, if placed in different storage pools, allow volume mirroring to eliminate impact to volume availability if one or more MDisk, or the entire storage pool fails.

When designing for a highly available SVC and FlashSystem deployment, it is recommended to have each storage pool built with MDisk coming from separate back-end storage subsystems. This allows the mirrored volumes to have each copy in a different storage pool. In this manner, volume mirroring can provide protection against planned or unplanned storage controller outages, as the volume continues to be available for read/write operations from hosts with the surviving copy.

If one of the mirrored volume copies becomes unavailable, updates to the volume are logged by SVC, allowing for resynchronization of the volume copies when the mirror is reestablished. The resynchronization between both copies is incremental and is started by the SAN Volume Controller automatically.

Note: The port allocation and masking for this scenario should be configured as in 2.1, “Usage considerations” on page 42.

In the SAN Volume Controller software stack, volume mirroring operates below cache and copy services. Therefore, FlashCopy, Metro Mirror, and Global Mirror have no awareness that a volume is mirrored. All operations that can be run on non-mirrored volumes can also be run on mirrored volumes. These operations include migration and expanding/shrinking. Similarly, the volume mirroring feature is above the thin provisioning, Real-time Compression, and Easy Tier services in the software stack. Each of the two copies of the volume can utilize different combinations of these characteristics. For example, a volume can be mirrored between a thin-provisioned copy residing in a storage pool with Easy Tier enabled, and its second copy can be fully provisioned and residing in a second storage pool with Easy Tier disabled.

As with non-mirrored volumes, each mirrored volume is owned by the preferred node within the I/O group. The preferred node performs all write operations to the back-end disks for both copies of the volume. Read operations can be serviced from either node in the I/O group that owns that volume. However, all read operations are serviced only from the primary copy of the volume mirror.

Note: The preferred node is set at the *volume creation* time, and can only be changed if the volume is moved across I/O groups, at which time a new preferred node from the target I/O group can be chosen. When preferred node placement is a requirement, careful planning should include:

- ▶ Balanced approach, evenly distributing the expected workload between the nodes in the I/O group
- ▶ Aligning of hosts with nodes, such that traffic between host and node traverses a single fabric

Example 2-1 shows how to define the preferred node to a mirrored volume at its creation time. In this example, a mirrored volume of 10 GB was created, selecting node 1 from I/O group 0 as its preferred node.

Example 2-1 Preferred node assignment to a mirrored volume at creation time

```
IBM_2145:SVC_FLASHSYSTEM:superuser>mkvdisk -mdiskgrp FLASHSYSTEM_A:FLASHSYSTEM_B -size 10
-unit gb -copies 2 -I/O group 0 -node 1 -name IBMX3650_1
Virtual Disk, id [1], successfully created

IBM_2145:SVC_FLASHSYSTEM:superuser>lsvdisk 1
id 1
name IBMX3650_1
I/O_group_id 0
I/O_group_name io_grp0
... [OUTPUT TRUNCATED]
preferred_node_id 1
... [OUTPUT TRUNCATED]
```

Figure 2-6 on page 53 shows how to set the preferred node to a volume at the time of its creation using the graphical user interface (GUI). Click **Advanced** to gain access to the advanced setup configuration for that volume before it is created.

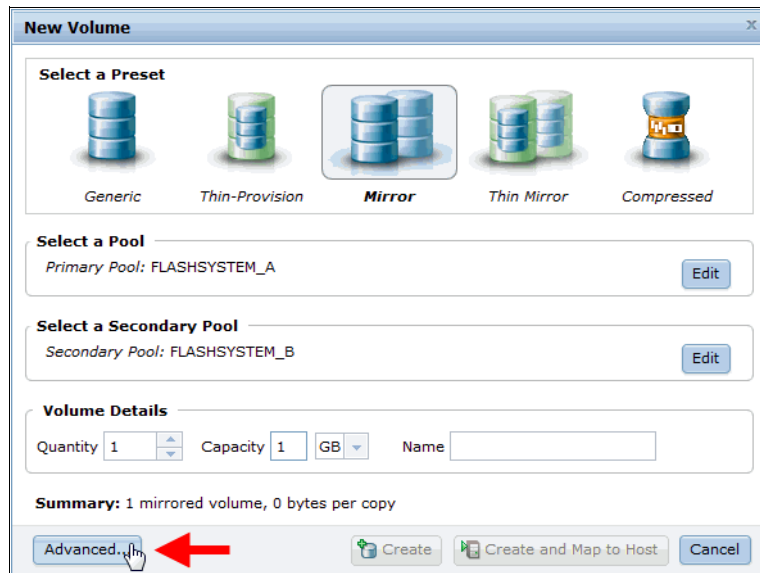


Figure 2-6 Creating a mirrored volume with a preferred node with the GUI

The Advanced Settings dialog is shown in Figure 2-7.

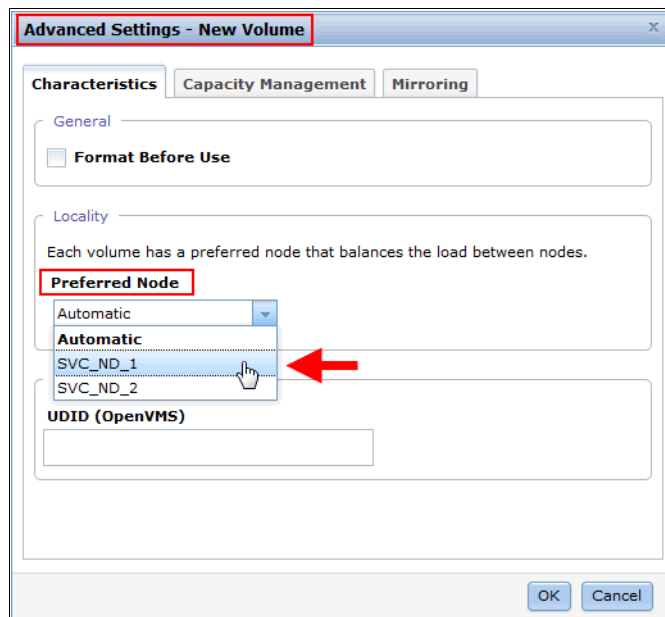


Figure 2-7 Advanced Settings dialog - selecting the preferred node for the volume

You can select the desired node to become the preferred for the volume being created. If not chosen, **Automatic** placement will be used, and the result is a round-robin balance between the two nodes of the chosen I/O group for every volume assigned to it. Click **OK** to return to the New Volume dialog and complete the volume creation.

Volume mirroring data flow

It is important to understand the volume mirroring data flow to make better decisions about the configuration to use to obtain the best results.

Read I/O operations data flow with a mirrored volume

For the *read* operations, volume mirroring implements an algorithm with one copy that is designated as the *primary* for all read operations. SVC reads the data from the primary copy and does not automatically distribute the read requests across both copies. The first copy that is created for a mirrored volume becomes the primary by default. You can change this setting by using the **chvdisk** CLI command, as shown in Example 2-2. In this example, Copy 0 was previously the primary, and we changed the primary to Copy 1.

Example 2-2 Use of the chvdisk command to change the primary copy of a mirrored volume

```
IBM_2145:SVC_FLASHSYSTEM:superuser>lsvdisk 2
id 2
name IBMX3650_2
... [OUTPUT TRUNCATED]

copy_id 0
status online
sync yes
primary yes
... [OUTPUT TRUNCATED]

copy_id 1
status online
sync yes
primary no
... [OUTPUT TRUNCATED]

IBM_2145:SVC_FLASHSYSTEM:superuser>chvdisk -primary 1 2

IBM_2145:SVC_FLASHSYSTEM:superuser>lsvdisk 2
id 2
name IBMX3650_2
... [OUTPUT TRUNCATED]

copy_id 0
status online
sync yes
primary no
... [OUTPUT TRUNCATED]

copy_id 1
status online
sync yes
primary yes
... [OUTPUT TRUNCATED]
```

The primary copy status can also be checked and changed using the SVC GUI. Figure 2-8 on page 55 shows the information about a mirrored volume in the Pools → Volumes by Pool view of the GUI, where its primary copy is denoted with an asterisk after the copy ID.

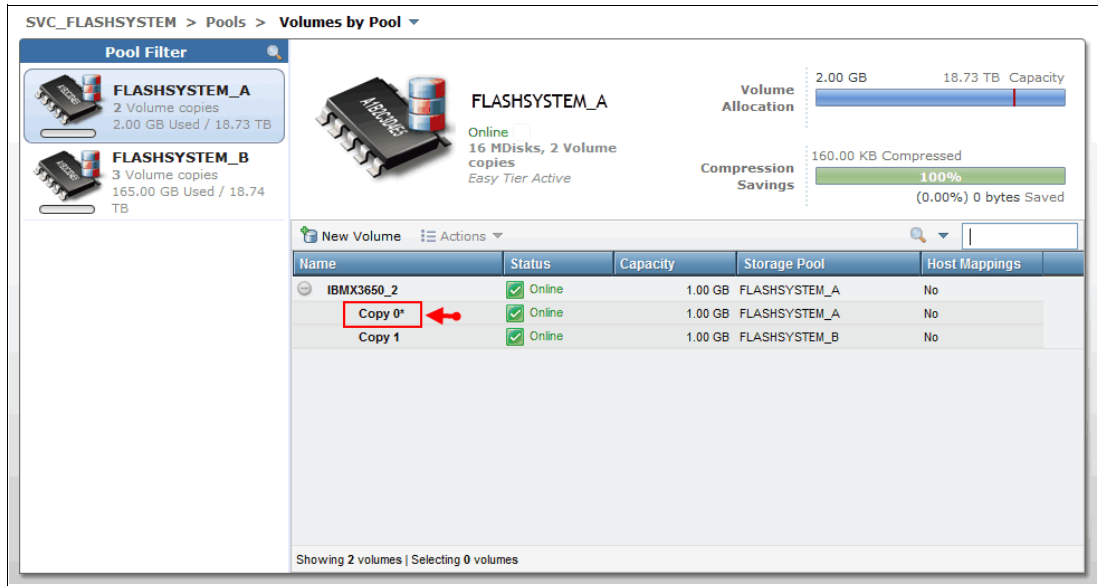


Figure 2-8 Checking the primary copy of a mirrored volume in the SVC GUI

In order to change the primary status to the other copy, right-click that copy and choose **Make Primary** in the pop-up menu, as shown in Figure 2-9.

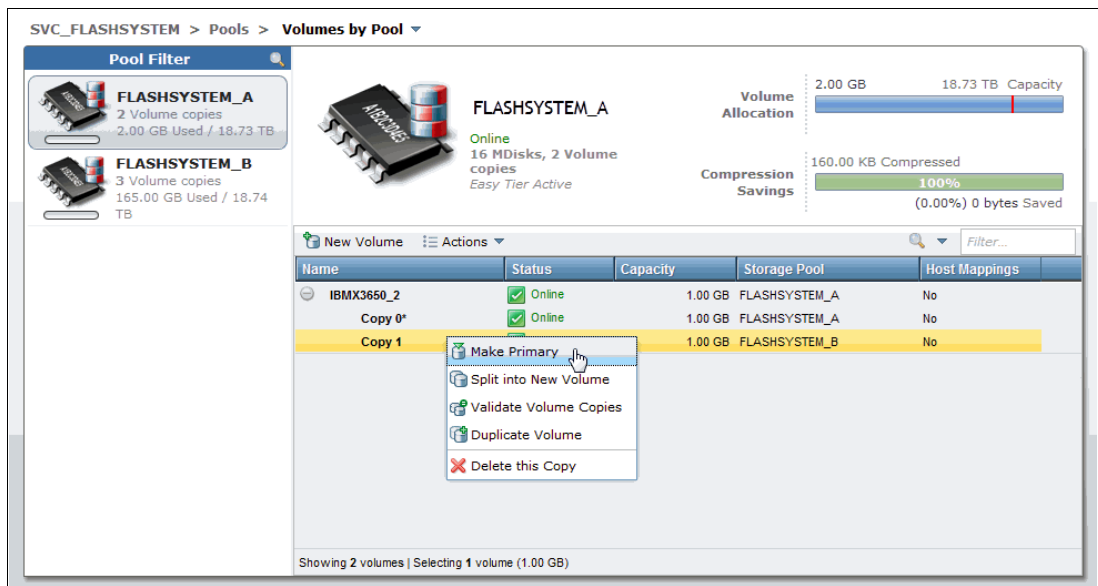


Figure 2-9 Changing the primary status of a volume copy using the SVC GUI

After the command completes, the new status will appear, as shown in Figure 2-10.

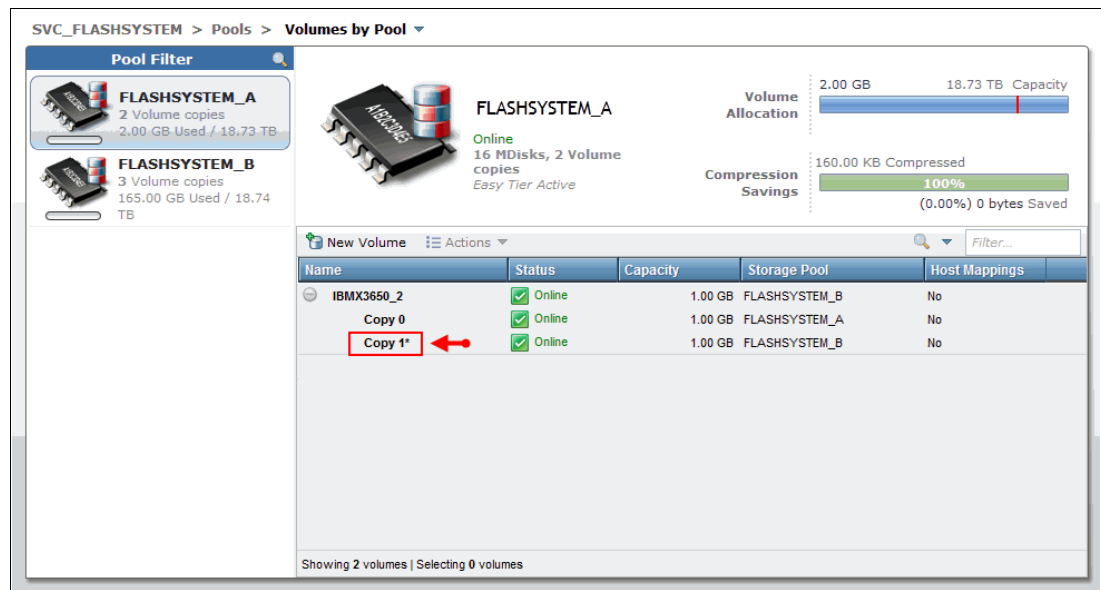


Figure 2-10 Checking the primary copy status of a mirrored volume in the SVC GUI

Write I/O operations data flow with a mirrored volume

For write I/O operations to a mirrored volume, the SVC *preferred node* definition, together with the multipathing driver on the host, are used to determine the preferred path. The host routes the I/Os via the preferred path, and the corresponding node is responsible to further destage written data from cache to both volume copies. Figure 2-11 shows the data flow for write I/O processing when using volume mirroring.

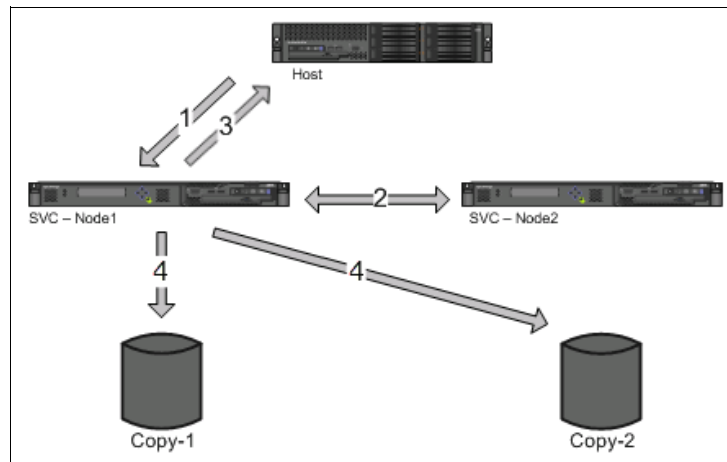


Figure 2-11 Data flow for write I/O processing in a mirrored volume in SVC

All the writes are sent by the host to the preferred node for each volume (1); then, the data is mirrored to the cache of the partner node in the I/O group (2) and then acknowledgement of the write operation is sent to the host (3). The preferred node then destages the written data to the two volume copies (4).

Depending on the **-mirrorwritepriority** flag that is used for each mirrored volume definition, the data is either synchronously destaged by the preferred node to both copies (**-mirrorwritepriority redundancy**), or the data is first destaged to the fastest copy and then

mirrored to the other copy asynchronously (**-mirrorwritepriority latency**). If not specified during the mirrored volume creation, latency is assumed by default. This choice is not available in the GUI, but can be changed using the CLI afterwards, if needed.

These options can directly affect the write latency from the cache to the mirrored copies, especially if they are separated by longer distances, as in a stretched cluster architecture, or when one of the volume copies resides in a lower-performance storage system. Example 2-3 shows how to check the current status of the **-mirrorwritepriority** flag for a mirrored volume using the CLI, and then change it.

*Example 2-3 Checking the **-mirrorwritepriority** flag status and changing it*

```
IBM_2145:SVC_FLASHSYSTEM:superuser>lsvdisk 2
id 2
name IBMX3650_2
... [OUTPUT TRUNCATED]
mirror_write_priority latency
... [OUTPUT TRUNCATED]

IBM_2145:SVC_FLASHSYSTEM:superuser>chvdisk -mirrorwritepriority redundancy 2

IBM_2145:SVC_FLASHSYSTEM:superuser>lsvdisk 2
id 2
name IBMX3650_2
status online
... [OUTPUT TRUNCATED]
mirror_write_priority redundancy
```

2.3 SAN Volume Controller volume mirroring use cases with FlashSystem

Depending on the storage system chosen for each of the two volume mirror copies, some details must be considered in order to meet the high performance and low latency expectations of this solution. We describe some usage scenarios for SVC volume mirroring when virtualizing one or more FlashSystem storage systems, as well as the potential benefits, and best practices for each.

2.3.1 Volume mirroring between two FlashSystem storage systems

This is the most simple scenario for SVC volume mirroring because both copies reside on the same type of storage systems. In this scenario, there is no expected impact to performance in the event of a failure if the two subsystems are close to the SVC nodes in a *Standard*² architecture. In this case, Fibre Channel link latency is not likely to present an issue, and the default configurations for preferred node, primary copy, and mirroring priority are simplified. However, care should still be taken to ensure balancing of the workload between all the available resources (nodes, paths, fabrics, and storage subsystems).

However, if an SVC *Stretched Cluster* architecture is being deployed in this scenario, careful planning should be performed to ensure that the link latency between the SVC nodes, the

² Here, we mean *not* a Stretched Cluster.

hosts, and the two FlashSystem subsystems do not negatively impact the I/O operations from the applications. Some recommendations should be followed in this case, whenever possible:

- ▶ The preferred node for each mirrored volume should be kept at the same site as the subsystem, which contains the primary copy of that mirrored volume
- ▶ The host that is performing I/O operations to the mirrored volume should reside in the same site as the preferred node for that volume
- ▶ The **-mirrorwritepriority** flag for the mirrored volume should be set to 1 at latency if the access to the volume copy across the stretched cluster link represents a significant percentage of the total I/O latency, because this can compromise the cache usage. Otherwise, the recommended value of redundancy still applies, as for every stretched cluster architecture, allowing the highest level of data concurrency at both sites for protection against failure in the environment

2.3.2 Volume mirroring between a FlashSystem storage system and a non-flash storage system

In this scenario, usually adopted for cost-reduction reasons, planning should be used to avoid the penalty that is represented by the slowest subsystem to the overall I/O latency. Some recommendations are:

- ▶ The primary copy of each mirrored volume should be set for the copy residing in the FlashSystem subsystem so that all the reads are directed by SVC to it. This is commonly referred to as a *Preferred Read* configuration.
- ▶ If both subsystems are close to the SVC nodes in a *Standard* architecture, the **-mirrorwritepriority** flag for the mirrored volume should be set to 1 at latency, so the destage to the volume copy in the slowest subsystem does not introduce a negative impact in the overall write latency and, consequently, to cache usage.

2.4 Using FlashSystem with SVC Easy Tier

This section covers considerations for using FlashSystem storage systems with Easy Tier behind SVC. Suggested practices for design and implementation for this scenario will also be covered in this section.

In this scenario, the FlashSystem storage system is used with Easy Tier to improve performance across all storage pools within an existing environment. Due to the complexity of this scenario, there are some important considerations that must be factored in when designing and planning to add a FlashSystem to an existing environment.

Note: It is recommended that FlashSystem storage systems with 2D Flash RAID data protection are used with Easy Tier. FlashSystem 720 and 820 FlashSystem models have this protection.

2.4.1 General considerations with Easy Tier

A common question is when to use FlashSystem storage over internal SSDs in SVC. The general recommendation is to use FlashSystem storage when your capacity requirements for Easy Tier exceed five SSDs. At this point, FlashSystem storage systems should be considered for cost efficiency and performance overhead reasons. For more information, refer to *Flash or SSD: Why and When to Use IBM FlashSystem*, REDP-5020.

When planning to use FlashSystem storage systems with Easy Tier, first use the IBM Storage Tier Advisor Tool (STAT) to obtain a comprehensive analysis of hot extents. This will allow you to estimate the amount of required FlashSystem capacity. For more information about using this tool, refer to *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933, or later.

When planning the MDisk layout for a FlashSystem to be used with Easy Tier, the amount of capacity on existing spindle-based storage pools should be considered. The recommended maximum MDisk allocation for a single FlashSystem is 16 MDisk. Use your Easy Tier capacity requirements, along with this recommendation, to determine the number of FlashSystem storage systems that are required.

Where possible, the previously recommended extent size of 1024 MB (1 GB) should be used³. When working with existing disk pools, the defined extent size for the existing storage pools may be different. It is a recommended best practice that all storage pools use the same extent size behind SVC because this allows for seamless volume migration. It is likely that the existing storage pool extent size will be 256 MB or 512 MB based on the suggested best practice for spindle-based storage pools. In these cases, use the extent size of the existing pool for the FlashSystem corresponding pool.

When configuring the FlashSystem MDisk, ensure that you add these MDisk as *SSD MDisk*s. Otherwise, Easy Tier will not be able to distinguish between the spindle-based generic hard disk drives (HDDs) and the FlashSystem SSD MDisk. This can be done using the `-tier generic_ssd` switch when adding MDisk to an MDisk group, as explained in 4.3, “SAN Volume Controller MDisk configuration” on page 111.

Refer to *Implementing the IBM System Storage SAN Volume Controller V6.3*, SG24-7933, or later for more information about Easy Tier.

Note: The port allocation and masking for this scenario should be configured as in 2.1, “Usage considerations” on page 42.

2.4.2 Separate FlashSystem storage pools and existing storage pools that are not using IBM Easy Tier

This section covers some brief considerations when using FlashSystem in an SVC environment where existing spindle-based storage pools exist that are not using Easy Tier, and the suggested recommendations.

General considerations

The main consideration if using FlashSystem storage systems in an environment where Easy Tier is *not* enabled is to treat it as any other virtualized storage device, placing it in its own storage pool with a dedicated SVC I/O group.

Regarding port masking for this scenario, it is recommended that the FlashSystem has a dedicated pair of Fibre Channel ports for each SVC I/O group. In an existing environment, this might require some reconfiguration of the existing port allocation, port masking, and zoning to accommodate this requirement.

³ The recommendation of a 1024 MB extent size is based on lab testing by IBM. An extent size of 1024 MB was shown to provide the best performance with Easy Tier.

Note: If not already applied, port masking should be applied to ensure that the dedicated FlashSystem Fibre Channel ports are not used for intracluster or replication communication. Refer to 2.1.3, “Port masking on SVC with IBM FlashSystem 820” on page 46 for more information about port masking.

The introduction of FlashSystem into the environment provides an increase in the SVC cluster’s overall potential performance. For maximum achievable performance, refer to *IBM SAN Volume Controller and IBM FlashSystem 820: Best Practice and Performance Capabilities*, REDP-5027.

Note: The port allocation and masking for this scenario should be configured as in 2.1, “Usage considerations” on page 42.

In scenarios where increased host port connectivity requirements (or existing host port connectivity) restrict your ability to dedicate Fibre Channel ports for FlashSystem connectivity, consider adding additional SVC nodes to the cluster.

An example of a suggested best practice port allocation for dual HBA SVC nodes is shown in Table 2-2. We assume that replication is in use.

Table 2-2 Suggested port allocation for dual HBA SVC nodes when using FlashSystem storage pools and existing storage pools and not using Easy Tier

HBA	Intracluster communication port	Host communication port	Replication port	FlashSystem back-end port
HBA 1	4	3	2	1
HBA 2	8	7	6	5

An example of a suggested best practice port allocation for single HBA SVC nodes is shown in Table 2-3. Again, we assume that replication is in use.

Table 2-3 Suggested port allocation for single HBA SVC nodes when using FlashSystem storage pools and existing storage pools and not using Easy Tier

HBA	Intracluster communication port	Host communication port	Replication port	FlashSystem back-end port
HBA 1	4 and 3	3	2	1

2.5 FlashSystem with SVC replication

This section explains, briefly, some considerations when using IBM FlashSystem 820 with SVC and using replication.

There are five main points of consideration:

- ▶ Additional latency overhead with synchronous Metro Mirror
- ▶ Distance of cross site links and additional latency overhead
- ▶ Adequate bandwidth for cross site links
- ▶ Amount of data to be replicated and its I/O rate
- ▶ Dedicated replication ports

IBM FlashSystem storage systems provide extremely low latency. As such, the latency of Metro Mirror links might affect FlashSystem storage systems to a greater degree than other traditional disk systems used with SVC. Metro Mirror replication distances in excess of 10 km should be carefully analyzed to ensure that they will not introduce bottlenecks or increase in response times when used with FlashSystem storage systems.

Dedicating ports for replication is strongly recommended with this solution. Isolating replication ports can disperse congestion and reduce latency for replication, while protecting other ports from the impacts of increased amounts of replication traffic.

Fibre Channel is the preferred connectivity method using a Dense Wavelength Division Multiplexer (DWDM), or equivalent device, between source and target sites. Further, the use of inter-switch links (ISLs) in a trunk or port channel to increase the aggregate bandwidth between the two sites is recommended. Also, size the connectivity between sites according to the amount of bandwidth that you are allocating on the SVC for replication, with additional bandwidth for future growth and peak loads. To summarize, consider the following factors when designing the replication connectivity:

- ▶ Current average and peak write activity to the volumes that you plan to replicate
- ▶ Desired recovery point objective (RPO) and recovery time objective (RTO) values
- ▶ Existing best practices for SVC replication, detailed in *IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services*, SG24-7574

Note: When using Fibre Channel over IP (FCIP) for replication in this scenario, consider using at least 10 Gb links between sites.

2.6 Failure protection capabilities

This section describes the failure protection capabilities of the hardware components of SVC and FlashSystem storage systems' solutions.

2.6.1 FlashSystem 820 hardware failure protection

The following section identifies the failure protection capabilities of the IBM FlashSystem 820 storage system.

The FlashSystem 820 is protected from outages of the following components through redundant hardware:

- ▶ Power supplies: Dual, hot-swappable power supplies
- ▶ RAID controllers: Dual Redundant Array of Independent Disks (RAID) controllers
- ▶ HBA cards: Dual HBA cards with hot-swappable SFPs
- ▶ Management control processors: Dual management control processors, each with its own Ethernet management port
- ▶ Fans: Fans are redundant and can sustain fan failures
- ▶ Batteries: Two batteries; one is required for system operation

Note: In the IBM FlashSystem 820 storage system, flash modules are hot-swappable and field-replaceable by IBM service.

The flash modules are protected by 2D Flash RAID and can sustain two module failures before an outage occurs. There are 12 flash modules in a standard IBM FlashSystem 820, which means roughly 17% of the system capacity can fail before overall system operation is affected.

RAID rebuild: If a module fails, the 2D Flash RAID protection transparently rebuilds the data on the failed module by using the built-in spare capacity. Overall system performance might be affected during the RAID rebuild process. Lab testing identified potential performance of up to 20%.

Firmware upgrades: The FlashSystem 820 storage system does not currently support concurrent firmware upgrades. For details about performing upgrades, see 5.2, “FlashSystem firmware update non-disruptively with the use of volume mirroring” on page 124.

Important note about high availability: Where high availability is a requirement, it is recommended that dual IBM FlashSystem 820 devices are used with volume mirroring. For further details, refer to 2.2.2, “FlashSystem with SAN Volume Controller volume mirroring” on page 51.

2.6.2 SVC hardware failure protection

The following section identifies the failure protection capabilities of the SAN Volume Controller.

Specifically, the SVC design includes built-in hardware protection at the node level. Further, the design groups nodes together in active/active pairs (I/O groups).

Upon node failure, the partner node in an SVC I/O group picks up the additional workload and continues to serve data without an outage. The potential for performance impact exists until the failed node is restored or replaced. During the failure, access to the back-end IBM FlashSystem 820s is maintained through the partner node. Depending on the port assignment, there might be a reduced number of SAN paths between the SVC and the FlashSystem during the outage.

2.6.3 SAN fabric failure considerations

A recognized best practice to protect against SAN fabric failure scenarios is to implement redundant SAN fabrics. This allows for the failure of a single fabric without disruption of the connected devices. If there is a SAN fabric failure, or a failure of a fabric component, access to the back-end storage systems is maintained through the redundant connections, but overall performance might be affected during the failure because of a reduction in paths.

In summary, the same principal applies to the SAN fabric as to the SVC I/O groups: If a hardware failure occurs, there is no outage, but there might be reduced performance capabilities.

More best practices can be found in the following IBM Redbooks publication:

IBM System Storage SAN Volume Controller Best Practices and Performance Guidelines, SG24-7521



Planning and installation of the IBM FlashSystem 820

In this chapter, we explain the physical planning and installation considerations for the IBM FlashSystem 820 for use with the IBM SAN Volume Controller (SVC), and that we used to successfully create our lab environment.

We start with physical planning and connectivity considerations, then provide some information on power and cooling specifications, and finally offer some performance guidelines and additional supportability information.

3.1 Physical device accessibility

This section describes the physical planning aspects when working with an IBM FlashSystem 820 and its accessibility for serviceability.

3.1.1 Physical layout

It is important to remember that although the IBM FlashSystem 820 storage system has high levels of redundancy built in, at some point, access to the internal hardware for replacement of failed components might be required. The storage system opens at the top of the unit, similar to an IBM System x server. For a diagram of the IBM FlashSystem 820 storage system with the lid removed, identifying the internal system components, see Figure 3-1.

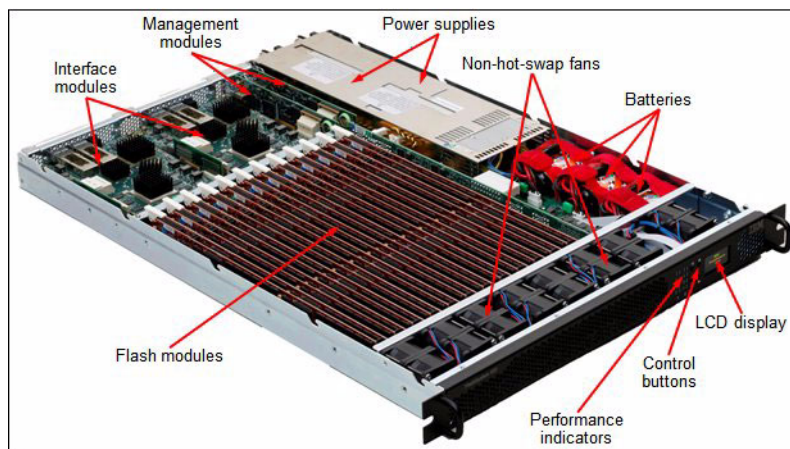


Figure 3-1 Shows an internal view of an IBM FlashSystem 820

Removal of the lid requires the ability to slide out the entire device on rails and have the room to take the top off to perform maintenance tasks. This is an important consideration.

Figure 3-2 provides a rear view of the IBM FlashSystem 820 storage system, detailing the power, Ethernet management port, and Fibre Channel (FC) port connectors.

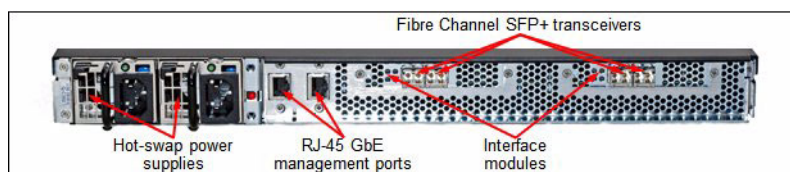


Figure 3-2 Shows the rear of an IBM FlashSystem 820

3.1.2 Rack placement and cabling considerations

The IBM FlashSystem 820 storage system contains numerous hot-swappable internal components, allowing for the service of the system while it is still connected and serving data. As such, there are several considerations to account for when planning the rack layout for the storage systems.

First, it is recommended to install the devices at an accessible height between RU 13 - 30, allowing for easy and safe access to the system internals if the systems require future service, as shown in Figure 3-3.

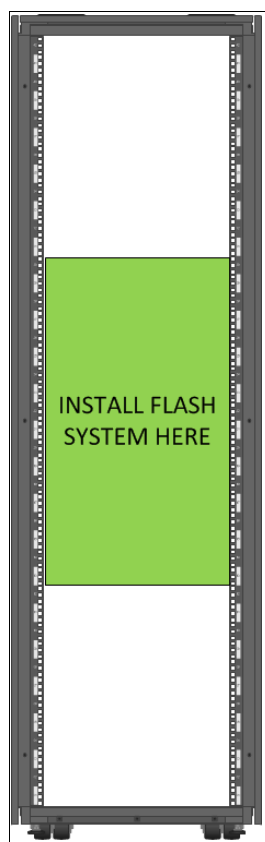


Figure 3-3 Shows the ideal installation area for IBM FlashSystem storage systems in a 42RU IBM rack

Second, it is important to ensure that all cabling to the systems includes sufficient slack to allow the systems to slide out fully from the rack without causing cable disconnection because the IBM FlashSystem 820 storage systems do not come with a cable tray. It is a good idea to measure the distance and test that the system can slide in and out from the rack freely before any configuration tasks are initiated. The device should be fully cabled up and powered on when this test is conducted and all connectivity verified when conducting the test to ensure it will work in a production situation.

These are important considerations that could affect the ability to carry out required maintenance on IBM FlashSystem 820 storage systems, which could in turn affect system uptime.

3.2 Physical cabling

This chapter details the considerations when planning and executing the physical installation of an IBM FlashSystem 820 for use with SAN Volume Controller. It also includes some notes and best practice considerations for fibre cabling, network cabling, physical installation, and configuration.

3.2.1 FC port speed settings

There are a couple of considerations when cabling the IBM FlashSystem 820 device to the SAN. The SAN switches should have 8 GB ports, where available. However, the IBM FlashSystem 820 FC ports will run at 4 GB and 2 GB, if required. The best practice recommendation is to manually set both the FlashSystem FC ports and the storage area network (SAN) switch FC ports to the highest mutually available speed, rather than using auto negotiate. This is done purely for consistency and stability. See Example 3-1.

*Example 3-1 Shows a truncated **switchshow** output from an IBM SAN switch, showing ports connected to an IBM FlashSystem 820 set to 8 GB*

```
> switchshow
.
.
Index Port Address Media Speed State      Proto
=====
.
.
 36 36  012400  id    8G   Online    FC  F-Port  20:04:00:20:c2:12:24:c0
 37 37  012500  id    8G   Online    FC  F-Port  21:04:00:20:c2:12:24:c0
 38 38  012600  id    8G   Online    FC  F-Port  20:08:00:20:c2:12:24:c0
 39 39  012700  id    8G   Online    FC  F-Port  21:08:00:20:c2:12:24:c0
.
.
```

For details about how to configure the FC controller port settings of the IBM FlashSystem 820 storage system, refer to 4.2, “Port masking and SAN zoning configuration” on page 108.

3.2.2 Cabling methods

The IBM FlashSystem 820 storage system supports either 8 Gb Fibre Channel or quadruple data rate (QDR) InfiniBand interfaces. In our example environment, our FlashSystem 820 storage systems are each configured with two dual port 8 Gb Fibre Channel cards, for a total of four 8 Gb FC ports. The optimal cabling scenario with this hardware configuration is to cable port 1 (P1) on each card to Fabric A and port 2 (P2) on each card to Fabric B, as shown in Figure 3-4 on page 67.

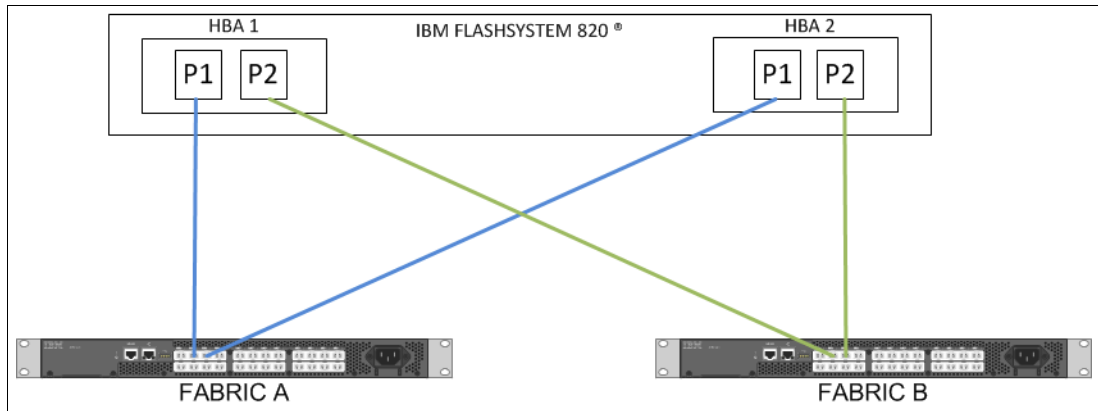


Figure 3-4 Cabling diagram based on dedicated port groups and dual fabrics

This configuration achieves fabric switch level redundancy and Fibre Channel card level redundancy, providing protection in the event of an issue or failure.

A further consideration is to distribute the IBM FlashSystem 820 Fibre Channel ports onto dedicated port groups, if possible. If this is not possible, it is recommended to at least distribute onto different port groups on the SAN switch to distribute the workload and provide the best available bandwidth.

If the environment only contains a single SAN switch, all four Fibre Channel ports must be cabled to this switch. This will work functionally, but does not provide any redundancy at the SAN switch layer, and can result in an outage if a switch failure occurs.

For optimal throughput and performance, four Fibre Channel switch ports for each IBM FlashSystem 820 storage system are required.

3.2.3 FC cable type

Another consideration is that OM4 standard cabling should be used where possible to provide the clearest possible connection. The connectors should also all be the LC connector standard.

3.2.4 Ethernet management cabling

The IBM FlashSystem 820 contains dual management control processors, each with its own Ethernet management port. See Figure 3-2 on page 64 for a picture that identifies the ports on the system. To ensure the maximum availability of system management when using this model, it is recommended that you cable and configure both management interfaces. Connecting them to different virtual local area networks (VLANs), or optimally, different network switches, will provide the network redundancy and offer protection from a network disruption. For further details, refer to the *IBM FlashSystem 820 User's Guide* and *IBM FlashSystem 720 and IBM FlashSystem 820*, TIPS1003.

The default speed setting of the FlashSystem Ethernet management network interface is *auto*, allowing the port to negotiate speed and duplex settings with the switch. The maximum configurable speed of the interface is 1 GB full duplex. If you are required to hard set the speed or duplex settings on your Ethernet network, the IBM FlashSystem 820 management ports can be explicitly configured. For more information about configuring the IBM FlashSystem Ethernet management ports, refer to section 4.1, "Setup and configuration of IBM FlashSystem for use with SAN Volume Controller" on page 78.

Note: You will require at least one IP address for each IBM FlashSystem 820 storage system for management purposes. For reliability and redundancy, *two static* IP addresses for each FlashSystem are recommended.

3.3 Power and cooling considerations

This section details the power consumption and cooling requirements of a single IBM FlashSystem 820 storage system. Unlike other disk subsystems, these devices require a relatively low amount of power. This reduces the cost and power consumption within your data center. An example of energy costs per hour to cool one system is also provided. This information is important when planning to scale out an IBM FlashSystem 820 implementation rapidly.

3.3.1 Power requirements

The IBM FlashSystem 820 storage system comes with dual, redundant AC power modules that connect to a 10 amp capable power supply. The unit ships with two IEC 320-C13 to C14 rack Power Distribution Unit (PDU) power cords. Refer to Figure 3-2 on page 64 for a graphic that identifies the storage system power connectors. For more detailed specifications on the IBM FlashSystem 820 storage system, refer to *IBM FlashSystem 720 and IBM FlashSystem 820*, TIPS1003.

A single IBM FlashSystem 820 will consume 300 watts of power, which equates to 1.25 amps.

Note: The IBM FlashSystem 820 device should be connected to an uninterruptible power supply (UPS) protected power source and each power supply should be on a different phase of power to provide power redundancy.

3.3.2 Cooling requirements

The IBM FlashSystem 820 storage system has a British Thermal Unit (BTU) per-hour rating of 1023. This equates to roughly 1.25 amps per hour of energy used to provide cooling for a single IBM FlashSystem 820 storage system.

It is recommended that the cooling vents be at the front of the rack so that the air flows from the front of the rack to the back.

3.4 Performance Guidelines

In this section, we briefly outline the performance guidelines and performance accelerators that can be expected with an IBM FlashSystem 820 storage system when virtualized behind a SAN Volume Controller.

Supported features such as vStorage API for Array Integration (VAAI) are used to accelerate VMware performance and off load storage-related tasks from the VMware hosts to the storage by performing operations more efficiently.

3.4.1 VMware VAAI

The SAN Volume Controller supports the following VAAI features.

Table 3-1

Official name	Description	Function	FCP
Atomic test & set (ATS)	Hardware Assisted Locking	Enables granular locking of block storage devices	Y
Cloning blocks	Full Copy/Extended Copy	Commands the array to make a mirror of a LUN (Used for Vmotion/cloning)	Y
Zeroing file blocks	Block Zeroing	Communication mechanism for thin provisioning arrays	Y

Note: For any implementation it is essential that you check the IBM System Storage Interoperation Center (SSIC) to ensure that the proposed implementation is supported:

<http://www.ibm.com/systems/support/storage/ssic/interoperability.wss>

When using VMware with the IBM FlashSystem 820 storage system virtualized behind SAN Volume Controller, it is important to remember that VAAI, if not enabled on all storage volumes, introduces additional overhead when transferring data from a VAAI-enabled storage volume to a non-VAAI enabled storage volume and vice versa. This impact can be seen on the CPU and memory resources of the ESX hosts, and the back-end performance is generally poor because VMware will try to use VAAI and cannot.

The best practice is to have VAAI fully enabled or fully disabled. It is also advised that you ensure that the environment firmware levels are fully supported by all VAAI features if you plan to use them.

For more information, see the SAN Volume Controller interoperability website:

<http://www.ibm.com/systems/storage/software/virtualization/svc/interop.html>

Also, refer to the following VMware Knowledge Base article KB1033665 about how to disable VAAI:

http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1033665

3.4.2 Performance Redpaper

For expected performance results, including the maximum recorded I/O and latency with an IBM SAN Volume Controller and IBM FlashSystem 820 storage system sample configuration, refer to the following IBM Redpaper publication *IBM SAN Volume Controller and IBM FlashSystem 820: Best Practice and Performance Capabilities*, REDP-5027.

3.4.3 Performance data and statistics collection

Performance monitoring and analysis are very important for high-performing systems such as IBM FlashSystem 820 and IBM SAN Volume Controller. In most cases when using this bundle, there is sufficient performance data provided by SVC and monitored by IBM Tivoli Productivity Center. However, for precise performance tuning and identification of bottlenecks, statistics from the IBM FlashSystem 820 should be collected and analyzed too.

There are two methods of performance monitoring in IBM FlashSystem 820:

- ▶ Using command-line interface (CLI) for current data
- ▶ Using graphical user interface (GUI) for current and historical data

For information about how to access the IBM FlashSystem 820 by GUI and CLI, refer to 4.1, “Setup and configuration of IBM FlashSystem for use with SAN Volume Controller” on page 78.

Using the CLI to access current data

Performance statistics management can be done by using **stats** command. See Example 3-2.

Example 3-2 Output of stats command

```
admin #: stats
```

```
Valid sub-commands are:
```

```
view
info
log
```

This command allows three options (subcommands):

- ▶ the **view** subcommand is used to read specific system component and statistic name combinations. Multiple combinations can be viewed at a time. The statistic values are refreshed once every second
- ▶ the **info** subcommand is used to view all system components with statistics available, as well as each statistic that is available for specific system components.
- ▶ the **log** subcommand is used to display information about statistics logging or to update the logged statistics list. Issuing the **stats log** command without arguments will display the current logged statistics.

The format of the **stats** command is the following:

```
stats view [<system_component> <statistic_name>]
```

To view the list of available components with statistics, use the **stats info** command. See Example 3-3 for reference:

Example 3-3 List of the components with statistics available

```
admin #: stats info
```

The following system components have statistics available:

```
bus
SuperEnv
SystemEnvA
SystemEnvB
mcp-1
mcp-2
flashcard-1
flashcard-2
flashcard-3
flashcard-4
flashcard-5
flashcard-6
flashcard-7
flashcard-8
flashcard-9
flashcard-10
flashcard-11
flashcard-12
fc-1
fc-1a
fc-1b
fc-2
fc-2a
fc-2b
```

For each component, a list of statistics is available. Use the **stats info [component_name]** command. See Example 3-4:

Example 3-4 List of the statistics that are available for the component "flashcard-1"

```
admin #: stats info flashcard-1
```

The following statistics are available for the 'flashcard-1' system component:

Statistic Name -----	Description
temperature	temperature
onboard.cap_sense_stat	(onboard) cap_sense_stat
onboard.cap_vin_stat	(onboard) cap_vin_stat
onboard.cap_adin_stat	(onboard) cap_adin_stat
onboard.fpgas_per_board	(onboard) fpgas_per_board
onboard.fpgas_in_chain	(onboard) fpgas_in_chain
onboard.cs_per_lane	(onboard) cs_per_lane
onboard.presented_size_align8	(onboard) presented_size_align8
onboard.presented_size_max8	(onboard) presented_size_max8
onboard.temperature	(onboard) temperature
onboard.health_state	(onboard) health_state
onboard.overall_health	(onboard) overall_health
onboard.available_remap	(onboard) available_remap
onboard.block_wear_max	(onboard) block_wear_max
onboard.block_wear_min	(onboard) block_wear_min

onboard.block_wear_avg	(onboard) block_wear_avg
onboard.available_brp	(onboard) available_brp

Example 3-5 shows the usage of the **stats** command to get the statistics of a specific component.

Example 3-5 Output of the component statistics

```
admin #: stats view system iops

system
iops
-----

2030.00
2100.00
1990.00
1270.00
```

The statistics update time is one second.

To obtain the list of the statistics logs that are available, use the **stats log** command. See Example 3-6 for reference.

Example 3-6 Log files that are available on the system

```
admin #: stats log

Logged Statistics:

Object-----Name-----Description-----
system      bw              Bandwidth
system      idle_cpu       CPU Idle
system      iops           IOs per second
system      iowait_cpu     CPU IO Wait Utilization
system      irq_cpu        CPU Hard IRQ Utilization
system      nice_cpu       CPU Nice Utilization
system      rb_bw_avg      Average Rebuild BW
system      softirq_cpu    CPU Soft IRQ Utilization
system      system_cpu     CPU System Utilization
system      temperature    Max System Temperature
system      user_cpu       CPU User Utilization
```

Log files present on the system can be obtained through the GUI. See “Using the GUI to access current and historical data” for reference. Statistics monitoring using CLI is intended for monitoring current data for a short time while doing performance tuning or while identifying a bottleneck. To obtain statistics for a specific period, scripting methods can be used.

Using the GUI to access current and historical data

The GUI provides a more convenient way to monitor and gather performance data. It is possible to monitor current activity, as well as retrieve historical data stored in the log files. Figure 3-5 on page 73 shows the initial view of the performance monitoring tool in the IBM FlashSystem 820 GUI.

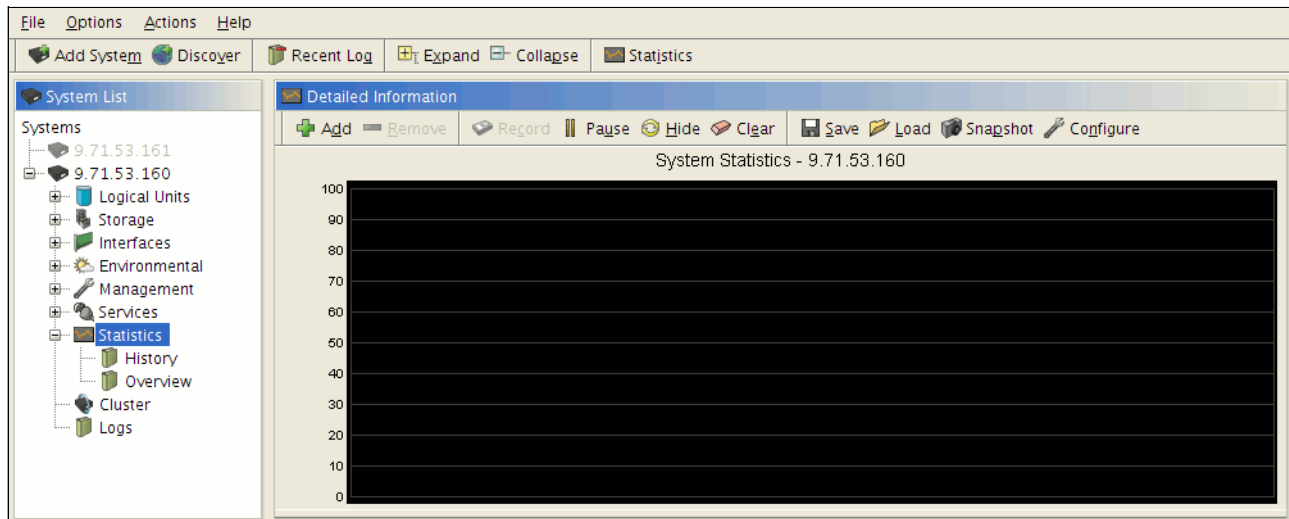


Figure 3-5 Overall view of the performance statistics option in IBM FlashSystem 820 GUI

This view allows you to monitor the current statistics and save the momentary values to a text file for future usage. You can also make a window capture of the current statistics graphs and save it to a JPG file. To do that, select **Snapshot** and **Save** on the window menu accordingly.

To add a system component statistic to monitor, use the **Add** button of the window menu. See Figure 3-6.

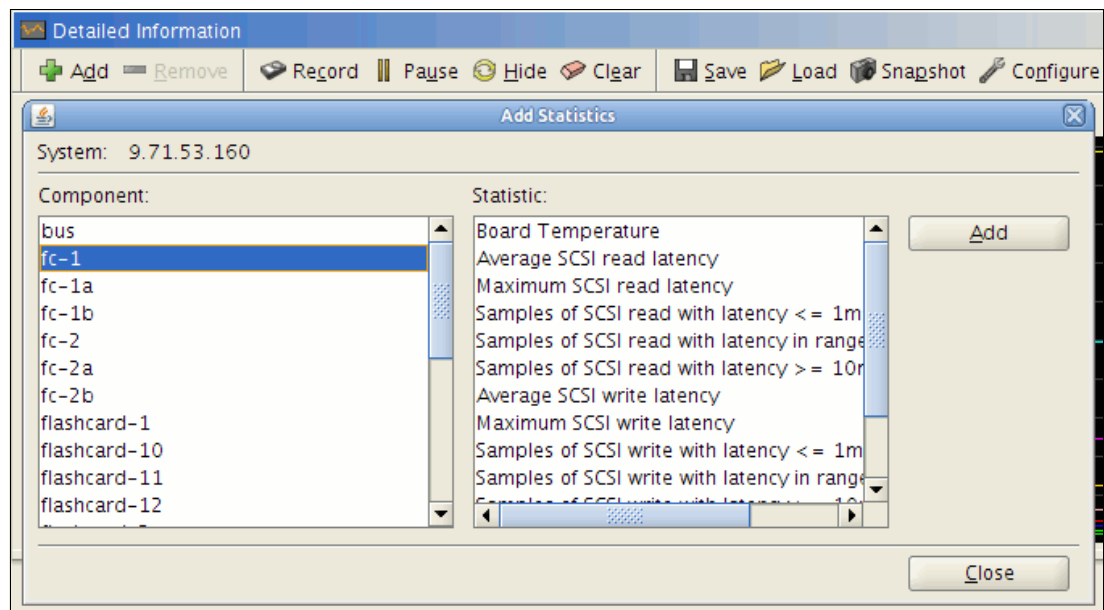


Figure 3-6 Add statistics dialog of the performance monitoring tool

Using this dialog, it is possible to add several components and component statistics at a time. The selected statistics are presented in different colors on the graph. This is shown in Figure 3-7 on page 74.

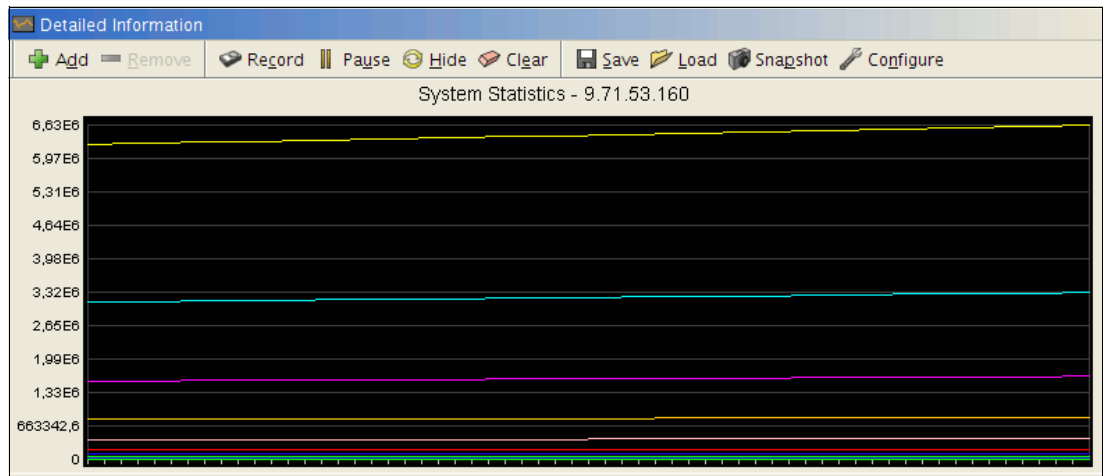


Figure 3-7 Statistics values represented on the graph

Color and scale can be adjusted for each value by using the **Configure** tab of the menu. The interval for statistics collection can be selected in the **Configure** dialog, as well. See Figure 3-8.

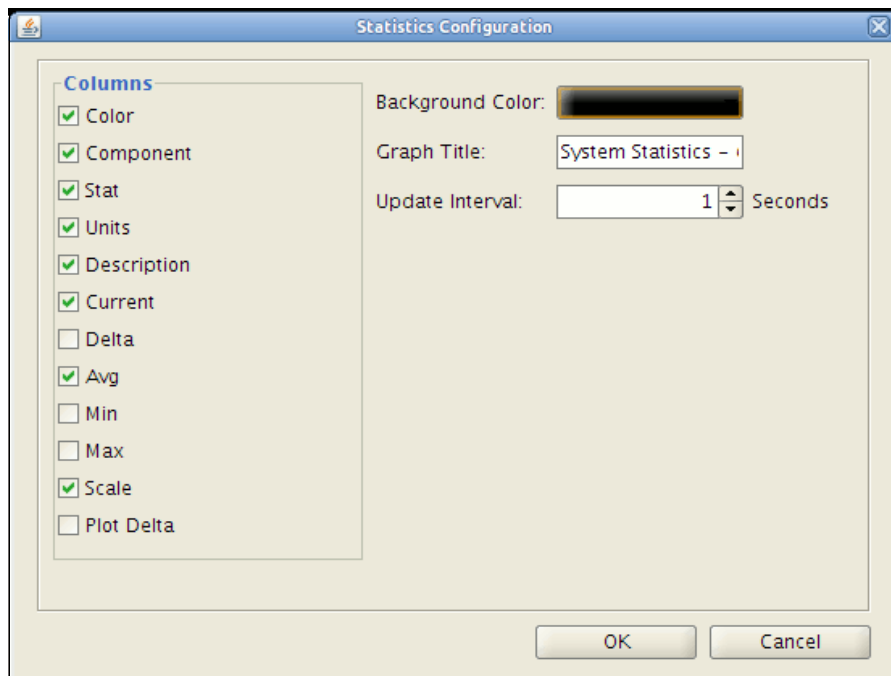


Figure 3-8 Configure dialog of the performance tool

To view historical performance data, use the **History** tab of the **Statistics** tab of the main management menu. This tab opens the list of the available statistics in log files, which can be viewed or saved. It is possible to perform multiple selections and combine several graphs in one to compare different values for better performance analysis. See Figure 3-9 on page 75.

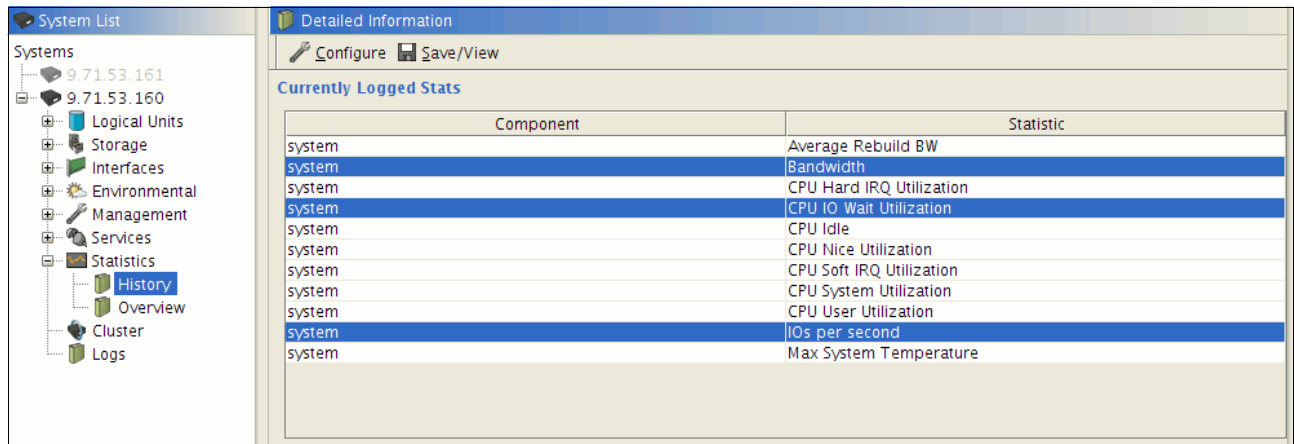


Figure 3-9 Historical statistics log files to view multiple selections

Select **Save/View** on the top menu to view selected statistics on the same graph for analysis and comparison. See Figure 3-10.

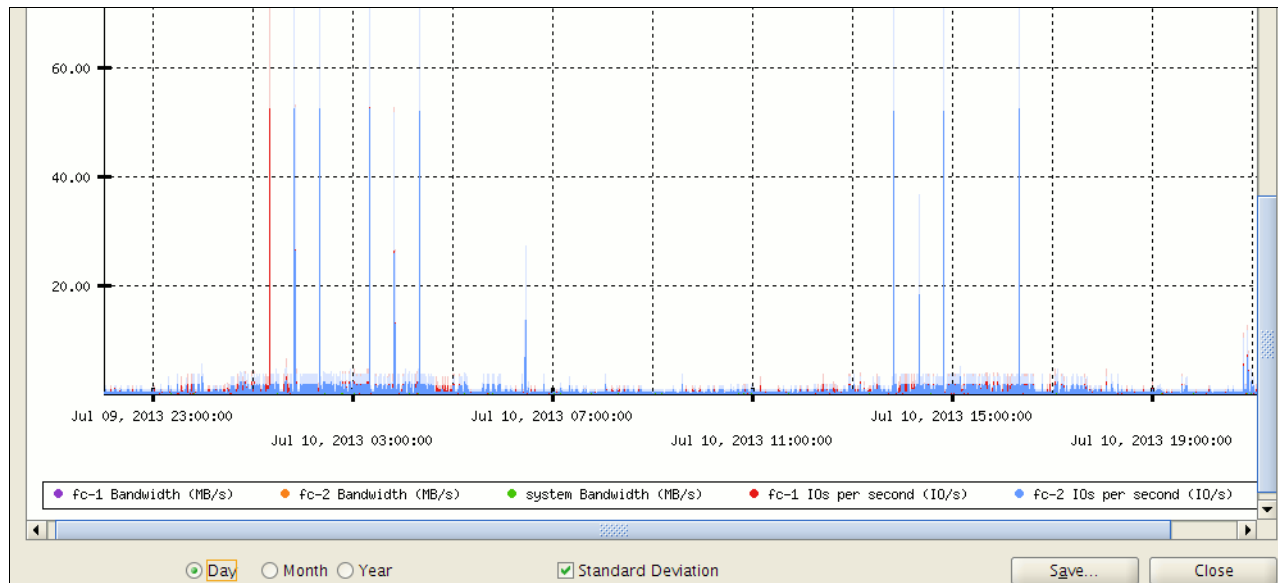


Figure 3-10 Graph with multiple statistics for analysis

To save selected data, select **Save**. Data will be saved in JPG file format.



Configuration and administration

This chapter provides information about the initial configuration of the environment composed by the IBM SAN Volume Controller virtualizing two IBM FlashSystem model 820 subsystems.

The following topics are covered:

- ▶ Initial setup of the environment
- ▶ Port masking and storage area network (SAN) zoning configuration
- ▶ Managed disk (MDisk) configuration

Note: For details about the sample environment that was used to write this chapter, refer to Appendix B, “Example environment details” on page 139.

4.1 Setup and configuration of IBM FlashSystem for use with SAN Volume Controller

After the IBM FlashSystem storage systems have been racked, cabled, and powered on, several steps must be performed in order to configure them optimally for use with the SAN Volume Controller.

4.1.1 Configuring the FlashSystem management IP addresses

In order to access the FlashSystem to perform management tasks, at least one of the management control processor Ethernet ports must be configured with an IP address. This allows you to access the Web Monitor (graphical user interface (GUI)) with your browser or the command-line interface (CLI) over Secure Shell (SSH). For more information about the GUI and CLI, refer to the *IBM FlashSystem 820 User's Guide*, and the *IBM FlashSystem 820 Command Line Interface Guide* (currently, the CLI guide is not available).

Note: The best practice recommendation when using the IBM FlashSystem 820 storage system is to cable and configure both management control processors. For more information, refer to 3.2.4, “Ethernet management cabling” on page 67.

Configuring the management IP address using the front panel display

The default factory IP address setting of the FlashSystem management controller is Dynamic Host Configuration Protocol (DHCP). In our examples, we use a static IP configuration on Eth0 of management control processor mc-1, and the second management control processor, mc-2, is not used. Before the FlashSystem is accessible on your network, the LCD display and controls on the front panel of the system, as shown in Figure 4-1, can be used to configure the IP address settings.

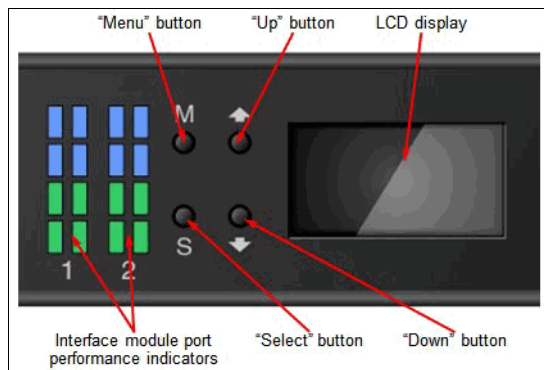


Figure 4-1 IBM FlashSystem front panel LCD and controls

To configure the IP address settings, use the front panel, following these steps:

1. Press the **Menu** button on the front panel. Use the **Down Arrow** button to scroll the cursor to the **Net Config** option and push the **Select** button.
2. If necessary, use the arrow buttons to navigate the cursor to the **mc-1** option. Push the **Select** button to configure the IP address of the management controller 1 (mc-1).
3. Use the **Down Arrow** button to scroll the cursor to the **Eth0 Config** option and push the **Select** button.

4. If necessary, use the arrow buttons to navigate the cursor to the **Static** option and push the **Select** button.
5. Set the IP address using the **Up** and **Down** buttons to move the cursor right or left and the **Select** button to cycle through the numbers **0 - 9**. You can abort your current changes at any time by pressing the **Menu** button and following the dialog.
6. After you have finished entering the IP address, scroll the cursor to the right using the **Up** button — this opens the **Subnet Mask** window. Using the same procedure as in step 5, enter the subnet mask.
7. After you have finished entering the subnet mask, scroll the cursor to the right again using the **Up** button — this brings up a window that indicates the values to enter for no gateway. Scroll to the right again using the **Up** button to access the **Gateway** window. Using the same procedure as in step 5, enter the gateway IP address.
8. After entering the gateway IP address, scroll to the right again using the **Up** button — this brings up the **Review Config** window, allowing you to review, apply, edit, or cancel the settings that you have entered.
9. To apply your changes, scroll the cursor to the **Apply** option and push the **Select** button. On the next window, push the **Down** button to apply the changes. The settings are saved and the network is automatically restarted.
10. Confirm that you can access the management interface of the FlashSystem by using the **ping** utility from a host machine, as shown in Example 4-1.

Example 4-1 Testing access to the IBM FlashSystem management port

```
>ping 9.71.53.160
```

```
Pinging 9.71.53.160 with 32 bytes of data:
Reply from 9.71.53.160: bytes=32 time=19ms TTL=60
Reply from 9.71.53.160: bytes=32 time<1ms TTL=60
Reply from 9.71.53.160: bytes=32 time<1ms TTL=60
Reply from 9.71.53.160: bytes=32 time=1ms TTL=60
```

```
Ping statistics for 9.71.53.160:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 19ms, Average = 5ms
```

Note about the above steps: The steps outlined above for configuring the management IP address via the front panel might differ slightly, depending on whether your IBM FlashSystem model contains a single management control processor, or dual management control processors, as in our example. For more information about the features of specific IBM FlashSystem models, refer to *IBM FlashSystem 720 and IBM FlashSystem 820*, TIPS1003, and *IBM FlashSystem 710 and IBM FlashSystem 810*, TIPS1002.

Note about port speed and duplex: The default speed and duplex settings of the management interface are both *auto*. In our example, we have not modified these settings. If your network requires that explicit speed and duplex settings be configured, you can do so via the **network** CLI command.

Important: The IBM FlashSystem 820 contains dual management control processors, each with its own Ethernet management port. To ensure the maximum availability of system management when using this model, it is recommended that you cable and configure both management interfaces. For more information, refer to the *IBM FlashSystem 820 User's Guide*, and *IBM FlashSystem 720 and IBM FlashSystem 820*, TIPS1003.

4.1.2 FlashSystem management tools

IBM FlashSystem products offer two methods of access for performing common management tasks. A GUI tool is accessible with a standard web browser via HTTP or HTTPS. A CLI is accessible over SSH or Telnet.

Note: Access to the management tools via the web browser, SSH, or Telnet, can be enabled or disabled on the FlashSystem via the **system services** CLI command or the Services window in the GUI. Additionally, you can dictate the use of HTTP or HTTPS for GUI access via the GUI login window, as shown in Figure 4-3 on page 82.

Managing the FlashSystem using the command-line interface

To access the FlashSystem CLI, your host system must support SSH or Telnet. To access the CLI, open an SSH or Telnet connection to the management IP address of the system. In our example, we use the PuTTY utility to connect over Telnet, as shown in Figure 4-2.

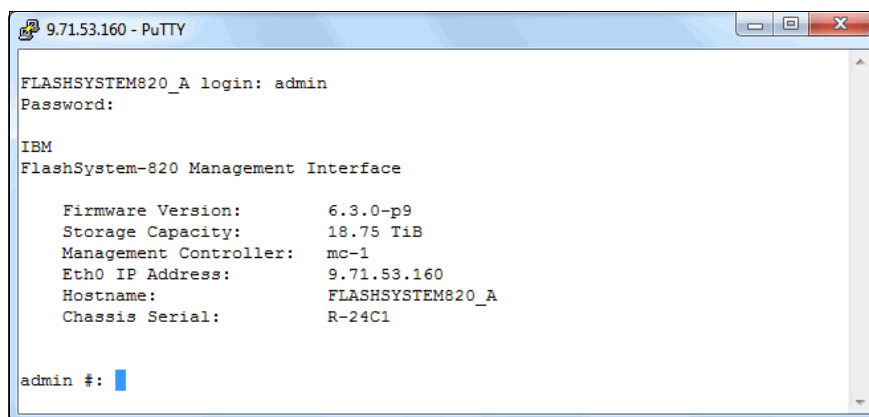


Figure 4-2 Accessing the FlashSystem CLI over Telnet

The IBM FlashSystem CLI contains a **help** command that provides detailed instructions on how to use the various commands. Issuing a simple **help** or **?** command lists the available commands, as shown in Example 4-2.

Example 4-2 Viewing the available IBM FlashSystem CLI commands

```
admin #: help
Top level commands:
  fc          Configure a Fibre Channel controller.
  network     View or configure network settings.
  user        Configure the user table.
  system      Read system information or perform system commands.
  support     Display contact information for local support team.
  exit        Exits this shell.
  lu          Configure the Logical Unit table.
```

help	Display help information.
time	Configure system time.
log	Read system log files.
confirm	Modify confirmation message behavior.
task	View information about tasks issued.
snmp	Configure SNMP settings.
mail	Configure mail settings.
license	Configure system feature licenses.
stats	View system component statistics.
who	List users currently logged into the system.
status	Display environmental sensor values and states.
diag	Run system diagnostics.
ib	Configure an InfiniBand controller.
storage	View/modify storage information.
sync	View and change various GbE sync properties.

Try `help <command>` for details.

Issuing the **help** command along with another CLI command provides information about the syntax and details of the command, as shown in Example 4-3.

Example 4-3 Viewing the detailed help information for the IBM FlashSystem CLI network command

```
admin #: help network
```

NAME

network - View or configure network settings

SYNOPSIS

```
network <management_controller> [<option> [<args>]]
```

DESCRIPTION

The network command configures the management port's Ethernet settings. After changing the network settings, you must issue the 'network restart' command to apply the changes. Issuing the network command without arguments will display the current network configuration.

Management controller specifies which controller to communicate with; for example, 'mc-1' or 'mc-2'. Typing 'network' will show configurations for both controllers and display which unit you are currently logged into.

OPTIONS

ldap	View or configure LDAP settings.
dns	Display the network DNS settings.
link_speed	Set the network link speed settings.
restart	Restart the network.
ip_assignment	Read or set the ethernet device IP assignment mode.
hostname	Read or configure the hostname.

Managing IBM FlashSystem using the GUI

To access the FlashSystem management GUI, your browser must support Sun Java version 1.5 or later. To access the GUI, point your browser to the management IP address of the system. After the Java applet loads, a login prompt opens, as shown in Figure 4-3.

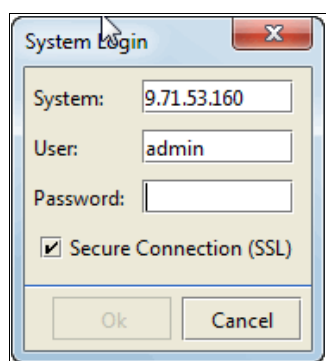


Figure 4-3 IBM FlashSystem web GUI login prompt

Enter the login and password, and click **Ok** to log in. For IBM FlashSystem storage systems, the default login is *admin* and the default password is *password*. Upon successful authentication, the main GUI window loads, as shown in Figure 4-4. We suggest that you change and record the default password to something else.

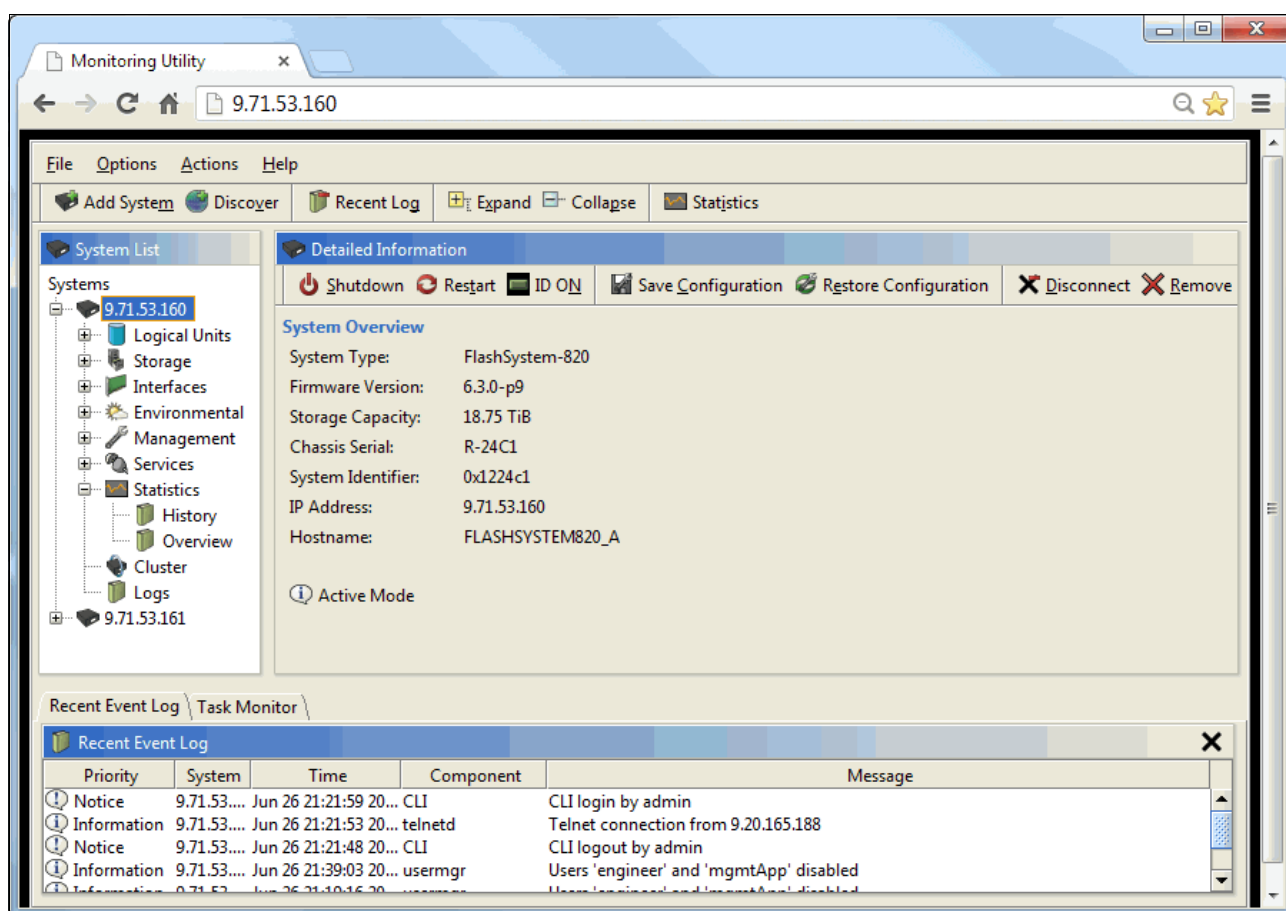


Figure 4-4 IBM FlashSystem GUI main window

Note the system tree navigation element in the left pane. The system tree consists of a root node (a labeled icon) that represents the storage system, and a nested series of nodes that represent components in the system and system management functions. You can click a node to see details and options that are related to the node.

Performing actions on nodes in the system tree

After you select a node in the system tree, you have several options for completing actions on the node. You can access available actions for a node in the system tree in several ways:

- ▶ Clicking the management interface toolbar buttons
- ▶ Right-clicking either a tree node or its pane
- ▶ Clicking the Actions menu

Each method shows a menu or a list of all available actions for the particular node in the tree.

Tip: Depending on the nature of the IBM FlashSystem configuration tasks being performed, there might be an advantage to using the GUI over the CLI, or vice versa. However, when configuring the IBM FlashSystem for use with SAN Volume Controller, there are distinct advantages to using the CLI for initial configuration tasks, specifically regarding logical unit number (LUN) and access policy creation. See 4.1.7, “Logical unit creation and access policies” on page 99 for more details.

4.1.3 IBM FlashSystem feature licenses

Certain IBM FlashSystem features are enabled via system licenses. Currently licensed features can be viewed via the **license** CLI command, as shown in Example 4-4.

Example 4-4 Viewing the installed licenses on an IBM FlashSystem via the CLI

```
admin #: license

Name-----Key-----Description-----
Configuration Restore *****6VLE Allows for save and restore of system configuration.
LDAP *****LNIA Allows login authentication via LDAP.
LU Masking *****UN8K Allows a Logical Unit to be masked to a specific host port.
Mail Service *****QSLA Mail service used to receive system alerts through email.
RAID 5 Mode *****4YEL System Storage Mode used to support RAID 5.
Statistic Log *****KSAY Activates the internal, long term logging of system statistics.
Terawatch *****6V2U Web monitoring application with multiple system support.
```

To view currently licensed features via the Management panel of the GUI, click the **Licenses** button, as shown in Figure 4-5 on page 84. Alternatively, you can click the **Management** node and type **ctrl-c** or right-click the **Management** node in the system tree and choose **Manage System Licenses...**

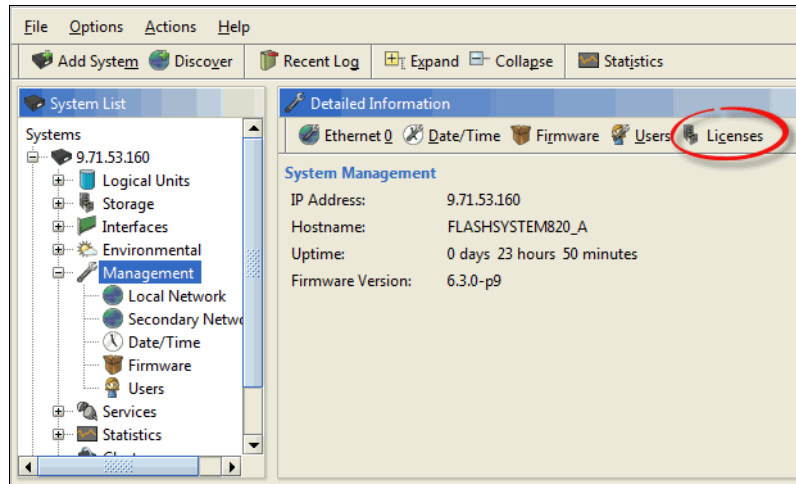


Figure 4-5 Administering system licenses in the FlashSystem GUI

Additionally, the CLI and GUI allow you to add or remove license keys from the system.

For optimal configuration when using FlashSystem storage systems with IBM SAN Volume Controller, it is recommended that the following features are licensed:

- LU Masking

The Logical Unit (LU) Masking license enables you to assign access policies to an LU on the storage system, allowing you to restrict access to specific host servers through a controller port. LU Masking is covered in greater detail in 4.1.7, “Logical unit creation and access policies” on page 99.

- RAID 5 Mode

The RAID 5 Mode license enables the RAID storage mode in the FlashSystem, allowing the system to create parity protection across flash modules to protect against the failure of an individual flash card. This is the recommended storage mode to configure when deploying the IBM FlashSystem with SAN Volume Controller. This publication assumes that your system is already configured to use the RAID 5 storage mode. For more information about FlashSystem 820 storage modes, refer to the *IBM FlashSystem 820 User's Guide*.

- Terawatch

The terawatch license allows you to manage multiple systems from a single web management interface. Although this is an optional feature, it allows for more efficient configuration and management when deploying multiple FlashSystems with SAN Volume Controller. The terawatch feature allows you to:

- Use the Add System button to log in to more systems. Use this option to manage multiple storage systems at the same time.
- Use the Discover button to start a network broadcast that discovers other systems on the network.

For more information about these licensed features, refer to the *IBM FlashSystem 820 User's Guide*.

4.1.4 Configuring additional network settings

In addition to the basic IP address settings configured in 4.1.1, “Configuring the FlashSystem management IP addresses” on page 78, we also need to configure a host name and Domain Name Service (DNS) settings for the system. The email notifications feature, described in 4.1.5, “email notifications and call home” on page 90, constructs the *from* address used in its emails by combining the system values for hostname and DNS domain, using the following syntax:

`<hostname>@<dns domain>`

Although the use of a DNS server is optional, the DNS domain value *must* be set to a valid domain in order for the email notifications feature to function properly.

Setting the management control processor host name

Each IBM FlashSystem 820 storage system management control processor has a host name, and the default value is `defaulthostname`. It is recommended that you assign a unique host name value to each management control processor.

Setting the host name using the CLI

To set the host name using the CLI, use the **network hostname** command, as shown in Example 4-5.

Example 4-5 Setting the host name using the IBM FlashSystem CLI

```
admin #: network hostname mc-1 FLASHSYSTEM820_A
FLASHSYSTEM820_A
```

```
Hostname set to 'FLASHSYSTEM820_A'
```

Note about the network CLI command: The syntax of the **network** command differs, depending on whether your IBM FlashSystem model contains a single management control processor, or dual management control processors, as in our example environment. To validate the syntax of the command for your system, use the **help network** command.

Setting the host name using the GUI

To set the management control processor host name using the GUI, click the **Ethernet 0** button on the **Local Network** node in the system tree, as shown in Figure 4-6 on page 86.

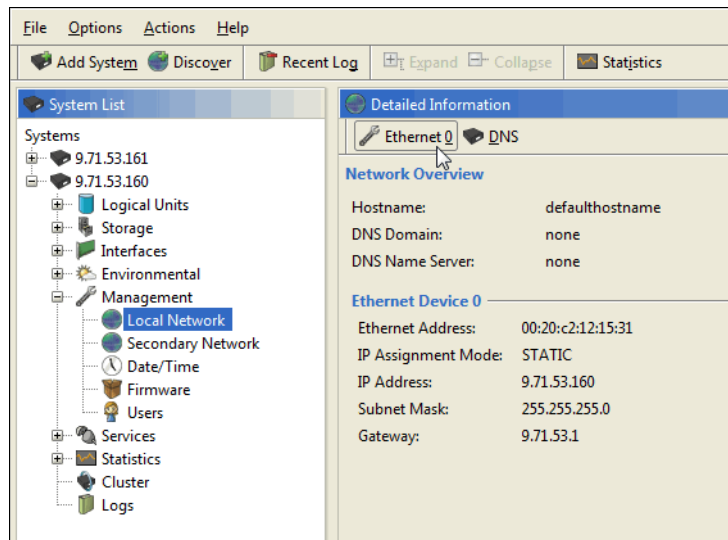


Figure 4-6 Setting the host name using the IBM FlashSystem GUI

Click **Next** on the **Overview** pop-up window, as shown in Figure 4-7.

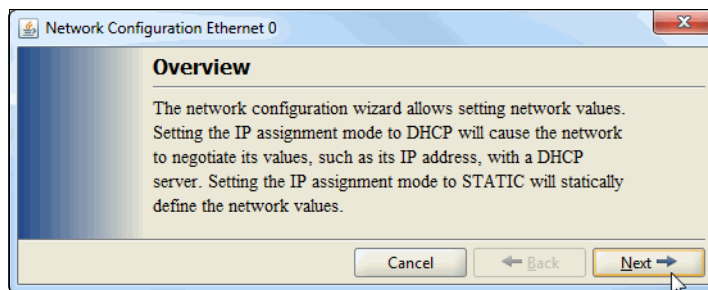


Figure 4-7 IBM FlashSystem network overview pop-up window

In the Hostname panel, specify the value that you want to use for the host name, then click **Next**, as shown in Figure 4-8.

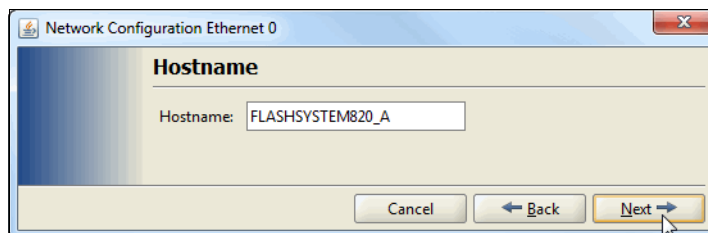


Figure 4-8 Configuring the host name in the IBM FlashSystem GUI

In the IP Address Assignment panel, as shown in Figure 4-9 on page 87, the selected value will reflect what you configured in 4.1.1, “Configuring the FlashSystem management IP addresses” on page 78. In the case of our example environment, this value is **STATIC**. You can modify this setting, as needed. Click **Next** to proceed.

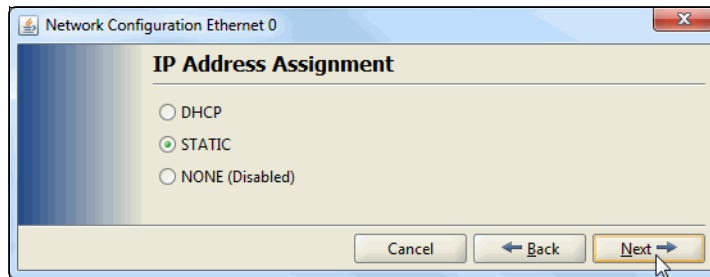


Figure 4-9 Configuring the IP address assignment setting in the IBM FlashSystem GUI

On the next window, confirm that the new Hostname value is correct, check the confirmation check box, and click **Finish** to apply the changes, as shown in Figure 4-10.

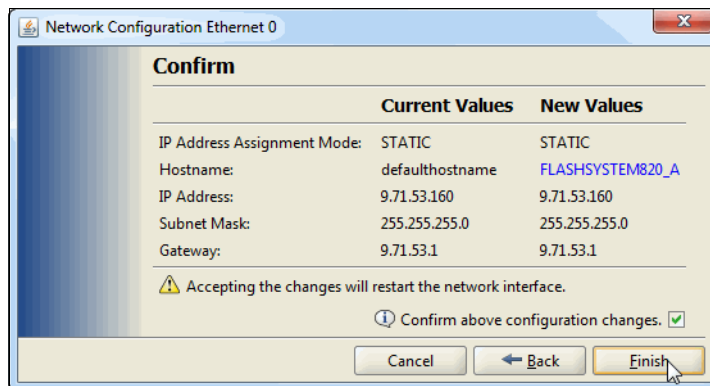


Figure 4-10 Confirming and applying the network settings in the IBM FlashSystem GUI

When you click Finish, the pop-up window closes, and you can confirm that the new Hostname value has been applied in the Local Network panel in the GUI, as shown in Figure 4-11.

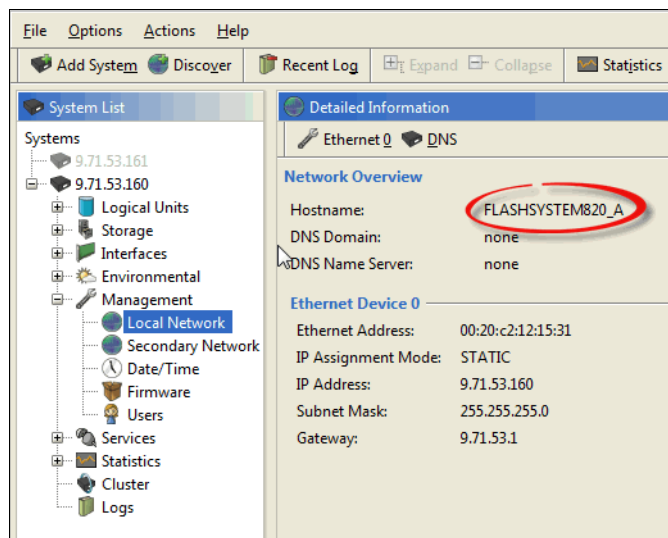


Figure 4-11 Confirming the host name setting in the IBM FlashSystem GUI

Setting the DNS domain

The DNS domain value is used in generating the *from* email address that is used by the email notifications feature, which is described in 4.1.5, “email notifications and call home” on page 90.

Setting the DNS domain using the CLI

To set the DNS domain using the CLI, use the **network dns** and **network restart** commands, as shown in Example 4-6.

Example 4-6 Setting the DNS domain using the IBM FlashSystem CLI

```
admin #: network dns domain mc-1 hurlsey.ibm.com

Network DNS search domain set to 'hurlsey.ibm.com'

admin #: network restart mc-1
Are you sure you want to restart the network? (y/n)
y
Network restarted
```

Note: When setting the DNS domain from the CLI, you must manually reset the network interface for the change to take effect, as shown in Example 4-6.

Setting the DNS domain using the GUI

To set the DNS domain using the GUI, click **DNS** on the **Local Network** node in the system tree, as shown in Figure 4-12.

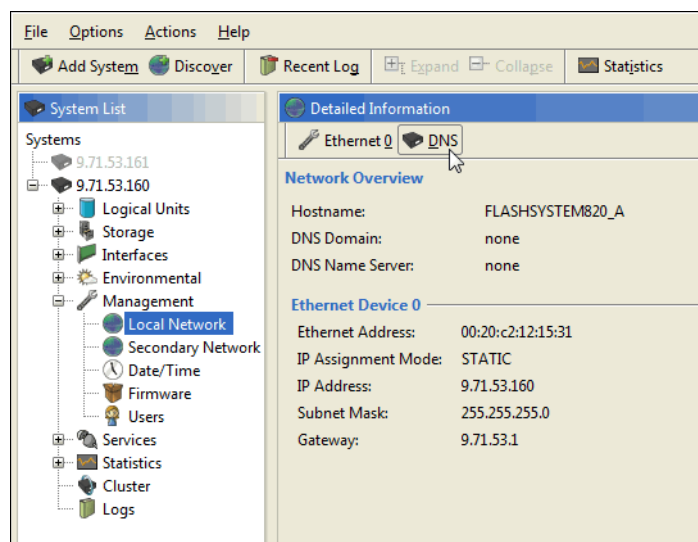


Figure 4-12 Setting the DNS domain using the IBM FlashSystem GUI

Click **Next** on the Overview pop-up window, as shown in Figure 4-13 on page 89.

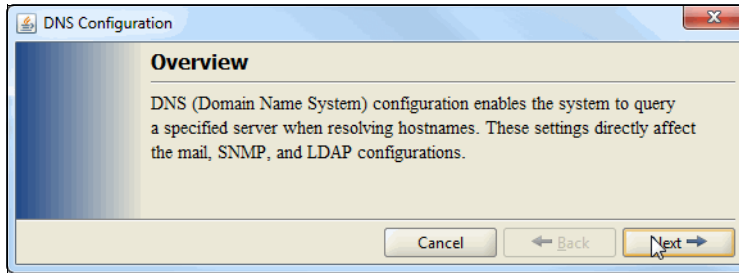


Figure 4-13 IBM FlashSystem DNS configuration overview pop-up window

On the next window, specify whether you want to automatically discover the DNS settings using DHCP, then click **Next**, as shown in Figure 4-14. In our example environment, we chose *not* to use DHCP.

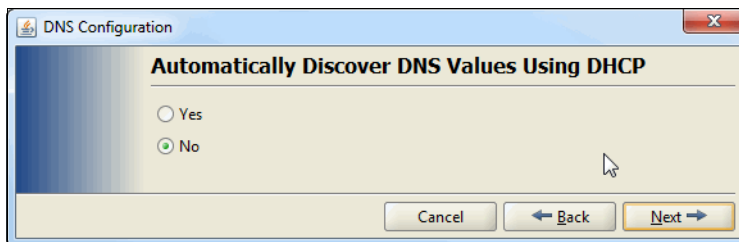


Figure 4-14 Configuring the DNS auto-discovery setting in the IBM FlashSystem GUI

In the DNS Configuration window, as shown in Figure 4-15, enter a valid **Domain** value. If your environment has a Domain Name Server, you can also enter its IP address. In our example environment, *no* Domain Name Server was available. Click **Next** to proceed.

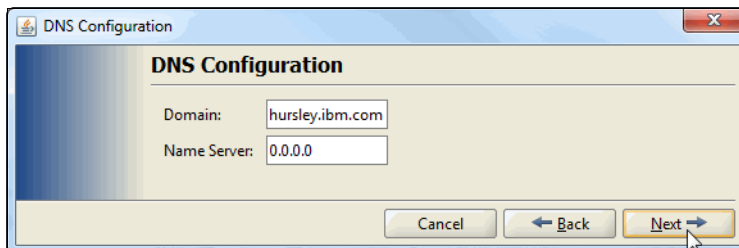


Figure 4-15 Configuring the DNS domain value in the IBM FlashSystem GUI

On the next window, confirm that the new DNS settings are correct, check the confirmation check box, and click **Finish** to apply the changes, as shown in Figure 4-16.

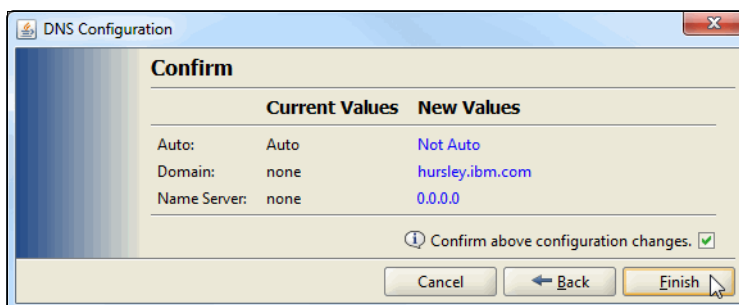


Figure 4-16 Confirming and applying the DNS settings in the IBM FlashSystem GUI

When you click **Finish**, the pop-up window closes, and you can confirm that the new DNS settings have been applied in the Local Network window in the GUI, as shown in Figure 4-17.

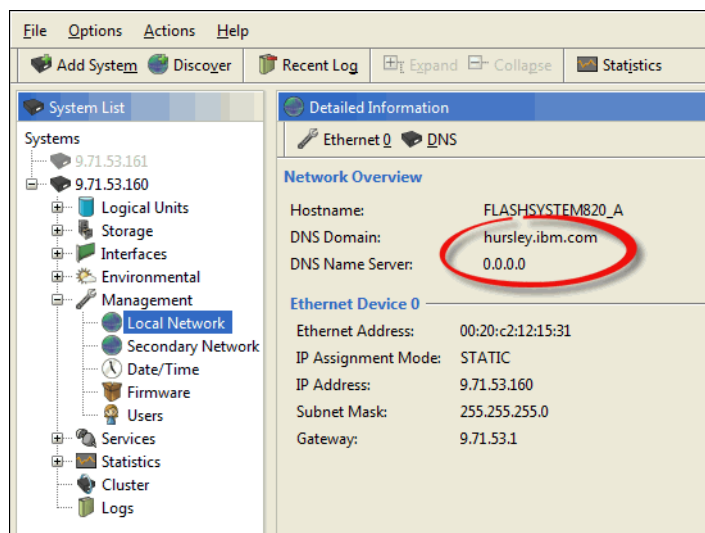


Figure 4-17 Confirming the DNS settings in the IBM FlashSystem GUI

DNS domain and host name settings are now complete, and you are ready to configure email notifications and call home.

4.1.5 email notifications and call home

It is highly recommended to configure the call home feature of the FlashSystem. This allows your system to automatically send preailure or failure notifications to the IBM Troubleshooting Ticketing system in the IBM Service Center. In addition, you can set up call home to automatically send alerts directly to you through the email notifications feature. The use of both of these features requires that the system can access an SMTP server over port 25 on your network, and that server must allow mail relaying to the sc1@vnet.ibm.com destination email address.

Configuring the call home feature

To configure call home via the CLI, use the **system callhome_config** command, with the first command parameter specifying the IP address of the mail server and the second, optional parameter specifying the *from* address, as shown in Example 4-7. Issuing the same command without any parameters will display the current call home configuration.

Example 4-7 Configuring the call home feature with the IBM FlashSystem CLI

```
admin #: system callhome_config 9.20.118.16 flashsystem820a@hursley.ibm.com
Email gateway set to '9.20.118.16'
```

```
admin #: system callhome_config
The call home feature will send emails from flashsystem820a@hursley.ibm.com via
9.20.118.16.
```


You can confirm or modify the call home heartbeat schedule, or disable the call home heartbeat feature using the **system callhome_heartbeat** command, as shown in Example 4-8.

Example 4-8 Configuring the call home heartbeat settings with the IBM FlashSystem CLI

```
admin #: system callhome_heartbeat 1 5
Call home heartbeat updated.

admin #: system callhome_heartbeat
The call home heartbeat will be sent at 5:00 every day. Full heartbeat will be
sent on Monday.

admin #: system callhome_heartbeat disable
Are you sure you want to disable 'call home'? (y/n)
y
Call home heartbeat disabled.
```

You can enable, disable, or test the event's call home feature by using the **system callhome_events** commands, as shown in Example 4-9.

Example 4-9 Configuring the call home events settings with the IBM FlashSystem CLI

```
admin #: system callhome_events disable
Are you sure you want to disable 'call home'? (y/n)
y
Call home events reporting disabled.

admin #: system callhome_events enable
Call home events reporting enabled.
admin #: system callhome_events test
Call home test message sent
```

Note about configuring the call home feature: The call home feature cannot be configured, or enabled or disabled via the GUI. You must use CLI.

Configuring the events notification feature

The next sections describe how to configure the events notification feature by using the CLI or GUI.

Configuring the events notification feature using the CLI

To configure the events notification feature via the CLI, use the **mail** command. First, enable email notifications, as shown in Example 4-10

Example 4-10 Enabling mail notifications with the IBM FlashSystem CLI

```
admin #: mail notifications enable
Mail notifications enabled
```

Next, specify the mail server to use for sending email notifications, as shown in Example 4-11.

Example 4-11 Configuring the mail notifications server with the IBM FlashSystem CLI

```
admin #: mail server 9.20.118.16
Mail server set to '9.20.118.16'
```

You can add or remove email addresses to or from the list of notification targets, or view the list of email targets, by using the **mail targets** command, as shown in Example 4-12.

Example 4-12 Configuring the mail notification targets with the IBM FlashSystem CLI

```
admin #: mail targets add myemailaddress@us.ibm.com
Added new mail target 'myemailaddress@us.ibm.com'
admin #: mail targets
mail targets
The following emails will receive system notifications:
myemailaddress@us.ibm.com
```

Finally, you can test the email notification feature by using the **mail test** command, as shown in Example 4-13.

Example 4-13 Testing the mail notifications feature with the IBM FlashSystem CLI

```
admin #: mail test
Test message sent
```

Configuring the events notification feature using the GUI

To configure the events notification feature using the GUI, go to the **Mail** window and click **Configure**, as shown in Figure 4-18. Alternatively, you can click the **Mail** node and type **ctrl-m** or right-click the **Mail** node in the system tree and choose **Configure Mail Service...**

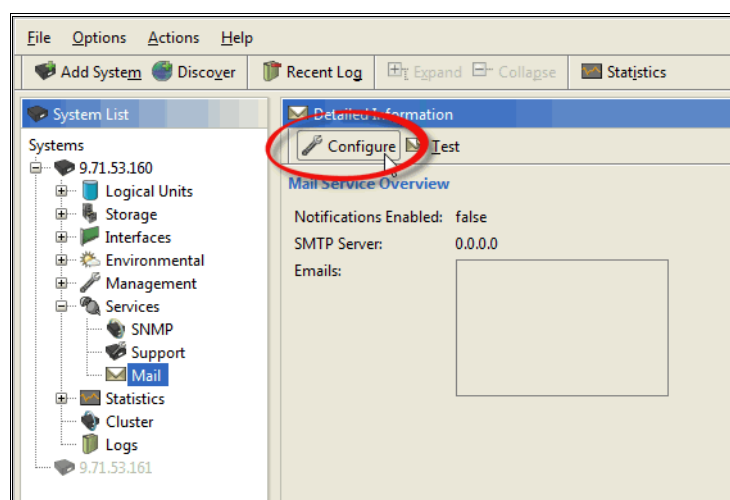


Figure 4-18 Configuring IBM FlashSystem mail notifications in the GUI

Click **Next** on the Overview pop-up window, as shown in Figure 4-19.

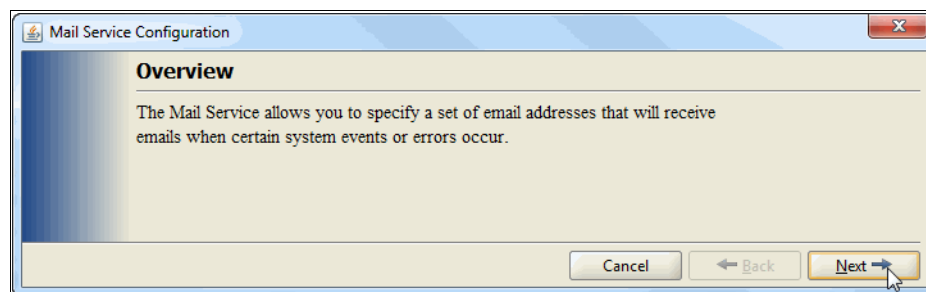


Figure 4-19 IBM FlashSystem mail notifications overview pop-up window

On the next window, ensure that the **Enable Notifications** check box is checked, and click **Next**, as shown in Figure 4-20.

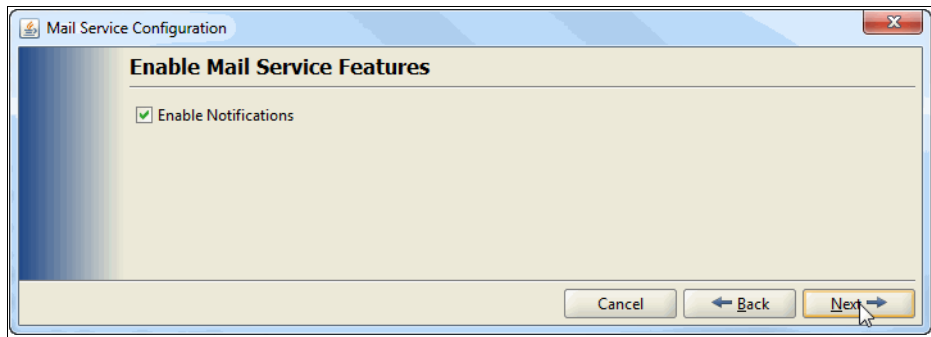


Figure 4-20 Enabling email notifications in the IBM FlashSystem GUI

In the Mail Service Settings window, specify the IP address of the mail server to use for sending email notifications, or specify that the system should automatically acquire the SMTP server address. Next, add or remove email addresses to the notifications list as required, then click **Next**, as shown in Figure 4-21.

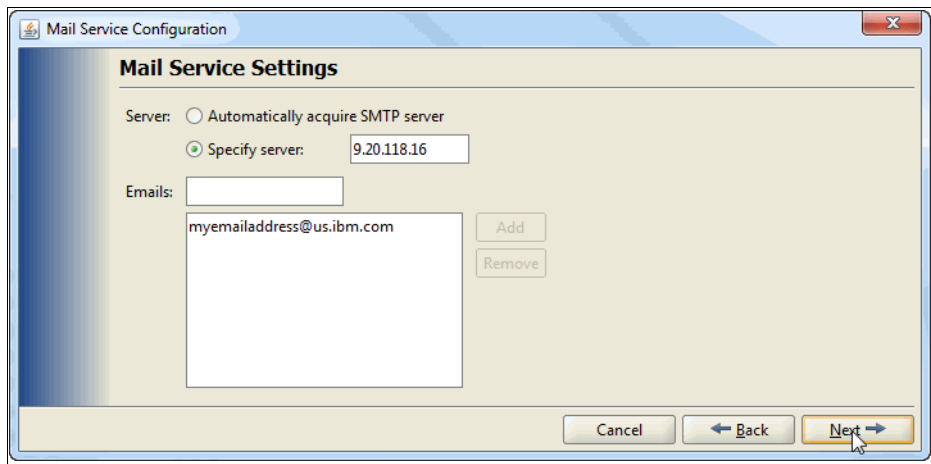


Figure 4-21 Configuring email notification settings in the IBM FlashSystem GUI

On the next window, confirm that the new values are correct, enter your FlashSystem password, and click **Finish** to apply the changes, as shown in Figure 4-22.

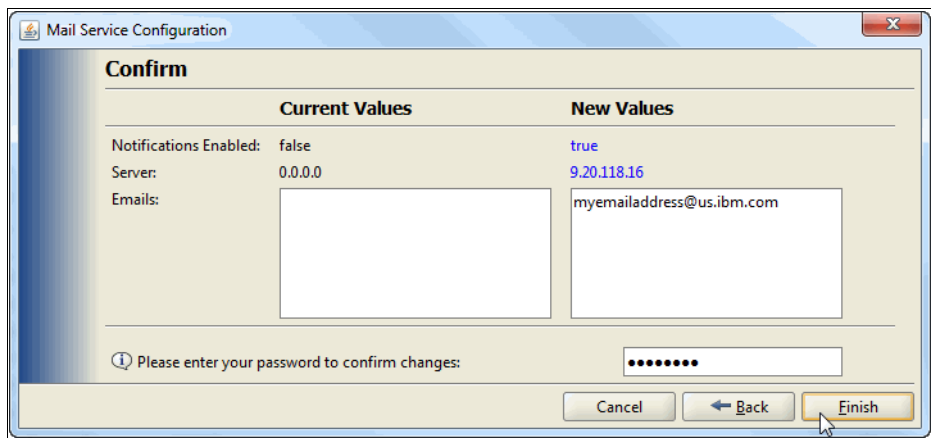


Figure 4-22 Confirming and applying email notification settings in the IBM FlashSystem GUI

When you click **Finish**, the pop-up window closes, and you can confirm that the new email notification settings have been applied in the Mail panel in the GUI, as shown in Figure 4-23.

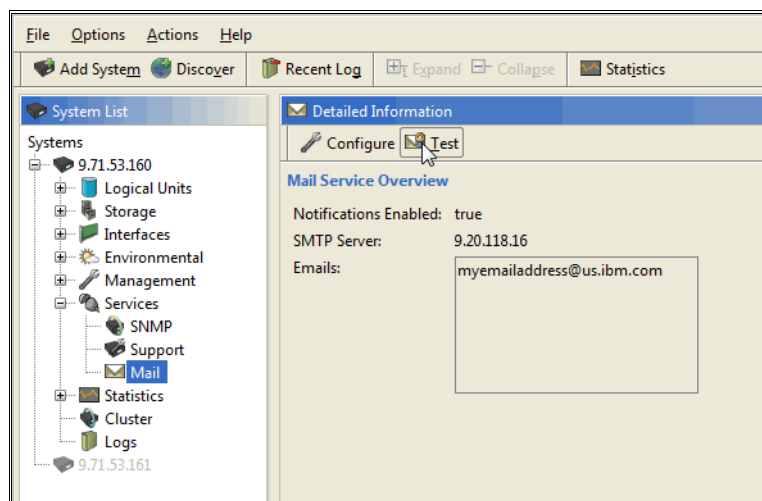


Figure 4-23 Confirming and testing email notification settings in the IBM FlashSystem GUI

Click **Test** on the Mail panel to send a test notification email, as shown in Figure 4-23. Alternatively, you can click the **Mail** node and type **ctrl-t** or right-click the **Mail** node in the system tree and choose **Test Mail Service...** A pop-up window indicates the results of the email notifications test, as shown in Figure 4-24.

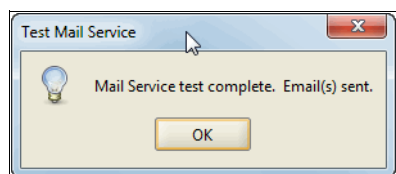


Figure 4-24 Successful email notifications test in the IBM FlashSystem GUI

4.1.6 Fibre Channel port settings

IBM FlashSystem storage systems support either 8 Gb Fibre Channel (FC) or quadruple data rate (QDR) InfiniBand interfaces. In our example environment, our FlashSystem 820 storage systems are each configured with four 8 Gb Fibre Channel ports. Each FlashSystem FC port supports the following settings.

Link Speed:

- ▶ Auto
- ▶ 2 Gb
- ▶ 4 Gb
- ▶ 8 Gb

Topology:

- ▶ Point-to-point (PP)
- ▶ Arbitrated Loop (AL, where either soft or hard loop ID assignment is supported)
- ▶ Auto

For optimal use with SAN Volume Controller, where a private SAN fabric is used to connect the FlashSystem storage systems and the SAN Volume Controller nodes, it is recommended

to configure both the FlashSystem FC ports and the corresponding SAN switch ports to the highest mutually supported speed setting, and to a mutually supported topology setting. In the case of our example environment, these settings are 8 Gb and PP, respectively.

Configuring FC port settings using the CLI

To configure FC ports using the IBM FlashSystem CLI, use the **fc** command. To view the detailed status of a port, use the **fc status** command, as shown in Example 4-14.

Example 4-14 Viewing the status of an FC controller port with the IBM FlashSystem CLI

```
admin #: fc status fc-1a
```

```
---fc-1a status---
```

```
Port Name      20:04:00:20:c2:12:24:c1
Node Name      10:00:00:20:c2:12:24:c1
Port ID        000000
Link State      OFFLINE
Link Speed      NONE
Topology        NONE
```

```
---fc-1a configuration---
```

```
Link Speed      4Gb
Topology        AUTO
```

To view or set the speed of a port, issue an **fc speed** command, as shown in Example 4-15.

Example 4-15 Setting and viewing the speed of an FC controller port in the IBM FlashSystem CLI

```
admin #: fc speed fc-1a 8Gb
```

```
Channel fc-1a's link speed has been changed to 8Gb
```

```
admin #: fc speed fc-1a
fc speed fc-1a
```

```
Configuration: 8Gb
Current State: NONE
```

```
Supported link speeds:
AUTO
2Gb
4Gb
8Gb
```

To view or set the topology of a port, issue an **fc topology** command, as shown in Example 4-16.

Example 4-16 Setting and viewing the topology of an FC controller port in the IBM FlashSystem CLI

```
admin #: fc topology fc-1a PP
```

```
Channel fc-1a's link topology has been changed to 'PP'
```

```
admin #: fc topology fc-1a
fc topology fc-1a
```

```
Configuration: PP
Current State: NONE
```

Supported link topologies:
PP
AL
AUTO

Note: After modifying the speed and topology settings of a FlashSystem FC port, you must issue an **fc link_reset** command to reset the port in order for it to log in to the switch with the new settings, as shown in Example 4-17.

Example 4-17 Resetting the link of an FC controller port in the IBM FlashSystem CLI

```
admin #: fc link_reset fc-1a
Are you sure you want to reset 'fc-1a'? (y/n)
y
```

The link has been reset on channel fc-1a

```
admin #: fc status fc-1a
```

```
---fc-1a status---
```

Port Name	20:04:00:20:c2:12:24:c1
Node Name	10:00:00:20:c2:12:24:c1
Port ID	000000
Link State	OFFLINE
Link Speed	NONE
Topology	NONE

```
---fc-1a configuration---
```

Link Speed	4Gb
Topology	PP

Issuing an **fc** command without any arguments outputs the high-level status of the FC ports in a FlashSystem. The high-level status of the FC ports for the two IBM FlashSystem storage systems (A and B) in our example environment are shown in Example 4-18 and Example 4-19 on page 97, respectively.

Example 4-18 FC port settings for the example IBM FlashSystem A used in this book

```
admin #: fc
fc
```

Controller	State	Firmware	Serial
fc-1	GOOD	6309	R-1C91
fc-2	GOOD	6309	R-1C92

Channel	Link_State	Link_Speed	Topology
fc-1a	ONLINE	8Gb	F_PORT
fc-1b	ONLINE	8Gb	F_PORT
fc-2a	ONLINE	8Gb	F_PORT
fc-2b	ONLINE	8Gb	F_PORT

Example 4-19 FC port settings for the example IBM FlashSystem B used in this book

admin #: fc

Controller	State	Firmware	Serial
fc-1	GOOD	6309	R-1C8F
fc-2	GOOD	6309	R-1C90

Channel	Link_State	Link_Speed	Topology
fc-1a	ONLINE	8Gb	F_PORT
fc-1b	ONLINE	8Gb	F_PORT
fc-2a	ONLINE	8Gb	F_PORT
fc-2b	ONLINE	8Gb	F_PORT

Configuring FC port settings using the GUI

To configure FC ports using the IBM FlashSystem GUI, go to the **Interfaces** node in the system tree and expand the node to see the FC controllers. Expand the corresponding FC controller node to see the ports for that controller, and click the node for the port that you want to configure. This will load the details window for that port. Next, click **Configure**, as shown in Figure 4-25. Alternatively, you can click the FC port node in the system tree and type **ctrl-c** or right-click the node in the system tree and choose **Configure...**

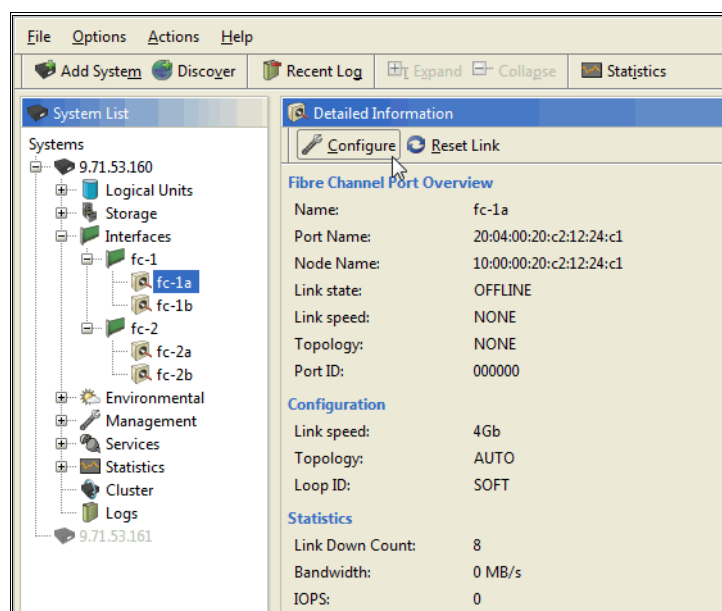


Figure 4-25 Configuring FC port details in the IBM FlashSystem GUI

The Overview pop-up window that appears provides some details about configuring FC port settings. When you are ready to proceed, click **Next**, as shown in Figure 4-26 on page 98.

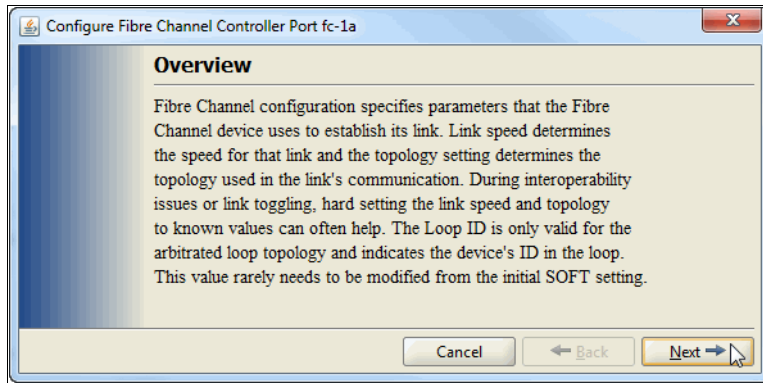


Figure 4-26 IBM FlashSystem FC port configuration overview pop-up window

In the Controller Port Configuration window, configure the settings that are appropriate to your environment, and then click **Next**, as shown in Figure 4-27. In our example environment, these settings are 8 Gb (speed) and PP (topology).

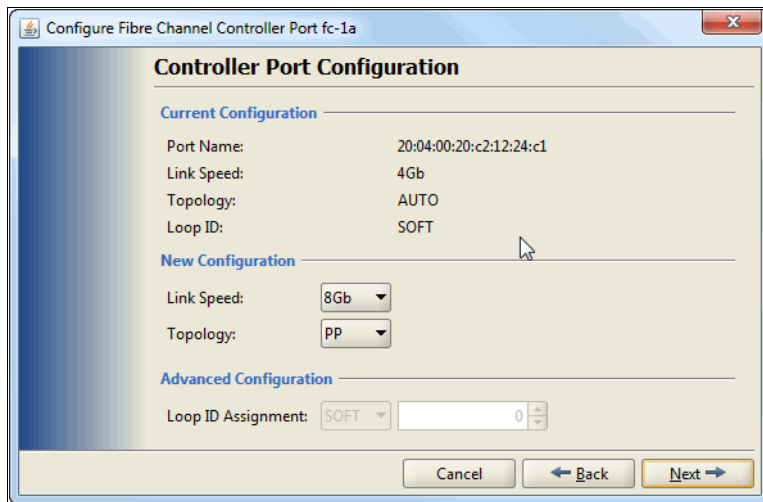


Figure 4-27 Configuring FC controller port settings in the IBM FlashSystem GUI

On the next window, confirm that the new values are correct, enter your FlashSystem password, and click **Finish** to apply the changes, as shown in Figure 4-28.

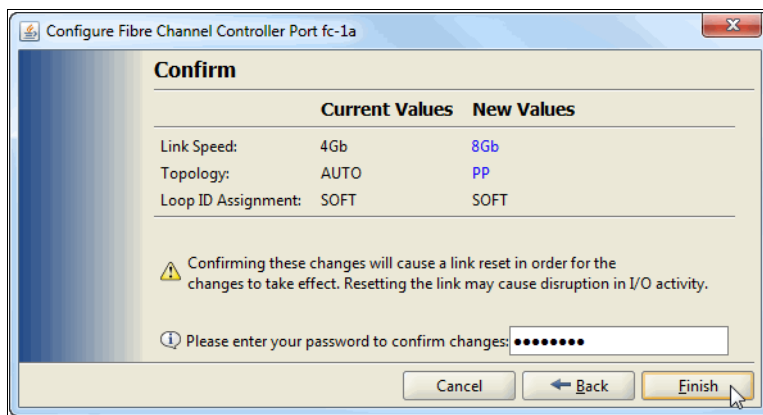


Figure 4-28 Confirming and applying FC controller port settings in the IBM FlashSystem GUI

When you click **Finish**, the pop-up window closes, and you can confirm that the new FC controller port settings have been applied in the port details panel in the GUI, as shown in Figure 4-29.

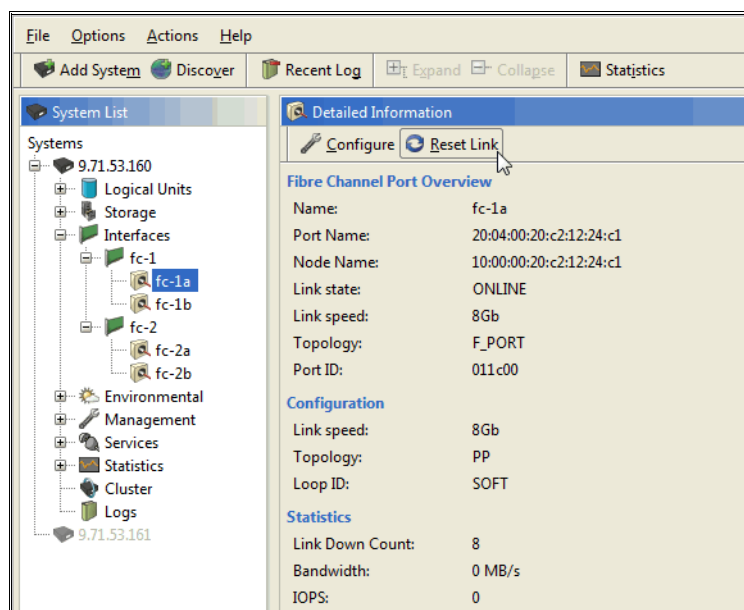


Figure 4-29 Confirming FC controller port settings in the IBM FlashSystem GUI

Note: When configuring FC controller port settings in the IBM FlashSystem GUI, the FC port link is automatically reset when you click **Finish** on the confirmation window. It might take several moments for the main window to display the updated port settings. If the updated settings do not appear, you can manually reset the FC port link by clicking **Reset Link** in the main window, as shown in Figure 4-29.

If you click **Reset Link** in the main window, a pop-up window appears, as shown in Figure 4-30. Enter your FlashSystem password and click **Yes** to reset the corresponding FC port link.

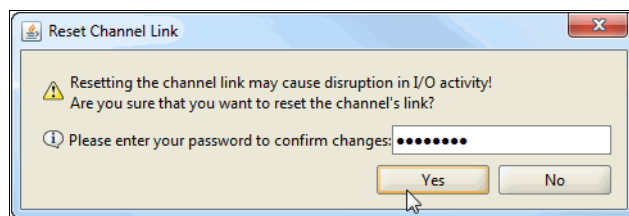


Figure 4-30 Manually resetting the FC controller port link in the IBM FlashSystem GUI

4.1.7 Logical unit creation and access policies

This section covers the steps that are required to create LUNs and associated access policies on the IBM FlashSystem. Concise examples are provided to help you understand the steps that are required. For a complete list of the commands that are used to configure the LUNs and access policies of both IBM FlashSystem 820 storage systems used with SAN Volume Controller in the example environment of this book, refer to Appendix A, “FlashSystem CLI commands used in the example environment” on page 129.

For more information about logical units and access policies, refer to the *IBM FlashSystem 820 User's Guide* and the *IBM FlashSystem Integration Guide*.

Creating LUNs using the CLI

As with any back-end storage array, in order to present usable storage from the FlashSystem to the SAN Volume Controller, you must create logical units on the system. For more information about the best practices for FlashSystem LUN creation for use with SAN Volume Controller, refer to 4.3, “SAN Volume Controller MDisk configuration” on page 111.

To create a LUN on the IBM FlashSystem by using the CLI, use the **lu create** command, as shown in Example 4-20. To view the details of the created LUN, use the **lu info** command.

Example 4-20 Creating a LUN by using the IBM FlashSystem CLI

```
admin #: lu create FLASHSYSTEM_A_SVC_1 1199g 0
Logical Unit 'FLASHSYSTEM_A_SVC_1' created with number 0 and size 1.17 TiB

admin #: lu info FLASHSYSTEM_A_SVC_1
Logical Unit 'FLASHSYSTEM_A_SVC_1'
      Size:      1.17 TiB
      Number:    0
      Device ID:
      State:     Good
      Offset:    0.00 B
      Sector Size: 512.00 B
```

Creating access policies using the CLI

As discussed in 4.1.3, “IBM FlashSystem feature licenses” on page 83, the LUN Masking feature of the IBM FlashSystem allows you to restrict access of the system LUNs to specific hosts. LUN Masking is accomplished on the FlashSystem through the creation of access policies. An access policy defines what host IDs can access a LUN through each FlashSystem FC controller port. The IBM FlashSystem supports creating logical aliases that map to specific host IDs. The use of aliases can speed the creation of access policies.

To create a host ID alias using the CLI, use the **lu alias add** command, as shown in Example 4-21. Further, to view the existing aliases using the CLI, use the **lu alias** command.

Example 4-21 Creating an alias using the IBM FlashSystem CLI

```
admin #: lu alias add SVC_ND_1_P1 50:05:07:68:01:40:ba:da
Added alias 'SVC_ND_1_P1' for host '50:05:07:68:01:40:ba:da'
```

```
admin #: lu alias
```

```
-----Port Aliases-----

-----Alias-----Host ID-----
SVC_ND_1_P1          50:50:07:68:01:40:ba:da
```

An advantage to using the CLI over the GUI for creating access policies is the ability to create access policy groups. An *access policy group* is simply a collection of FlashSystem port and host ID pairs with a logical name. By using groups, you can quickly add or remove access for a specific host or group of hosts to a LUN. To create an access policy group using the CLI, use the **lu access group add** command, as shown in Example 4-22 on page 101. The same

command can be repeated with additional FlashSystem port and host ID pairs to add additional entries to the access policy group. To view the entries of an access policy group, use the **lu access group** command, also shown in Example 4-22.

Example 4-22 Creating an access policy group using the IBM FlashSystem CLI

```
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P1_FC-1A fc-1a 50:05:07:68:01:40:ba:da
Added Access Policy Entry 'SVC_ND_1_P1_FC-1A' to Group 'SVC_FLASHSYSTEM'

admin #: lu access group

-----Policy Group Table-----

-Group- -Entry- --Controller-- -----Host ID-----
SVC_FLASHSYSTEM
      SVC_ND_1_P1_FC-1A
              fc-1a              50:05:07:68:01:40:ba:da
```

After you have defined your access policy group, it is a simple step to add the access policy group to the desired LUN, effectively allowing access to the LUN based on the ports and host IDs defined in the group. To do so, use the **lu access add group** command, as shown in Example 4-23.

Example 4-23 Creating an access policy using the IBM FlashSystem CLI

```
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_1
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_1'
```

Tip: The FlashSystem CLI offers a distinct advantage over the GUI regarding access policy configuration through the access policy group feature. Using this feature when configuring the IBM FlashSystem with SAN Volume Controller can speed your deployment.

Creating LUNs and access policies using the GUI

To create a LUN on the IBM FlashSystem using the GUI, click the **Logical Units** node in the system tree, then click **Create**, as shown in Figure 4-31.

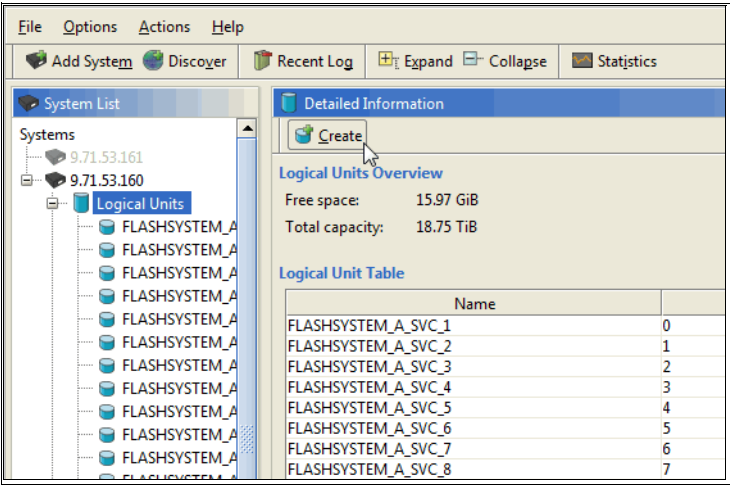


Figure 4-31 Creating a LUN by using the IBM FlashSystem GUI

The Overview pop-up window that appears provides some details about creating logical units. When you are ready to proceed, click **Next**, as shown in Figure 4-32 on page 102.

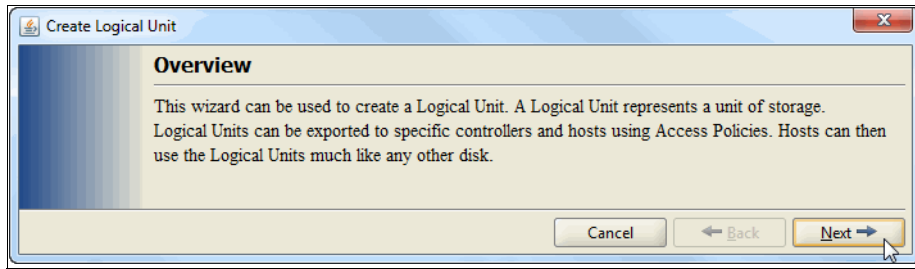


Figure 4-32 IBM FlashSystem logical unit creation overview pop-up window

In the Setup parameters window, enter the details for your LUN, then click **Next**, as shown in Figure 4-33. See 4.3, “SAN Volume Controller MDisk configuration” on page 111 for details about optimal FlashSystem LUN settings for use with SAN Volume Controller. For more information about configuring logical units, refer to the *IBM FlashSystem 820 User's Guide*, and the *IBM FlashSystem Integration Guide*.

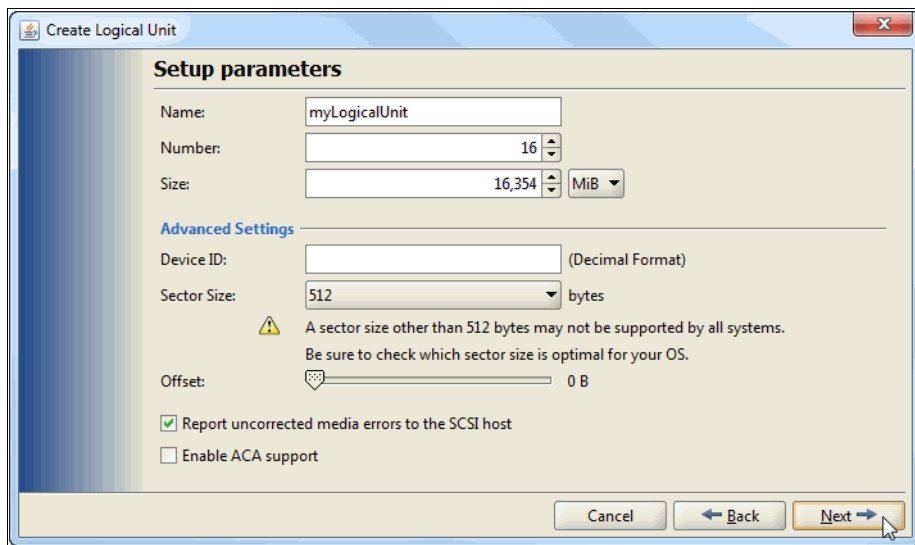


Figure 4-33 Configuring logical unit settings in the IBM FlashSystem GUI

On the next window, confirm that the configuration parameters of your LUN are correct, click the confirmation check box, and click **Finish** to create the LUN, as shown in Figure 4-34 on page 103.

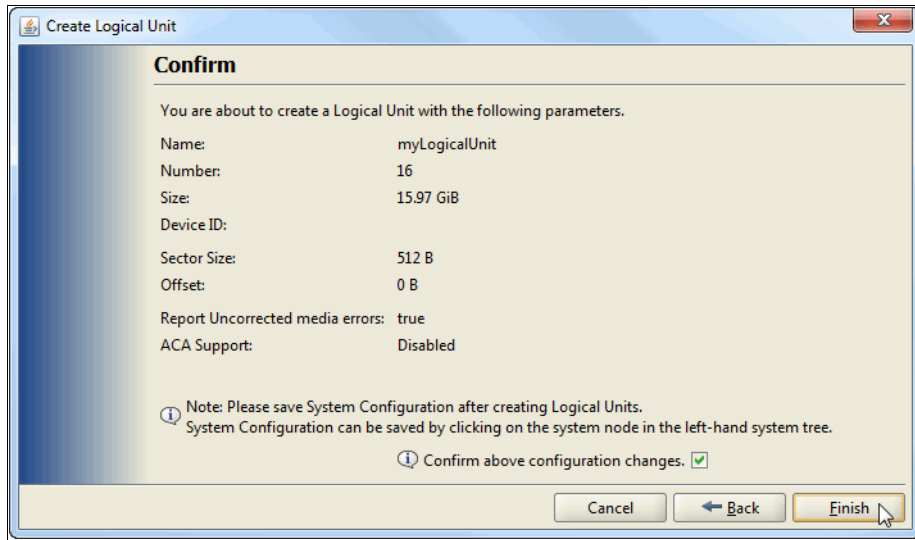


Figure 4-34 Confirming and applying logical unit settings in the IBM FlashSystem GUI

When you click **Finish**, the pop-up window closes, and you can confirm that the new LUN has been created in the Logical Units node of the system tree in the GUI, as shown in Figure 4-35. The LUN is highlighted in yellow, indicating that there is no host access that is configured for the LUN.

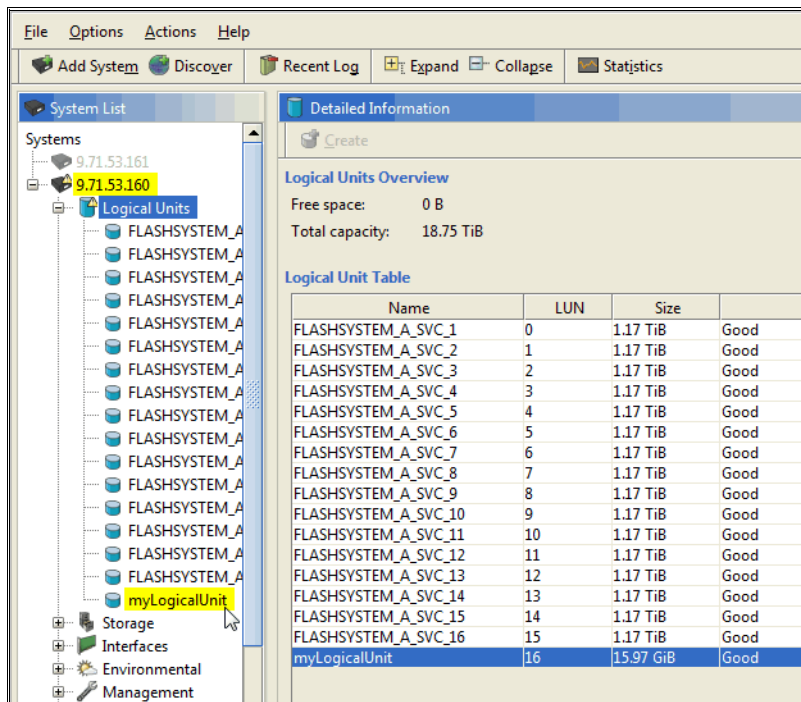


Figure 4-35 Confirming logical unit creation in the IBM FlashSystem GUI

Click the new LUN in the system tree to see and confirm its properties, as shown in Figure 4-36 on page 104. To configure host access to the new LUN, click **Access** in the details panel. Alternatively, you can click the LUN in the system tree and type **ctrl-s** or right-click the LUN in the system tree and choose **Logical Unit Access...**

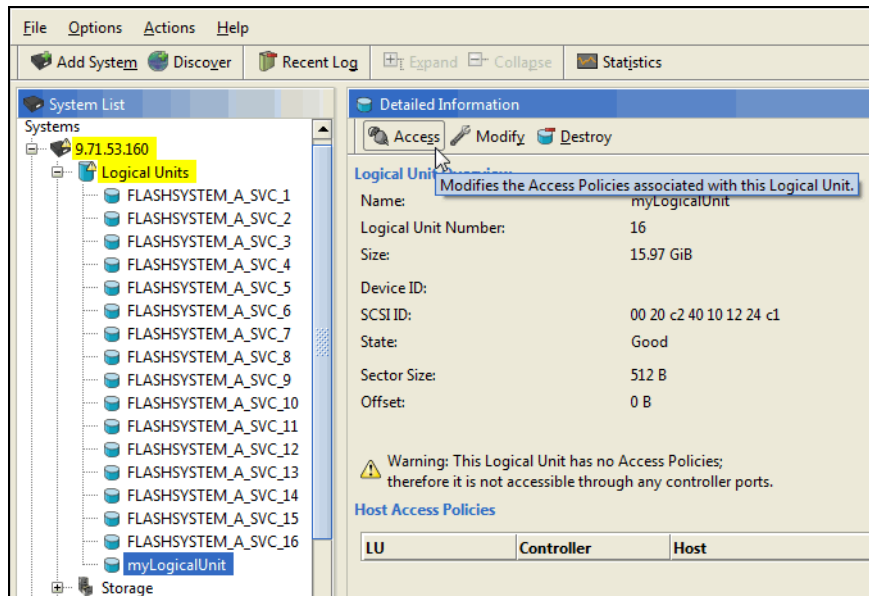


Figure 4-36 Confirming logical unit creation and configuring access in the IBM FlashSystem GUI

The Overview pop-up window that appears provides some details about configuring LUN access policies. When you are ready to proceed, click **Next**, as shown in Figure 4-37.

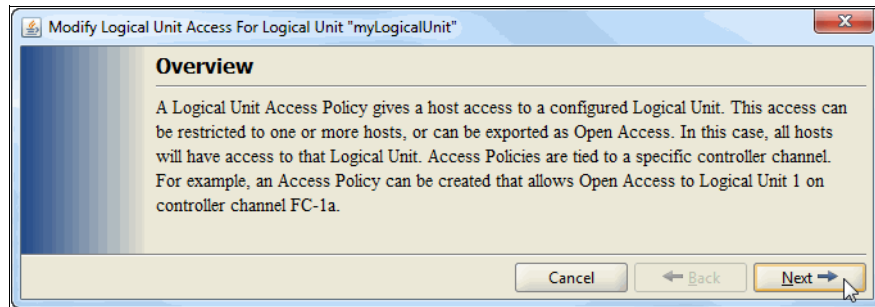


Figure 4-37 IBM FlashSystem Logical Unit Access overview pop-up window

The Modify Access Policies window that is shown is used to configure access to the LUN for the desired FlashSystem ports and host IDs. To simplify configuration, use the Alias feature to create logical aliases for the host IDs. Click **New Alias**, as shown in Figure 4-38 on page 105. For more information about configuring logical unit access policies, refer to the *IBM FlashSystem 820 User's Guide*, and the *IBM FlashSystem Integration Guide*.

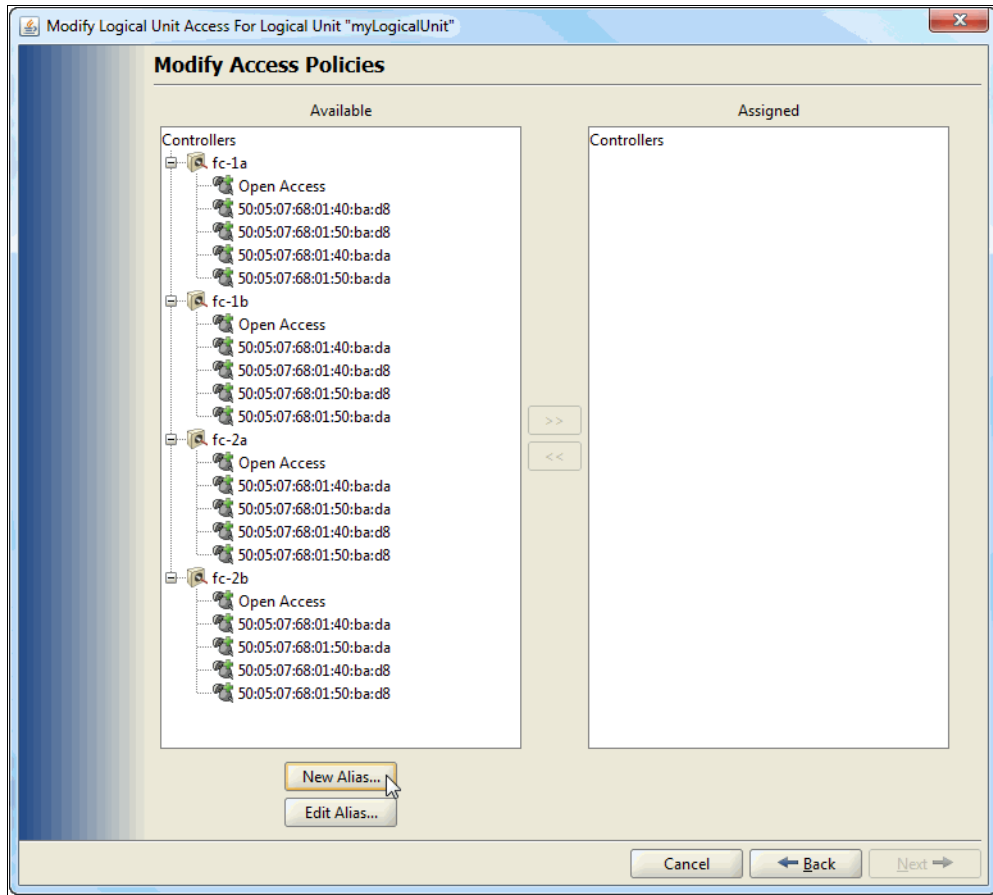


Figure 4-38 IBM FlashSystem Modify Access Policies pop-up window

In the Add Host ID Alias pop-up window that is displayed, enter the host ID and a logical name (alias) to represent that host ID, then click **Add**, as shown in Figure 4-39.

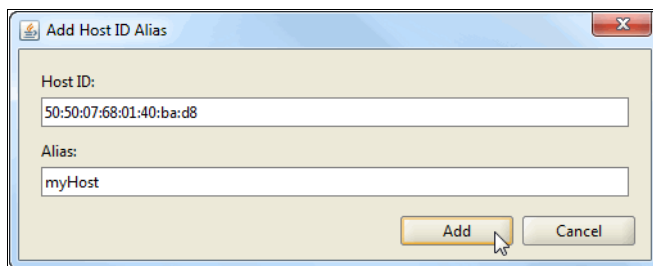


Figure 4-39 Adding a host ID alias in the IBM FlashSystem GUI

The Modify Access Policies window now displays the alias in parenthesis next to the host ID for all corresponding FlashSystem FC controller ports, as shown in Figure 4-40 on page 106. Highlight the host ID entries that correspond to the host IDs and FlashSystem ports that you want to provide with access to this LUN and click the >> button.

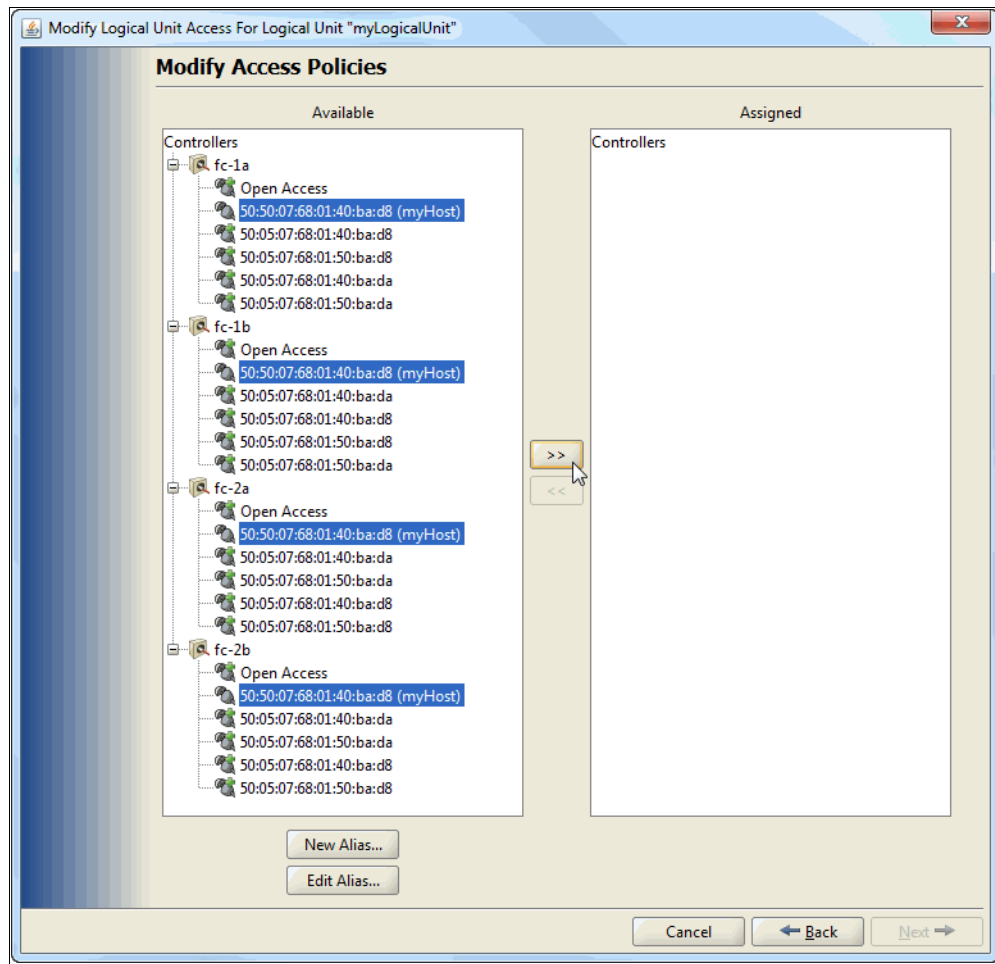


Figure 4-40 Selecting host ID aliases for LUN access in the IBM FlashSystem GUI

The Modify Access Policies window now displays the selected host ID aliases in green in the *Assigned* pane, as shown in Figure 4-41 on page 107. When you have finished configuring the desired access, click **Next**.

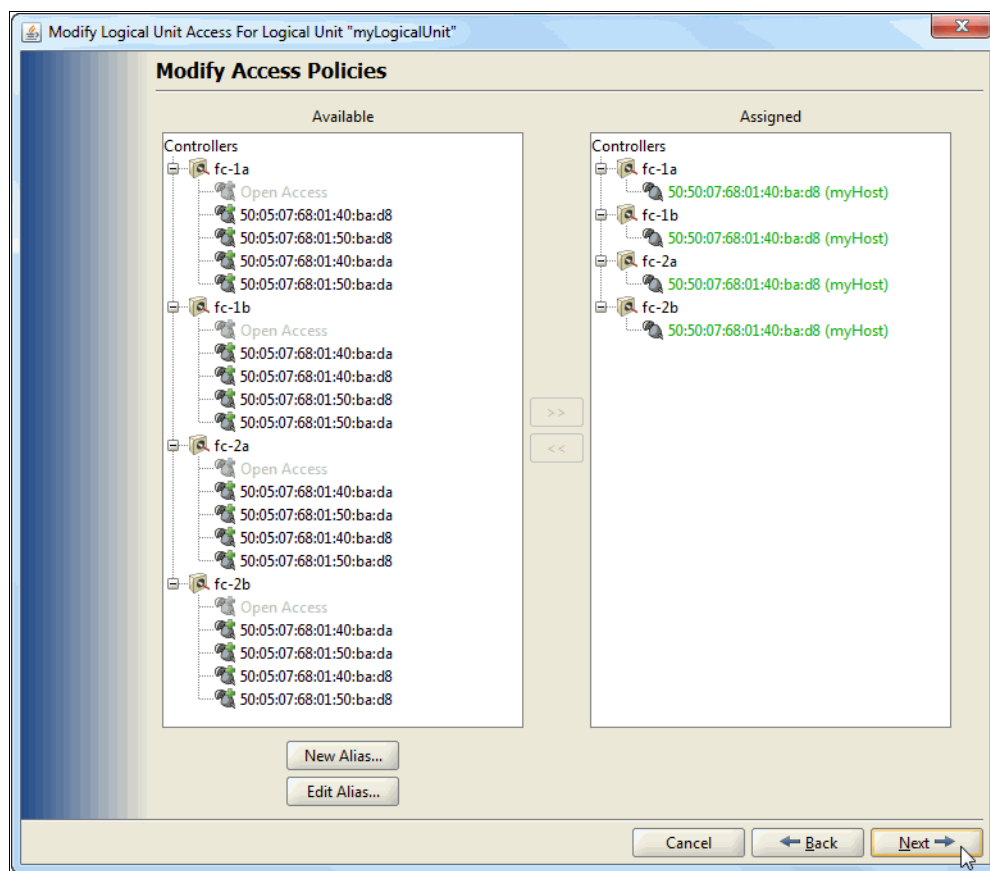


Figure 4-41 LUN access policy configured in the IBM FlashSystem GUI

On the next window, confirm that the access policy configuration matches your requirements, click the confirmation check box, and click **Finish** to create the LUN access policy, as shown in Figure 4-42.

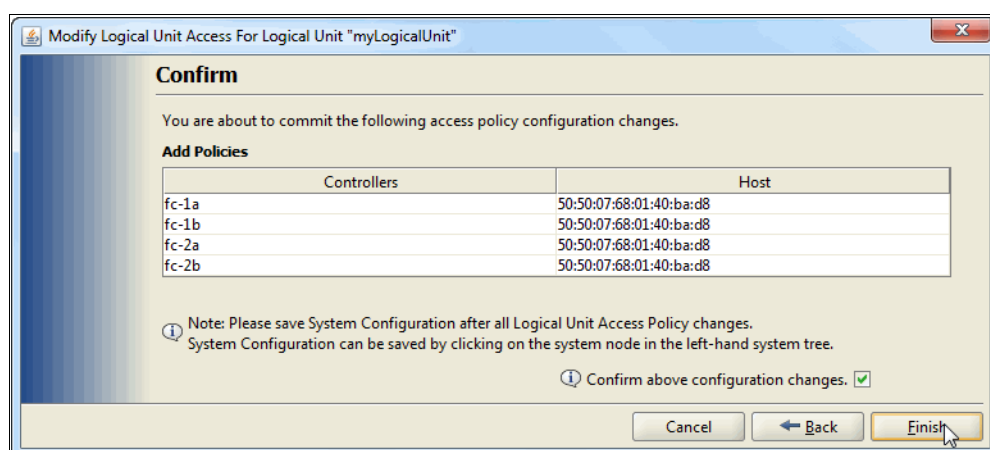


Figure 4-42 Confirming and applying logical unit access settings in the IBM FlashSystem GUI

When you click **Finish**, the pop-up window closes, and you can confirm that the host access to the LUN has been created in the details panel for the LUN, as shown in Figure 4-43 on page 108. The LUN should no longer be highlighted in yellow, indicating that host access is now configured for the LUN.

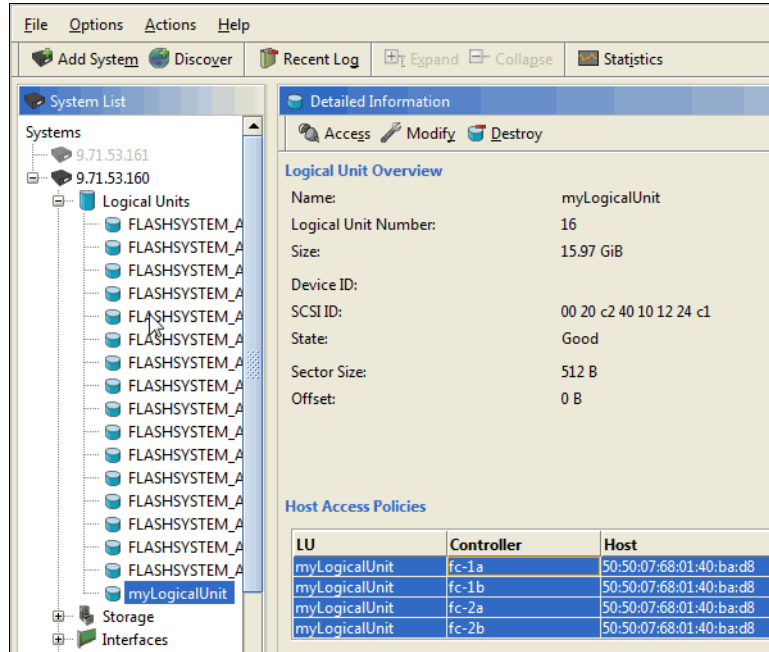


Figure 4-43 Confirming logical unit access in the IBM FlashSystem GUI

Note: When configuring LUN access policy settings in the IBM FlashSystem GUI, it might take several moments for the main window to display the updated access settings.

At this point, all the steps that are required to configure the IBM FlashSystem 820 for use with the SAN Volume Controller have been reviewed. To review the comprehensive set of CLI commands executed to configure our two example FlashSystem 820 storage systems, refer to Appendix A, “FlashSystem CLI commands used in the example environment” on page 129.

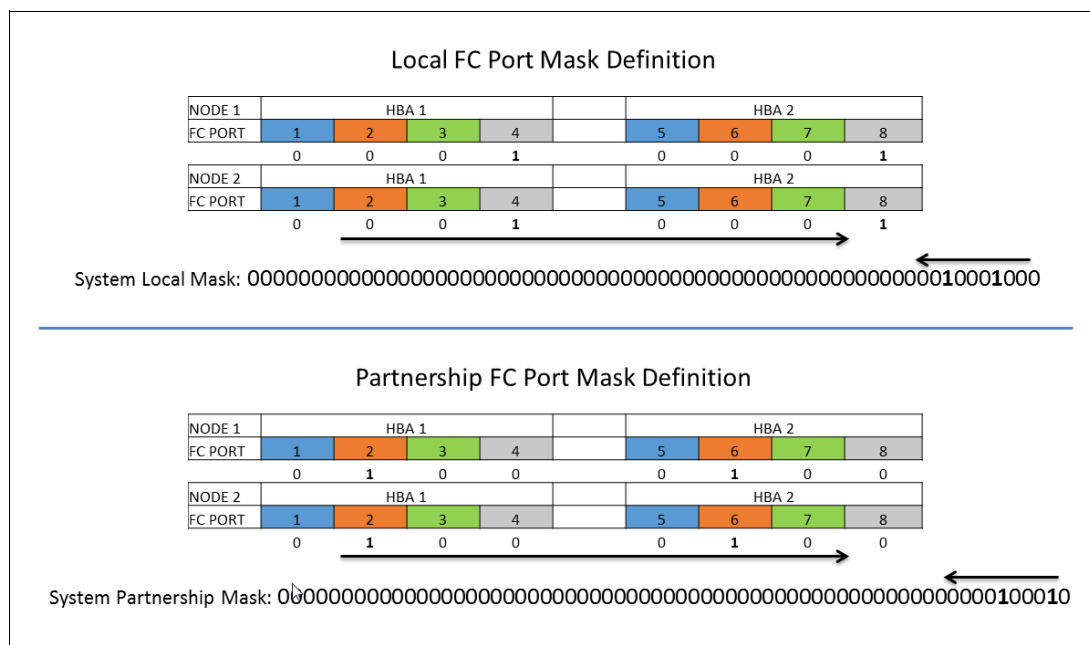
4.2 Port masking and SAN zoning configuration

This section provides information about how the Fibre Channel port masking can be configured, and together with the SAN zoning, establish communication and policies for data isolation. More information about SAN Volume Controller Fibre Channel port masking can be found in Chapter 2, “Usage considerations and scenarios” on page 41.

In this section, we assume that the SAN Volume Controller software is at level 7.1, and the SAN Volume Controller CG8 nodes have two 4-port adapters, for a total of eight Fibre Channel ports per node.

4.2.1 SAN Volume Controller Fibre Channel port masking setup

In the test environment, it was chosen to restrict local (node-to-node) communication to ports 4 and 8 in each node, and partnership (replication) data traffic to ports 2 and 6. Figure 4-44 on page 109 shows a diagram of the Fibre Channel port masks created to help implement this.



At SAN Volume Controller software level 7.1, Fibre Channel port masking is configured through the CLI. Example 4-24 shows how to use the **chsystem** command to set the local (node-to-node) Fibre Channel port masking, as defined in Figure 4-44.

[illegible]

Example 4-25 Using SAN Volume Controller CLI to set partnership Fibre Channel port masking

[illegible]

4.2.2 Host Fibre Channel port masking setup

Additionally to the *local* and *partnership* Fibre Channel port masking introduced with the SAN Volume Controller software level 7.1, it is also supported to have *host* port masking as well. Host port masking is set when a host object is created in the SAN Volume Controller or it can be changed later, and specifies which node target ports the host can access. Using this port mask, worldwide port names (WWPNs) on the host object must access volumes through the node ports that are included in the mask only.

In the test environment, it was chosen to restrict host-to-node communication from the host called *IBMX3650* to ports 3 and 7 in each node. Figure 4-45 shows a diagram of the host Fibre Channel port mask created to help implement this.

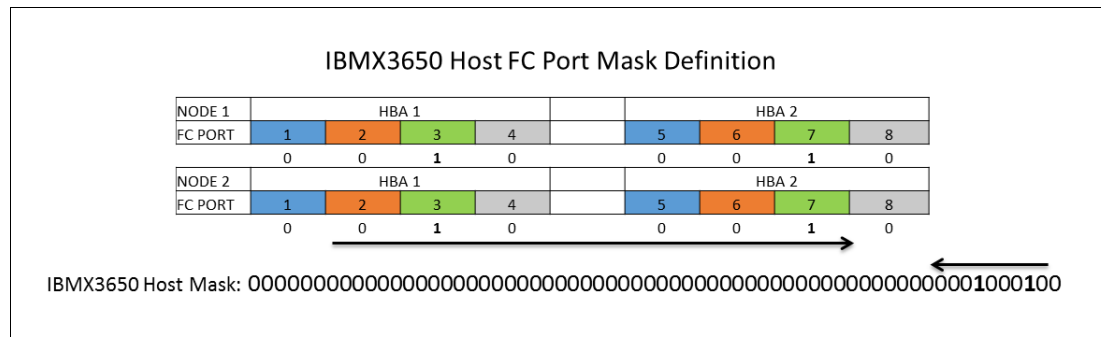


Figure 4-45 Host Fibre Channel port masking definition for the test environment

Host Fibre Channel port masking is configured through the CLI. Example 4-26 shows how to list current host properties with the **lshost** command, and then change the host port mask using the **chhost** command.

Example 4-26 Using SAN Volume Controller CLI to set host Fibre Channel port masking

[illegible]

4.2.3 SAN zoning setup

The SAN zoning complements Fibre Channel port masking in order to isolate data communication between the SAN Volume Controller nodes and the back-end storage subsystems. In the test environment, it was chosen to dedicate ports 1 and 5 in each node for the SAN Volume Controller to communicate with the two FlashSystem 820 subsystems. Table 4-1 on page 111 lists the entire schema for the test environment.

Table 4-1 SAN Volume Controller port isolation schema for the test environment

Ports in each node	Data transfer type	Method of enforcement
1, 5	Back-end storage (FlashSystem 820)	Public storage zone
2, 6	Replication (partnership)	Partnership port mask
3, 7	Host (IBM System x3650)	Host port mask and public zone
4, 8	Node-to-node (local)	Local port mask and private zone

Note: When planning the SAN zoning to establish communication between the SAN Volume Controller nodes with software at level 7.1, and the back-end storage subsystems, keep in mind that node ports 7 and 8 cannot be used. However, those ports are available for all other types of data transfer (local, remote, or with hosts).

Although a partnership Fibre Channel port mask was created, there was no remote replication in the test environment. Therefore, the ports reserved for partnership (2 and 6) were not considered in the zoning schema. Table 4-2 shows the SAN zones created for the test environment.

Table 4-2 SAN zones created in the test environment

Zone name	Fabric A participant FC ports					Fabric B participant FC ports				
	SVC 1	SVC 2	Flash A	Flash B	Host	SVC 1	SVC 2	Flash A	Flash B	Host
SVC_Private	4	4				8	8			
FlashA_SVC	1	1	1,3			5	5	2,4		
FlashB_SVC	1	1		1,3		5	5		2,4	
Host_SVC	3	3			1,3	7	7			2,4

4.3 SAN Volume Controller MDisk configuration

This section describes the best practices and considerations when designing and planning the SAN Volume Controller (SVC) storage pool (MDisk group) setup using IBM FlashSystem 820. There are several considerations when planning and configuring the MDisk and storage pools for IBM FlashSystem 820 behind SVC.

4.3.1 MDisk LUN configuration on IBM FlashSystem 820

When using IBM FlashSystem 820 behind SVC, it is important to remember certain things when looking to design and create MDisk for use in storage pools. In this case, queue and cache assignment is not as relevant as they would be with traditional spindle-based disk systems due to the speed that IBM FlashSystem 820 is able to process I/O requests.

Key things to remember are that now, the maximum supported IBM FlashSystem 820 LU size is 2 TB (except when running SVC code 7.1.0.2 or higher) and the maximum of MDisk per SVC cluster is 4096 MDisk.

The other key considerations are that the MDisk should be created in the following number range for optimal performance. The number of MDisk should always be divisible by the

number 4, have a minimum of 4 MDisks, and a maximum of 16 MDisks per IBM FlashSystem 820.

It is recommended to use 4, 8, or 16 MDisks because all are numbers that are divisible by 2, which means that even numbers of MDisks will be allocated to SVC CPU cores, which are an optimal number for CPU processing.

Note: The SAN Volume Controller cannot currently detect MDisks configured with a 4 K (4096 B) block size. Use 512 B when creating LUs on IBM FlashSystem storage systems.

The main reason that this is recommended is due to the MDisk-to-CPU processor core mappings, as shown in Figure 4-46.

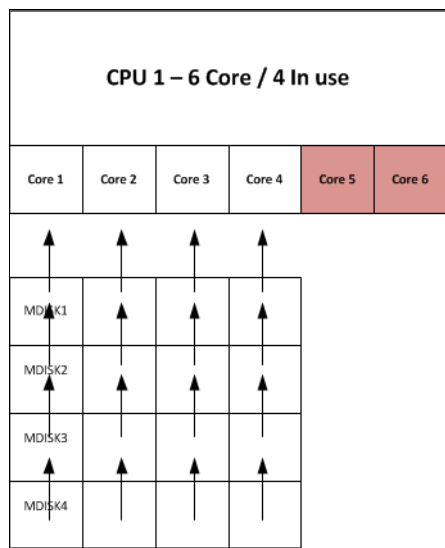


Figure 4-46 CPU cores to MDisk mapping theory example using 16 MDisks

Note: Cores 5 and 6 on the SVC’s first CPU are reserved for Real-time Compression (RtC) functionality.

A secondary reason is that the IBM FlashSystem 820 currently has four HBA ports to service I/O. This means that there is a totally balanced environment for CPU core allocation, MDisk numbers, and paths to HBAs with everything being divisible by four.

When working with a FlashSystem to define what the LU size should be in your configuration, apply this process to it.

Note: The useable capacity of the FlashSystem can be obtained from the GUI under Logical Units, or in the system report in the the storage overview section under TOTAL_SIZE=.

As an example, if you decide that you want 8 MDisks for an IBM FlashSystem 820 which has a usable capacity of 18.75 TiB, you can work out what that is going to be in GB by multiplying this number by 1024. This gives us 19199.97 GB (or 19200 rounded up), then divide this by the number of MDisks. In this example it is 8 MDisks, which gives us 2399.996 GB. Then you need to multiply it by 1024 again to get the number of MB, which is 2,457,596 MB rounded up (actual 2,457,595.90).

This will leave the smallest possible amount of free space on the FlashSystem.

Note: The suggested LU size for an 8 MDisk FlashSystem behind an SVC is 2,457,596 MiB, and for 16 LU the suggested size is 1,228,798 MiB. This will leave the smallest possible amount of free space on the Flashsystem. You must create the LU specifying the MB amount.

Note: You can use either the GUI or the CLI to configure LU's as the calculations are the same on both.

4.3.2 Storage pool configuration guidelines

When using IBM FlashSystem 820 behind SVC, the best practice is to create a single storage pool for each IBM FlashSystem 820 device.

A single storage pool for each FlashSystem device is the recommended best practice to increase storage pool availability and reduce the impact if a failure was to occur. It is also because it increases the flexibility to move volumes around if required to different pools if one FlashSystem becomes overloaded. If there are multiple IBM FlashSystem devices in a single pool, there are fewer storage pools that are available to relocate volumes too.

Below are some further reasons why it is not recommended to stripe across multiple arrays:

- Data on these systems, if important enough, would be required to have a DR position and protection, which negates the need for striping across multiple arrays for protection.
- There is no performance advantage of wide striping across multiple IBM FlashSystem 820 arrays, generally because they stripe effectively on a single internal subsystem.

The same principals should be applied that exist in the SVC best practices documentation. The main consideration is that a single IBM FlashSystem 820 should be used only with a single SVC cluster.

Refer to Chapter 5 in the following IBM Redbooks publication for more information about storage pool configuration: *IBM System Storage SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521.

4.3.3 Quorum disk allocation

When adding the first IBM FlashSystem 820 back-end system, the SVC will automatically allocate three MDisk as quorum disks on the first IBM FlashSystem 820 to be connected. To check what MDisk have been allocated, run the **lsquorum** command.

However, when adding a second or third IBM FlashSystem 820 or storage back-end device the recommendation is to change the quorum MDisk settings manually so that *the quorum disks are spread across multiple storage systems for redundancy*. This can be done by using the **chquorum** command, as shown in Example 4-27.

Example 4-27 How to change the quorum disk when a second system is added using the CLI

Before second system is added run **lsquorum**

quorum_index	status	id	name	controller_id	controller_name	active	object_type	override
0	online	0	FlashSystem_A_SVC_1	0	FlashSystem_A	yes	MDisk	no
1	online	1	FlashSystem_A_SVC_2	0	FlashSystem_A	no	MDisk	no

```
2          online 2  FlashSystem_A_SVC_3 0          FlashSystem_A no      MDisk      no
```

After the second system is added run a **lscontroller** to show the controller id and rename if required using **chcontroller -name %new name% %controller id%**

Example: **chcontroller -name FlashSystem_B 1**

```
id controller_name ctrl_s/n      vendor_id      product_id_low  product_id_high
0  FlashSystem_A 1224c00000      IBM           FlashSys       tem
1  FlashSystem_B 1224c10000      IBM           FlashSys       tem
```

You need to run an **lsMDisk** to list the available MDisk and select an MDisk on another controller to use as the quorum disk. Below is a truncated extract of an **lsMDisk** output.

```
id name                status mode      MDisk_grp_id MDisk_grp_name capacity ctrl_LUN_#
8  FlashSystem_B_SVC_1 online managed 1          FlashSystem_B 1.2TB 0000000000000000
9  FlashSystem_B_SVC_2 online managed 1          FlashSystem_B 1.2TB 0000000000000001
10 FlashSystem_B_SVC_3 online managed 1          FlashSystem_B 1.2TB 0000000000000002
11 FlashSystem_B_SVC_4 online managed 1          FlashSystem_B 1.2TB 0000000000000003
12 FlashSystem_B_SVC_5 online managed 1          FlashSystem_B 1.2TB 0000000000000004
```

Run **chquorum** to change second quorum disk to be on second subsystem, we will use MDisk 10 in this example.

chquorum -MDisk %MDisk number% %quorum disk number%

Example: **chquorum -MDisk 10 1**

Output of **lsquorum** after changes applied

```
quorum_index status id name                controller_id controller_name active object_type override
0            online 0  FlashSystem_A_SVC_1 0          FlashSystem_A yes  MDisk      no
1            online 10 FlashSystem_B_SVC_3 1          FlashSystem_B no   MDisk      no
2            online 2  FlashSystem_A_SVC_3 0          FlashSystem_A no   MDisk      no
```

A best practice example using two IBM FlashSystem 820 devices is shown in Example 4-28.

Example 4-28 An example lsquorum output

```
quorum_index status id name                controller_id controller_name active object_type override
0            online 0  FlashSystem_A_SVC_1 0          FlashSystem_A yes  MDisk      no
1            online 10 FlashSystem_B_SVC_3 1          FlashSystem_B no   MDisk      no
2            online 2  FlashSystem_A_SVC_3 0          FlashSystem_A no   MDisk      no
```

Note: The **chquorum** command is strongly recommended to be run prior to any volume that is being provisioned from the additional FlashSystems that are added after the first IBM FlashSystem 820.

For more information, see section 5.2 in the Redbooks publication *IBM System Storage SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521.

This can also be set using the GUI. From the **Pools** → **MDisks by Pools** page on the SVC GUI, select the three or more MDisk *in the priority order you would like them to be* that you want to make the quorum disks then right-click and select **Edit Quorum**, as shown in Figure 4-47 on page 115.

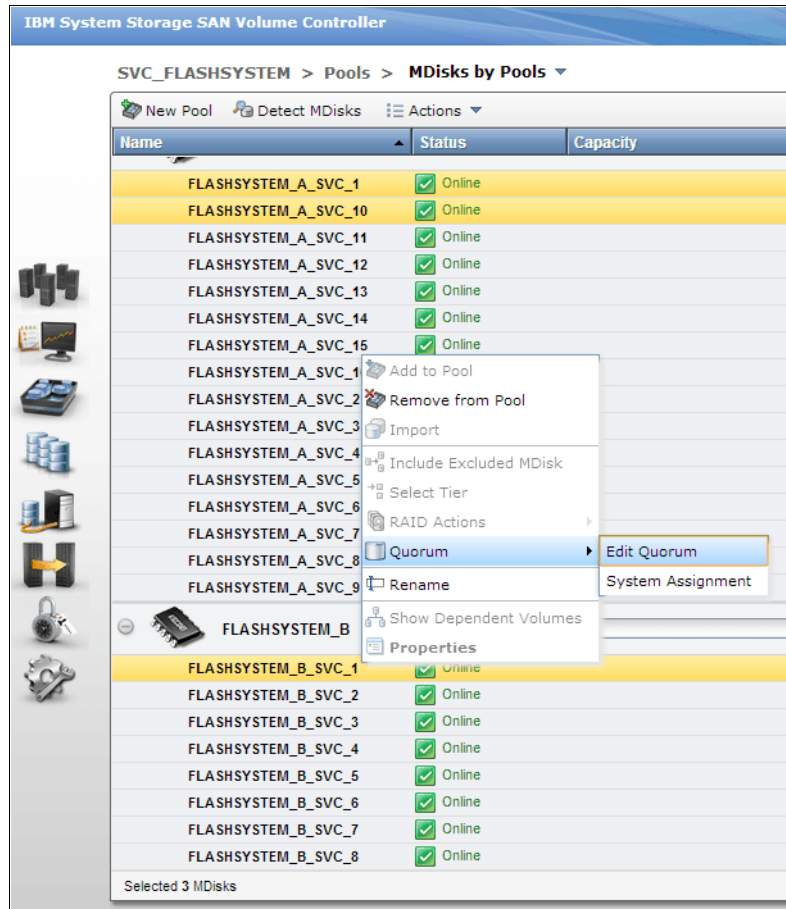


Figure 4-47 MDisks by pools page showing the change quorum menu

This wizard window will be presented. If this is a *single site configuration*, ensure that the corresponding radio button is selected. Check the MDisks that are to become the *quorum MDisks* and click **Edit Quorum**, as shown in Figure 4-48.

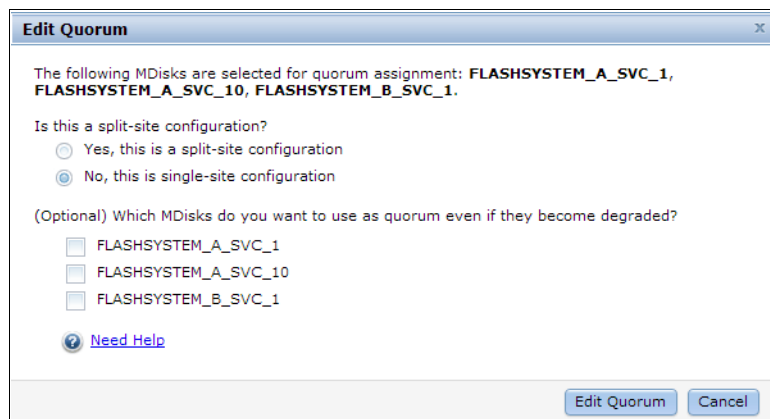


Figure 4-48 Edit Quorum wizard

The command will execute and complete the changes to the quorum MDisks, as shown in Figure 4-49.

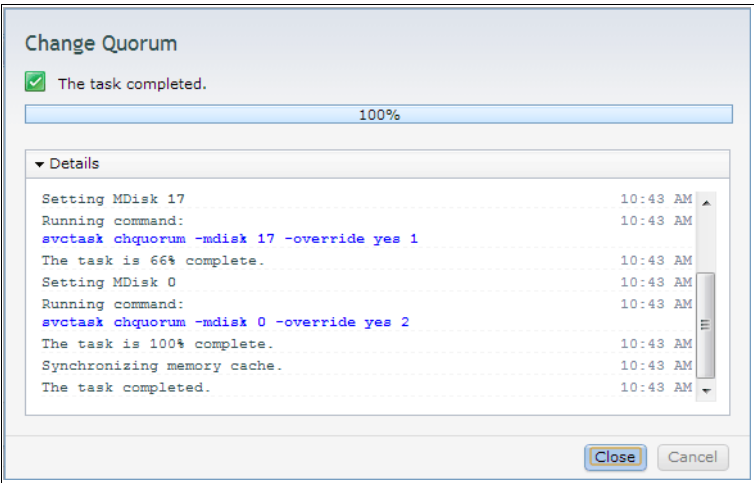


Figure 4-49 Change quorum changes are complete

Note: In Figure 4-48 on page 115, the “Is this a split-site configuration” question refers to stretched cluster configurations. The quorum disk is a very important consideration for stretched clusters. Refer to the following IBM Redbooks publications for more information:

- ▶ *IBM SAN and SVC Stretched Cluster and VMware Solution Implementation*, SG24-8072
- ▶ *IBM SAN Volume Controller Stretched Cluster with PowerVM and PowerHA*, SG24-8142

4.3.4 Storage pool extent size

When working with an all IBM FlashSystem 820 behind SVC configuration, the extent size can be left at the default of 1 GB (1024 KB). This is because the IBM FlashSystem 820’s performance with random I/O workload does not require the extent size to be smaller than this.

If there are existing systems and existing storage pools, to be able to use transparent volume migration between pools you need to have the same extent size for all pools. If you have different extent sizes, you can use volume mirroring to create a copy of the disk and then promote it to the master volume when the copy is completed. However, this is a manual process and takes slightly longer to complete.

Example 4-29 lsmdiskgroup truncated output to show same extent size

id	name	status	MDisk_count	volume_count	capacity	extent_size	free_capacity	virtual_capacity
0	IBM FlashSystem 820_A	online	8	1	18.74TB	1024	18.63TB	4.00TB
1	IBM FlashSystem 820_B	online	16	1	18.73TB	1024	18.61TB	4.00TB

4.3.5 MDisk mapping and storage pool creation using the CLI

We now briefly describe the process to follow when mapping MDiskS from an IBM FlashSystem 820 to the SVC.

Note: Image mode disks should only ever be used for migration purposes.

Once LUs have been created on the IBM FlashSystem and presented to the SVC, the **detectmdisk** command needs to be run from the CLI to detect available MDiskS that have been presented.

When the **detectmdisk** command has completed, the **lsmdisk** command needs to be run to show all detected and available MDiskS. An example output is shown in Example 4-30.

Example 4-30 lsmdisk output (truncated) shows detected and available MDiskS

id	name	status	mode	MDisk_grp_id	MDisk_grp_name	capacity	ctrl_LUN_#	controller_name	UID
0	FlashSystem_B_SVC_1	online	managed	0	FlashSystem_B	2.3TB	0000000000000000	FlashSystem_B	0020c240001224c00
1	FlashSystem_B_SVC_2	online	managed	0	FlashSystem_B	2.3TB	0000000000000001	FlashSystem_B	0020c240011224c00
2	FlashSystem_B_SVC_3	online	managed	0	FlashSystem_B	2.3TB	0000000000000002	FlashSystem_B	0020c240021224c00
3	FlashSystem_B_SVC_4	online	managed	0	FlashSystem_B	2.3TB	0000000000000003	FlashSystem_B	0020c240031224c00
4	FlashSystem_B_SVC_5	online	managed	0	FlashSystem_B	2.3TB	0000000000000004	FlashSystem_B	0020c240041224c00
5	FlashSystem_B_SVC_6	online	managed	0	FlashSystem_B	2.3TB	0000000000000005	FlashSystem_B	0020c240051224c00
6	FlashSystem_B_SVC_7	online	managed	0	FlashSystem_B	2.3TB	0000000000000006	FlashSystem_B	0020c240061224c00
7	FlashSystem_B_SVC_8	online	managed	0	FlashSystem_B	2.3TB	0000000000000007	FlashSystem_B	0020c240071224c00
8	FlashSystem_A_SVC_1	online	unmanaged			1.2TB	0000000000000000	FlashSystem_A	0020c240001224c10
9	FlashSystem_A_SVC_2	online	unmanaged			1.2TB	0000000000000001	FlashSystem_A	0020c240011224c10
10	FlashSystem_A_SVC_3	online	unmanaged			1.2TB	0000000000000002	FlashSystem_A	0020c240021224c10
11	FlashSystem_A_SVC_4	online	unmanaged			1.2TB	0000000000000003	FlashSystem_A	0020c240031224c10
12	FlashSystem_A_SVC_5	online	unmanaged			1.2TB	0000000000000004	FlashSystem_A	0020c240041224c10
13	FlashSystem_A_SVC_6	online	unmanaged			1.2TB	0000000000000005	FlashSystem_A	0020c240051224c10
14	FlashSystem_A_SVC_7	online	unmanaged			1.2TB	0000000000000006	FlashSystem_A	0020c240061224c10
15	FlashSystem_A_SVC_8	online	unmanaged			1.2TB	0000000000000007	FlashSystem_A	0020c240071224c10
16	FlashSystem_A_SVC_9	online	unmanaged			1.2TB	0000000000000008	FlashSystem_A	0020c240081224c10
17	FlashSystem_A_SVC_10	online	unmanaged			1.2TB	0000000000000009	FlashSystem_A	0020c240091224c10
18	FlashSystem_A_SVC_11	online	unmanaged			1.2TB	000000000000000A	FlashSystem_A	0020c2400a1224c10
19	FlashSystem_A_SVC_12	online	unmanaged			1.2TB	000000000000000B	FlashSystem_A	0020c2400b1224c10
20	FlashSystem_A_SVC_13	online	unmanaged			1.2TB	000000000000000C	FlashSystem_A	0020c2400c1224c10
21	FlashSystem_A_SVC_14	online	unmanaged			1.2TB	000000000000000D	FlashSystem_A	0020c2400d1224c10
22	FlashSystem_A_SVC_15	online	unmanaged			1.2TB	000000000000000E	FlashSystem_A	0020c2400e1224c10
23	FlashSystem_A_SVC_16	online	unmanaged			1.2TB	000000000000000F	FlashSystem_A	0020c2400f1224c10

Unmanaged mode MDiskS can be added to a new storage pool. *Managed* mode MDiskS mean that they are already a member of a storage pool.

This output shows that the MDiskS have been renamed. It is very important to rename the MDiskS when they are added to the SVC for identification purposes. The output also shows that the unmanaged MDiskS are allocated from *FlashSystem_A* (which was renamed earlier), and that their controller LUN ID after the number 9 converts to a hexadecimal number on the SVC.

When naming an MDisk, the recommended naming convention is “%controller name_lun id on disk system%”. This defines the name of the controller that the LUN is presented from, and the local identifier that is referenced on the source disk system. These are very important when troubleshooting.

The output also shows that there is an existing storage pool called *FlashSystem_B*, which is made up of eight 2.3 TB MDiskS.

The next step would be to create a new storage pool by using the **mkmdiskgrp** command with some additional parameters to set the disk tier and the extent size, as shown in Example 4-31 on page 118.

The extent size is important when setting up multiple pools because it allows seamless volume migration, as explained in 4.3.4, “Storage pool extent size” on page 116. Setting the

disk tier options helps the SVC identify where to put hot extents when using Easy Tier. See Example 4-31.

Example 4-31 Example mkmdiskgrp command

```
IBM_2145:SVC_FlashSystem:superuser>mkmdiskgrp -name FlashSystem_A -tier generic_ssd -ext 1024
MDisk Group, id [1], successfully created
```

Note: You can also add MDisks in a single `mkmdiskgrp` command with the `-mdisk` switch if desired.

The next step is to add the MDisks to the storage pool by using the `addmdisk` command. A reference to the MDisk ID or name from the `lsmdisk` output is required along with the storage pool ID or name, as shown in Example 4-32.

Example 4-32 Example addmdisk command

```
IBM_2145:SVC_FlashSystem:superuser>addMDisk -MDisk 8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23 -tier generic_ssd 1
```

The storage pool creation is now complete. It can be verified that all MDisks have been correctly assigned by re-running the `lsmdisk` command, as shown in Example 4-33.

Example 4-33 Shows 16 MDisks, now a member of Storage Pool 1

id	name	status	mode	MDisk_grp_id	MDisk_grp_name	capacity	ctrl_LUN_#	controller_name	UID
8	FlashSystem_A_SVC_1	online	managed	1	FlashSystem_A	1.2TB	0000000000000000	FlashSystem_A	0020c240001224c10
9	FlashSystem_A_SVC_2	online	managed	1	FlashSystem_A	1.2TB	0000000000000001	FlashSystem_A	0020c240011224c10
10	FlashSystem_A_SVC_3	online	managed	1	FlashSystem_A	1.2TB	0000000000000002	FlashSystem_A	0020c240021224c10
11	FlashSystem_A_SVC_4	online	managed	1	FlashSystem_A	1.2TB	0000000000000003	FlashSystem_A	0020c240031224c10
12	FlashSystem_A_SVC_5	online	managed	1	FlashSystem_A	1.2TB	0000000000000004	FlashSystem_A	0020c240041224c10
13	FlashSystem_A_SVC_6	online	managed	1	FlashSystem_A	1.2TB	0000000000000005	FlashSystem_A	0020c240051224c10
14	FlashSystem_A_SVC_7	online	managed	1	FlashSystem_A	1.2TB	0000000000000006	FlashSystem_A	0020c240061224c10
15	FlashSystem_A_SVC_8	online	managed	1	FlashSystem_A	1.2TB	0000000000000007	FlashSystem_A	0020c240071224c10
16	FlashSystem_A_SVC_9	online	managed	1	FlashSystem_A	1.2TB	0000000000000008	FlashSystem_A	0020c240081224c10
17	FlashSystem_A_SVC_10	online	managed	1	FlashSystem_A	1.2TB	0000000000000009	FlashSystem_A	0020c240091224c10
18	FlashSystem_A_SVC_11	online	managed	1	FlashSystem_A	1.2TB	000000000000000A	FlashSystem_A	0020c2400a1224c10
19	FlashSystem_A_SVC_12	online	managed	1	FlashSystem_A	1.2TB	000000000000000B	FlashSystem_A	0020c2400b1224c10
20	FlashSystem_A_SVC_13	online	managed	1	FlashSystem_A	1.2TB	000000000000000C	FlashSystem_A	0020c2400c1224c10
21	FlashSystem_A_SVC_14	online	managed	1	FlashSystem_A	1.2TB	000000000000000D	FlashSystem_A	0020c2400d1224c10
22	FlashSystem_A_SVC_15	online	managed	1	FlashSystem_A	1.2TB	000000000000000E	FlashSystem_A	0020c2400e1224c10
23	FlashSystem_A_SVC_16	online	managed	1	FlashSystem_A	1.2TB	000000000000000F	FlashSystem_A	0020c2400f1224c10

4.3.6 MDisk mapping and storage pool creation using the GUI

As per the CLI method of creating storage pools, the same process is followed with the GUI. The first step is to click **Detect MDisks** from the GUI to detect newly presented MDisks, as shown in Figure 4-50.

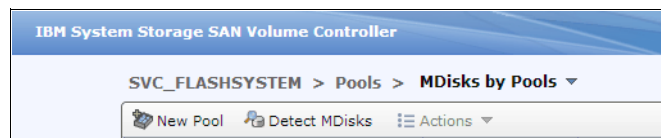
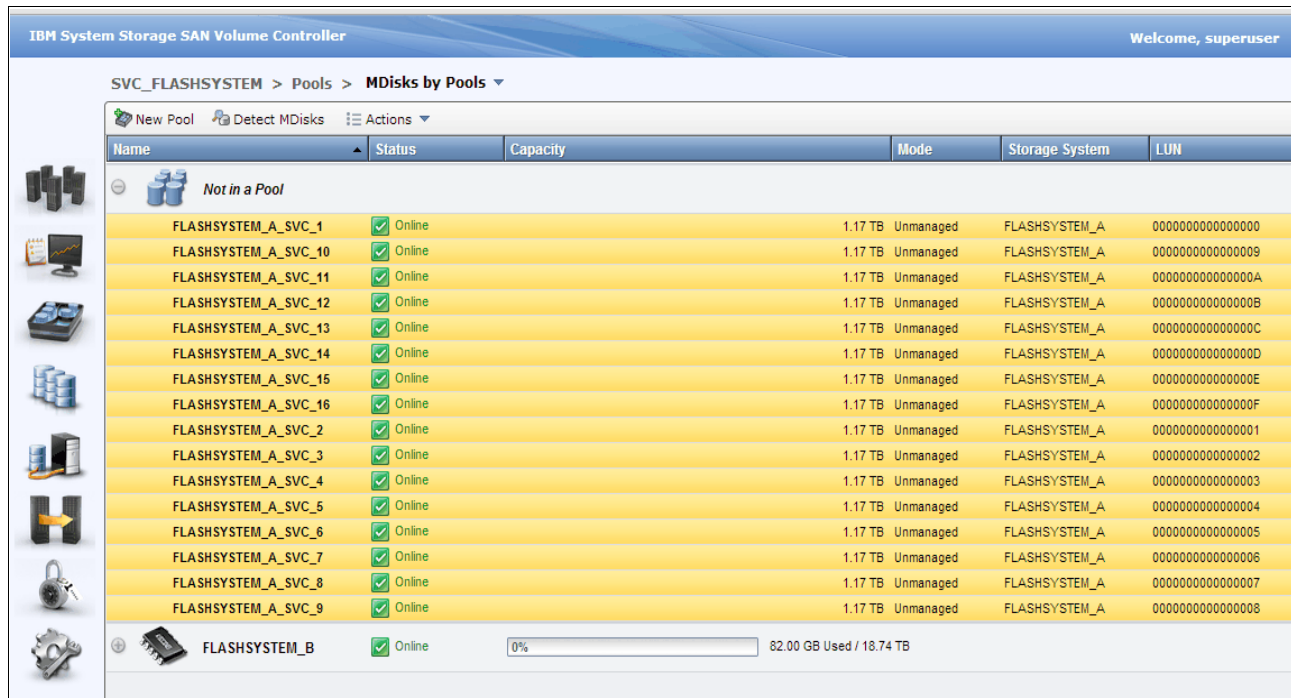


Figure 4-50 Detect MDisks command button in the GUI

When this detection has completed, a list of MDisks will appear, as shown in Figure 4-51 on page 119. From this output, you can see that the MDisks have been renamed. It is very important to rename the MDisks when they are added, for identification purposes.

You can see that the unmanaged MDisks are allocated from FlashSystem_A (which was renamed earlier), and that their controller LUN ID after the number 9 converts to a hexadecimal number. When naming an MDisk, the recommended naming convention is “%controller name_lun id on disk system%”. This defines the name of the controller that the

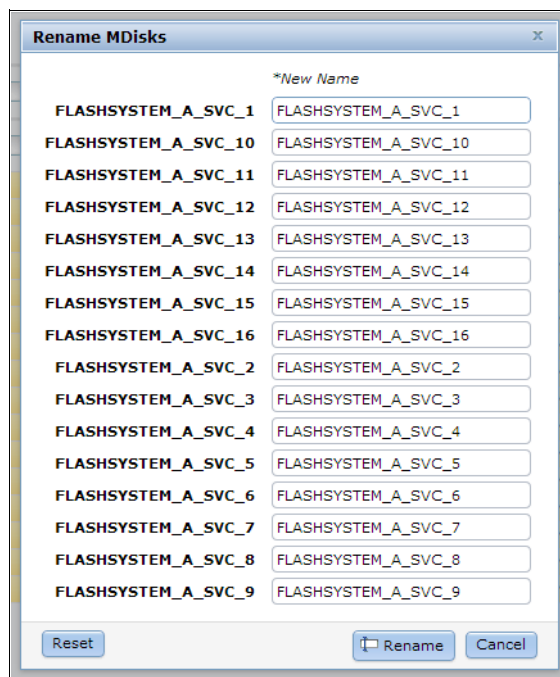
LUN is presented from, and the local identifier that is referenced on the source disk system. These are very important when troubleshooting.



Name	Status	Capacity	Mode	Storage System	LUN
Not in a Pool					
FLASHSYSTEM_A_SVC_1	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000000
FLASHSYSTEM_A_SVC_10	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000009
FLASHSYSTEM_A_SVC_11	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	000000000000000A
FLASHSYSTEM_A_SVC_12	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	000000000000000B
FLASHSYSTEM_A_SVC_13	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	000000000000000C
FLASHSYSTEM_A_SVC_14	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	000000000000000D
FLASHSYSTEM_A_SVC_15	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	000000000000000E
FLASHSYSTEM_A_SVC_16	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	000000000000000F
FLASHSYSTEM_A_SVC_2	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000001
FLASHSYSTEM_A_SVC_3	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000002
FLASHSYSTEM_A_SVC_4	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000003
FLASHSYSTEM_A_SVC_5	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000004
FLASHSYSTEM_A_SVC_6	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000005
FLASHSYSTEM_A_SVC_7	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000006
FLASHSYSTEM_A_SVC_8	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000007
FLASHSYSTEM_A_SVC_9	Online	1.17 TB	Unmanaged	FLASHSYSTEM_A	0000000000000008
FLASHSYSTEM_B	Online	0%			82.00 GB Used / 18.74 TB

Figure 4-51 Highlighted MDisks to be renamed

When all the MDisks are selected, the MDisks need to be renamed by using the **right-click** → **Rename** menu. A Rename MDisks window is displayed, as shown in Figure 4-52. Type the new names of the MDisks and, once completed, click **Rename**.



Rename MDisks

*New Name

FLASHSYSTEM_A_SVC_1 FLASHSYSTEM_A_SVC_1

FLASHSYSTEM_A_SVC_10 FLASHSYSTEM_A_SVC_10

FLASHSYSTEM_A_SVC_11 FLASHSYSTEM_A_SVC_11

FLASHSYSTEM_A_SVC_12 FLASHSYSTEM_A_SVC_12

FLASHSYSTEM_A_SVC_13 FLASHSYSTEM_A_SVC_13

FLASHSYSTEM_A_SVC_14 FLASHSYSTEM_A_SVC_14

FLASHSYSTEM_A_SVC_15 FLASHSYSTEM_A_SVC_15

FLASHSYSTEM_A_SVC_16 FLASHSYSTEM_A_SVC_16

FLASHSYSTEM_A_SVC_2 FLASHSYSTEM_A_SVC_2

FLASHSYSTEM_A_SVC_3 FLASHSYSTEM_A_SVC_3

FLASHSYSTEM_A_SVC_4 FLASHSYSTEM_A_SVC_4

FLASHSYSTEM_A_SVC_5 FLASHSYSTEM_A_SVC_5

FLASHSYSTEM_A_SVC_6 FLASHSYSTEM_A_SVC_6

FLASHSYSTEM_A_SVC_7 FLASHSYSTEM_A_SVC_7

FLASHSYSTEM_A_SVC_8 FLASHSYSTEM_A_SVC_8

FLASHSYSTEM_A_SVC_9 FLASHSYSTEM_A_SVC_9

FLASHSYSTEM_B

Reset Rename Cancel

Figure 4-52 The GUI rename menu

The next step is to create a new storage pool by clicking **New Pool**, as shown in Figure 4-53.

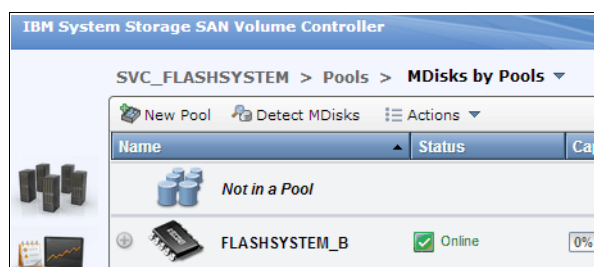


Figure 4-53 The new pool command button

The next window is used to configure the storage pool name and extent size. Ensure that **Advanced Settings** is clicked to show the extent size menu. The default is 1 GB (1024 MB), which can be left as the default, as explained in section 4.3.4, “Storage pool extent size” on page 116. The name of the storage pool is also required, as shown in Figure 4-54. When complete, click **Next**.

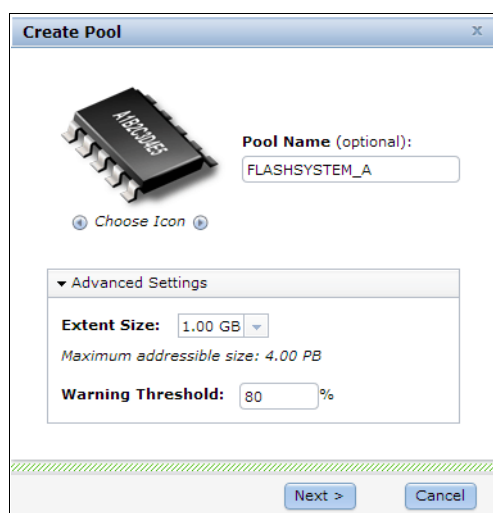


Figure 4-54 The new pool wizard and advanced settings displayed

When this is complete, the *MDisks* that will be in this *storage pool* need to be added. Select all the *MDisks* that will be members of this group. In this case, we have 16 *MDisks*, which are all to become members of the *storage pool* that we are creating. Click **Create** to begin the creation of the *storage pool*, as shown in Figure 4-55 on page 121.

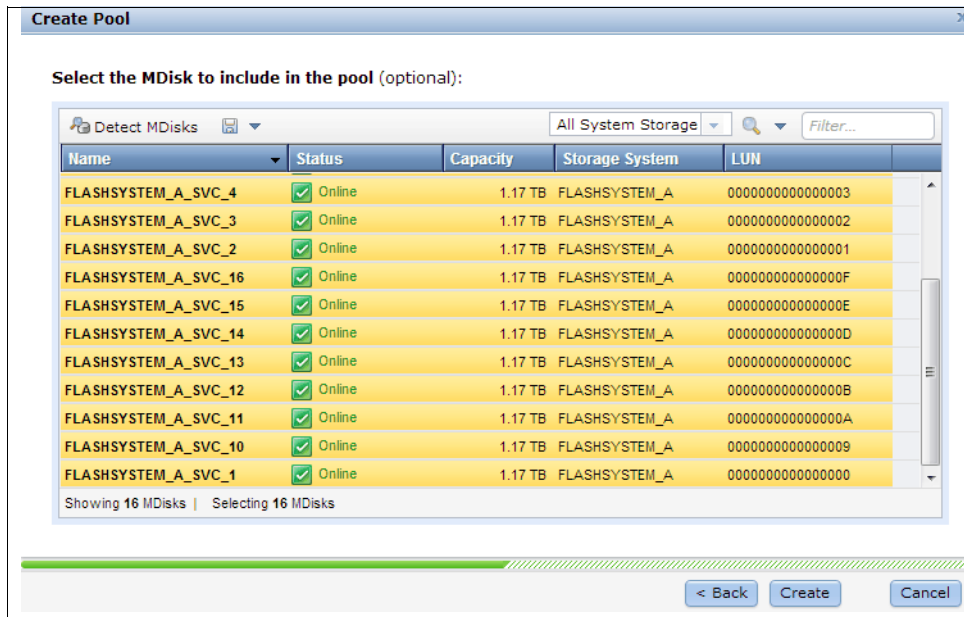


Figure 4-55 Showing highlighted MDisks to be added to the storage pool

The process will begin, as shown in Figure 4-56.

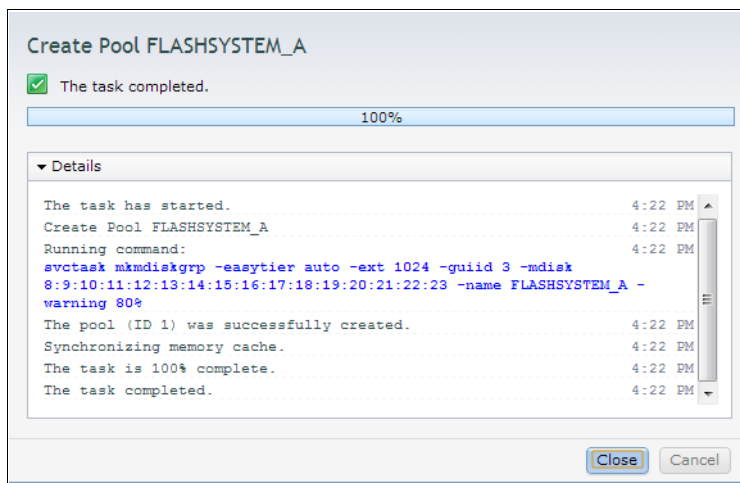


Figure 4-56 Command to create a storage pool being processed and the storage pool being created

When the commands have completed, the *storage pool* has been created and the *MDisks* are now a member of the *storage pool*, as shown in Figure 4-57 on page 122.

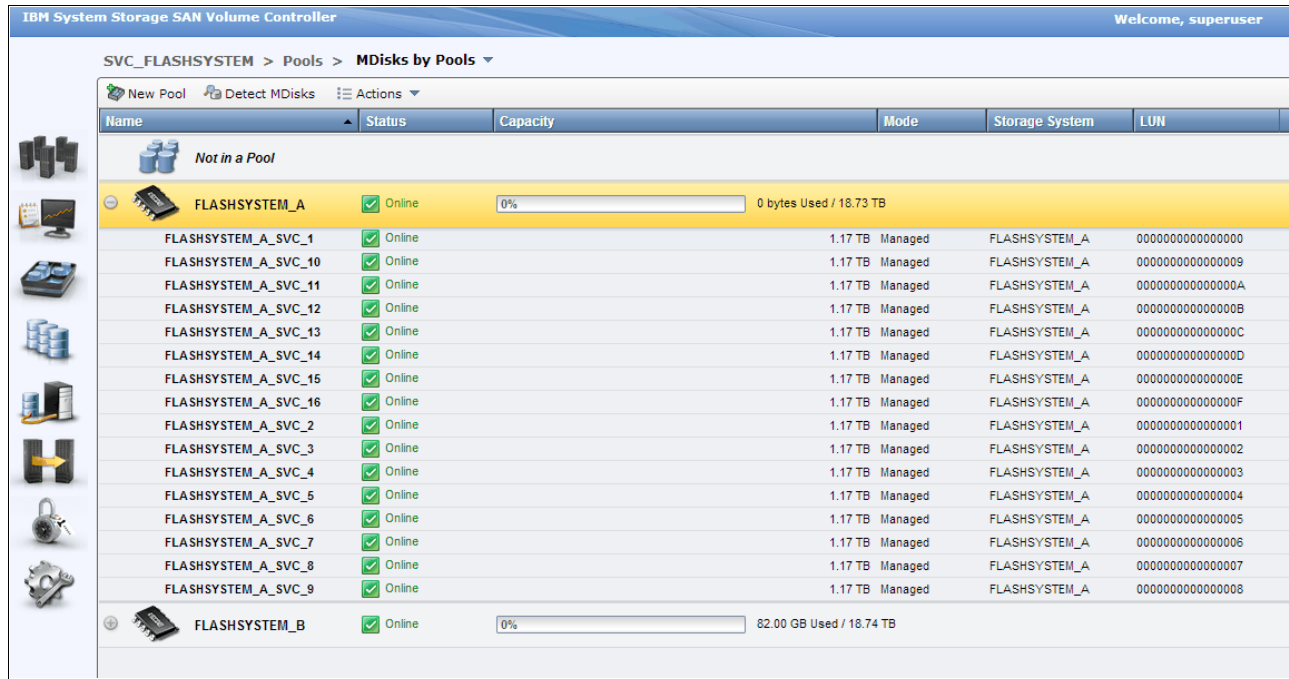


Figure 4-57 GUI that shows the newly created storage pool group and its MDisk members

This completes the definition of the MDisks and one storage pool in the test environment. If a second storage pool is necessary for volume mirroring, or for more volume allocation, repeat the above steps to create the second storage pool. In our case, we created a second storage pool called *FlashSystem_B*, which was formed by MDisks coming from the other FlashSystem 820 device that was available.



Diagnostics, planned outages, and troubleshooting

This chapter provides basic information about the diagnostics and troubleshooting tools that are available to help identify and solve common problems that can affect environments comprised of the IBM SAN Volume Controller (SVC) and IBM FlashSystem model 820 storage systems.

The following topics are covered:

- ▶ I/O group volume migration for planned outages
- ▶ Non-disruptive FlashSystem firmware updates with volume mirroring
- ▶ Volume migration to another storage pool for planned outages
- ▶ Call home features of both SAN Volume Controller and FlashSystem storage systems
- ▶ Easy Tier and IBM FlashSystem storage system outage
- ▶ Hardware and troubleshooting guide for IBM FlashSystem storage systems

5.1 I/O group volume migration for planned outages

This section details how to move a volume between I/O groups.

SVC code level 6.4 introduced a new feature to enable live, nondisruptive migration of volumes between I/O groups. Example scenarios where moving a volume might be required include the following:

- ▶ SVC node failure in a high performance I/O group
- ▶ The occurrence of performance issues within an I/O group
- ▶ Device or software limitations have been reached in a particular I/O group (that is, number of volumes, host connections, or compressed volumes)

Note: Ensure that your host platform supports nondisruptive volume moves, and that you have the latest supported version of the multipathing driver (SDD is recommended whenever possible) installed on your hosts. Support for volume migration with VMware is supported via the native VMware driver. See the “Non Disruptive Volume Move” section of the Supported Hardware List, Device Driver, Firmware, and Recommended Software Levels for your SVC version at the following website:

<http://www.ibm.com/support/docview.wss?uid=ssg1S1003658>

Take the following steps to execute a volume migration:

1. Use the **addvdiskaccess** command in the SVC command-line interface (CLI) to add the destination I/O group to the access list of I/O groups for the desired volume.
2. Use the **movevdisk** command in the SVC CLI to move the desired volume to the destination I/O group.
3. Issue the appropriate commands on the hosts that are mapped to the volume to detect the new paths to the volume in the destination I/O group.
4. When you confirm that the new paths are online, remove access from the old I/O group, using the **rmvdiskaccess** command.
5. Issue the appropriate commands on the hosts that are mapped to the volume to remove the paths to the old I/O group.

Note: Ensure that you rescan your disks from your host after running this procedure and ensure that all paths are visible to the volume on the SVC. Some host platforms do not support removing paths to volumes dynamically. For those platforms, the old paths are only removed in its next reboot.

This process is instantaneous, and there is no wait time when changing I/O groups because there is no on-disk migration. Rather, the workload to the original volume is simply transferred to the new I/O group.

5.2 FlashSystem firmware update non-disruptively with the use of volume mirroring

This section details how to perform a firmware upgrade on an IBM FlashSystem device, without incurring an outage, utilizing volume mirroring.

As seen in 2.3, “SAN Volume Controller volume mirroring use cases with FlashSystem” on page 57, SVC volume mirroring can protect data availability against storage subsystem planned and unplanned outages. At the time this book was written, firmware upgrades to IBM FlashSystem storage systems required a planned outage. As such, SVC volume mirroring can be used to maintain the availability of applications that would otherwise be affected by those outages, where each volume that contains a copy in the FlashSystem storage system also contains a copy that resides on an alternate disk subsystem.

With all mirrored volumes synchronized, each with both volumes’ copies online, FlashSystem firmware upgrades can be performed on the FlashSystem storage system. When the FlashSystem device being upgraded becomes temporarily unavailable due to the upgrade, the SVC for the affected volumes automatically switches the I/O operations in the back end to the copies that are outside of the FlashSystem storage system.

In effect, all host I/O operations subsequent to the moment of FlashSystem unavailability are frozen until SVC gives up trying to access the primary copies and the I/O operations are redirected to the alternate copies of the affected volumes. The duration of this freeze of I/O operations depends on the workload that the system is experiencing at that moment and can vary.

The time that the total workload is redirected to the alternate subsystem is the time that the FlashSystem takes until it returns to operation after its firmware is updated. Consequently, the time to resynchronize the updated data in the end of the process is as short as possible and done transparently. As explained before, this resynchronization is incremental to minimize its duration because it copies only the grains that have been written to since the copies became out of sync.

Once the FlashSystem becomes available again after its firmware is updated, all the settings for primary copies automatically return to the previous state, which means that the primary copies are redistributed between the two systems as before the maintenance started.

5.3 Volume migration to another storage pool for planned outages

This section details how to use seamless volume migration on the SVC to move volumes to another storage pool to assist with a planned outage of an IBM FlashSystem storage system.

A feature of the SVC that is useful when having to perform maintenance on a selected disk system is the ability to seamlessly migrate volumes from one storage pool to another. The maximum capacity of a single FlashSystem storage system is approximately 20 TB, making this a useful feature to use when having to perform maintenance on a FlashSystem device that requires an outage that is not set up for high availability.

Note: To use the `migratevdisk` command, all storage pools must have the same extent size.

Depending on the target disk system and the I/O workload on the active volumes on the FlashSystem, the migration time will vary as it is essentially moving the entire volume, extent by extent, to another disk subsystem.

We suggest working with your local IBM representative to estimate the time to perform this migration if you are planning to use this feature for an entire FlashSystem storage system containing up to 20 TB of data.

The process has no impact on running systems and can be performed during business hours. However, if you plan to schedule downtime for the device, account for the appropriate amount of time for the migration to complete before you are able to power off the system.

To migrate a volume from one pool to another, use the **migratevdisk** command, as shown in Example 5-1.

Example 5-1 Command syntax for moving a volume to another storage pool

```
migratevdisk -vdisk %id or name% -mdiskgrp %target group id or name% -threads %no of threads%
```

```
migratevdisk -vdisk VMWARE01 -mdiskgrp FLASHSYSTEM_B -threads 4
```

Note: The number of threads used in the migration can be varied based on the priority that volume is required to be migrated at. The higher the number of threads, the faster the volume will be migrated.

For more information about volume migration, see the following IBM Redbooks publication:

Implementing the IBM System Storage SAN Volume Controller V6.3, SG24-7933

5.4 Call home features of SAN Volume Controller and FlashSystem storage systems

This section details the built-in call home functions for both SVC and IBM FlashSystem storage systems.

5.4.1 SVC call home

The SVC contains a call home feature, that, when configured correctly, will automatically notify the IBM Support Center when a system event occurs. In most cases, this will log a support ticket within IBM and an IBM service representative will contact you to assist with the issue. If this does not occur within a few hours, call IBM support directly. If this is an urgent issue, do not wait; call IBM immediately.

Note: When performing maintenance on the SVC, you must disable event notifications to prevent call home events from being sent. It is also important to ensure that they are re-enabled once the maintenance is completed.

For more information about how to set up and configure SVC call home, see the following book:

Implementing the IBM System Storage SAN Volume Controller V6.3, SG24-7933

Note: The SVC also supports Simple Network Management Protocol (SNMP) trap monitoring and alerting.

5.4.2 IBM FlashSystem call home and event notification

Like SVC, the IBM FlashSystem storage system has a built-in, email-based, call home and event notification feature that, when configured correctly, will automatically notify the IBM Support Center when a system event occurs. In most cases, this will log a support ticket within IBM and an IBM service representative will contact you to assist with the issue. If this does not occur within a few hours, call IBM support directly. If this is an urgent issue, do not wait; call IBM immediately.

Note: When performing maintenance on the IBM FlashSystem, you must disable call home from being sent. It is also important to ensure that they are re-enabled once the maintenance is completed.

For more information about how to configure call home and event notification on FlashSystem storage systems, see 4.1.5, “email notifications and call home” on page 90.

5.5 Easy Tier and FlashSystem planned outages

This section details how to configure Easy Tier for use when an outage is required for an IBM FlashSystem.

If an outage is required on a FlashSystem storage system that it is being used as Easy Tier capacity for a storage pool, the extents residing on it will need to be relocated in advance of the outage. To achieve this, take the following steps:

1. Ensure that the hard disk drives in the storage pool contain enough free capacity to allow the FlashSystem extents to be migrated off of the FlashSystem MDisks and on to the HDD MDisks.
2. If Easy Tier is set to “auto” for the storage pool, change the setting to “on”. By setting Easy Tier to “on” for the storage pool during this process, the heat maps will be retained, and the duration of the overall process will be minimized.
3. Remove the FlashSystem MDisks from the storage pool to force the extents to be migrated to the HDD MDisks.
4. Ensure that the extent migration has completed, and that the FlashSystem MDisks are now “unmanaged” in the SVC. This step can take up to 48 hours to complete, as the data has to be migrated from the SSDs to the HDDs and the migration is performed at a controlled pace to avoid overloading the storage. The actual duration of this step will depend, in part, on the size of the FlashSystem MDisks that were removed.
5. Perform the required FlashSystem maintenance.
6. Add the FlashSystem MDisks back into the storage pool. Within 24 hours, Easy Tier will begin migrating hot extents to the FlashSystem MDisks. This step can take up to 48 hours to complete because Easy Tier works on a 24-hour sliding window and can move up to 2 TB an hour. The actual duration of this step will depend, in part, on the size of the FlashSystem MDisks added to the pool.

Retaining heat data: In step 2, above, if the setting is left on “auto”, Easy Tier will lose all knowledge about what data is hot. This information will have to be re-learned when the SSDs are re-added. By changing the setting to “on”, this heat data is retained, allowing Easy Tier to make more intelligent and informed decisions when it starts to move data back on to the FlashSystem MDisks in step 6.

Important: Verify that Easy Tier extents have been migrated before shutting down and performing maintenance. Manually verify that all Easy Tier extents have been placed back in their original location after the maintenance process. If there have been other environmental workload changes, the extents may not be placed back in the same location.

5.6 Hardware replacement guide for IBM FlashSystem storage systems

This section details the hardware replacement methods for IBM FlashSystem storage systems.

The IBM FlashSystem storage system might at some point require the replacement of hardware. This work is performed by IBM support personnel. In some instances, the device will be required to be powered down in order to replace components.

Depending on what data is stored on the FlashSystem, it might need to be migrated to another disk system or Easy Tier might need to be reconfigured and the hot extents migrated automatically to another storage pool or volume mirroring be used to avoid data access outages.

Refer to the following guide for more information about hardware replacement with the IBM FlashSystem storage system:

<https://www.ibm.com/support/entdocview.wss?uid=ssg1S7004340>

Note: This website requires an IBM ID for access. If you do not have access, create an account the first time that you visit the site.



FlashSystem CLI commands used in the example environment

In this appendix, we provide the complete command-line interface (CLI) commands used to configure the two IBM FlashSystem 820 storage systems that are used in the example environment throughout the book.

Configuring the example IBM FlashSystem storage systems using the CLI

The following commands were used to configure the IBM FlashSystem storage systems used in our example environment.

Setting the management control processor host name

The commands to set the management control processor host names differ for each example IBM FlashStorage storage system.

Setting the host name on FLASHSYSTEM_A

In Example A-1, we show the CLI commands to set the host name on FLASHSYSTEM_A.

Example A-1 Setting the host name on FLASHSYSTEM_A using the CLI

```
admin #: network hostname mc-1 FLASHSYSTEM820_A
FLASHSYSTEM820_A
```

```
Hostname set to 'FLASHSYSTEM820_A'
```

Setting the host name on FLASHSYSTEM_B

In Example A-2, we show the CLI commands to set the host name on FLASHSYSTEM_B.

Example A-2 Setting the host name on FLASHSYSTEM_B using the CLI

```
admin #: network hostname mc-1 FLASHSYSTEM820_B
FLASHSYSTEM820_B
```

```
Hostname set to 'FLASHSYSTEM820_B'
```

Note about the network CLI command: The syntax of the **network** command differs, depending on whether your IBM FlashSystem model contains a single management control processor, or dual management control processors, as in our example environment. To validate the syntax of the command for your system, use the **help network** command.

Setting the Domain Name Service domain

The same commands were used on each IBM FlashSystem to configure this setting.

In Example A-3, we show the CLI commands to set the Domain Name Service (DNS) domain and restart the network interface.

Example A-3 Setting the DNS domain using the CLI

```
admin #: network dns domain mc-1 hursley.ibm.com
Network DNS search domain set to 'hursley.ibm.com'
```

```
admin #: network restart mc-1
Are you sure you want to restart the network? (y/n)
y
Network restarted
```

Configuring the call home feature

The same commands were used on each IBM FlashSystem to configure this feature.

Configuring the call home mail server

Example A-4 shows the commands that were used to configure and confirm the call home mail server.

Example A-4 Configuring the call home feature with the IBM FlashSystem CLI

```
admin #: system callhome_config 9.20.118.16 flashsystem820a@hursley.ibm.com
Email gateway set to '9.20.118.16'
```

```
admin #: system callhome_config
The call home feature will send emails from flashsystem820a@hursley.ibm.com via
9.20.118.16.
```

Configuring the call home heartbeat schedule

Example A-5 shows the commands that were used to configure and confirm the call home heartbeat schedule.

Example A-5 Configuring the call home heartbeat settings with the IBM FlashSystem CLI

```
admin #: system callhome_heartbeat 1 5
Call home heartbeat updated.
```

```
admin #: system callhome_heartbeat
The call home heartbeat will be sent at 5:00 every day. Full heartbeat will be
sent on Monday.
```

Configuring the call home events notification

Example A-6 shows the command used to enable the system call home events feature.

Example A-6 Configuring the call home events settings with the IBM FlashSystem CLI

```
admin #: system callhome_events enable
Call home events reporting enabled.
```

Configuring the events notification feature

The same commands were used on each IBM FlashSystem to configure this feature.

Enabling the events notification feature

Example A-7 shows the command that was used to enable email notifications.

Example A-7 Enabling mail notifications with the IBM FlashSystem CLI

```
admin #: mail notifications enable
Mail notifications enabled
```

Configuring the events notification mail server

Example A-8 on page 132 shows the command that was used to configure the mail server to use for sending email notifications.

Example A-8 Configuring the mail notifications server with the IBM FlashSystem CLI

```
admin #: mail server 9.20.118.16
Mail server set to '9.20.118.16'
```

Configuring the events notification target addresses

Example A-9 shows the commands that were used to add an email address to the list of notification targets, and to view the list of email targets.

Example A-9 Configuring the mail notifications targets with the IBM FlashSystem CLI

```
admin #: mail targets add myemailaddress@us.ibm.com
Added new mail target 'myemailaddress@us.ibm.com'

admin #: mail targets
mail targets
The following emails will receive system notifications:
myemailaddress@us.ibm.com
```

Configuring Fibre Channel port settings using the CLI

The same commands were used on each IBM FlashSystem to configure this feature.

Configuring the speed setting of the FC controller ports

Example A-10 shows the commands that were used to set the speed of the Fibre Channel (FC) controller ports.

Example A-10 Setting the speed of the FC controller ports in the IBM FlashSystem CLI

```
admin #: fc speed fc-1a 8Gb
Channel fc-1a's link speed has been changed to 8Gb
admin #: fc speed fc-1b 8Gb
Channel fc-1b's link speed has been changed to 8Gb
admin #: fc speed fc-2a 8Gb
Channel fc-2a's link speed has been changed to 8Gb
admin #: fc speed fc-2b 8Gb
Channel fc-2b's link speed has been changed to 8Gb
```

Configuring the topology setting of the FC controller ports

Example A-11 shows the commands used to set the topology of the FC controller ports.

Example A-11 Setting the topology of the FC controller ports in the IBM FlashSystem CLI

```
admin #: fc topology fc-1a PP
Channel fc-1a's link topology has been changed to 'PP'
admin #: fc topology fc-1b PP
Channel fc-1b's link topology has been changed to 'PP'
admin #: fc topology fc-2a PP
Channel fc-2a's link topology has been changed to 'PP'
admin #: fc topology fc-2b PP
Channel fc-2b's link topology has been changed to 'PP'
```

Note: After modifying the speed or topology settings of a FlashSystem FC port, you must issue an **fc link_reset** command to reset the port in order for it to log in to the switch with the new settings, as shown in Example A-12.

Resetting the FC controller ports

Example A-12 shows the commands that were used to reset the links of the FC controller ports.

Example A-12 Resetting the links of the FC controller ports in the IBM FlashSystem CLI

```
admin #: fc link_reset fc-1a
Are you sure you want to reset 'fc-1a'? (y/n)
y
```

The link has been reset on channel fc-1a

```
admin #: fc link_reset fc-1b
Are you sure you want to reset 'fc-1b'? (y/n)
y
```

The link has been reset on channel fc-1b

```
admin #: fc link_reset fc-2a
Are you sure you want to reset 'fc-2a'? (y/n)
y
```

The link has been reset on channel fc-2a

```
admin #: fc link_reset fc-2b
Are you sure you want to reset 'fc-2b'? (y/n)
y
```

The link has been reset on channel fc-2b

Confirming the status of the FC controller ports

Example A-13 shows the command that was used to confirm the settings of the FC controller ports.

Example A-13 Viewing the FC port settings using the IBM FlashSystem CLI

```
admin #: fc
```

Controller	State	Firmware	Serial
fc-1	GOOD	6309	R-1C91
fc-2	GOOD	6309	R-1C92

Channel	Link_State	Link_Speed	Topology
fc-1a	ONLINE	8Gb	F_PORT
fc-1b	ONLINE	8Gb	F_PORT
fc-2a	ONLINE	8Gb	F_PORT
fc-2b	ONLINE	8Gb	F_PORT

Configuring logical unit numbers and access policies using the CLI

The two IBM FlashSystem storage systems that are referenced in the example environment were configured with differing numbers of logical unit numbers (LUNs) and LUN sizes. Accordingly, the commands that were used to create the LUNs and some of the commands used to create the associated access policies were different for each system. The differences are noted, where applicable.

Creating LUNs using the CLI

The LUN creation commands differed for each example IBM FlashSystem.

Creating LUNs on FLASHSYSTEM_A

In Example A-14, we show the CLI commands that were used to create 16 LUNs, each of size 1.17 TiB on FLASHSYSTEM_A.

Example A-14 Creating LUNs on FLASHSYSTEM_A using the CLI

```
admin #: lu create FLASHSYSTEM_A_SVC_1 1199g 0
Logical Unit 'FLASHSYSTEM_A_SVC_1' created with number 0 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_2 1199g 1
Logical Unit 'FLASHSYSTEM_A_SVC_2' created with number 1 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_3 1199g 2
Logical Unit 'FLASHSYSTEM_A_SVC_3' created with number 2 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_4 1199g 3
Logical Unit 'FLASHSYSTEM_A_SVC_4' created with number 3 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_5 1199g 4
Logical Unit 'FLASHSYSTEM_A_SVC_5' created with number 4 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_6 1199g 5
Logical Unit 'FLASHSYSTEM_A_SVC_6' created with number 5 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_7 1199g 6
Logical Unit 'FLASHSYSTEM_A_SVC_7' created with number 6 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_8 1199g 7
Logical Unit 'FLASHSYSTEM_A_SVC_8' created with number 7 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_9 1199g 8
Logical Unit 'FLASHSYSTEM_A_SVC_9' created with number 8 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_10 1199g 9
Logical Unit 'FLASHSYSTEM_A_SVC_10' created with number 9 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_11 1199g 10
Logical Unit 'FLASHSYSTEM_A_SVC_11' created with number 10 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_12 1199g 11
Logical Unit 'FLASHSYSTEM_A_SVC_12' created with number 11 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_13 1199g 12
Logical Unit 'FLASHSYSTEM_A_SVC_13' created with number 12 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_14 1199g 13
Logical Unit 'FLASHSYSTEM_A_SVC_14' created with number 13 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_15 1199g 14
Logical Unit 'FLASHSYSTEM_A_SVC_15' created with number 14 and size 1.17 TiB
admin #: lu create FLASHSYSTEM_A_SVC_16 1199g 15
Logical Unit 'FLASHSYSTEM_A_SVC_16' created with number 15 and size 1.17 TiB
```

Creating LUNs on FLASHSYSTEM_B

In Example A-15 on page 135, we show the CLI commands that were used to create eight LUNs, each of size 2.34 TiB on FLASHSYSTEM_B.

Example A-15 Creating LUNs on FLASHSYSTEM_B using the CLI

```
admin #: lu create FLASHSYSTEM_B_SVC_1 2399g 0
Logical Unit 'FLASHSYSTEM_B_SVC_1' created with number 0 and size 2.34 TiB
admin #: lu create FLASHSYSTEM_B_SVC_2 2399g 1
Logical Unit 'FLASHSYSTEM_B_SVC_2' created with number 1 and size 2.34 TiB
admin #: lu create FLASHSYSTEM_B_SVC_3 2399g 2
Logical Unit 'FLASHSYSTEM_B_SVC_3' created with number 2 and size 2.34 TiB
admin #: lu create FLASHSYSTEM_B_SVC_4 2399g 3
Logical Unit 'FLASHSYSTEM_B_SVC_4' created with number 3 and size 2.34 TiB
admin #: lu create FLASHSYSTEM_B_SVC_5 2399g 4
Logical Unit 'FLASHSYSTEM_B_SVC_5' created with number 4 and size 2.34 TiB
admin #: lu create FLASHSYSTEM_B_SVC_6 2399g 5
Logical Unit 'FLASHSYSTEM_B_SVC_6' created with number 5 and size 2.34 TiB
admin #: lu create FLASHSYSTEM_B_SVC_7 2399g 6
Logical Unit 'FLASHSYSTEM_B_SVC_7' created with number 6 and size 2.34 TiB
admin #: lu create FLASHSYSTEM_B_SVC_8 2399g 7
Logical Unit 'FLASHSYSTEM_B_SVC_8' created with number 7 and size 2.34 TiB
```

Creating aliases using the CLI

The same commands were used on each IBM FlashSystem to create the aliases.

In Example A-16, we show the CLI commands used to add logical aliases for the four SAN Volume Controller FC ports used to connect to the IBM FlashSystem storage systems.

Example A-16 Creating aliases using the IBM FlashSystem CLI

```
admin #: lu alias add SVC_ND_1_P1 50:05:07:68:01:40:ba:da
Added alias 'SVC_ND_1_P1' for host '50:05:07:68:01:40:ba:da'
admin #: lu alias add SVC_ND_1_P5 50:05:07:68:01:50:ba:da
Added alias 'SVC_ND_1_P5' for host '50:05:07:68:01:50:ba:da'
admin #: lu alias add SVC_ND_2_P1 50:05:07:68:01:40:ba:d8
Added alias 'SVC_ND_2_P1' for host '50:05:07:68:01:40:ba:d8'
admin #: lu alias add SVC_ND_2_P5 50:05:07:68:01:50:ba:d8
Added alias 'SVC_ND_2_P5' for host '50:05:07:68:01:50:ba:d8'
```

Creating access policies using the CLI

The same commands were used on each IBM FlashSystem to create the access policy groups.

Creating the access policy group

In Example A-17, we show the CLI commands that were used to create the access policy group that will allow the SAN Volume Controller FC ports to access the LUNs that we created in Example A-14 on page 134, and Example A-16. The first CLI command in the example creates the access policy group with a single entry. The subsequent 15 CLI commands add additional entries this policy group.

Example A-17 Creating access policy group entries on the IBM FlashSystem using the CLI

```
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P1_FC-1A fc-1a 50:05:07:68:01:40:ba:da
Added Access Policy Entry 'SVC_ND_1_P1_FC-1A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P1_FC-1B fc-1b 50:05:07:68:01:40:ba:da
Added Access Policy Entry 'SVC_ND_1_P1_FC-1B' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P1_FC-2A fc-2a 50:05:07:68:01:40:ba:da
Added Access Policy Entry 'SVC_ND_1_P1_FC-2A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P1_FC-2B fc-2b 50:05:07:68:01:40:ba:da
```

```

Added Access Policy Entry 'SVC_ND_1_P1_FC-2B' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P5_FC-1A fc-1a 50:05:07:68:01:50:ba:da
Added Access Policy Entry 'SVC_ND_1_P5_FC-1A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P5_FC-1B fc-1b 50:05:07:68:01:50:ba:da
Added Access Policy Entry 'SVC_ND_1_P5_FC-1B' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P5_FC-2A fc-2a 50:05:07:68:01:50:ba:da
Added Access Policy Entry 'SVC_ND_1_P5_FC-2A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_1_P5_FC-2B fc-2b 50:05:07:68:01:50:ba:da
Added Access Policy Entry 'SVC_ND_1_P5_FC-2B' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P1_FC-1A fc-1a 50:05:07:68:01:40:ba:d8
Added Access Policy Entry 'SVC_ND_2_P1_FC-1A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P1_FC-1B fc-1b 50:05:07:68:01:40:ba:d8
Added Access Policy Entry 'SVC_ND_2_P1_FC-1B' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P1_FC-2A fc-2a 50:05:07:68:01:40:ba:d8
Added Access Policy Entry 'SVC_ND_2_P1_FC-2A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P1_FC-2B fc-2b 50:05:07:68:01:40:ba:d8
Added Access Policy Entry 'SVC_ND_2_P1_FC-2B' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P5_FC-1A fc-1a 50:05:07:68:01:50:ba:d8
Added Access Policy Entry 'SVC_ND_2_P5_FC-1A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P5_FC-1B fc-1b 50:05:07:68:01:50:ba:d8
Added Access Policy Entry 'SVC_ND_2_P5_FC-1B' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P5_FC-2A fc-2a 50:05:07:68:01:50:ba:d8
Added Access Policy Entry 'SVC_ND_2_P5_FC-2A' to Group 'SVC_FLASHSYSTEM'
admin #: lu access group add SVC_FLASHSYSTEM SVC_ND_2_P5_FC-2B fc-2b 50:05:07:68:01:50:ba:d8
Added Access Policy Entry 'SVC_ND_2_P5_FC-2B' to Group 'SVC_FLASHSYSTEM'

```

Note: At the time this book was written, the CLI **lu access group add** command was not accepting the port aliases as a parameter. Therefore, Example A-17 on page 135 shows that command using the worldwide port name (WWPN) instead.

Adding the access policy group to each LUN

The commands to add the access policies to the LUNs differed for each example IBM FlashStorage storage system.

Adding access policies to the LUNs on FLASHSYSTEM_A

In Example A-18, we show the CLI commands that were used to add the access policy group to each of the 16 LUNs that were created in Example A-14 on page 134.

Example A-18 Adding the access policy groups to the LUNs on FLASHSYSTEM_A using the CLI

```

admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_1
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_1'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_2
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_2'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_3
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_3'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_4
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_4'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_5
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_5'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_6
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_6'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_7
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_7'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_8
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_8'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_9
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_9'

```

```
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_10
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_10'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_11
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_11'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_12
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_12'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_13
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_13'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_14
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_14'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_15
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_15'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_A_SVC_16
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_A_SVC_16'
```

Adding access policies to the LUNs on FLASHSYSTEM_B

In Example A-19, we show the CLI commands that were used to add the access policy group to each of the eight LUNs that were created in Example A-16 on page 135.

Example A-19 Adding the access policy groups to the LUNs on FLASHSYSTEM_B using the CLI

```
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_1
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_1'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_2
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_2'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_3
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_3'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_4
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_4'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_5
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_5'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_6
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_6'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_7
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_7'
admin #: lu access add group SVC_FLASHSYSTEM FLASHSYSTEM_B_SVC_8
Access Policy Group 'SVC_FLASHSYSTEM' added to 'FLASHSYSTEM_B_SVC_8'
```



B

Example environment details

This appendix provides detailed information about the example environment that was used to write this book. The following topics are covered:

- ▶ Example environment hardware components
- ▶ Example environment firmware and software levels
- ▶ Example environment topology

Example environment hardware components

All the references to the example environment in this book are related to the lab that was built to run the tests. All the components were installed, configured, and provided the operational environment for all the activities documented in this book, such as window captures, command-line interface (CLI) command executions and results, and validation of the sequence of tasks.

List of hardware components

IBM FlashSystem 820

Two subsystems are required, each one with the following specifications and configuration options:

- ▶ Machine type/model 9831-AE2
- ▶ Raw maximum capacity: 33 TB/30 TiB
- ▶ Storage mode: RAID-5
- ▶ Storage usable capacity: 20.61 TB/18.75 TiB
- ▶ 4 x 8 Gbps Fibre Channel ports
- ▶ 4 x 40 Gbps quadruple data rate (QDR) InfiniBand ports (not used in this example environment)
- ▶ 1U rack space

IBM SAN Volume Controller (SVC) engines

Two virtualization engines are required, each one with the following specifications and options:

- ▶ Machine type/model 2145-CG8
- ▶ 8x 8 Gbps Fibre Channel ports
- ▶ 48 GB of processor memory
- ▶ Two Intel Xeon 5600 Series six-core processors
- ▶ 2U rack space (including the uninterruptible power supply (UPS))

IBM SAN24B-5 SAN switches for SVC private SAN

- ▶ Machine type/model 2498-F24
- ▶ 24x 16 Gbps Fibre Channel ports
- ▶ 1U rack space

IBM SAN48B-5 SAN switches for SVC public SAN

- ▶ Machine type/model 2498-F48
- ▶ 48x 16 Gbps Fibre Channel ports
- ▶ 1U rack space

IBM System x3650 M4 server

- ▶ Machine type/model 7915-85G
- ▶ 4x 8 Gbps Fibre Channel ports
- ▶ 64 GB of RAM
- ▶ Two Intel Xeon E5-2600 six-core 2.0 GHz processors
- ▶ 2U rack space

Example environment firmware and software levels

IBM FlashSystem 820

Firmware level: 6.3.0-p9

IBM SAN Volume Controller software

Software level: 7.1.0.2

IBM SAN24B-5 SAN switches

Fabric-OS version: v7.0.2a

IBM SAN48B-5 SAN switches

Fabric-OS version: v7.0.2a

IBM System x3650 M4 server

VMware ESXi 5.1

Example environment topology

Figure B-1 shows a logical diagram of the example environment.

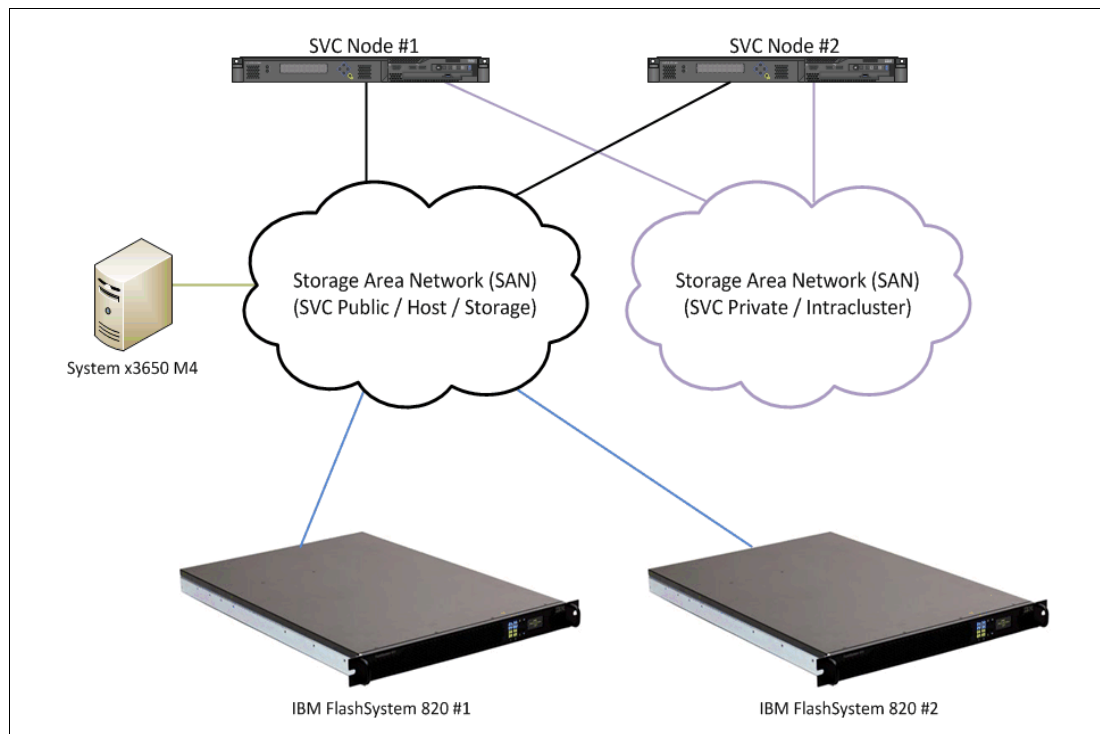


Figure B-1 Diagram of the example environment

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

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- ▶ *IBM SAN Volume Controller and IBM FlashSystem 820: Best Practices and Performance Capabilities*, REDP-5027
- ▶ *IBM FlashSystem 710 and IBM FlashSystem 810*, TIPS1002
- ▶ *IBM FlashSystem 720 and IBM FlashSystem 820*, TIPS1003
- ▶ *Flash or SSD: Why and When to Use IBM FlashSystem*, REDP-5020
- ▶ *IBM FlashSystem in a Virtual Desktop Environment Solution Guide*, TIPS1029
- ▶ *IBM FlashSystem in OLAP Database Environments*, TIPS0974
- ▶ *IBM FlashSystem in OLTP Database Environments*, TIPS0973
- ▶ *IBM System Storage Solutions Handbook*, SG24-5250
- ▶ *IBM SAN and SVC Stretched Cluster and VMware Solution Implementation*, SG24-8072
- ▶ *IBM SAN Volume Controller Stretched Cluster with PowerVM and PowerHA*, SG24-8142
- ▶ *IBM System Storage SAN Volume Controller and Storwize V7000 Replication Family Services*, SG24-7574
- ▶ *IBM System Storage SAN Volume Controller Best Practices and Performance Guidelines*, SG24-7521

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- ▶ *IBM FlashSystem 820 User's Guide*, GI11-9896
- ▶ *IBM FlashSystem Web Interface Guide*
- ▶ *IBM FlashSystem Integration Guide*
- ▶ *IBM FlashSystem SNMP Guide*
- ▶ *IBM FlashSystem Installing the IBM FlashSystem 820*, GI11-9896
- ▶ *IBM FlashSystem Troubleshooting and Service*

Online resources

These websites are also relevant as further information sources:

- ▶ IBM Flash Storage and Solutions home page
<http://www.ibm.com/systems/storage/flash>
- ▶ IBM SAN Volume Controller product home page
<http://www.ibm.com/systems/storage/software/virtualization/svc>
- ▶ IBM SAN Volume Controller support matrixes portal
<http://www.ibm.com/support/docview.wss?uid=ssg1S1003658>
- ▶ IBM SAN Volume Controller Information Center
<http://pic.dhe.ibm.com/infocenter/svc/ic/index.jsp>
- ▶ IBM Intelligent Cluster
<http://www-03.ibm.com/systems/x/hardware/largescale/cluster>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Implementing the IBM SAN Volume Controller and FlashSystem 820

(0.2"spine)
0.17"<->0.473"
90<->249 pages



Implementing the IBM SAN Volume Controller and FlashSystem 820

Enhance storage capabilities with sophisticated virtualization

Move data among virtualized storage systems

Optimize flash storage deployments automatically

In today's 24 x 7 world, there is likely not a business on this planet, IBM Smarter Planet or not, that finds that their storage requirements are growing too fast and demand is starting to outpace supply. Not only this, but in this cost-conscious environment of today, the costs of managing this growth are likely to be eating into the IT budget.

One way to make better use of existing storage without adding more complexity to the infrastructure is the IBM System Storage SAN Volume Controller (SVC). For many years now this has helped business become more flexible, agile, and introduced an extremely efficient storage environment. SAN Volume Controller is designed to deliver the benefits of storage virtualization in environments from large enterprises to small businesses and midmarket companies.

Now, with IBM FlashSystem storage, SAN Volume Controller is enabled to extend its reach and benefit all virtualized storage. For example, IBM Easy Tier optimizes use of flash storage. And IBM Real-time Compression enhances efficiency even further by enabling the storage of up to five times as much active primary data in the same physical disk space.

In this IBM Redbooks publication, we show how to integrate the IBM FlashSystem 820 to provide storage to the SAN Volume Controller, and show how they are designed to operate seamlessly together, reducing management effort.

In this book, which is aimed at pre- and post-sales support, storage administrators, and people that want to get an overview of this new and exciting technology, we show the steps required to implement the IBM FlashSystem 820 in an existing SAN Volume Controller environment. We also highlight some of the new features in SAN Volume Controller that increase performance.

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

SG24-8172-00

ISBN 0738438634