

Faster Microsoft SQL Server Performance with IBM FlashSystem Storage

IBM Redbooks Solution Guide

This IBM® Redbooks® Solution Guide provides an overview of how to identify storage performance bottlenecks and improve Microsoft SQL Server database performance by using IBM® FlashSystem™ storage to accelerate the most resource-intensive data operations.

Fast storage and low latency are vital to maximizing SQL Server performance. Adding servers or processing power often does not improve SQL performance because the system is limited by the speed of data access time. The result is a massive performance gap, often felt most by database servers, which typically carry out more I/O transactions than other systems. Super-fast processors and massive amounts of bandwidth are often wasted as storage devices take several milliseconds to access requested data.

Figure 1 depicts the importance of latency, which applications see as time waiting. Input/output (operations) per second (IOPS), defined as a measure of work versus time, is important, but low latency is key to increased SQL Server performance.

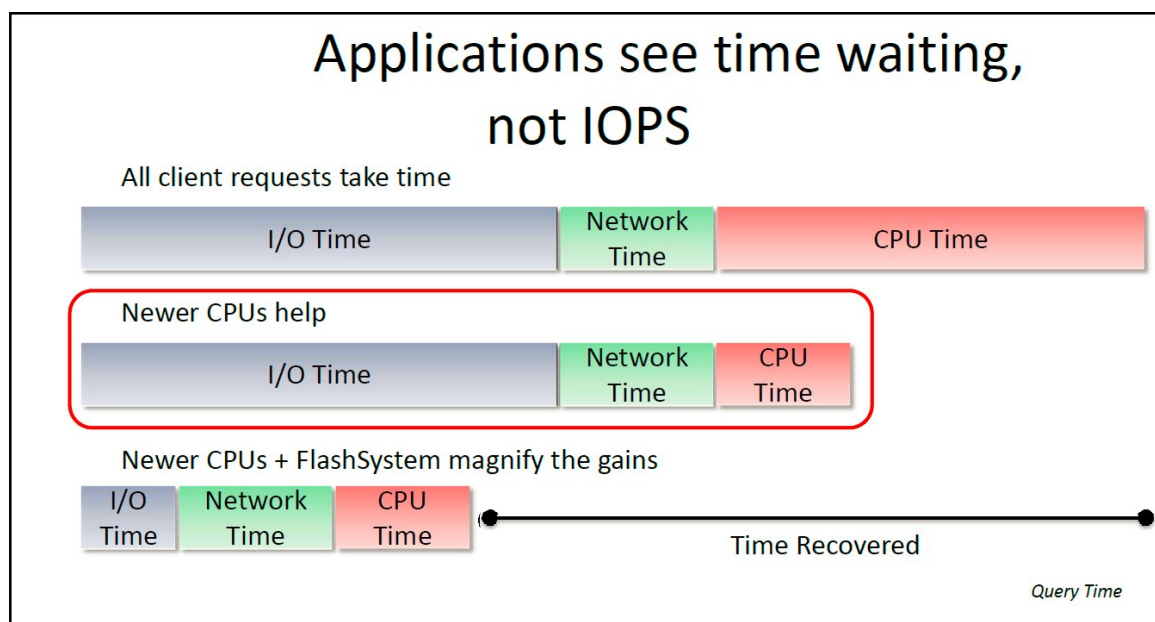


Figure 1. FlashSystem products minimize wait time

Did you know?

- While traditional storage solutions take several milliseconds to access data, FlashSystem products use IBM MicroLatency™ to access storage in a matter of microseconds.
- You can move your most I/O intensive applications to FlashSystem products to improve the performance of your entire environment with minimal impact to existing configurations.
- Consolidating your storage onto FlashSystem products can help you reduce floor space, server costs, and software licensing fees.

Business value

IBM FlashSystem products provide extremely fast storage performance with an extremely small footprint. They offer extraordinarily high random I/O performance because of their low access times and high bandwidth. You can deliver more and have faster SQL transactions with FlashSystem storage.

Decreasing application performance under heavy user loads is not a new story for most enterprises. In the last three years, there have been dramatic changes in demands placed upon database servers. While the number of users of database systems has increased, so has the average amount of data stored in databases. Additionally, the demand for more complex business analysis has increased the complexity of the work done by database servers. The combination of more users, greater volume of data, and more complex queries has frequently resulted in slower database response. Administrators often consider the following solutions for database performance problems:

- Server and processor performance. One of the first steps that most IT shops do when performance decreases is to add processors and memory to servers or add servers to server farms. Adding servers and processors has a minimal impact on database performance that is storage bound, and compounds the resources that are wasted as even more processing power waits on the same slow storage.
- SQL statements. Enterprises invest millions of dollars squeezing every bit of efficiency out of their SQL statements. The software tools that assist programmers with the assessment of their SQL statements can cost tens of thousands of dollars. The personnel required for evaluating and iterating the code costs much more. In the last decade, many consulting firms have appeared that specialize in system tuning, and their number-one billable service is SQL tuning.

Tuning SQL can create performance improvements, but even the best SQL cannot make up for poor storage I/O. In many cases, features that rely heavily on disk I/O cannot be supported by applications. In particular, programs that result in large queries and those that return large data sets are often removed from applications to protect application performance.

In many cases, the money spent in these two pursuits can be significant, but the return is often disappointing. Server performance and SQL tuning alone do not often address the true cause of poor database performance; the gap between processor performance and storage performance. Adding servers and processors will have minimal impact on database performance and will instead compound the resources wasted, as more processing power waits on the same slow storage.

When system administrators look to storage, they frequently try two approaches to resolve performance problems:

- Increase the number of disks. Adding disks to JBOD (just a bunch of disks) or RAID is one way to improve storage performance. This approach can increase the I/O capability because I/O growth is a linear function when increasing the drive count, but has a trivial impact on the bottleneck if the performance issue is related to latency. You can scale I/O and bandwidth with disks if you have the space and the budget to do so, but you cannot gain the benefit of microsecond latency no matter how many disks you add. FlashSystem storage systems are capable of providing over one million IOPS within a single 2U appliance. FlashSystem storage combines low latency with writes at 90 microseconds and reads at 135 microseconds, to provide extreme performance and MicroLatency for critical business needs.
- Select and implement a hybrid RAID array. A common reaction is to look into hybrid storage architectures that include a mix of solid state drive (SSD) and hard disk drive (HDD). These approaches have a few pitfalls. Hybrid arrays are nearly all optimized for HDD and cannot take advantage of the speed of flash hybrid arrays; due to their use of commodity SSDs, are relatively space, power, and performance inefficient. Conversely, the hardware-only data path in FlashSystem products, with extremely low-latency custom FPGA-based data movement decreases latency and reduces the need to continually increase the number of hard disks required to reach high IOPS.

The best solution to the performance gap is to implement all flash storage rather than HDD or hybrid solutions for the most frequently accessed database components.

While other components in the data chain move in terms of computation times and the raw speed of electricity through a circuit, hard drives move mechanically, relying on physical movement around a magnetic platter to access information. Form factor solid-state drive (SSD) technologies are limited by the speed of SAS controllers, and many locally attached PCI Express (PCIe) solutions require additional software to manage reliability and scaling. FlashSystem products provide a hardware-only data path that offers much faster shared data access than any of these other solutions.

When servers wait on storage, users wait on servers. This is I/O wait time. FlashSystem arrays are designed to minimize I/O wait time by providing a fast, hardware-only data path. Depending on the model, FlashSystem products offer 37 times faster access times (135 microseconds instead of 5 milliseconds) and 2750 times more I/O transactions per second (1,100,000 instead of up to 400) than a hard disk drive. Although multiple hard disk drives can be stacked to obtain thousands of IOPS, this type of solution soon reaches a point of diminishing returns where the costs of power, floor space, and air conditioning become prohibitive. Figure 2 shows latency for flash products compared to other types of storage technologies.

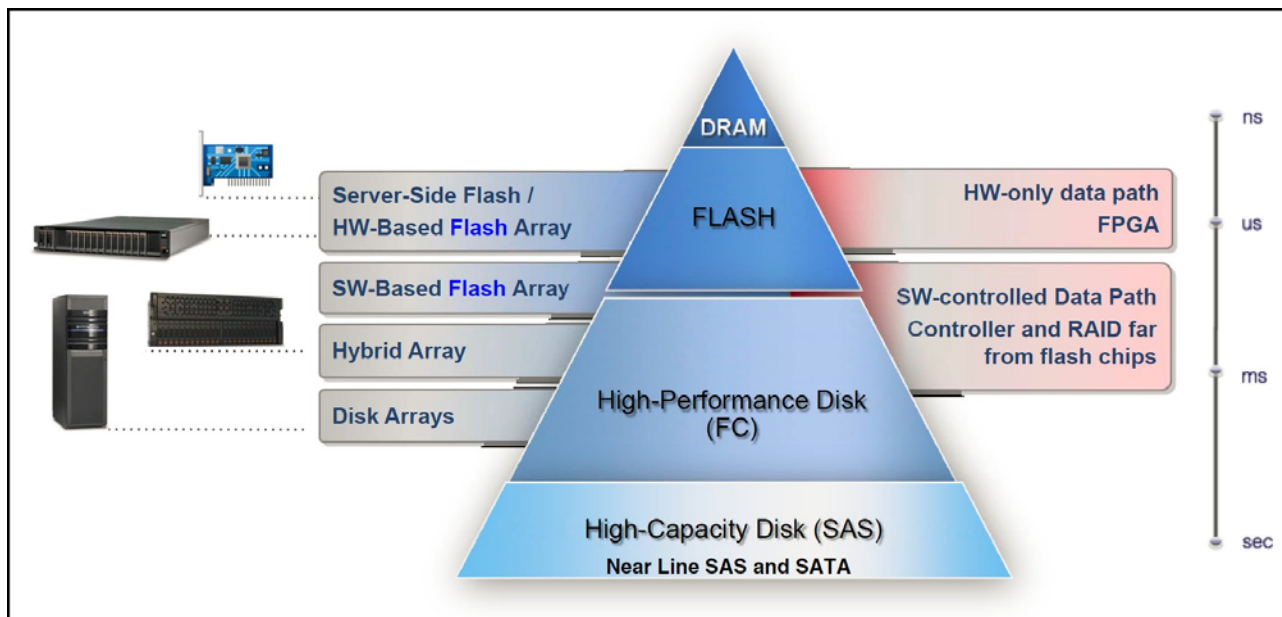


Figure 2. Latencies seen in storage technologies

FlashSystem products provide four key benefits:

- Extreme performance: Optimize scalability, aggregate IOPS, and bandwidth to ensure that as database workloads increase, FlashSystem continues to scale performance while minimizing latency degradation.
- MicroLatency: Radically reduce I/O wait time with access times below 200 microseconds.
- Macro efficiency: Save money on IT staffing, power, and cooling.
- Enterprise reliability: Durable and reliable designs use enterprise grade flash, Variable Stripe RAID™ technology to reduce maintenance events, two-dimensional RAID technology, and hot-swap capability.

Read more about IBM FlashSystem reliability in the white paper at the following link:

www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&htmlfid=TS03139USEN

Solution overview

Disk latency, which is defined as a measure of the delay from the time a disk I/O request is created until the time the disk I/O request is completed, directly affects Microsoft SQL Server performance. IBM FlashSystem storage products can reduce bottlenecks caused by frequent disk access. This section discusses methods for identifying I/O performance bottlenecks and lists components that are the best candidates for migration to a flash storage appliance.

Although flash can speed up almost any SQL Server database, it is most needed in installations where servers experience I/O wait time. Several elements are involved in server I/O. The PCI (or other) bus, host bus adapter, interface, storage network switch, RAID controller, and hard disk drives are all involved in every I/O operation between server and storage. Theoretically, any one of these points can cause an I/O bottleneck. In practice, however, the hard disk drives are the most likely culprit. Simply put, every component in the I/O process is solid state except for the hard disk drives. Therefore, when you identify I/O wait time, hard disk drives are the most likely cause. Adding flash storage can significantly reduce I/O wait time.

The best way to identify I/O wait time is to look at operating system performance. The tools to evaluate operating system performance vary by operating system.

For Microsoft Windows operating systems, the best tool for system performance analysis is the Performance Monitor (Perfmon) program. This tool uses counters to monitor the *speed* of disk transfers and the I/O operations that are generated by SQL Server.

Unfortunately, the Performance Monitor program does not provide the actual I/O wait time statistic, but it does include actual processor performance levels. The “Processor: % Processor Time” statistic shows the actual work that is being done by the processor. If transactions have a large impact on your system, but the “% Processor Time” statistic is well under 100%, it is possible to infer an I/O wait problem. Systems that implement flash storage show high “% Processor Time” numbers.

Figure 3 shows the “Processor: % Processor Time” for a Windows system that is running Iometer software performing 100% random writes to a hard disk drive. In this exhibit, the processor utilization is less than 1%. Running additional applications on this system might increase the processor utilization only marginally, because the processor is waiting for I/O from the hard disk drive. In the following example, Iometer shows that on average, there were 150 writes per second (150 IOPS) to the disk drive.

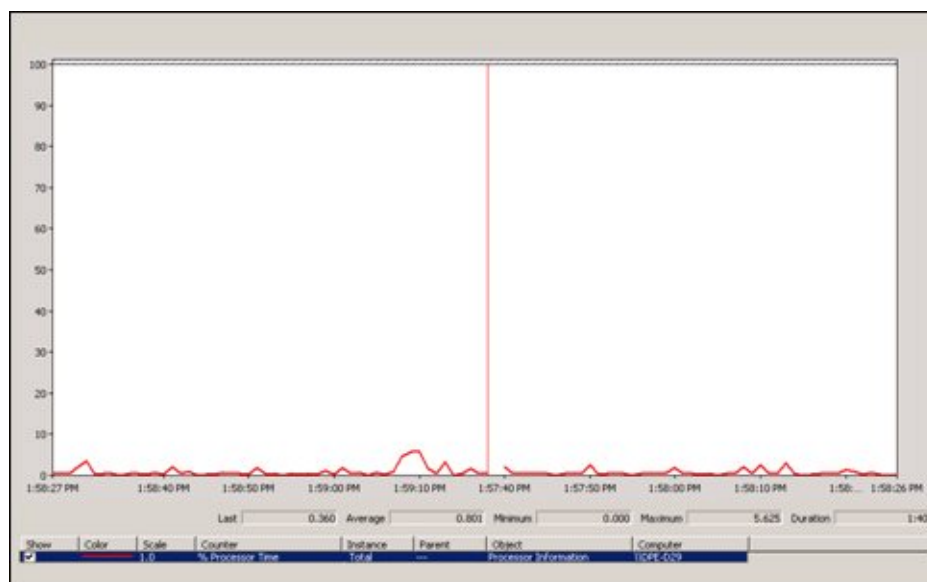


Figure 3. Processor performance when writing to hard disk

Figure 4 shows the same system with the same access specifications in the Iometer software running against FlashSystem storage. In this example, the processor averages 68% utilization. The Iometer software shows that 138,000 writes per second are going to the FlashSystem array (138,000 IOPS). As it turns out, this IOPS number is limited by the host bus adapter or it might be even higher. Nonetheless, this example clearly illustrates how flash storage can improve processor utilization for a Windows-based system.

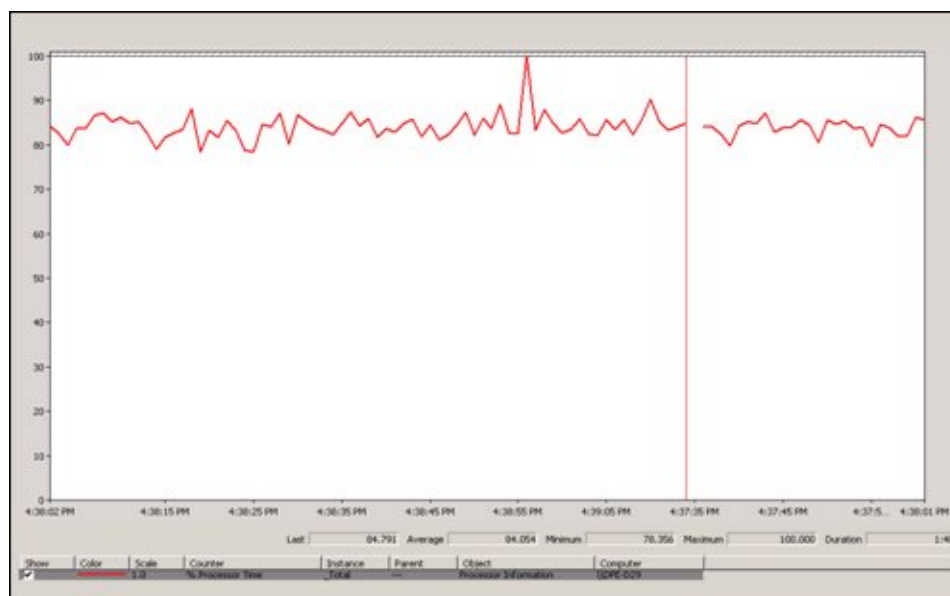


Figure 4. Processor performance when writing to FlashSystem

In addition to processor indicators included in Windows performance Monitor, you can look at the “Physical Disk: Average Queue Length” and “Physical Disk: Disk Bytes Per Second” indicators to detect bottlenecks in the disk subsystem. If the “Physical Disk: Average Disk Queue Length” is consistently high, consider moving files that are located on that disk to flash storage. A “Disk Queue Length” greater than 3 (per disk in the RAID set) indicates a problem.

The “Physical Disk: Disk Bytes Per Second” indicator helps you visibly determine the level of performance you are getting when your disk array levels off. The point at which disk bytes per second level off and physical disk queue lengths increase is the point when your system is demonstrating an I/O bottleneck. In addition, if you simultaneously graph Processor Utilization, you notice that processor time tends to decrease when your system experiences an I/O bottleneck. After you narrow down the time range when I/O problems occur, further refine your search for the problem by using Perfmon’s ability to narrow details by reads/writes and by drive arrays. In this way, you can quickly determine which array is causing I/O problems.

For statistics to confirm the I/O wait witnessed by the application, you can also load the SQL Server: “Wait Statistics” objects into the Windows Perfmon program. These counters give an application’s perception of Wait categories for real-time queries. The most prominent waits are typically the Page I/O Latch waits, and Log buffer and Log Write waits.

With these tools, you can observe the I/O wait time for a Windows-based system.

Solution architecture

FlashSystem products are storage area network (SAN) solutions with Fibre Channel, iSCSI, Fibre Channel over Ethernet (FCoE), or InfiniBand connectivity. Base building blocks are available in a small 2U form factor, and larger scalable systems are also available.

Figure 5 shows FlashSystem architectural components.

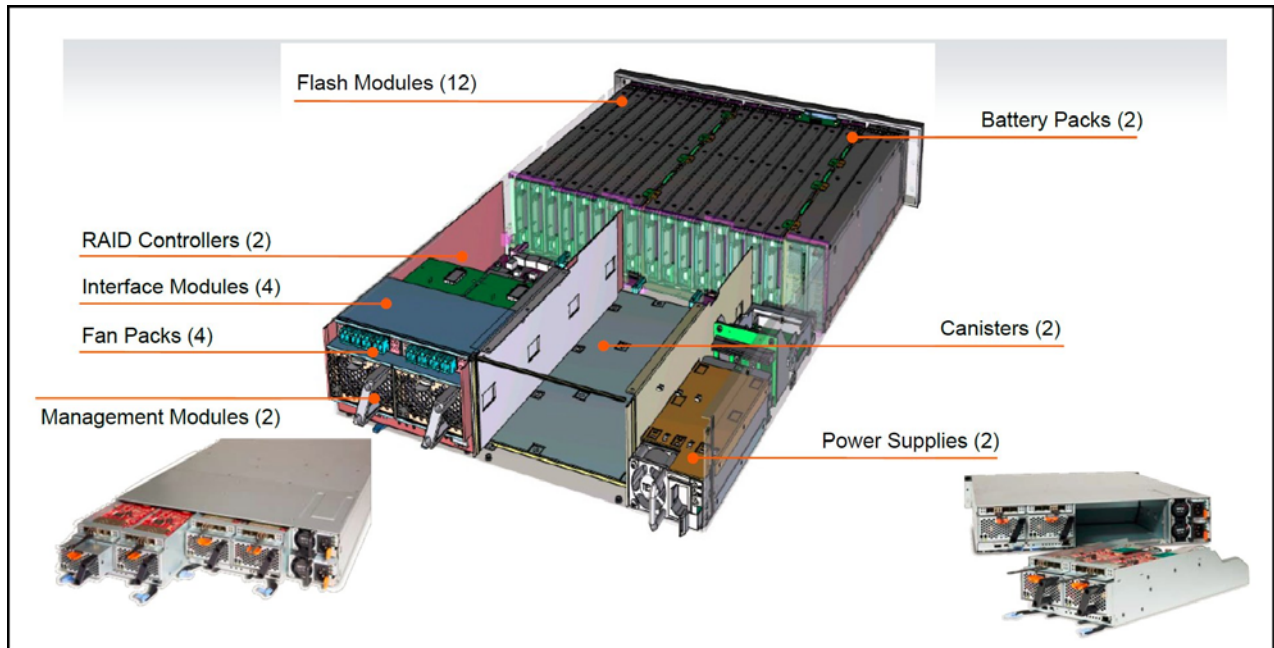


Figure 5 FlashSystem components

Usage scenarios

IBM FlashSystem technology can provide benefits across any enterprise environment where higher database performance and lower IT costs are crucial for success. The following usage scenarios address various aspects of database performance, including transaction logs, temporary space, indexes, and even entire databases.

Improving database performance

If you determine that your system has I/O subsystem problems, the next step is to identify which components of your SQL Server database are experiencing the highest I/O and in turn contributing the most to I/O wait time. Consider moving several components like transaction logs, temporary database (tempdb), indexes, and frequently accessed tables to FlashSystem storage, or for simplicity and maximum benefit, you can even move the entire databases to FlashSystem.

Entire database

Some databases should have all of their files moved to flash storage. These databases tend to have at least one of the following characteristics:

- High concurrent access. Consider moving all data to flash storage for databases that are used by a large number of concurrent users. This ensures that storage is not a bottleneck for the application and maximizes the utilization of servers and networks. I/O wait time will be minimized and servers and bandwidth will be fully utilized.
- Frequent random accesses to all tables. For some databases, identifying a subset of files that are frequently accessed is not possible. Many times these databases are effectively large indices.
- Small to medium size databases. Given the fixed costs associated with buying traditional storage systems, buying FlashSystem storage to store small to medium sized databases is often economical. FlashSystem products offer a variety of capacity points, so you can often start with a few terabytes of storage for less than the price of most enterprise RAID systems.
- Large read-intensive databases. Given the fixed costs associated with architecting a traditional storage system for performance (buying large cache and buying a lot of spindles for striping), it is economical and much simpler to buy a flash solution to accelerate large read-intensive databases. FlashSystem can be deployed in preferred read mirror configurations to preserve the value and resiliency of the existing storage infrastructure while still providing extreme performance acceleration in read-oriented environments.
- Database performance is the key to company profitability. Some subset of databases help companies make more money, lose less money, or improve customer satisfaction if they process faster. In such environments an investment in flash storage can directly contribute to the top or bottom line.

Transaction logs

Transaction logs are one of the most important factors in the write-performance for the SQL Server databases. When a database write occurs, SQL Server creates a transaction log entry. Each SQL Server database has at least one transaction log (*.ldf). The SQL Server considers operations committed when the write to the transaction logs is complete.

Because the transaction logs are a source of constant I/O during database operation, it is recommended to store the transaction logs on the fastest storage system possible. Writing a transaction log to FlashSystem storage is a natural way to improve overall database performance. The combination of performance and two-dimensional resiliency provided by FlashSystem technology makes it a great replacement for the inefficient RAID 10 disk (striped and mirrored) storage configurations frequently used for transaction logs.

Temporary database (tempdb)

The SQL Server uses tempdb to support temporary data during certain operations. The tables support complex queries, joins, and index creations. Because tempdb supports many kinds of operations, locating them on the fastest possible storage is essential.

When complex operations occur, they complete more quickly if the tempdb is moved to FlashSystem storage. Because the I/O to the tempdb can be frequent, disk drives cannot handle them easily.

Indexes

An index is a data structure that speeds up access to database records. An index is usually created for each table in a database. The SQL Server updates these indexes when adding records and modifying identifying data for records. When a read occurs, the SQL Server consults an index to access the correct record quickly. Furthermore, many concurrent users might read any index simultaneously. This activity to the disk drive creates frequent, small, and random transactions, which makes the disk drives unable to keep up with demand and I/O wait time results.

Storing indexes on flash can increase the entire application's performance. For online transaction processing (OLTP) systems with a high number of concurrent users, this results in faster database access. Historically, indexes have been a common SQL Server component moved to flash because of their high volume, small size, and random transaction characteristics.

Frequently accessed tables

Sometimes, only 5 - 10% of tables account for a large percentage of all database activity, and thus I/O to storage. When a large number of users hit a table, they are likely going after different records and different attributes. As a result, the activity on that table is random. Disk drives are notoriously bad at servicing random requests for data. In fact, the peak performance of a disk drive drops as much as 95% when servicing random transactions. When a table experiences frequent access, transaction queues develop where other transactions are waiting on the disk to service the next request. These queues are another sign that the system is experiencing excessive I/O wait time.

Moving frequently accessed tables to FlashSystem storage makes sense in this case. Unlike with traditional hard drives, random transactions do not affect flash performance. Additionally, flash has faster access times than disk drives. Based on latency comparisons alone, moving frequently accessed tables to flash can improve application performance up to 20 times, though of course such benefits depend on the overall scalability of your applications.

SQL Server supports moving specific tables to specific file groups, which enables you to more easily move the most frequently accessed tables to flash storage.

Using the Microsoft SQL Server Profiler tool to identify the most frequently accessed SQL Server files and tables

The SQL Server Profiler tool allows database administrators to monitor events in SQL Server. The profiler monitors SQL Server activities by tracing selected events, including database objects. Database objects include databases, tables, indexes, views, and stored procedures. The SQL Server Profiler specifically supports monitoring when a table is opened. Tables are opened during Select, Insert, and Delete statements. By using the profiler, a database administrator can monitor the number of times all tables are opened during the trace period. The results of the trace can be sent to a table for further analysis to determine which tables were most frequently opened during the trace period and thus which tables are the best candidates for flash acceleration.

Object-related traces can impose a significant overhead on database performance because of the frequency with which objects are accessed. If you have a test environment that models production activity and transaction mixes, this might be the best place to use an SQL Server Profiler "Object" trace. Otherwise, ensure you do not use the Profiler during peak transaction periods unless you are willing to take the performance hit.

For more information about SQL Server Profiler, visit the Microsoft MSDN website:
<http://msdn.microsoft.com/en-us/library/ms181091.aspx>

To improve performance, physically isolate database objects, such as tables and indexes, into unique file groups. With file groups, management and monitoring become more granular than at the database level, and database administrators are able to avoid the high overhead of the SQL Server Profiler to find hot spots. Instead, this monitoring is done with Dynamic Management Views (DMVs).

The DMV `dm_io_virtual_file_stats` (Figure 6) is a query of I/O counters and cumulative wait times for each type of I/O. These counters represent the aggregate values since the last boot of the SQL Server instance. Because these are running counters, a difference in query results is needed to get a proper assessment of I/O wait. Record one query result before a busy period of the application, and record a second query result after the busy period. Recording each query result in a spreadsheet application, such as Microsoft Excel, is ideal for this analysis.

```
select CURRENT_TIMESTAMP as SnapShotTime,
db.name as DatabaseName,
mf.physical_name as FilePath,
mf.name as LogicalName,
vfs.*
from sys.dm_io_virtual_file_stats(-1,-1) as vfs
join sys.master_files
    as mf on vfs.database_id = mf.database_id and vfs.file_id= mf.file_id
join sys.databases
    as db on vfs.database_id= db.database_id;
```

Figure 6. Running a DMV `dm_io_virtual_file_stats` query

From this query, you can extract the counters for every database with their respective file statistics. The results can then be presented into a more visual form using Microsoft Excel with a Pivot Chart to group files into file groups.

Figure 7 is an example that shows the number of I/O stalls logged against each file's size in an IBM Tivoli® Storage Productivity Center benchmark database.

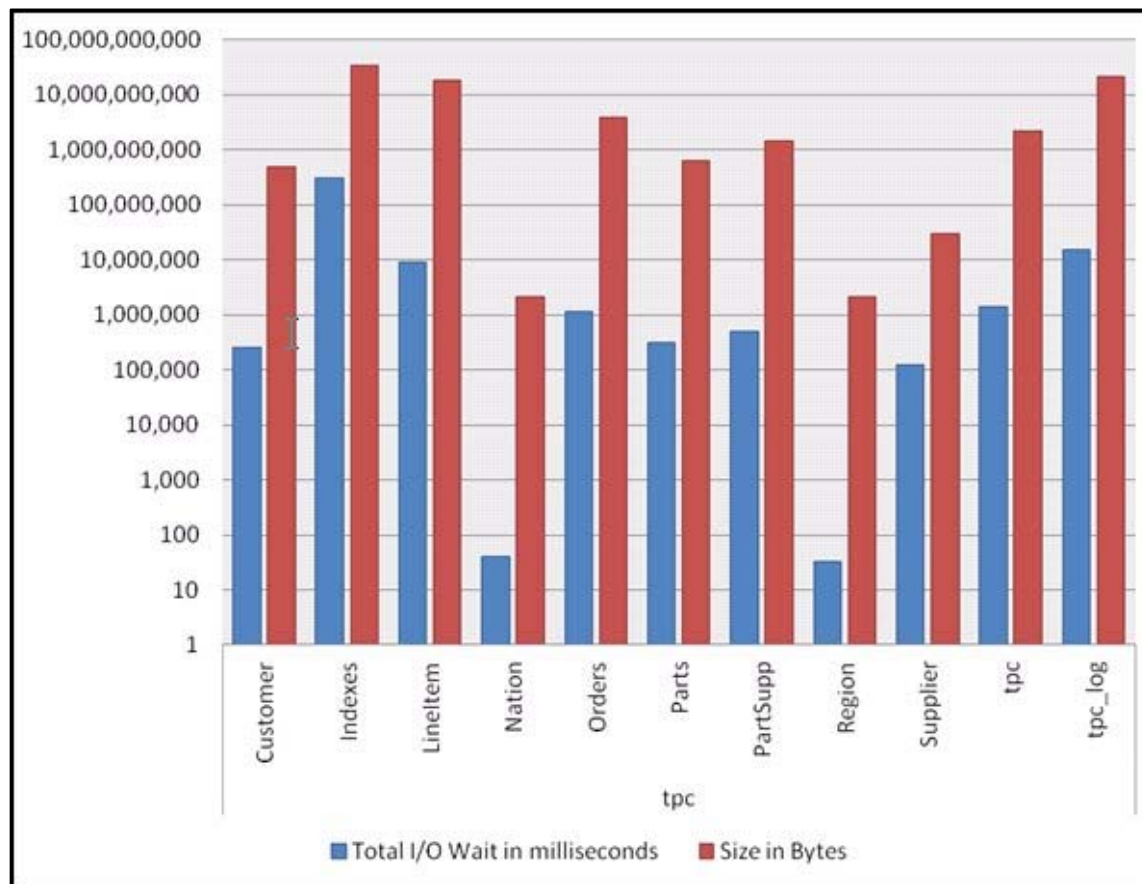


Figure 7. Graphic view of query

Integration

FlashSystem products can be easily integrated into your existing storage environment, whether as a Tier 0 storage pool or as part of a Tier 1 feature-rich deployment, with the following key features:

- Thin provisioning to allocate storage "just in time"
- Improved utilization to harvest all SAN capacity
- Disaster avoidance with location-proof data availability
- IBM 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 about FlashSystem integration with Independent Software Vendors (ISV) and other IBM solutions, visit the FlashSystem Ecosystem at:

<http://ibm.com/storage/flash/ecosystem/>

Supported platforms

Supported hardware and platforms for the FlashSystem product portfolio are listed at the IBM System Storage Interoperation Center (SSIC):

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

Ordering information

FlashSystem storage products can be ordered as (IBM FlashSystem 840) or the IBM FlashSystem V840 Enterprise Performance Solution (FlashSystem V840) may be ordered to provide advanced functionality, such as thin provisioning, IBM Easy Tier, FlashCopy, and Real-time Compression.

For FlashSystem ordering information, see the following IBM Redbooks® Product Guides:

- *IBM FlashSystem 840 Product Guide*, TIPS1079
<http://www.redbooks.ibm.com/abstracts/tips1079.html>
- IBM FlashSystem V840 Enterprise Performance Solution, TIPS1158
<http://www.redbooks.ibm.com/abstracts/tips1158.html>

Related information

For more information, see the following documents:

- IBM FlashSystem family product page
<http://www.ibm.com/storage/flash>
- IBM Redbooks publication, Implementing IBM FlashSystem 840, SG24-8189-01
<http://www.redbooks.ibm.com/redpieces/abstracts/sg248189.html?Open>
- IBM Redbooks Solution Guides for IBM FlashSystem family:
<http://www.redbooks.ibm.com/redbooks.nsf/searchsite?SearchView&query=flashss>
- IBM Support Portal
<http://ibm.com/support/entry/portal/>
- IBM Offering Information page (to search on announcement letters, sales manuals, or both):
http://www.ibm.com/common/ssi/index.wss?request_locale=en

On this page, enter **FlashSystem**, select the information type, and then click **Search**. On the next page, narrow your search results by geography and language.

Notices

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

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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 measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

© 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.

This document was created or updated on August 22, 2014.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
ibm.com/redbooks
- Send your comments in an e-mail to:
redbook@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 U.S.A.

This document is available online at <http://www.ibm.com/redbooks/abstracts/tips1038.html> .

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:

Easy Tier®
FlashCopy®
FlashSystem™
IBM®
IBM FlashSystem™
MicroLatency™
Redbooks®
Redpaper™
Redbooks (logo)®
System Storage®
Tivoli®
Variable Stripe RAID™

The following terms are trademarks of other companies:

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

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