HITACHI
Inspire the Next

Microsoft GOLD CERTIFIED Partner

Hitachi Adaptable Modular Storage 2000 Family Best Practices with Microsoft® SQL Server

For Online Transaction Processing Applications

By Eduardo Freitas, Reginal A. Hay, Jr., and Rob Simmons

February 2009

Intended Audience

This paper is written for IT professionals who need information on how to configure and monitor the Hitachi Adaptable Modular Storage 2000 family to optimize storage performance, availability and cost for their Microsoft SQL Server applications either as a dedicated platform or in a shared application environment. This document assumes a base level of storage, storage area network (SAN) and Microsoft SQL Server knowledge.

Contributors

The information included in this document represents the expertise, feedback, and suggestions of a number of skilled practitioners. The authors would like to recognize and sincerely thank the following contributors and reviewers of this document

- Alan Benway, Technical Operations
- John Montgomery, Global Solutions Engineering
- Gil Rangel, Technical Operations

For best results use Acrobat Reader 8.0.

Executive Summary

The Hitachi Adaptable Modular Storage 2000 family brings enterprise-class features and availability to the modular storage space, making it easier to deploy and manage storage resources for Microsoft® SQL Server.

With a new management interface and significant controller enhancements, the 2000 family simplifies the disk provisioning and management process for SQL Server, allowing system administrators to focus their time and energy on other issues.

This document describes recommendations and guidelines for configuring and monitoring the 2000 family for SQL Server online transaction processing (OLTP) applications to achieve optimal performance and availability in shared application environment. It is written for IT professionals who need to configure and monitor the 2000 family to optimize storage performance, availability and cost for SQL Server applications.

Contents

Hitachi Adaptable Modular Storage 2000 Family Best Practices with Microsoft® SQL Server

For Online Transaction Processing Applications

By Eduardo Freitas, Reginal A. Hay, Jr., and Rob Simmons

Businesses large and small recognize the need for available, scalable storage foundations for their growing, business critical applications. The Hitachi Adaptable Modular Storage 2000 family brings enterprise-class availability, performance and ease of management to organizations of all sizes. It also offers a robust storage solution that is easy to install and integrate with Microsoft SQL Server, Exchange and other common applications on Windows Server 2008.

Decades of experience are engineered into the 2000 family's intelligent microcode that directs the low-level activities of the storage system. While many options exist to uniquely tune a system's behavior for the most demanding SQL Server applications, the 2000 family controllers make effective I/O decisions, minimizing the need to over-engineer storage configurations. Several considerations are important for all SQL Server deployments, including these:

- Application performance and response time requirements, such as consideration for multiple, shared applications
- Database size and expected growth rate
- Data protection preferences based on recovery point and time objectives

While availability, performance and budget requirements often compete when selecting a storage solution, considering these factors can help you determine the best choice of disk drive technology, RAID configuration and data protection to meet your application needs.

Shared Application Platform Solution Profile

Many organizations use a shared storage platform to serve multiple critical applications. The SQL Server solutions discussed in this paper are intended for organizations that typically have multiple online transaction processing (OLTP) databases, each of which requires less than 500GB of storage. These databases often reside on the same SAN-based storage that supports other applications.

OLTP database application workloads, like accounting, order-entry and manufacturing applications, can have widely ranging read-versus-write ratios that are typically determined by the application design. This ratio influences application performance and storage design because the read-intensive nature of the applications can degrade response times due to unnecessary physical disk access or accelerate response times through the effective use of server and storage cache.

Hitachi Adaptable Modular Storage 2000 Family Features

The Hitachi Adaptable Modular Storage 2000 family provides a reliable, flexible, scalable and cost-effective modular storage system for SQL Server. The 2000 family is ideal for more demanding application requirements and delivers enterprise-class performance, capacity and functionality at a midrange price.

The 2000 family is the only midrange storage product with symmetric active-active controllers that provide integrated, automated hardware-based front-to-back-end I/O load balancing. This ensures I/O traffic to back-end disk devices is dynamically managed, balanced and shared equally across both controllers, even if the I/O load to specific logical units (LUs) is skewed. Storage administrators are no longer required to manually define specific affinities between LUs and controllers, simplifying overall administration. In addition, this new controller design is fully integrated with standard host-based multipathing, thereby eliminating mandatory requirements to implement proprietary multipathing software.

No other midrange storage product that scales beyond 100TB has a serial attached SCSI (SAS) drive interface. The new point-to-point back-end design virtually eliminates I/O transfer delays and contention associated with Fibre Channel arbitration and provides significantly higher bandwidth and I/O concurrency. It also isolates any component failures that might occur on back-end I/O paths.

Flexibility

- Choice of Fibre Channel and iSCSI server interfaces or both
- Resilient performance using LUs that can be configured to span multiple drive trays and back-end paths
- Choice of high-performance SAS and low-cost SATA disk drives
- Lowered costs using SAS or SATA drives that can be intermixed in the same tray
- Support for all major open systems operating systems, host bus adapters (HBAs) and switch models from major vendors

Scalability

- Ability to add capacity, connectivity and performance as needed
- Concurrent support of large heterogeneous open systems environments using up to 2048 virtual ports with host storage domains and 4096 LUs
- Ability to scale capacity to 472TB
- Ability to scale performance to more than 900K IOPS
- Seamless expansion due to data-in-place upgrades from Adaptable Modular Storage 2100 to Adaptable Modular Storage 2300 and to Adaptable Modular Storage 2500
- Large-scale disaster recovery and data migration using integration with Hitachi Universal Storage Platform™ V and Hitachi Universal Storage Platform VM
- Complete lifecycle management solutions within tiered storage environments

Availability

- Outstanding performance and nondisruptive operations using Hitachi Dynamic Load Balancing Controller
- 99.999% data availability
- No single point of failure
- Hot swappable major components

- Dual-battery backup for cache
- Nondisruptive microcode updates
- Flexible drive sparing with no copy back required after a RAID rebuild
- Host multipathing capability
- In-system SQL Server and Exchange backup and snapshot support through Windows Volume Shadow Copy Service
- Remote site replication
- RAID-5, RAID-1, RAID-1+0 and RAID-0 (SAS drives) support
- RAID-6 dual parity support for enhanced reliability when using large SATA and SAS drives
- Hi-Track® Remote Maintenance Tool support

Performance

- No performance bottlenecks in highly utilized controllers due to Hitachi Dynamic Load Balancing Controller
- Point-to-point SAS backplane with a total bandwidth of 96 gigabits per second (Gb/sec) and no overhead from loop arbitration
- Full duplex 3Gb/sec SAS drive interface that can simultaneously send and receive commands or data on the same link
- Up to 32 concurrent I/O paths provide up to 9600 megabytes per second (MB/sec) of total system bandwidth
- 4Gb/sec host Fibre Channel connections
- Cache partitioning and cache residency to optimize or isolate unique application workloads

Simplicity

- Simplified RAID group placement using SAS backplane architecture
- Highly intuitive management software that includes easy-to-use configuration and management utilities
- Command line interface (CLI) and command control interface (CCI) that match graphical user interface (GUI) functionality
- Seamless integration with Hitachi storage systems, managed with a single set of tools using Hitachi Storage Command Suite software
- Consistency among most Hitachi software products whether run on Hitachi modular storage systems or the Hitachi Universal Storage Platform family

Security

- Role-based access to Adaptable Modular Storage management systems
- Ability to track all system changes with audit logging
- Ability to apply system-based "write once, read many" (WORM) data access protection to logical volumes to provide regulatory compliant protection
- Encrypted communications between management software and storage system using SSL and TSL
- Internet Protocol version 6 (IPv6) and Internet Protocol Security (IPsec) compliant maintenance ports

SQL Server in a SAN Environment

SQL Server has multiple components, each of which has a unique I/O profile based on the workload from the application server and the specific design of the application database. The three primary components are database files, transaction logs and `tempdb`. Table 1 from Microsoft highlights some of the different I/O patterns and sizes.

Table 1. Input/Output Patterns of SQL Server 2005

Operation	*Random / Sequential*	*Read / Write*	*Size Range*
OLTP – Log	Sequential	Write	Sector Aligned up to 60K
OLTP – Log	Sequential	Read	Sector Aligned up to 120K
OLTP – Data (Index Seeks)	Random	Read	8K
OLTP – Lazy Writer	Random	Write	Any multiple of 8K up to 256K
OLTP – Checkpoint	Random	Write	Any multiple of 8K up to 256K
Read Ahead (DSS, Index/Table Scans)	Sequential	Read	Any multiple of 8K up to 256K (512K for Enterprise)
Bulk Insert	Sequential	Write	Any multiple of 8K up to 128K
Read Ahead (DSS, Index, Scans)	Sequential	Read	8K Multiples to 256K
Create Database	Sequential	Write	Up to 4MB (only log file initialized in SQL 2005
Backup	Sequential	Read / Write	64K multiples to 4MB
Restore	Sequential	Read / Write	64K multiples to 4MB
DBCC CheckDB	Sequential	Read	8K to 64K
ALTER INDEX REBUILD (Write Phase)	Sequential	Write	Any multiple of 8K up to 128K
ALTER INDEX REBUILD (Read Phase)	Sequential	Read	Any multiple of 8K up to 256K

Note: These values may change in the future based on optimizations made to take advantage of modern storage enhancements.

Source: "Microsoft SQL Server and SAN – Lessons Learned & Best Practices" by Mike Ruthruff and Prem Mehra, SQL Server Customer Advisory Team

Database File Considerations

When deploying SQL Server, it is important to have an overall understanding of the key file types that are essential for a database. Understanding the type of workload that each file type has, along with the database type and size enables both storage and database administrators (DBAs) to establish storage requirements for the SQL Server environment.

SQL Server Database Files

SQL Server OLTP database file I/O is composed of random small record reads and writes. A database might include only a single database file, while those designed to support heavy transactional workloads or large schemas might use a variety of filegroup architectures to improve performance, operational convenience or availability. Acceptable database I/O response times typically range between 5 and 20 milliseconds. In general, whether your database maintains a single `.mdf` file or leverages multiple secondary database files (`.ndf`) and filegroups, the same storage design rules apply because of the random nature of the database I/O profile.

SQL Server Transaction Log File

Every SQL Server database has at least one log file that records database modifications made by each transaction. It is a critical component of the database for availability and data integrity. In the event of a system failure, the active transaction log, at a minimum, is required to bring the database back to a consistent state. The transaction logs are written before the data records are updated to the database file via the checkpoint process. The logs can be used to roll back transactions if corruption is later discovered or to recover a committed transaction if an error writing to the database occurs.

Response time and performance are critical considerations for separating the transaction log from the database files. Microsoft suggests aiming for log I/O response times between 1 and 5 milliseconds. Hitachi's optimized caching and proper storage design ensure that the logs can be written without delay.

One of the features built into the 2000 family is the ability to optimize physical I/O based on recognition of I/O patterns. The 2000 family controller can make optimized timing decisions about when to move the data between mirrored, protected cache and physical disks when it encounters a series of I/O requests. By combining multiple logical I/O requests into a single physical I/O or by optimizing the order of individual reads and writes, the 2000 family can significantly increase overall performance.

SQL Server `tempdb` Files

SQL Server `tempdb` files are used for storage of temporary data structures and can be subject to intense and unpredictable I/O. Many best practice recommendations suggest locating `tempdb` files on separate RAID groups from the database and using the fastest disks available. This is generally a safe recommendation because the load on `tempdb` is highly dependent upon database and application design. However, if the `tempdb` load is well understood and monitored regularly, testing shows that it can reside on the same RAID group as the primary database without adverse effects. Accordingly, if the environment does not have sufficient physical disk I/O resources to meet the combined requirements of `tempdb` and the database files, performance for all databases in the SQL Server instance degrades. Therefore, you need a good understanding of your `tempdb` usage, regardless of where you choose to place the `tempdb` files.

Key Considerations

- Place database and log files on physically separate RAID groups.
- Place log files on RAID-1 or RAID-1+0, depending on their capacity requirement.
- Select the appropriate RAID type for database files based on performance, cost and availability requirements. RAID-1+0 provides the best performance and availability, but at a higher cost. RAID-5 can provide good performance with slightly lower availability and at a lower cost. RAID-6 can also be considered if the higher availability is important enough to justify reduced write performance due to the additional RAID-6 write overhead.
- For cost-sensitive environments, use RAID-5 for transaction logs rather than RAID-1+0. In some cases, RAID-5 performs as well as RAID-1+0 on the 2000 family for sequential workloads like Microsoft SQL Server log files.
- Create one `tempdb` file per CPU core and make all files equal in size.
- Place `tempdb` files on a separate RAID-1+0 group from the database and log files with the fastest hard disk drives available. However, with caution, `tempdb` files can reside on the same RAID group as the database files. To simplify monitoring, always place `tempdb` files in a dedicated LU, regardless of RAID group placement. `tempdb` LU usage and growth must also be evaluated prior to deployment and monitored regularly.

Storage Considerations

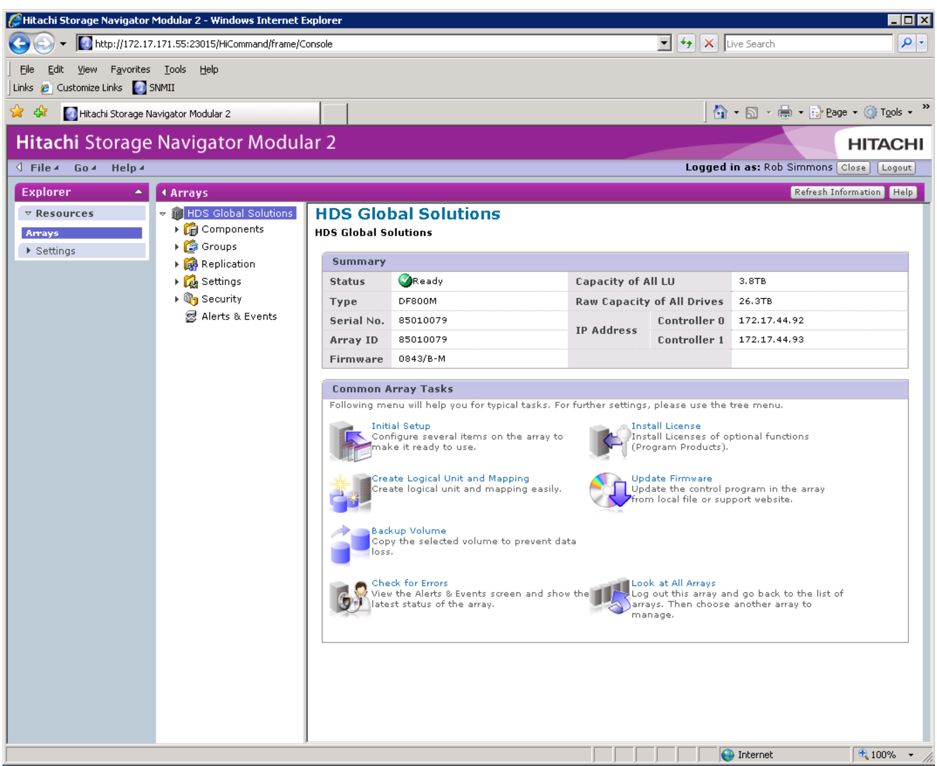
With the Hitachi Adaptable Modular Storage 2000 family, the new Dynamic Load Balancing Controller architecture allows host access to a LU from any of the front-end host ports on the two controller modules. This type of design is commonly referred to as symmetrical active-active.

To ensure that both controllers maintain optimal performance, the system automatically balances the workload when it detects an imbalance of CPU utilization between the two controllers. This feature eliminates bottlenecks that occur when one of the controllers reaches its maximum throughput level while the other controller remains underutilized.

For example, I/O received by a port on Controller 0 can be transferred to Controller 1 over an efficient and high-speed PCI-express data path. The performance impact of this transfer is minimal to non-detectable. Therefore, system administrators do not need to spend time analyzing their resource requirements to set the ownership of servers to the ports on the controllers.

Also new with the 2000 family is the release of Hitachi Storage Navigator Modular 2, which features a new look and feel that is familiar to Hitachi Storage Command Suite users. Along with user-supported maintenance and GUI-based replication setup, new wizards simplify the most commonly executed tasks such as RAID group and LU creation. The new version also offers a simplified online microcode update procedure.

Figure 1. Storage Navigator Modular 2



Drive Considerations

Selecting a drive type is as important as selecting a RAID type for a storage implementation. With the introduction of the 2000 family, users can choose between SAS and SATA drives to meet their applications' performance or capacity requirements. Hitachi recommends SAS drives for solutions where high performance is critical for business operations. SATA drives continue to be the low-cost option for environments where high performance is not critical to business operations and higher capacities are required. Due to performance and availability requirements, Hitachi recommends the use of SAS drives for SQL Server database, log and `tempdb` files and SATA drives for secondary copies, dump and scratch space for cost savings.

When using one of the 2000 family storage systems, use Table 2 as a guideline for calculating the number of spare drives needed; keep in mind that these are the minimum number of spares recommended, and in some cases such as remote offices additional spare drives might be appropriate.

Table 2. Minimum Spare Disk Drive Recommendations for SAS or SATA

Number of Hard Disk Drives per Type	Minimum Recommended Spare Drives
5–30	1
31–50	2
51–100	3
101–480	4

Key Considerations

- Use SAS drives for SQL Server database, log and `tempdb` files.
- Use several smaller physical disks in a RAID group to improve performance through increased spindle count.
- Use Table 2 as a guideline for calculating the number of spare drives to use for a given environment.

Host Considerations

To ensure optimal performance from your storage system, it is important to understand disk alignment, NTFS allocation unit size, path redundancy, HBA placement and queue depth during the planning process.

Disk Alignment

Both disk alignment offset and NTFS allocation unit size must be set when the LUNs are partitioned and formatted at the operating system level before Microsoft SQL Server database files are created. Windows Server 2008 eliminates the need to align the partition, due to the use of 1024K as the default partition offset. For Windows Server 2003, use `Diskpart.exe` to ensure that the disk tracks are sector and cache aligned. Use an offset of 64KB (128 sectors) for every disk that is part of the SQL environment. It is important to remember that this is a destructive process and must be performed before database files are created to prevent data loss.

NTFS Allocation Unit Size

Choose an allocation unit size that matches the I/O profile of your application to allow more efficient I/O on your disk subsystem. When formatting your SQL LUNs in Windows Server, override the default setting and specify an allocation unit size of 64KB.

HBA Drivers

When choosing HBA drivers, ensure that you are using the current recommended drivers for the 2000 family. Major HBA vendors allow you to download current drivers for Hitachi storage systems. For a list of currently supported HBAs and drivers, see the Interoperability Information section on the Hitachi Data Systems Web site.

HBA Queue Depth Settings

Queue depth settings determine how many command data blocks can be sent to a port at one time. Setting queue depth too low can artificially restrict an application's performance, while setting it too high might cause a slight reduction in I/O. Setting queue depth correctly allows the controllers on the Hitachi storage system to optimize multiple I/Os to the physical disk. This can provide significant I/O improvement and reduce response time.

Many applications that use SQL Server are I/O intensive and can have many concurrent, outstanding I/O requests. For that reason, better performance is generally achieved with higher queue depth settings. However, this must be balanced with the available command data blocks on each front-end port of the storage system.

The Adaptable Modular Storage 2000 family has a maximum of 512 command data blocks available on each front-end port. This means that at any one time up to 512 active host channel I/O commands can be queued for service on a front-end port. The 512 command data blocks on each front-end port are used by all LUs presented on the port, regardless of the connecting server. When calculating queue depth settings for your Microsoft SQL Server HBAs, you must also consider queue depth requirements for other LUs presented on the same front-end ports to all servers. Hitachi recommends setting HBA queue depth on a per-target basis rather than per-port basis.

To calculate queue depth, use the following formula:

512 ÷ total number of LUs presented through the front-end port = HBA queue depth per host

For example, suppose that four servers share a front-end port on the storage system, and between the four servers, 16 LUs are assigned through the shared front-end port and all LUs are constantly active. The maximum dynamic queue depth per HBA port is 32, that is:

512 command data blocks ÷ 16 LUs presented through the front-end port = 32 HBA queue depth setting

Key Considerations

- Use a 64KB allocation unit size for database, transaction logs and `tempdb` files.
- Use the most current Hitachi supported HBA drivers.
- Select the proper HBA queue depth using the formula described above.

Planning for Availability

Business critical SQL Server deployments require SAN data path redundancy with multiple path access to storage resources. Many of these deployments also require local quick recovery data protection for creating secondary copies of the database and log files.

SAN Data Path Redundancy

Hitachi recommends the use of dual SAN fabrics, multiple HBAs and host-based multipathing software when deploying business critical SQL Server applications. Two or more paths from the SQL Server connecting to two independent SAN fabrics is essential for ensuring the redundancy required for critical applications.

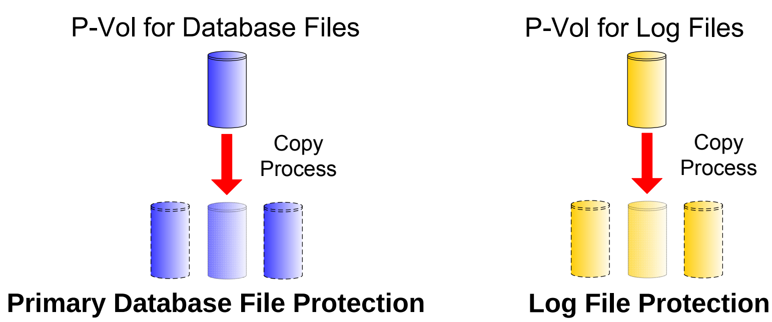
Multipathing software such as Hitachi Dynamic Link Manager and Microsoft Windows Server 2008 native MPIO are critical components of a highly available system. Multipathing software allows the Windows operating system to see and access multiple paths to the same LU, enabling data to travel down any available path for increased performance or continued access to data in the case of a failed path. While multiple load balancing settings exist in both Hitachi Dynamic Link Manager and Windows Server 2008 native multipath I/O (MPIO), the symmetrical active-active controller feature of the 2000 family enables either controller to respond to I/O, regardless of the originating HBA port and without having to select a host load balancing option. However, if the workload is large enough to consume more bandwidth than a single HBA port can handle, Hitachi recommends using the round robin load-balancing algorithm in both Hitachi Dynamic Link Manager and Windows Server 2008 native MPIO to distribute load evenly over all available HBAs.

Basic Local Data Protection Using In-System Replication Technology

The ability to test or debug applications with current data that is truly a replica of production data is critical. The clones created in-system by Hitachi ShadowImage® Replication software are exactly this. Hitachi Protection Manager[1], ShadowImage and Hitachi Copy-on-Write Snapshot software using Microsoft's Virtual Device Interface (VDI) framework allow users to create hardware-based clones and snapshots with minimal effect on the production SQL Server environment. These technologies can create multiple copies, each of which can be used for a variety of purposes, such as point-in-time backup, reporting, test and development, or decision support. In addition, the clones can be updated on a regular basis to include any changes or new data.

Figure 2 highlights basic local data protection using ShadowImage. Each primary volume (P-VOL) and secondary volume (S-VOL) container in the following figure represents a single SQL Server storage volume (for example, database file volume or log volume). Two P-VOL containers correspond to the two primary volumes: one for the database and the other for logs. The S-VOL containers represent the three volumes that contain the point-in-time recovery copies of these files. The S-VOL containers that are dashed in the figure represent multiple copies, which Hitachi Data Systems recommends to improve recoverability in the case of corruption or other data loss that is not discovered immediately. Each additional S-VOL provides an additional recovery point that enhances the protection level of the overall SQL Server database.

Figure 2. Basic Replication Capabilities of Hitachi ShadowImage Replication Software



The most recent ShadowImage copy of the database can be mounted and used offline to validate data integrity of the database and to identify potential corruption before more serious damage can occur. The ShadowImage copy can also be used to create a full backup without affecting the production system.

Key Considerations

- Use at least two HBAs and place them on different buses within the server to distribute the workload over the server's PCI bus architecture.
- Use at least two Fibre Channel switch fabrics to provide multiple independent paths to the 2000 family storage systems to prevent configuration errors from bringing down the entire SAN infrastructure.
- For large bandwidth requirements that surpass a HBA's port capability, use round robin as the load-balancing setting for either Hitachi Dynamic Link Manager or Windows Server 2008 native MPIO.

[1] SQL Server 2008 will be supported in a future release of Hitachi Protection Manager

- Maintain multiple copies of your database and log files with ShadowImage or Copy-on-Write Snapshot software using the Microsoft VDI framework for quick recovery.

Performance Monitoring

A complete, end-to-end picture of your SQL Server environment and continual monitoring of capacity and performance are key components of a sound database management strategy. Monitor servers, operating systems, SQL Server instances, databases, database applications, storage and IP networks, and the 2000 family using tools such as Windows Performance Monitor (PerfMon), Hitachi Performance Monitor feature and SQL Server stored procedures, Dynamic Management Views (DMV) and Dynamic Management Functions (DMF).

Note that while PerfMon provides good overall I/O information about a SQL Server environment and specific SQL Server performance characteristics, it cannot identify all possible bottlenecks in an environment. For a good overall understanding of the I/O profile of a given SQL Server host, monitor the storage system's performance with Hitachi Performance Monitor feature. Combining data from at least two performance-monitoring tools provides a more complete picture of the SQL Server environment. Remember that PerfMon is a per-server monitoring tool and cannot provide a holistic view of the storage system. For a complete view, use PerfMon to monitor all servers that are sharing a RAID group.

Windows Performance Monitor

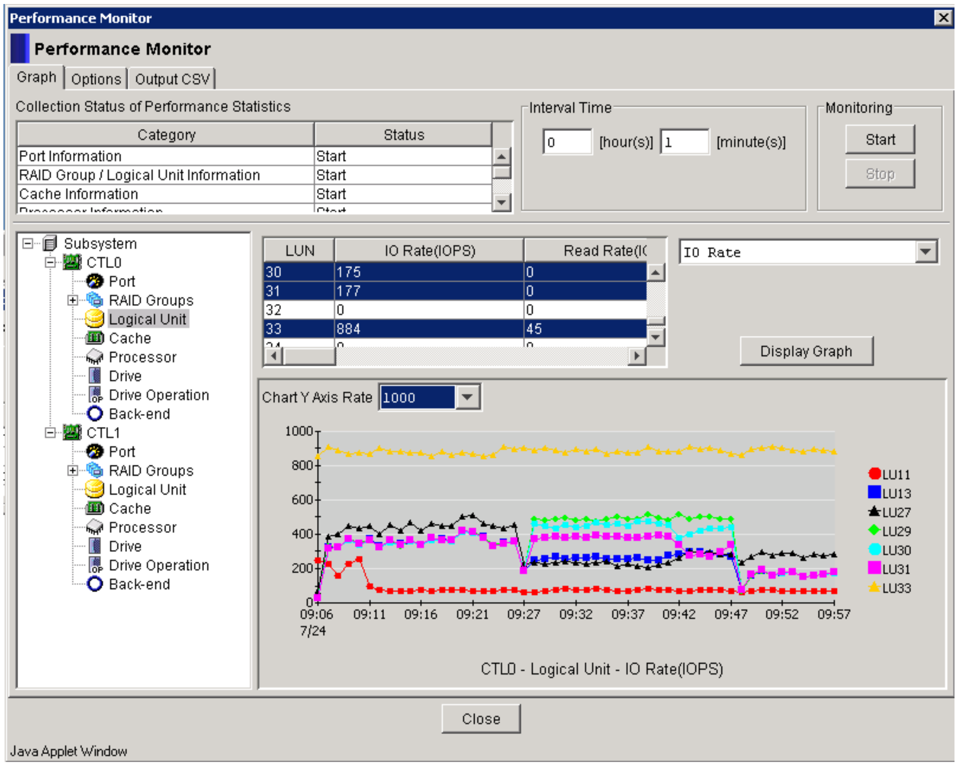
PerfMon is a Windows-based application that allows administrators to monitor the performance of a system using counters or graphs, in logs or as alerts on the local or remote host. SQL Server performance counters are included in PerfMon's counters. Use these counters to identify and troubleshoot possible bottlenecks that the SQL Server might be experiencing. For more information about monitoring host-related counters, see the "Monitoring I/O Performance Using System Monitor" section of Microsoft's *Predeployment I/O Best Practices for Microsoft SQL Server* article on TechNet.

Hitachi Performance Monitor Feature

Hitachi Performance Monitor feature is a controller-based software application, enabled through Hitachi Storage Navigator 2, which monitors the performance of RAID groups, logical units and other elements of the disk subsystem while tracking utilization rates of resources such as hard disk drives and processors (see Figure 3). Information is displayed using line graphs in the Performance Monitor windows and can be saved in comma-separated value (`.csv`) files.

When the disk subsystem is monitored using Hitachi Performance Monitor feature, utilization rates of resources in the disk subsystem (such as load on the disks and ports) can be measured. When a problem such as slow response occurs in a host, an administrator can use Hitachi Performance Monitor feature to quickly determine if the disk subsystem is the source of the problem.

Figure 3. Hitachi Performance Monitor Feature



SQL Server Stored Procedures

SQL Server uses DMV and DMF information to provide DBAs with instrumentation to monitor their SQL Server environment. DMV and DMF information contains useful views and functions that enable users not only to monitor the health of their SQL Server instance, but most importantly provide useful information that enables DBAs to diagnose and tune performance related problems on the database design. Using the DMV information, DBAs can obtain extremely granular information on databases, query statistics, I/O statistics and operating system level information such as stacks, tasks and threads currently running on the SQL server.

For more information on troubleshooting SQL Server, see *Troubleshooting Performance Problems in SQL Server 2005* and *SQL Server 2005 Waits and Queues*.

Table 3 provides an overall view of the DMV and DMF information available on SQL Server. For a complete list of all available DMV and DMF along with their description, see the SQL Server TechCenter available at Microsoft TechNet's Web site.

Table 3. Prefixes for DMVs and DMFs

Prefix	***Description***
dm_clr*	Common language runtime related DMV
dm_db*	General database statistics
dm_exec*	Execution related DMV
dm_fts*	Full-text search related DMV
dm_io*	I/O related DMV
dm_qn*	Query notification related DMV
dm_repl	Replication related DMV
dm_broker*	Service broker related DMV
dm_os*	SQL Server operating system related DMV
dm_tran*	Transaction related DMV

It is important to note that dynamic management views and functions return internal, implementation specific state data. This means that the schema and data they return might change in future releases of SQL Server at Microsoft's discretion. For that reason, keep in mind that future releases of DMV and DMF might not be compatible with their current SQL Server 2005 release.

DMV and DMF Examples

Following are some examples of the type of SQL code that can be written to use DMV and DMF and the output they can provide. Keep in mind that this is a very limited view of DMV and DMF capabilities.

Monitoring Database File Information

To track databases file information, use the `sp_helpfile` stored procedure.

```
use HDS_testdb
go
sp_helpfile
```

Running this SQL code produces the results shown in Figure 4.

Figure 4. `sp_helpfile` ***Output***

Results | Messages

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	MSSQL_HDS_testdb_root	1	H:\HDS_testdb_DATA\Root\MSSQL_HDS_testdb_root.mdf	PRIMARY	10240 KB	Unlimited	0 KB	data only
2	MSSQL_HDS_testdb_log_1	2	L:\HDS_testdb\HDS_testdb_Log_1.ldf	NULL	36488960 KB	2147483648 KB	1048576 KB	log only
3	Misc_1	3	H:\HDS_testdb_DATA\Misc\Misc_1.ndf	misc_fg	51200 KB	Unlimited	10240 KB	data only
4	Misc_2	4	I:\HDS_testdb_DATA\Misc\Misc_2.ndf	misc_fg	51200 KB	Unlimited	10240 KB	data only
5	Misc_3	5	J:\HDS_testdb_DATA\Misc\Misc_3.ndf	misc_fg	51200 KB	Unlimited	10240 KB	data only
6	Broker1	6	I:\HDS_testdb_DATA\Broker1\Broker_1.ndf	broker_fg	31795200 KB	Unlimited	10240 KB	data only
7	Broker2	7	H:\HDS_testdb_DATA\Broker2\Broker_2.ndf	broker_fg	31784960 KB	Unlimited	10240 KB	data only
8	Broker3	8	J:\HDS_testdb_DATA\Broker3\Broker_3.ndf	broker_fg	31784960 KB	Unlimited	10240 KB	data only
9	Market1	9	J:\HDS_testdb_DATA\Market1\Market_1.ndf	market_fg	1024000 KB	Unlimited	10240 KB	data only
10	Market2	10	I:\HDS_testdb_DATA\Market2\Market_2.ndf	market_fg	1024000 KB	Unlimited	10240 KB	data only
11	Market3	11	H:\HDS_testdb_DATA\Market3\Market_3.ndf	market_fg	1024000 KB	Unlimited	10240 KB	data only
12	Customer1	12	H:\HDS_testdb_DATA\Customer1\Customer_1.ndf	customer_fg	6748160 KB	Unlimited	10240 KB	data only
13	Customer2	13	J:\HDS_testdb_DATA\Customer2\Customer_2.ndf	customer_fg	6748160 KB	Unlimited	10240 KB	data only
14	Customer3	14	I:\HDS_testdb_DATA\Customer3\Customer_3.ndf	customer_fg	6748160 KB	Unlimited	10240 KB	data only

This information can help determine what files, filegroups and file identifiers are used by the selected database. For this particular output, note the size of the files and their growth in kilobytes (KB). Check to see that all database files for a filegroup are about the same size and grow at the same rate.

Monitoring `tempdb` *Usage*

Use the following SQL code to monitor the number of user objects, internal objects, version stores and free space left in the `tempdb` file. This code uses the database DMVs.

```
SELECT  SUM (user_object_reserved_page_count)     *8 as user_objects_kb
      , SUM (internal_object_reserved_page_count)*8 as internal_objects_kb
      , SUM (version_store_reserved_page_count)  *8 as version_store_kb
      , SUM (unallocated_extent_page_count)      *8 as freespace_kb
      , SUM (mixed_extent_page_count)            *8 as mixedextent_kb
From sys.dm_db_file_space_usage
WHERE database_id = 2
```

Running this SQL code produces the results shown in Figure 5. This information is useful for determining the usage of `tempdb` (`database_id = 2` uniquely identifies `tempdb`).

Figure 5. `tempdb` ***Size Information***

Results | Messages

	user_objects_kb	internal_objects_kb	version_store_kb	freespace_kb	mixedextent_kb
1	512	64	0	15074560	1152

This information can help determine the usage of `tempdb`. A high number of user or internal objects might indicate heavy usage of variables, temporary tables, sort objects and other objects signifying heavy use of `tempdb`. Also note the free space remaining for `tempdb`.

Monitoring Cache Usage

The following SQL code provides the number of data pages each database has in the buffer cache.

```
SELECT   count(*) AS cached_pages
       , CASE database_id
         WHEN 32767 THEN 'ResourceDb'
         ELSE db_name(database_id)
         END
         AS database_name
FROM  sys.dm_os_buffer_descriptors
GROUP BY db_name(database_id)
       , database_id
ORDER BY cached_pages DESC
```

Running this SQL code produces the results shown in Figure 6.

Figure 6. **cached_pages** ***Output***

Results | Messages

	cached_pages	database_name
1	1341967	HDS_testdb
2	38826	tempdb
3	150	ResourceDb
4	124	msdb
5	88	master
6	61	model
7	53	AdventureWorks
8	39	AdventureWorksDW

This information can help DBAs determine which databases are consuming the most cache at a given time.

Monitoring Latch Waits

I/O bottlenecks can sometimes be identified by examining latch waits. A latch wait is a short-term, lightweight synchronization object that accounts for the physical I/O waits when accessing a page for reading or writing and the page is not available in the buffer pool. When the page is not found in the buffer pool, an asynchronous I/O is posted and then the status of the I/O is checked. If I/O is complete, the worker process proceeds normally. Otherwise, it waits on `PAGEIOLATCH_EX` or `PAGEIOLATCH_SH (exclusive or shared respectively)`, depending upon the type of request.

The following SQL query provides I/O wait latch statistics.

```
SELECT   wait_type
       , waiting_tasks_count
       , wait_time_ms
FROM  sys.dm_os_wait_stats
WHERE wait_type like 'PAGEIOLATCH%'
ORDER BY wait_type
```

Running this SQL code produces the results in Figure 7. This DMV table, `sys.dm_os_wait_stats`, lists all wait latch types. This example uses the `WHERE` clause to limit the search to only page latch types.

Figure 7.* `wait_type` *and* `wait_time` *Output

	wait_type	waiting_tasks_count	wait_time_ms
1	PAGEIOLATCH_DT	0	0
2	PAGEIOLATCH_EX	19	303
3	PAGEIOLATCH_KP	0	0
4	PAGEIOLATCH_NL	0	0
5	PAGEIOLATCH_SH	138	4961
6	PAGEIOLATCH_UP	24	249

Note that this information is cumulative and gathered from the time the server starts. To clear these values, either restart the SQL Server instance or make a call to the following ODBC function:

```
DBCC SQLPERF ('sys.dm_os_wait_stats', CLEAR)
```

This function resets all wait statistic information, not just the pagelatch information.

For more information about wait types, see [Microsoft's Help and Support Knowledge Base article 822101]() and the Microsoft Knowledge Base article [SQL Server 2005 Waits and Queues]().

Monitoring I/O Writes and Sizes

When the I/O of one or more databases is suspected to be the cause of a performance issue, the following SQL code can be executed to determine which database or databases and files have the largest I/O and longest average response time.

```
SELECT db_name(database_id) AS Database_Name
       ,file_id
       ,CAST(num_of_reads AS BIGINT) AS [total_num_reads]
       ,CAST(num_of_writes AS BIGINT) AS [total_num_writes]
       ,CAST(io_stall_read_ms AS BIGINT)
        /  CAST(CASE WHEN num_of_reads=0
             THEN 1
             ELSE num_of_reads
             END AS BIGINT) AS [avg_read_stall]
       ,CAST(io_stall_write_ms AS BIGINT)
        /  CAST(CASE WHEN num_of_writes=0
             THEN 1
             ELSE num_of_writes
             END AS BIGINT) AS [avg_write_stall]
       ,CAST(num_of_bytes_read AS BIGINT)  AS [total_bytes_read]
       ,CAST(num_of_bytes_written AS BIGINT) AS [total_bytes_written]
       ,CAST(num_of_bytes_read AS BIGINT)
        /  CAST(CASE WHEN num_of_reads=0
             THEN 1
             ELSE num_of_reads
             END AS BIGINT)
        / 1024  AS [avg_read_size_Kbytes]
       ,CAST(num_of_bytes_written AS BIGINT)
        /  CAST(CASE WHEN num_of_writes=0
             THEN 1
             ELSE num_of_writes
             END AS BIGINT)
        /  1024  AS [avg_write_size_Kbytes]
FROM sys.dm_io_virtual_file_stats(NULL,NULL)
```

Running this SQL code produces the results in Figure 8. This DMV table lists a number of I/O related totals and averages for all databases in a SQL instance. This can help determine whether a specific database might be causing an I/O problem for the entire SQL environment.

Figure 8. Output

Results | Messages

	Database_Name	file_id	total_num_reads	total_num_writes	avg_read_stall	avg_write_stall	total_bytes_read	total_bytes_written	avg_read_size_Kbytes	avg_write_size_Kbytes
1	master	1	38	1	15	20	2367488	8192	60	8
2	master	2	13	19	13	19	475136	81920	35	4
3	tempdb	1	101	15	4	0	6438912	172032	62	11
4	tempdb	2	13	132	0	0	475136	7382016	35	54
5	tempdb	3	14	101	1	0	229376	835584	16	8
6	tempdb	4	14	101	0	0	229376	843776	16	8
7	model	1	45	2	6	19	2777088	16384	60	8
8	model	2	9	7	6	12	458752	36864	49	5
9	msdb	1	34	1	17	82	2056192	8192	59	8
10	msdb	2	13	4	16	33	475136	20480	35	5
11	AdventureWorksDW	1	19	1	15	95	1073152	8192	55	8
12	AdventureWorksDW	2	9	5	17	52	458752	24576	49	4

Note that this information is cumulative and gathered from the time the server starts.

Monitoring I/O Pending

The following SQL query provides I/O pending for each database in a SQL server instance. The DMF takes two parameters, database number and file number, both of which can be null, and produces very granular information on database reads, writes, I/O stalls, number of bytes on disk and more.

```
SELECT  database_id
      , file_id
      , io_stall
      , io_pending_ms_ticks
      , scheduler_address
FROM  sys.dm_io_virtual_file_stats(NULL, NULL) stats
      , sys.dm_io_pending_io_requests as requests
WHERE stats.file_handle = requests.io_handle
```

As shown in Figure 8, an empty result data set is an indication that the databases and queries are well constructed or the databases are not currently very busy.

Figure 9. I/O **`stall`** ***and*** **`pending`** ***Output***

Results | Messages

	database_id	file_id	io_stall	io_pending_ms_ticks	scheduler_address

SQL Server 2005 Performance Dashboard Reports

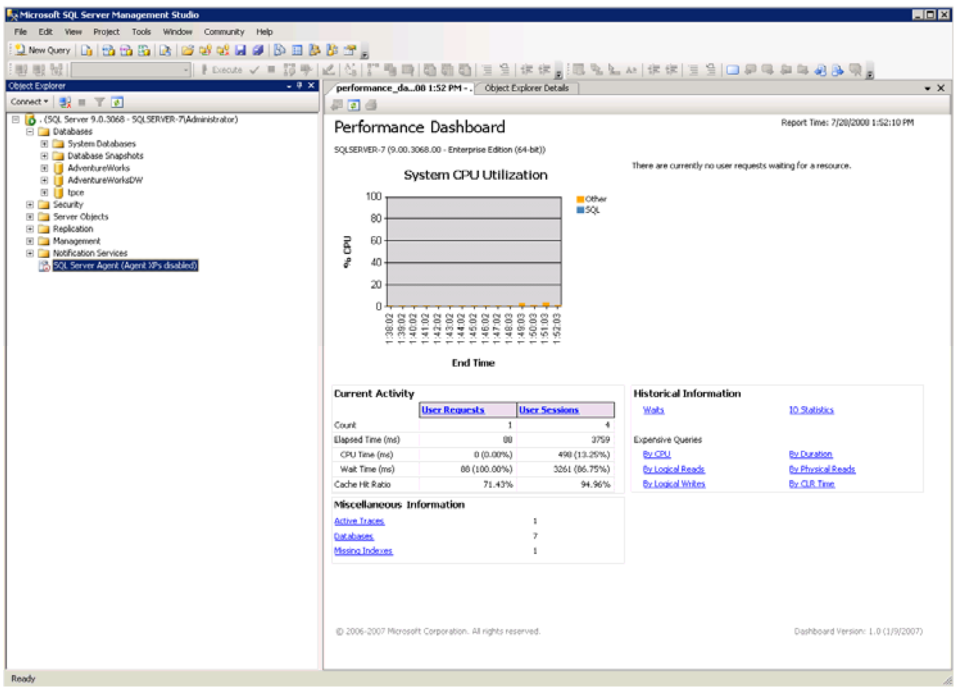
SQL Server 2005 Performance Dashboard Reports contains Reporting Services report files that allow quick identification of bottlenecks within a SQL Server environment. These customized reports are created through the use of SQL Server's DMV, which means that the DBA has no need to perform additional tracing or captures. The reports' drill-down capabilities provide an easy-to-navigate environment that allows DBAs to quickly identify possible performance issues within their SQL environments (see Figure 10).

The reports can help resolve common performance problems such as:

- CPU bottlenecks and queries consuming the most CPU
- I/O bottlenecks and queries performing the most I/O
- Index recommendations generated by the query optimizer (missing indexes)
- Blocking
- Latch contention

Keep in mind that SQL Server 2005 Performance Dashboard Reports does not store a history of performance over time. SQL Server 2005 Performance Dashboard Reports requires SQL Server 2003 Service Pack 2 or later and is available online.

Figure 10. SQL Server 2005 Performance Dashboard Reports Main Screen



Using Performance Monitoring Counters

Best practice is to establish a baseline for your SQL Server environment using performance monitoring tools and then monitor and analyze the environment on an ongoing basis.

Table 4 describes the key PerfMon counters to use when monitoring and analyzing disk-related interactions between the SQL Server hosts and the storage system.

Table 4. Key PerfMon Counters for Host Storage System Analysis

Performance Monitor Counter	Microsoft Description	Hitachi Data Systems Notes
Disk Reads/sec **Disk Writes/sec**	Number of I/Os per second (IOPs) being issued against a particular disk or volume. This number varies based on the size of I/Os issued. Consult the hardware vendor for an estimation of the number of I/Os per second support per spindle on their particular hardware.	Total number of read or write operations taking place per second for the selected logical or physical disk. This field indicates the quantity of I/O, not the size of the I/O.
Average Disk sec/Read **Average Disk sec/Write**	Measure of disk latency. Lower values are better but this can vary and is dependent on the size and nature of the I/Os being issued. Numbers also vary across different storage configurations (cache size/utilization can impact this greatly). On well-tuned I/O subsystems, ideal values would be: 1–5 ms for Log (ideally 1 ms on arrays with cache) 4–20 ms for Data on OLTP systems (ideally 10 ms or less) 30 ms or less on DSS (decision support system) type. Latencies here can vary significantly depending on the number of simultaneous queries being issued against the system. Sustained values of more than this when the total throughput is less than expected should be investigated. Consider these in combination with what is normal for your particular system. Make sure to monitor disk latencies for trend analysis. The number of I/Os and latency specific to SQL Server data files can be found by using the `sys.dm_io_virtual_file_stats` dynamic management view in SQL Server 2005.	Average response time in milliseconds for the selected logical or physical disk. This average can be affected by I/O size, RAID configuration and other factors in the data path.
Average Disk Bytes/Read **Average Disk Bytes/Write**	Size of I/Os being issued. Impacts disk latency. Large I/O sizes may result in slightly higher latency. This will tell you the average size of the I/Os SQL is issuing to fill query requests.	Average size in bytes of the I/O being issued.

Performance Monitor Counter	Microsoft Description	Hitachi Data Systems Notes
Average Disk Queue Length	Average number of outstanding I/O requests. The general rule of thumb is <=2 per spindle but this may be hard to measure due to storage virtualization, differences in RAID level between configurations, and so on. Focus on higher than average disk queue length in combination with higher than average disk latencies. This combination could indicate that the storage array cache is being over utilized or spindle sharing with other applications is impacting performance.	A high average disk queue length in combination with a high average disk/sec read (response time) might indicate that the HBA queue depth is set incorrectly for your environment. The controller on the storage system might be able to further optimize physical I/O when using an increased queue depth.
Disk Read Bytes/sec **Disk Write Bytes/sec**	Measure of the total bandwidth for a particular disk or LU.	Total number of bytes either read or written per second for the selected logical or physical disk.

Table 5 describes the key counters to use when monitoring and analyzing SQL Server-related operations.

Table 5. Key PerfMon Counters for SQL Server-related Operations

Category	Performance Monitor Counter	Microsoft Description	Hitachi Data Systems Notes
Buffer Manager	Buffer cache hit ratio	Percentage of pages found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups over the last few thousand page accesses. After a long period of time, the ratio moves very little. Because reading from the cache is much less expensive than reading from disk, you want this ratio to be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory available to SQL Server.	A high number indicates that the system is mainly utilizing cached data for its operations rather than going to the physical disk for data.
	Page life expectancy	Number of seconds a page will stay in the buffer pool without references.	As the server requests more memory for other processes, this value decreases.
	Page reads/sec	Number of physical database page reads that are issued per second. This statistic displays the total number of physical page reads across all databases. Because physical I/O is expensive, you may be able to minimize the cost, either by using a larger data cache, intelligent indexes and more efficient queries, or by changing the database design.	This counter is for all databases, including `tempdb`.
	Page writes/sec	Number of physical database page writes issued per second.	Total number of pages written, including split pages.

Category	Performance Monitor Counter	Microsoft Description	Hitachi Data Systems Notes
	Read ahead pages/sec	Number of pages read per second in anticipation of use.	The SQL Server engine looks at the type of query being performed and makes decisions as to what pages to read ahead of time. This can lead to a deceivingly small buffer hit rate.
Databases	Percent log used	Percentage of space in the log that is in use.	High log space utilization rates indicate a need for more storage resources for logs, that logs need to be backed up, or that they are not being truncated.
	Transactions/sec	Number of transactions started for the database per second.	The transactions per second and the amount of pages being read or written indicate the actual workload on SQL Server. OLTP is characterized as more reads than writes. Transactions per second and batch requests per second are an indication of server usage.
General Statistics	Processes blocked	Number of currently blocked processes.	This counter monitors processes waiting for pages that are locked by other processes. The lock might be a page, table or database lock preventing a process from accessing a page. A high number of blocked processes does not necessarily indicate I/O fault.
Latches	Average latch wait time (ms)	Average latch wait time (in milliseconds) for latch requests that had to wait.	This counter indicates the average number of milliseconds the latches waited, regardless of the reasons for the delays.
	Latch waits/sec	Number of latch requests that could not be granted immediately.	When transactions and batch requests are high and latch waits per second are low, the server is operating efficiently.
	Total latch wait time (ms)	Total latch wait time (in milliseconds) for latch requests in the last second.	
Locks	Average wait time (ms)	Average amount of wait time (in milliseconds) for each lock request that resulted in a wait.	High average wait time might be an indication of a page, table or database lock.
	Lock wait time (ms)	Total wait time (in milliseconds) for locks in the last second.	
Memory Manager	Memory grants pending	Total number of processes waiting for a workspace memory grant.	Even if a process has all the pages it needs, it might still be waiting for a workspace to complete a process. A high number can be an indication that many concurrent queries need workspace memory grants at the same time. Some examples of queries supported by workspace grants are sorts, hash joints and aggregations.

Category	*Performance Monitor Counter*	*Microsoft Description*	*Hitachi Data Systems Notes*
SQL Statistics	Batch requests/sec	Number of Transact-SQL command batches received per second. This statistic is affected by all constraints (such as I/O, number of users, cache size, complexity of requests and so on). High amount of batch requests means good throughput.	Batch requests and transactions per second indicate how busy the server is.
	SQL Compilations/sec	Number of SQL compilations per second. Indicates the number of times the compile code path is entered. Includes compiles caused by statement-level recompilations in SQL Server 2005. After SQL Server user activity is stable, this value reaches a steady state.	Compiles and recompiles are caused by changes to the environment. If these counters are high, it is an indication that something is causing the system's environment to change, like un-parameterized SQL statements, memory pressure on the processor's cache and more. Performing a `dbcc freeproccache` causes the buffers to flush and all processes to recompile and rerun.
	SQL Re-compilations/sec	Number of statement recompiles per second. Counts the number of times statement recompiles are triggered. Generally, you want the recompiles to be low. In SQL Server 2005, recompilations are statement-scoped instead of batch-scoped recompilations as in Microsoft SQL Server 2000. Therefore, direct comparison of values of this counter between SQL Server 2005 and earlier versions is not possible.	
Wait Statistics	Average wait time (ms)/Lock waits	Average time for processes waiting on a lock.	
	Average wait time (ms)/Log write waits	Average time for log buffer to be written.	
	Average wait time (ms)/Network I/O waits	Average time for wait on network I/O.	
	Average wait time (ms)/Page I/O latch waits	Average time for page I/O latches.	
	Average wait time (ms)/Page latch waits	Average time for page latches, not including I/O latches.	

Use the Hitachi Performance Monitor feature counters described in Table 6 when monitoring or analyzing a SQL Server environment. Keep in mind that the performance of the storage system is monitored from the controller level down to each of its main sections (port, RAID groups, logical unit, cache, processor, drive, drive operation and back end). When both controllers are being used, select the following counters from both controllers.

Table 6. Hitachi Performance Monitor Feature Counters

Performance Monitor Counter	Description	Normal Value
Port IO Rate (IOPS)	Total number of commands (reads and writes) per second on the selected ports. Port read/write I/O rates can be individually monitored as described using other port counters.	Varies
Port Read Rate **Port Write Rate (IOPS)**	Number of read/write commands per second on the selected ports.	Varies
Port Read Hit **Port Write Hit (%)**	Indication of cache-hitting within the received read/write command on the selected ports. The read/write hit counters can also be monitored at a RAID group or logical unit level.	Read – Varies[1] Write – 100%
Port Transfer Rate (MB/sec)	As with the port I/O rates, port read/write transfer rates can also be individually monitored for a given port.	Varies
RAID Group IO Rate (IOPS)	Total number of commands (reads and writes) per second on the selected RAID group. As with the port I/O rates, RAID group read/write I/O rates can also be individually monitored for a given RAID group.	Varies
RAID Group Read Rate **RAID Group Write Rate (IOPS)**	Number of commands (reads or writes) per second on the selected RAID groups.	Varies
Cache Write Pending Rate (%)	Indication of cache being used to buffer writes on the selected controller.	1% to 25%[2]
Processor Usage (%)	Processor utilization rate on the selected controller.	1% to 50%[3]
Drive IO Rate (IOPS)	Total number of commands (reads and writes) per second on the selected physical drives.	Varies

[1] Port Read Hit counter is highly dependent on the type of workload and overall user load the database has.

[2] Above 25% might be an indication of insufficient spindles for one or more RAID groups or that one or more RAID groups were created on an incorrect RAID type (for example, RAID-5 instead of RAID-1+0).

[3] If one controller reaches a high utilization rate while the other is underutilized, the Hitachi Dynamic Controller Load Balancing Controller architecture automatically balances workload between controllers.

Reference Configuration Examples

The recommendations provided in this document are generally applicable to databases ranging in size from 50GB to 480GB. For each of the reference configurations described in the following section, up to four database (`.mdf`) files are placed on a given RAID group, depending on the environment. It is important to note that while placed in the same RAID group, these files are contained within dedicated LUs.

The optimal storage configuration is a tradeoff based on cost, availability and performance. Disk drive type and capacity, RAID level, amount of cache and overall storage configuration all contribute to the ability of the architecture to meet business requirements and the constraints of a budget.

For the large database configurations presented in this paper (250GB and 480GB databases) each database LU is deployed on a separate RAID-1+0 (2D+2D or 4D+4D) RAID group for both performance and availability. The transaction logs for these databases are hosted on dedicated RAID-1+0 (2D+2D) RAID groups. On small database configurations (50GB and 100GB databases), transaction logs are hosted on dedicated RAID-1

(1D+1D) RAID groups. In regard to performance, the number of spindles and the RAID level determine the physical IOPS capability of the RAID group for disk drives of a specific type. The disks used in the configurations are all 146GB 15,000RPM SAS drives.

The following sections describe common components that are used in the reference configurations described in this section along with storage assignment details and illustrations of the physical disk allocations for each configuration. The reference configurations presented in this section are built on a shared environment in which other SQL Server databases are hosted, and one in which a Microsoft Exchange Server environment is hosted on the storage system.

Hardware Components

- Hitachi Adaptable Modular Storage 2100 storage system:
  - 8GB cache
  - 4 Fibre Channel 4Gb ports
  - 146GB 15,000RPM SAS physical disks
- Hitachi Adaptable Modular Storage 2300 storage system:
  - 16GB cache
  - 8 Fibre Channel 4Gb ports
  - 146GB 15,000RPM SAS physical disks
- Server hardware:
  - Quad dual core 64-bit server
  - 16GB memory
  - 2 Emulex LPe11000 4Gb HBAs

Software and Service Components

- Microsoft Windows Server 2008 Enterprise x64 Edition
- Microsoft .NET Framework 2.0, Service Pack 1, or version 3.0
- Microsoft SQL Server 2005 Enterprise Edition, Service Pack 2
- Microsoft Management Console (MMC) 3.0
- Emulex Storport Miniport Driver version 7-2.01A4

For more information, see Microsoft SQL Server 2005 system requirements.

Testing I/O on a Storage System

As documented in the *Microsoft Predeployment Best Practices* paper, it is important to understand the existing or expected SQL Server I/O workload and to evaluate the specific storage environment before deploying SQL Server-based applications. It is also necessary to identify the workloads presented by other applications, such as Microsoft Exchange or SharePoint if those applications share storage resources with SQL Server.

I/O throughput on the 2000 family increases dramatically as cache read hits increase, which highlights the importance of database application design. Application designs that can take advantage of server and storage caching generally achieve much higher transaction throughput compared to those that require frequent physical disk access. For many OLTP applications, locality of reference provides a natural cache-friendly benefit. For example, when performance issues arise during frequent access of the most popular products in an order entry application, examine the read cache hit percentage and cache write pending counters on the storage system for troubleshooting clues. Additional performance measurements that might be useful are shown in the Performance Monitoring section.

While the reference architectures in this paper feature the Hitachi Adaptable Modular Storage 2100 and 2300 systems, its recommendations apply to any of the models in the 2000 family because none of the test configurations generate enough I/O to stress any of these storage systems. Any member of the Hitachi Adaptable Modular Storage 2000 family is a good choice for SQL Server implementations, either as dedicated storage systems or as shared application storage platforms.

Reference Architectures

The following figures and tables describe the relationship between SQL Server structures, RAID groups, LUs, ports and controllers. For some of the configurations, the assignment numbers alternate (even and odd) to allow easy identification of the controller on which the LU and RAID group reside. In general, the goal is to balance the load across the ports and controllers. In addition, the port assignment for the alternate path for each LUN assumes the use of Hitachi Dynamic Load Manager for automatic failover in the event of a failure along the primary path.

Database Layout Recommendations

Database files (`.mdf` and `.ndf`) and `tempdb` are located on separate LUs on the same RAID groups. Log files (`.ldf`) are on RAID groups separated from database and `tempdb` I/O both to improve performance and to improve availability. Mixing small block random I/O and large block sequential I/O can circumvent some of the advanced performance accelerators inherent in the 2000 family microcode. For that reason, use caution when creating additional LUs on RAID groups where LUs hosting database files are located.

250GB 2100 Reference Configuration

Table 7 and Table 8 describe the reference architecture for a 250GB capacity database environment. Table 7 describes the RAID groups created to sustain the capacity required for the environment.

Table 7. RAID Groups for the 250GB Database Capacity Environment

RAID Group Number	RAID Level	Configuration	Number of Hard Disk Drives
0	1+0	2+2	4
1	1	1+1	2

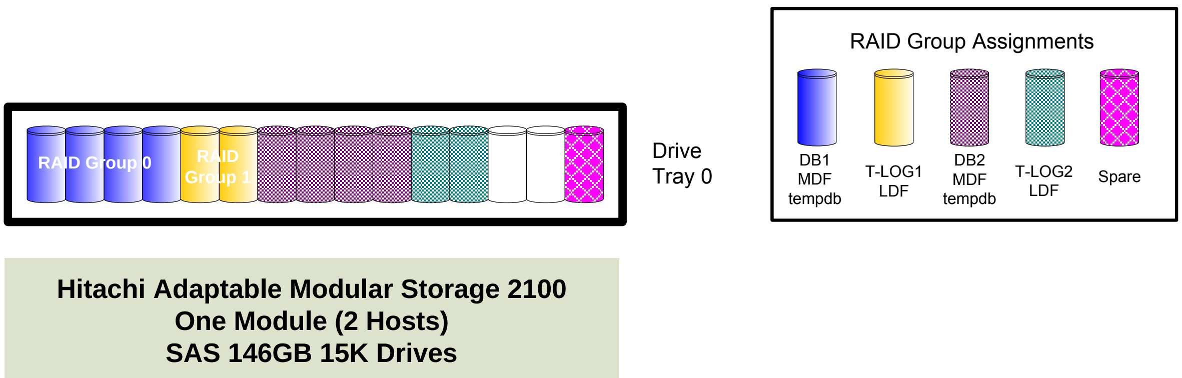
Table 8 describes the capacity and port assignment of each LU.

Table 8. LU Sizes for the 250GB Database Capacity Environment

LU Assignment	LUN	RAID Group Number	LU Size (GB)	Port Assignment	
				Primary Path	Secondary Path
Database	0	0	250	0A	1A
Transaction log	1	1	62	1A	0A
`tempdb`	2	0	25	0A	1A

Figure 11 illustrates the RAID group layout on the storage system for the 250GB reference architecture. Note that the shaded RAID groups are not part of this reference architecture.

Figure 11. Reference Configuration for the 250GB Database Capacity Environment



480GB 2300 Reference Configuration

Tables 9 and 10 describe the reference architecture for a 480GB capacity database environment. Table 9 describes the RAID groups created to sustain the capacity required for the environment.

Table 9. RAID Groups for the 480GB Database Capacity Environment

RAID Group Number	RAID Level	Configuration	Number of Hard Disk Drives
2	1+0	4+4	8
3	1+0	2+2	4

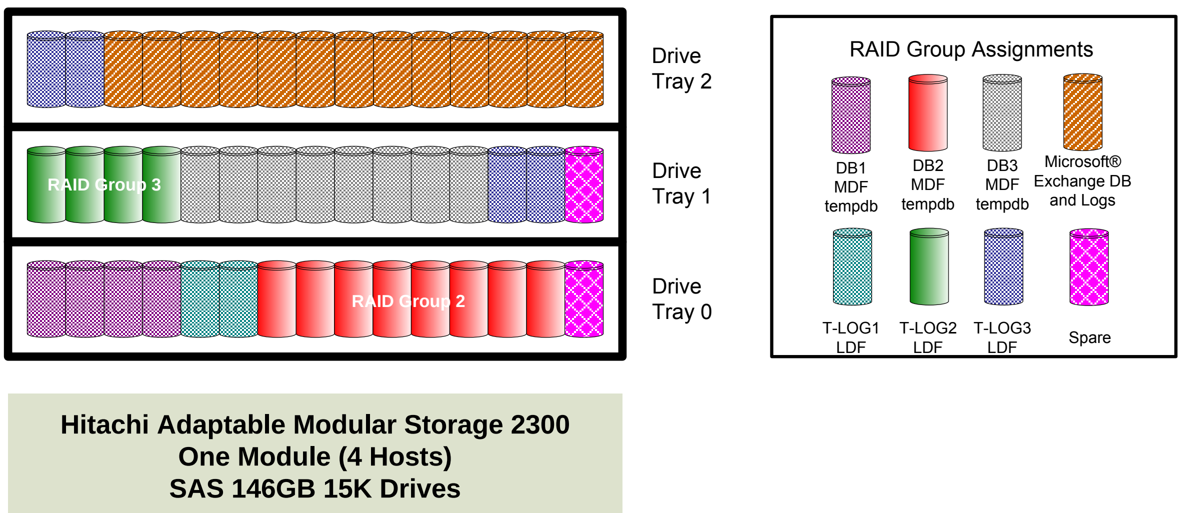
Table 10 describes the capacity and port assignment of each LU.

Table 10. LU Sizes for the 480GB Database Capacity Environment

LU Assignment	LUN	RAID Group Number	LU Size (GB)	Port Assignment	
				Primary Path	Secondary Path
Database	0	2	482	0A	1A
Transaction log	1	3	125	1A	0A
`tempdb`	2	2	50	0A	1A

Figure 12 illustrates the RAID group layout on the storage system for the 480GB reference architecture. Note that the shaded RAID groups are not part of this reference architecture.

Figure 12. Reference Configuration for the 480GB Database Capacity Environment



250GB 2300 Reference Configuration

Tables 11 and 12 describe the reference architecture for a 250GB capacity database environment with three independent 50GB databases. Table 11 describes the RAID groups created to sustain the capacity required for the environment.

Table 11. RAID Groups for Three Databases with 50GB Capacity Each

RAID Group Number	RAID Level	Configuration	Number of Hard Disk Drives
0	1+0	2+2	4
1	1	1+1	2

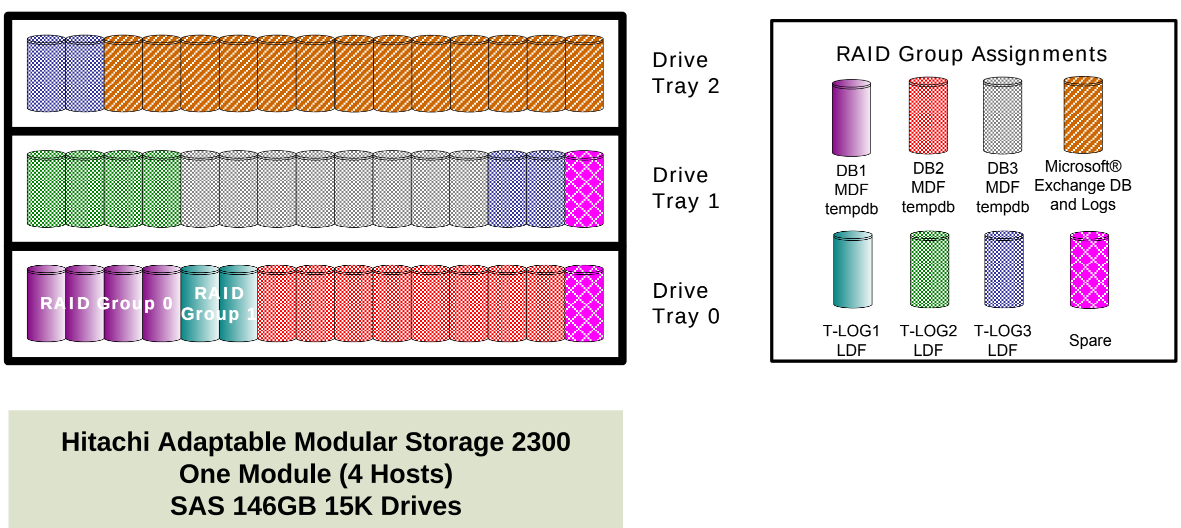
Table 12 describes the capacity and port assignment of each LU.

Table 12. LU Sizes for Three Databases with 50GB Capacity Each

LU Assignment	LUN	RAID Group Number	LU Size (GB)	Port Assignment	
				Primary Path	Secondary Path
Database	0	0	52	0B	1B
Transaction log	1	1	12	1B	0B
tempdb	2	0	5	0B	1B
Database	3	0	52	0B	1B
Transaction log	4	1	12	1B	0B
tempdb	5	0	5	0B	1B
Database	6	0	52	0B	1B
Transaction log	7	1	12	1B	0B
tempdb	8	0	5	0B	1B

Figure 13 illustrates the RAID group layout on the storage system for the 250GB reference architecture with three independent 50GB databases. Note that the shaded RAID groups are not part of this reference architecture.

Figure 13. Reference Configuration for Three Databases with 50GB Capacity Each



400GB 2300 Reference Configuration

Tables 13 and 14 describe the reference architecture for a 400GB capacity database environment with four independent 100GB databases. Table 13 describes the RAID groups created to sustain the capacity required for the environment.

Table 13. RAID Groups for Four Databases with 100GB Capacity Each

RAID Group Number	RAID Level	Configuration	Number of Hard Disk Drives
4	1+0	4+4	8
5	1+0	2+2	4

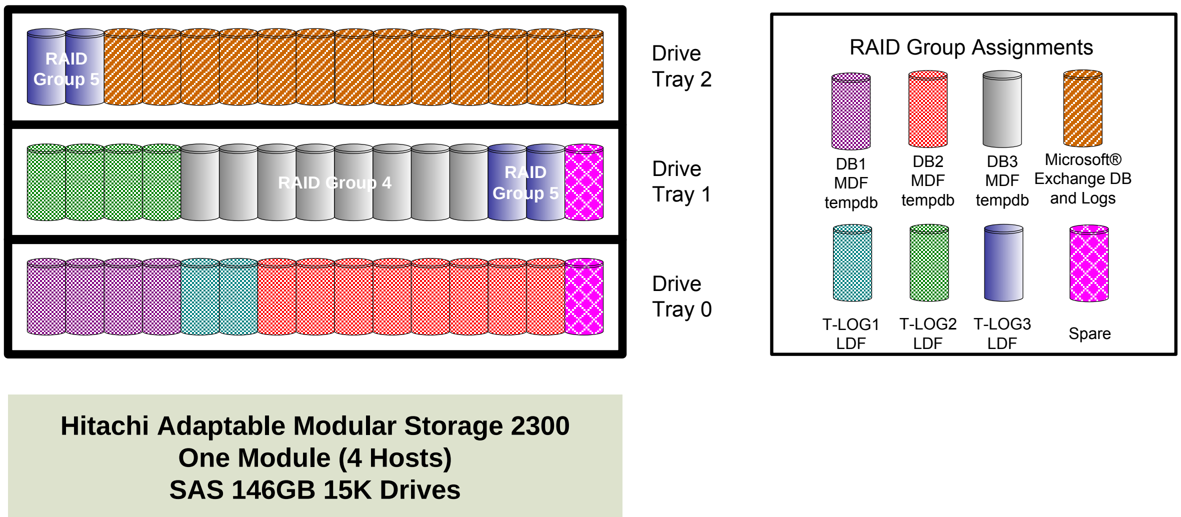
Table 14 describes the capacity and port assignment of each LU.

Table 14. LU Sizes for Four Databases with 100GB Capacity Each

LU Assignment	LUN	RAID Group Number	LU Size (GB)	Port Assignment	
				Primary Path	Secondary Path
Database	0	4	105	0C	1C
Transaction log	1	5	25	1C	0C
`tempdb`	2	4	10	0C	1C
Database	3	4	105	0C	1C
Transaction log	4	5	25	1C	0C
`tempdb`	5	4	10	0C	1C
Database	6	4	105	0C	1C
Transaction log	7	5	25	1C	0C
`tempdb`	8	4	10	0C	1C
Database	9	4	105	0C	1C
Transaction log	10	5	25	1C	0C
`tempdb`	11	4	10	0C	1C

Figure 14 illustrates the RAID group layout on the storage system for the 400GB reference architecture with four independent 100GB databases. Note that the shaded RAID groups are not part of this reference architecture.

Figure 14. Reference Configuration for Four Databases with 100GB Capacity Each



Conclusion

With its new Hitachi Dynamic Load Balancing Controller architecture, symmetrical active-active controllers and simplified Hitachi Storage Navigator Modular 2, the Hitachi Adaptable Modular Storage 2000 family makes deploying and managing storage for Microsoft SQL Server fast and easy. As either a dedicated SQL Server platform or a shared storage environment, the 2000 family is the right choice to meet your demanding SQL Server application needs.

For more information on the exciting new family of Hitachi Adaptable Modular Storage systems or to learn more about how Hitachi solutions can help you save time and money, contact your local Hitachi TrueNorth Channel Partner or Hitachi Data Systems sales team or visit us on the Web to see why Hitachi storage is the right choice for Microsoft SQL Server.

Hitachi Data Systems Corporation

Corporate Headquarters 750 Central Expressway, Santa Clara, California 95050-2627 USA
Contact Information: + 1 408 970 1000 www.hds.com / info@hds.com

Asia Pacific and Americas 750 Central Expressway, Santa Clara, California 95050-2627 USA
Contact Information: + 1 408 970 1000 www.hds.com / info@hds.com

Europe Headquarters Sefton Park, Stoke Poges, Buckinghamshire SL2 4HD United Kingdom
Contact Information: + 44 (0) 1753 618000 www.hds.com / info.emea@hds.com