

**EMC Solutions for Microsoft SQL Server 2005**  
**EMC<sup>®</sup> Celerra<sup>®</sup> NS Series iSCSI**  
**Applied Best Practices Guide**

**EMC NAS Product Validation**  
*Corporate Headquarters*  
Hopkinton, MA 01748-9103  
1-508-435-1000  
[www.EMC.com](http://www.EMC.com)

Copyright © 2007 EMC Corporation. All rights reserved.

Published October, 2007

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

**Microsoft SQL Server 2005 EMC Celerra NS Series iSCSI Applied Best Practices Guide**

**P/N H2370.3**

	About this Document .....	5
Chapter 1	SQL Server Best Practices .....	9
	General performance .....	10
	Recommendation #1: Plan for storage performance, not for capacity .....	10
	Recommendation #2: Use clusters for high availability .....	10
	Recommendation #3: Make the SQL Server part of an Active Directory domain. ....	10
	Recommendation #4: Enable SQL Server to keep pages in memory. ....	10
	Recommendation #5: Enable Windows fast file initialization.....	11
	Recommendation #6: Do not allow database and log files to share physical spindles .....	11
	Recommendation #7: Set database file sizes and autogrow increments .....	11
	Recommendation #8: Plan database filegroups based on the workload .....	12
	Recommendation #9: Plan the location, layout, and size of the tempdb.....	12
	Recommendation #10: Use defaults for processors and memory .....	13
	Recommendation #11: Use failover - aware applications.....	13
	Recommendation #12: Disable hyperthreading on Microsoft SQL Server 2005 servers .....	13
	Backup and restore.....	14
	Recommendation #13: For point-in-time recovery use database log backups.....	14
	Recommendation #14: When possible, schedule backups for minimal disruption.....	14
	Recommendation #15: Do a full backup when you change the database recovery model ...	14
	Data protection: Database mirroring .....	15
	Recommendation #16: If using MSCS at the principal, do not use a witness server.....	15
	Recommendation #17: Plan for high I/O levels at the mirror site .....	15
	Recommendation #18: Ensure that inter-database consistency is NOT needed .....	15
	Recommendation #19: Use other methods to sync some SQL Server objects.....	15
	Recommendation #20: Consult additional resources for additional information.....	15
Chapter 2	Microsoft Windows Server 2003 Best Practices .....	17
	Recommendation #21: Only use Microsoft approved hardware.....	18
	Recommendation #22: Use the latest verified NIC driver .....	18
	Recommendation #23: When using MSCS, reboot the passive node occasionally .....	18
	Recommendation #24: Use a dedicated VLAN for cluster heartbeat connectivity.....	18
Chapter 3	Network Best Practices .....	19
	Recommendation #25: Use Gigabit Ethernet switches with VLAN capabilities.....	20
	Recommendation #26: Use CAT6 cables for GbE connectivity.....	20

	Recommendation #27: Manually set network speed and duplexing.....	20
Chapter 4	Storage Systems Best Practices .....	21
	Recommendation #28: Plan storage layouts for performance, not for capacity.....	22
	Recommendation #29: Use DISKPART to align the LUNs for best performance .....	22
	Recommendation #30: Set the NTFS allocation unit to 64 KB. ....	24
	Recommendation #31: Do not exceed 80% utilization of LUNs.....	24
	Recommendation #32: Use Gigabit Ethernet for iSCSI connections.....	24
	Recommendation #33: Create persistent iSCSI target connections and bindings.....	25
	Recommendation #34: Verify that your TcpWindowSize setting is correct.....	25
	Recommendation #35: Increase your iSCSI time-out value. ....	25
	Recommendation #36: Use MC/S for iSCSI high performance and high availability.....	26
	Recommendation #37: Use the most recently available Celerra software. ....	26
	Recommendation #38: Use dedicated Celerra file systems for iSCSI LUNs.....	26
	Recommendation #39: Disable DNS on the storage network .....	26
	Recommendation #40: Use Celerra storage pools for volume management.....	26
	Recommendation #41: Configure the system for high availability.....	27
	Recommendation #42: Fail over Data Movers before rebooting.....	27
Appendix A	Performance Monitoring and Tuning .....	29
	Overview .....	30
	Windows performance monitoring .....	30
Appendix B	RAID Group Planning .....	33
	Overview .....	34
	RAID level attributes .....	34
	Estimating required performance.....	36
	Calculating disk spindle requirements .....	38
	Summary .....	40
Appendix C	File Group Planning .....	41
	Overview .....	42
	TempDB .....	42
	User databases.....	42
	Log files .....	42

This document summarizes a series of Best Practices which were discovered, validated or otherwise encountered during the validation of a solution for using Microsoft SQL Server with Celerra iSCSI.

### Purpose

Information in this document can be used as the basis for a solution build, white paper, best practices document, or training. Information in this document can also be used by other EMC organizations (for example, the technical services or sales organization) as the basis for producing documentation for a technical services or sales kit.

### Audience

The intended audience includes IT administrators, database administrators, data architects, and system engineers who have an interest in implementing Microsoft SQL Server 2005 using EMC Celerra systems.

It is assumed that the reader has a general knowledge of Microsoft SQL Server and EMC Celerra features and terminologies.

### Scope

While designing a Microsoft SQL solution, it is important to plan a solution that supports scalability while maintaining acceptable performance, high availability, and an efficient mechanism for disaster recovery. This document can be applied in many situations, and is meant to be a resource guide of recommendations for Microsoft SQL Server 2005 storage configuration on EMC Celerra through iSCSI.

---

**Note:** The recommendations in this document are mostly derived from running a TPCC workload. A TPCC workload is meant to be representative of an OLTP workload. However no two workloads are the same. The only way to be sure of the effect that a change in an environment will have is to test that change in that environment. Microsoft SQL Server is an environment, not an application, so the databases that it hosts can vary widely. Therefore, it is possible that a recommendation may help in one environment while it may have little effect or even a negative effect, on a different environment. Using “rules of thumb” or best practices is not a valid substitute for proper planning, design, and architecture.

---

## Related documents

The following documents provide additional, relevant information. Access to these documents is based on your login credentials. If you do not have access the content listed below, contact your EMC representative:

- ◆ *EMC Solutions for Microsoft SQL Server 2005 EMC Celerra NS20 over iSCSI - Reference Architecture*
- ◆ *EMC Solutions for Microsoft SQL Server 2005 EMC Celerra NS20 over iSCSI - Validation Test Report*
- ◆ *Celerra Network Server 5.5 Best Practices for Performance - Best Practices Planning white paper*
- ◆ *Best Practices for Celerra iSCSI: Considerations to Understand When Deploying Celerra iSCSI within Your Environment*
- ◆ *Microsoft iSCSI Software Initiator User's Guide*
- ◆ *Microsoft Storage Technologies: Deploying iSCSI SANs*
- ◆ *Microsoft SQL Server 2000 Operations Guide: Capacity and Storage Management*

## Terminology

This section defines the terms used in this document.

**Table 1** Terminology

Term	Definition
Automatic Volume Management (AVM)	A feature of the Celerra Network Server that creates and manages volumes automatically, without manual volume management by an administrator. AVM organizes volumes into pools of storage that can be allocated to file systems
Data Mover	A Celerra Network Server cabinet component running the data access in real time (DART) operating system, which retrieves files from a storage device and makes the files available to a network client.
Disk volume	On Celerra systems, a physical storage unit exported from the storage array. All other volume types are created from disk volumes.
Internet SCSI (iSCSI)	A protocol for sending SCSI packets over TCP/IP networks.
iSCSI initiator	An iSCSI endpoint, identified by a unique iSCSI-recognized name, which begins an iSCSI session by issuing a command to the other endpoint (the iSCSI target).
iSCSI target	An iSCSI endpoint, identified by a unique iSCSI-recognized name, which executes commands issued by the iSCSI initiator.
Logical unit number (LUN)	The identifying numbers of a SCSI or iSCSI object that processes SCSI commands. The LUN is the last part of the SCSI address for a SCSI object. The LUN is an ID for the logical unit, but the term is sometimes used to refer to the logical unit itself.
Microsoft Cluster Services (MSCS)	A cluster is a group of computers called nodes that function as a single computer/system to provide high availability and high fault tolerance for

Term	Definition
	applications and services. If one member node of the cluster is unavailable, the other computers carry the load so that applications or services are always (with a small interruption) available.
Multiple Connections per Session (MCS)	Microsoft iSCSI Initiator version 2.x provides MCS, which allows multiple TCP/IP connections between the initiator (server) and target (storage array) during the same iSCSI session, either on the same or a different physical link. This allows load balancing and failover among multiple network interface cards (NICs).
MPIO	Microsoft iSCSI Initiator version 2.x provides MPIO multipathing support for iSCSI, which allows the initiator to log in to multiple sessions on the same target, providing load balancing for storage devices. Multipathing is a high-availability function, since it provides multiple paths from a host to an external storage device.
RAID	Redundant array of inexpensive disks. A method for storing information where the data is stored on multiple disk drives to increase performance and storage capacities and to provide redundancy and fault tolerance
RAID 1	RAID method that provides data integrity by mirroring (copying) data onto another disk. This RAID type provides the greatest assurance of data integrity at the greatest cost in disk space.
RAID 1 with striping	The volume creation method described in the EMC Solutions for Microsoft SQL Server 2005 EMC Celerra NS20 over iSCSI Reference Architecture that allows the creation of a stripe of mirrors. This configuration is functionally similar to a RAID 1+0 configurations.
RAID 5	Data is striped across disks in large stripes. Parity information is stored so data can be reconstructed if needed. One disk can fail without data loss. Performance is good for reads but slower for writes.
RAID group	The CLARiiON <sup>®</sup> storage system term for a Celerra disk group. In a CLARiiON storage system, a RAID group is a set of physical disks with a RAID type on which one or more LUNs are bound. Each RAID group supports only the RAID type of the first LUN bound on it; any other LUNs bound on it have that same RAID type. LUNs are distributed equally across all the disks in the RAID group.
SP	Storage processor on a CLARiiON or an integrated Celerra storage system. On a CLARiiON storage system, a circuit board with memory modules and control logic that manages the storage-system I/O between the host's Fibre Channel adapter and the disk modules.
SP A	Storage processor A. A generic term for the first storage processor in a CLARiiON storage system.
SP B	Storage processor B. A generic term for the second storage processor in a CLARiiON storage system.
System database	A database that is installed as part of the installation of Microsoft SQL Server. The system databases include master, model, msdb, tempdb, and others.
User database	A non-system database that is put on the server after the installation of Microsoft SQL Server. Examples include an OLTP application database or data warehouse.
Volume Shadow Copy Service (VSS)	A Windows Service that provides an infrastructure that enables third-party storage management programs, business programs, and hardware providers to create and manage consistent point-in-time copies of data, called shadow copies.



## Chapter 1 SQL Server Best Practices

This section details recommendations for the configuration of Microsoft SQL Server 2005 on Windows Server 2003. This chapter presents these topics:

General performance .....	10
Recommendation #1: Plan for storage performance, not for capacity .....	10
Recommendation #2: Use clusters for high availability .....	10
Recommendation #3: Make the SQL Server part of an Active Directory domain. ....	10
Recommendation #4: Enable SQL Server to keep pages in memory. ....	10
Recommendation #5: Enable Windows fast file initialization.....	11
Recommendation #6: Do not allow database and log files to share physical spindles .....	11
Recommendation #7: Set database file sizes and autogrow increments .....	11
Recommendation #8: Plan database filegroups based on the workload .....	12
Recommendation #9: Plan the location, layout, and size of the tempdb.....	12
Recommendation #10: Use defaults for processors and memory .....	13
Recommendation #11: Use failover - aware applications.....	13
Recommendation #12: Disable hyperthreading on Microsoft SQL Server 2005 servers .....	13
Backup and restore .....	14
Recommendation #13: For point-in-time recovery use database log backups.....	14
Recommendation #14: When possible, schedule backups for minimal disruption.....	14
Recommendation #15: Do a full backup when you change the database recovery model ...	14
Data protection: Database mirroring .....	15
Recommendation #16: If using MSCS at the principal, do not use a witness server. ....	15
Recommendation #17: Plan for high I/O levels at the mirror site .....	15
Recommendation #18: Ensure that inter-database consistency is NOT needed .....	15
Recommendation #19: Use other methods to sync some SQL Server objects.....	15
Recommendation #20: Consult additional resources for additional information.....	15

## General performance

### Recommendation #1: Plan for storage performance, not for capacity

The most common error made while planning the storage for Microsoft SQL Server is designing for storage capacity and not for performance or I/Os per Second (IPOS). With advances in disk technology, the increase in storage capacity of a disk drive has outpaced the increase in IOPS by almost 1,000:1. With this effect it is rare to find a system that, when planned for performance, does not meet the storage capacity requirements for the workload. Hence, the IOPS capacity is the standard to be used while planning Microsoft SQL Server storage configurations. Only after considering the IOPS capacity of a configuration should the storage capacity (GB) be considered.

### Recommendation #2: Use clusters for high availability

A cluster is a collection of servers known as nodes that together provide a single, highly available system for hosting applications such as Microsoft SQL Server 2005. Microsoft Windows Server 2003 Enterprise Edition 64-bit R2 supports clusters of up to eight nodes. This solution was validated using two nodes.

Microsoft clusters provide a highly available environment that can protect against Microsoft SQL Server 2005 server failures of hardware, operating systems, device drivers, or applications. If one of the nodes in a cluster is unavailable as a result of a failure, another node immediately begins providing service.

### Recommendation #3: Make the SQL Server part of an Active Directory domain.

The primary recommended method for security and account management in SQL Server is through Active Directory domain user accounts, using integrated security. This allows for greater security, at multiple levels, and makes user management easier.

The SQL Server should not be a domain controller, except in certain unusual circumstances. The added overhead of being a DC is likely to have a negative impact on the SQL Server performance.

### Recommendation #4: Enable SQL Server to keep pages in memory.

Microsoft SQL Server dynamically allocates and deallocates memory based on the current state of the server, in an attempt to prevent memory pressure and swapping. However, if a process suddenly attempts to grab a substantial amount of memory, SQL Server may not be able to react quickly enough and the OS may swap some of SQL Server's memory to disk. Unfortunately, there is a good probability that the memory that was swapped to disk contains part of what SQL Server will soon be deallocating to decrease its memory use in response to the newly created memory pressure.

It is recommended that SQL Server be enabled to prevent its memory from being swapped. This is known as locking pages in memory. To do this, the account that the Microsoft SQL Server service is running under must be given the "Lock pages in memory" user right.

## Recommendation #5: Enable Windows fast file initialization

When Microsoft SQL Server creates or expands a file, the file must be initialized. Previous versions of SQL Server had only one option and that was to initialize the space by writing all “0” zeros to the space, which would cause a substantial performance impact if a file growth occurred. In Microsoft SQL Server 2005 there is now support for fast file initialization, which just sets a file end pointer, and then is complete. This operation is nearly instant and minimizes the impact of file growth on production systems. Fast file initialization is enabled at the OS level, by granting the user right “Perform volume maintenance tasks” to the account that Microsoft SQL Server service is running under. By default, this right is granted to administrators.

## Recommendation #6: Do not allow database and log files to share physical spindles

It is highly recommended to ensure that the database data files and log files do NOT share the same physical spindles. This helps to prevent the loss of data due to loss of multiple drives, and in many cases improves performance.

## Recommendation #7: Set database file sizes and autogrow increments

Microsoft SQL Server 2005 supports the ability to automatically grow both data and log files as they fill. However, this should not be misconstrued as a method of database sizing. It is a best practice to set the file sizes appropriately and grow them manually at times of minimal system use, on a planned basis. Autogrowth should only be used as a safety net to prevent the files from becoming full and making the database read-only, at times when unpredicted substantial growth occurs.

---

**Note:** When database files are expanded there is an impact to performance. This impact is minimized but not nullified through fast file initialization.

---

Additionally, the file autogrowth increments should be set such that the time it takes for the growth to occur is short enough to minimize its impact to performance, but large enough to prevent many small allocations that invite file fragmentation. An adequate increase in file size that prevents fragmentation usually impacts the performance of the database. Hence, it is recommended that the file sizes be changed during periods of lowered activity on the database.

Log files have an additional issue, as there are virtual log files within a physical log file, and a virtual log file cannot span file growth increments. Thus, if the log file were set to grow at 1 MB increments, then the virtual log file would not be able to exceed 1 MB either. This will have a performance impact as discussed above due to the impact of file expansion. This limit may also make certain transactions impossible to complete.

For all files, because of the impact to performance, it is recommended that an absolute growth increment (in MB or GB) be used instead of a percentage growth. It is also recommended that autoshrink should never be enabled. A file should not be shrunk, unless absolutely necessary, and only through a controlled manual action.

## Recommendation #8: Plan database filegroups based on the workload

SQL Server provides users with many options for organizing database tables and other structures on disk. The primary structure to control this behavior is a filegroup. Database structures are assigned to filegroups, which contain files on disk where that data can be stored. The placement of these data files is critical to the I/O performance of the database, and the recommendations for the best ways to set up filegroups vary based on the database workload. Please see Appendix B RAID Group Planning for a detailed discussion of how various database workloads impact your selection of a physical storage structure, and Appendix C File Group Planning for a discussion of how to plan your filegroups.

## Recommendation #9: Plan the location, layout, and size of the tempdb

By default the tempdb database is rather small and gets its characteristics from the model database. Each time the Microsoft SQL Server service starts, the tempdb is dropped and recreated with its initial parameters. Thus, if tempdb is initially 128 MB and during operations it autogrows to 4 GB, on restart it will be 128 MB again. Then it will have to go through the autogrow again, which will impact the performance of your database. To minimize this impact it is recommended that tempdb be sized appropriately for the environment. The easiest way to size the tempdb database is the following:

1. Start with a reasonable size tempdb for the size of databases that are in the same SQL Server instance. For example, a 1 GB tempdb database is a reasonable starting place for a sum total of instance databases between 10 GB and 100 GB, but not for 1 TB. A good starting place is to sum the total size of the databases in the instance and size tempdb between 1% and 10% of that size.
2. Set a valid autogrow increment that will allow the tempdb to grow without heavy fragmentation. The best thing here is to set the autogrow to between 10% to 20% of the tempdb initial size. Do not use a percentage for the growth parameter, calculate the MB growth that corresponds to the percentage and set that as the autogrowth size. You should also make sure that fast file initialization is enabled.
3. Periodically check the size and utilization of the tempdb database to see if it has grown significantly.
4. Reset the size of the tempdb database to something close to its size, before a shutdown. If our tempdb database from the example above had grown from 1 GB to 5 GB, then resetting it to start at 5 GB would be advantageous, unless the new size is obviously excessive. For example, if the sum total of user databases was 10 GB and tempdb was 15 GB, this would seem excessive. It is possible that an odd set of scenarios came together to cause uncharacteristic tempdb growth. If you suspect that this may be the case, then the starting size should be set to something smaller than the current size. If the tempdb repeatedly grows to larger than is initially considered reasonable, then it is possible that this is simply the size of tempdb that is needed for your workload. From here, a DBA could diagnose what is causing the excessive growth and then determine if it is valid, or if anything needs tuning.

It is recommended that tempdb be placed on its own spindles, where tempdb and user database activity cannot cause physical disk contention with each other. The number of spindles will be determined on a case-by-case basis using the same principles that are applied to designing storage for user databases.

## Recommendation #10: Use defaults for processors and memory

When Microsoft SQL Server is first installed, most of its tunable parameters are set to automatic and it is recommended that, on a server dedicated to SQL Server's use, these parameters should be left at their automatic defaults. The only time they should really be changed is if there are other workloads running on the same server, or if issues arise from the use of the defaults.

By default SQL Server will run at a standard priority and make all processors in the system available for use. Also, by default SQL Server will use as much memory as it needs until it notices that memory pressure is starting to build. If other processes start consuming memory, SQL Server will begin decreasing its memory footprint appropriately to decrease the possibility of swapping occurring.

## Recommendation #11: Use failover - aware applications

When a Microsoft SQL Server failover occurs, using MSCS clustering, Database Mirroring, or other technologies, all database connections are lost and any "in-flight" transactions are rolled back. To minimize data loss, it is recommended that all applications be failover-aware and have reconnect/retry logic. Thus, in case of a failover, the application will attempt to reconnect, and once it successfully reconnects, it will retry the transaction that was previously rolled back.

## Recommendation #12: Disable hyperthreading on Microsoft SQL Server 2005 servers

Intel hyperthreading technology allows multithreaded operating systems to view a single physical processor as if it were two logical processors. A processor that incorporates this technology shares CPU resources among multiple threads. In theory, this enables faster enterprise-server response times and provides additional CPU processing power to handle larger workloads. As a result, server performance will improve. However, testing has shown that hyperthreading can have a negative impact on many Microsoft SQL Server 2005-based processor loads. Unless it can be proven that hyperthreading helps the performance of a particular Microsoft SQL Server 2005 workload, it is recommended that hyperthreading be disabled.

Hyperthreading must be disabled at the hardware (BIOS setting) level, not through the application of processor affinity or other software means.

---

**Note:** For more information, refer to the Slava Oks's WebLog (<http://blogs.msdn.com/slavao/archive/2005/11/12/492119.aspx>) site.

---

## Backup and restore

### Recommendation #13: For point-in-time recovery use database log backups

A full database backup combined with a chain of log backups allows a database to be restored to a given point in time, at the granularity of a transaction. This is the highest level of granularity that it is possible to achieve with Microsoft SQL Server. The full backup may be taken with Microsoft SQL Server native backup functionality, third-party tools, or a snapshot tool like Replication Manager. However, Replication Manager (RM) cannot perform transaction log (in SQL Server terms) backups. To achieve point-in-time recoverability, RM full backups would need to be combined with SQL Server log backups.

### Recommendation #14: When possible, schedule backups for minimal disruption

When a backup is initiated, Microsoft SQL Server 2005 will do a checkpoint to flush all dirty pages to disk. When this is done on a machine that has a large amount of RAM (possibly most of which is dirty pages) that is also under a heavy I/O load, the backup may take substantially longer - sometimes as much as two to 20 times longer. Try to schedule your backups for times when the system is not under its heaviest load. It is also recommended that backup overhead be taken into account when designing RAID groups.

---

**Note:** It is intuitively obvious that a checkpoint process will run before a full or differential backup runs. However the process will also run for all transaction log backups.

---

### Recommendation #15: Do a full backup when you change the database recovery model

There are times when users might change their recovery model to simple or bulk logged, and then back to FULL. The change to FULL does not take complete effect until after a full database backup is performed. Therefore, a database that is changed from simple recovery model to FULL may lose data if the backup is taken before the recovery model change, instead of after the change.

A database does not start maintaining a log in FULL recovery mode until a full backup is done and will only maintain that recovery mode as long as nothing is done to break the log chain. For example, if the command `BACKUP LOG WITH TRUNCATE ONLY` is issued, then the database log will no longer operate in FULL recovery mode, because the log chain was broken. The only way to bring the log back into FULL recovery mode is to then take another full database backup.

## Data protection: Database mirroring

**Recommendation #16:** If using MSCS at the principal, do not use a witness server.

When running in High Availability mode (Synchronous with a witness), in the case of a cluster failover, it is likely that the database will fail over to the Mirror, before the cluster failover can complete. Therefore, when using MSCS, it is recommended that only High Protection (Synchronous without a witness) or High Performance (Asynchronous) be used.

**Recommendation #17:** Plan for high I/O levels at the mirror site

The method that database mirroring uses to commit transactions at the mirror causes substantially more write I/O than occurs on the principle. Therefore, depending on the data load, more storage resources may be required at the mirror than at the principle. Some tests have indicated that the mirror site might need to handle as much as four times the level of I/O as the principal site. It is recommended that you monitor the mirror site for performance bottlenecks that may impact your data protection and recovery plan.

**Recommendation #18:** Ensure that inter-database consistency is NOT needed

Database mirroring only maintains intra-database consistency. There is no mechanism in database mirroring to maintain consistency between multiple databases. If your application requires inter-database consistency then database mirroring as it is implemented in Microsoft SQL Server 2005 is not a recommended technology for data protection in that environment.

**Recommendation #19:** Use other methods to sync some SQL Server objects

Database mirroring only operates within the realm of a single database, and cannot be used for system databases like master. Therefore, in addition to database mirroring a separate mechanism must be used to keep other objects like user accounts, jobs and security assignments above the database level, and the system databases in sync from the principle to the mirror system. This can be done with a SQL Server job that runs at a regular interval or with some third party tools that help in this. Either way it is important that it be realized that dependant objects need to be synced as well.

**Recommendation #20:** Consult additional resources for additional information

Refer to the *Microsoft SQL Server 2005 Database Mirroring – Applied Technology Guide* for more details.



## Chapter 2 Microsoft Windows Server 2003 Best Practices

This section details recommendations for the configuration of Windows Server 2003 for use with a Microsoft SQL Server instance. This chapter presents these topics:

Recommendation #21: Only use Microsoft approved hardware.....	18
Recommendation #22: Use the latest verified NIC driver .....	18
Recommendation #23: When using MSCS, reboot the passive node occasionally .....	18
Recommendation #24: Use a dedicated VLAN for cluster heartbeat connectivity.....	18

### Recommendation #21: Only use Microsoft approved hardware

Using hardware that is on the Windows Hardware Compatibility List (WHCL) decreases the possibility of compatibility problems and increases the level of support that Microsoft will provide, should a problem occur.

### Recommendation #22: Use the latest verified NIC driver

For best performance and stability it is recommended to install the latest vendor NIC driver that has been validated for use with Windows 2003.

### Recommendation #23: When using MSCS, reboot the passive node occasionally

Many times configuration changes, especially disk configuration changes, may not be detected by the passive node until a reboot is completed. If the passive node has not detected these changes then a cluster failover may not succeed.

### Recommendation #24: Use a dedicated VLAN for cluster heartbeat connectivity

If MSCS is used, it is recommended that the cluster heartbeat network be physically isolated from other networks. For example, in a two-way cluster it is common for a crossover cable to be used between the two machines as the heartbeat network.

# Chapter 3 Network Best Practices

This section details recommendation for the configuration of your IP networks for use with Microsoft SQL Server 2005. This chapter presents these topics:

- Recommendation #25: Use Gigabit Ethernet switches with VLAN capabilities..... 20
- Recommendation #26: Use CAT6 cables for GbE connectivity..... 20
- Recommendation #27: Manually set network speed and duplexing..... 20

### Recommendation #25: Use Gigabit Ethernet switches with VLAN capabilities

Gigabit Ethernet (GbE) switches capable of setting up virtual LANs (VLAN) to segment different types of traffic should be used for best performance.

### Recommendation #26: Use CAT6 cables for GbE connectivity

It is recommended to use CAT6 cables for best performance and reliability as it showed superior results than CAT5E cables when used for 1000 Mb connectivity.

### Recommendation #27: Manually set network speed and duplexing.

After the system is set up and it has been verified that the infrastructure supports GbE properly, then the switch ports and NIC ports should be configured to 1Gbps and FULL duplex. During setup it may be necessary to use AUTO settings to ensure that everything works properly in a new environment, however the proper speed and duplex settings should be set explicitly in production systems.

## Chapter 4 Storage Systems Best Practices

This section details recommendations for the configuration of Microsoft SQL Server 2005 on Windows Server 2003. This chapter presents these topics:

Recommendation #28: Plan storage layouts for performance, not for capacity.....	22
Recommendation #29: Use DISKPART to align the LUNs for best performance .....	22
Recommendation #30: Set the NTFS allocation unit to 64 KB. ....	24
Recommendation #31: Do not exceed 80% utilization of LUNs.....	24
Recommendation #32: Use Gigabit Ethernet for iSCSI connections.....	24
Recommendation #33: Create persistent iSCSI target connections and bindings.....	25
Recommendation #34: Verify that your TcpWindowSize setting is correct.....	25
Recommendation #35: Increase your iSCSI time-out value. ....	25
Recommendation #36: Use MC/S for iSCSI high performance and high availability.....	26
Recommendation #37: Use the most recently available Celerra software. ....	26
Recommendation #38: Use dedicated Celerra file systems for iSCSI LUNs .....	26
Recommendation #39: Disable DNS on the storage network.....	26
Recommendation #40: Use Celerra storage pools for volume management. ....	26
Recommendation #41: Configure the system for high availability.....	27
Recommendation #42: Fail over Data Movers before rebooting. ....	27

## Recommendation #28: Plan storage layouts for performance, not for capacity

The most common error made while planning the storage for Microsoft SQL Server, is designing for capacity and not for performance or I/Os per Second (IOPS). To properly plan the disk layout there must be an estimate as to the number of IOPS that need to be supported on a sustained basis, the peak IOPS, and the duration of the peak.

Many customers gather data while the application is running, then use a 90th percentile to determine the level that should be planned for. There are three primary variables used for determining the number of spindles for database storage:

- ◆ IOPS (or sometimes MBps, if a serial workload)
- ◆ RAID Level -- When planning for performance, striped RAID 1 (RAID 10) will require fewer spindles than RAID for almost all read/write workloads. They are approximately equal in a read-only workload. Please see Appendix B on RAID Group Planning for additional information.
- ◆ Table 2 displays Microsoft guidelines for optimal performance with SQL Server 2005:

**Table 2** Microsoft suggested latency goals

	Read	Write
Average Latency	20ms	10ms
Max Latency	50ms	50ms

With advances in disk technology, the increase in storage capacity of a disk drive has outpaced the increase in IOPS by almost 1,000:1. With this effect it is rare to find a system that when planned for performance does not meet the storage capacity requirements for the workload. Hence, the IOPS capacity is the standard to be used while planning Microsoft SQL Server storage configurations. Only after considering the IOPS capacity of a configuration should the storage capacity (GB) be considered.

## Recommendation #29: Use DISKPART to align the LUNs for best performance

It is recommended to align the disk partition using DISKPART. When a Windows partition is created, it is created starting at the 64th sector. This misaligns the partition with the physical disk, which can cause the I/O operation to straddle stripe element boundaries and result in a significant reduction in performance. Performance improvement as high as 40% was observed on partitioning the drive using DISKPART and aligning the disk.

---

**Note:** In-depth discussion on this subject can be found in the *Using DISKPAR and DISKPART to align partitions on Windows Basic and Dynamic Disks* white paper on EMC® Powerlink®

---

The following Microsoft TechNet article also covers the topic:

<http://www.microsoft.com/technet/prodtechnol/exchange/Guides/StoragePerformance/fa839f7d-f876-42c4-a335-338a1eb04d89.msp?mfr=true>

After the LUN creation process is complete on the production system, the active MSCS node should be able to see the LUN as a raw volume.

Partition the LUN using the Microsoft command line utility DISKPART ensuring that the partition is created using ALIGN=64 switch.

The following example uses DISKPART against drive 4.

```
C:\>Diskpart
```

```
Microsoft DiskPart version 5.2.3790.1830
```

```
Copyright (C) 1999-2001 Microsoft Corporation.
```

```
On computer: JC27Q91X32
```

```
DISKPART> list disk
```

Disk ###	Status	Size	Free	Dyn	Gpt
-----	-----	-----	-----	---	---
Disk 1	Online	136 GB	112 GB		
Disk 2	Online	267 GB	0 B		
Disk 3	Online	267 GB	0 B		
Disk 4	Online	600 GB	600 GB		

```
DISKPART> select disk 4
```

```
Disk 4 is now the selected disk.
```

```
DISKPART> create partition primary align=64
```

```
DISKPART succeeded in creating the specified partition.
```

Using the Microsoft Disk Manager select the Drive Letter or Mount Point to be associated with the corresponding LUN.

### Recommendation #30: Set the NTFS allocation unit to 64 KB.

When formatting a new drive using Disk Administrator, the allocation unit size, or block size, chosen will affect application performance. For Microsoft SQL Server 2005, Microsoft recommends using a 64k block size.

---

**Note:** For more information refer to the Predeployment I/O Best Practices (<http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/pdplioebp.mspx>) site.

---

### Recommendation #31: Do not exceed 80% utilization of LUNs.

For the best performance, the utilized drive (NTFS formatted) capacity must not exceed 80 percent. There will be performance bottlenecks if this threshold is exceeded. This is because NTFS needs additional space to work efficiently. If the space is not available, NTFS cannot function to its full potential and performance can degrade. This can, in turn, cause excessive disk fragmentation, which can add to the performance degradation.

If an iSCSI LUN that is servicing Microsoft SQL Server 2005 reaches 80 percent utilized drive capacity, do one or more of the following:

- ◆ Remove unnecessary data from the disk
- ◆ Move some of the data to disks with more space
- ◆ Add more disk space

Following this recommendation will also provide you with some protection against application failure if there is unexpected growth in your database.

### Recommendation #32: Use Gigabit Ethernet for iSCSI connections

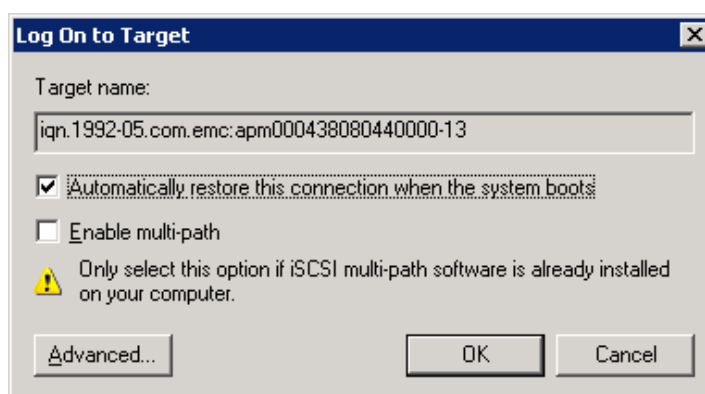
Maintaining optimal network performance is crucial to the deployment of Microsoft SQL Server 2005 on iSCSI because there is considerable network traffic generated by Microsoft SQL Server 2005. For optimal network performance, use Gigabit Ethernet cabling, switches, and network interface cards for network connections between Microsoft SQL Server 2005 servers and EMC Celerra<sup>®</sup> systems.

Use a dedicated iSCSI VLAN or a completely separate, private iSCSI network for storage.

### Recommendation #33: Create persistent iSCSI target connections and bindings.

If a service or application uses an iSCSI target volume or device, that volume or device must be bound in order for it to be available when the service or application is started by Windows.

To make target volumes and devices persistent, issue the **PersistentLoginTarget** command from the command line or select **Automatically restore the connection when the system boots** in the Log On to Target dialog box when configuring Microsoft iSCSI Initiator.



**Figure 1** Log On to Target dialog box with automatic restore option selected

To bind persistent target volumes and devices, issue the **BindPersistentVolumes** and **BindPersistentDevices** commands from the command line or use the **Bound Volumes/Devices** tab in the **iSCSI Initiator Properties** dialog box.

### Recommendation #34: Verify that your TcpWindowSize setting is correct

The **TcpWindowSize** parameter of the Windows TCP/IP stack determines the amount of available buffer on the receiver side. **TcpWindowSize** should be set to 0x0000faf0 (64240) in the registry. The Windows Registry entry is shown below:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\
TcpWindowSize (REG_DWORD)
```

---

**Note:** This recommendation is also cited as recommendation #105 (specifically, iSCSI recommendation #52) in the *Celerra Network Server 5.5 Best Practices for Performance - Best Practices Planning* white paper.

---

### Recommendation #35: Increase your iSCSI time-out value.

By default, the Microsoft iSCSI Initiator time-out value is set to 60 seconds. The time-out value determines how much time the initiator will hold a request for before reporting an iSCSI connection error. The value can be increased to accommodate longer outages, such as a Data Mover or cluster failure event.

The recommendation for this setting will vary depending on how your environment is set up to respond to a failure. In most cases EMC recommends a 600 - second timeout at the iSCSI initiator level.

To change the time-out value, search the Windows Registry for the MaxRequestHoldTime entry under HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet, and change the value to 600.

The following is an example of the Windows Registry entry:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\ {4D36E97B-  
E325-11CE-BFC1-08002BE10318}\0001\Parameters
```

```
MaxRequestHoldTime = 0x00000258 (DWORD) (600)
```

### Recommendation #36: Use MC/S for iSCSI high performance and high availability.

According to Microsoft, when a target supports Multiple Connections per Session, (MC/S) as Celerra does, and the Microsoft iSCSI Initiator is being used, Microsoft recommends using MC/S instead of MPIO.

The solution was validated with Microsoft iSCSI Software Initiator 2.04 using MC/S in “Round Robin” mode. Please check the E-Lab™ Interoperability Navigator for the latest supported version of the Microsoft iSCSI initiator.

---

**Note:** For more information, refer to the *Microsoft iSCSI Software Initiator User's Guide*.

---

### Recommendation #37: Use the most recently available Celerra software.

Install the latest available Celerra release code or patch to take advantage of new features, functionality, and bug fixes. Refer to the most recent Celerra release notes for detailed information.

---

**Note:** The solution was validated using DART 5.5.30

---

### Recommendation #38: Use dedicated Celerra file systems for iSCSI LUNs

Use file systems created for Microsoft SQL Server 2005 operations only for those operations and not for any other I/O operations. This ensures more predictable performance from Microsoft SQL Server 2005.

### Recommendation #39: Disable DNS on the storage network

The validated solution does not require DNS service on the storage network. Further, operational conflicts can occur if DNS service is not configured properly. For these reasons disabling DNS service on the storage network is recommended.

### Recommendation #40: Use Celerra storage pools for volume management.

The volume layout described in the reference architecture is designed to accommodate the I/O requirements of SQL Server database and log files while maintaining the physical separation of these elements consistent with accepted industry standards.

---

**Note:** The *Celerra Network Server 5.5 Best Practices for Performance for Performance - Best Practices Planning* white paper provides additional information.

---

## Recommendation #41: Configure the system for high availability

Export the iSCSI LUNs to two or more portals on the iSCSI target for high availability.

Use two or more separate paths from the Microsoft SQL Server 2005 cluster to different Data Mover ports for high availability.

Set up the Data Movers in active/passive mode (with one Data Mover acting as a standby) for high availability.

## Recommendation #42: Fail over Data Movers before rebooting.

The primary Data Mover will automatically fail over to the standby Data Mover if the primary Data Mover panics or fails. However, the primary Data Mover will not automatically fail over if it is rebooted. Therefore, before rebooting the primary Data Mover, manually fail over to the standby Data Mover and make sure the database is operating properly, then reboot the primary Data Mover and perform a failback operation.

Similarly, fail over to the standby Data Mover prior to performing any maintenance on the primary Data Mover.



## Appendix A Performance Monitoring and Tuning

This appendix presents these topics:

Overview .....	30
Windows performance monitoring .....	30

## Overview

Two principle items can have tremendous impact on a SQL Server storage subsystem: Server memory and I/O Subsystem setup (number of physical disks, RAID level, paths to the storage, etc.).

Server memory serves as the SQL Server's primary cache. In general, the more memory that is available for database caching, the fewer I/O operations the storage subsystem will need to service. The number of physical disk drives and RAID level used for the area where the database data files and log files will be stored determines the sustainable I/O rate the database can use without exceeding acceptable latencies. Adding more drives will generally increase the I/O rate, provided the storage connection bandwidth is not exceeded. Using the proper RAID level will also benefit how many sustained I/Os the system can handle. Also, additional storage connection paths and/or additional storage arrays can be added to lighten the load of saturated storage components. Monitoring and tuning SQL Server database installations should be a standard practice in all deployments.

There are several different tools and methodologies that can be used to help monitor and tune performance, including SQLTrace (SQL Server Profiler is the GUI), DMVs (Dynamic Management Views), Perfmon, and others. The intent of this section is not to exhaustively discuss any of these tools or any methodologies. This section is intended to simply provide more detailed information about some of the most commonly used Perfmon counters.

## Windows performance monitoring

The Windows 2003 System Monitor allows administrators to view or collect real-time performance counter information on a wide variety of operating-system and SQL Server components. Interesting counters and brief descriptions are provided in Table 3.

**Table 3 Terminology for Windows performance monitoring**

Counter name	Performance Object	Description
% Processor Time	Processor	An indication of how busy the system processors are; if the processors are very busy, the I/O system is not likely to impact overall system performance. This counter is from the Processor object. It is important to note that when using an Intel-based system with HyperThreading turned on, this counter is no longer accurate. In such a system, if this counter is hovering near 50%, it is probable that the machine is processor bound.
Pages / sec	Memory	This is one of the key indicators of Operating System level memory pressure. Some level of occasional paging is normal for most systems, however, if there are sustained periods of substantial paging or the paging occurs during periods of poor performance, then this is a strong indication of a shortage of memory.
% Idle time	PhysicalDisk or LogicalDisk	An indication of how much time a given disk/volume is not busy. If it is busy almost all the time (near 0%), the disk/volume may be a bottleneck.
Avg. Disk Queue length	PhysicalDisk or LogicalDisk	Average number of read and write requests outstanding, over the sample interval, on the disk/volume. Large queues and high disk-busy times usually indicate performance bottlenecks. When using RAID arrays, this number can be much higher than for a single physical disk drive, because a

Counter name	Performance Object	Description
		RAID array is backed by many physical disks. A semi-useful rule of thumb is to try and keep the queue length less than 2 per physical disks in the RAID array.
Current Disk Queue length	PhysicalDisk or LogicalDisk	The instantaneous number of outstanding requests on the disk, at the exact moment of sampling. Useful for tracking down temporal hot spots. If the disk queue varied between 0 and 128 during a sample interval, then this counter could be anything from 0 to 128, even though the average over the sample period might be 16. For example, if during the sample interval the disk queue length samples were (0,0,0,128), the average disk queue length would be 32, but the current disk queue length would be 128. For most situations, this is not a useful counter.
Avg. Disk sec / transfer	PhysicalDisk or LogicalDisk	Average response time, in milliseconds (ms), for disk read and write operations. Typical values of fewer than 10 ms are very good. Consistent values greater than 20 ms may indicate a problem.
Avg. Disk sec / Read	PhysicalDisk or LogicalDisk	Average response time (ms) for read operations. Useful for further isolating general response time issues. An average of less than 20 ms is desirable.
Avg. Disk sec / Write	PhysicalDisk or LogicalDisk	Average response time (ms) for write operations. Useful for further isolating general response time issues. An average of less than 10 ms is desirable.
Disk Bytes / sec	PhysicalDisk or LogicalDisk	Number of bytes transferred to or from the disk/volume per second.
Disk Transfers/sec	PhysicalDisk or LogicalDisk	Number of transfers to or from the disk/volume, regardless of transfer size. Otherwise known as IOPS.
Avg. Disk Bytes / Transfer	PhysicalDisk or LogicalDisk	A measure of the relative I/O composition of the system. This is an average but, on disks/volumes that exclusively contain database data files, it will tend toward 8 KB for most random data workloads. If the value is significantly higher than 8 KB, the workload may be more sequential in nature and benefit from additional caching. For disks/volumes that contain database log files, this value can vary substantially depending upon the workload.
Buffer Cache Hit Ratio	SQLServer : Buffer Manager	This object is useful for helping to determine whether SQL Server has sufficient memory available to it. Values of 98% or higher are excellent, 94% or higher are acceptable, and lower values are either an indication of insufficient memory or an extremely random data workload.
Page Life Expectancy	SQLServer : Buffer Manager	
Page Lookups / sec	SQLServer : Buffer Manager	Buffer Cache Hit Ratio equals (Page Lookups per sec – Page Reads per sec) divided by Page Lookups per sec. ReadAheads are not considered a cache miss, since they are not immediately being requested by a query processor, however they are disk reads and make the Buffer Cache Hit Ratio a bit misleading. For example, if all pages needed from disk, were prefetched by the read ahead manager, then the Cache Hit Ratio would be at or near 100%, which is only semi-true, as a Cache Hit Ratio of 100% would imply no physical reads would be occurring. Therefore, it is not uncommon to use an alternate calculation that takes ReadAheads into account, such as (Page Lookups per sec – (Page Reads per sec + ReadAhead Pages per sec)) divided

Counter name	Performance Object	Description
		by Page Lookups per sec. However this metric is not directly reported by the system.
Page Reads / sec	SQLServer : Buffer Manager	This object is also useful for helping to determine whether SQL Server has sufficient memory available to it. Values of less than 300 (5 minutes) are usually an indication of insufficient memory.
ReadAhead Pages / sec	SQLServer : Buffer Manager	Number of requests to find a page in the buffer pool. This counter includes pages that are requested from the buffer pool, not found, and read in from disk (Page Read).
Page Writes / sec	SQLServer : Buffer Manager	Number of physical database page reads issued. Pages reads in this counter are a direct result of needing the page to finish the execution of a query.

## Appendix B RAID Group Planning

This appendix presents these topics:

Overview .....	34
RAID level attributes .....	34
Estimating required performance .....	36
Calculating disk spindle requirements.....	38
Summary .....	40

## Overview

All RAID groups have two important quantities that can be consumed by an application workload. These are storage capacity and performance capacity. It is possible to have a RAID group whose capacity (size in gigabytes) is fully used, but the performance capacity of the RAID group is underutilized. However it is far more common for a RAID group to be under a demanding performance load, while using only a portion of its storage capacity. Since the increased capacity of disk spindles has dramatically outpaced the increase in the performance of those spindles, one of the most common errors encountered in SQL Server deployments is to find a RAID group that has been designed for capacity instead of performance.

When designing a RAID group, the first thing to consider is the level of performance that is needed, then verify if the needed capacity is available. The methods of designing a RAID group discussed herein take a conservative view of RAID group design and do not rely on features such as caching or prefetching for sustained performance.

### RAID level attributes

There are three primary RAID levels that are often discussed and offer fault tolerance: RAID 1, RAID 5, and RAID 1 with striping (RAID 10).

---

**Note:** RAID 0 is sometimes discussed, but seldom recommended because it does not provide fault tolerance. Any spindle failure in a RAID 0 group will render the entire group unusable.

---

The discussion of the various RAID levels is usually done in the rudimentary terms of capacity, whereas RAID 1 and 10 are thought of as  $2n$  or needing twice the number of disks to store a given amount of information when compared with a non RAID protected system; and RAID 5 is thought of as  $n+1$  or needing one additional spindle than is required outside of a RAID implementation to store the data. These facts are true, but are seldom the most important part of a design discussion. As previously presented, the most common limitation of a disk array for SQL Server is its performance capacity, not storage capacity.

Each RAID level has a performance impact based on the type of fault tolerance that RAID level implements. This performance impact is normally only seen during writes. For example, RAID 1 and 10 both use mirroring, therefore everything that is written to a given spindle must also be written to its partner mirror. For this reason, each logical IO issued by the host server actually turns into two physical I/Os inside the storage array and RAID 1 and 10 are thought of as having a  $2x$  write penalty.

RAID 5 is more difficult to understand. One might think that RAID 5 would have some sort of  $x+1$  write penalty, like its  $n+1$  overhead in storage capacity. However, RAID 5 actually has a  $4x$  write penalty. The specifics of of this calculation are out of scope for this document, but easily obtainable from trusted sources. Effectively each logical I/O from the host is broken down into four physical I/Os

- ◆ Read data disk being written
- ◆ Read parity disk for the stripe parity value
- ◆ Write new data to data disk
- ◆ Write new stripe parity to the parity disk

So in summary: RAID protection has an impact to the usable storage capacity, and performance capacity of an array of disks. The impact is determined by the RAID level that is selected for the array.

As the percentage of writes increases from 0% (read only) to 100% (write only), the gap in performance between RAID 1 or 10 and RAID 5 widens rapidly. The following graph shows the effects of RAID level on different % write / % read workloads.

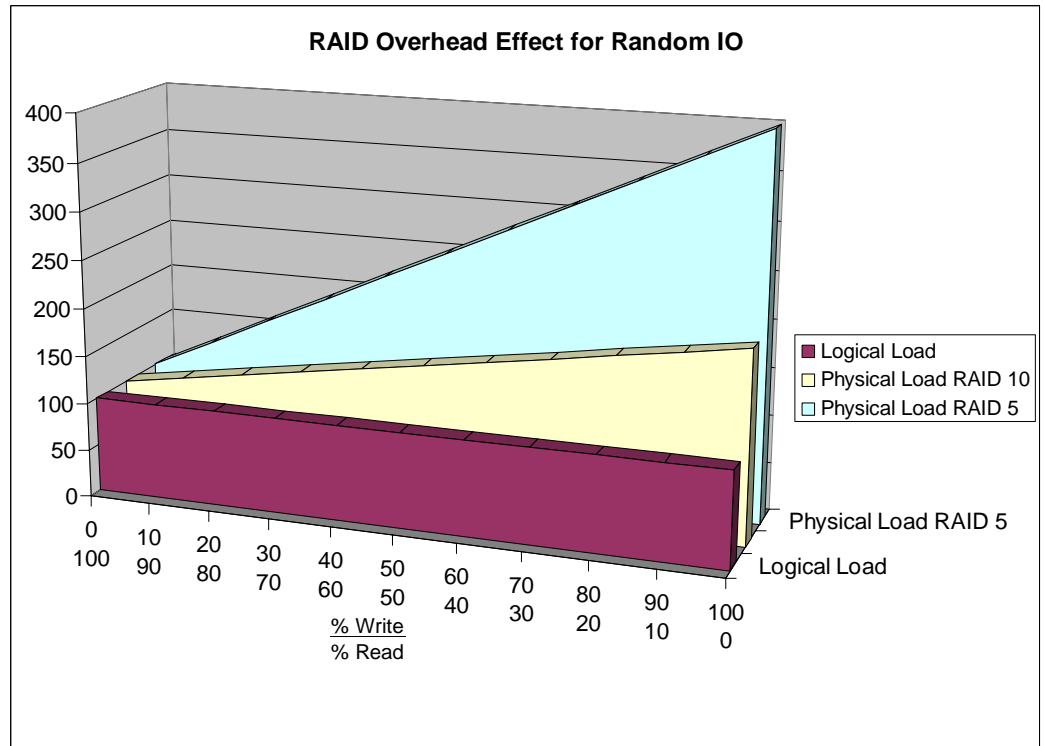


Figure 2 RAID overhead effect for random I/O

Table 4 on page 36 presents some facts about the three RAID levels that are being discussed.

**Table 4 RAID Level Performance Characteristics**

RAID Level	Random	Serial	Read	Write
1 <sup>1</sup>	Good <sup>1</sup>	Good <sup>1</sup>	Good <sup>1</sup>	Good <sup>1</sup>
5 <sup>2</sup>	Moderate	Good	Excellent	Poor <sup>3</sup>
10	Excellent	Excellent	Excellent	Better <sup>1</sup>

**Note:**

1. All RAID 1 groups are by definition limited to two drives. This places a distinct upper limit on their potential performance. RAID 10 is a method for striping data across multiple RAID 1 groups to avoid this limit.
2. RAID 5 takes a substantial performance impact during the failure of a drive and subsequent rebuild of its replacement. Therefore, this should be taken into account when planning.
3. Although RAID 5 writes perform poorly, because of the 4x penalty, there is a special case that is rarely found, where RAID 5 writes can actually outperform RAID 1 or 10 called a "full stripe write". This occurs when the write is aligned with the stripe and is the exact same width as a full stripe. For example, if each disk held 32 KB per stripe and a 4+1 array was created, then a full stripe write would need to be 128 KB in size and be aligned such that it created one full stripe across the disks. This level of performance should not be counted upon, unless testing, for the particular workload, shows that it occurs almost exclusively.

## Estimating required performance

One of the often misunderstood facets of Microsoft SQL Server is that it is not an application. It is an environment that houses databases of various types and attributes. The performance characteristics of one database can vary substantially from another database. Therefore, it is not possible to discuss how Microsoft SQL Server will perform in general for all possible workloads, but it is possible to discuss the performance of a given database when the workload characteristics are defined. Databases are usually broken down into two general classes, OLTP (OnLine Transaction Processing) and OLAP (OnLine Analytical Processing).

The usual attributes of the I/O patterns involved with each type of database, as well as tempdb, are discussed in Table 5 on page 37.

**Table 5 Microsoft SQL Server file types and performance attributes**

File type	Performance attributes
User Database Data File (OLTP)	The database data file for most OLTP (On-Line Transaction Processing) type applications usually has the following characteristics: <ul style="list-style-type: none"> <li>• Smaller I/Os</li> <li>• Random I/Os</li> <li>• High percentage of writes compared to reads</li> <li>• Not usually a very large database (aged data is usually archived to a data warehouse)</li> </ul> Based on this, RAID 10 will usually provide the best performance for a given # of spindles. Or said another way, the needed performance can usually be achieved with fewer spindles using RAID 10, rather than RAID 5.
User Database Data File (OLAP or Data Warehouse)	The database data file for most OLAP (On-Line Analytical Processing) type applications usually has the following characteristics: <ul style="list-style-type: none"> <li>• Larger I/Os</li> <li>• Serial I/Os</li> <li>• Low percentage of writes compared to reads, sometimes read-only</li> <li>• Usually a very large database</li> </ul> Based on this, RAID 5 will usually provide adequate performance and much more usable space for a given number of spindles.
Database Log File	The database log file(s) for all databases have the following characteristics: <ul style="list-style-type: none"> <li>• Smaller I/Os (some multiple of 512 bytes)</li> <li>• Highly Serialized I/Os</li> <li>• Almost exclusively writes, with occasional reads during large rollbacks or log backups</li> <li>• Size is dependent upon several factors and difficult to predict without more details about the database workload.</li> <li>• A log file is the single most important piece of information for database recovery from either a crash or database restore.</li> <li>• Every transaction that modifies data is limited by log write speed.</li> </ul> Because of the critical nature of the log files both in terms of performance and recoverability, RAID 10 is the recommended standard for database logs. There are times when RAID 5 may provide adequate performance (because of full stripe writes), but upon drive failure, RAID 5 performance will likely drop below needed levels. Also, RAID 5 cannot survive a double drive fault, while it is possible that RAID 10 will survive such a failure.

File type	Performance attributes
tempdb data file	<p>The database data file(s) for tempdb usually has the following characteristics;</p> <ul style="list-style-type: none"> <li>• Smaller or larger I/Os, depending upon usage, but many times it is larger I/Os</li> <li>• Serial or random I/Os, although a given workload might be somewhat serial, many workloads running simultaneously may give tempdb more of a random I/O appearance</li> <li>• Usually a near 50/50 split of writes and reads</li> <li>• Size can vary wildly.</li> </ul> <p>Based on the unpredictable nature of tempdb combined with its usually large percentage of writes, RAID 10 will usually provide the best performance for a given number of spindles.</p>

Please remember that these are general characteristics and that a specific user databases might generate an I/O workload that varies substantially from those presented above. Therefore, the only real way to determine the IO performance needs of a given database is to run tests with that database.

For example, in an OLTP type database, it is critical to know what level of I/O performance will be needed from the data and log RAID groups in terms of IOPS. SQL Server has many inherent buffering algorithms that are used to try and decrease I/O levels and the efficiency of these algorithms is entirely database and workload dependent.

Therefore, to get accurate performance estimates it is best to run tests with as close to “real world” conditions as possible. During these tests, Performance Monitor Logs can be used to capture the characteristics (Reads per second and Writes per second) of the volumes used for storing database files. This information can then be used to do an initial RAID group design.

---

**Note:** IOPS counters averaged over time should not be used as the basis for a RAID group design. It is recommended to find the 90th percentile of the IOPS samples and design for that performance level. This will allow your system to respond well to spikes in demand.

**Note:** The IOPS requirements for a RAID group should be calculated independently for both reads and writes.

---

## Calculating disk spindle requirements

Once the read and write IOPS are known, they can be plugged into the following formula.

$$\#ofSpindles = \frac{ReadsPerSecond + (WritesPerSecond * RAIDMultiplier)}{RecommendedIopsPerSpindle}$$

---

**Note:** Your spindle count may need to be adjusted to conform to the requirements for the RAID level you have selected. For example you cannot build a seven-spindle RAID 10 set. In such a case you would need to build an eight-spindle RAID 10 set.

---

Now that we know the formula and two of the three variables needed (read and write IOPS), the only variable still needed is the number of IOPS that a given rotational speed spindle can support.

This number is even harder to compute than any of the others discussed so far and has a very broad range of possibilities. Primarily the number of IOPS that a spindle can support is derived either by observation of workloads or computed mathematically with certain assumptions made. The primary factor that influences the number of IOPS a spindle can support is the distance between where the current I/O is occurring and where the next I/O will occur. This is referred to as the “locality of reference.”

Since it is impossible to know the locality of reference ahead of time in all non-serial workloads, certain assumptions are common to estimate how many IOPS a spindle is capable of maintaining. These assumptions usually involve taking an average between the minimum distance that would need to be moved and the maximum distance that would need to be moved and then averaging them.

---

**Note:** The exact math involved in making these assumption and estimates is beyond the scope of this paper. It is generally accepted that a 15,000 rpm disk spindle can support approximately 180 IOPS under a variety of common workloads and conditions while maintaining an acceptable latency. We will use this estimate.

---

Table 6 on page 40 shows some estimated spindle counts for a few example workloads, using the formula supplied above. Notice that in almost every example the number of spindles required for RAID 10 to hit a certain performance level is lower than that of RAID 5. Therefore, it could be said that since fewer spindles are needed, when it comes to non-read only workloads, RAID 10 is less expensive than RAID 5.

Table 6 on page 40 shows the number of spindles that would be needed for various workloads using either RAID 5 or RAID 10.

**Table 6** Number of spindles required for a series of sample workloads

Total IOPS	%Read	%Write	Read IOPS	Write IOPS	RAID 5	RAID 10
1000	100	0	1000	0	7	6
1000	75	25	750	250	11	8
1000	50	50	500	500	15	10
1000	25	75	250	750	19	10
2000	100	0	2000	0	13	12
2000	75	25	1500	500	21	14
2000	50	50	1000	1000	29	18
2000	25	75	500	1500	37	20

After a RAID group initial configuration is determined, it is then recommended that it be tested under the specific workload that it will be performing to be sure that it meets the needed performance level.

Another important point to note here is that I/O levels are being discussed at the RAID group level. Therefore, if multiple LUNs exist on a given RAID group, then the aggregate sum of the needed IOPS performance from all LUNs on that RAID group would need to be used in the equation above for its design.

## Summary

All of the data in your database environment must at some point pass through the disk subsystem. At present, for most implementations, this will have some level of RAID protection. An understanding of the various RAID protection levels and their impact to both the storage capacity and performance capacity of your disks is critical when designing a database for your workload.

## Appendix C File Group Planning

This appendix presents these topics:

TempDB .....	42
User databases .....	42
Log files .....	42

## Overview

There have been many discussions about what is the right number of data files per file group (database, if a single file group is used) and the correct answer, as always, is “it depends.”

## TempDB

In SQL Server 2000 the recommendation for tempdb was to have one file for every CPU core. Therefore, if you have a machine with four CPU sockets and used dual core CPUs, it would be recommended that tempdb be broken into eight files. This is primarily because of an issue with contention at the GAM and SGAM areas of the files. That contention has been decreased substantially in SQL Server 2005, but there is still a possibility for contention, especially since tempdb has the potential to be used much more than previous versions, because of new features like row versioning. As with all recommendations there is a cost associated with multiple files per file group. This is caused by the fact that SQL Server will stripe data across all of the files in a given file group (called proportional fill) and will likely cause all of the files to be accessed simultaneously. If the files are located on LUNs that are on the same RAID group (set of spindles), this will induce head movement, which increases latency and decreases throughput. Therefore, as with most recommendations, a conscious decision must be made to balance the decreased contention of multiple files against the increased I/O load. A good midway starting point might be to break tempdb into a number of files equal to half the number of CPU cores.

---

**Note:** The article, *Concurrency enhancements for the tempdb database* (<http://support.microsoft.com/kb/328551/en-us>) on the Microsoft website discusses this in more detail.

---

## User databases

For user databases a similar decision must be made and similar criteria should be used. However, the contention of user databases is usually much lower, and hence, we should start with a smaller number of files like one or two and then increase, if needed.

## Log files

Increasing the number of files available to a SQL Server database does not enhance performance. If a database has two log files, Microsoft SQL Server will fill the first log file before beginning to use the second log file. Therefore, the only use of a second log file is to expand a database’s logs onto a new volume.

---

**Note:** For a description of the physical architecture of the Microsoft SQL Server 2005 NS Series iSCSI solution, including a topology diagram, see the *EMC Solutions for Microsoft SQL Server 2005 EMC Celerra NS20 over iSCSI - Reference Architecture*.

---