

CONFIGURING MICROSOFT SQL SERVER 2005 ON DELL SERVERS AND STORAGE

BY WENDY CHEN
SUMNER STEWART
PAUL RUBIN
CHETHAN KUMAR
ALVIN PENG

Following best practices from Dell and Microsoft can help enterprises design a high-performance Microsoft® SQL Server™ 2005 database on Dell™ PowerEdge™ servers and Dell PowerVault™ storage. This article provides specific guidance and best-practice recommendations in the areas of storage, OS, and database configuration.



Applying best practices derived from laboratory and real-world experiences can help IT professionals design and configure reliable, high-performance database environments. Microsoft and Dell offer several best practices for implementing, configuring, and managing Microsoft SQL Server 2005 database solutions in enterprise architectures that include Dell PowerEdge servers, Dell PowerVault MD3000 Serial Attached SCSI (SAS) storage enclosures, and the Microsoft Windows Server® 2003 OS. Dell PowerEdge servers are designed to deliver high performance for critical enterprise applications such as databases, messaging, Web services, and infrastructure. The Dell PowerVault MD3000 is a high-performance storage array built for critical applications running on one, two, or a pair of clustered PowerEdge servers. By combining Dell PowerEdge servers with Dell PowerVault MD3000 storage systems, administrators can easily deploy PowerEdge servers as building blocks of database solutions. Working in tandem, Dell servers and storage can help consolidate and virtualize computing and storage resources for SQL Server–based environments.

Figure 1 illustrates an architecture for SQL Server 2005 on Dell hardware. This architecture comprises the following components:

- **Client systems:** The clients access data stored within the SQL Server 2005 database.

- **Client-server network:** The connecting network consists of network controllers, cables, and switches.
- **Dell PowerEdge servers:** The servers run Windows Server 2003 and SQL Server 2005.
- **Server-storage interconnect:** The servers and storage devices connect with SAS.
- **Dell PowerVault storage:** The storage hardware supports up to 45 drives.

Preparing the storage systems

The physical disks in the Dell PowerVault MD3000 storage array provide physical storage capacity for the SQL Server 2005 database. Before data can be stored, the PowerVault MD3000 physical storage capacity must be configured into components known as *disk groups* and *virtual disks*. A disk group is a set of physical disks that are logically grouped and assigned a RAID level. Virtual disks are logical entities that the server uses to store data. Each disk group provides the overall capacity needed to create one or more virtual disks.

SQL Server 2005 uses three specific storage areas for tempdb, the transaction log file, and data for user-defined databases. Tempdb is a shared working area for all databases on the server and for activities such as creating temporary tables, sorting, and processing sub-queries. The transaction log file stores the details of modifications made

Related Categories:

Database

Dell PowerEdge servers

Dell PowerVault storage

Microsoft SQL Server 2005

Storage

Visit DELL.COM/PowerSolutions
for the complete category index.

“Working in tandem, Dell servers and storage can help consolidate and virtualize computing and storage resources for SQL Server-based environments.”

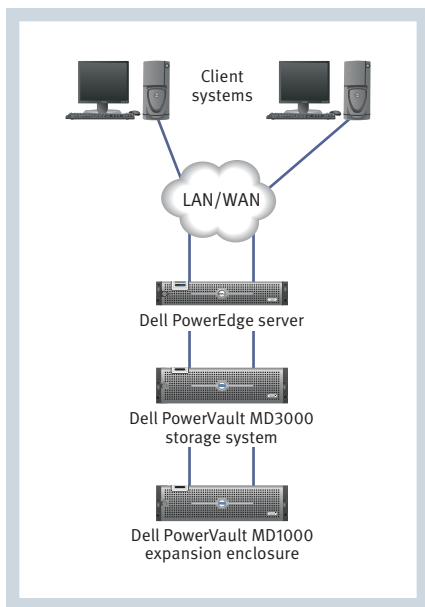


Figure 1. Architectural overview for Microsoft SQL Server 2005 on Dell hardware

to a SQL Server database along with the details of the transactions that performed each modification. Data for user-defined databases is stored in a separate data storage area. This data file contains user-defined database objects such as tables, indexes, and stored procedures.

Microsoft best practices recommend separating the tempdb and transaction log files onto separate physical disks—which, for the Dell PowerVault MD3000, means separating these files onto their own virtual disks on separate disk groups. This separation can enhance I/O performance by ensuring that these files do not share the same physical disks. Figure 2 illustrates a

sample disk group and virtual disk configuration that separates tempdb, transaction logs, and data files onto distinct physical disks.

Creating disk groups and virtual disks

Microsoft recommends two common best practices with respect to storage configuration for SQL Server 2005. The first best practice is to spread database objects across multiple disk spindles; the second is to limit the number of data files based on the number of processor cores.¹ For design simplicity and to help avoid the complexity introduced by having too many virtual disks, the sample configurations described in this article also map the number of virtual disks for data to the number of processor sockets. For example, a two-socket system (with either dual-core or quad-core processors) would have two virtual disks for data. Figure 3 illustrates sample virtual disk configurations and sizes for four-socket host servers.

RAID-10 is considered the optimal choice for SQL Server 2005 virtual disk implementation because it offers high performance and fault tolerance by combining mirroring and striping.² When possible, the disk groups that contain tempdb, the transaction logs, and virtual disks for the data files should be configured with RAID-10.

Because additional drives are required to implement RAID-10, it may not be the preferred choice for certain databases. In these cases, RAID-0 can be used for the tempdb virtual disk. If RAID-0 is used, however, loss of a tempdb disk will affect system availability.

For the transaction log virtual disk, RAID-1 can be used instead of RAID-10 because it provides protection against drive hardware failure. For the virtual disks for the data files, RAID-5 provides a cost-effective alternative, especially for predominantly read-only workloads such as a data warehouse database. However, RAID-5 is not suitable for heavy write workloads such as an online transaction processing database, because it can have significantly lower write performance than other RAID levels due to the reading and writing of parity blocks in addition to the reading and writing of database data.

Configuring the OS and the file system

As mentioned in the preceding section, Microsoft recommends that the number of

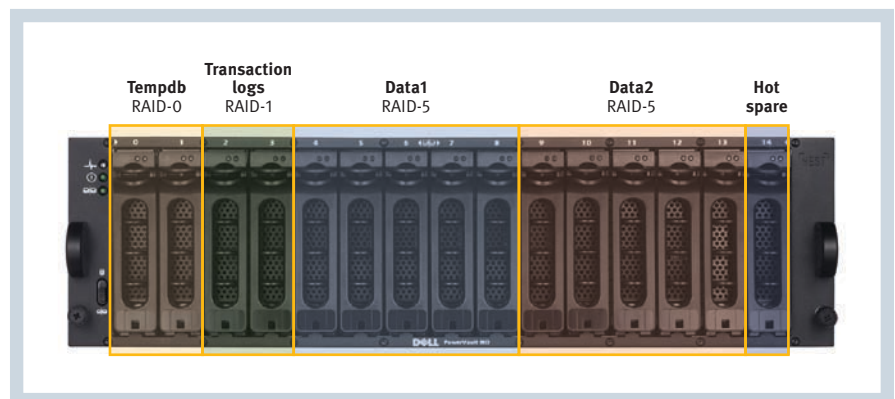


Figure 2. Separation of tempdb, transaction logs, and data files onto distinct physical disks within a Dell PowerVault MD3000 storage array

¹“Storage Top 10 Best Practices,” by Microsoft, *Microsoft TechNet*, October 17, 2006, www.microsoft.com/technet/prodtechnol/sql/bestpractice/storage-top-10.mspx.

²“Physical Database Storage Design,” by Kathy Lu and Lewis Bruck, *Microsoft TechNet*, February 12, 2007, www.microsoft.com/technet/prodtechnol/sql/2005/physdbstor.mspx.

Virtual disk	Minimum size	RAID level	Use
Data1	25% of the database size	RAID-10; for read-only, RAID-5	Data
Data2	25% of the database size	RAID-10; for read-only, RAID-5	Data
Data3	25% of the database size	RAID-10; for read-only, RAID-5	Data
Data4	25% of the database size	RAID-10; for read-only, RAID-5	Data
Tempdb	Variable (depends on the required tempdb size)	RAID-10 or RAID-0	Tempdb
Transaction logs	10% of the database size	RAID-10 or RAID-1	Logs

Figure 3. Recommended disk configurations for four-socket servers running Microsoft SQL Server 2005

Drive letter	Folder names	Format	Use
G:	MSS_DATA_1, MSS_DATA_2	NTFS 64 KB	Data
H:	MSS_DATA_3, MSS_DATA_4	NTFS 64 KB	Data
I:	MSS_DATA_5, MSS_DATA_6	NTFS 64 KB	Data
J:	MSS_DATA_7, MSS_DATA_8	NTFS 64 KB	Data
K:	TEMPDB_DATA_1, TEMPDB_DATA_2, TEMPDB_DATA_3, TEMPDB_DATA_4, TEMPDB_DATA_5, TEMPDB_DATA_6, TEMPDB_DATA_7, TEMPDB_DATA_8	NTFS 64 KB	Tempdb
L:	MSS_LOG1, TEMPDB_LOG1	NTFS 64 KB	Logs

Figure 4. Sample file system configurations for four-socket, dual-core servers running Microsoft SQL Server 2005

data files configured for a SQL Server 2005 database equal the total number of processor cores installed on the server to achieve optimal performance and scalability for heavy workloads. Also, in the configurations described in this article, the number of virtual disks for data is equal to the number of physical processor sockets. Each virtual disk created in the storage system is mapped to an OS file partition. Therefore, within each partition used for data, the recommended number of data files should equal the number of cores per processor socket. For example, a four-socket, dual-core system should have two data files for each data partition. The number of tempdb data files should also equal the number of processor cores.

Each OS file partition is assigned a unique drive letter. Microsoft best practices recommend that all partitions be formatted using NT

File System (NTFS) with 64 KB clusters. Figure 4 illustrates sample file system configurations for four-socket, dual-core host servers.

Using Microsoft Windows Address Windowing Extensions

Microsoft Windows Address Windowing Extensions (AWE) extend Windows memory management to allow applications to address more memory than the standard 32-bit addressing space of 2–3 GB. With AWE, SQL Server 2005

can support up to 64 GB of physical memory on Windows Server 2003.

AWE applies to 32-bit operating systems with more than 4 GB of physical memory. By default, the 32-bit Windows Server 2003 OS reserves 2 GB of memory for the OS and leaves 2 GB of available process address space—known as user-mode virtual address space—to applications. Windows Server 2003 has a boot.ini switch called /3GB that provides applications access to 3 GB of process address space and limits the OS to 1 GB of physical memory. For a system with less than 16 GB of physical memory, a /3GB switch and a Physical Address Extension switch (/PAE) should be added to the boot.ini file. For a system with more than 16 GB of physical memory, a /PAE switch should be added to the boot.ini file.

To enable AWE for SQL Server 2005, administrators should make sure that the “Maximize data throughput for network applications” option is selected in the Network Connections section of the Control Panel. This option may limit the available memory to SQL Server operations because it gives priority to applications that perform buffered I/O operations by caching their I/O pages in system cache.

Configuring Microsoft SQL Server 2005

Installing SQL Server 2005 builds a single tempdb data file by default. When the database is initially created, the tempdb files are placed on the server’s internal drives. Microsoft best practices recommend moving the tempdb files to external disk drives dedicated for tempdb files. Also, creating additional tempdb data files can help avoid latch contention on allocation pages and help solve I/O performance

“By combining Dell PowerEdge servers with Dell PowerVault MD3000 storage systems, administrators can easily deploy PowerEdge servers as building blocks of database solutions.”

“The best practices described here are intended to help achieve optimal performance of SQL Server 2005, help simplify operations, improve resource utilization, and cost-effectively scale systems as an organization’s needs change over time.”

issues. The recommended number of tempdb data files should match the number of processor cores of the server. To help effectively utilize the allocation mechanism, all tempdb data files should be equal in size.

By default, the autogrow option is enabled for tempdb files. However, expanding tempdb too frequently can lead to performance degradation. To avoid this issue, Microsoft recommends pre-allocating the tempdb space with enough space to accommodate the expected workload and setting the FILEGROWTH increment large enough to minimize tempdb expansions. The Microsoft-recommended FILEGROWTH increment setting is 10 percent.³

When a new data file is created or a data file expands as a result of the autogrow function, the pages in the file are initialized—zeros are automatically written to the file before it can be used. This file-zeroing process can be time-consuming and can lead to application time-out, especially during the autogrow activity.


One of the major improvements in SQL Server 2005 is the instant file initialization feature. This feature skips the file-zeroing process; instead, the pages in the data file are overwritten with new data when pages are actually allocated. In February 2007, Microsoft conducted in-house tests of file creation and growth processes that showed a significant performance increase when using this feature. For example, the time required to

add a 100 MB file dropped from 2.8 seconds to 0.2 seconds with instant data file initialization.⁴

Enabling Address Windowing Extensions

To enable AWE on SQL Server 2005, the SQL Server 2005 engine must run under a Windows account that has the “Lock pages in memory” option assigned (this option is disabled by default). Enabling this option allows SQL Server 2005 to keep data in physical memory and prevents it from sending data to virtual memory. Next, administrators should use the `sp_configure` command to set the `awe_enabled` option to 1, and restart SQL Server 2005.

Following best practices for optimal database performance

Maximizing the performance of databases and their supporting storage systems while reducing management complexity can be critical for a growing enterprise. This article provides a blueprint for configuring a stand-alone Microsoft SQL Server 2005 database on Dell PowerEdge servers and Dell PowerVault MD3000 storage arrays. The best practices described here are intended to help achieve optimal performance of SQL Server 2005, simplify operations, increase resource utilization, and cost-effectively scale systems as an organization’s needs change over time. 

Wendy Chen is a systems engineer with the Dell Database, Applications, and High-Availability team. Her fields of interest include Oracle® database backup and recovery, disaster recovery, and Oracle Real Application Clusters. Her experience includes Oracle database administration, application development, and database solutions engineering. Wendy has an M.S. in Computer Science from Brigham Young University.

Sumner Stewart is a senior database administrator with Dell IT Global Database Management Systems. He currently consults and supports the Dell manufacturing business segment for global applications. Sumner’s primary focuses include multitiered architectures with zero-downtime tolerances and high-performance transaction systems.

Paul Rubin is a senior product manager in the Dell Software Solutions marketing group. He is responsible for planning Dell database solutions. Paul has a B.A. in Computer Science from the University of Texas at Austin.

MORE

ONLINE

DELL.COM/PowerSolutions

QUICK LINKS

Microsoft SQL Server 2005:
www.microsoft.com/sql

Microsoft SQL Server TechCenter:
technet.microsoft.com/sqlserver

Dell database solutions for Microsoft SQL Server 2005:
DELL.COM/SQL

³For more information about configuring tempdb files, see “Working with tempdb in SQL Server 2005,” by Wei Xiao, Matt Hink, Mirek, and Sunil Agarwal, *Microsoft TechNet*, June 26, 2006, www.microsoft.com/technet/prodtech-nol/sql/2005/workingwithtempdb.mspx.

⁴“Physical Database Storage Design,” by Kathy Lu and Lewis Bruck, *Microsoft TechNet*, February 12, 2007, www.microsoft.com/technet/prodtechnol/sql/2005/physdbstor.mspx.