

Realtime
publishers

The Shortcut Guide[™] To



Storage Considerations for Microsoft SharePoint

sponsored by



Wendy Henry

Chapter 2: SharePoint Storage Architecture	21
Components of a SharePoint Server	22
Microsoft Windows OS and Virtual Memory Management	22
Microsoft IIS Web Delivery	28
Microsoft SQL Server Relational Database Management System	30
SharePoint Databases	36
Managing SharePoint Configuration Databases and Logs	37
Managing SharePoint Content Databases and Logs	39
Managing SharePoint Administration Databases and Logs	42
SharePoint Search/Index and External Content	43
Managing SharePoint Search Databases and Logs	43
Examining Full Text and Optional Metadata Indexes	44
Interfacing External Content via SharePoint	44
Storage Options for SharePoint	45
Internal Disks and DAS	45
NAS and SAN Strategies	47
How to Choose Appropriate Storage	48
Summary	50

Copyright Statement

© 2009 Realtime Publishers, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers, Inc or its web site sponsors. In no event shall Realtime Publishers, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

[Editor's Note: This eBook was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology eBooks and guides from Realtime Publishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 2: SharePoint Storage Architecture

Based on the information explored in Chapter 1, it should be apparent that analyzing SharePoint storage demands more than just a single focus. Because there are so many Microsoft applications and services at work, not to mention the Windows OS itself, it is important to take each individual product into account when determining the best storage solution. From the use of Virtual Memory Manager in Windows to the art of optimizing SQL Server 2005 databases, there are many considerations to examine. This chapter will dissect the OS and each application supporting a SharePoint installation—in particular, the many SQL Server 2005 databases required for SharePoint will figure prominently as well as IIS optimization and SharePoint indexing behavior.

When it comes to managing storage on any Windows Server OS, there is more than meets the eye. And although it may seem simple to concentrate solely on the I/O subsystem of hardware and the disk controller driver software, doing so would only expose half of the picture. In Microsoft, as with many platforms, storage management has an additional influence that can affect readings and decrease disk efficiency. This influence is memory management or, more specifically, the Memory Manager. The first section of this chapter will expose the Memory Manager and its effect on storage performance.

No SharePoint discussion would be complete without including SQL Server, so the second section of this chapter will focus on optimizing both system databases as well as SharePoint-specific databases in a SQL Server 2005 instance. Because SharePoint installs so many databases, each with a very different responsibility, there is no one single solution to distributing SharePoint database files or maintaining fault tolerance for transaction logs. Additionally, indexes play a large role in the efficiency, or lack thereof, as well as the performance of SharePoint. Therefore, we will also take a look at full text index management and the challenges of crawling external content.

Lastly, we will focus on the different storage products available on the market and which lend themselves most advantageously to SharePoint and SQL Server. This chapter will examine internal disks, compare NAS and SAN storage strategies, and discuss factors in choosing appropriate storage for both your budget and your goals. SharePoint is not too unlike other enterprise-class network applications in that bigger is not always better and flexible scaling and availability is paramount to the long-term maintenance of your SharePoint implementation. We have a lot to cover, so let's get started!

Components of a SharePoint Server

Before beginning a lengthy discussion of the individual actors in a SharePoint play, it is important to understand how they work together to provide interface into the shared storage of a SharePoint collaborative environment. The Windows OS is the fundamental abstraction layer for all the necessary network applications and would be influential on any of the servers in a scaled SharePoint solution. Internet Information Server (IIS) can impede or enhance rendering of SharePoint Web pages and has its own distinct resource needs. SQL Server 2005 is the true workhorse of the entire SharePoint installation as it stores and delivers not only the user content submitted to SharePoint but also all the configuration and layout page information necessary for Windows SharePoint Services to function. Figure 2.1 illustrates the roles each component plays in delivering requested SharePoint resources to users.

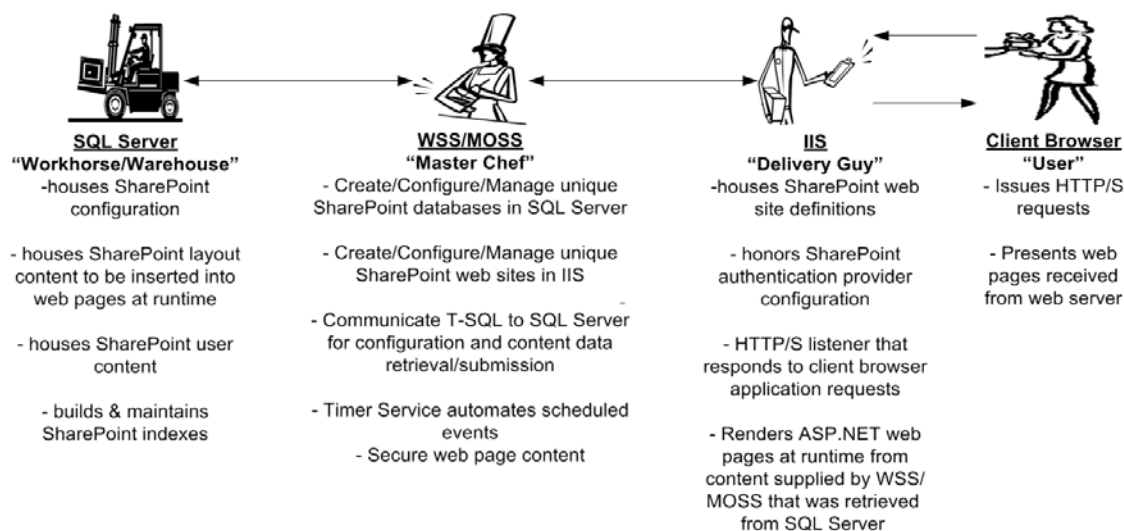


Figure 2.1: Network application roles in a SharePoint environment.

Microsoft Windows OS and Virtual Memory Management

As with any discussion, it is always prudent to start at the beginning, and the beginning for this chapter is the OS on which SharePoint will be installed. Windows Server 2003 and 2008 are the supported Windows OSs for Windows SharePoint Services v3.0 and Microsoft Office SharePoint Server 2007, each with its own disparate minimum requirements (see Table 2.1). SharePoint itself has robust minimum requirements beginning with the dependent WSSv3 requirements that can be even higher if all enterprise-class features of Microsoft Office SharePoint Server 2007 (MOSS 2007) are installed. Be sure to purchase beyond minimum requirements to adequately support a production SharePoint environment.

Component	Windows Server 2003 R2 Enterprise Edition.	Windows Server 2008 Enterprise Edition	Windows SharePoint Services v3.0
Processor	133MHz(x86)	1GHz(x86)/1.4GHz(x64)	2.5GHz
Memory	128MB	512MB	1024MB
Available Disk	2.9GB	10GB	3GB (NTFS)

Table 2.1: Comparison of minimum requirements by OS and WSSv3.

One notable requirement from Table 2.1 is the NTFS file system demanded by Windows SharePoint Services v3.0. Windows Server 2003 and 2008 support both FAT and NTFS file systems, but NTFS performs better on large disk volumes regardless of the underlying storage hardware platform or configuration. Alas, there is always room for improvement and NTFS can be tuned to better handle large files that tend to grow over time, such as the SQL Server database files of SharePoint.

Each NTFS volume maintains a master file table (MFT) that records metadata about every file on the volume, including the MFT itself. During volume formatting, 12.5% (by default) of the total volume size will be allocated for the MFT file. This initial space reservation can be increased using the Windows CLI utility fsutil.exe in an attempt to keep the MFT as contiguous on the volume as possible (see Sample 2.1). However, this value cannot be set lower than the default 12.5% of the volume size. When the MFT exceeds its initial reserved footprint, NTFS will grow the MFT file, thus potentially causing fragmentation. Conversely, as files are deleted from the volume, their metadata is removed from the MFT generating empty space within the table. The onboard Windows Disk Defragmenter utility cannot operate on the MFT, so consider purchasing a third-party enterprise-class defragmenter utility that handles MFT files and offers automation to keep things running smoothly.

Warning

Defragmenting and compacting the MFT can improve disk seek times, but the MFT is incapable of releasing empty space back to the volume for other purposes. Avoid saving then deleting unnecessary files on an NTFS volume causing the MFT to grow but never shrink.

In addition to maintaining the MFT, you can further improve NTFS performance by modifying practices that generate undue disk activity. For example, use `fsutil.exe` to disable 8.3 alias generations for long file names on volumes containing more than 300,000 objects that do not support legacy applications. You may also develop a file-naming convention that places differentiating characters at the beginning of names to reduce metadata reads when multiple similarly named files reside in the same directory. Defragment volatile volumes often, perhaps even on an automated schedule, using the onboard Windows Disk Defragmenter and Scheduled Tasks utilities or with a third-party application. You might also opt to disable Last Access Time via `fsutil.exe`, eliminating write activity to the MFT every time a file is accessed even if just for read purposes. Lastly, consider increasing the default 4KB cluster size to 16KB, 32KB, or 64KB (depending on volume size) when initially formatting the volume if files tend to be large and grow over time such as SQL Server database files. Deviating from a 4KB cluster size will disable NTFS software compression, but you could invoke hardware compression from your storage provider instead. These are just a few examples of strategies that can improve overall NTFS file system performance in any Windows Server OS of NT architecture.

```

fsutil behavior set mftzone {x}          --increase MFT space reservation
                                         --possible values of {x}:
                                         1 = 12.5% (default)
                                         2 = 25%
                                         3 = 37.5%
                                         4 = 50%

fsutil behavior set disable8dot3 1       --disables 8.3 alias creation

fsutil behavior set disablelastaccess 1   --disables Last Access write

format {volume} /A:{cluster size}        --sets cluster size during format
                                         --possible values of {cluster size}:
                                         512    2048    8192    32K
                                         1024   4096    16K     64K

```

Sample 2.1: Syntax examples for `fsutil.exe` and `format.exe` CLI utilities.

Another revealing piece of information from Table 2.1 is that the memory requirement of SharePoint exceeds the OS memory requirements by double or more. Yet even if more than 1GB of physical RAM is installed in the server, supporting several consumers including the OS, various services, and multiple applications will cause contention for the limited resource. So how does the Windows OS handle competing requests for memory space?

- A paging file named `pagefile.sys` is created during bootup on the volume dictated in System Properties (by default, the volume that the OS resides on, also known as the `%systemdrive%` variable).
- The Memory Manager operates in Windows kernel mode and issues memory space to requests from the user-mode subsystems that support programs. Programs are issued virtual memory addresses regardless of physical RAM free (defaults: 2GB on x86; 8TB on x64).
- Memory Manager writes data from stagnant memory pages to the paging file on the hard drive when additional requests for memory are received but insufficient free physical memory exists. This process is called paging and generates disk I/O (write) activity. Some OS data is ineligible for paging and so remains in the NonPaged Pool of memory, one of two dynamic virtual memory pools created during boot up.
- If a program attempts to collect data that has already been written to disk, Memory Manager will retrieve the data from the paging file and place it back into memory (but not necessarily back into the same address from which it was paged). This event is reported as a Page Fault and generates disk I/O (read) activity.
- If no memory space is available upon Page Fault to read the disk data back into, then Memory Manager may first have to free up some memory space by performing paging on other stagnant pages. Such processing generates both read and write disk I/O activity.

Note

In the discussion of memory pages, the term “stagnant” does not necessarily refer to that data that has been in RAM for the longest period of time. Instead, “stagnant” refers to data in RAM that has met the OS’ criteria for being written to disk. Windows Server OSs use the Least Recently Updated (LRU) algorithm to determine which page of memory to write to disk during the paging process. LRU is beyond the scope of this guide.

As you can see, virtual memory management and the paging process can generate disk activity beyond what I/O resources are required by IIS, SQL Server, and SharePoint. Insufficient physical RAM in a busy server degrades both OS and application performance. Procure enough physical RAM to accommodate all applications concurrently as well as the OS and its services. Additionally, you can implement virtual memory optimization strategies across multiple physical disks such as:

- Set the maximum page file size equal to the initial size to alleviate autogrowth overhead
- Place the pagefile.sys paging file on a separate physical disk than that hosting the Windows OS files to reduce contention (note that this action disables full memory dumping for troubleshooting)
- Add a paging file to multiple physical disks to give the OS more virtual memory space without contention

Generating multiple paging files on separate disks can improve the speed of the paging process as the OS will take advantage of paging to multiple disks concurrently. However, increasing the number of files will add workload to the processor resulting from concurrent disk activity. You must try to strike a balance between processor overhead and improved paging performance or risk diminishing return. Use the System Monitor object counters shown in Table 2.2 to identify excessive paging (sometimes referred to as disk thrashing) and processor activity.

Object	Counter	Description	Preferable Values
Memory	Pages/Sec	Rate at which pages are read from or written to disk to resolve hard page faults	0-20 avg
Memory	Available MBytes	Physical memory available to processes including free, zeroed, and standby pages	>20% RAM
Paging File	% Usage	Amount of page file in use	<80
Physical Disk	% Disk Time	Percentage of time interval disk was busy with read or write requests	<50
Process	Page File Bytes	Number of bytes the selected process has used in the paging file(s)	n/a—use to determine glutton exe
Processor	% Processor Time	Percentage of time interval processor spent executing non-idle threads	<75 constant
Processor	% Privileged Time	Percentage of time interval processor spent executing code in privileged mode (kernel threads)	<50 constant

Table 2.2: Virtual Memory management and optimization monitoring counters.

Keep in mind that the pagefile.sys file, like any other file experiencing growth, can become fragmented over time. Unfortunately, the onboard Windows Disk Defragmenter utility cannot operate on pagefile.sys for the same reason it does not defragment NTFS MFT files: because system files are in constant use by the OS running Disk Defragmenter itself. Consider employing the Microsoft Windows SysInternals utility called PageDefrag to defragment paging files or purchase a third-party enterprise-class defragmenter utility that handles paging files. As always, back up the registry before making changes and continue monitoring to verify benefits.

Microsoft IIS Web Delivery

Now that we have fine-tuned the Windows Server OS for network application support, let's take a look at the first dependent application, IIS. When it comes to optimizing this Web server application, most documentation available from Microsoft focuses on the processor and memory because these are the two hardware resources most heavily utilized by IIS services. So don't forget all the virtual memory management enhancement you just read in the previous section!

Storage requirements demanded by IIS in a SharePoint environment depend on the server's role in the SharePoint farm. Recall from Chapter 1 of this guide that SharePoint can be deployed on a single server where SQL optimization takes precedence over IIS optimization or in a scaled farm where tactics can be employed on the IIS Web front-end server to distinctly optimize IIS performance. Simple tricks with configuration file administration, queue management, and authentication can improve IIS disk utilization.

- **Configuration File Administration**—Location: Both IIS version 6 and 7 use XML-formatted files to store configuration settings. Version 6 loads its MetaBase.xml and MBSchema.xml data into memory at startup, unlike version 7, which operates from the disk resident applicationhost.config and other files. Both versions benefit from reduced disk contention with the paging file when you place pagefile.sys on a separate disk, albeit version 7 enjoys the stronger performance advantage.

History: Both IIS 6 and 7 automatically maintain histories of their configuration files. Although histories enhance diagnostic and recovery efforts, they can also usurp disk space. Consider reducing the number of histories kept or disabling histories if not needed (see Sample 2.2).

IIS 6: %windir%\system32\inetsrv\metabase.xml

```
<IISComputer ... EnableHistory="1" ... MaxHistoryFiles="10" >
  --to disable automatic histories, set EnableHistory value to "0"
  --to reduce number of histories kept, alter value of MaxHistoryFiles
```

IIS 7: %windir%\system32\inetsrv\config\iis_schema.xml

```
<sectionSchema name="system.applicationHost/configHistory">
  <attribute name="enabled" type="bool" defaultValue="true" />
  ... <attribute name="maxHistories" type="uint" defaultValue="10"
validationType="integerRange" validationParameter="1,4294967295"/>
  ... </sectionSchema>
  -- to disable automatic histories, set enabled defaultValue to "false"
  -- to reduce number of histories kept, alter maxHistories defaultValue
```

Sample 2.2: Configuration history settings in IIS 6 and 7.

- Queue Management—Maintaining adequate request queue sizes and ASP.NET session state information will reduce roundtrips to the SharePoint databases in SQL Server, conserving disk read access. Insufficient queue size reveals itself as 503 errors from the WWW service and can be remedied in both versions of IIS by modifying the performance properties of the reporting application object (see Figure 2.2). SharePoint creates its ASP.NET Web applications with SQL Server session state settings. Do not alter these settings on a SharePoint Web site's properties or session information may become corrupt. Always use SharePoint Central Administration to scale multiple WFE Web servers in your farm or risk incorrect configuration of the ASP.NET application path on some servers resulting in session state failures.
- Authentication—Consider carefully before deciding to employ an ASP.NET provider for SharePoint authentication rather than using Windows authentication. Creating an additional SQL Server database via ASP.NET registration (aspnet_regsql.exe) for the purpose of housing non-Windows users and roles will require additional hard drive space and generate significant read access upon user authentication into IIS sites. Though this model is sometimes appropriate for public or extranet SharePoint environments, its use demands additional disk space and usage.

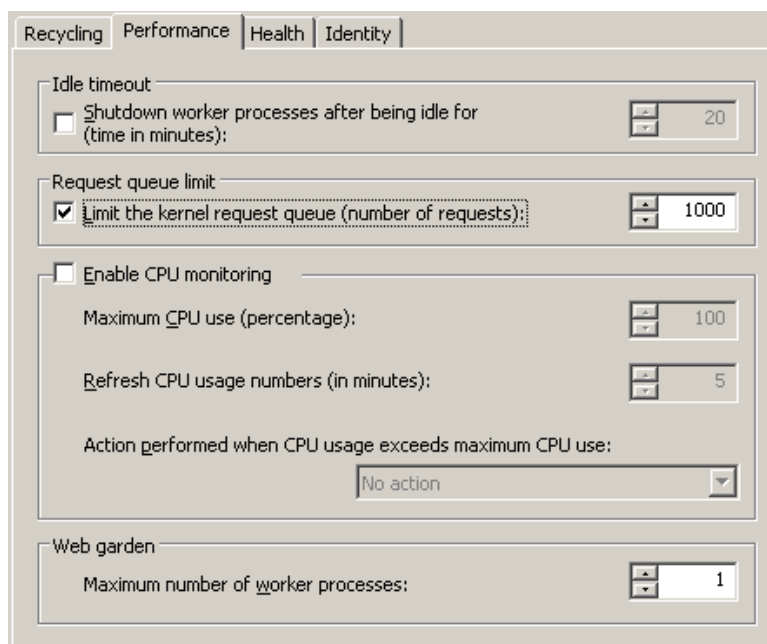


Figure 2.2: Properties window of IIS 6 application pool object—configure request queue limit.

Microsoft best Practices for SharePoint strongly discourage direct editing of IIS and the SharePoint Web applications within via IIS Manager. Instead, Microsoft suggests that any configuration changes to IIS or its objects be performed solely through stsadm.exe or SharePoint Central Administration. So consider the preceding information a bit of nickel knowledge and know that you are skating at your own risk if you choose to employ such IIS optimization strategies.

Microsoft SQL Server Relational Database Management System

Inexorably, we must now scrutinize the largest resource consumer in a SharePoint enterprise: SQL Server. Microsoft's relational database system not only requires the most disk space of the SharePoint components but also is a large patron of processor and memory. Such an appetite is the reason the medium farm discussed in the previous chapter of this guide is a very common SharePoint enterprise design. Although both Windows Server 2003 and 2008 are still available in 32-bit versions, as are SQL Server 2005 and 2008, purchasing such a limited architecture for SharePoint would be a great disservice to your network. 64-bit systems can utilize far beyond the 4GB maximum memory addressing supported by 32-bit systems. Adding physical RAM to a SQL Server and running a 64-bit version of the OS and SQL Server on 64-bit hardware boosts both memory and processing prowess.

Slow database response time, historically a factor in Web service design, tends not to pose as much of a threat in SharePoint Web environments. ASP.NET can be vulnerable to slow responses from remote data sources resulting in worker thread waits that stall the WWW service. Figure 2.1 earlier in this chapter illustrates how WSS intercedes for SharePoint Web applications, alleviating much of the direct communication between IIS and SQL Server. Time-out configuration settings available in SharePoint Central Administration can be leveraged to accommodate slow SQL Server responses and reduce Web service interruptions. Even so, you should still endeavor to procure the most capable hardware and medium possible.

Stuck in 32-bit land?

You might already own an x86 32-bit system, and it is possible you have invested in great RAM but still suffer excessive disk thrashing due to virtual memory paging. Windows Server 2003 and 2008 (32-bit) only support up to 4GB of memory addresses divided into 2GB for kernel use and 2GB for user mode. If you want to tilt this ratio or if more than 4GB of physical RAM is installed on an IA32 system, consider the memory enhancements to the OS and SQL that may reduce thrashing provided in Table 2.3.

Tool	Description
/3GB	Windows OS boot.ini switch that changes the default 2GB kernel vs. 2GB user scale to 1GB kernel vs. 3GB user so that user-mode applications receive up to 3GB of possible memory addresses (physical and/or virtual); does not necessarily alleviate excessive paging or disk activity
/PAE	Windows OS Enterprise or Datacenter edition boot.ini switch that employs Physical Address Extension capability of IA32 processor to recognize physical memory above 4GB; user-mode applications configured to reserve more physical memory (see AWE) will generate less paging, thus reducing disk activity; PAE is automatically invoked on systems with hot-add memory devices and does not need to be expressed in boot.ini
AWE	Address Windowing Extensions user-mode application APIs that allow programs to reserve more than 4GB of memory as non-paged; must be enabled in application (see Figure 2.3); SQL Server only takes advantage of AWE for data caching, not processing memory needs

Table 1.3: Memory options on 32-bit Windows OSs.

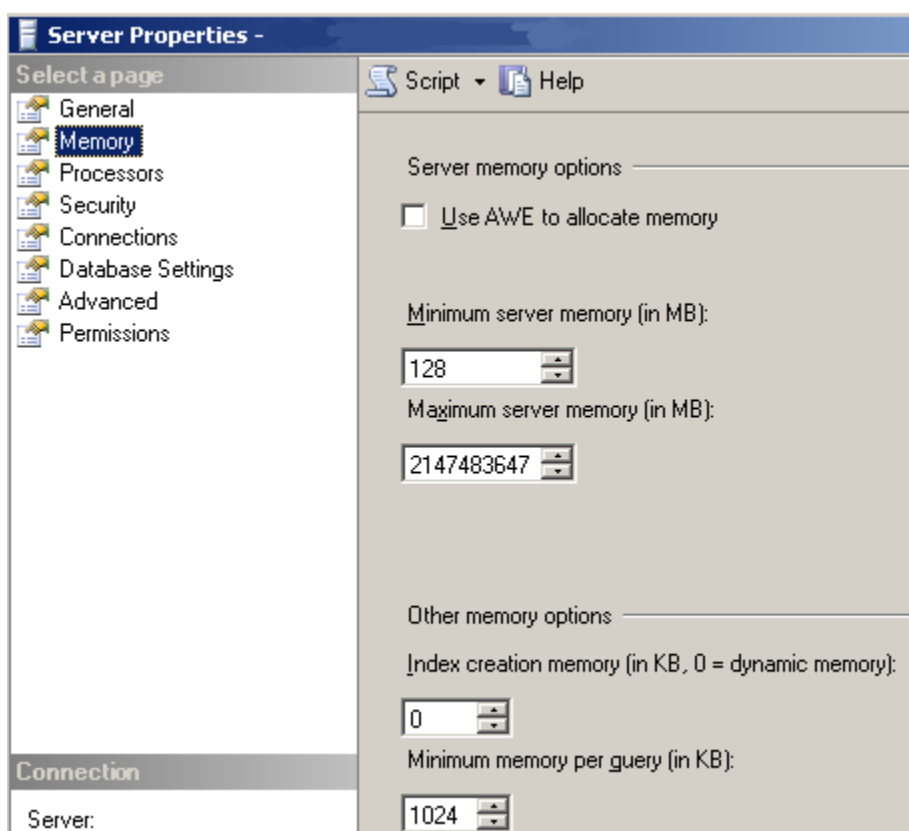


Figure 2.3: Setting AWE and other memory options on SQL Server.

SQL Server stores database table row and index bits onto disk volumes in 64kb “extents” comprised of eight 8KB “pages” (see Figure 2.4). A single row cannot span multiple pages, so no table row can be larger than 8KB. Large data objects are stored outside of the row’s data page yet within the database data file requiring only a small pointer in the row’s page while still contributing to overall database size.

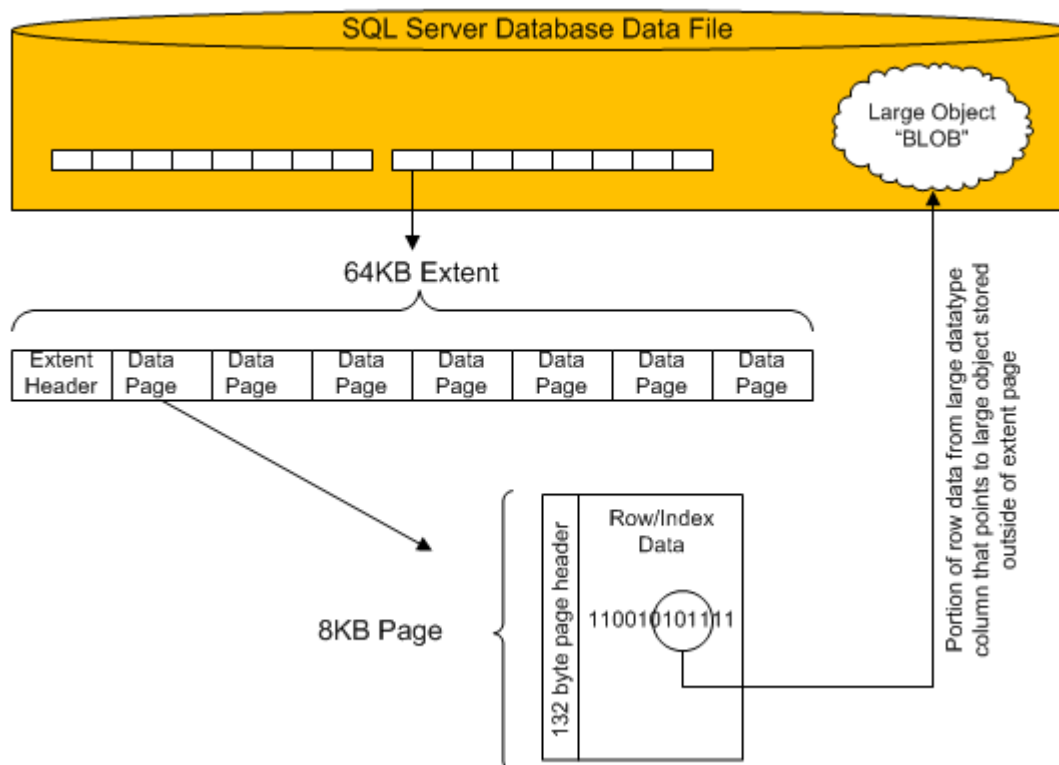


Figure 2.4: SQL Server data storage.

Note

Don't be confused by the second use of the word *page* in the previous discussion. Microsoft often recycles vocabulary for very different purposes and SQL data pages are not Windows OS memory pages.

Warning

Compression utilities such as those found in Windows NTFS or SQL Server 2008 don't really help SharePoint SQL Server database files. Why? Because the large object blobs do not compress and the only benefit is derived from non-library list items that compress only up to about 20%. Unless your SharePoint sites are hosting mostly non-library lists, compression isn't worth the overhead it generates on the server.

SQL Server employs write-ahead caching in the form of database log files to manage data manipulation. All transactions executed against database data are written to the database's log file(s) first before being committed into the data file itself (see Figure 2.5). A minimum of one log file must exist for every SQL Server database, so a single log file will be generated by default during database creation. Though log files only contain the changes being made to the data, they can still become quite large on an OLTP database. All SharePoint databases are set to Simple recovery model by default, which truncates log files automatically immediately following each checkpoint process (step 4 in Figure 2.5). Log files can be configured to automatically grow if necessary and should be monitored for disk utilization like their data file counterparts.

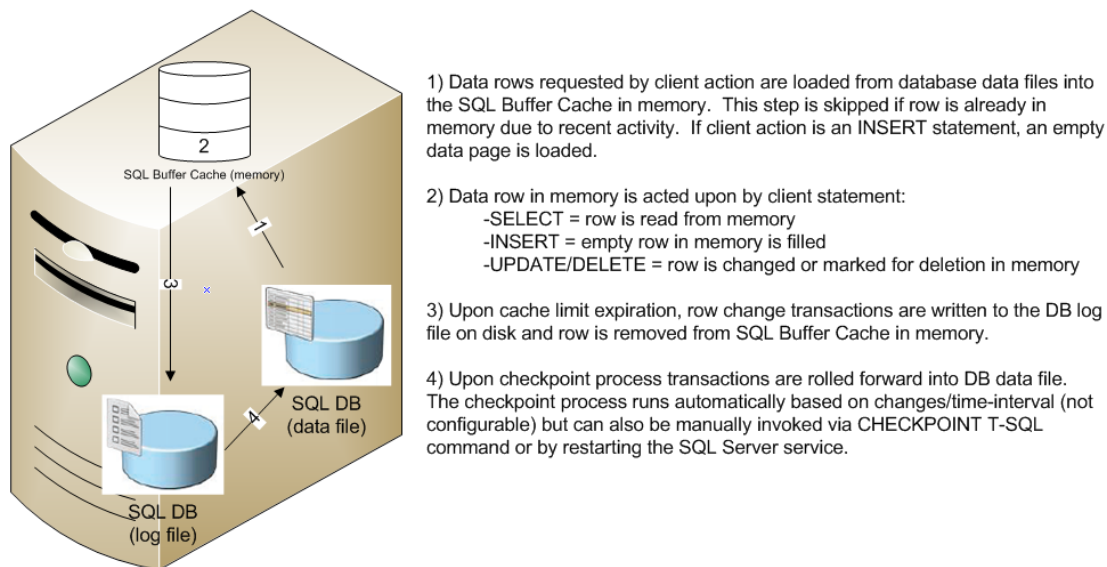


Figure 2.5: SQL Server data process.

Cross-Reference

A *transaction* is a body of work that must succeed in its entirety or not at all. SQL Server adheres to ISO standards for Atomic, Consistent, Isolated, Durable (ACID) database management. For more information about the ACID test and transactional behavior, research the <http://www.iso.org> Web site.

To determine whether SQL Server is putting excessive stress on the I/O subsystem, watch the SQLServer:BufferManager:Buffer Cache Hit Ratio counter in System Monitor. Ideally, this counter should maintain somewhere around 85 to 95 percent consistency. Buffer cache hits are experienced when client requests are successfully served from the buffer cache in memory (see Figure 2.5) rather than requiring hard drive access. A low cache hit ratio translates to more frequent disk access and can only be remedied by making more RAM available to SQL Server. Low cache hits can be temporarily relieved by configuring SQL Server to initially consume a larger memory footprint (see Figure 2.6) but ultimately more RAM will need to be installed or SQL Server will need to be moved onto a cluster solution.

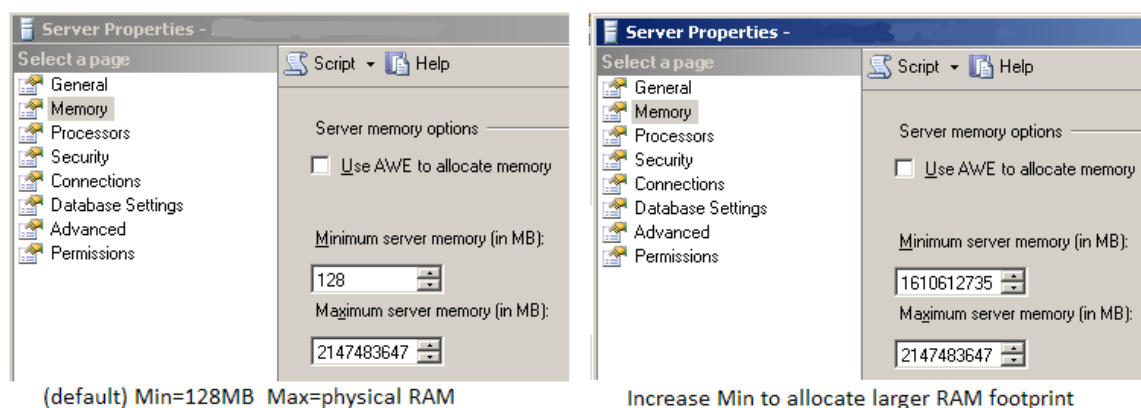


Figure 2.6: Dynamic Memory Management in SQL Server 2005.

A candidate for potential improvement on SQL Servers supporting SharePoint is the tempdb system database. This database provides temporary work space for T-SQL statements and hosts user objects deliberately built as temporary, internal objects, and version stores. The tempdb database has the following characteristics:

- Each SQL Server instance can have only one tempdb system database
- The tempdb is dynamically built anew at each MSSQLSERVER service start
- Tempdb has the same default location as the other system databases
- Initial data and log file sizes are derived from Model system database
- Default initial data file size is 2MB; default initial log file size is 1MB
- Default auto-growth of data and log files: 10% increments to *unlimited* size
- No user-defined filegroups (thus does not support table partitioning)
- Supports adding user-defined data and log files
- No CHECKSUM option, DBCC CHECKALLOC, or DBCC CHECKCATALOG

Managing disk space utilized by tempdb is crucial to the health of a SQL Server instance. You might set a maximum limit on autogrowth but beware—if tempdb runs out of space, the SQL instance is rendered unstable and multiple errors (such as 3958, 3959, 3966, and 3967) begin to appear in the Windows OS event log. Microsoft offers a bevy of System Monitor counters and SQL Server DMVs to assist with monitoring and correcting tempdb behavior (see Table 2.4). Precede System Monitor object names with the name of the SQL instance being analyzed and filter the counter instance to only the tempdb database for a more manageable result set. There is no way to deliberately separate internal objects from user objects or version stores without filegroups, so monitor object allocation as a whole in tempdb to determine which objects occupy the most space. From SharePoint, expect mostly internal objects and pacify tempdb by granting it the most disk space possible.

System Monitor Object or SQL DMV	System Monitor Counter	Description
Database	Log File(s) Size (KB)	Cumulative size of all log files (counts both used and free space in log files)
Database	Log File(s) Used Size (KB)	Cumulative used size of all log files (exposes density of log files)
Transactions	Free Space in tempdb (KB)	Total free space left in tempdb (useful for a Performance Alert to warn when value falls below acceptable KB value)
Transactions	Version Store Size (KB)	Cumulative size of both version stores (if it never shrinks, something is preventing version clean up process)
AccessMethods	Workfiles Created/Sec	Number of work files created per second (such as temp results of join)
AccessMethods	Worktables Created/Sec	Number of work tables created per second (such as temp results of XML variables, query spools, LOB variables or cursors)
GeneralStatistics	Temp Tables Creation Rate	Number of temporary tables and table variables (user objects) created per second
GeneralStatistics	Temp Tables For Destruction	Number of temporary tables and table variables waiting to be destroyed by the next cleanup system thread (if elevated something is preventing clean up process)
sys.dm_db_file_space_usage	n/a	Details space used by object type
sys.dm_db_session_space_usage	n/a	Identifies usurping session

Table 2.4: Tools for monitoring SQL Server tempdb system database.

Cross-Reference

The default tempdb location is the DATA folder of the SQL Server installation path. The default installation path is the C volume as is the default installation path of the Windows OS. This will cause major issues of tempdb exceeds available space! Consider reading the SQL Server Books Online article entitled “Moving Database Files” for instructions on moving tempdb to its own dedicated volume—perhaps even onto a SAN for better performance and scalability.

SharePoint Databases

If optimizing the OS, virtual memory, IIS, and the SQL instance fail to fill your idle hours, you are in luck because there is more! Recall from the previous chapter that within the SQL instance, the SharePoint installation process creates several databases, each with a specific purpose, that experience different activity. Correctly identifying the OLTP versus OLAP nature of each database and tweaking it for just such access can be the difference between mediocre and outstanding performance. Keep in mind that all SharePoint databases will be created using the Model system database for settings not supplied by the creation statements. This includes the default 1MB autogrowth increment for data files, which would be a good candidate for increasing immediately after installation.

A Caveat or Two

Volumes have been written about optimizing SQL Server 2005 databases...literally several massive technical books. A quick Internet search on the topic will validate this fact. However, this Shortcut Guide is not the appropriate time or place to wax philosophic on the details of how to distribute databases and partition tables or how to weight query plans for efficient execution of T-SQL statements.

Suffice it to say that any Microsoft-endorsed strategies for fine-tuning OLTP and OLAP databases should be tested in a lab before unleashing potential destruction on your production SharePoint collaboration system. For example, interrupting log backup procedures by manually creating well-intentioned maintenance plans for SharePoint databases can inadvertently compromise a SharePoint Web environment.

If you work in a highly segregated IT department where non-SharePoint DBAs will exercise full dominion over all databases on the production SQL Server, including the SharePoint databases, then please share these words of caution with them:

TEST SQL CHANGES FIRST!

Now if I can just find my way down off of my soapbox...

Managing SharePoint Configuration Databases and Logs

Windows SharePoint Services v3.0 creates several databases during installation, not the least of which is the `SharePoint_Config_{GUID}` database. This database holds all the configuration settings for the WSS product. WSS is a precedent for a Microsoft Office SharePoint Server 2007 install, so you can expect to see the `SharePoint_Config_{GUID}` database on MOSS servers as well. Additionally, the MOSS 2007 installation process will generate a second configuration database called `{SharedServiceName}_DB_{GUID}` to house configuration settings of the Shared Service organizational level. Both of these databases are heavily written to during initial build but thereafter are only heavily read from unless administrative or structural changes are made to SharePoint.

To properly manage the Shared Service configuration database, you must understand the role of the Shared Service Provider (SSP) in a MOSS environment. Shared Services are an additional abstraction layer of configuration independent of Web applications for MOSS features such as user profiles and audiences, enterprise search, excel services, and the business data catalog (see Figure 2.7). Every MOSS implementation must have at least one SSP, so the installation process will generate one named `SharedServices1` by default. Only consider creating additional SSPs if you have a large SharePoint environment wherein groups of Web applications need to be configured with differing setting values for any of the aforementioned features (such as Enterprise Search content sources). Upon adding subsequent SSPs, you can enlist Central Administration to associate a given Web Application with a specific SSP to gain its enterprise settings (again, see Figure 2.7). Each SharePoint Web Application can only be associated with one SSP at a time. Remember that creating additional SSPs creates additional databases in SQL Server thus requiring more disk space.

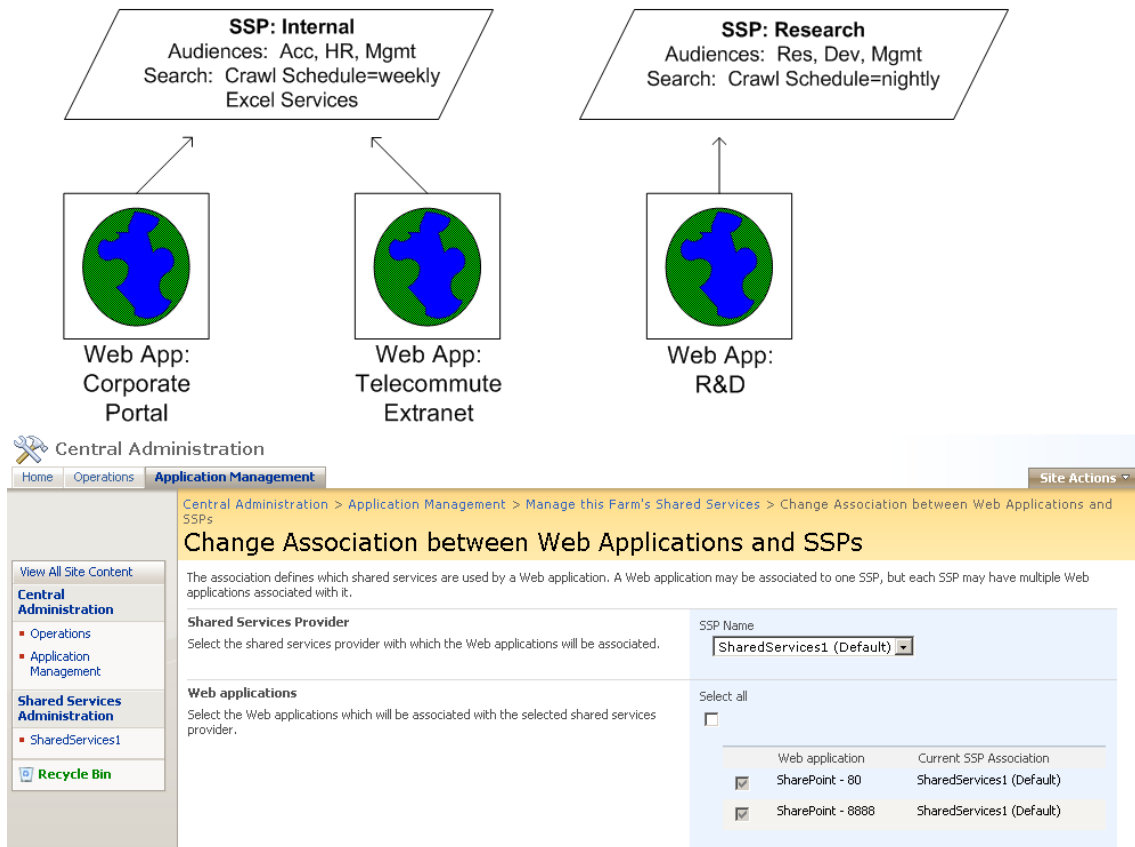


Figure 2.7: Web Application to SSP relationship and setting the associations in SharePoint Central Administration.

Optimize configuration databases by placing their data files on dedicated disk sets separate from the OS, paging files, log files, and SQL Server application files. Remember that enhancing configuration databases for transaction performance may be overkill for these two databases as they are not very busy with writing data during day-to-day SharePoint activity. If you disable log file autogrowth on configuration databases to conserve processor overhead, you should configure SQL Server Agent Service to monitor the SQL Server:Databases:Percent Log Used counter for each configuration database instance and run a job with error control that manually grows the log file in the event the counter breaches your desired threshold. Just because configuration databases suffer mostly read activity does not imply they are read-only databases; there will occasionally be writing activity that will generate log file content. Configuration databases are read most heavily during initial user login to a SharePoint site, upon consumer changes such as opening a document library item in a client-side application, at content upload, and each time the Timer service initiates a job. Of course, fault tolerance of configuration is paramount, so consider a RAID or SAN solution that offers disaster recovery along with fast read performance, such as RAID 1 mirror sets or RAID 5 stripe sets with parity. Placing the log file on a separate disk from the data files reduces disk contention.

Altering SQL Database Layout on SharePoint Databases

Many SQL Server optimization strategies suggest building or moving the data and log files of the databases to separate disks. Because SharePoint initially builds its own databases during installation, we human administrators have little control over the initial placement of the files. Moreover, tables within the SharePoint databases are immediately built and populated during installation, challenging deliberate distribution of database objects across multiple database files and thus multiple disks.

Moving database files to alternate disk locations should ideally be performed prior to unleashing SharePoint to users (while the files are small). SharePoint services should be stopped while moving database files to avoid corruption. After physically copying or moving the physical database file via a Windows OS tool, the logical change of file path on the database object in SQL Server must be accomplished using the following T-SQL statement:

```
-----
ALTER DATABASE {database name}
MODIFY FILE (NAME={file logical name},FILENAME={new path\physical file
name})
-----
```

New logical file paths take effect upon SQL Server service restart.

Managing SharePoint Content Databases and Logs

The most active of all SharePoint databases, the content databases of WSS and MOSS host what users see when visiting any non-administration pages in the collaboration environment. Everything from the master page that dictates the Web page borders to the Web parts that populate the Web page work frame to the actual items created or uploaded into a list or library reside in the content databases. Upon creation of the first SSP, a SharedServicesContent_{GUID} database is generated by default. Similarly upon creation of the first Web application, a WSS_Content database is generated automatically by default. Subsequent additional SSPs or Web applications created will generate their own content databases following default naming convention and appending a unique GUID as a suffix to the name. SQL Server limitations to keep in mind when planning SharePoint content databases are:

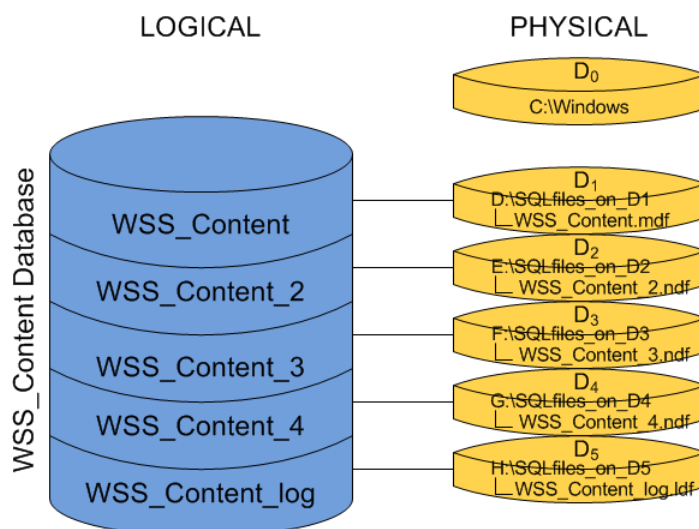
- Maximum database size = 1,048,516TB
- Maximum databases per SQL Server instance = 32,767
- Maximum filegroups per database = 32,767
- Maximum files per database = 32,767
- Maximum data file size = 16TB
- Maximum log file size = 2TB

First and foremost, the content database is highly normalized and very volatile, the epitome of an OLTP candidate. Therefore, it benefits from the most exclusive use of disk space possible and the most spindles available. Transfer rate is a significant priority for content databases, so purchase the fastest disk architecture you can afford and consider beefing up the cache on the disk controller hardware. Mature strategies using hardware RAID stripe sets to exploit multiple mechanical drives accessing data simultaneously still abound, but be aware of the performance overhead associated with processing the logical volume sets as well as the administrative demands of maintaining the RAID definitions. Alternatively, a well-built clustered SAN can offer the same multiple platter benefits while providing replication for fault tolerance. Within the SQL Server application itself, you might consider taking this *striping* idea a step further by implementing logical secondary data files to stretch each content database across multiple volumes (see Figure 2.8). Organizing multiple data files into separate filegroups enables partial database backups, online restores, table partitioning, and deliberate object placement.

Be aware, however, that existing data will automatically relocate across newly created data files or filegroups. Initially, all SharePoint data is placed into the only single default primary data file (.mdf) in the only single default PRIMARY filegroup. Until you move database objects (such as tables, views, and indexes) onto your new data files, SQL Server will not be taking full advantage of the additional disks.

Caution

Moving populated tables, indexes, and indexed views between data files in SQL Server 2005 is a complex high-level administrative process requiring table partitioning and table/view index re-creation. Even experienced SQL Server DBAs should consult SQL Server Books Online or Microsoft TechNet before attempting to redistribute data.



Logical File Name	Physical Path	Physical File Name
WSS_Content	D:\SQLfiles_on_D1	WSS_Content.mdf
WSS_Content_2	E:\SQLfiles_on_D2	WSS_Content_2.ndf
WSS_Content_3	F:\SQLfiles_on_D3	WSS_Content_3.ndf
WSS_Content_4	G:\SQLfiles_on_D4	WSS_Content_4.ndf
WSS_Content_log	H:\SQLfiles_on_D5	WSS_Content_log.ldf

Figure 2.8: Single WSS_Content database logically striped within SQL Server across five disks.

Content databases also benefit from being configured with multiple transaction log files (.ldf) located on separate disks. Preferably, these log files should not reside on the same disk as the data files of their database or risk causing disk contention. And as the log files are heavily written to during transaction processing from the SQL buffer cache and heavily read from during the checkpoint process, they probably should not share disk spindle with the Windows OS paging file either. The good news is that SQL Server 2005 has a maximum log file size of only 2TB, so your smallest disks can be reserved for log files (well, if you consider 2TB “small”).

Note

Don't fear the default unlimited autogrowth setting of the content database's log. By default, SharePoint databases are configured with the Simple recovery model that truncates log files upon checkpoint. This model keeps log files at a manageable size but prevents transaction log backups and thus restoring to the point of failure. However, do fear the default autogrowth increment on all SharePoint database logs. The increment is only 10% thanks to the default Model system database settings. To avoid debilitating overhead, increase the autogrowth increment to reduce frequency of growth processing. To alleviate the need for frequent autogrowth procedures, consider setting the size of the log file as large as the free disk space can afford initially and consider disabling autogrowth altogether.

Managing SharePoint Administration Databases and Logs

The only administration database created by default during SharePoint installation is the SharePoint_AdminContent_{GUID} database. Content data for the SharePoint 3.0 Central Administration Web application is stored in this database. The SharePoint 3.0 Central Administration Web application is the first Web application created during SharePoint installation whether installing only the free WSSv3 services or the entire MOSS2007 product. The IIS Web site for Central Administration is configured with a random port number to ensure security, although by default no SSL is required.

Unlike the WSS and SSP content databases, the SharePoint_AdminContent_{GUID} database is not a volatile database. In fact, Microsoft best practices discourage modifying any of the administration page content in Central Administration to avoid future SharePoint service packs or critical security patches setting the content back to factory defaults. Though you will see plenty of read activity against the administration database, very rarely will its transaction log even deviate from the initial size let alone grow significantly due to write activity on the database. Therefore, optimizing disk utilization by SharePoint_AdminContent_{GUID} follows the same theories as optimizing the SharePoint configuration databases.

SharePoint Search/Index and External Content

There is one last aspect of SharePoint databases left to examine: search and the full text index. Search is such a selling point for SharePoint that Microsoft split MOSS 2007 For Search software off to its own SKU for sale. One of the advantages SharePoint search offers is its ability to index data outside of the SharePoint enterprise. Thanks to custom content sources and the MOSS Enterprise Edition business data catalog you can configure SharePoint to index almost all data in your heterogeneous environment giving users a single SharePoint search site from which they can find anything in the realm. And even though the new Windows Search Server 2008 product rivals the enterprise searching capabilities of MOSS, this guide would be remiss if it did not give the search databases of SharePoint their due. So here we go.

Managing SharePoint Search Databases and Logs

Upon installation, SharePoint automatically generates two separate search databases. The WSS_Search_{hostname} database hosts user-defined search settings as well as the metadata and ACLs of all indexed content that will eventually make up search results. The {Shared Service Provider name}_Search_DB_{GUID} database hosts SSP-level search settings for content sources, crawl schedules and histories, and search reporting data. Of the two, the busier database is the WSS search database thanks to content changes and index crawls. The SSP search database is modified when the SharePoint administrators need to change the breadth of the enterprise that will be exposed to SharePoint search or the frequency of the data crawls to build new content source data into the SharePoint indexes.

Simply put, the WSS_Search_{hostname} search database needs to be on an isolated multi-spindle storage solution. Create multiple transaction logs for this database and store each on separate disks away from the database files. Any search environment is only as productive as its indexes are current, so you can anticipate aggressive crawling of new and updated content will be necessary to optimize SharePoint search success. All of this crawling will affect the I/O performance of the WSS_Search_{hostname} database. And keep an eye on the medium between the Index Server in a medium or large SharePoint farm and the SQL Server housing the data that will be crawled. The path should be fast and persistent.

Examining Full Text and Optional Metadata Indexes

Chapter 1 discussed Microsoft Deployment Guides that suggest indexing servers in a SharePoint farm will need 50% of the anticipated content database size for indexes. However, this capacity need can grow exponentially depending on what content sources are named and exactly how much data the index server gathers from each. Compound this with index maintenance concerns such as fragmentation and you might find yourself laying awake at night worrying about your SharePoint indexes.

Full text indexing will need to be enabled in WSSv3 if you plan to crawl all text content in the named search content sources. If it is used, SharePoint benefits from placing the full text index catalog of SQL Server on an isolated volume in a SAN or RAID design optimizing write activity. You might even lean on your storage provider for lightening fast hardware RAID with robust caching controllers. Every change to text-type metadata values as well as the text within documents that are stored in libraries will be among the information that must be written to the full text index. Indexing should be split onto its own SharePoint server in a medium farm architecture to isolate the full text index onto its own processors.

Engaging indexing on specific list and library columns increases indexing activity and disk space needs and can add to the full text index size dilemma if the column datatype assignment is text based. Use caution before indexing a column and be certain it will benefit user search behavior. You can determine the most popular search phrases from the MOSS 2007 Search Reports available in the Shared Service Provider administration site.

Interfacing External Content via SharePoint

The enterprise search capabilities of MOSS 2007 allow SharePoint administrators to add custom content sources that will provide search results of data from outside the SharePoint environment. The possible custom content source types are file shares, Microsoft Exchange public folders, Web addresses, or business data from the Business Data Catalog of MOSS 2007 Enterprise Edition. Before adding a custom content source, consider the additional storage space that will be required to accommodate the metadata of the external data in the SharePoint search databases.

Administrators may choose to add custom content sources associated with finite data sets within the SharePoint environment to fine tune the search database in preparation for designing custom search scopes. This practice allows for easier scope creation by non-administrative business users because they can build a scope by simply pointing it to an identifiable content source from a list instead of having to know the exact URL to the desired scope content. However, adding custom content sources that are essentially duplicates of the already existing entire SharePoint farm content source (albeit smaller individual sections of it) might cause undesired growth of the WSS_Search_{hostname} database.

Storage Options for SharePoint

Now that we have at least a basic idea of how much storage we might need for a successful SharePoint deployment, it's time to go hardware shopping! When deciding on storage architecture, remember that one size does not fit all and each component of a SharePoint environment may benefit from different storage options. Disks that reside within the local machine may best serve the Windows Server OS while SQL Server might perform most admirably on network interfaced storage. Regardless of the physical architecture you decide on, there are four criteria to always keep in mind when purchasing hard drives:

- **Spin Rate (RPM)**—Indicates the speed at which the disk platter revolves on the spindle. Higher RPM drives spin the platter to the location desired by the actuator more quickly but also generate more heat within the system.
- **Seek Time (ms)**—Average number of milliseconds it takes to spin the platter to the desired location and put the read heads of the actuator in position to read data from the drive. Short seek times less than 10 milliseconds translate to faster read performance from the drive.
- **Transfer Rate (MB/Sec)**—The megabytes per second transfer onto (write) or off of (read) the disk is the overall indicator of disk speed performance. Keep in mind that advertised transfer rates may be degraded by compromised disk controllers or poorly written applications.
- **Capacity (GB, TB, etc.)**—Available disk space for firmware, file system overhead, and actual data storage. Try to buy for the future by purchasing larger capacity drives than called for during deployment planning.

Internal Disks and DAS

Internal disks are those directly connected to the system board through a specific interface such as SATA or SCSI. The disks and their controllers are hidden from view because they are contained within the computer chassis (see Figure 2.9). Direct Access Storage (DAS) devices are portable and connect to the system board via an external interface such as USB or FireWire. Still considered a bus device, these DAS units are bound by the same limitations as internal disks, such as bus speed and circuit congestion. Both disk types are very easy to purchase and set up in a Windows OS thanks to the plug-and-play nature of DAS driver sets.

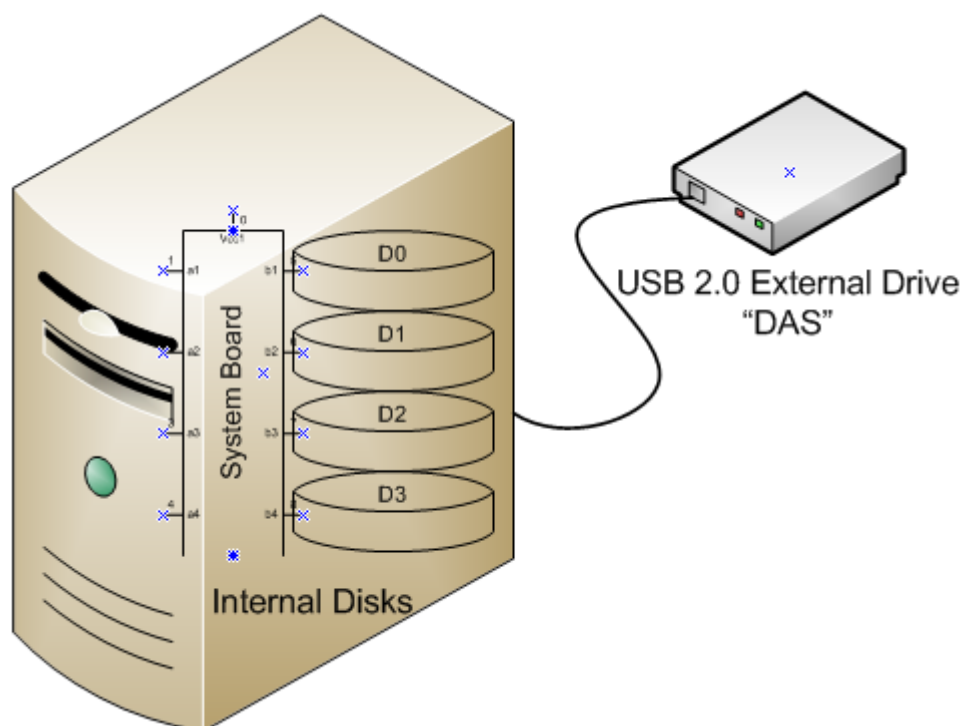


Figure 2.9: Internal disks and DAS.

Both internal disks and DAS devices can be made members of software RAID sets devised in Windows Server 2003 or 2008. However, consider the ease with which DAS units can be disconnected before using them in a RAID design because removing them from the computer could cause disk set failure (RAID 0), redirection (RAID 1), or regeneration (RAID 5). Table 2.5 outlines the advantages and disadvantages of internal and DAS disks.

Note

When it comes to implementing RAID, this author advises choosing hardware RAID from your storage provider instead of the Windows OS software RAID offerings. Software RAID, whether involving internal disks, DAS, or network accessible disks, performs slower than hardware RAID and leaves your system vulnerable to OS failures resulting in data unavailability despite your fault-tolerant RAID design! Of course, these performance and stability concerns could be eliminated by implementing a multiple-disk clustered SAN with replication instead of RAID.

NAS and SAN Strategies

NAS and SANs are often confused by the similarities of their acronyms and the fact that both represent storage remote from servers and across the network. But in reality, NAS and SAN storage architectures operate very differently from one another. NAS solutions use network-connected disk vaults to provide a distant locale for servers to place files onto. They are file system-based and must be accessed via protocols such as Windows SMB over TCP/IP on an Ethernet network. Several platforms can utilize NAS solutions by different protocols (Novell NCP, IBM LANManager) but it was Sun Microsystems who first coined the phrase “Network File System” for their NFS protocol and the label stuck to become an IT industry accepted nickname for any NAS storage facility. NAS devices became popular for expanding simple file storage beyond the restrictions of internal disk but are limited by their protocol-based access. For instance, the Windows OS treats NAS destinations as logical drive letters; however, their remoteness from the server prevents many Microsoft network applications (including SQL Server) from being able to use them because they are networked drives.

SAN solutions, in contrast, use mass storage protocols to transfer data from the initiator server to the target SAN LUN and back (see Figure 2.10). A Logical Unit Number (LUN) is a logical division of the SAN physical disks that can be accessed by the host server. In addition to providing working space, LUNs make granular management possible, such as security settings and quota limits, if such tools are available from your storage provider. And although the SAN disk LUNs may actually be formatted in, say, NTFS (from Microsoft Windows), the server does not see the destination SAN LUN as anything but an additional local drive thanks to the block-level transport protocols. Therefore, all Microsoft network applications can take advantage of SAN storage! The most common storage interfaces and mass storage protocols are:

Storage Interfaces

- **Parallel SCSI**—An older topology requiring a correctly terminated SCSI controller in the management server, this protocol has reached its End of Technology (EOT) and is a candidate for upfit to another protocol
- **Serial Attached SCSI (SAS)**—Replacement for Parallel SCSI that offers nearline connectivity to both SCSI and SATA disk sets; SAS can be enhanced with Fibre Channel or iSCSI interfaces on the front end of the array to provide fast connectivity to a management server(s)

Mass Storage Protocols

- **Fibre Channel**—Praised for its speed over long distances, can use twisted pair copper or fiber-optic medium to transport SCSI commands over gigabit Ethernet networks via the Fibre Channel Protocol (FCP)
- **iSCSI**—Standardized in 2004 (RFC 3720), the iSCSI protocol operates between TCP and SCSI to layer translation instructions for the actual TCP/IP packet that will traverse the medium between the initiator server and the target SAN LUN; more on iSCSI will appear in the next chapter of this guide

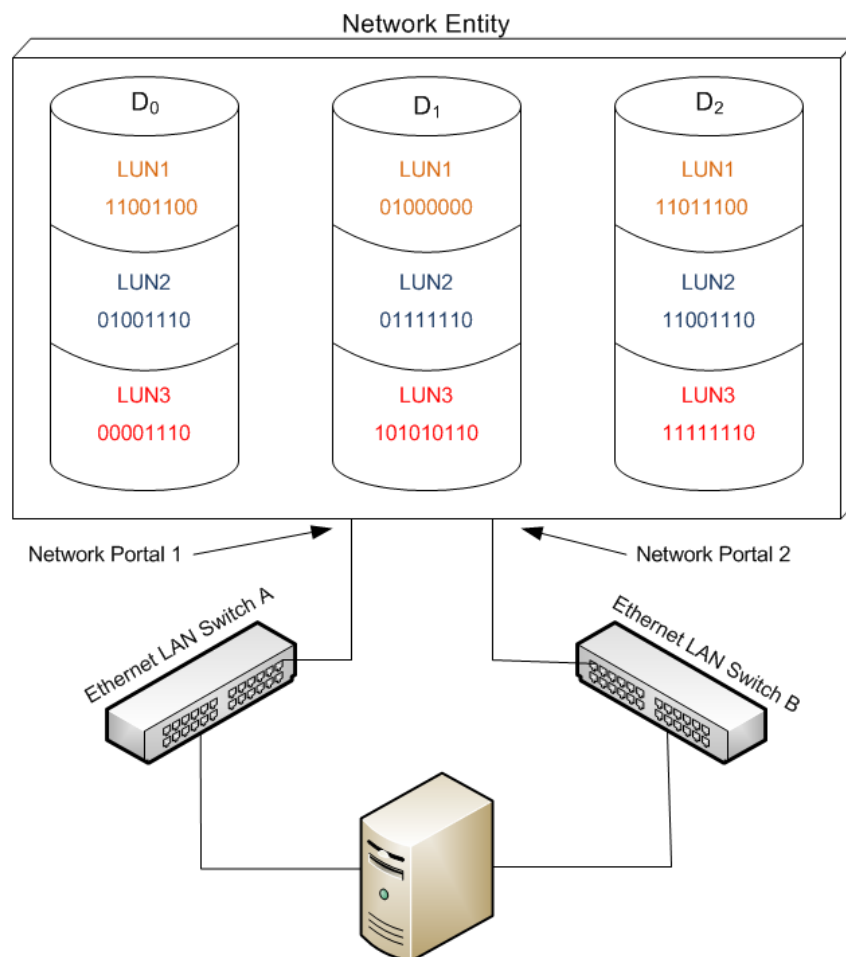


Figure 2.10: Typical SAN storage layout with dual switch for fault tolerance.

How to Choose Appropriate Storage

When it comes to choosing the appropriate storage architecture for your SharePoint environment, you must consider cost and expertise currently on staff. Additional concerns such as data availability requirements and disaster recovery strategies will also influence your choice of storage, especially if your IT department must adhere to a strict Service Level Agreements (SLAs). The OS component of SharePoint can live on either internal disks or on a SAN if the storage provider offers boot on SAN capabilities. Paging files for the Windows OS can reside anywhere but the IIS services will be hosted within the Windows System 32 directory and therefore will reside where the OS was installed. SQL Server performs best on a scaled SAN but could work on the internal disks of a small business server. DAS and NAS options are not recommended for SQL Server.

Consider carefully the advantages and disadvantages of each storage solution architecture before purchase. Your primary concern should not be initial purchase price but rather the total cost of maintaining the system. Internal storage is cheap, but when your collaborative SharePoint enterprise overstates its welcome on your limited disk space (see previous chapter of this guide), reconfiguring or adding to internal disk configurations is time consuming and disruptive to user connections. Purchase for flexibility and scalability, two of the most important and inevitable concerns of an organic system. Table 2.5 outlines the advantages and disadvantages of internal disks, DAS, NAS, and SAN solutions as a quick reference for choosing the appropriate storage.

Note

This author recommends a flexible, bootable SAN solution for the ultimate in performance, fault tolerance, disaster recovery, and manageability of OS, SQL Server, and SharePoint files.

Type	Advantages	Disadvantages
Internal disks and DAS	Inexpensive	Inflexible (reconfiguration is disruptive)
	Internally managed	Not scalable (number of disks is limited by hardware)
	Simple installation	System board and OS issues that may not be disk related can still affect data availability
	Simple management	Few disaster recovery options
NAS	Scales beyond internal disk and DAS limits	Inflexible (reconfiguration is disruptive)
	Lower cost than SAN	Slow at handling large files
	More disaster recovery options than internal/DAS	Requires file-based protocols and offers little or no local management
SAN	Divisible (LUNs)	Expensive
	Flexible (reconfiguration need not impact servers)	Higher TCO compared with internal/DAS notwithstanding scaling.
	Recognized by front servers as local drives	Might require service contract if IT expertise is insufficient to maintain SAN LUNs
	Fast and redundant	
	Sophisticated high-availability/disaster recovery options.	

Table 2.5: Advantages and disadvantages of different storage architectures.

Summary

This chapter examined the components of a SharePoint environment and their individual storage needs and behaviors. From managing memory in the OS to optimizing IIS performance to controlling the size of SQL Server databases unique to SharePoint, there are several facets to administering a SharePoint server. Several tips and tricks for software optimization beyond storage concerns, such as placement of database files and configuration of database options, were also introduced to assist you in overall server performance tuning.

The next chapter will dig deep into one of the most popular SAN architectures on the market today, the iSCSI mass storage protocol solution. In the final chapter, we will examine disaster recovery and high-availability options for your SharePoint environment and maintenance issues that arise over the life span of a SharePoint server.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.