Managing Multi-Terabyte Content Databases with Microsoft® SharePoint® 2010

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Managing Multi-Terabyte Content Databases with Microsoft SharePoint 2010

Bill Baer
Microsoft Corporation

June 2011

Applies to: Microsoft® SharePoint® Foundation 2010, Microsoft SharePoint Server 2010, Microsoft SQL Server® 2008, Windows Server® 2008

Summary: Managing large content databases in SharePoint 2010 requires careful planning and consideration of capacity management, performance, and data protection. This white paper provides guidance to support these considerations.

Contents

Introduction

The space required for unstructured content (**B**inary **L**arge **Ob**jects) hosted within a SharePoint 2010 environment can increase over time as the result of changing business requirements, expanding the scope of the service, or as additional workloads are introduced. As the unstructured content grows, the strategies and planning for dealing with that content will change.

Managing large content databases with SharePoint 2010 requires careful planning and consideration, including capacity management, performance, and data protection. This white paper provides an overview of managing content databases at several stages, with each stage representing a range of sizing considerations and some of the requirements associated with that range. The guidance in this white paper is cumulative and therefore should be read in its entirety.

For all ranges, the size includes both structured data (that is, metadata) stored and unstructured data (that is, BLOBs) hosted both inline and/or externalized.

Database Sizing Considerations

Content Databases Less Than 200 GB

Content databases up to 200 GB benefit from their flexibility in that they can support rapid backup and recovery due to their relatively small size. Content databases up to 200 GB are suitable for all workloads such as Collaboration, personalization sites, Document and Records Management and Business Intelligence.

Content database up to 200 GB benefit from lower overall per volume I/O requirements as the I/O can be distributed across a number of drives and their size more effectively supporting RAID implementations such as RAID 1+0 where performance and lower storage costs are requirements. For example, the baseline SharePoint 2010 I/O requirement of 0.25 I/O per sec. per GB of content stored results in less overall I/O performance required, providing access to a greater array of disk architectures such as smaller RAID 1+0 arrays or RAID 5 arrays where the additional resiliency of RAID 1+0 is not required.

NOTE

The minimum recommend I/O/sec. required for Microsoft SharePoint 2010 content databases is 0.25 I/O per sec. per GB content stored, optimal performance requires a minimum 2 I/O per sec. per GB stored.

Content databases up to 200 GB will require no addition considerations beyond those categorized above or within existing documentation in the TechNet library.

Content Databases between 200 GB and 4.0 TB

Content databases exceeding 200 GB and up to 4 TB support much of the same flexibility as content databases <200 GB. However, additional planning should be considered in order to effectively manage those databases in respect to operational and hardware considerations.

You should upgrade to Service Pack 1 when supporting content databases up to 4 TB. Capabilities such as the Site Recycle Bin, shallow copy, and StorMan.aspx will reduce management overhead and provide greater insight into storage allocation within site collections hosted in the database.

4 TB represents the largest supported content database size for all workloads with the exception of Document and Record Center implementations where the content database can exceed 4 TB. In scenarios where the 4 TB threshold is expected to be exceeded, the following requirements and recommendations should be considered:

1. Archive content into Document Center archives using Content Organizer. For additional information about Content Organizer and metadata routing, see Metadata-based routing and storage overview (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/ff608093.aspx).
2. Redistribute site collections across content databases. For additional information about redistributing site collections across content databases see Move site collections between databases (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/cc825328.aspx).

The planning section of this document is designed to provide guidance when considering content databases exceeding 200 GB. This guidance, however, should not be considered to be an exhaustive list of all recommendations and best practices associated with working with data sets of this size.

Content Databases Exceeding 4 TB

SharePoint 2010 supports content databases that exceed 16 TB. However, this support is subject to several limitations and constraints as documented herein, such as:

1. The content database must be distributed across multiple data files. SQL Server maximum capacity for a single data file is 16 TB.
2. Site collections hosted within content databases exceeding 16 TB are supported only when using either the Document or Record Center templates.
3. Site collections hosted within content databases exceeding 16 TB are supported only where less than 5% of the content is actively read and less than 1% actively written.
4. Alerts, Workflow (with the exception of flows related to Content Organizer (http://msdn.microsoft.com/en-us/library/ee558288.aspx)), Link Fix-up, and Item Level Security are not supported on content databases exceeding 16 TB.

The planning section of this document is designed to provide guidance when considering content databases exceeding 200 GB. This guidance, however, should not be considered to be an exhaustive list of all recommendations and best practices associated with working with data sets of this size.

Planning

Capacity Management

When considering multi-terabyte content databases with SharePoint 2010 is it important to understand the limitations and constraints when working within the boundaries their schema imposes.

SharePoint 2010 content databases do not support partitioning. Therefore, when creating content databases you are limited to a single (primary) filegroup containing one or more data and log files. Due to these limitations, many SQL Server features designed to support multi-terabyte databases cannot be supported with SharePoint 2010 content databases such as partial database availability. When planning multi-terabyte content databases, you should consider distributing the database across multiple data files, allocating each data file to a separate physical disk to optimize I/O distribution. As a best practice, you should map the number of data files to the number of available cores to the instance of SQL Server supporting those databases. For additional information about Database Files, see Physical Database Files and Filegroups (http://msdn.microsoft.com/en-us/library/ms179316(SQL.90).aspx).

For additional information about SQL Server capacity planning with SharePoint 2010 Products, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/cc298801.aspx).

When considering multi-terabyte databases, you should not only consider the impact on storage capacity for hosting the database, but also consider the impact of storing backup sets and storage requirements associated with technologies such as Log Shipping (such as space allocation for Transaction Log Backups).

Log Shipping

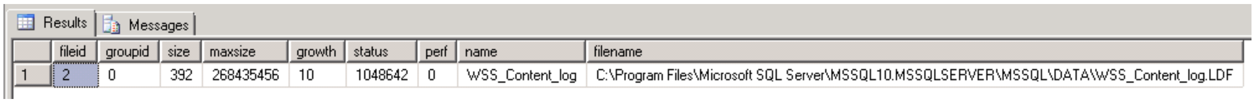
Using Transact-SQL you can measure and predict the storage requirements for logs with the following sample statement for each content database:

```
USE [Database Name]
GO
SELECT * FROM sysfiles WHERE name LIKE '%LOG%'
GO
```

The result of executing the Transact-SQL statement above is presented in Figure 1 sys.sysfiles (Transact-SQL) below.

The **size** value represents the current size of the Transaction Log in 8-KB pages, the **maxsize** value represents the maximum file size of the Transaction Log in 8-KB pages, for example 268435456 indicates the log file will grow to a maximum size of 4 TB. (see Figure 1 sys.sysfiles (Transact-SQL)).

Figure 1. sys.sysfiles (Transact-SQL)



	fileid	groupid	size	maxsize	growth	status	perf	name	filename
1	2	0	392	268435456	10	1048642	0	WSS_Content_log	C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\WSS_Content_log.LDF

NOTE

The amount of storage required is a product of usage patterns, amount of transaction log generated, and backup retention configuration. The recommendation above is not a replacement for careful planning and monitoring.

Backup and Restore

Consider a common SLA of maintaining up to three weeks of data on disk for rapid recovery, prior to the transition of data to tape. This scenario would require at least three full and 18 differential backup sets, effectively requiring 3+ x the amount of content storage. For example, a 16 TB database would require 48+ TB of available backup storage to accommodate this scenario.

Data Protection

As the size of the database increases, the overall time to back up and recover that database increases as well. Prior to planning for large databases, you should consider this impact on SLAs related to Recovery Point Objectives, and more critically Recovery Time Objectives.

If your system exceeds the limits outlined below, or if backing up your system exceeds the maintenance window that you have available, we recommend that you consider using alternative backup and recovery tools, such as SQL Server 2008 or System Center Data Protection Manager.

Limits

- Content databases larger than 4 TB.
- Site collections that are larger than 15 GB that you want to back up by using Windows PowerShell, or the SharePoint Object Model.

NOTE

Site collections larger than 15 GB up to 100 GB should be isolated within a single content database to facilitate efficient backup and recovery. In the event a site collection between 15 GB and 100 GB is contained within a content database with smaller site collections, it may be necessary to redistribute the smaller site collections across other content databases in order to isolate the larger site collection. These operations can be performed using the Move-SPSite cmdlet. For additional information about the Move-SPSite cmdlet, see Move-SPSite (http://technet.microsoft.com/en-us/library/ff607915.aspx).

SQL Server 2008 includes capabilities and features designed to support making it easier to work with multi-terabyte databases that can be implemented to improve the performance of backup operations.

SQL Server backup operations are I/O intensive. Backup compression writes less pages to disk compared to uncompressed backup. As long as the system is not bottlenecked on CPU, backup compression should execute faster than backup without compression.

Backup Compression

Backup compression was introduced in SQL Server 2008 Enterprise. Beginning in SQL Server 2008 R2, backup compression is supported by SQL Server 2008 R2 Standard and all higher editions. Every edition of SQL Server 2008 and later can restore a compressed backup.

NOTE

Monitor the output of backup operations and note the performance when planning multi-terabyte databases to ensure backups can be completed within a standard maintenance window or the impact of running backup operations does not significantly impact performance on 24x7 systems. The following example illustrates the output of a backup operation.

```
Processed 3512 pages for database 'WSS_Content', file 'WSS_Content' on file 1.
Processed 2 pages for database 'WSS_Content', file 'WSS_Content_log' on file 1.
BACKUP DATABASE successfully processed 3514 pages in 1.021 seconds (26.887 MB/sec)
```

Backup compression requires less device I/O due to the smaller size of the backup when compared to an uncompressed backup. However, it increases CPU usage that can potentially impact other concurrent operations. For collecting data about processor activity, monitor the counters that are listed in Table 1. The % Processor Time counter is the primary indicator of processor activity, displaying the average percentage of busy time observed during the sample interval.

Table 1. Recommended Processor Activity Counters

Counter	Description
System\Processor Queue Length	The number of threads in the processor queue. Shows ready threads only, not threads that are running. Even multiprocessor computers have a single queue for processor time. Thus, for multiprocessors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than two threads per processor is normally acceptable, depending upon the workload.
Processor (_Total)\% Processor Time	The average percentage of time that all processors are active. This counter is the primary indicator of average processor activity. This value is derived by calculating the percentage of time during the sample interval that all processors spent in executing their idle thread (which consumes cycles on a processor when no other threads are ready to run), and then subtracting the result from 100 percent. Values exceeding 65% consistently should be evaluated to determine whether a processor bottleneck exceeds.

Processor\% Processor Time	The average percentage of processor use for each processor (#0, #1, and so on).
Processor\% Privileged Time	The average percentage of processor time that is spent in privileged mode. Values exceeding 75% consistently indicate a processor bottleneck.

To measure the amount of compression that can be realized on one or more databases, use the information from the backup_size and compressed_backup_size columns in the backupset history table for a database that has at least on prior backup set.

Use the SELECT statement to query the backupset history table:

```
SELECT backup_size/compressed_backup_size FROM msdb..backupset;
```

Backup compression can be enabled through Transact-SQL or the SQL Server Management Studio. To configure backup compression through Transact-SQL, see backup compression default Option (http://technet.microsoft.com/en-us/library/bb677250.aspx). To configure backup compression through SQL Server Management Studio, see How to: View or Change the backup compression default Option (SQL Server Management Studio) (http://technet.microsoft.com/en-us/library/bb933863.aspx).

Consider performing backup operations during periods of low utilization or within scheduled maintenance windows where the impact on users is during periods of low activity. Optionally, low-priority compressed backups are recommended in sessions where CPU usage is limited through Resource Governor. For information about configuring Resource Governor to limit CPU usage, see the "Configuring Resource Governor to Limit CPU Usage" section of "How to: Use Resource Governor to Limit CPU Usage by Backup Compression (Transact-SQL)" (http://technet.microsoft.com/en-us/library/cc280384.aspx#configure_RG).

An important step in understanding backup I/O requirements is monitoring for the impact these operations will have and their unique characteristics through isolating and monitoring backup I/O through monitoring performance counters in the following table.

Table 2 Recommended Windows I/O Counters

Counter	Description
LogicalDisk|PhysicalDisk\Avg. Disk sec/Transfer	Indicates how fast data is being moved (in seconds). Measures the average time of each data transfer, regardless of the number of bytes read or written. Shows the total time of the read or write, from the moment it leaves the Diskperf.sys driver to the moment it is complete.

	This value should generally be less than 10ms. A high value for this counter might mean that the system is retrying requests due to lengthy queuing or, less commonly, disk failures. To determine the percentage of disk access time spent paging (indicating physical memory is a bottleneck), multiply the result of Physical Disk\Avg. Disk sec/Transfer and Memory\Pages/sec counters. Results below 10% are acceptable. To analyze transfer data further, use Avg. Disk sec/Read and Avg. Disk sec/Write.
LogicalDisk|PhysicalDisk\Avg. Disk Queue Length	Tracks the number of requests that are queued and waiting for a disk during the sample interval, as well as requests in service. As a result, this might overstate activity. The value should not exceed the number of spindles x 2; otherwise disk may be a bottleneck. To analyze queue length data further, use Avg. Disk Read Queue Length and Avg. Disk Write Queue Length.
LogicalDisk|PhysicalDisk\Disk Bytes/sec	Indicates the rate at which bytes are transferred and is the primary measure of disk throughput. To analyze transfer data based on reads and writes, use Disk Read Bytes/sec and Disk Write Bytes/sec, respectively. Disk/Read Bytes/sec and Disk Write Bytes/sec should not exceed 85% of disk capacity. Consult the disk vendor to determine capacity. The result of Disk/Read Bytes/sec and Disk Write Bytes/sec will vary depending on the RAID implementation. Use the following formulas to accurately measure the results of these counters: RAID 1+0

	I/O per Disk = (reads + (2 x writes)) / [number of physical disks in array] RAID 5 I/O per Disk = (reads + (4 x writes)) / [number of physical disks in array]
LogicalDisk|PhysicalDisk\Disk Transfers/sec	Indicates the number of read and writes completed per second, regardless of how much data they involve. Measures disk utilization. If value exceeds 50 (per physical disk in the case of a striped volume), then a bottleneck might be developing. To analyze transfer data based on reads and writes, use Disk Read/sec and Disk Writes/sec, respectively.
LogicalDisk|PhysicalDisk\% Disk Time	Reports the percentage of time that the selected disk drive is busy servicing read or write requests. Because this counters data can span more than one sample, and consequently overstate disk utilization, compare this value against % Idle Time for a more accurate picture. Values greater than 50% indicate a disk bottleneck.
LogicalDisk|PhysicalDisk\% Idle Time	Reports the percentage of time that the disk system was not processing requests and no work was queued. Notice that this counter, when added to % Disk Time, might not equal 100 percent, because % Disk Time can exaggerate disk utilization.

Table 3. Recommended SQL Server Backup Counters

Counter	Description
SQL Server:Backup Device | Device Throughput Bytes/sec	Throughput of read and write operations (in bytes per second) for a backup device used when backing up or restoring databases. This counter exists only while the backup or restore

	operation is executing.
SQL Server:Databases | Backup/Restore Throughput/sec	Read/write throughput for backup and restore operations of a database per second. For example, you can measure how the performance of the database backup operation changes when more backup devices are used in parallel or when faster devices are used. Throughput of a database backup or restore operation allows you to determine the progress and performance of your backup and restore operations.

Network Backup

Backing up over the network can be a time intensive operation. You should consider at least the following network optimizations when backing up over the network.

Jumbo Frames

Jumbo Frame support enhances Ethernet throughput and reduces CPU utilization when transferring large files through enabling more efficient larger payloads per packet. Sending larger payloads per packet results in fewer packets routed, thereby reducing CPU utilization and improving network throughput.

Jumbo Frames are Ethernet frames with more than 1500 bytes of payload, data packets on an Ethernet link comprise a frame. Maximum efficiency can be obtained using the maximum allowed payload size (9014 bytes).

Buffer Count

Buffer Count refers to I/O buffers to be used for the backup or restore operation. Increasing Buffer Count can help improve the overall performance of backup and restore operations. A recommended value of 512 should be used when considering large multi-terabyte databases.

NOTE

Large numbers of buffers may result in out of memory errors due to insufficient virtual address space in the SQLSERVER.EXE process.

Configuring the BUFFERCOUNT database option should be carefully planned and configured as the result of testing under the conditions the represent those where BUFFERCOUNT will be adjusted. BUFFERCOUNT is determined by the number of backup devices and database volumes. To determine what the current BUFFERCOUNT is for your environment, you will need to run a sample backup set with trace flags 3605 and 3213 configured.

Use the following DBCC statement to set the TRACEON database options:

```
DBCC TRACEON/TRACEOFF (3605)
DBCC TRACEON/TRACEOFF (3213)
```

- FlagNo 3605 sends trace output to the error log.
- FlagNo 3213 traces SQL Server activity during backup process.

Running a sample backup set with trace flags 3605 and 3213 configured will result in output to the ERRORLOG containing the number of buffers used for backup operations.

For additional information about configuring BUFFERCOUNT, see Data Transfer Options in the BACKUP Transact-SQL statement documentation at BACKUP (Transact-SQL) (http://msdn.microsoft.com/en-us/library/ms186865.aspx), or the Advanced BACKUP and RESTORE Options SQL Server Magazine article at http://www.sqlmag.com/article/database-backup-and-recovery/advanced-backup-and-restore-options-129834.

File Distribution

Distributing the database across multiple data files can increase the performance of write, backup, and recovery operations through isolating and distributing I/O to disks on which data files reside in addition to enabling the distribution of files across separate network paths during backup operations over the network—for example, passing two data files across each Network Interface Card installed on the server. The maximum capacity specification for a single data file is 16 TB.

Backup Server

In a multi-node cluster environment, each cluster node can have backup server capabilities which mean that they can take ownership of resources. This enables the offloading of the I/O intensive operation to a node that does not directly serve the environment, thus mitigating the performance impact of these operations.

Performance

The density of a database determined by the number of rows can result in increased I/O due to query processing and other actions to include property promotion/demotion, workflow, upgrade sequences, or operations through which database objects are queries and/or iterated. When planning your information architecture, you should design your information layout to accommodate the limiting of density through distributing objects such as site collections and sites across multiple databases, or optionally, distributing libraries and lists across site collections and sites. A properly planned information architecture that maintains itself within the documented boundaries of 30,000,000 items will minimize row and table locks, improve the upgrade experience, and reduce wait times for operations to complete, whether initiated indirectly as the result of timer job definitions on the server farm or directly through end user interaction.

RAID

As the size of a content database increases, the number of operations that include read and write operations performed can increase. In order to support these operations, you should consider write-optimizing the array or arrays where the data files that comprise the database reside.

A RAID 1+0 set is recommended when considering multi-terabyte databases to provide the performance benefits associated with disk striping in addition to the redundancy benefits of mirroring. RAID 1+0 sets provide the highest read and write performance when compared to other RAID levels such as RAID 5. However, that performance will be at the expense of overall storage capacity and cost per GB.

NOTE

System RAID can be used, though is not recommended. When considering RAID, it is recommended to implement a hardware-based solution.

RAID 5 can be considered in Document Archive and Records Management scenarios where data is frequently read and infrequently written.

Measuring I/O

I/O requirements can change over time due to changes in user concurrency, content propagation, and workload characteristics. Understanding which activities and components generate I/O is an important consideration when planning multi-terabyte content databases, and how to configure your storage to support them requires calculating disk I/O requirements for your users. Calculating your disk I/O requirements allows you to optimize your disk subsystem to best support your users.

NOTE

The minimum recommend I/O/sec. required for Microsoft SharePoint 2010 content databases is 0.25 I/O per sec. per GB content stored. Optimal performance requires a minimum 2 I/O per sec. per GB stored.

To determine the relative size of a content database, you can use the following formula:

```
Database Size=((([Number of Documents x [Number of Versions])) x [Size of Documents in KB]) + ((10KB x ([Number of List Items + ([Number of Documents] x 2)))
```

To determine the amount of I/O available to a logical disk, you can use the values of various performance counters as follows:

$$I/O = {}^{(\text{Logical Disk: Avg. Disk Bytes/Read} + (2 \text{ x Logical Disk: Avg. Disk sec/Write}))}/_{2}$$

For additional performance best practices, see Predeployment I/O Best Practices at http://technet.microsoft.com/en-us/library/cc966412.aspx.

Maintenance

As the size of a database increases, you will need to carefully evaluate both the impact of maintenance operations and how those operations will be performed to mitigate downtime, interruption, and performance penalties associated with those operations.

A common maintenance plan includes consistency checks, which are designed to check logical and physical integrity of all objects in one or more databases.

Database consistency checks on multi-terabyte databases can result in lengthy operations that cannot be accomplished within a standard maintenance window. In such scenarios you will want to reduce the need for frequent consistency checks. The following recommendations can help reduce the need for frequent consistency checks although these recommendations do not mitigate the need for consistency checks.

Set the PAGE_VERIFY Database Option to CHECKSUM

CHECKSUM provides a high-level of data-file integrity checking. When the PAGE_VERIFY database option is set to CHECKSUM, SQL Server calculates a checksum over the contents of the whole page and stores the value in the page header when a page is written to disk. When the page is read from disk, the checksum is recomputed and compared to the checksum value that is stored in the page header.

Use the ALTER DATABASE statement to set the PAGE_VERIFY database option to CHECKSUM:

```
ALTER DATABASE [DatabaseName] SET PAGE_VERIFY CHECKSUM
```

Backup Databases with CHECKSUM

CHECKSUM ensures when pages are written to backup media, BACKUP has verified the page providing a high degree of confidence in the integrity of your backup sets. Restore operations can also use the backup checksum to validate that the backup is not corrupt.

Use the BACKUP DATABASE statement to enable backup checksum:

```
BACKUP DATABASE [DatabaseName] TO [Option] = ‘PATH’ WITH CHECKSUM
```

NOTE

Backup checksum is the enabled by default when using backup compression.

DBCC CHECKDB

DBCC should be implemented only where the operation can be completed within a standard maintenance window. As the size of the database increases, the duration of consistency checks will increase as well. CHECKSUM can help reduce the frequency of consistency checks; however, you should consider a regular maintenance plan that includes consistency checks. To reduce the duration of consistency checks you should consider using the PHYSICAL_ONLY database option. Using the PHYSICAL_ONLY argument checks the physical consistency of the database to include torn pages and checksum errors.

NOTE

DBCC using any REPAIR argument is not recommended with Microsoft SharePoint 2010. In the event consistency checks return errors, it is recommended to restore a previous version of the database from the most recent backup set.

Use the DBCC CHECKDB statement to use PHYSICAL_ONLY:

```
DBCC CHECKDB [DatabaseName] PHYSICAL_ONLY
```

NOTE

The PHYSICAL_ONLY argument does not include FILESTREAM data.

When considering your consistency check plans you can use the ESTIMATEONLY argument to estimate the amount of tempdb space required to complete the operation without performing the actual database check. As the size of your database increases, additional space will be required to support these operations.

For additional information about using DBCC with SharePoint 2010 Products, see Database maintenance for SharePoint Server 2010 (http://technet.microsoft.com/en-us/library/cc262731.aspx).

MAXDOP

MAXDOP or max degree of parallelism can be used to improve the performance of resource intensive operations to include consistency checks, index reorganization, and index rebuild operations. Highly parallelized operations are recommended during standard maintenance windows.

NOTE

Index reorganization operations are always single-threaded.

To configure MAXDOP for specific maintenance operations and to learn more about MAXDOP, see max degree of parallelism Option (http://msdn.microsoft.com/en-us/library/ms181007.aspx).

In the event maintenance operations on multi-terabyte content databases cannot be completed or supported within the maintenance window or hardware constraints unique to your environment, you may consider solutions such as utilizing SAN-based snapshots mounted to separate server computers where the penalties associated with maintenance can be abstracted from the production environment.

Timer Job Definitions

Microsoft SharePoint 2010 includes several maintenance operations executed through scheduled timer job definitions, when considering multi-terabyte databases the timer job definitions listed in the following table should be disabled, executed during maintenance windows when there is low user concurrency, or replaced through DBA-managed maintenance plans.

Table 4 Timer Job Definitions Recommended to be Disabled

Timer Job Definition	Description
Cell Storage Data Cleanup Timer Job	Deletes temporary cell storage data and frees SQL Server disk space.
Cell Storage User Data Deletion Job	Deletes user data that is stored as cell storage.

	This job should only be run only if the SQL Server database server is running out of disk space.

Upgrade

As the size of the content database increases, the amount of time required to process its content will increase. Upgrade is one of those scenarios. Upgrade should be thoroughly tested to ensure it can be completed within the allotted maintenance window of your organization.

Requirements

Microsoft SharePoint 2010

For Microsoft SharePoint 2010 hardware and software requirements, see Hardware and software requirements (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/cc262485.aspx).

SQL Server

For SQL Server hardware and software requirements, see Hardware and Software Requirements for Installing SQL Server 2008 R2 (http://msdn.microsoft.com/en-us/library/ms143506.aspx).

These requirements must be met to support the basic deployment of a server farm environment. As workloads change and demand increases it may be required to provide additional resources through scaling-out or scaling-up your topology.

The minimum recommended I/O/sec. required for Microsoft SharePoint 2010 content databases is 0.25 I/O per sec. per GB content stored. Optimal performance requires a minimum 2 I/O per sec. per GB stored. In either profile when using Microsoft SharePoint 2010, the storage subsystem must measure 20ms or less TTFB.

TTFB or **T**ime to **F**irst **B**yte represents the duration from the database server making a request to the first byte received by the database server from the storage subsystem. This comprises the time taken to send the request and the time taken to request the first byte.

Conclusion

Microsoft SharePoint 2010 can scale to meet the demands of organizations of all sizes. However, as an environment is scaled, the operational processes and procedures necessary to maintain that environment will expand in their scope and require management tools and solutions beyond those provided with the product or within the product's immediate technology stack. This white paper is intended to provide some of the guidance and considerations that should be applied as you scale your deployment.

Additional Resources

TechNet Article - Performance and capacity management (SharePoint Server 2010)

TechNet Article – SQL Server and storage (SharePoint Server 2010)

About the Author

Bill Baer is a Senior Technical Product Manager and Microsoft Certified Master for SharePoint in the SharePoint product group in Redmond, Washington. He was previously a Hewlett-Packard Technology Solutions Group MVP with a background in infrastructure engineering and enterprise deployments of SharePoint Products and Technologies.