

Estimate Performance and Capacity Requirements for Large Scale Document Repositories

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.

Some examples depicted herein are provided for illustration only and are fictitious.  No real association or connection is intended or should be inferred.

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.

© 2010 Microsoft Corporation. All rights reserved.

Estimate Performance and Capacity Requirements for Large Scale Document Repositories

Quentin Christensen, Ken LaToza, Hari Kizhakekalam

Microsoft Corporation

May 2010

**Applies to:** Microsoft SharePoint Server 2010

Summary

This white paper provides guidance on performance and capacity planning for large scale document management solutions. The focus of this document is on the performance characteristics of document libraries as size increases and throughput of high volume operations, such as policy update and retention. Testing conducted by Microsoft has shown that single document libraries can support tens of millions of documents. This limit should be considered as the limit for documents in a single site, site collection, and content database.

Capacity planning testing has indicated that there are certain limits to the scale of large document libraries and the throughput of certain high volume operations. Considerations must be made for these limitations. Document management repositories are heavily dependent on Microsoft® SQL Server® performance because list operations, such as queries and document uploads, consume SQL Server resources. Front-end Web servers can be scaled out to support additional throughput, but in most cases for this scenario the instance of SQL Server will become the bottleneck. To continue to scale, content must be split across multiple instances of SQL Server databases.

High volume operations, such as policy update, retention actions, and document uploading, have a throughput limitation because the operations are single threaded. To increase throughput you can split content across multiple content databases. To ensure that items are uploaded appropriately and expire appropriately, proper planning of the input and expiration of content must be conducted.

In large document repository scenarios, planning for throughput will involve considering how quickly content can be uploaded and how quickly it can expire without a large effect on overall performance. The record center site template has characteristics that make it ideal to use as a site for large scale document repositories.

For general information about how to plan and run your capacity planning for Microsoft SharePoint® Server 2010, see [Capacity management and sizing for SharePoint Server 2010](http://technet.microsoft.com/en-us/library/cc261700(office.14).aspx). To learn more about large lists, see [Designing Large Lists and Maximizing List Performance](http://technet.microsoft.com/en-us/library/ff608068(office.14).aspx).

Contents

[Test farm characteristics 4](#_Toc262811032)

[Workload 4](#_Toc262811033)

[Test definitions 4](#_Toc262811034)

[Test mix 5](#_Toc262811035)

[Test load 6](#_Toc262811036)

[Hardware settings and topology 6](#_Toc262811037)

[Farm #1 Large Document Library tests 6](#_Toc262811038)

[Farm #2 Scale Out and Document Parser tests 9](#_Toc262811039)

[Test results 11](#_Toc262811040)

[Farm #1 Large Document Library 11](#_Toc262811041)

[Performance as list size increases 11](#_Toc262811042)

[Document uploading performance 12](#_Toc262811043)

[Farm #2 Scale Out 13](#_Toc262811044)

[Effect of front-end Web server scale on throughput 13](#_Toc262811045)

[Farm #2: Document Center and Record Center site template comparison 15](#_Toc262811046)

[Policy update 15](#_Toc262811047)

[Retention actions 16](#_Toc262811048)

[Recommendations 18](#_Toc262811049)

[Hardware recommendations 18](#_Toc262811050)

[Scaled-up and scaled-out topologies 18](#_Toc262811051)

[Estimating throughput targets 18](#_Toc262811052)

[Troubleshooting performance and scalability 19](#_Toc262811053)

[Common bottlenecks and their causes 19](#_Toc262811054)

[Performance monitoring 20](#_Toc262811055)

[Web servers 20](#_Toc262811056)

[Database servers 20](#_Toc262811057)

[See Also 20](#_Toc262811058)

# Test farm characteristics

This white paper is the result of a series of performance tests that were conducted with SharePoint Server 2010. Most of the tests were conducted in a similar manner. This section includes an explanation of the testing methodology that was used for tests that are discussed in this paper. Deviations from this methodology are noted where data is presented.

## Workload

Testing for this scenario was designed to help develop estimates of how different farm configurations are affected by large document repositories. This document shows how performance responds to changes to the following variables:

* Additional front-end Web servers for one database sever
* Document Center versus Record Center site templates

It is important to note that the specific capacity and performance figures presented in this article are different from the figures in real-world environments. The figures that are presented are intended to provide a starting point for the design of an appropriately scaled environment. After you have completed your initial system design, test the configuration to determine whether your system will support the factors in your environment.

### Test definitions

This section defines the test scenarios and provides an overview of the test process that was used for each scenario. Detailed information such as test results and specific parameters are given in each of the test results sections later in this white paper.

|  |  |
| --- | --- |
| **Test name** | **Test description** |
| **Document Upload** | 1.     Upload a document. |
| 2.     Edit and update the properties of the document. |
| **Document Upload and Route** | 1.     Upload a document. |
| 2.     Edit and update the properties of the document. |
| 3.     Route the document matching a routing rule. |
| **Document Download** | 1.     Download a document. |
| **Access Document Library** | 1.     Access a document library list view page. |
| **Access Home Page with Content Query Web Parts** | 1.     Access a document center home page that has 3 content query Web Parts. |
| 2.     Cached content query Web Part returns 15 highest rated documents. |
| 3.     Cached content query Web Part returns the 15 newest documents. |
| 4.     Non-cached content query Web Part returns the 15 most recent items modified by the current user. |
| **Managed Metadata Fallback Query** | 1.     A list view query that returns more than 5,000 results filtering on a single value, managed metadata column. A fallback query returns more results than the list view threshold, so only a subset of the results are returned. |
| **Managed Metadata Selective Query** | 1.     A list view query that returns 4,000 results filtering on a single value, managed metadata column. A selective query returns fewer results than the list view threshold, so all results that match are returned. |
| **Content Type Fallback Query** | 1.     A list view query that returns more than 5,000 results filtering by content type. |

### Test mix

|  |  |  |
| --- | --- | --- |
| **Test Mix** | **Test Name** | **% in the mix** |
| **1** | Document Upload | 5% |
| Document Upload and Route | 5% |
| Document Download | 10% |
| Access Document Library | 16% |
| Access Home Page with Content Query Web Parts | 24% |
| Managed Metadata Fallback Query | 10% |
| Managed Metadata Selective Query | 10% |
| Content Type Fallback Query | 10% |
| Content Type Selective Query | 10% |
| **2** | Document Upload | 15% |
| Document Upload and Route | 0% |
| Document Download | 6% |
| Access Document Library | 10% |
| Access Home Page with Content Query Web Parts | 15% |
| Managed Metadata Fallback Query | 11% |
| Managed Metadata Selective Query | 16% |
| Content Type Fallback Query | 16% |
| Content Type Selective Query | 11% |

The test mixture that was used for a test varied, based on the particular test. Tests were conducted by using a Visual Studio Test System. Specific data points for each test were populated, and then the test mix was run for 2 minutes of warm up and 10 minutes of data collection. The results that are presented in this document are averaged over those 10 minutes.

|  |
| --- |
| **C:\Users\kvice\AppData\Local\Temp\DxEditor\DduePreview\Default\ad986110-8676-44ca-a97e-ded47ba2a97a\local\note.gifNote:** |
| The mixture of operations that was used to measure performance for the purpose of this white paper is artificial. All results are only intended to illustrate performance characteristics in a controlled environment under a specific set of conditions. These test mixtures are made up of an uncharacteristically high amount of list queries that consume a large amount of SQL Server resources compared to other operations. Your specific environmental variables and mix of operations will vary. |

### Test load

Tests were conducted at an optimal load point, or Green Zone, with a general mix of operations. To measure particular changes, tests were conducted at each point that a variable was altered. To find the optimal load point, additional threads were added to saturate the environment while remaining under the following metrics:

* 75th percentile latency is less than 1 second
* Front-end Web server CPU is less than 50%
* SQL Server CPU is less than 50%
* Application server CPU is less than 50%
* Failure rate is less than 0.01

## Hardware settings and topology

To provide a high level of test-result detail, several farm configurations were used for testing. Farm configurations ranged from one to five front-end Web servers, one application server, and one database server running SQL Server 2008. Testing was performed with two client computers. All Web server computers and the database server computers were 64-bit, and the client computers were 32-bit. Two separate farms were used for testing. One farm was for testing a document library with up to 30 million documents, and another farm was for scaling front-end Web servers and comparing document center and record center site template performance.

One aspect of the test configuration was significantly different from most real world deployments. The application servers contained SQL Server instances used as a logging database. This was done to reduce load on the main SQL Server instance because the logging level was much higher than in real world deployments.

### Farm #1 Large Document Library tests

The following table lists the specific hardware that was used for testing.

|  |  |  |
| --- | --- | --- |
|  | Front-End-Web Servers (2) | Application Server (1) |
| Processor(s) | 2 quad core @2.33 GHz | 2 quad core @2.33GHz |
| RAM | 8 GB | 8 GB |
| Operating System | Windows Server® 2008 R2, 64 bit | Windows Server 2008 R2, 64 bit |
| Size of the SharePoint drive | 50 GB + 18 GB + 205 GB 15K SAS Disk 1: Operating System Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory | 50 GB + 18 GB + 300 GB 15K SAS Disk 1: Operating System Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory |
| Number of NICs | 2 | 2 |
| NIC Speed | 1 Gigabit | 1 Gigabit |
| Authentication | Windows NTLM | Windows NTLM |
| Load balancer type | Hardware load balancing | Hardware load balancing |
| Software version | SharePoint Server 2010 (pre-release version) | SharePoint Server 2010 (pre-release version) |
| Services running locally | N/A | Search Crawler  Search Query  Logging Database |

|  |  |
| --- | --- |
|  | 1 SQL Server Database Server |
| Processor(s) | 4 quad core @3.2 GHz |
| RAM | 32 GB |
| Operating System | Windows Server 2008 R2, 64-bit |
| Storage and geometry | 15x450GB 15K SAS |
| Number of NICs | 2 |
| NIC Speed | 1 Gigabit |
| Authentication | NTLM |
| Software version | SQL Server 2008 R2 CTP3 |

#### Topology





#### Dataset #1 Large Document Library

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Documents** | **500,000** | **5 Million** | **10 Million** | **20 Million** | **30 Million** |
| **Database size** | 50 GB | .5 TB | 1 TB | 2 TB | 3 TB |
| **Average Document Size** | 20 KB | 20 KB | 20 KB | 20 KB | 20 KB |
| **Number of Folders** | 25 | 250 | 500 | 1,000 | 1,500 |
| **Number of Documents Per Folder** | 20,000 | 20,000 | 20,000 | 20,000 | 20,000 |

### Farm #2 Scale Out and Document Parser tests

The following table lists the specific hardware that was used for testing. The main difference between this configuration and Farm #1 is that the SQL Server database server has half the memory.

|  |  |  |
| --- | --- | --- |
|  | Front-End Web Servers (2-4) | Application Server (1) |
| Processor(s) | 2 quad core @2.33 GHz | 2 quad core @2.33GHz |
| RAM | 8 GB | 8 GB |
| Operating System | Windows Server 2008 R2, 64 bit | Windows Server 2008 R2, 64 bit |
| Size of the SharePoint drive | 50 GB + 18 GB + 205 GB 15K SAS Disk 1: Operating System Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory | 50 GB + 18 GB + 300 GB 15K SAS Disk 1: Operating System Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory |
| Number of NICs | 2 | 2 |
| NIC Speed | 1 Gigabit | 1 Gigabit |
| Authentication | Windows NTLM | Windows NTLM |
| Load balancer type | Hardware load balancing | Hardware load balancing |
| Software version | SharePoint Server 2010 (pre-release version) | SharePoint Server 2010 (pre-release version) |
| Services running locally | N/A | Search Crawler  Search Query  Logging Database |

|  |  |
| --- | --- |
|  | SQL Server Database Server |
| Processor(s) | 4 quad core @3.2 GHz |
| RAM | 16 GB |
| Operating System | Windows Server 2008 R2, 64-bit |
| Storage and geometry | 15x450GB 15K SAS |
| Number of NICs | 2 |
| NIC Speed | 1 Gigabit |
| Authentication | NTLM |
| Software version | SQL Server 2008 R2 CTP3 |

#### Topology



# Test results

The following tables show the test results with a mix of operations being run against document libraries in SharePoint Server 2010. For each group of tests, only certain specific variables are changed to show the progressive effect on farm performance.

Note that all the tests reported on in this article were conducted without think time, a natural delay between consecutive operations. In a real-world environment, each operation is followed by a delay as the user performs the next step in the task, although there might be many users so load might be continuous. In this testing, each operation was immediately followed by the next operation, which resulted in a continuous even load on the farm. This load introduced database contention and other factors that can adversely affect performance.

For information about bottlenecks with large scale document repositories in SharePoint Server 2010, see the Common bottlenecks and their causes section later in this article.

## Farm #1 Large Document Library

### Performance as list size increases

SharePoint Server 2010 supports document libraries with tens of millions of items. As list size increases, performance gradually decreases. It is important to plan your solution appropriately. List size is one of several factors that affect performance of document libraries. Testing was conducted on document libraries ranging in size from 500,000 to 30 million documents by using Test Mix 1. Overall throughput gradually decreased and the latency of specific operations increased.

### Document uploading performance

The following graph shows the performance of uploading documents into SharePoint. Two factors are measured that affect performance. The record center template has document parsing disabled, which makes it much faster to upload content (at the cost of no promotion/demotion, which isn’t used to ensure that the file isn’t modified). There is also a difference between uploading content directly and using the OfficialFile.asmx web service and matching a content organizer rule. The tests that are marked with upload show the results of uploading content directly to a final location. The tests that are marked with send to show the submitting of content to the OfficialFile.asmx web service and the content matches a content organizer rule to move the document to a final location. The graph shows the average performance of uploading 100,000 documents and 500,000 documents, which shows a slight degradation as the content size increases, in part due to the uploading being performed without the benefit of what would generally be regularly scheduled database maintenance.

## Farm #2 Scale Out

### Effect of front-end Web server scale on throughput

Operations on document libraries have a larger effect on SQL Server resources in general compared to other types of operations, such as cached web pages. In heavy document management workloads, scaling front-end Web servers might result in minimal performance gains because the bottleneck is IO and potentially lock contention in SQL Server. The following graphs show the performance effect of adding additional front-end Web servers. Test Mix 2 was used to load the farm.

The following graph shows that under the specific test conditions, SQL Server performance becomes the limitation when using between two and three front-end web servers and additional servers will not increase performance. The test mix used is heavily weighted towards operations that have a significant effect on SQL Server performance that is not realistic of most actual customer environments. Scaling up the SQL Server or a more typical real world workload will support more front-end Web servers to achieve higher throughput. This graph is included only to illustrate that heavy document management loads are database intensive and that there is a point where scaling front-end Web servers will not increase throughput.

## Farm #2: Document Center and Record Center site template comparison

Two site templates were used to conduct the following tests; the key difference is the document parser. In all site templates except the Record Center site template, the document parser is enabled. The document parser is disabled to ensure that the file isn’t modified by SharePoint’s metadata promotion/demotion mechanism. As a side effect, disabling the document parser results in better performance for certain operations, such as upload. This makes the Record Center site template more suitable than other site templates for storing large amounts of content, such as an archive.

### Policy update

Policy update is a long running operation that is handled by a timer job. Policy update occurs when a policy change, such as adding a retention action, is made to a content type. The performance of policy updates is affected by the number of content databases, the capabilities of the hardware that is running SQL Server, and the site template the item is in. To update items faster, you can split content across multiple site collections with separate content databases.

Testing has shown that policy update has a minimal effect on overall farm performance, using approximately 2 percent to 5 percent of SQL Server CPU when the timer job runs. Our test results showed that approximately 11,000 items could be processed per hour and about two million items in a week by using our lab configuration. Very large policy changes are uncommon, but the update timer job might run for a long period of time if this occurs.

### Retention actions

Retention actions give an administrator control over the processing and expiration of content to reduce storage size, eDiscovery costs, and legal liability. SharePoint Server 2010 supports multistage content type and location-based retention actions. Several out of box retention actions are available.

In large scale document repositories a large number of items might have policy and retention actions configured. Retention actions are long running operations because a large number of items might need to be processed and have the appropriate retention action applied. In cases where a large amount of content will expire, the rate of processing might be a limitation. Proper planning must be done for the rate of expiration.

The rate of expiration is somewhat fixed for a single content database. To improve the rate of expiration, split content across multiple databases. The rate of expiration varies based on the action and site template.

# Recommendations

This section provides general performance and capacity recommendations. Use these recommendations to determine the capacity and performance characteristics of the starting topology that you created and to decide whether you have to scale out or scale up the starting topology.

## Hardware recommendations

For specific information about minimum and recommended system requirements, see [Hardware and software requirements (SharePoint Server 2010)](http://technet.microsoft.com/en-us/library/cc262485(office.14).aspx).

|  |
| --- |
| **C:\Users\kvice\AppData\Local\Temp\DxEditor\DduePreview\Default\ad986110-8676-44ca-a97e-ded47ba2a97a\local\note.gifNote:** |
| Memory requirements for Web servers and database servers depend on the size of the farm, the number of concurrent users, and the complexity of features and pages in the farm. The memory recommendations in the following table might be sufficient for a small or light usage farm. However, memory usage should be carefully monitored to determine whether more memory must be added. |

## Scaled-up and scaled-out topologies

You can estimate the performance of your starting-point topology by comparing your topology to the starting-point topologies that are provided in [Plan for availability (SharePoint Server 2010)](http://technet.microsoft.com/en-us/library/cc748824(office.14).aspx). Doing so can help you quickly determine whether you must scale up or scale out your starting-point topology to meet your performance and capacity goals.

To increase the capacity and performance of one of the starting-point topologies, you can do one of two things. You can either scale up by increasing the capacity of your existing server computers or scale out by adding additional servers to the topology. This section describes the general performance characteristics of several scaled-out topologies. The sample topologies represent the following common ways to scale out a topology for an InfoPath Forms Services scenario:

* To provide for more user load, add Web server computers.
* To provide for more data load, add capacity to the database server role by increasing the capacity of a single (clustered or mirrored) server or by adding clustered or mirrored servers.
* Maintain a ratio of no more than eight Web server computers to one (clustered or mirrored) database server computer. Although testing in our lab yielded a specific optimum ratio of Web servers to database servers for each test scenario, deployment of more robust hardware, especially for the database server, can yield better results in your environment.

## Estimating throughput targets

Many factors can affect throughput. These factors include the number of users; the type, complexity, and frequency of user operations; the number of postbacks in an operation; and the performance of data connections. Each of these factors can have a major effect on farm throughput. You should carefully consider each of these factors when you plan your deployment.

SharePoint Server 2010 can be deployed and configured in a wide variety of ways. As a result, there is no simple way to estimate how many users can be supported by a given number of servers. Therefore, make sure that you conduct testing in your own environment before you deploy SharePoint Server 2010 in a production environment.

## Troubleshooting performance and scalability

### Common bottlenecks and their causes

During performance testing, several different common *bottlenecks* were revealed. A bottleneck is a condition in which the capacity of a particular constituent of a farm is reached. This causes a plateau or decrease in farm throughput.

The following table lists some common bottlenecks and describes their causes and possible resolutions.

|  |  |  |
| --- | --- | --- |
| Bottleneck | Cause | Resolution |
| Database contention (locks) | Database locks prevent multiple users from making conflicting modifications to a set of data. When a set of data is locked by a user or process, no other user or process can modify that same set of data until the first user or process finishes modifying the data and relinquishes the lock. | To help reduce the incidence of database locks, you can:   * Distribute content across multiple content databases. * Scale up the database server. * Tune the database server hard disk for read/write. |
| Database server disk I/O | When the number of I/O requests to a hard disk exceeds the disk’s I/O capacity, the requests will be queued. As a result, the time to complete each request increases. | Distributing data files across multiple physical drives allows for parallel I/O. The blog [SharePoint Disk Allocation and Disk I/O](http://go.microsoft.com/fwlink/?LinkId=129557) (http://go.microsoft.com/fwlink/?LinkId=129557) contains useful information about resolving disk I/O issues. |
| Web server CPU utilization | When a Web server is overloaded with user requests, average CPU utilization will approach 100 percent. This prevents the Web server from responding to requests quickly and can cause timeouts and error messages on client computers. | This issue can be resolved in one of two ways. You can add additional Web servers to the farm to distribute user load, or you can scale up the Web server or servers by adding higher-speed processors. |

### Performance monitoring

To help you determine when you have to scale up or scale out your system, use performance counters to monitor the health of your system. Use the information in the following tables to determine which performance counters to monitor and to which process the performance counters should be applied.

### Web servers

The following table shows performance counters and processes to monitor for Web servers in your farm.

|  |  |  |
| --- | --- | --- |
| Performance counter | Apply to object | Notes |
| Processor time | Total | Shows the percentage of elapsed time that this thread used the processor to execute instructions. |

### Database servers

The following table shows performance counters and processes to monitor for database servers in your farm.

|  |  |  |
| --- | --- | --- |
| Performance counter | Apply to object | Notes |
| Average disk queue length | Hard disk that contains the appropriate database files | Deep queue depths can indicate a problem if the latencies are also high. However, if the queue is deep, but latencies are low that is, getting emptied and refilled very quickly), this might just indicate an active and efficient system. A high queue length does not necessarily imply a performance problem. |
| Processor time | SQL Server process | Average values greater than 80 percent indicate that processor capacity on the database server is insufficient. |
| Processor time | Total | Shows the percentage of elapsed time that this thread used the processor to execute instructions. |
| Memory utilization | Total | Shows the average utilization of system memory. |

## See Also

[Designing Large Lists and Maximizing List Performance](http://technet.microsoft.com/en-us/library/ff608068(office.14).aspx)