Capacity planning for Microsoft SharePoint Server 2010

Microsoft Corporation

Published: January 2011

Author: Microsoft Office System and Servers Team (itspdocs@microsoft.com)

Abstract

This book provides information about planning for capacity and performance requirements for deploying Microsoft SharePoint Server 2010. Subjects include sizing, performance testing, software boundaries, and capacity case studies. The audiences for this book are business application specialists, line-of-business specialists, information architects, IT generalists, program managers, and infrastructure specialists who are planning a solution based on SharePoint Server 2010. This book is part of a set of four planning guides that provide comprehensive IT planning information for SharePoint Server.

For information about planning the architecture of a SharePoint Server 2010 deployment, see Planning guide for server farms and environments for Microsoft SharePoint Server 2010 (http://go.microsoft.com/fwlink/?LinkID=189513).

For information about planning for sites and solutions created by using SharePoint Server, see Planning guide for sites and solutions for Microsoft SharePoint Server 2010, Part 1 (http://go.microsoft.com/fwlink/?LinkID=196150) and Planning guide for sites and solutions for Microsoft SharePoint Server 2010, Part 2 (http://go.microsoft.com/fwlink/?LinkId=208024).

The content in this book is a copy of selected content in the SharePoint Server 2010 technical library (http://go.microsoft.com/fwlink/?LinkId=181463) as of the publication date. For the most current content, see the technical library on the Web.

Microsoft

Contents

Getting help

Every effort has been made to ensure the accuracy of this book. This content is also available online in the Office System TechNet Library, so if you run into problems you can check for updates at:

http://technet.microsoft.com/office

If you do not find your answer in our online content, you can send an e-mail message to the Microsoft Office System and Servers content team at:

itspdocs@microsoft.com

If your question is about Microsoft Office products, and not about the content of this book, please search the Microsoft Help and Support Center or the Microsoft Knowledge Base at:

http://support.microsoft.com

Performance and capacity management (SharePoint Server 2010)

Performance and capacity planning is the process of mapping your solution design for Microsoft SharePoint Server 2010 to a farm size and set of hardware that will support your business goals.

The articles in this section include:

- Capacity management and sizing for SharePoint Server 2010

  This article walks you through the process of determining the hardware requirements for a single farm, and provides an overview of the planning process.

- SharePoint Server 2010 capacity management: Software boundaries and limits

  This article provides a starting point for planning the performance and capacity of your system. This article includes performance and capacity testing results and guidelines for acceptable performance.

- Performance and capacity technical case studies (SharePoint Server 2010)

  This article provides links to key technical case study articles that contain performance and capacity details for specific environments running SharePoint Server 2010.

- Performance and capacity test results and recommendations (SharePoint Server 2010)

  This article provides links to articles that provide test results and recommendations for specific feature sets in SharePoint Server 2010.

- Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

  This article describes a process for planning storage and SQL Server capacity for a SharePoint Server 2010 deployment.

The following resources can also be helpful for capacity planning:

- Hardware and software requirements (SharePoint Server 2010)
- Technical diagrams:
  - Topologies for SharePoint Server 2010
  - Search Architectures for Microsoft SharePoint Server 2010
  - Design Search Architectures for Microsoft SharePoint Server 2010
  - Search Environment Planning for Microsoft SharePoint Server 2010

  To download these models, see Technical diagrams (SharePoint Server 2010).

Capacity management and sizing for SharePoint Server 2010

The articles in this section help you to make the following decisions regarding the appropriate capacity for your Microsoft SharePoint Server 2010 environment:

- Understand the concepts behind effective capacity management.
- Define performance and capacity targets for your environment.
- Select the appropriate data architecture.
- Choose hardware to support the number of users and the features you intend to deploy.
- Test, validate, and adjust your environment to achieve your performance and capacity targets.
- Monitor and adjust your environment to match demand.

In this section:

Capacity management and sizing overview for SharePoint Server 2010

This article provides an overview of how to effectively plan and manage the capacity of Microsoft SharePoint Server 2010 environments. This article also describes how to maintain a good understanding of the capacity needs and capabilities of your deployment, by analysis of performance and volume data. It also reviews the major application impacts that affect capacity, including content characteristics and usage.

Capacity management is an ongoing process, because no implementation remains static with regard to content and usage. You need to plan for growth and change, so that your SharePoint Server 2010–based environment can continue to deliver an effective business solution.

Capacity Planning is only one part of the capacity management cycle. It is the initial set of activities that brings the design architect to the point where there is an initial architecture that the architect believes will best serve the SharePoint Server 2010 deployment. The capacity management model includes additional steps to help you validate and tune the initial architecture, and provides a feedback loop for re-planning and optimizing the production environment until it can support design goals with optimal choices of hardware, topology, and configuration.

In this article:

Glossary

The following specialized terms are used in SharePoint Server 2010 capacity management documentation.

- **RPS** Requests per second. The number of requests received by a farm or server in one second. This is a common measurement of server and farm load. The number of requests processed by a farm is greater than the number of page loads and end-user interactions. This is because each page contains several components, each of which creates one or more requests when the page is loaded. Some requests are lighter than other requests with regard to transaction costs. In our lab

tests and case study documents, we remove 401 requests and responses (authentication handshakes) from the requests that were used to calculate RPS because they have insignificant impact on farm resources.

- **Peak hours** The time or times during the day when load on the farm is at its maximum.
- **Peak load** The average maximum daily load on the farm, measured in RPS.
- **Load spike** Transient load peaks that fall outside usual peak hours. These can be caused by unplanned increases in user traffic, decreased farm throughput because of administrative operations, or combinations of such factors.
- **Scale up** To scale up means to add resources such as processors or memory to a server.
- **Scale out** To scale out means to add more servers to a farm.

Who should read capacity management articles?

Consider the following questions to determine whether you should read this content.

Evaluating SharePoint Server 2010

I am an IT pro or business decision maker, and I am looking for a solution to specific business problems. SharePoint Server 2010 is an option for my deployment. Can it provide features and scalability that meet my specific requirements?

For information about how SharePoint Server 2010 scales to meet the demands of specific solutions and how to determine the hardware that will be required to support your requirements, see the following sections later in this article:

- Key differences: SharePoint Server 2010 versus Office SharePoint Server 2007
- Software limits and boundaries

For information about how to evaluate SharePoint Server 2010 for your specific business requirements, see the following articles:

- Product evaluation for SharePoint Server 2010
- SharePoint Server 2010 capacity management: Software boundaries and limits

Upgrading from Office SharePoint Server 2007

I am currently using Office SharePoint Server 2007. What has changed in SharePoint Server 2010, and what do I have to consider if I upgrade? What effect will the upgrade have on my topology's performance and scale?

For information about how performance and capacity factors are different for Office SharePoint Server 2007 and SharePoint Server 2010, see the following section later in this article:

- Key differences: SharePoint Server 2010 versus Office SharePoint Server 2007

For information about more general upgrade considerations and guidance on how to plan and execute an upgrade from Office SharePoint Server 2007, see the following article:

- Upgrading to SharePoint Server 2010

Tuning and optimizing a live SharePoint-based environment

I have deployed SharePoint Server 2010, and I want to make sure I have the appropriate hardware and topology in place. How do I validate my architecture and maintain it correctly?

For information about monitoring and performance counters for Microsoft SharePoint Server 2010 farms, see the following article:

- Monitoring and maintaining SharePoint Server 2010

For information about how to use the health monitoring tools built into the Central Administration interface, see the following article:

- Health monitoring (SharePoint Server 2010)

I have deployed SharePoint Server 2010, and I am experiencing performance issues. How do I troubleshoot and optimize my environment?

For information about monitoring and performance counters for Microsoft SharePoint Server 2010 farms, see the following article:

- Monitoring and maintaining SharePoint Server 2010

For information about troubleshooting by using the health monitoring tools built into the Central Administration interface, see the following article:

- Solving problems and troubleshooting (SharePoint Server 2010)

For a list of capacity management articles that are available for many specific SharePoint Server 2010 services and features (more articles will be added as they become available), see the following article:

- Performance and capacity test results and recommendations (SharePoint Server 2010)

For information about database sizing and performance, see the following article:

- Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

For information about Remote BLOB Storage (RBS), see the following article:

- Plan for Remote BLOB Storage (RBS) (SharePoint Server 2010)

Beginning to end

I want to know everything about SharePoint Server 2010 capacity management. Where do I start?

For information about the general concepts behind capacity management and links to additional documentation and resources, see the following article:

- Performance and capacity management (SharePoint Server 2010)

For additional information about capacity management, see the following companion articles to this overview article:

- Capacity planning for SharePoint Server 2010
- Performance testing for SharePoint Server 2010

- Monitoring and maintaining SharePoint Server 2010

You should now have a good understanding of the concepts. For information the limits and boundaries of SharePoint Server 2010, see the following article:

- SharePoint Server 2010 capacity management: Software boundaries and limits

When you are ready to identify a starting point topology for your SharePoint Server 2010–based environment, you can look through the library of available technical case studies to find the one that most closely matches your requirements. For a list of the case studies (more case studies will be added as they become available), see the following article:

- Performance and capacity technical case studies (SharePoint Server 2010)

For a list of capacity management articles that are available for many specific SharePoint Server 2010 services and features (more articles will be added as they become available), see the following article:

- Performance and capacity test results and recommendations (SharePoint Server 2010)

For information about database sizing and performance, see the following article:

- Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

For information about Remote BLOB Storage (RBS), see the following article:

- Plan for Remote BLOB Storage (RBS) (SharePoint Server 2010)

For information about health monitoring and troubleshooting by using the health monitoring tools built into the Central Administration interface, see the following articles:

- Health monitoring (SharePoint Server 2010)
- Solving problems and troubleshooting (SharePoint Server 2010)

For information about general performance tuning guidelines and a variety of specific performance and capacity subjects (more articles will be added as they become available), see the following article:

- Use search administration reports (SharePoint Server 2010)

For more information about how to virtualize SharePoint Server 2010–based servers, see the following article:

- Virtualization planning (SharePoint Server 2010)

Four fundamentals of performance

Capacity management focuses on the following four major aspects of sizing your solution:

- **Latency** For the purposes of capacity management, latency is defined as the duration between the time that a user initiates an action, such as clicking a hyperlink, and the time until the last byte is transmitted to the client application or Web browser.
- **Throughput** Throughput is defined as the number of concurrent requests that a server or server farm can process.
- **Data scale** Data scale is defined as the content size and data corpus that the system can host. The structure and distribution of the content databases has a significant effect on the time it takes

the system to process requests (latency) and the number of concurrent requests it can serve (throughput).

- **Reliability** Reliability is a measurement of the ability of the system to meet the targets set for the latency and throughput over time.

The main goal of managing your environment's capacity is to establish and maintain a system that meets your organization's latency, throughput, data scale, and reliability targets.

Latency

Latency, also known as end-user perceived latency, is composed of three major components:

- The time it takes the server to receive and process the request.
- The time it takes the request and the server response to transfer over the network.
- The time it takes the response to render on the client application.

Different organizations define different latency goals based on business requirements and user expectations. Some organizations can afford latency of several seconds, whereas other organizations require very fast transactions. Optimizing for very fast transactions is usually more costly, and usually requires more powerful clients and servers, more recent browser and client application versions, high-bandwidth network solutions, and possibly development investments and page tuning.

Some major factors that contribute to longer end-user perceived latencies, and examples of some common problems, are described in the following list. These factors are especially relevant in scenarios where the clients are geographically distant from the server farm, or are accessing the farm across a low-bandwidth network connection.

- Features, services, or configuration parameters that are not optimized might delay the processing of requests and impact latency for both remote and local clients. For more information, see Throughput and Reliability later in this article.
- Web pages that generate unnecessary requests to the server to download required data and resources. Optimization would include downloading the minimum number of resources to draw the page, reducing the sizes of images, storing the static resources in folders that enable anonymous access, clustering requests and enabling page interactivity while resources are downloaded asynchronously from the server. These optimizations are important for achieving an acceptable first time visit browse experience.
- Excessive volume of data being transmitted over the network contributes to latency and throughput degradation. For example, images and other binary objects on a page should use a compressed format such as .png or .jpg instead of bitmaps when possible.
- Web pages that are not optimized for second-access page loads. Page Load Time (PLT) improves for second-access page loads because some page resources are cached on the client, and the browser must only download dynamic uncached content. Unacceptable second-access page load latencies are often caused by incorrect Binary Large Object (BLOB) cache configuration or local browser caching being disabled on client computers. Optimizations would include correct caching of resources on the client.

- Web pages that have non-optimized custom JavaScript code. This might slow rendering of the page on the client. Optimization would delay JavaScript from being processed on the client until the rest of the page has loaded, and preferably calling scripts instead of adding JavaScript inline.

Throughput

Throughput is described by the number of requests that a server farm can process in a unit of time, and is also often used to measure the scale of operations that the system is expected to sustain based on the size of the organization and its usage characteristics. Every operation has a specific cost in server farm resources. Understanding the demand and deploying a farm architecture that can consistently satisfy demand requires estimating the expected load, and testing the architecture under load to validate that latency does not fall below target when concurrency is high and the system is under stress.

Some common examples of low throughput conditions include the following:

- **Inadequate hardware resources** When the farm receives more requests than it can process concurrently, some requests are queued, which cumulatively delays the processing of each subsequent request until demand is reduced enough for the queue to be cleared. Some examples of optimizing a farm to sustain higher throughput include the following:
  - Ensure that the processors on farm servers are not over-utilized. For example, if CPU usage during peak hours or load spikes consistently exceeds 80 percent, add more servers or redistribute services to other farm servers.
  - Ensure that there is sufficient memory on application servers and Web servers to contain the complete cache. This will help to avoid calls to the database to serve requests for uncached content.
  - Ensure that database servers are free of bottlenecks. If total available disk IOPS are insufficient to support peak demand, add more disks or redistribute databases to underutilized disks. See the Removing Bottlenecks section of the Monitoring and Maintaining SharePoint Server 2010 Products and Technologies article for more information.
  - If adding resources to existing computers is insufficient to resolve throughput issues, add servers and redistribute affected features and services to the new servers.
- **Non-optimized custom Web pages** Adding custom code to frequently used pages in a production environment is a common cause of throughput issues. Adding custom code might generate additional round trips to the database servers or Web services to service data requests. Customization of infrequently used pages might not significantly impact throughput, but even well-optimized code can decrease farm throughput if it is requested thousands of times a day. SharePoint Server 2010 administrators can enable the Developer Dashboard to identify custom code that requires optimization. Some examples of optimizing custom code include the following:
  - Minimize the number of Web service requests and SQL queries.
  - Fetch the minimum required data in each trip to the database server while minimizing the number of necessary round trips.

    - Avoid adding custom code to frequently used pages.
    - Use indexes when you are retrieving a filtered amount of data.
- **Untrusted solutions** Deploying custom code in bin folders can cause slow server performance. Every time that a page that contains untrusted code is requested, SharePoint Server 2010 must perform security checks before the page can be loaded. Unless there is a specific reason to deploy untrusted code, you should install custom assemblies in the GAC to avoid unnecessary security checking.

Data scale

Data scale is the volume of data the server or server farm can store while meeting latency and throughput targets. Generally, the greater the data volume on the farm, the greater the impact on overall throughput and user experience. The method that is used to distribute data across disks and database servers can also affect farm latency and throughput.

Database sizing, data architecture, and sufficient database server hardware are all very important to an optimal database solution. In an ideal deployment, content databases are sized according to limits guidance and are distributed across physical disks so that requests are not queued because of disk overutilization, and database servers are able to support peak loads and unexpected spikes without exceeding resource utilization thresholds.

Also, certain operations can lock certain tables during the operation. An example of this is large site deletion, which can cause the related tables in the content database where the site resides to be locked until the delete operation is completed.

Some examples of optimizing a farm for data and storage performance include the following:

- Ensure that databases are properly distributed across the database servers, and that database server resources are sufficient to support the volume and distribution of data.
- Separate database volumes into unique Logical Units (LUNs), consisting of unique physical disk spindles. Use multiple disks that have low seek time and appropriate RAID configurations to satisfy database server storage demands.
- You can use Remote BLOB Storage (RBS) if your corpus contains many Binary Large Objects (BLOBs). RBS can provide the following benefits:
    - BLOB data can be stored on less expensive storage devices that are configured to handle simple storage.
    - The administration of the BLOB storage is controlled by a system that is designed specifically to work with BLOB data.
    - Database server resources are freed for database operations.

    These benefits are not free. Before you implement RBS with SharePoint Server 2010, you should evaluate whether these potential benefits override the costs and limitations of implementing and maintaining RBS.

    For more information, see Plan for Remote BLOB Storage (RBS) (SharePoint Server 2010).

For more information about how to plan data scale, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

Reliability

Reliability is the aggregate measurement of the server farm's capacity to meet established latency, throughput, and data capacity targets over time. A reliable farm is one for which uptime, responsiveness, failure rate, and frequency and amplitude of latency spikes are within established targets and operational requirements. A reliable farm can also consistently sustain latency and throughput targets during peak load and peak hours, or when system operations such as crawling or daily backups take place.

A major factor in sustaining reliability is the effect of common administrative operations on performance targets. During certain operations, such as rebuilding the database indexes, maintenance timer jobs, or deleting multiple sites that have large volume of content, the system might be unable to process user requests as quickly. In this case, both latency and throughput of end-user requests can be affected. The impact on the farm depends on the frequency and transaction cost of such less common operations, and whether they are run during normal operating hours.

Some examples of how to sustain a more reliable system include the following:

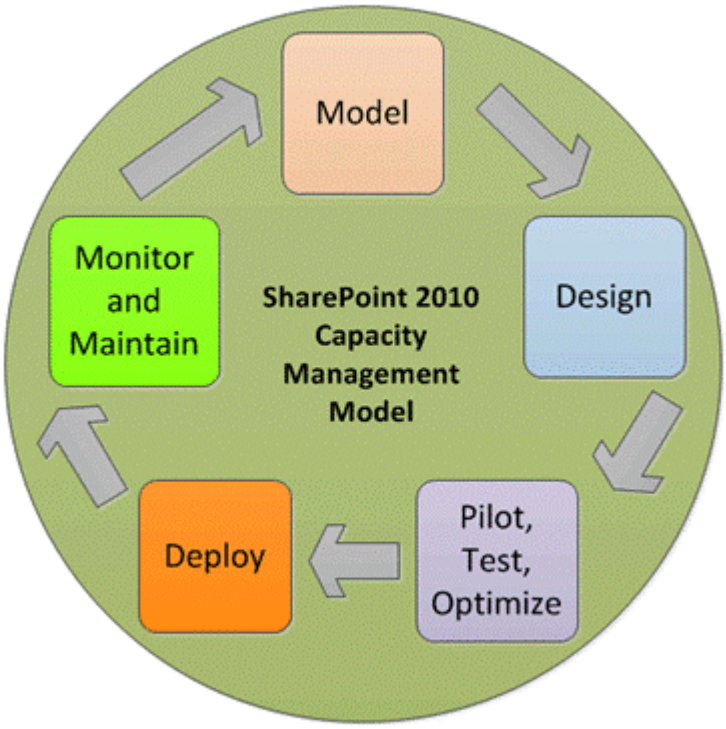
- Schedule resource-intensive timer jobs and administrative tasks during off-peak hours.
- Scale up hardware on existing farm servers, or scale out by adding Web servers, application servers or additional database servers.
- Distribute resource-intensive services and features to dedicated servers. You can also use a hardware load balancer to direct feature-specific traffic to a Web server dedicated to specific features or services.

Capacity management versus capacity planning

Capacity management extends the concept of capacity planning to express a cyclical approach in which the capacity of a SharePoint Server 2010 deployment is continually monitored and optimized to accommodate changing conditions and requirements.

SharePoint Server 2010 offers increased flexibility and can be configured to sustain usage scenarios in a wide variety of different scale points. There is no single deployment architecture. Therefore, system designers and administrators must understand the requirements for their specific environments.

SharePoint Server 2010 capacity management model



- **Step 1: Model** Modeling is the process by which you decide the key solutions that you want your environment to support, and establish all important metrics and parameters. The output of the modeling exercise should be a list of all the key data that you need to design your environment.
  - Understand your expected workload and dataset.
  - Set farm performance and reliability targets.
  - Analyze the SharePoint Server 2010 IIS logs.
- **Step 2: Design** Once you have collected the data from Step 1, you can design your farm. Outputs are detailed data architecture and physical and logical topologies.
  - Determine your starting point architecture.
  - Select your hardware.

- **Step 3: Pilot, Test, and Optimize** If you have designed a new deployment, you need to deploy a pilot environment for testing against your workload and expected usage characteristics. For an existing farm, testing is advised when major changes are being made to the infrastructure, but regular optimization based on monitoring results might be necessary to maintain performance targets. The output from this phase is analysis of test results against targets, and an optimized architecture able to sustain established performance and capacity targets.
  - **Pilot** Deploy a pilot environment.
  - **Test** Test against latency and throughput targets.
  - **Optimize** Gather test results and make any required changes to the farm resources and topology.
- **Step 4: Deploy** This step describes implementing the farm, or deploying changes to an existing farm. Output for a new design is a completed deployment to live production, including all content and user migrations. Output for an existing farm is revised farm maps and updates to maintenance plans.
- **Step 5: Monitor and maintain** This step describes how to set up monitoring, and how to predict and identify bottlenecks and perform regular maintenance and bottleneck mitigation activities.

Oversizing versus undersizing

Oversizing describes an approach to farm design in which targets are achieved without full utilization of hardware, and the resources in the SharePoint Server 2010 farm are significantly and consistently underutilized. In an oversized deployment, memory, CPU, and other indicators on the farm's resources show that it can well serve the demand with fewer resources. The downside of oversizing is increased hardware and maintenance expenditures and can impose greater power and space demands.

Undersizing describes an approach to farm design in which performance and capacity targets are not achievable because hardware resources in the SharePoint Server 2010 farm are over-utilized. Undersizing a farm is sometimes done to reduce hardware costs, but generally results in high latency leading to a poor user experience, low satisfaction, frequent escalations, high support costs, and unnecessary spending for troubleshooting and tuning the environment.

When you design your farm, make sure that your farm can meet established performance and capacity targets, both under regular peak load and unexpected spikes. Design, testing, and optimization will help you ensure that your farm has the correct hardware.

To maintain performance targets and accommodate growth, it is always more desirable to have more resources than you need to meet your targets. The cost of overinvestment in hardware is usually far less than the cumulative expenses related to troubleshooting problems cause by undersizing.

You should always size a system to respond adequately during peak demand, which might be different for specific services at different times. To effectively estimate capacity requirements, you need to identify the worst case demand period for all resources. There might be increased load on various features and services at certain times of the day, such as first thing in the morning or after lunch.

The farm also must be able to support unplanned peaks, such as when organization-wide announcements are made and an unusually high number of users access a site at the same time. During such periods of high demand, users will experience high latency or not get a response from the farm at all unless sufficient farm resources are available to satisfy the increased load on the farm.

Farm capacity should also be revisited when additional users will be provisioned in the enterprise. Situations such as a merger or acquisition characterized by new employees or members accessing the farm can have adverse effects on performance if not planned and estimated in advance.

Operational states: Green Zone and Red Zone

When we describe the load of a production system, we refer to two major operational states: the "*Green Zone*" state in which the system is operating under the normal, expected load range, and the "*Red Zone*" state, which is a state in which the farm experiences very high transient resource demand that can only be sustained for limited periods until failures and other performance and reliability issues occur.

Green Zone This is the state at which the server or farm is operating under normal load conditions, up to expected daily peak loads. A farm operating in this range should be able to sustain response times and latency within acceptable parameters.

Red Zone The operating range in which load is greater than normal peak load, but can still service requests for a limited period. This state is characterized by greater than normal latency and possible failures caused by saturation of system bottlenecks.

The ultimate goal of farm design is to deploy an environment that can consistently support Red Zone load without service failure and within acceptable latency and throughput targets.

Software limits and boundaries

In SharePoint Server 2010, there are certain limits that are by design and cannot be exceeded, and other limits that are set to default values that can be changed by the farm administrator. There are also certain limits that are not represented by a configurable value, such as the number of site collections per Web application.

Boundaries are absolute limits that cannot be exceeded by design. It is important to understand these limits to ensure that you do not make incorrect assumptions when you design your farm.

An example of a boundary is the 2 GB document size limit. You cannot configure SharePoint Server 2010 to store documents that are larger than 2 GB. This is a built-in absolute value, and cannot be exceeded by design.

Thresholds are those that have a default value that cannot be exceeded unless the value is modified. Thresholds can, in certain circumstances, be exceeded to accommodate variances in your farm design. However, it is important to understand that doing this might affect the performance of the farm and the effective value of other limits.

The default value of certain thresholds can only be exceeded up to an absolute maximum value. A good example is the document size limit again. By default, the document size limit is set to 50 MB, but can be changed to a maximum value of 2 GB.

Supported limits define the tested value for a given parameter. The default values for these limits were defined by testing, and represent the known limitations of the product. Exceeding supported limits could cause unexpected results, significant performance degradation, or other detrimental effects.

Some supported limits are configurable parameters that are set by default to the recommended value, whereas other limits relate to parameters that are not represented by a configurable value.

An example of a supported limit is the number of site collections per Web application. The supported limit is 500,000, which is the largest number of site collections per Web application that met performance benchmarks during testing.

It is important to note that many of the limit values that are provided in this document represent a point in a curve that describes an increasing resource load and concomitant performance degradation as the value increases. Therefore, exceeding certain limits, such as the number of site collections per Web application, might only result in a fractional decrease in farm performance. However, in most cases, operating at or near an established limit is not a best practice, as acceptable performance and reliability targets are best achieved when a farm's design provides for a reasonable balance of limits values.

Thresholds and supported limits guidelines are determined by performance. In other words, you can exceed the default values of the limits, but as you increase the limit value, farm performance and the effective value of other limits might be affected. Many limits in SharePoint Server 2010 can be changed. However, it is important to understand how changing a given limit affects other parts of the farm.

If you contact Microsoft Customer Support Services about a production system that does not meet the published minimum hardware specifications as described in Hardware and software requirements (SharePoint Server 2010), support will be limited until the system is upgraded to the minimum requirements.

How limits are established

In SharePoint Server 2010, thresholds and supported limits are established through testing and observation of farm behavior under increasing loads up to the point where farm services and operations reach their effective operational limits. Some farm services and components can support a higher load than others. Therefore, in some cases you must assign a limit value that is based on an average of several factors.

For example, observations of farm behavior under load when site collections are added indicate that certain features exhibit unacceptably high latency while other features are still operating within acceptable parameters. Therefore, the maximum value assigned to the number of site collections is not absolute, but is calculated based on an expected set of usage characteristics in which overall farm performance would be acceptable at the given limit under most circumstances.

If other services are operating under parameters that are higher than those used for limits testing, the maximum effective limits of other services will be reduced. Therefore, it is important to execute rigorous

capacity management and scale testing exercises for specific deployments to establish effective limits for that environment.

For more information about boundaries and limits and how they affect the capacity management process, see SharePoint Server 2010 capacity management: Software boundaries and limits.

Key differences: SharePoint Server 2010 versus Office SharePoint Server 2007

SharePoint Server 2010 offers a richer set of features and a more flexible topology model than earlier versions. Before you use this more complex architecture to deliver more powerful features and functionality to users, you must carefully consider their effect upon your farm's capacity and performance.

In Office SharePoint Server 2007, there were four major services that you could enable in SSPs (Shared Service Providers): Search Service, Excel Calculation Service, User Profile Service, and the Business Data Catalog Service. Additionally, there was a relatively smaller set of clients that could directly interface with Office SharePoint Server 2007.

In SharePoint Server 2010, there are more available services, known as SSAs (SharePoint Service Applications), and SharePoint Server 2010 offers a much broader range of client applications that can interact with the farm, including several new Office applications, mobile devices, designer tools, and browsers. Some examples of how expanded client interactions impact the capacity considerations include the following:

- SharePoint Server 2010 includes social applications that integrate with Outlook, which enable Outlook 2010 clients to display information about e-mail recipients that is pulled from the SharePoint Server 2010 farm when e-mail messages are viewed in the Outlook client. This introduces a new set of traffic patterns and server load for which should be accounted.
- Some new Microsoft Office 2010 client capabilities automatically refresh data against the SharePoint Server 2010 farm, even when the client applications are open but are not actively being used. Such clients as SharePoint Workspace and OneNote will also introduce some new traffic patterns and server load for which should be accounted.
- SharePoint Server 2010 new Web interactivity capabilities, such as Office Web Apps, which enable editing of Office files directly from the browser, use AJAX calls that introduce some new traffic patterns and server load which should be considered.

In Office SharePoint Server 2007, the primary client used to interact with the server was the Web browser. Given the richer feature set in SharePoint Server 2010, the overall requests per second (RPS) is expected to grow. Further, the percentage of requests coming from the browser is expected to be smaller than in Office SharePoint Server 2007, which makes room for the growing percent of new traffic coming from other clients as they are widely adopted throughout the organization.

Additionally, SharePoint Server 2010 introduces new functionality such as native embedded video support which can add stress to the farm. Some functionality has also been expanded to support a larger scale than previous versions.

The following section describes these client interactions, services and features, and their overall performance and capacity implications on the system that you should consider when you design your solution.

For more information about how to upgrade to SharePoint Server 2010, see Upgrading to SharePoint Server 2010.

Services and features

The following table provides a simplified high level description of the resource requirements for the different services on each tier. Blank cells indicate that the service does not run on or impact that tier.

X – Indicates minimal or insignificant cost on the resource. The service can share this resource with other services.

XX – Indicates medium cost on the resource. The service could share this resource with other services that have minimal impact.

XXX – Indicates high cost on the resource. The service should generally not share this resource with other services.

For more information about how to plan SQL Server databases, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

For a list of capacity management articles that are available for many specific SharePoint Server 2010 services and features (more articles will be added as they become available), see Performance and capacity test results and recommendations (SharePoint Server 2010).

Service Application	Web server CPU	Web server RAM	Application server CPU		Application server RAM	SQL Server CPU	SQL Server IOPS	SQL Server storage
SharePoint Foundation Service	XXX	XXX				XX	XXX	XXX
Central Admin service			XX		XX	X	X	X
Logging Service *	XX	XX				XX	XXX	XXX
SharePoint Search Service	XXX	XXX	XXX		XXX	XXX	XXX	XXX
Word Viewing Service Application *	X	X	XXX		XX			
PowerPoint	XX	XX	XXX		XX			

Service Application	Web server CPU	Web server RAM	Application server CPU		Application server RAM	SQL Server CPU	SQL Server IOPS	SQL Server storage
Service *								
Excel Calculation Service	XX	X	XX		XXX			
Visio Service *	X	X	XXX		XXX	X	X	X
Access Service *	X	X	XXX		XX	X	X	X
User Profile Service	X	XX	XX		XX	XXX	XXX	XX
Managed Metadata Service *	X	XX	XX		XX	X	X	XX
Web Analytics Service *	X	X				XXX	XXX	XXX
Business Connection Service *	XX	XX	XXX		XXX			
InfoPath Forms Service	XX	XX	XX		XX	X	X	X
Word Conversion Service	X	X	XXX		XX	X	X	X
PerformancePoint Service Application *	XX	XX	XXX		XXX	X	X	X
Project Service *	X	X	X		X	XXX	XXX	XX
Sandboxed Solutions *	X	X	XXX		XXX			
Workflow capabilities *	XXX	XXX						
Timer Service	XX	XX	XX		XX			
PowerPivot *	X	X	XXX		XXX	XX	XX	XXX

Note:

An asterisk (*) indicates a new service in SharePoint Server 2010.

- **SharePoint Foundation Service** The core SharePoint service for content collaboration. In large SharePoint Server 2010 deployments, we recommend that you allocate redundant Web servers based on expected traffic load, properly size the SQL Server–based computers that service the content databases, and properly allocate storage based on the size of the farm.
- **Central Admin Service** The administration service. This service has relatively small capacity requirements. We recommend that you enable this service on multiple farm servers to ensure redundancy.
- **Logging Service** The service that records usage and health indicators for monitoring purposes. This is a write-intensive service, and can require relatively large disk space depending on the number of indicators and the frequency at which they are logged. In large SharePoint Server 2010 deployments, we recommend that you isolate the usage database from the content databases on different SQL Server–based computers.
- **SharePoint Search Service Application** The shared service application that provides indexing and querying capabilities. Generally this is a relatively resource intensive service, that can scale to serve very large content deployments. In large SharePoint Server 2010 deployments where enterprise search is very important, we recommend that you use a separate "service farm" to host search service applications, with dedicated database resources, use multiple application servers servicing specific search functions (crawl or query), and dedicated target Web servers on the content farms to ensure acceptable throughput for crawling and querying. You can also enable the FAST Service Applications as your Search Service Application. Choose to create one or more FAST Search Connectors for indexing content with FAST Search Server 2010 for SharePoint and create another FAST Search Query (SSA) for querying content that is crawled by the FAST Search Connectors.
- **Word Viewing Service Application** Enabling this service lets you view Word documents directly from the browser. This service is added when you install Office Web Apps in addition to SharePoint Server 2010. This service requires an application server to prepare the original files for browser viewing. In large SharePoint Server 2010 deployments, we recommend that you scale out the service to multiple application servers for redundancy and throughput.

  Note:
  Browser editing for Word and OneNote are enabled when you install Office Web Apps on the SharePoint Server 2010 farm. However, this feature runs on the farm Web servers and does not use any service applications.

- **PowerPoint Service Application** This service displays and lets users edit PowerPoint files directly in the browser, and also enables you to broadcast and share live PowerPoint presentations. This service is added when you install Office Web Apps on SharePoint Server 2010. This service requires an application server to prepare the original files for browser viewing. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, we recommend that you deploy multiple application servers to ensure acceptable redundancy and throughput, and add more Web servers when PowerPoint Broadcast is frequently used as well.

- **Excel Calculation Service Application** This service displays Excel worksheets directly in the browser and performs Excel calculations on the server. It also enables editing of worksheets directly from the browser when you install Office Web Apps on SharePoint Server 2010. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, we recommend that you allocate a sufficient number of application servers that have sufficient RAM to ensure acceptable performance and throughput.
- **PowerPivot for SharePoint** The service to display PowerPivot enabled Excel worksheets directly from the browser. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, we recommend that you allocate a sufficient number of application servers that have sufficient RAM and CPU to ensure acceptable performance and throughput. For more information, see Hardware and Software Requirements (PowerPivot for SharePoint).
- **Visio Service Application** The service to display dynamic Visio diagrams directly in the browser. This service has a dependency on the Session State Service Application, which requires a relatively small SQL Server database. The Visio service requires an application server to prepare the original Visio files for browser viewing. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, we recommend that you scale out the service to multiple application servers that have sufficient CPU and RAM to ensure acceptable performance and throughput.
- **Access Service Application** The service to host Access solutions inside SharePoint Server 2010. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, we recommend that you scale out to multiple application servers that have sufficient RAM for acceptable performance and throughput. The Access service uses SQL Reporting Services, which will require a SQL Server database that can be co-located with other databases.
- **User Profile Service Application** The service that powers the social scenarios in SharePoint Server 2010 and enables My Sites, Tagging, Notes, Profile sync with directories and other social capabilities. The profile service requires three relatively resource intensive databases: the synchronization, Profile, and Social Tagging databases. This service is dependent on the Managed Metadata Service Application. In large SharePoint Server 2010 deployments, you should consider distributing this service to a shared services farm, and correctly size the database server tier to ensure acceptable performance of the common transactions and directory synchronization jobs.
- **Managed Metadata Service Application** The service that powers the central metadata store and allows the syndication of content types across the enterprise. The service can be federated to a dedicated services farm. It requires a database that can be co-located with other databases.
- **Web Analytics Service Application** The service that aggregates and stores statistics on the usage characteristics of the farm. This service has relatively high SQL Server resource and storage demands. The service can be federated to a dedicated services farm. In large SharePoint Server 2010 deployments, we recommend that you isolate the Web Analytics databases from other very important or resource intensive databases by hosting them on different database servers.
- **Business Connection Service Application** The service that enables the integration of various organizational line-of-business applications together with SharePoint Server 2010. This service requires an application service to maintain data connections to external resources. In large

SharePoint Server 2010 deployments where this is a frequently used capability, we recommend that you allocate a sufficient number of application servers that have sufficient RAM for acceptable performance.

- **InfoPath Forms Service Application** The service that enables browser-based forms in SharePoint Server 2010 and the integration with the InfoPath client application for form creation. This service requires an application server and has a dependency on the Session State Service Application, which requires a relatively small database. This service can be co-located with other services and has relatively small capacity requirements that can grow depending on the frequency of use of this capability.
- **Word Automation Service Application** The service that enables conversion of Word files from one format, such as .doc, to another format, such as .docx or .pdf. This service requires an application server. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, we recommend that you scale out the service to multiple application servers that have sufficient CPU resources to achieve acceptable conversion throughput. This service also requires a relatively small database to maintain the queue of conversion jobs.
- **PerformancePoint Service Application** The service that enables PerformancePoint BI capabilities in SharePoint Server 2010 and enables you to create analytic visualizations. This service requires an application server and a database. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, we recommend that you allocate sufficient RAM to the application servers for acceptable performance and throughput.
- **Project Service Application** The service that enables all the Microsoft Project Server 2010 planning and tracking capabilities in addition to SharePoint Server 2010. This service requires an application server and a relatively resource intensive database. In large SharePoint Server 2010 deployments where this is a frequently used capability, you should dedicate a database server for the Project Server database and even consider a dedicated SharePoint Server 2010 farm for the Project Server management solutions.
- **Timer Service** The process responsible of executing the various scheduled tasks on the different servers in the farm. There are various timer jobs that the system executes, some running on all farm servers, and some running only on specific servers depending on the server's role. Some of these timer jobs are resource intensive and can potentially create load on both the local server and the database servers, depending on their activity and how much content they are operating against. In large SharePoint Server 2010 deployments where timer jobs can potentially impact end-user latency, we recommend that you dedicate a server to isolate the execution of the more resource intensive jobs.
- **Workflow** The capability that enables integrated workflows in SharePoint Server 2010, and executes workflows on the Web server. Resource utilization is dependent on the complexity of the workflows and the total number of events they handle. In large SharePoint Server 2010 deployments where this is a frequently used capability, you should consider adding Web servers or isolating a server to handle only the workflow timer service to ensure end-user traffic is not affected and that workflow operations are not delayed.

- **Sandboxed Solutions** The service that enables isolation of custom code to dedicated farm resources. In large SharePoint Server 2010 deployments where this becomes a frequently used capability, you should consider dedicating additional Web servers if custom code begins to impact server performance.

New client applications interactions with SharePoint Server 2010

This section describes some new client-server interactions that SharePoint Server 2010 supports and their capacity planning implications.

The following table provides a simplified high level description of the typical load that these new capabilities introduce on the system:

X – Indicates minimal or insignificant load on the system's resources

XX – Indicates medium load on the system's resources

XXX – Indicates high load on the system's resources

Client	Traffic	Payload
Office Web Apps	XXX	XX
PowerPoint Broadcast	XXX	X
Word and PowerPoint 2010 client application	XX	X
OneNote client application	XXX	XXX
Outlook Social Connector	XX	XX
SharePoint Workspace	XXX	XX

- **Office Web Apps** Web viewing and editing of Word, PowerPoint, Excel, and OneNote files is a subset of browser requests, with slightly different traffic characteristics, this kind of interaction introduces a relatively high load of traffic necessary for enabling capabilities like co-authoring. In large SharePoint Server 2010 deployments where these capabilities are enabled, you should expect additional load on the Web servers.
- **PowerPoint Broadcast** The set of requests associated with viewing live PowerPoint presentation in the Web browser is another subset of browser requests. During live PowerPoint broadcast sessions, participating clients request changes from the service. In large SharePoint Server 2010 deployments where this is a frequently used capability, you should expect additional load on the Web servers.
- **Word and PowerPoint 2010 client applications** The Word and PowerPoint 2010 clients have new features that take advantage of the SharePoint Server 2010 farm. One example is document co-authoring, in which all client applications participating in a co-authoring session frequently

upload and download updates to and from the server. In large SharePoint Server 2010 deployments where this is a frequently used capability, you should expect additional load on the Web servers.

- **OneNote 2010 client application** The OneNote 2010 client interacts with the SharePoint Server 2010 farm in a similar manner to the previous OneNote version, and uses SharePoint Server 2010 to share and enable co-authoring of OneNote notebooks. This scenario introduces load on SharePoint Server 2010 even when the client is open but not actively being used. In large SharePoint Server 2010 deployments where this is a frequently used capability, you should expect additional load on the Web servers.
- **Outlook 2010 client application** Outlook 2010 has a new feature — the Outlook Social Connector — that takes advantage of the SharePoint Server 2010 farm (this component can be added to previous versions of Outlook as well). This feature enables you to view social activity requested from the SharePoint Server 2010 farm directly in e-mails. In large SharePoint Server 2010 deployments where this capability is enabled, you should expect additional load on the Web servers.
- **SharePoint Workspace** SharePoint Workspace 2010 clients has new features that take advantage of the SharePoint Server 2010 farm and enable you to sync Web sites, lists, and document libraries to the client for offline use. SharePoint Workspace 2010 regularly synchronizes with the attached server objects when the client is running, regardless of whether it is actively being used. In large SharePoint Server 2010 deployments where this is a frequently used capability, you should expect additional load on the Web servers.

SharePoint Server 2010 deployment key differentiators

Each SharePoint Server 2010 deployment has a key set of characteristics that will make it unique and different from other farms. These key differentiators can be described by these four major categories:

- **Specification** Describes the farm's hardware, and the farm topology and configuration.
- **Workload** Describes the demand on the farm, including the number of users and the usage characteristics.
- **Dataset** Describes content sizes and distribution.
- **Health and performance** Describes the farm's performance against latency and throughput targets.

Specifications

Hardware

Hardware is the computer's physical resources such as processors, memory, and hard disks. Hardware also includes physical network components such as NICs (Network Interface Cards), cables, switches, routers and hardware load balancers. Many performance and capacity issues can be resolved by

making sure that the correct hardware is being used. Conversely, a single misapplication of a hardware resource, such as insufficient memory on a server, can affect performance across the entire farm.

Topology

Topology is the distribution and interrelationships of farm hardware and components. There are two kinds of topology:

- **Logical topology** The map of software components such as services and features in a farm.
- **Physical topology** The map of servers and physical resources.

Typically, the number of users and usage characteristics determine the physical topology of a farm, and business requirements such as the need to support specific features for expected load drives the logical topology.

Configuration

We use the term configuration to describe software settings and how parameters are set. Also, configuration refers to caching, RBS, how configurable limits are set, and any part of the software environment that can be set or modified to meet specific requirements.

Workload

Workload defines the key operational characteristics of the farm, including the user base, concurrency, features that are being used, and the user agents or client applications that are used to connect with the farm.

Different SharePoint Server 2010 features have different associated costs on the farm's resources. Popularity of more costly features can potentially significantly impact the performance and the health of the system. Understanding your expected demand and usage characteristics will enable you to correctly size your implementation, and reduce the risk of constantly running the system in an unhealthy condition.

User Base

The user base of a SharePoint Server 2010–based application is a combination of the total number of users and how they are geographically distributed. Also, within the total user base, there are subgroups of users who might use given features or services more heavily than other groups. Concurrency of users is defined as the total percentage of users actively using the system at a given time. Indicators that define the user base include the number of total unique users and number of concurrent users.

Usage Characteristics

A farm's performance can be affected not only by the number of users interacting with the system, but also by their usage characteristics. Two organizations that have the same number of users might have significantly different requirements based on how often users access farm resources, and whether resource-intensive features and services are enabled on the farm. Indicators that describe the usage characteristics include the frequency of unique operations, the overall operational mix (the ratio of read and write operations and administrative operations), and the usage patterns and load against new features that are enabled on the farm (such as My Site Web sites, Search, Workflows, and Office Web Apps).

Dataset

The volume of content that is stored in the system and the characteristics of the architecture in which it is stored can have a significant effect on the overall health and performance of the system. Understanding the size, access frequency, and distribution of data will enable you to correctly size the storage in the system and prevent it from becoming the bottleneck that slows down user interactions with farm services and affects the end-user experience.

To correctly estimate and design the storage architecture of a SharePoint Server 2010–based solution, you need to know the volume of data that you will store on the system, and how many users are requesting data from different data sources. The volume of the content is an important element of sizing disk capacity, because it can influence the performance of other features, and can also potentially affect network latency and available bandwidth. Indicators that define the dataset include total size of content, total number of documents, total number of site collections, and average and maximum sizes of site collection.

Health and performance

SharePoint Server 2010 farm health is basically a simplified measurement or score that reflects the reliability, stability, and performance of the system. How well the farm performs against targets is basically dependent on the first three differentiators. The health and performance score can be tracked and described by a distillation of a set of indicators. For more information, see Monitoring and maintaining SharePoint Server 2010. These indicators include the system's uptime, end-user perceived latency, page failure rates, and resource utilization indicators (CPU, RAM).

Any significant change in hardware, topology, configuration, workload, or dataset can significantly vary the reliability and responsiveness of the system. The health score can be used to track performance over time and to assess how changing operating conditions or system modifications affect farm reliability.

Reference architectures

SharePoint Server 2010 is a complex and powerful product, and there is no one-size-fits-all architecture solution. Each SharePoint Server 2010 deployment is unique, and is defined by its usage and data characteristics. Every organization needs to perform a thorough capacity management process and effectively take advantage of the flexibility that the SharePoint Server 2010 system offers to customize a correctly sized solution that best satisfies the organizational needs.

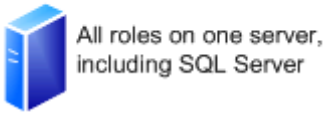
The concept of reference architectures is meant to describe and illustrate the different major categories of SharePoint Server 2010 deployments, and not to provide a recipe for architects to use to design their solutions. This section focuses on describing the vectors on which SharePoint Server 2010 deployments usually scale.

The architectures listed here are provided as a useful way to understand the general differentiators between these generic categories, and to distinguish them by general cost factors and scale of effort.

Single server deployment

The single server deployment architecture consists of one server that is running SharePoint Server 2010 and a supported version of SQL Server. This architecture might be appropriate for evaluation purposes, developers or for an isolated non-mission-critical departmental implementation with only a few users. However, we do not recommend its use for a production environment.

One-server farm



Small farm deployment

A small farm deployment consists of a single database server or cluster and one or two SharePoint Server 2010–based computers. The major architecture characteristics include limited redundancy and failover, and a minimal set of SharePoint Server 2010 capabilities enabled.

A small farm is useful to serve only limited deployments, with a minimal set of service applications enabled, a relatively small user base, a relatively low usage load (a few requests per minute up to very few requests per second), and a relatively small volume of data (10 or more gigabytes).

Two-tier small farm

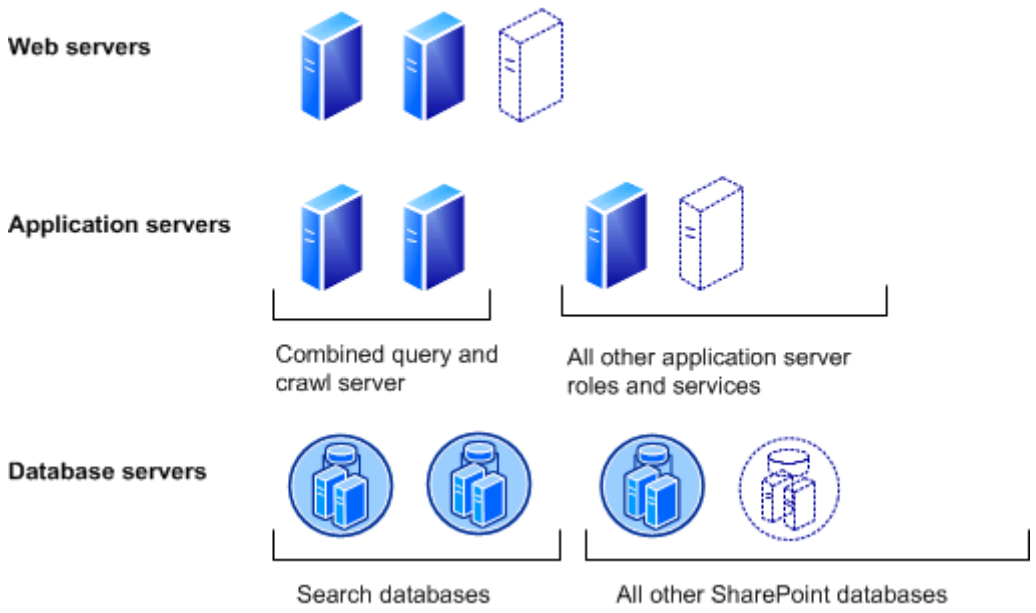
Medium farm deployment

This architecture introduces the breakdown of the topology into three tiers: dedicated Web servers, dedicated application servers, and one or more database servers or clusters. Separating the front end server tier from the application server tier allows greater flexibility in service isolation and helps balancing the load across the system.

This is the most common architecture, and includes a wide spectrum of service topologies and farm sizes. A medium farm deployment is useful to serve environments that have the following:

- Several service applications distributed across multiple servers. A typical set of features might include the Office Web Apps Service, User Profile Service, Managed Metadata Service, and Excel Calculation Service.
- A user base of tens of thousands of users and a load of 10 to 50 requests per second.
- A data store of one or two terabytes.

Medium farm

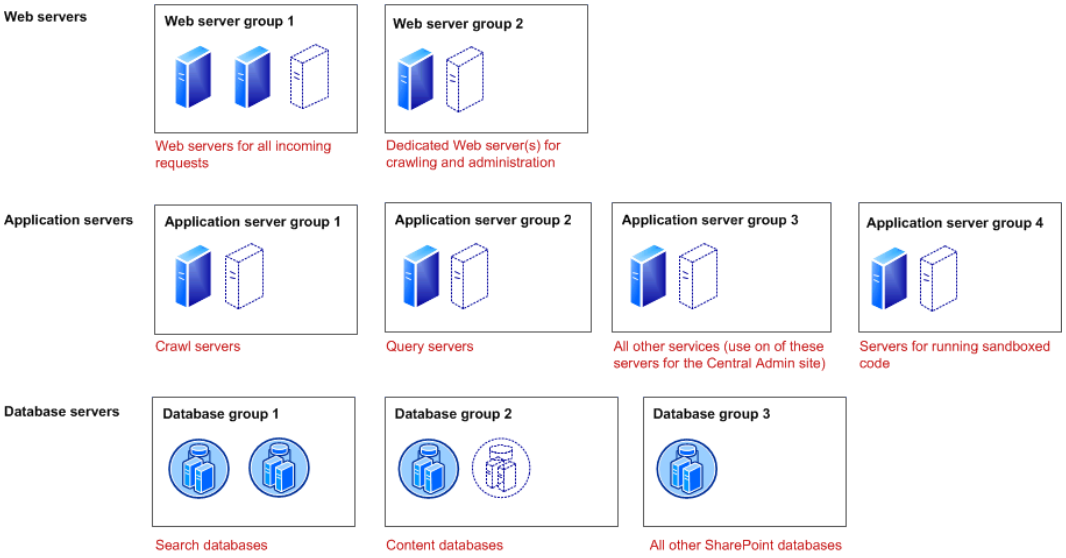


Large farm deployment

Large farm deployments introduce the breakdown of services and solutions across multiple farms, and additional scaling out the tiers on a single farm. Several SharePoint Server 2010 services can be deployed on a dedicated services farm that serves requests from multiple consuming farms. In these large architectures, there are typically Web servers, multiple application servers, depending on the usage characteristic of each of the local (non-shared) services, and multiple SQL Server–based servers or SQL Server clusters, depending on the content size and the application services databases that are enabled on the farm. Large farm architectures are expected to serve deployments that have the following:

- Several service applications federated and consumed from dedicated services farm, typically the User Profile Service, Search, Managed Metadata service, and Web Analytics.
- Most other service applications are enabled locally.

- A user base in the range of hundreds of thousands of users.
- A usage load in the range of hundreds of requests per second.
- A dataset in the range of ten or more terabytes.



See Also

Capacity planning for SharePoint Server 2010

Performance testing for SharePoint Server 2010

Monitoring and maintaining SharePoint Server 2010

SharePoint Server 2010 capacity management: Software boundaries and limits

Performance and capacity test results and recommendations (SharePoint Server 2010)

Performance and capacity technical case studies (SharePoint Server 2010)

Hardware and software requirements (SharePoint Server 2010)

Capacity planning for SharePoint Server 2010

This article describes how to plan the capacity of a Microsoft SharePoint Server 2010 farm. When you have a good appreciation and understanding of capacity planning and management, you can apply your knowledge to system sizing. Sizing is the term used to describe the selection and configuration of appropriate data architecture, logical and physical topology, and hardware for a solution platform. There is a range of capacity management and usage considerations that affect how you should determine the most appropriate hardware and configuration options.

Before you read this article, you should read Capacity management and sizing overview for SharePoint Server 2010.

In this article, we describe the steps you should take to undertake effective capacity management for your environment. Each step requires certain information for successful execution, and has a set of deliverables that you will use in the subsequent step. For each step, these requirements and deliverables are outlined in tables.

In this article:

Step 1: Model

Modeling your SharePoint Server 2010-based environment begins with analyzing your existing solutions and estimating the expected demand and targets for the deployment you are planning to set up. You start by gathering information about your user base, data requirements, latency and throughput targets, and document the SharePoint Server 2010 features you want to deploy. Use this section to understand what data you should collect, how to collect it, and how it can be used in subsequent steps.

Understand your expected workload and dataset

Proper sizing of a SharePoint Server 2010 implementation requires that you study and understand the demand characteristics that your solution is expected to handle. Understanding the demand requires that you be able to describe both the workload characteristics such as number of users and the most frequently used operations, and dataset characteristics such as content size and content distribution.

This section can help you understand some specific metrics and parameters you should collect and mechanisms by which they can be collected.

Workload

Workload describes the demand that the system will need to sustain, the user base and usage characteristics. The following table provides some key metrics that are helpful in determining your workload. You can use this table to record these metrics as you collect them.

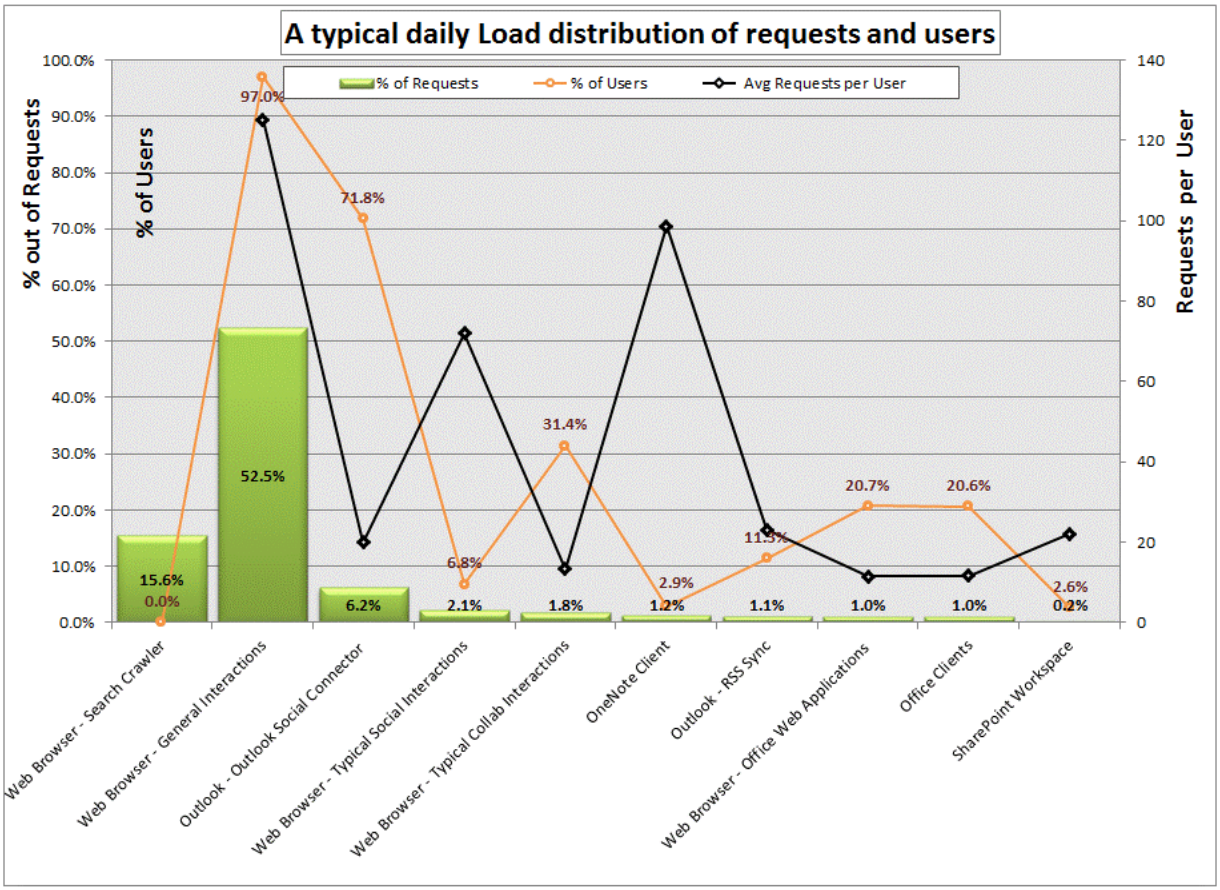
Workload Characteristics	Value	
Average daily RPS		
Average RPS at peak time		
Total number of unique users per day		
Average daily concurrent users		
Peak concurrent users at peak time		
Total number of requests per day		
Expected workload distribution	No. of Requests per day	%
Web Browser - Search Crawl		
Web Browser - General Collaboration Interaction		
Web Browser - Social Interaction		
Web Browser - General Interaction		
Web Browser - Office Web Apps		
Office Clients		
OneNote Client		
SharePoint Workspace		
Outlook RSS Sync		
Outlook Social Connector		
Other interactions(Custom Applications/Web services)		

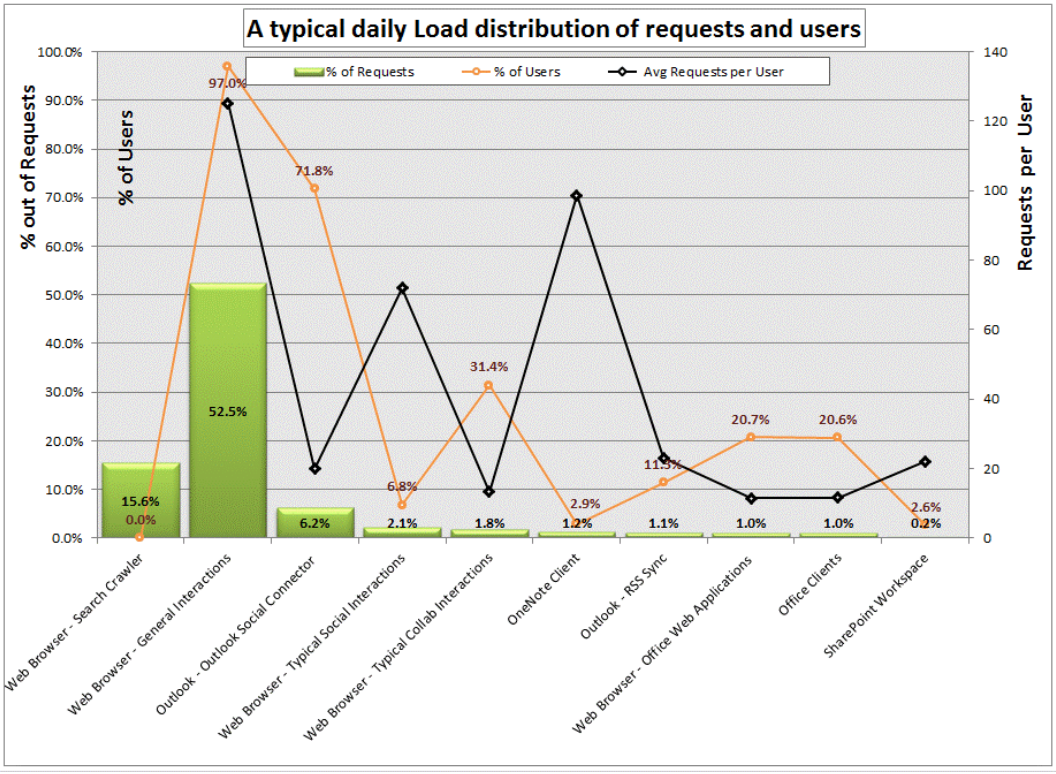
- **Concurrent users** – It is most common to measure the concurrency of operations executed on the server farm as the number of distinct users generating requests in a given time frame. The key metrics are the daily average and the concurrent users at peak load.

- **Requests per second (RPS)** – RPS is a commonly used indicator used to describe the demand on the server farm expressed in the number of requests processed by the farm per second, but with no differentiation between the type or size of requests. Every organization's user base generates system load at a rate that is dependent on the organization's unique usage characteristics. See the **Glossary** section in Capacity management and sizing overview for SharePoint Server 2010 for more information on this term.
- **Total daily requests** – Total daily requests is a good indicator of the overall load the system will need to handle. It is most common to measure all requests except authentication handshake requests (HTTP status 401) over a 24 hour period.
- **Total daily users** - Total users is another key indicator of the overall load the system will need to handle. This measurement is the actual number of unique users in a 24 hour period, not the total number of employees in the organization.

  Note:
  The number of total daily users can indicate the growth potential of the load on the farm. For example, if the number of potential users is 100k employees, 15k daily users indicates that the load may significantly grow over time as user adoption increases.
- **Workload Distribution** – Understanding the distribution of the requests based on the clients applications that are interacting with the farm can help predict the expected trend and load changes after migrating to SharePoint Server 2010. As users transition to more recent client versions such as Office 2010, and start using the new capabilities new load patterns, RPS and total requests are expected to grow. For each client we can describe the number of distinct users using it in a time frame of a day, and the amount of total requests that the client or feature generates on the server.

  For example, the chart below shows a snapshot of a live internal Microsoft environment serving a typical social solution. In this example, you can see that the majority of the load is generated by the search crawler and typical end user web browsing. You can also observe that there is significant load introduced by the new Outlook Social Connector feature (6.2 percent of the requests).

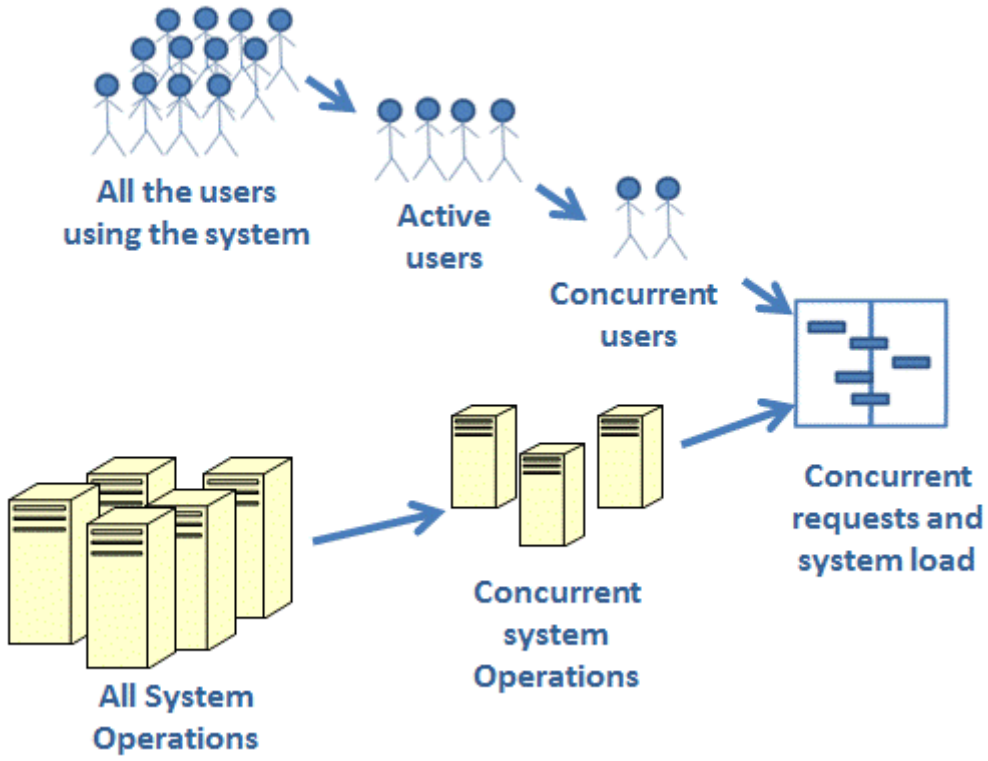

A typical daily Load distribution of requests and users
% of Requests
% of Users
Avg Requests per User
% out of Requests
% of Users
Requests per User
100.0%
90.0%
80.0%
70.0%
60.0%
50.0%
40.0%
30.0%
20.0%
10.0%
0.0%
140
120
100
80
60
40
20
0
97.0%
71.8%
31.4%
20.7%
20.6%
15.6%
0.0%
52.5%
6.2%
2.1%
1.8%
1.2%
2.9%
1.1%
11.5%
1.0%
1.0%
2.6%
0.2%
Web Browser - Search Crawler
Web Browser - General Interactions
Outlook - Outlook Social Connector
Web Browser - Typical Social Interactions
Web Browser - Typical Collab Interactions
OneNote Client
Outlook - RSS Sync
Web Browser - Office Web Applications
Office Clients
SharePoint Workspace



Estimating your production workload

In estimating the required throughput your farm needs to be able to sustain, begin with estimating the mix of transactions that will be used in your farm. Focus on analyzing the most frequently used transactions the system will serve, understanding how frequently they will be used and by how many users. That will help you validate later whether the farm can sustain such load in pre-production testing.

The following diagram describes the relationship of the workload and load on the system:



To estimate your expected workload, collect the following information:

- Identify user interactions such as typical web page browses, file downloads and uploads, Office Web Application views and edits in the browser, co-authoring interactions, SharePoint Workspace site syncs, Outlook Social Connections, RSS sync (in Outlook or other viewers), PowerPoint Broadcasts, OneNote shared notebooks, Excel Service shared workbooks, Access Service shared applications and others. See the **Services and Features** section of the article Capacity management and sizing overview for SharePoint Server 2010 for more information. Focus on the identifying the interactions that may be unique to your deployment, and recognize the expected impact of such load, examples can be significant use of InfoPath Forms, Excel Service Calculations and similar dedicated solutions.
- Identify system operations such as Search incremental crawls, daily backups, profile sync timer jobs, web analytics processing, logging timer jobs and others.
- Estimate the total number of users per day that are expected to utilize each capability, derive the estimated concurrent users and high level Requests per second, there are some assumptions you will be making such as present concurrency and the factor of RPS per concurrent users that is different across capabilities, you should use the workload table earlier in this section for your estimates. It is important to focus on peak hours, rather than average throughput. Planning for peak activity, you are able to proper size your SharePoint Server 2010-based solution.

If you have an existing Office SharePoint Server 2007 solution, you can mine the IIS log files or look to other Web monitoring tools you have to better understand some of the expected behaviors from the existing solution or see the instructions in the section below for more details. If you are not migrating from an existing solution, you should fill out the table using rough estimates. In later steps you will need to validate your assumptions and tune the system.

Analyzing your SharePoint Server 2010 IIS Logs

To discover key metrics about an existing SharePoint Server 2010 deployment, such as how many users are active, how heavily they are using the system, what kind of requests are coming in, and from what kind of clients they originate, it is necessary to extract data from ULS and IIS logs. One of the easiest ways to acquire this data is to use **Log Parser**, a powerful tool available free for download from Microsoft. Log Parser can read and write to a number of textual and binary formats, including all the IIS formats.

For detailed information about how to analyze SharePoint Server 2010 usage using Log Parser, read Analyzing Microsoft SharePoint Products and Technologies Usage (http://www.microsoft.com/downloads/details.aspx?familyid=f159af68-c3a3-413c-a3f7-2e0be6d5532e&displaylang=en&tm).

You can download Log Parser 2.2 at http://www.microsoft.com/downloads/details.aspx?FamilyID=890CD06B-ABF8-4C25-91B2-F8D975CF8C07&displaylang=en.

Dataset

Dataset describes the volume of content stored in the system and how it can be distributed in the data store. The following table provides some key metrics that are helpful in determining your dataset. You can use this table to record these metrics as you collect them.

Object	Value
DB size (in GB)	
Number of Content DBs	
Number of site collections	
Number of web apps	
Number of sites	
Search index size (# of items)	
Number of docs	
Number of lists	
Average size of sites	

Object	Value
Largest site size	
Number of user profiles	

- **Content size** – Understanding the size of the content that you expect to store in the SharePoint Server 2010 system is important for planning and architecting the system storage, and also for properly sizing the Search solution that will crawl and index this content. The content size is described in total disk space. If you are migrating content from an existing deployment you might find it simple to identify the total size that you will move; while planning you should leave room for growth over time based on the predicted trend.
- Total number of documents – Other than the data corpus size, it is important to track the overall number of items. The system reacts differently if 100 GB of data is composed of 50 files of 2 GB each versus 100,000 files of 1 KB each. In large deployments, the less stress there is on a single item, document or area of documents, the better performance will be. Widely distributed content like multiple smaller files across many sites and site collection is easier to serve then a single large document library with very large files.
- **Maximum site collection size** – It is important to identify what is the biggest unit of content that you will store in SharePoint Server 2010; usually it is an organizational need that prevents you from splitting that unit of content. Average size of all site collections and the estimated total number of site collections are additional indicators that will help you identify your preferred data architecture.
- **Service applications data characteristics** – In addition to analysing the storage needs for the content store, you should analyse and estimate the sizes of other SharePoint Server 2010 stores, including:
- Total size of the Search index
- The profile database total size based on the number of user in the profile store
- The social database total size based on the expected number of tags, colleagues and activities
- The metadata store size
- The size of the usage database
- The size of the Web Analytics data base

For more information on how to estimate database sizes, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

Setting Farm Performance and Reliability Targets

One of the deliverables of Step 1: Model is a good understanding of the performance and reliability targets that best fit the needs of your organization. A properly designed SharePoint Server 2010 solution should be able to achieve "four nines" (99.99%) of uptime with sub-second server responsiveness.

The indicators used to describe the performance and reliability of the farm can include:

- **Server availability** – Usually described by the percent of overall uptime of the system. You should track any unexpected downtime and compare the overall availability to the organizational target you set. The targets are commonly described by a number of nines (i.e. 99%, 99.9%, 99.99%)
- **Server responsiveness** – The time it takes the farm to serve requests is a good indicator to track the health of the farm. This indicator is usually named server side latency, and it is common to use the average or median (the 50th percentile) latency of the daily requests being served. The targets are commonly described in sub seconds or seconds. Note that if your organization has a target to serve pages from SharePoint Server 2010 in less than two seconds, then the server side goal needs to be sub seconds to leave time for the page to reach the client over the network and time to render in the browser. Also in general longer server response times are an indication of an unhealthy farm, as this usually as an impact on throughput and rarely can RPS keep up if you spend more than a second on the server on most requests
- **Server spikiness** – Another good server side latency indicator worth tracking is the behaviour of the slowest 5% of all requests. Slower requests are usually the requests that hit the system when it is under higher load or even more commonly, requests that are impacted by less frequent activity that occur while users interact with the system; a healthy system is one that has the slowest requests under control as well. The target here is similar to Server Responsiveness, but to achieve sub-second response on server spikiness, you will need to build the system with a lot of spare resources to handle the spikes in load.
- **System resource utilization** – Other common indicators used to track the health of the system are a collection of system counters that indicate the health of each server in the farm topology. The most frequently used indicators to track are % CPU utilization and Available Memory; however, there are several additional counters that can help identify a non-healthy system; more details can be found in Step 5: Monitor and Maintain.

Step 2: Design

Now that you have finished collecting some facts or estimates on the solution you need to deliver, you are ready to start the next step of designing a proposed architecture that you predict will be able to sustain the expected demand.

By the end of this step you should have a design for your physical topology and a layout for your logical topology, so you should be able to go ahead with any necessary purchase orders.

The hardware specifications and the number of machines you layout are tightly related, to handle a specific load there are several solutions you can choose to deploy. It is common to either use a small set of strong machines (scale up) or a larger set of smaller machines (scale out); each solution has its advantages and disadvantages when it comes to capacity, redundancy, power, cost, space, and other considerations.

We recommend that you begin this step by determining your architecture and topology. Define how you plan to layout the different farms and the different services in each farm, and then pick the hardware specifications for each of the individual servers in your design. You can also execute this process by

identifying the hardware specifications you are expected to deploy (many organizations are constrained to a certain company standard) and then define your architecture and topology.

Use the following table to record your design parameters. The data included is sample data, and should not be used to size your farm. It is intended to demonstrate how to use this table for your own data.

Role	Type (Standard or virtual)	# of machines	Procs	RAM	IOPS need	Disk size OS+Log	Data drive
Web servers	Virtual	4	4 cores	8	N/A	400 GB	N/A
Content database server	Standard	1 cluster	4 quad-core 2.33 (GHz)	48	2k	400 GB	20 disks of 300GB @ 15K RPM
Application servers	Virtual	4	4 cores	16	N/A	400 GB	N/A
Search Crawl Target Web server	Virtual	1	4 cores	8	N/A	400 GB	N/A
Search Query server	Standard	2	2 quad-core 2.33 (GHz)	32	N/A	400 GB	500 GB
Search Crawler server	Standard	2	2 quad-core 2.33 (GHz)	16	400	400 GB	N/A
Search Crawl database server	Standard	1 cluster	4 quad-core 2.33 (GHz)	48	4k (tuned for read)	100 GB	16 disks of 150GB @ 15K RPM
Search Property Store database + Administration database server	Standard	1 cluster	4 quad-core 2.33 (GHz)	48	2k (tuned for write)	100 GB	16 disks of 150GB @ 15K RPM

Determine your starting point architecture

This section describes how to select a starting point architecture.

When you deploy SharePoint Server 2010, you can choose from a range of topologies to implement your solution; you may deploy a single server or scale out many servers to a SharePoint Server 2010 farm with clustered or mirrored database servers and discreet application servers for various services. Later you will select the hardware configurations based on the requirements of each of the roles, based on your capacity, availability, and redundancy needs.

Start by reviewing the different reference architectures and figure out your farm structure, decide if you should split your solution across multiple farms, or federate some services, such as search, on a dedicated farm. See the **Reference Architectures** section in Capacity management and sizing overview for SharePoint Server 2010 for more information.

SharePoint Server 2010 Technical Case Studies

Capacity management guidance for SharePoint Server 2010 includes a number of technical case studies of existing production environments that present a detailed description of existing SharePoint Server 2010-based production environments. Additional technical case studies will be published over time; these can serve as a reference on how to design a SharePoint Server 2010-based environment for specific purposes.

You can use these case studies as a reference while designing the architecture of your SharePoint Server 2010 solutions especially if you find the description of these deployment specific key differentiators similar to the demands and targets of the solution you are architecting.

These documents describe the following information for each documented case study:

- **Specifications**, such as hardware, farm topology and configuration;
- **Workload** including the user base, and the usage characteristics;
- **Dataset**, including contents sizes, content characteristics and content distribution
- **Health and performance** including a set of recorded indicators describing the farm's reliability and performance characteristics

For more information, download relevant documents from the Performance and capacity technical case studies page (http://technet.microsoft.com/en-us/library/cc261716(Office.14)aspx).

Select your hardware

Selecting the right specifications for the machines in your farm is a crucial step to ensure proper reliability and performance of your deployment, one key concept to keep in mind is that you should plan for peak load and peak hours; in other words, when your farm is operating under average load conditions, there should be enough resources available to handle the greatest expected demand while still hitting latency and throughput targets.

The core capacity and performance hardware features of servers reflect four main categories: processing power, disk performance, network capacity, and memory capabilities of a system.

Another thing to consider is using virtualized machines. A SharePoint Server 2010 farm can be deployed using virtual machines. Although it has not been found to add any performance benefits, it does provide manageability benefits. Virtualizing SQL Server-based computers is generally not recommended, but there may be certain benefits to virtualizing the Web server and application server tiers. For more information, see Virtualization planning (http://technet.microsoft.com/en-us/library/71c203cd-7534-47b0-9122-657d72ff0080(Office.14).aspx).

Hardware Selection Guidelines

Choosing Processors

SharePoint Server 2010 is only available for 64-bit processors. In general, more processors will enable you to serve higher demand.

In SharePoint Server 2010, individual Web servers will scale as you add more cores (we have tested up to 24 cores); the more cores the server has the more load it can sustain, all else being equal. In large SharePoint Server 2010 deployments, it is recommended to allocate either multiple 4-core Web servers (which can be virtualized), or fewer stronger (8-/16-/24-cores) Web servers.

Application servers' processor capacity requirements differ depending on the role of the server and the services it is running. Some SharePoint Server 2010 features demand greater processing power than others. For example, the SharePoint Search Service is highly dependent on the processing power of the application server. For more information on the resource requirements of SharePoint Server 2010 features and services, see Services and Features earlier in this document.

The processor capacity requirements for SQL Server also depend on the service databases that a SQL Server-based computer is hosting. For more information on the typical behavior and requirements of each database, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

Choosing Memory

Your servers will require varying amounts of memory, depending on server function and role. For example, servers running Search crawl components will process data more quickly if they have a large amount of memory because documents are read into memory for processing. Web servers that leverage many of the caching features of SharePoint Server 2010 may require more memory as well.

In general, Web server memory requirements are highly dependent on the number of application pools enabled in the farm and the number of concurrent requests being served. In most production SharePoint Server 2010 deployments, it is recommended to allocate at least 8 GB RAM on each Web server, with 16 GB recommended for servers with higher traffic or deployments with multiple application pools set up for isolation.

Application servers' memory requirements differ as well; some SharePoint Server 2010 features have greater memory requirements on the application tier than others. In most production SharePoint Server 2010 deployments it is recommended to allocate at least 8 GB RAM on each application server; 16 GB, 32 GB and 64 GB application servers are common when many application services are enabled on the

same server, or when services that are highly dependent on memory, such as the Excel Calculation Service and SharePoint Server 2010 Search Service, are enabled.

The memory requirements of database servers are tightly dependent on the database sizes. For more information on choosing memory for your SQL Server-based computers, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

Choosing Networks

In addition to the benefit offered to users if clients have fast data access through the network, a distributed farm must have fast access for inter-server communication. This is particularly true when you distribute services across multiple servers or federate some of the services to other farms. There is significant traffic in a farm across the Web server tier, the application server tier, and the database server tier, and network can easily become a bottleneck under certain conditions like dealing with very large files or very high loads.

Web servers and application servers should be configured with at least two network interface cards (NICs): one NIC to handle end-user traffic and the other to handle the inter-server communication. Network latency between servers can have a significant impact on performance, so it is important to maintain less than 1 millisecond of network latency between the Web server and the SQL Server-based computers hosting the content databases. The SQL Server-based computers that host each service application database should be as close as possible to the consuming application server as well. The network between farm servers should have at least 1 Gbps of bandwidth.

Choosing Disks and Storage

Disk management is not simply a function of providing adequate space for your data. You must assess the on-going demand and growth, and ensure that the storage architecture is not slowing the system down. You should always ensure that you have at least 30 percent additional capacity on each disk, above your highest data requirement estimate, to leave room for future growth. Additionally, in most production environments, disk speed (IOps) is crucial to providing sufficient throughput to satisfy the servers' storage demands. You must estimate the amount of traffic (IOps) the major databases will require in your deployment and allocate enough disks to satisfy that traffic.

For more information on how to choose disks for database servers, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

The Web and application servers have storage requirements as well. In most production environments, it is recommended to allocate at least 200 GB disk space for OS and temp and 150 GB of disk space for logs.

Step 3: Pilot, Test and Optimize

The testing and optimization stage is a critical component of effective capacity management. You should test new architectures before you deploy them to production and you should conduct acceptance testing in conjunction with following monitoring best practices in order to ensure the architectures you design achieve the performance and capacity targets. This allows you to identify and optimize potential bottlenecks before they impact users in a live deployment. If you are upgrading from

an Office SharePoint Server 2007 environment and plan to make architectural changes, or are estimating user load of the new SharePoint Server 2010 features, then testing particularly important to make sure your new SharePoint Server 2010-based environment will meet performance and capacity targets.

Once you have tested your environment, you can analyze the test results to determine what changes need to be made in order to achieve the performance and capacity targets you established in Step 1: Model.

These are the recommended sub steps you should follow for pre-production:

- Create the test environment that mimics the initial architecture you designed in Step 2: Design.
- Populate the storage with the dataset or part of the dataset that you've identified in Step 1: Model.
- Stress the system with synthetic load that represents the workload you've identified in Step 1: Model.
- Run tests, analyze results, and optimize your architecture.
- Deploy your optimized architecture in your data center, and roll out a pilot with a smaller set of users.
- Analyze the pilot results, identify potential bottlenecks, and optimize the architecture. Retest if needed.
- Deploy to the production environment.

Test

Testing is a critial factor in establishing the ability of your system design to support your workload and usage characteristics. See Performance testing for SharePoint Server 2010 for detailed information on testing your SharePoint Server 2010 deployment.

- Create a test plan
- Create the test environment
- Create Tests and Tools

Deploy the pilot environment

Before you deploy SharePoint Server 2010 to a production environment, it is important that you first deploy a pilot environment and thoroughly test the farm to ensure that it can meet capacity and performance targets for your expected peak load. We recommend that the pilot environment is first tested with synthetic load particularly for large deployments, and then stressed by a small set of live users and live content. The benefit of analyzing a pilot environment with a small set of live users is the opportunity to validate some of the assumptions you made about the usage characteristics and the content growth before you go fully into production.

Optimize

If you cannot meet your capacity and performance targets by scaling your farm hardware or making changes to the topology, you may need to consider revising your solution. For example, if your initial requirements were for a single farm for collaboration, Search and Social, you may need to federate some of the services such as search to a dedicated services farm, or split the workload across more farms. One alternative is to deploy a dedicated farm for social and another for team collaboration.

Step 4: Deploy

Once you have executed your final round of tests and confirmed that the architecture you have selected can achieve the performance and capacity targets you established in Step 1: Model, you can deploy your SharePoint Server 2010-based environment to production.

The appropriate rollout strategy will vary depending upon the environment and situation. While SharePoint Server 2010 deployment in general is outside the scope of this document, there are certain suggested activities that may come out of the capacity planning exercise. Here are some examples:

- **Deploying a new SharePoint Server 2010 farm:** The capacity planning exercise should have guided and confirmed plans for a design and deployment of SharePoint Server 2010. In this case, the rollout will be the first broad deployment of SharePoint Server 2010. It will require moving or rebuilding the servers and services that were used during the capacity planning exercises into production. This is the most straight-forward scenario because there aren't any upgrades or modifications needed to an existing farm.
- **Upgrading an Office SharePoint Server 2007 farm to SharePoint Server 2010:** The capacity planning exercise should have validated the design for a farm that can meet existing demands and scale up to meet increased demand and usage of a SharePoint Server 2010 farm. Part of the capacity planning exercise should have included test migrations to validate how long the upgrade process will take, whether any custom code needs to be modified or replaced, whether any third-party tools need updating, etc. At the conclusion of capacity planning you should have a validated design, and understanding of how much time it will take to upgrade, and a plan for how best to work through the upgrade process – for example, an in-place upgrade, or migrating content databases into a new farm. If you're doing an in-place upgrade then during capacity planning you may have found that additional or upgraded hardware will be needed, and considerations for downtime. Part of the output from the planning exercise should be a list of the hardware changes that are needed and a detailed plan for deploying the hardware changes to the farm first. Once the hardware platform that was validated during capacity planning is in place, you can move forward with the process of upgrading to SharePoint Server 2010.
- **Improving the performance of an existing SharePoint Server 2010 farm:** The capacity planning exercise should have helped you to identify the bottlenecks in your current implementation, devise ways to reduce or eliminate those bottlenecks, and validate an improved implementation that meets your business requirements for SharePoint Server 2010 services. There are different ways in which performance issues could have been resolved, from something as simple as reallocating services across existing hardware, upgrading existing hardware, or adding additional hardware and adding

additional services to it. The different approaches should be tested and validated during the capacity planning exercise, and then a deployment plan formulated depending on the results of that testing.

Step 5: Monitor and Maintain

To maintain system performance, you must monitor your server to identify potential bottlenecks. Before you can monitor effectively, you must understand the key indicators that will tell you if a specific part of your farm requires attention, and know how to interpret these indicators. If you find that your farm is operating outside the targets you have defined, you can adjust your farm by adding or removing hardware resources, modifying your topology, or changing how data is stored.

See Monitoring and maintaining SharePoint Server 2010 for a list of the settings that you can modify to monitor your environment in its early stages, which will help you determine if any changes are needed. Keep in mind that increasing your monitoring capabilities will affect the amount of disk space that your usage database will require. Once the environment is stable and this detailed monitoring is no longer required, you may want to reverse the settings below to their defaults.

For more information about health monitoring and troubleshooting using the health monitoring tools built into the SharePoint Server 2010 Central Administration interface, read the following:

Health monitoring (SharePoint Server 2010)

Solving problems and troubleshooting (http://technet.microsoft.com/en-us/library/ee748639(office.14).aspx)

See Also

Capacity management and sizing overview for SharePoint Server 2010

Performance testing for SharePoint Server 2010

Monitoring and maintaining SharePoint Server 2010

Health monitoring (SharePoint Server 2010)

Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

Performance testing for SharePoint Server 2010

This article describes how to test the performance of Microsoft SharePoint Server 2010. The testing and optimization stage is a critical component of effective capacity management. You should test new architectures before you deploy them to production and you should conduct acceptance testing in conjunction with following monitoring best practices in order to ensure the architectures you design achieve the performance and capacity targets. This allows you to identify and optimize potential bottlenecks before they impact users in a live deployment. If you are upgrading from an Microsoft Office SharePoint Server 2007 environment and plan to make architectural changes, or are estimating user load of the new SharePoint Server 2010 features, then testing particularly important to make sure your new SharePoint Server 2010-based environment will meet performance and capacity targets.

Once you have tested your environment, you can analyze the test results to determine what changes need to be made in order to achieve the performance and capacity targets you established in Step 1: Model of Capacity planning for SharePoint Server 2010.

These are the recommended sub steps you should follow for pre-production:

- Create the test environment that mimics the initial architecture you designed in Step 2: Design.
- Populate the storage with the dataset or part of the dataset that you've identified in Step 1: Model.
- Stress the system with synthetic load that represents the workload you've identified in Step 1: Model.
- Run tests, analyze results, and optimize your architecture.
- Deploy your optimized architecture in your data center, and roll out a pilot with a smaller set of users.
- Analyze the pilot results, identify potential bottlenecks, and optimize the architecture. Retest if needed.
- Deploy to the production environment.

Before you read this article, you should read Capacity management and sizing overview for SharePoint Server 2010.

In this article:

Create a Test Plan

Verify that your plan includes:

- Hardware that is designed to operate at expected production performance targets. Always measure the performance of test systems conservatively.

- If you have custom code or custom component, it is important that you test the performance of those components in isolation first to validate their performance and stability. After they are stable, you should test the system with those components installed and compare performance to the farm without them installed. Custom components are often a major culprit of performance and reliability problems in production systems.
- Know the goal of your testing. Understand ahead of time what your testing objectives are. Is it to validate the performance of some new custom components that were developed for the farm? Is it to see how long it will take to crawl and index a set of content? Is it to determine how many requests per second your farm can support? There can be many different objectives during a test, and the first step in developing a good test plan is deciding what your objectives are.
- Understand how to measure for your testing goal. If you are interested in measuring the throughput capacity of your farm for example, you will want to measure the RPS and page latency. If you are measuring for search performance then you will want to measure crawl time and document indexing rates. If your testing objective is well understood, that will help you clearly define what key performance indicators you need to validate in order to complete your tests.

Create the Test Environment

Once your test objectives have been decided, your measurements have been defined, and you have determined what the capacity requirements are for your farm (from steps 1 and 2 of this process), the next objective will be to design and create the test environment. The effort to create a test environment is often underestimated. It should duplicate the production environment as closely as possible. Some of the features and functionality you should consider when designing your test environment include:

- **Authentication** – Decide whether the farm will use Active Directory Domain Services (AD DS), forms-based authentication (and if so with what directory), claims-based authentication, etc. Regardless of which directory you are using, how many users do you need in your test environment and how are you going to create them? How many groups or roles are you going to need and how will you create and populate them? You also need to ensure that you have enough resources allocated to your authentication services that they don't become a bottleneck during testing.
- **DNS** – Know what the namespaces are that you will need during your testing. Identify which servers will be responding to those requests and make sure you've included a plan that has what IP addresses will be used by which servers, and what DNS entries you will need to create.
- **Load balancing** – Assuming you are using more than one server (which you normally would or you likely wouldn't have enough load to warrant load testing), you will need some kind of load balancer solution. That could be a hardware load balancing device, or you could use software load balancing like Windows NLB. Figure out what you will use and write down all of the configuration information you will need to get it set up quickly and efficiently. Another thing to remember is that load test agents typically try and resolve the address to a URL only once every 30 minutes. That means that you should not use a local hosts file or round robin DNS for load balancing because the test agents will likely end up going to the same server for every single request, instead of balancing around all available servers.

- **Test servers** – When you plan your test environment, you not only need to plan for the servers for the SharePoint Server 2010 farm, you also need to plan for the machines needed to execute the tests. Typically that will include 3 servers at a minimum; more may be necessary. If you are using Visual Studio Team System (Team Test Load Agent) to do the testing, one machine will be used as the load test controller. There are generally 2 or more machines that are used as load test agents. The agents are the machines that take the instructions from the test controller about what to test and issue the requests to the SharePoint Server 2010 farm. The test results themselves are stored on a SQL Server-based computer. You should not use the same SQL Server-based computer that is used for the SharePoint Server 2010 farm, because writing the test data will skew the available SQL Server resources for the SharePoint Server 2010 farm. You also need to monitor your test servers when running your tests, the same way as you would monitor the servers in the SharePoint Server 2010 farm, or domain controllers, etc. to make sure that the test results are representative of the farm you're setting up. Sometimes the load agents or controller can become the bottleneck themselves. If that happens then the throughput you see in your test is typically not the maximum the farm can support.
- **SQL Server** –In your test environment, follow the guidance in the sections "Configure SQL Server" and "Validate and monitor storage and SQL Server performance" in the article Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).
- **Dataset validation** – As you decide what content you are going to run tests against, remember that in the best case scenario you will use data from an existing production system. For example, you can back up your content databases from a production farm and restore them into your test environment, then attach the databases to bring the content into the farm. Anytime you run tests against made up or sample data, you run the risk of having your results skewed because of differences in your content corpus.

If you do have to create sample data, there are a few considerations to keep in mind as you build out that content:

- All pages should be published; nothing should be checked out
- Navigation should be realistic; don't build beyond what you would reasonably expect to use in production.
- You should have an idea of the customizations the production site will be using. For example, master pages, style sheets, JavaScript, etc. should all be implemented in the test environment as closely as possible to the production environment.
- Determine how many SharePoint groups and/or permission levels you are going to need, and how you are going to associate users with them.
- Figure out whether you'll need to do profile imports, and how long that will take.
- Determine whether you'll need Audiences, and how you'll create and populate them.
- Determine whether you need additional search content sources, and what you will need to create them. If you won't need to create them, determine whether you'll have network access to be able to crawl them.

- Determine whether you have enough sample data – documents, lists, list items, etc. If not, create a plan for how you will create this content.
- Have a plan for enough unique content to adequately test search. A common mistake is to upload the same document – maybe hundreds or even thousands of times – to different document libraries with different names. That can impact search performance because the query processor will spend an ordinate amount of time doing duplicate detection that it wouldn't otherwise have to in a production environment with real content.

Create Tests and Tools

After the test environment is functional, it is time to create and fine-tune the tests that will be used to measure the performance capacity of the farm. This section will at times make references specifically to Visual Studio Team System (Team Test Load Agent), but many of the concepts are applicable irrespective of which load test tool you use. For more information about Visual Studio Team System, see Visual Studio Team System at MSDN (http://msdn.microsoft.com/en-us/library/fda2bad5.aspx").

You can also use the SharePoint Load Test Kit (LTK) in conjunction with VSTS for load testing of SharePoint 2010 farms. The Load Test Kit generates a Visual Studio Team System 2008 load test based on Windows SharePoint Services 3.0 and Microsoft Office SharePoint Server 2007 IIS logs. The VSTS load test can be used to generate synthetic load against SharePoint Foundation 2010 or SharePoint Server 2010 as part of a capacity planning exercise or a pre-upgrade stress test.

The Load Test Kit is included in the Microsoft SharePoint 2010 Administration Toolkit v1.0, available from the Microsoft Download Center (http://www.microsoft.com/downloads/details.aspx?FamilyId=718447d8-0814-427a-81c3-c9c3d84c456e&displaylang=en).

A key criterion to the success of the tests is to be able to effectively simulate a realistic workload by generating requests across a wide range of the test site data, just as users would access a wide range of content in a production SharePoint Server 2010 farm. In order to do that, you will typically need to construct your tests such that they are data driven. Rather than creating hundreds of individual tests that are hard-coded to access a specific page, you should use just a few tests that use data sources containing the URLs for those items to dynamically access that set of pages.

In Visual Studio Team System (Team Test Load Agent), a data source can come in a variety of formats, but a CSV file format is often easiest to manage and transport between development and test environments. Keep in mind that creating CSV files with that content might require the creation of custom tools to enumerate the SharePoint Server 2010-based environment and record the various URLs being used.

You may need to use tools for tasks like:

- Creating users and groups in Active Directory or other authentication store if you're using forms based authentication
- Enumerating URLs for sites, lists and libraries, list items, documents, etc. and putting them into CSV files for load tests

- Uploading sample documents across a range of document libraries and sites
- Creating site collections, webs, lists, libraries, folders and list items
- Creating My Sites
- Creating CSV files with usernames and passwords for test users; these are the user accounts that the load tests will execute as. There should be multiple files so that, for example, some contain only administrator users, some contain other users with elevated privileges (like author / contributor, hierarchy manager, etc.), and others are only readers, etc.
- Creating a list of sample search keywords and phrases
- Populating SharePoint groups and permission levels with users and Active Directory groups (or roles if you are using forms based authentication)

When creating the web tests, there are other best practices that you should observe and implement. They include:

- Record simple web tests as a starting point. Those tests will have hard-coded values in them for parameters like URL, ID's, etc. Replace those hard-coded values with links from your CSV files. Data binding those values in Visual Studio Team System (Team Test Load Agent) is extremely easy.
- Always have validation rules for your test. For example, when requesting a page, if an error occurs you will often get the error.aspx page in response. From a web test perspective it appears as just another positive response, because you get an HTTP status code of 200 (successful) in the load test results. Obviously an error has occurred though so that should be tracked differently. Creating one or more validation rules allows you to trap when certain text is sent as a response so that the validation fails and the request is marked as a failure. For example, in Visual Studio Team System (Team Test Load Agent) a simple validation rule might be a ResponseUrl validation – it records a failure if the page that is rendered after redirects is not the same response page that was recorded in the test. You could also add a FindText rule that will record a failure if it finds the word "access denied", for example, in the response.
- Use multiple users in different roles for tests. Certain behaviors such as output caching work differently depending on the rights of the current user. For example, a site collection administrator or an authenticated user with approval or authoring rights will not get cached results because we always want them to see the most current version of content. Anonymous users, however, will get the cached content. You need to make sure that your test users are in a mix of these roles that approximately matches the mix of users in the production environment. For example, in production there are probably only two or three site collection administrators, so you should not create tests where 10% of the page requests are made by user accounts that are site collection administrators over the test content.
- Parsing dependent requests is an attribute of a Visual Studio Team System (Team Test Load Agent) that determines whether the test agent should attempt to retrieve just the page, or the page and all associated requests that are part of the page, such as images, style sheets, scripts, etc. When load testing, we usually ignore these items for a few reasons:
    - After a user hits a site the first time these items are often cached by the local browser

    - These items don't typically come from SQL Server in a SharePoint Server 2010-based environment. With BLOB caching turned on, they are instead served by the Web servers so they don't generate SQL Server load.

If you regularly have a high percentage of first time users to your site, or you have disabled browser caching, or for some reason you don't intend to use the blob cache, then it may make sense to enable parsing dependent requests in your tests. However this is really the exception and not the rule of thumb for most implementations. Be aware that if you do turn this on it can significantly inflate the RPS numbers reported by the test controller. These requests are served so quickly it may mislead you into thinking that there is more capacity available in the farm than there actually is.

- Remember to model client application activity as well. Client applications, such as Microsoft Word, PowerPoint, Excel and Outlook generate requests to SharePoint Server 2010 farms as well. They add load to the environment by sending the server requests such as retrieving RSS feeds, acquiring social information, requesting details on site and list structure, synchronizing data, etc. These types of requests should be included and modeled if you have those clients in your implementation.
- In most cases a web test should only contain a single request. It's easier to fine-tune and troubleshoot your testing harness and individual requests if the test only contains a single request. Web tests will typically need to contain multiple requests if the operation it is simulating is composed of multiple requests. For example, to test this set of actions you will need a test with multiple step: checking out a document, editing it, checking it in and publishing it. It also requires reserving state between the steps – for example, the same user account should be used to check it out, make the edits, and check it back in. Those multi-step operations that require state to be carried forward between each step are best served by multiple requests in a single web test.
- Test each web test individually. Make sure that each test is able to complete successfully before running it in a larger load test. Confirm that all of the names for web applications resolve, and that the user accounts used in the test have sufficient rights to execute the test.

Web tests comprise the requests for individual pages, uploading documents, view list items, etc. All of these are pulled together in load tests. A load test is where you plug in all of the different web tests that are going to be executed. Each web test can be given a percentage of time that it will execute – for example, if you find that 10% of requests in a production farm are search queries, then in the load test you would configure a query web test to run 10% of the time. In Visual Studio Team System (Team Test Load Agent), load tests are also how you configure things like the browser mix, network mix, load patterns, and run settings.

There are some additional best practices that should be observed and implemented for load tests:

- Use a reasonable read/write ratio in your tests. Overloading the number of writes in a test can significantly impact the overall throughput of a test. Even on collaboration farms, the read/write ratios tend to have many more reads than writes. For more information, see the Performance and capacity technical case studies page (http://technet.microsoft.com/en-us/library/cc261716(Office.14)aspx).

- Consider the impact of other resource intensive operations and decide whether they should be occurring during the load test. For example, operations like backup and restore are not generally done during a load test. A full search crawl may not be usually run during a load test, whereas an incremental crawl may be normal. You need to consider how those tasks will be scheduled in production – will they be running at peak load times? If not, then they should probably be excluded during load testing, when you are trying to determine the maximum steady state load you can support for peak traffic.
- Don't use think times. Think times are a feature of Visual Studio Team System (Team Test Load Agent) that allow you to simulate the time that users pause between clicks on a page. For example a typical user might load a page, spend three minutes reading it, then click a link on the page to visit another site. Trying to model this in a test environment is nearly impossible to do correctly, and effectively doesn't add value to the test results. It's difficult to model because most organizations don't have a way to monitor different users and the time they spend between clicks on different types of SharePoint sites (like publishing versus search versus collaboration, etc.). It also doesn't really add value because even though a user may pause between page requests, the SharePoint Server 2010-based servers do not. They just get a steady stream of requests that may have peaks and valleys over time, but they are not waiting idly as each user pauses between clicking links on a page.
- Understand the difference between users and requests. Visual Studio Team System (Team Test Load Agent) has load pattern where it asks you to enter the number of users to simulate. This doesn't have anything to do with application users, it's really just how many threads are going to be used on the load test agents to generate requests. A common mistake is thinking that if the deployment will have 5,000 users for example, then 5,000 is the number that should be used in Visual Studio Team System (Team Test Load Agent) – it is not! That's one of the many reasons why when estimating capacity planning requirements, the usage requirements should be based on number of requests per second and not number of users. In a Visual Studio Team System (Team Test Load Agent) load test, you will find that you can often generate hundreds of requests per second using only 50 to 75 load test "users".
- Use a constant load pattern for the most reliable and reproducible test results. In Visual Studio Team System (Team Test Load Agent) you have the option of basing load on a constant number of users (threads, as explained in the previous point), a stepped up load pattern of users, or a goal based usage test. A stepped load pattern is when you start with a lower number of users and then "step up" adding additional users every few minutes. A goal based usage test is when you establish a threshold for a certain diagnostic counter, like CPU utilization, and test attempts to drive the load to keep that counter between a minimum and maximum threshold that you define for it. However, if you are just trying to determine the maximum throughput your SharePoint Server 2010 farm can sustain during peak load, it is more effective and accurate to just pick a constant load pattern. That allows you to more easily identify how much load the system can take before starting to regularly exceed the thresholds that should be maintained in a healthy farm.

Each time you run a load test remember that it is changing data in the database. Whether that's uploading documents, editing list items, or just recording activity in the usage database, there will be

data that is written to SQL Server. To ensure a consistent and legitimate set of test results from each load test, you should have a backup available before you run the first load test. After each load test is complete the backup should be used to restore the content back to the way it was before the test was started.

See Also

Capacity management and sizing overview for SharePoint Server 2010

Capacity planning for SharePoint Server 2010

Monitoring and maintaining SharePoint Server 2010

Health monitoring (SharePoint Server 2010)

Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

Monitoring and maintaining SharePoint Server 2010

This article provides information about monitoring and performance counters for Microsoft SharePoint Server 2010 farms. To maintain SharePoint Server 2010 system performance, you must monitor your server to identify potential bottlenecks. Before you can monitor effectively, you must understand the key indicators that will tell you if a specific part of your farm requires attention, and know how to interpret these indicators. If you find that your farm is operating outside the targets you have defined, you can adjust your farm by adding or removing hardware resources, modifying your topology, or changing how data is stored.

The information in this section is intended to help administrators manually configure performance counters and other settings. For more information about health monitoring and troubleshooting using the health monitoring tools built into the SharePoint Central Administration interface, read the following articles:

- Health monitoring (SharePoint Server 2010)
- Solving problems and troubleshooting (SharePoint Server 2010)

Before you read this article, you should read Capacity management and sizing overview for SharePoint Server 2010.

In this article:

Configuring Monitoring

Below is a list of the settings that you can modify to monitor your environment in its early stages, which will help you determine if any changes are needed. Keep in mind that increasing your monitoring capabilities will affect the amount of disk space that your usage database will require. Once the environment is stable and this detailed monitoring is no longer required, you may want to reverse the settings below to their defaults.

Setting	Value	Notes
Event Log Flooding Protection	Disabled	The default value is **Enabled**. It can be disabled to collect as much monitoring data as possible. For normal operations, it should be enabled.

Setting	Value	Notes
Timer Job Schedule		
Microsoft SharePoint Foundation Usage Data Import	5 minutes	The default value is **30 minutes**. Lowering this setting imports the data into the usage database more frequently, and is particularly useful when troubleshooting. For normal operations, it should be 30 minutes.
Diagnostic Providers		
Enable all diagnostic providers	Enabled	The default value is **Disabled** except for the "Search Health Monitoring - Trace Events" provider. These providers collect health data for various features and components. For normal operations, you may want to revert to the default.
Set "job-diagnostics-performance-counter-wfe-provider" and "job-diagnostics-performance-counter-sql-provider" Schedule Intervals	1 minute	The default value is **5 minutes**. Lowering this setting can poll data more frequently, and is particularly useful when troubleshooting. For normal operations, it should be 5 minutes.
Miscellaneous		
Enable stack tracing for content requests	Enabled	The default value is **Disabled**. Enabling this setting allows diagnosis of content requests failures using the process stack trace. For normal operations, it should be disabled.
Enable the Developer Dashboard	Enabled	The default value is **Disabled**. Enabling this setting allows diagnosis of slow pages, or other problems by using the Developer Dashboard. For normal

Setting	Value	Notes
		operations, and once troubleshooting is no longer necessary, it should be disabled.
Usage Data Collection		
Content Import Usage Content Export Usage Page Requests Feature Use Search Query Use Site Inventory Usage Timer Jobs Rating Usage	Enabled	Enabling the logging of this set of counters allows you to collect more usage data across the environment and to better understand the traffic patterns in the environment.

Performance Counters

If you are making use of the usage database, then you can add the performance counters that assist you in monitoring and evaluating your farm's performance to the usage database, such that they are logged automatically at a specific interval (30 minutes by default). Given that, you can query the usage database to retrieve these counters and graph the results over time. Here's an example of using the **Add-SPDiagnosticsPerformanceCounter** PowerShell cmdlet to add the % Processor Time counter to the usage database. This only needs to be run on one of the Web servers:

```
Add-SPDiagnosticsPerformanceCounter -Category "Processor" -Counter "% Processor Time" -Instance "_Total" -WebFrontEnd
```

There are a number of generic performance counters that you should monitor for any server system. The following table outlines these performance counters.

Performance Counter	Description
Processor	You should monitor processor performance to ensure that all processor usage does not remain consistently high (over 80 percent) as this indicates that the system would not be able to handle any sudden surges of activity. And that in the common state, you will not see a domino effect if one component failure will bring the remaining components to a malfunctioning state. For

Performance Counter	Description
	example – if you have three Web servers, you should make sure the average CPU across all servers is under 60% so that if one fails, there is still room for the other two to pick up the extra load.
Network Interface	Monitor the rate at which data is sent and received via the network interface card. This should remain below 50 percent of network capacity.
Disks and Cache	There are a number of logical disk options that you should monitor regularly. The available disk space is essential in any capacity study, but you should also review the time that the disk is idle. Dependent on the types of applications or services you are running on your servers, you may review disk read and write times. Extended queuing for write or read function will affect performance. The cache has a major impact on read and write operations. You must monitor for increased cache failures.
Memory and Paging File	Monitor the amount of physical memory available for allocation. Insufficient memory will lead to excessive use of the page file and an increase in the number of page faults per second.

System Counters

The following table provides information on system objects and counters that you could add to the set of counters monitored in the usage database using the **SPDiagnosticPerformanceCounter** on a web server.

Objects and Counters	Description
Processor	
% Processor Time	This shows processor usage over a period of time. If this is consistently too high, you may find performance is adversely affected. Remember to count "Total" in multiprocessor systems. You can

Objects and Counters	Description
	measure the utilization on each processor as well, to ensure balanced performance between cores.
Disk	
- Avg. Disk Queue Length	This shows the average number of both read and write requests that were queued for the selected disk during the sample interval. A bigger disk queue length may not be a problem as long as disk reads/writes are not suffering and the system is working in a steady state without expanding queuing.
Avg. Disk Read Queue Length	The average number of read requests that are queued.
Avg. Disk Write Queue Length	The average number of write requests that are queued.
Disk Reads/sec	The number of reads to disk per second.
Disk Writes/sec	The number of writes to disk per second.
Memory	
- Available Mbytes	This shows the amount of physical memory available for allocation. Insufficient memory will lead to excessive use of the page file and an increase in the number of page faults per second.
- Cache Faults/sec	This counter shows the rate at which faults occur when a page is sought in the file system cache and is not found. This may be a soft fault, when the page is found in memory, or a hard fault, when the page is on disk. The effective use of the cache for read and write operations can have a significant effect on server performance. You must monitor for increased cache failures, indicated by a reduction in the **Async Fast Reads/sec** or **Read Aheads/sec**.
- Pages/sec	This counter shows the rate at which pages are read from or written to disk to resolve hard page faults. If this rises, it indicates system-wide performance problems.

Objects and Counters	Description
Paging File	
- % Used and % Used Peak	The server paging file, sometimes called the swap file, holds "virtual" memory addresses on disk. Page faults occur when a process has to stop and wait while required "virtual" resources are retrieved from disk into memory. These will be more frequent if the physical memory is inadequate.
NIC	
- Total Bytes/sec	This is the rate at which data is sent and received via the network interface card. You may need to investigate further if this rate is over 40-50 percent network capacity. To fine-tune your investigation, monitor **Bytes received/sec** and **Bytes Sent/sec**.
Process	
- Working Set	This counter indicates the current size (in bytes) of the working set for a given process. This memory is reserved for the process, even if it is not in use.
- % Processor Time	This counter indicates the percentage of processor time that is used by a given process.
Thread Count (_Total)	The current number of threads.
ASP.NET	
Requests Total	The total number of requests since the service was started.
Requests Queued	Microsoft SharePoint Foundation 2010 provides the building blocks for HTML pages that are rendered in the user browser over HTTP. This counter shows the number of requests waiting to be processed.
Request Wait Time	The number of milliseconds that the most recent request waited in the queue for processing. As the number of wait events increases, users will experience degraded page-rendering performance.
Requests Rejected	The total number of requests not executed

Objects and Counters	Description
	because of insufficient server resources to process them. This counter represents the number of requests that return a 503 HTTP status code, indicating that the server is too busy.
Requests Executing (_Total)	The number of requests currently executing.
Requests/Sec (_Total)	The number of requests executed per second. This represents the current throughput of the application. Under constant load, this number should remain within a certain range, barring other server work (such as garbage collection, cache cleanup thread, external server tools, and so on).
.NET CLR Memory	
# Gen 0 Collections	Displays the number of times the generation 0 objects (that is, the youngest, most recently allocated objects) are garbage collected since the application started. This number is useful as a ratio of #Gen 0: #Gen 1: #Gen 2 to make sure that the number of Gen 2 collections does not greatly exceed Gen 0 collections, optimally by a factor of 2.
# Gen 1 Collections	Displays the number of times the generation 1 objects are garbage collected since the application started.
# Gen 2 Collections	Displays the number of times the generation 2 objects are garbage collected since the application started. The counter is incremented at the end of a generation 2 garbage collection (also called a full garbage collection).
% Time in GC	Displays the percentage of elapsed time that was spent performing a garbage collection since the last garbage collection cycle. This counter usually indicates the work done by the garbage collector to collect and compact memory on behalf of the application. This counter is updated only at the end of every garbage collection. This counter is not an average; its value reflects the last observed value. This counter should be under 5% in normal

Objects and Counters	Description
	operation.

SQL Server Counters

The following table provides information on SQL Server objects and counters.

Objects and Counters	Description
General Statistics	This object provides counters to monitor general server-wide activity, such as the number of current connections and the number of users connecting and disconnecting per second from computers running an instance of SQL Server.
User Connections	This counter shows the amount of user connections on your instance of SQL Server. If you see this number rise by 500 percent from your baseline, you may see a performance reduction.
Databases	This object provides counters to monitor bulk copy operations, backup and restore throughput, and transaction log activities. Monitor transactions and the transaction log to determine how much user activity is occurring in the database and how full the transaction log is becoming. The amount of user activity can determine the performance of the database and affect log size, locking, and replication. Monitoring low-level log activity to gauge user activity and resource usage can help you to identify performance bottlenecks.
Transactions/sec	This counter shows the amount of transactions on a given database or on the entire SQL Server instance per second. This number is to help you create a baseline and to help you troubleshoot issues.
Locks	This object provides information about SQL Server locks on individual resource types.
Number of Deadlocks/sec	This counter shows the number of deadlocks on the SQL Server per second. This should normally

Objects and Counters	Description
	be 0.
Average Wait Time (ms)	This counter shows the average amount of wait time for each lock request that resulted in a wait.
Lock Wait Time (ms)	This counter shows the total wait time for locks in the last second.
Lock Waits/sec	This counter shows the number of locks per second that could not be satisfied immediately and had to wait for resources.
Latches	This object provides counters to monitor internal SQL Server resource locks called latches. Monitoring the latches to determine user activity and resource usage can help you to identify performance bottlenecks.
Average Latch Wait Time (ms)	This counter shows the average latch wait time for latch requests that had to wait.
Latch Waits/sec	This counter shows the number of latch requests per second that could not be granted immediately.
SQL Statistics	This object provides counters to monitor compilation and the type of requests sent to an instance of SQL Server. Monitoring the number of query compilations and recompilations and the number of batches received by an instance of SQL Server gives you an indication of how quickly SQL Server is processing user queries and how effectively the query optimizer is processing the queries.
SQL Compilations/sec	This counter indicates the number of times the compile code path is entered per second.
SQL Re-Compilations/sec	This counter indicates the number of times statement recompiles are triggered per second.
Plan Cache	This object provides counters to monitor how SQL Server uses memory to store objects such as stored procedures, ad hoc and prepared Transact-SQL statements, and triggers.
Cache Hit Ratio	This counter indicates the ratio between cache hits

Objects and Counters	Description
	and lookups for plans.
Buffer Cache	This object provides counters to monitor how SQL Server uses memory to store data pages, internal data structures, and the procedure cache, as well as counters to monitor the physical I/O as SQL Server reads and writes database pages.
Buffer Cache Hit Ratio	This counter shows the percentage of pages found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since an instance of SQL Server was started.

Removing Bottlenecks

System bottlenecks represent a point of contention where there are insufficient resources to service user transaction requests. These may be physical hardware, operating environment, or application-based. Often, the reason for the bottleneck will be inefficient custom code or 3rd party solutions, and a review of those could yield better results than adding hardware. Another common cause for bottlenecks is a misconfiguration of the farm, or an inefficient solution implementation that structures data in a way that requires more resources than necessary. For a system administrator, it is essential to manage bottlenecks by constantly monitoring performance. When you identify a performance issue, you must assess the best resolution for removing the bottleneck. The performance counters and other performance monitoring applications, such as System Center Operations Manager (SCOM), are the key tools in tracking and analyzing problems, so that you can develop a solution.

Physical Bottleneck Resolution

Physical bottlenecks are based on processor, disk, memory, and network contention: too many requests are contending for too few physical resources. The objects and counters described in the Monitoring Performance topic indicate where the performance problem is located, for example, hardware processor or ASP.NET. Bottleneck resolution requires that you identify the issue and then make a change or changes that mitigate the performance problem.

Problems seldom happen instantaneously; there is usually a gradual performance degradation that you can track if you monitor regularly, using your performance monitor tool or a more sophisticated system, such as SCOM. For both of these options, to varying degrees, you can embed solutions within an alert, in the form of advisory text or scripted commands.

You may have to resolve bottleneck issues by making changes to hardware or system configurations, once you have determined that they are not caused by a misconfiguration, inefficient custom code or

third party solutions, or inefficient solution implementation. The following tables identify problem threshold and possible resolution options. Some of the options suggest hardware upgrades or modifications.

Objects and Counters	Problem	Resolution Options
Processor		
Processor - % Processor Time	Over 75-85%	Upgrade processor Increase number of processors Add additional server(s)
Disk		
Avg. Disk Queue Length	Gradually increasing, system not in a steady state and queue is backing up	Increase number or speed of disks Change array configuration to stripe Move some data to an alternative server
% Idle Time	Greater than 90%	Increase number of disks Move data to an alternative disk or server
% Free Space	Less than 30%	Increase number of disks Move data to an alternative disk or server
Memory		
Available Mbytes	Less than 2GB on a Web server.	Add memory. **Note:** SQL server available memory will be low, by design, and does not always indicate a problem.
Cache Faults/sec	Greater than 1	Add memory Increase cache speed or size if possible Move data to an alternative disk or

Objects and Counters	Problem	Resolution Options
		server
Pages/sec	Greater than 10	Add memory
Paging File		
% Used and % Used Peak	The server paging file, sometimes called the swap file, holds "virtual" memory addresses on disk. Page faults occur when a process has to stop and wait while required "virtual" resources are retrieved from disk into memory. These will be more frequent if the physical memory is inadequate.	Add memory
NIC		
Total Bytes/sec	Over 40-50% of network capacity. This is the rate at which data is sent and received via the network interface card.	Investigate further by monitoring Bytes received/sec and Bytes Sent/sec. Reassess network interface card speed Check number, size, and usage of memory buffers
Process		
Working Set	Greater than 80% of total memory	Add memory
% Processor Time	Over 75-85%.	Increase number of processors Redistribute workload to additional servers
ASP.NET		
Application Pool Recycles	Several per day, causing intermittent slowness.	Make sure that you have not implemented settings that automatically recycle the application pool unnecessarily throughout the day.
Requests Queued	Hundreds or thousands of	Implement additional Web servers

Objects and Counters	Problem	Resolution Options
	requests queued.	The default maximum for this counter is 5,000, and you can change this setting in the Machine.config file
Request Wait Time	As the number of wait events increases, users will experience degraded page rendering performance.	Implement additional Web servers
Requests Rejected	Greater than 0	Implement additional Web servers

See Also

Capacity management and sizing overview for SharePoint Server 2010

Performance testing for SharePoint Server 2010

Capacity planning for SharePoint Server 2010

Health monitoring (SharePoint Server 2010)

Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

SharePoint Server 2010 capacity management: Software boundaries and limits

This document describes software boundaries and limits of Microsoft SharePoint Server 2010. These include the following:

- **Boundaries:** Static limits that cannot be exceeded by design
- **Thresholds:** Configurable limits that can be exceeded to accommodate specific requirements
- **Supported limits:** Configurable limits that have been set by default to a tested value.

Note:
The capacity planning information in this document provides guidelines for you to use in your planning. It is based on testing performed at Microsoft, on live properties. However, your results are likely to vary based on the equipment you use and the features and functionality that you implement for your sites.

In this article:

- Visio Services limits
- PerformancePoint Services limits
- Word Automation Services limits
- SharePoint Workspace limits
- OneNote limits
- Office Web Application Service limits
- Project Server limits

Overview of boundaries and limits

This article contains information to help you understand the tested performance and capacity limits of SharePoint Server 2010, and offers guidelines for how limits relate to acceptable performance. Use the information in this article to determine whether your planned deployment falls within acceptable performance and capacity limits, and to appropriately configure limits in your environment.

The test results and guidelines provided in this article apply to a single SharePoint Server 2010 farm. Adding servers to the installation might not increase the capacity limits of the objects that are listed in the tables in the Limits and boundaries section later in this topic. On the other hand, adding server computers increases the throughput of a server farm, which might be necessary to achieve acceptable performance with many objects. In some cases, the requirements for high numbers of objects in a solution might require more servers in the farm.

Note that there are many factors that can affect performance in a given environment, and each of these factors can affect performance in different areas. Some of the test results and recommendations in this article might be related to features or user operations that do not exist in your environment, and therefore do not apply to your solution. Only thorough testing can give you exact data related to your own environment.

Boundaries, thresholds and supported limits

In SharePoint Server 2010, there are certain limits that are by design and cannot be exceeded, and other limits that are set to default values that may be changed by the farm administrator. There are also certain limits that are not represented by a configurable value, such as the number of site collections per Web application.

- Boundaries are absolute limits that cannot be exceeded by design. It is important to understand these limits to ensure that you do not make incorrect assumptions when you design your farm.

  An example of a boundary is the 2 GB document size limit; you cannot configure SharePoint Server 2010 to store documents that are larger than 2 GB. This is a built-in absolute value, and cannot be exceeded by design.
- Thresholds are those that have a default value that cannot be exceeded unless the value is modified. Thresholds can, in certain circumstances, be exceeded to accommodate variances in your farm design, but it is important to understand that doing this may affect the performance of the farm in addition to the effective value of other limits.

The default value of certain thresholds can only be exceeded up to an absolute maximum value. A good example is the document size limit. By default, the default document size threshold is set to 50MB, but can be changed to support the maximum boundary of 2GB.

- Supported limits define the tested value for a given parameter. The default values for these limits were defined by testing, and represent the known limitations of the product. Exceeding supported limits may cause unexpected results, significant decrease in performance, or other harmful effects.

  Some supported limits are configurable parameters that are set by default to the recommended value, while other supported limits relate to parameters that are not represented by a configurable value.

An example of a supported limit is the number of site collections per Web application. The supported limit is 250,000, which is the largest number of site collections per Web application that met performance benchmarks during testing.

It is important to be aware that many of the limit values that are provided in this document represent a point in a curve that describes an increasing resource load and concomitant decrease in performance as the value increases. Therefore, exceeding certain limits, such as the number of site collections per Web application, may only result in a fractional decrease in farm performance. However, in most cases, operating at or near an established limit is not a best practice, as acceptable performance and reliability targets are best achieved when a farm's design provides for a reasonable balance of limits values.

Thresholds and supported limits guidelines are determined by performance. In other words, you can exceed the default values of the limits, but as you increase the limit value, farm performance and the effective value of other limits may be affected. Many limits in SharePoint Server 2010 can be changed, but it is important to understand how changing a given limit affects other parts of the farm.

How limits are established

In SharePoint Server 2010, thresholds and supported limits are established through testing and observation of farm behavior under increasing loads up to the point where farm services and operations reach their effective operational limits. Some farm services and components can support a higher load than others so that in some cases you must assign a limit value based on an average of several factors.

For example, observations of farm behavior under load when site collections are added indicate that certain features exhibit unacceptably high latency while other features are still operating within acceptable parameters. Therefore, the maximum value assigned to the number of site collections is not absolute, but is calculated based on an expected set of usage characteristics in which overall farm performance would be acceptable at the given limit under most circumstances.

Obviously, if some services are operating under parameters that are higher than those used for limits testing, the maximum effective limits of other services will be reduced. It is therefore important to execute rigorous capacity management and scale testing exercises for specific deployments in order to establish effective limits for that environment.

Note: We do not describe the hardware that was used to validate the limits in this document, because the limits were collected from multiple farms and environments. For descriptions of the farms we used

in testing, see Performance and capacity test results and recommendations (SharePoint Server 2010) and Performance and capacity technical case studies (SharePoint Server 2010).

The Equalizer Metaphor

You can consider thresholds and supported limits as sliders on a graphic equalizer, with each limit representing a certain frequency. In this metaphor, increasing the value of one limit may decrease the effective value of one or more other limits.

Imagine that one slider represents the maximum number of documents per library, a supported limit with a maximum tested value of around 30 million. However, this value depends on another slider, which represents the maximum size of documents in the farm, a threshold with a default value of 50 MB.

If you change the maximum size of documents to 1 GB to accommodate videos or other large objects, the number of documents your library can serve to users efficiently is reduced accordingly. For example, a given farm's hardware configuration and topology may support 1 million documents up to 50 MB. However, the same farm with the same number of documents cannot meet the same latency and throughput targets if the farm is serving a larger average document size because the file size limit was set to 1 GB.

The degree to which the maximum number of documents is reduced in this example is difficult to predict and is based on the number of large files in the library, the volume of data that they contain, the farm's usage characteristics, and the availability of hardware resources.

Limits and boundaries

This section lists the objects that can be a part of a solution and provides guidelines for acceptable performance for each kind of object. Acceptable performance means that the system as tested can support that number of objects, but that the number cannot be exceeded without some decrease in performance or a reduction in the value of related limits. Objects are listed both by scope and by feature. Limits data is provided, together with notes that describe the conditions under which the limit is obtained and links to additional information where available.

Use the guidelines in this article to review your overall solution plans. If your solution plans exceed the recommended guidelines for one or more objects, take one or more of the following actions:

- Evaluate the solution to ensure that compensations are made in other areas.
- Flag these areas for testing and monitoring as you build your deployment.
- Redesign or partition the solution to ensure that you do not exceed capacity guidelines.

Limits by hierarchy

This section provides limits sorted by the logical hierarchy of a SharePoint Server 2010 farm.

Web application limits

The following table lists the recommended guidelines for Web applications.

Limit	Maximum value	Limit type	Notes
Content database	300 per Web application	Supported	With 300 content databases per Web application, end user operations such as opening the site or site collections are not affected. But administrative operations such as creating a new site collection will experience decrease in performance. We recommend that you use Windows PowerShell to manage the Web application when a large number of content databases are present, because the management interface becomes slow and difficult to navigate.
Zone	5 per Web application	Boundary	The number of zones defined for a farm is hard-coded to 5. Zones include Default, Intranet, Extranet, Internet, and custom.
Managed path	20 per Web application	Supported	Managed paths are cached on the Web server, and CPU resources are used to process incoming requests against the managed path list. Exceeding 20 managed paths per Web application adds more load to the Web server for each request. If you plan to exceed twenty managed paths in a given Web application, we recommend that you test for acceptable system performance.
Solution cache size	300 MB per Web	Threshold	The solution cache allows the

Limit	Maximum value	Limit type	Notes
	application		InfoPath Forms service to hold solutions in cache in order to speed up retrieval of the solutions. If the cache size is exceeded, solutions are retrieved from disk, which may slow down response times. You can configure the size of the solution cache by using the Windows PowerShell cmdlet Set-SPInfoPathFormsService. For more information, see Set-SPInfoPathFormsService.

Web server and application server limits

The following table lists the recommended guidelines for Web servers on the farm.

Limit	Maximum value	Limit type	Notes
Application pools	10 per Web server	Supported	The maximum number is determined by hardware capabilities. This limit is dependent largely upon: • The amount of RAM allocated to the Web servers • The workload that the farm is serving, that is, the user base and the usage characteristics (a single highly active application pools can reach 10 GB or more)

Content database limits

The following table lists the recommended guidelines for content databases.

Limit	Maximum value	Limit type	Notes
Content database size	200 GB per content database	Supported	We strongly recommended limiting the size of content databases to 200 GB to help ensure system performance. Content database sizes up to 1 terabyte are supported only for large, single-site repositories and archives with non-collaborative I/O and usage patterns, such as Records Centers. Larger database sizes are supported for these scenarios because their I/O patterns and typical data structure formats have been designed for, and tested at, larger scales. For more information about large-scale document repositories, see "Estimate Performance and Capacity Requirements for Large Scale Document Repositories", available from Performance and capacity test results and recommendations (SharePoint Server 2010), and "Typical large-scale content management scenarios", available from Enterprise content storage planning (SharePoint Server 2010). A site collection should not exceed 100 GB unless it is

Limit	Maximum value	Limit type	Notes
			the only site collection in the database.
Site collections per content database	2,000 recommended 5,000 maximum	Supported	We strongly recommended limiting the number of site collections in a content database to 2,000. However, up to 5,000 site collections in a database are supported. These limits relate to speed of upgrade. The larger the number of site collections in a database, the slower the upgrade. The limit on the number of site collections in a database is subordinate to the limit on the size of a content database that has more than one site collection (200 GB). Therefore, as the number of site collections in a database increases, the average size of the site collections it contains must decrease. Exceeding the 2,000 site collection limit puts you at risk of longer downtimes during upgrades. If you plan to exceed 2,000 site collections, we recommend that you have a clear upgrade strategy, and obtain additional hardware to speed up upgrades and software updates that affect databases. To set the warning level for the number of sites in a content database, use the

Limit	Maximum value	Limit type	Notes
			Windows PowerShell cmdlet Set-SPContentDatabase with the -WarningSiteCount parameter. For more information, see Set-SPContentDatabase.
Remote BLOB Storage (RBS) storage subsystem on Network Attached Storage (NAS)	Time to first byte of any response from the NAS cannot exceed 20 milliseconds	Boundary	When SharePoint Server 2010 is configured to use RBS, and the BLOBs reside on NAS storage, consider the following boundary. From the time that SharePoint Server 2010 requests a BLOB, until it receives the first byte from the NAS, no more than 20 milliseconds can pass.

Site collection limits

The following table lists the recommended guidelines for site collections.

Limit	Maximum value	Limit type	Notes
Web site	250,000 per site collection	Supported	The maximum recommended number of sites and subsites is 250,000 sites. You can create a very large total number of Web sites by nesting subsites. For example, in a shallow hierarchy with 100 sites, each with 1,000 subsites, you would have a total of 100,000 Web sites. Or a deep hierarchy with 100

Limit	Maximum value	Limit type	Notes
			sites, each with 10 subsite levels would also contain a total of 100,000 Web sites. Note: Deleting or creating a site or subsite can significantly affect a site's availability. Access to the site and subsites will be limited while the site is being deleted. Attempting to create many subsites at the same time may also fail.
Site collection size	100 GB per site collection	Supported	A site collection should not exceed 100 GB unless it is the only site collection in the database. Certain site collection actions, such as site collection backup/restore or the Windows PowerShell cmdlet Move-SPSite, cause large Microsoft SQL Server operations which can affect performance or fail if other site collections are active in the same database. For more information, see Move-SPSite.

List and library limits

The following table lists the recommended guidelines for lists and libraries. For more information, see the "Designing Large Lists and Maximizing List Performance" white paper available from Performance and capacity test results and recommendations (SharePoint Server 2010).

Limit	Maximum value	Limit type	Notes
List row size	8,000 bytes per row	Boundary	Each list or library item can only occupy 8000 bytes in total in the database. 256 bytes are reserved for built-in columns, which leaves 7744 bytes for end-user columns. For details on how much space each kind of field consumes, see Column limits.
File size	2 GB	Boundary	The default maximum file size is 50 MB. This can be increased up to 2 GB, however a large volume of very large files can affect farm performance.
Documents	30,000,000 per library	Supported	You can create very large document libraries by nesting folders, or using standard views and site hierarchy. This value may vary depending on how documents and folders are organized, and by the type and size of documents stored.
Major versions	400,000	Supported	If you exceed this limit, basic file operations—such as file open or save, delete, and viewing the version histor— may not succeed.
Items	30,000,000 per list	Supported	You can create very large lists using standard views, site hierarchies, and metadata navigation. This value may vary depending on the number of columns in the list and the usage of the list.
Rows size limit	6 table rows internal to the database used for a list or library item	Supported	Specifies the maximum number of table rows internal to the database that can be used for a list or library item. To accommodate wide lists with many columns, each item may be wrapped over several internal table rows, up to six rows by default. This is configurable by farm administrators through the object model only. The object model method is SPWebApplication.MaxListItemRowStorage.
Bulk operations	100 items per	Boundary	The user interface allows a maximum of 100

Limit	Maximum value	Limit type	Notes
	bulk operation		items to be selected for bulk operations.
List view lookup threshold	8 join operations per query	Threshold	Specifies the maximum number of joins allowed per query, such as those based on lookup, person/group, or workflow status columns. If the query uses more than eight joins, the operation is blocked. This does not apply to single item operations. When using the maximal view via the object model (by not specifying any view fields), SharePoint will return up to the first eight lookups.
List view threshold	5,000	Threshold	Specifies the maximum number of list or library items that a database operation, such as a query, can process at the same time outside the daily time window set by the administrator during which queries are unrestricted.
List view threshold for auditors and administrators	20,000	Threshold	Specifies the maximum number of list or library items that a database operation, such as a query, can process at the same time when they are performed by an auditor or administrator with appropriate permissions. This setting works with Allow Object Model Override.
Subsite	2,000 per site view	Threshold	The interface for enumerating subsites of a given Web site does not perform well as the number of subsites surpasses 2,000. Similarly, the All Site Content page and the Tree View Control performance will decrease significantly as the number of subsites grows.
Coauthoring in Microsoft Word and Microsoft PowerPoint for .docx, .pptx and .ppsx files	10 concurrent editors per document	Threshold	Recommended maximum number of concurrent editors is 10. The boundary is 99. If there are 99 co-authors who have a single document opened for concurrent editing, any user after the 100th user sees a "File in use" error and have to view a read-only copy. More than 10 co-editors will lead to a gradually degraded user experience with

Limit	Maximum value	Limit type	Notes
			more conflicts and users will have to go through more iterations to get their changes to upload successfully.
Security scope	1,000 per list	Threshold	The maximum number of unique security scopes set for a list should not exceed 1,000. A scope is the security boundary for a securable object and any of its children that do not have a separate security boundary defined. A scope contains an Access Control List (ACL), but unlike NTFS ACLs, a scope can include security principals that are specific to SharePoint Server 2010. The members of an ACL for a scope can include Windows users, user accounts other than Windows users (such as forms-based accounts), Active Directory groups, or SharePoint groups.

Column limits

SharePoint Server 2010 data is stored in SQL Server tables. To allow for the maximum number of possible columns in a SharePoint list, SharePoint Server 2010 will create several rows in the database when data will not fit on a single row. This is called row wrapping.

Each time that a row is wrapped in SQL Server, an additional query load is put on the server when that item is queried because a SQL join must be included in the query. To prevent too much load, by default a maximum of six SQL Server rows are allowed for a SharePoint item. This limit leads to a particular limitation on the number of columns of each type that can be included in a SharePoint list. The following table describes the limits for each column type.

The row wrapping parameter can be increased beyond six, but this may result in too much load on the server. Performance testing is recommended before exceeding this limit. For more information, see the "Designing Large Lists and Maximizing List Performance" white paper that can be accessed from Performance and capacity test results and recommendations (SharePoint Server 2010).

Each column type has a size value listed in bytes. The sum of all columns in a SharePoint list cannot exceed 8,000 bytes. Depending on column usage, users can reach the 8,000 byte limitation before reaching the six-row row wrapping limitation.

Limit	Maximum value	Limit type	Size per column	Notes
Single line of text	276	Threshold	28 bytes	SQL Server row wrapping occurs after each 64 columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 384 Single line of text columns per SharePoint list (6 * 64 = 384). However, because the limit per SharePoint list item is 8000 bytes, of which 256 bytes are reserved for built-in SharePoint columns, the actual limit is 276 Single line of text columns.
Multiple Lines of Text	192	Threshold	28 bytes	SQL Server row wrapping occurs after each 32 columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 192 Multiple lines of text columns per SharePoint list (6 * 32 = 192).
Choice	276	Threshold	28 bytes	SQL Server row wrapping occurs

Limit	Maximum value	Limit type	Size per column	Notes
				after each 64 columns in a SharePoint list. The default row wrapping value of 6 allows for a maximum of 384 Choice columns per SharePoint list (6 * 64 = 384);); however because the limit per SharePoint list item is 8000 bytes, of which 256 bytes are reserved for built-in SharePoint columns, the actual limit should be 276 Choice columns.
Number	72	Threshold	12 bytes	SQL Server row wrapping occurs after each 12 columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 72 Number columns per SharePoint list (6 * 12 = 72).
Currency	72	Threshold	12 bytes	SQL Server row wrapping occurs after each 12 columns in a SharePoint list. The default row wrapping value of

Limit	Maximum value	Limit type	Size per column	Notes
				six allows for a maximum of 72 Currency columns per SharePoint list (6 * 12 = 72).
Date and Time	48	Threshold	12 bytes	SQL Server row wrapping occurs after each eight columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 48 Date and Time columns per SharePoint list (6 * 8 = 48).
Lookup	96	Threshold	4 bytes	SQL Server row wrapping occurs after each 16 columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 96 single value Lookup columns per SharePoint list (6 * 16 = 96).
Yes / No	96	Threshold	5 bytes	SQL Server row wrapping occurs after each 16 columns in a SharePoint list. The default row wrapping value of six allows for a

Limit	Maximum value	Limit type	Size per column	Notes
				maximum of 96 Yes / No columns per SharePoint list (6 * 16 = 96).
Person or group	96	Threshold	4 bytes	SQL Server row wrapping occurs after each 16 columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 96 Person or Group columns per SharePoint list (6 * 16 = 96).
Hyperlink or picture	138	Threshold	56 bytes	SQL Server row wrapping occurs after each 32 columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 192 Hyperlink or Picture columns per SharePoint list (6 * 32 = 192)); however because the limit per SharePoint list item is 8000 bytes, of which 256 bytes are reserved for built-in SharePoint columns, the actual limit should be 138

Limit	Maximum value	Limit type	Size per column	Notes
				Hyperlink or Picture columns.
Calculated	48	Threshold	28 bytes	SQL Server row wrapping occurs after each eight columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 48 Calculated columns per SharePoint list (6 * 8 = 48).
GUID	6	Threshold	20 bytes	SQL Server row wrapping occurs after each column in a SharePoint list. The default row wrapping value of six allows for a maximum of 6 GUID columns per SharePoint list (6 * 1 = 6).
Int	96	Threshold	4 bytes	SQL Server row wrapping occurs after each 16 columns in a SharePoint list. The default row wrapping value of six allows for a maximum of 96 Int columns per SharePoint list (6 * 16 = 96).
Managed	94	Threshold	40 bytes for the first, 32 bytes for	The first Managed Metadata field

Limit	Maximum value	Limit type	Size per column	Notes
metadata			each subsequent	added to a list is allocated four columns: • A lookup field for the actual tag • A hidden text field for the string value • A lookup field for the catch all • A lookup field for spillover of the catch all Each subsequent Managed Metadata field added to a list adds two more columns: • A lookup field for the actual tag • A hidden text field for the string value The maximum number of columns of Managed Metadata is calculated as (14 + (16 * (n-1))) where n is the row mapping value (default of 6).

External Data columns have the concept of a primary column and secondary columns. When you add an external data column, you can select some secondary fields of the external content type that you want to be added to the list. For example, given an External Content Type “Customer” which has fields like “ID”, “Name”, “Country”, and “Description”, when you add an External Data column of type “Customer” to a list, you can add secondary fields to show the “ID”, “Name” and “Description” of the Customer. Overall these are the columns that get added:

- Primary column: A text field.
- Hidden Id column: A multi-line text field.
- Secondary columns: Each secondary column is a text/number/Boolean/multi-line text that is based on the data type of the secondary column as defined in the Business Data Catalog model. For example, ID might be mapped to a *Number* column; Name might be mapped to a *Single line of text column*; Description might be mapped to a *Multiple lines of text* column.

Page limits

The following table lists the recommended guidelines for pages.

Limit	Maximum value	Limit type	Notes
Web parts	25 per wiki or Web part page	Threshold	This figure is an estimate based on simple Web Parts. The complexity of the Web parts dictates how many Web Parts can be used on a page before performance is affected.

Security limits

Limit	Maximum value	Limit type	Notes
Number of SharePoint groups a user can belong to	5,000	Supported	This is not a hard limit but it is consistent with Active Directory guidelines. There are several things that affect this number: • The size of the user token

Limit	Maximum value	Limit type	Notes
			• The groups cache: SharePoint Server 2010 has a table that caches the number of groups a user belongs to as soon as those groups are used in access control lists (ACLs). • The security check time: as the number of groups that a user is a member of increases, the time that is required for the access check increases also.
Users in a site collection	2 million per site collection	Supported	You can add millions of people to your Web site by using Microsoft Windows security groups to manage security instead of using individual users. This limit is based on manageability and ease of navigation in the user interface. When you have many entries (security groups of users) in the site collection (more than one thousand), you should use Windows PowerShell to manage users instead of the UI. This will provide a better management

Limit	Maximum value	Limit type	Notes
			experience.
Active Directory Principles/Users in a SharePoint group	5,000 per SharePoint group	Supported	SharePoint Server 2010 enables you to add users or Active Directory groups to a SharePoint group. Having up to 5,000 users (or Active Directory groups or users) in a SharePoint group provides acceptable performance. The activities most affected by this limit are as follows: • Fetching users to validate permissions. This operation takes incrementally longer with growth in number of users in a group. • Rendering the membership of the view. This operation will always require time.
SharePoint groups	10,000 per site collection	Supported	Above 10,000 groups, the time to execute operations is increased significantly. This is especially true of adding a user to an existing group, creating a new group, and rendering group views.
Security principal: size of	5,000 per Access	Supported	The size of the scope

Limit	Maximum value	Limit type	Notes
the Security Scope	Control List (ACL)		affects the data that is used for a security check calculation. This calculation occurs every time that the scope changes. There is no hard limit, but the bigger the scope, the longer the calculation takes.

Limits by feature

This section lists limits sorted by feature.

Search limits

The following table lists the recommended guidelines for Search.

Limit	Maximum value	Limit type	Notes
SharePoint search service applications	20 per farm	Supported	Multiple SharePoint search service applications can be deployed on the same farm, because you can assign search components and databases to separate servers. The recommended limit of 20 is less than the maximum limit for all service applications in a farm.
Crawl databases and database Items	10 crawl databases per search service application 25 million items per crawl database	Threshold	The crawl database stores the crawl data (time/status, etc) about all items that have been crawled. The supported limit is 10 crawl databases per SharePoint Search service application. The recommended limit is 25 million items per crawl database (or a total of four crawl

Limit	Maximum value	Limit type	Notes
			databases per search service application).
Crawl components	16 per search service application	Threshold	The recommended limit per application is 16 total crawl components; with two per crawl database, and two per server, assuming the server has at least eight processors (cores). The total number of crawl components per server must be less than 128/(total query components) to minimize propagation I/O degradation. Exceeding the recommended limit may not increase crawl performance; in fact, crawl performance may decrease based on available resources on the crawl server, database, and content host.
Index partitions	20 per search service application; 128 total	Threshold	The index partition holds a subset of the search service application index. The recommended limit is 20. Increasing the number of index partitions results in each partition holding a smaller subset of the index, reducing the RAM and disk space that is needed on the query server hosting the query component assigned to the index partition. The boundary for the total number of index partitions is 128.
Indexed items	100 million per search service application; 10 million per index partition	Supported	SharePoint Search supports index partitions, each of which contains a subset of the search index. The recommended

Limit	Maximum value	Limit type	Notes
			maximum is 10 million items in any partition. The overall recommended maximum number of items (e.g., people, list items, documents, Web pages) is 100 million.
Crawl log entries	100 million per search application	Supported	This is the number of individual log entries in the crawl log. It will follow the "Indexed items" limit.
Property databases	10 per search service application;128 total	Threshold	The property database stores the metadata for items in each index partition associated with it. An index partition can only be associated with one property store. The recommended limit is 10 property databases per search service application. The boundary for index partitions is 128.
Query components	128 per search application; 64/(total crawl components) per server	Threshold	The total number of query components is limited by the ability of the crawl components to copy files. The maximum number of query components per server is limited by the ability of the query components to absorb files propagated from crawl components.
Scope rules	100 scope rules per scope; 600 total per search service application	Threshold	Exceeding this limit will reduce crawl freshness, and delay potential results from scoped queries.
Scopes	200 per site	Threshold	This is a recommended limit per site. Exceeding this limit may reduce crawl efficiency and, if the scopes are added to the display group, affect end-user

Limit	Maximum value	Limit type	Notes
			browser latency. Also, display of the scopes in the search administration interface degrades as the number of scopes passes the recommended limit.
Display groups	25 per site	Threshold	Display groups are used for a grouped display of scopes through the user interface. Exceeding this limit starts degrading the scope experience in the search administration interface.
Alerts	1,000,000 per search application	Supported	This is the tested limit.
Content sources	50 per search service application	Threshold	The recommended limit of 50 can be exceeded up to the boundary of 500 per search service application. However, fewer start addresses should be used, and the concurrent crawl limit must be followed.
Start addresses	100 per content source	Threshold	The recommended limit can be exceeded up to the boundary of 500 per content source. However, the more start addresses you have, the fewer content sources should be used. When you have many start address, we recommend that you put them as links on an html page, and have the HTTP crawler crawl the page, following the links.
Concurrent crawls	20 per search application	Threshold	This is the number of crawls underway at the same time. Exceeding this number may cause the overall crawl rate to

Limit	Maximum value	Limit type	Notes
			decrease.
Crawled properties	500,000 per search application	Supported	These are properties that are discovered during a crawl.
Crawl impact rule	100	Threshold	Recommended limit of 100 per farm. The recommendation can be exceeded; however, display of the site hit rules in the search administration interface is degraded. At approximately 2000 site hit rules, the Manage Site Hit Rules page becomes unreadable.
Crawl rules	100 per search service application	Threshold	This value can be exceeded; however, display of the crawl rules in the search administration interface is degraded.
Managed properties	100,000 per search service application	Threshold	These are properties used by the search system in queries. Crawled properties are mapped to managed properties.
Mappings	100 per managed property	Threshold	Exceeding this limit may decrease crawl speed and query performance.
URL removals	100 removals per operation	Supported	This is the maximum recommended number of URLs that should be removed from the system in one operation.
Authoritative pages	1 top level and minimal second and third level pages per search service application	Threshold	The recommended limit is one top-level authoritative page, and as few second -and third-level pages as possible to achieve the desired relevance. The boundary is 200 per relevance level per search application, but adding additional pages may not

Limit	Maximum value	Limit type	Notes
			achieve the desired relevance. Add the key site to the first relevance level. Add more key sites at either second or third relevance levels, one at a time, and evaluate relevance after each addition to ensure that the desired relevance effect is achieved.
Keywords	200 per site collection	Supported	The recommended limit can be exceeded up to the maximum (ASP.NET-imposed) limit of 5000 per site collection given five Best Bets per keyword. If you exceed this limit, display of keywords on the site administration user interface will degrade. The ASP.NET-imposed limit can be modified by editing the Web.Config and Client.config files (MaxItemsInObjectGraph).
Metadata properties recognized	10,000 per item crawled	Boundary	This is the number of metadata properties that can be determined and potentially mapped or used for queries when an item is crawled.

User Profile Service limits

The following table lists the recommended guidelines for User Profile Service.

Limit	Maximum value	Limit type	Notes
User profiles	2,000,000 per service application	Supported	A user profile service application can support up to 2 million user profiles with full social

Limit	Maximum value	Limit type	Notes
			features functionality. This number represents the number of profiles that can be imported into the people profile store from a directory service, and also the number of profiles a user profile service application can support without leading to performance decreases in social features.
Social tags, notes and ratings	500,000,000 per social database	Supported	Up to 500 million total social tags, notes and ratings are supported in a social database without significant decreases in performance. However, database maintenance operations such as backup and restore may show decreased performance at that point.

Content deployment limits

The following table lists the recommended guidelines for content deployment.

Limit	Maximum value	Limit type	Notes
Content deployment jobs running on different paths	20	Supported	For concurrently running jobs on paths that are connected to site collections in the same source content database, there is an increased risk of deadlocks on the database. For jobs that must run concurrently, we recommend that you

Limit	Maximum value	Limit type	Notes
			move the site collections into different source content databases. **Note:** Concurrent running jobs on the same path are not possible. If you are using SQL Server snapshots for content deployment, each path creates a snapshot. This increases the I/O requirements for the source database. For more information, see About deployment paths and jobs.

Blog limits

The following table lists the recommended guidelines for blogs.

Limit	Maximum value	Limit type	Notes
Blog posts	5000 per site	Supported	The maximum number of blog posts is 5000 per site.
Comments	1000 per post	Supported	The maximum number of comments is 1000 per post.

Business Connectivity Services limits

The following table lists the recommended guidelines for Business Connectivity Services.

Limit	Maximum value	Limit type	Notes
ECT (in-memory)	5000 per Web Server (per tenant)	Boundary	Total number of external content type (ECT) definitions loaded in memory at a given point in time on a Web server.
External system connections	500 per Web server	Boundary	Number of active/open external system connections at a given point in time. The default maximum value is 200; the boundary is 500. This limit is enforced at the Web Server scope, regardless of the kind of external system (for example, database, .NET assembly, and so on) The default maximum is used to restrict the number of connections. An application can specify a larger limit via execution context; the boundary enforces the maximum even for applications that do not respect the default.
Database items returned per request	2,000 per database connector	Threshold	Number of items per request the database connector can return. The default maximum of 2,000 is used by the database connector to restrict the number of result that can be returned per page. The application can specify a larger limit via execution

Limit	Maximum value	Limit type	Notes
			context; the Absolute Max enforces the maximum even for applications that do not respect the default. The boundary for this limit is 1,000,000.

Workflow limits

The following table lists the recommended guidelines for workflow.

Limit	Maximum value	Limit type	Notes
Workflow postpone threshold	15	Threshold	15 is the maximum number of workflows allowed to be executing against a content database at the same time, excluding instances that are running in the timer service. When this threshold is reached, new requests to activate workflows will be queued to be run by the workflow timer service later. As non-timer execution is completed, new requests will count against this threshold. This is limit can be configured by using the Set-SPFarmConfig Windows PowerShell cmdlet. For more information, see Set-SPFarmConfig. Note: This limit does not refer to the total number of workflow instances that

Limit	Maximum value	Limit type	Notes
			can be in progress. Instead, it is the number of instances that are being processed. Increasing this limit increases the throughput of starting and completing workflow tasks but also increases load against the content database and system resources.
Workflow timer batch size	100	Threshold	The number of events that each run of the workflow timer job will pick up and deliver to workflows. It is configurable by using Windows PowerShell. To allow for additional events, you can run additional instances of the Microsoft SharePoint Foundation Workflow Timer Service.

Managed Metadata term store (database) limits

The following table lists the recommended guidelines for managed metadata term stores.

Limit	Maximum value	Limit type	Notes
Maximum number of levels of nested terms in a term store	7	Supported	Terms in a term set can be represented hierarchically. A term set can have up to seven levels of terms (a parent term, and six levels of nesting below it.)
Maximum number of term sets in a term store	1000	Supported	You can have up to 1000 term sets in a term store.

Limit	Maximum value	Limit type	Notes
Maximum number of terms in a term set	30,000	Supported	30,000 is the maximum number of terms in a term set. **Note:** Additional labels for the same term, such as synonyms and translations, do not count as separate terms.
Total number of items in a term store	1,000,000	Supported	An item is either a term or a term set. The sum of the number of terms and term sets cannot exceed 1,000,000. Additional labels for the same term, such as synonyms and translations, do not count as separate terms. **Note:** You cannot have both the maximum number of term sets and the maximum number of terms simultaneously in a term store.

Visio Services limits

The following table lists the recommended guidelines for instances of Visio Services in Microsoft SharePoint Server 2010.

Limit	Maximum value	Limit type	Notes
File size of Visio Web drawings	50 MB	Threshold	Visio Services has a configuration setting that enables the administrator to change the maximum size of Web drawings

Limit	Maximum value	Limit type	Notes
			that Visio processes. Larger file sizes have the following side effects: • Increase in the memory footprint of Visio Services. • Increase in CPU usage. • Reduction in application server requests per second. • Increase overall latency. • Increase SharePoint farm network load
Visio Web drawing recalculation time-out	120 seconds	Threshold	Visio Services has a configuration setting that enables the administrator to change the maximum time that it can spend recalculating a drawing after a data refresh. A larger recalculation time-out leads to: • Reduction in CPU and memory availability. • Reduction in application requests per second. • Increase in average latency across all documents. A smaller recalculation time-out leads to: • Reduction of the

Limit	Maximum value	Limit type	Notes
			complexity of diagrams that can be displayed. • Increase in requests per second. • Decrease in average latency across all documents.
Visio Services minimum cache age (data connected diagrams)	Minimum cache age: 0 to 24hrs	Threshold	Minimum cache age applies to data connected diagrams. It determines the earliest point at which the current diagram can be removed from cache. Setting Min Cache Age to a very low value will reduce throughput and increase latency, because invalidating the cache too often forces Visio to recalculate often and reduces CPU and memory availability.
Visio Services maximum cache age (non-data connected diagrams)	Maximum cache age: 0 to 24hrs	Threshold	Maximum cache age applies to non-data connected diagrams. This value determines how long to keep the current diagram in memory. Increasing Max Cache Age decreases latency for commonly requested drawings. However, setting Max

Limit	Maximum value	Limit type	Notes
			Cache Age to a very high value increases latency and slows throughput for items that are not cached, because the items already in cache consume and reduce available memory.

PerformancePoint Services limits

The following table lists the recommended guidelines for PerformancePoint Services in Microsoft SharePoint Server 2010.

Limit	Maximum value	Limit type	Notes
Cells	1,000,000 per query on Excel Services data source	Boundary	A PerformancePoint scorecard that calls an Excel Services data source is subject to a limit of no more than 1,000,000 cells per query.
Columns and rows	15 columns by 60,000 rows	Threshold	The maximum number of columns and rows when rendering any PerformancePoint dashboard object that uses a Microsoft Excel workbook as a data source. The number of rows could change based on the number of columns.
Query on a SharePoint list	15 columns by 5000 rows	Supported	The maximum number of columns and row when rendering any PerformancePoint dashboard object that uses

Limit	Maximum value	Limit type	Notes
			a SharePoint list as a data source. The number of rows could change based on the number of columns.
Query on a SQL Server data source	15 columns by 20000 rows	Supported	The maximum number of columns and row when rendering any PerformancePoint dashboard object that uses a SQL Server table data source. The number of rows could change based on the number of columns.

Word Automation Services limits

The following table lists the recommended guidelines for Word Automation Services.

Limit	Maximum value	Limit type	Notes
Input file Size	512 MB	Boundary	Maximum file size that can be processed by Word Automation Services.
Frequency with which to start conversions (minutes)	1 minute (recommended) 15 minutes (default) 59 minutes (boundary)	Threshold	This setting determines how often the Word Automation Services timer job executes. A lower number leads to the timer job running faster. Our testing shows that it is most useful to run this timer job once per minute.
Number of conversions to start per conversion process	For PDF/XPS output formats: 30 x MFor all other output formats: 72 x M Where M is the value of Frequency with which to start conversions (minutes)	Threshold	The number of conversions to start affects the throughput of Word Automation Services. If these values are set higher than the recommended levels then some conversion items may start to fail intermittently and user permissions may expire. User permissions expire 24 hours from the time that a conversion

Limit	Maximum value	Limit type	Notes
			job is started.
Conversion job size	100,000 conversion items	Supported	A conversion job includes one or more conversion items, each of which represents a single conversion to be performed on a single input file in SharePoint. When a conversion job is started (using the ConversionJob.Start method), the conversion job and all conversion items are transmitted over to an application server which then stores the job in the Word Automation Services database. A large number of conversion items will increase both the execution time of the Start method and the number of bytes transmitted to the application server.
Total active conversion processes	N-1, where N is the number of cores on each application server	Threshold	An active conversion process can consume a single processing core. Therefore, customers should not run more conversion processes than they have processing cores in their application servers. The conversion timer job and other SharePoint activities also require occasional use of a processing core. We recommend that you always leave 1 core free for use by the conversion timer job and SharePoint.
Word Automation Services database size	2 million conversion items	Supported	Word Automation Services maintains a persistent queue of conversion items in its database. Each conversion request generates one or more records. Word Automation Services does not delete records from the database automatically, so the database can grow indefinitely without maintenance. Administrators can manually remove conversion job history by using the Windows PowerShell cmdlet Remove-

Limit	Maximum value	Limit type	Notes
			SPWordConversionServiceJobHistory. For more information, see Remove-SPWordConversionServiceJobHistory.

SharePoint Workspace limits

The following table lists the recommended guidelines for Microsoft SharePoint Workspace 2010.

Limit	Maximum value	Limit type	Notes
SharePoint Workspace synchronization	30,000 items per list	Boundary	SharePoint Workspace will not synchronize lists that have more than 30,000 items. This restriction exists because the time to download a list that has more than 30,000 items is very long, and resource usage is high.
SharePoint Workspace synchronization	1800 documents limit in SharePoint Workspace	Boundary	Users are warned when they have more than 500 documents in SharePoint Workspace, but they can continue to add documents.

OneNote limits

The following table lists the recommended guidelines for Microsoft OneNote Services.

Limit	Maximum value	Limit type	Notes
Number of Sections and Section Groups in a OneNote Notebook (on	See limit for "Documents" in List and library limits		Each section counts as one folder and one document in the list. Each section group counts as one folder and one document in the

Limit	Maximum value	Limit type	Notes
SharePoint)			list.
Maximum size of a section	See limit for "File size" in List and library limits		This maximum excludes any images, embedded files, and XPS printouts to OneNote that are larger than 100 KB. Images and embedded files larger than 100 KB are split out into their own binary files. This means that a section with 100 KB of typed data and four embedded Word documents of 1 MB each will be considered a 100 KB section.
Maximum size of an image, embedded file, and XPS OneNote printout in a OneNote section.	See limit for "File size" in List and library limits		Each item is stored as a separate binary file and is therefore subject to file size limits. Each print operation to OneNote will result in one XPS printout binary, even if the printout contains multiple pages.
Maximum size of all images, embedded files, and XPS printouts in a single OneNote page.	Default limit is double the "File size" limit.	Threshold	This applies to embedded content in a single OneNote page, not a Section or Notebook. If users encounter this, they will see the following error in OneNote: jerrcStorageUrl_HotTableFull (0xE0000794). Users can work around this by splitting embedded content into different pages and deleting previous versions of the page. If users have to adjust this value (“Max Hot Table Size”), the effective limit is half of the absolute value they define (for example, specifying a 400 MB max hot table size means that the maximum size of all embedded content on a page is limited to 200 MB).

Limit	Maximum value	Limit type	Notes
Merge operations	One per CPU core per Web server	Boundary	OneNote merges combine changes from multiple users who are co-authoring a notebook. If no CPU core is available to run a merge, a conflict page is generated instead, which forces the user perform the merge manually). This limit applies whether OneNote is running as a client application or as a Microsoft Office Web Apps.

Office Web Application Service limits

The following table lists the recommended guidelines for Office Web Apps. Office client application limits also apply when an application is running as a Web app.

Limit	Maximum value	Limit type	Notes
Cache size	100 GB	Threshold	Space available to render documents, created as part of a content database. By default, the cache available to render documents is 100 GB. We do not recommend that you increase the available cache.
Renders	One per document per second per CPU core per application server (maximum eight cores)	Boundary	This is the measured average number of renders that can be performed of "typical" documents on the application server over a period of time.

Project Server limits

The following table lists the recommended guidelines for Microsoft Project Server. For more information about how to plan for Project Server, see Planning and architecture for Project Server 2010.

Limit	Maximum value	Limit type	Notes
End of project time	Date: 12/31/2049	Boundary	Project plans cannot extend past the date 12/31/2049.
Deliverables per project plan	1500 deliverables	Boundary	Project plans cannot contain more than 1500 deliverables.
Number of fields in a view	256	Boundary	A user cannot have more than 256 fields added to a view that they have defined in Project Web App.
Number of clauses in a filter for a view	50	Boundary	A user cannot add a filter to a view that has more than 50 clauses in it.

Performance and capacity technical case studies (SharePoint Server 2010)

This section contains technical case studies that describe specific deployments of Microsoft SharePoint Server 2010. Compare the scenarios in these documents to your planned workload and usage characteristics. If your planned design is similar, you can use the documented deployment as a starting point for your own installation.

These articles include information about environments, such as:

- Environment specifications, such as hardware, farm topology, and configuration
- The workload used for data generation, including the number and class of users, and farm usage characteristics
- Farm dataset, including database contents, Search indexes, and external data sources
- Health and performance data specific to the environment
- Performance data and recommendations for how to determine the hardware, topology, and configuration you need to deploy a similar environment, and how to optimize your environment for appropriate capacity and performance characteristics

Before reading these articles, it is important that you understand the key concepts behind capacity management in SharePoint Server 2010. For more information, see Capacity management and sizing for SharePoint Server 2010.

In this section:

- **Publishing**
  - Microsoft SharePoint Server 2010 enterprise intranet publishing environment: Technical case study

    Describes the published intranet environment used by employees at Microsoft.
- **Enterprise Intranet Collaboration**
  - Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Technical case study

    Describes an environment that hosts mission-critical team sites and publishing portals for internal organizations, teams, and projects.
  - Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Lab study

    Describes lab testing performed on a similar environment to the enterprise Intranet collaboration environment.
- **Departmental Collaboration**
  - Microsoft SharePoint Server 2010 departmental collaboration environment: Technical case study:

Describes a departmental collaboration environment used by employees of one department inside Microsoft.

- Microsoft SharePoint Server 2010 divisional portal environment: Lab study

  Describes lab testing performed on a similar environment to the departmental collaboration environment.

- **Social**
  - Microsoft SharePoint Server 2010 social environment: Technical case study

    Describes an environment that hosts My Sites that present employee information to the wider organization. The environment serves as a central location for personal sites and shared documents.

Microsoft SharePoint Server 2010 enterprise intranet publishing environment: Technical case study

This document describes a specific deployment of Microsoft SharePoint Server 2010. It includes the following:

- Technical case study environment specifications, such as hardware, farm topology and configuration
- The workload that includes the number, and types, of users or clients, and environment usage characteristics
- Technical case study farm dataset that includes database contents and Search indexes
- Health and performance data that is specific to the environment

In this article:

Prerequisite information

Before reading this document, make sure that you understand the key concepts behind SharePoint Server 2010 capacity management. The following documentation will help you learn about the recommended approach to capacity management and provide context for helping you understand how to make effective use of the information in this document, and also define the terms used throughout this document.

For more conceptual information about performance and capacity that you might find valuable in understanding the context of the data in this technical case study, see the following documents:

- Capacity management and sizing overview for SharePoint Server 2010
- SharePoint Server 2010 capacity management: Software boundaries and limits

Introduction to this environment

This white paper describes an actual SharePoint Server 2010 environment at Microsoft. Use this document to compare with your planned workload and usage characteristics. If your planned design is similar, you can use the deployment described here as a starting point for your own installation.

This document includes the following:

- **Specifications**, which include hardware, topology and configuration
- **Workload**, which is the demand on the farm that includes the number of users, and the usage characteristics
- **Dataset** that includes database sizes
- **Health and performance** data that is specific to the environment

This document is part of a series of Performance and capacity technical case studies (SharePoint Server 2010) about SharePoint environments at Microsoft.

The SharePoint Server 2010 environment described in this document is a production environment at a large, geographically distributed company. Employees view various content, such as news, technical articles, employee profiles, documentation, and training resources. Employees also use this environment to perform search queries against all the SharePoint environments within the company. Employees receive daily e-mails with links to articles on the environment and many employees set this environment as their browser home page.

As many as 48,000 unique users visit the environment on a busy day, generating up to 345 requests per second (RPS) during peak hours. Because this is an intranet site, all users are authenticated. Although content is published using a single environment author-in-place model, the environment's publishing procedure specifies that all draft content is published at a single time during the night in off-peak hours.

The information that is provided in this document reflects the enterprise intranet publishing environment on a typical day.

Specifications

This section provides detailed information about the hardware, software, topology, and configuration of the case-study environment.

Hardware

This section provides details about the server computers that were used in this environment.

Note

- This environment is scaled to accommodate pre-release builds of SharePoint Server 2010 and other products. Hence, the hardware deployed has bigger capacity than necessary to serve the demand typically experienced by this environment. This hardware is described only to provide additional context for this environment and serve as a starting point for similar environments.
- It is important to conduct your own capacity management based on your planned workload and usage characteristics. For more information about the capacity management process, see Capacity management and sizing overview for SharePoint Server 2010.

Web Servers

There are four Web servers in the farm, each with identical hardware. Three serve content, and the fourth is a dedicated search crawl target.

Web Server	WFE1-4
Processor(s)	2 quad core @ 2.33 GHz
RAM	32 GB
Operating system	Windows Server 2008, 64 bit
Size of the SharePoint drive	300 GB
Number of network adapters	2
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Load balancer type	Hardware load balancing
Software version	SharePoint Server 2010 (pre-release version)
Services running locally	Central Administration Microsoft SharePoint Foundation Incoming E-Mail Microsoft SharePoint Foundation Web Application Microsoft SharePoint Foundation Workflow Timer

Web Server	WFE1-4
	Service Search Query and Site Settings Service SharePoint Server Search
Services consumed from a federated services farm	User Profile Service Web Analytics Web Service Business Data Connectivity Service Managed Metadata Web Service

Application Server

There are no application servers in the farm. Local services are hosted on the Web servers. Other services are consumed from a federated services farm.

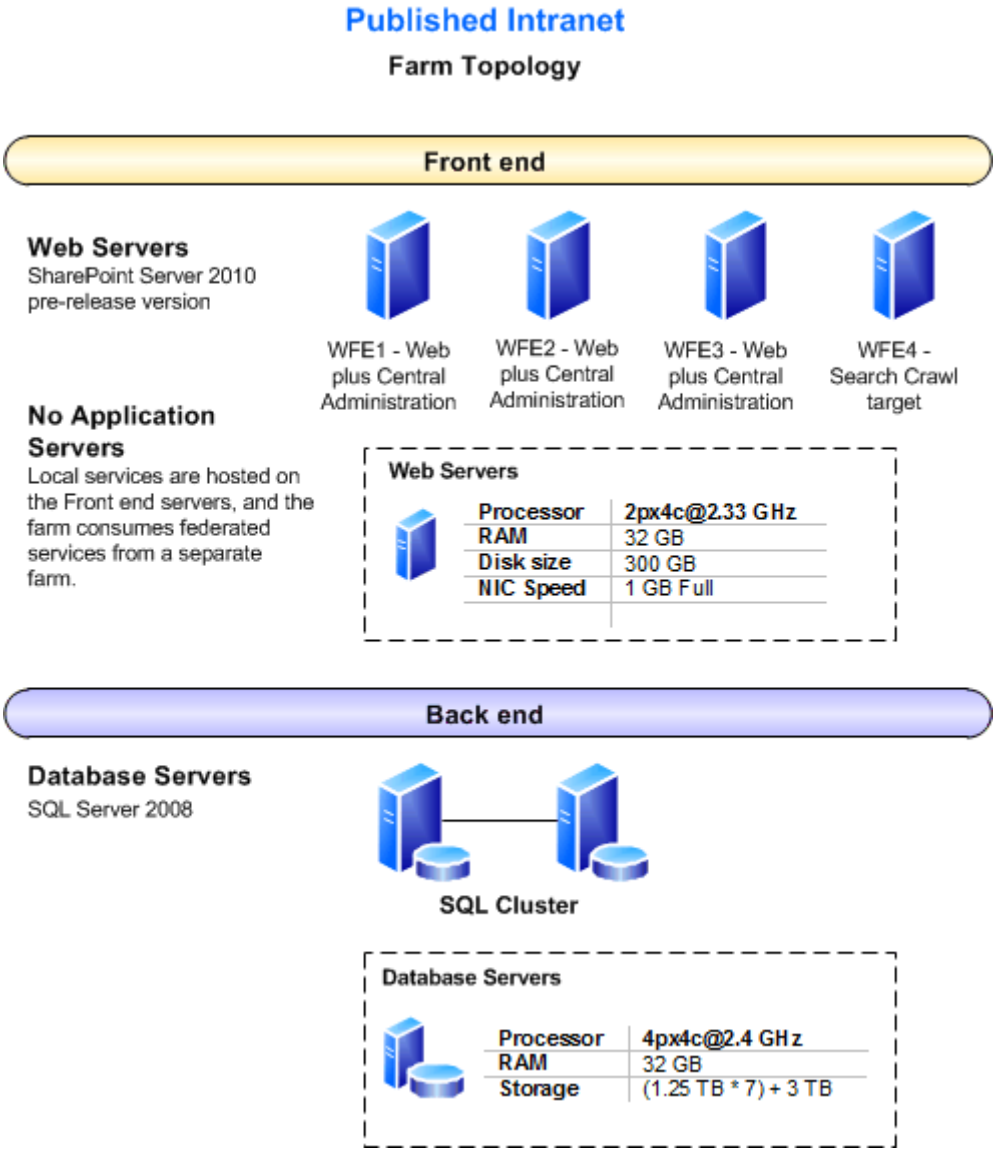
Database Servers

There is a SQL cluster with two database servers, each with identical hardware. One of the servers is active and the other is passive for redundancy.

Database Server	DB1-2
Processor(s)	4 quad core @ 2.4 GHz
RAM	32 GB
Operating system	Windows Server 2008, 64 bit
Storage and geometry	(1.25 TB * 6) + 3 TB Disk 1-4: SQL Data Disk 5: Logs Disk 6: TempDB
Number of network adapters	2
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Software version	SQL Server 2008

Topology

The following diagram shows the topology for this farm.



Configuration

The following table enumerates settings that were changed that affect performance or capacity in the environment.

Setting	Value	Notes
Site Collection Administration: Site collection output cache	On	Enabling the output cache improves server efficiency by reducing calls to the database for data that is frequently requested.
Site collection cache profile (select)	Intranet (Collaboration Site)	“Allow writers to view cached content” is checked, bypassing the ordinary behavior of not letting people with edit permissions to have their pages cached.
Object Cache (Off | n MB)	On – 500 MB	The default is 100 MB. Increasing this setting enables additional data to be stored in the front-end Web server memory.
Usage Service: Trace Log – days to store log files (default: 14 days)	5 days	The default is 14 days. Lowering this setting can save disk space on the server where the log files are stored.
Query Logging Threshold: Microsoft SharePoint Foundation Database – configure QueryLoggingThreshold to 1 second	1 second	The default is 5 seconds. Lowering this setting can save bandwidth and CPU on the database server.
Database Server – Default Instance: Max degree of parallelism	1	The default is 0. To ensure optimal performance, we strongly recommend that you set max degree of parallelism to 1 for database servers that host SharePoint Server 2010 databases. For more information about how to set max degree of parallelism, see max degree of parallelism Option(http://go.microsoft.com/fwlink/?LinkId=189030).

Workload

This section describes the workload, which is the demand on the farm that includes the number of users, and the usage characteristics.

Workload Characteristics	Value
Average Requests per Second (RPS)	100
Average RPS at peak time (11 AM-3 PM)	226
Total number of unique users per day	33,580
Average concurrent users	172
Maximum concurrent users	376
Total # of requests per day	3,800,000

This table shows the number of requests for each user agent.

User Agent	Requests	Percentage of Total
Browser	3,261,563	97.09%
DAV	2,418	0.07%
Search (crawl)	92,322	2.75%
OneNote	1,628	0.05%
Outlook	961	0.03%
Word	449	0.01%

Dataset

This section describes the case study farm dataset that includes database sizes and Search indexes.

Dataset Characteristics	Value
Database size (combined)	49.9 GB
BLOB size	22.2 GB
Number of content databases	3
Number of Web applications	3
Number of site collections	4
Number of sites	797

Dataset Characteristics	Value
Search index size (number of items)	275,000

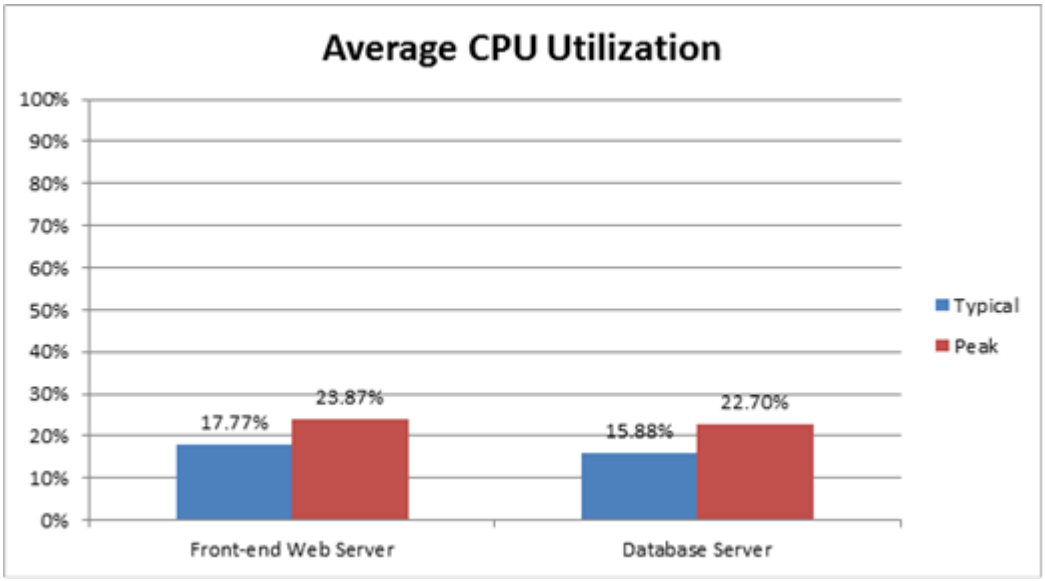
Health and Performance Data

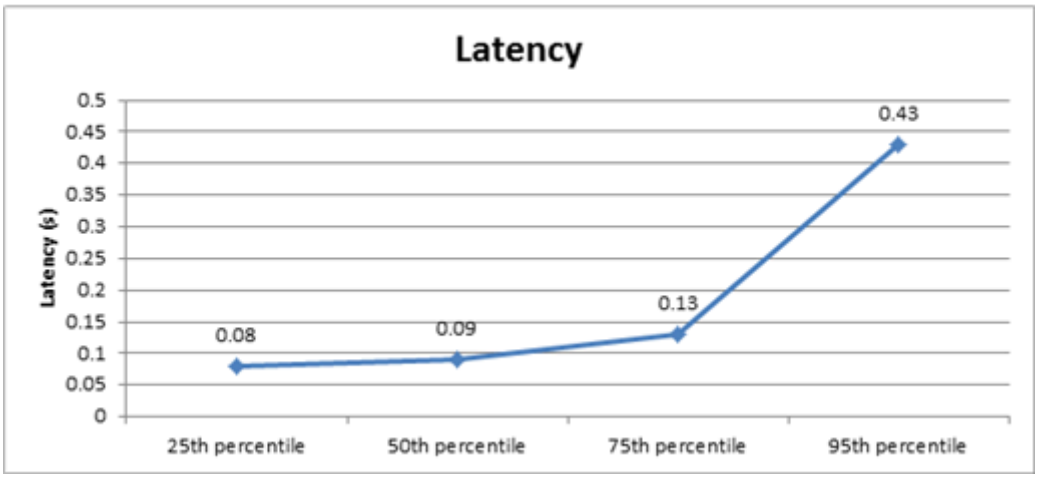
This section provides health and performance data that is specific to the case study environment.

General Counters

Metric	Value
Availability (uptime)	99.95%
Failure Rate	0.05%
Average memory used	1.08 GB
Maximum memory used	2.60 GB
Search Crawl % of Traffic (Search client requests / total requests)	6%
ASP.NET Requests Queued	0.00

The following charts show the average CPU utilization and latency for this environment.





In this document, latency is divided into four categories. The 50th percentile latency is typically used to measure the server's responsiveness. It means that half of the requests are served within that response time. The 95th percentile latency is typically used to measure spikes in server response times. It means that 95% of requests are served within that response time, and therefore, 5% of the requests experience slower response times.

Database Counters

When interpreting database statistics for this enterprise publishing environment, be aware that most of the visitors have read-only permissions.

Metric	Value
Read/Write Ratio (IO Per Database)	99.999 : 0.001
Average Disk queue length	0.35
Disk Queue Length: Reads	34
Disk Queue Length: Writes	2.5
Disk Reads/sec	131.33
Disk Writes/sec	278.33
SQL Compilations/second	2.36
SQL Re-compilations/second	94.80
SQL Locks: Average Wait Time	0 ms

Metric	Value
SQL Locks: Lock Wait Time	0 ms
SQL Locks: Deadlocks Per Second	0
SQL Latches: Average Wait Time	0.25 ms
SQL Cache Hit Ratio	>99%

Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Technical case study

This article describes a specific deployment of Microsoft SharePoint Server 2010, that includes the following:

- Technical case study environment specifications, such as hardware, farm topology and configuration.
- The workload, that includes the number, and types, of users or clients, and environment usage characteristics.
- Technical case study farm dataset, that includes database contents and search indexes.
- Health and performance data that is specific to the environment.

In this article:

Prerequisite information

Before reading this document, make sure that you understand the key concepts behind SharePoint Server 2010 capacity management. The following documentation will help you learn about the recommended approach to capacity management and provide context for helping you understand how to make effective use of the information in this document, and also define the terms used throughout this document.

For more conceptual information about performance and capacity that you might find valuable in understanding the context of the data in this technical case study, see the following documents:

- Capacity management and sizing for SharePoint Server 2010
- SharePoint Server 2010 capacity management: Software boundaries and limits

Introduction to this environment

This white paper describes an actual SharePoint Server 2010 environment at Microsoft. Use this document to compare to your planned workload and usage characteristics. If your planned design is similar, you can use the deployment described here as a starting point for your own installation.

This document includes the following:

- **Specifications**, which include hardware, topology, and configuration.
- **Workload**, which is the demand on the farm that includes the number of users, and the usage characteristics.
- **Dataset** that includes database sizes.
- **Health and performance** data that is specific to the environment.

This document is part of a series of Performance and capacity technical case studies (SharePoint Server 2010) about SharePoint environments at Microsoft.

The SharePoint environment described in this document is a production environment at a large, geographically distributed company. The environment hosts very important team sites and publishing portals for internal teams for enterprise collaboration, organizations, teams, and projects. Sites created in this environment are used as communication portals, applications for business solutions, and general collaboration. Self-service site creation is used to provision site collections. Custom code is not permitted.

As many as 88,900 unique users visit the environment on a busy day, generating up to 669 requests per second (RPS) during peak hours. Because this is an intranet site, all users are authenticated.

The information that is provided in this document reflects the enterprise intranet publishing environment on a typical day.

Specifications

This section provides detailed information about the hardware, software, topology, and configuration of the case study environment.

Hardware

This section provides details about the server computers that were used in this environment.

Note

- This environment is scaled to accommodate pre-release builds of SharePoint Server 2010 and other products. Hence, the hardware deployed has larger capacity than necessary to serve the demand typically experienced by this environment. This hardware is described only to provide additional context for this environment and serve as a starting point for similar environments.
- It is important to conduct your own capacity management based on your planned workload and usage characteristics. For more information about the capacity management process, see Capacity management and sizing for SharePoint Server 2010.

Web Servers

There are four Web servers in the farm, each with identical hardware. Three serve content, and the fourth is a dedicated search crawl target.

Web Server	WFE1-4
Processor(s)	2 quad core @ 2.33 GHz
RAM	32 GB
OS	Windows Server® 2008, 64 bit
Size of the SharePoint drive	205 GB
Number of network adapters	2
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Load balancer type	Hardware load balancing
Software version	SharePoint Server 2010 (pre-release version)
Services running locally	Central Administration Microsoft SharePoint Foundation Incoming E-Mail Microsoft SharePoint Foundation Web Application Microsoft SharePoint Foundation Workflow Timer Service Search Query and Site Settings Service SharePoint Server Search
Services consumed from a federated Services	User Profile Service

Web Server	WFE1-4
farm	Web Analytics Web Service Business Data Connectivity Service Managed Metadata Web Service

Application Server

There are four application servers in the farm, each with identical hardware.

Application Server	APP1-4
Processor(s)	4 six core @ 2.40 GHz
RAM	64 GB
OS	Windows Server 2008, 64 bit
Size of the SharePoint drive	300 GB
Number of network adapters	1
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Load balancer type	Hardware load balancing
Software version	SharePoint Server 2010 (pre-release version)
Services running locally	Office Web Apps Excel PowerPoint Secure Store Usage and Health State Service

Database Servers

There is a SQL cluster with 2 database servers, each with identical hardware, one of the servers is active and the other is passive for redundancy.

Database Server	DB1-2
Processor(s)	4 quad core @ 2.4 GHz
RAM	32 GB
OS	Windows Server 2008, 64-bit
Storage and geometry	(1.25 TB * 7) + 3 TB Disk 1-4: SQL Data Disk 5: Logs Disk 6: TempDB
Number of network adapters	2
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Software version	SQL Server 2008

Topology

The following diagram shows the topology for this farm.

Intranet Collaboration

Farm Topology

Front end

Web Servers
SharePoint Server 2010
pre-release version

Application Servers
SharePoint Server 2010
pre-release version

Also uses federated services from the Services farm.

Services hosted:
Central Administration, Office Web Apps, Excel, Secure Store, Usage, PowerPoint, State Service

Web Servers

Processor	2px4c@2.33 GHz
RAM	32 GB
NIC Speed	1 GB Full

Application Servers

Processor	4px6c@2.40 GHz
RAM	64 GB
NIC Speed	1 GB Full

Back end

Database Servers
SQL Server 2008

SQL Cluster

Database Servers

Processor	4px4c@2.4 GHz
RAM	32 GB
Storage	(1.25 TB * 7) + 3 TB

Configuration

The following table enumerates settings that were changed that affect performance or capacity in the environment.

Setting	Value	Notes
Usage Service Trace Log – days to store log files (default: 14 days)	5 days	The default is 14 days. Lowering this setting can save disk space on the server where the log files are stored.
QueryLoggingThreshold SharePoint Foundation Database – change QueryLoggingThreshold to 1 second	1 second	The default is 5 seconds. Lowering this setting can save bandwidth and CPU on the database server.
Database Server – Default Instance Max degree of parallelism	1	The default is 0. To ensure optimal performance, we strongly recommend that you set max degree of parallelism to 1 for database servers that host SharePoint Server 2010 databases. For more information about how to set max degree of parallelism, see max degree of parallelism Option(http://go.microsoft.com/fwlink/?LinkId=189030).

Workload

This section describes the workload, which is the demand on the farm that includes the number of users, and the usage characteristics.

Workload Characteristics	Value
Average Requests per Second (RPS)	157
Average RPS at peak time (11 AM-3 PM)	350
Total number of unique users per day	69,702
Average concurrent users	420
Maximum concurrent users	1,433
Total # of requests per day	18,866,527

This table shows the number of requests for each user agent.

User Agent	Requests	Percentage of Total
Search (crawl)	6,384,488	47%
DAV	2,446,588	18%
Browser	730,139	5%
OneNote	665,334	5%
Office Web Applications	391,599	3%
SharePoint Designer	215,695	2%

Dataset

This section describes the case study farm dataset that includes database sizes and Search indexes.

Dataset Characteristics	Value
Database size (combined)	7.5 TB
BLOB size	6.8 TB
Number of content databases	87
Number of Web applications	2
Number of site collections	34,423
Number of sites	101,598
Search index size (number of items)	23 million

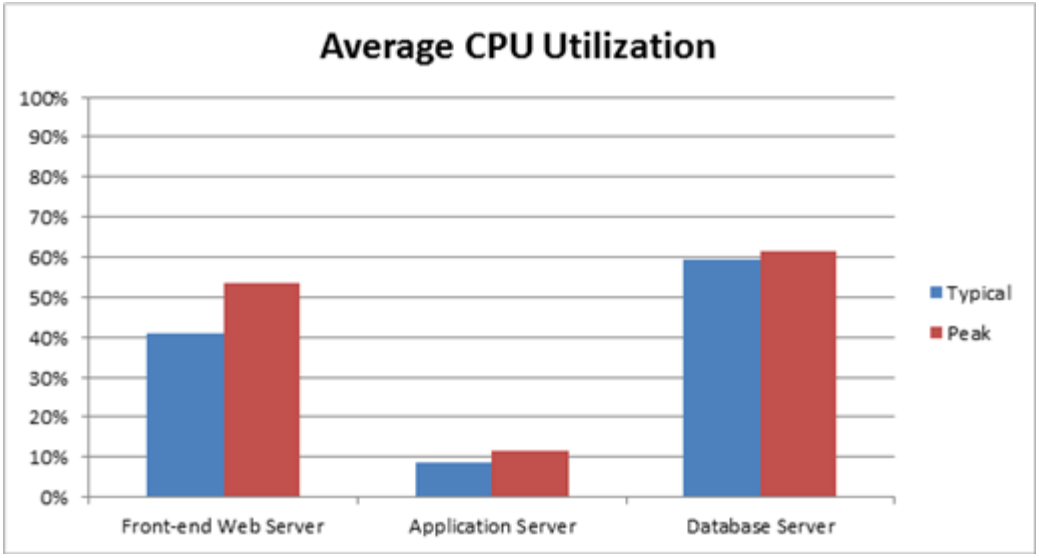
Health and Performance Data

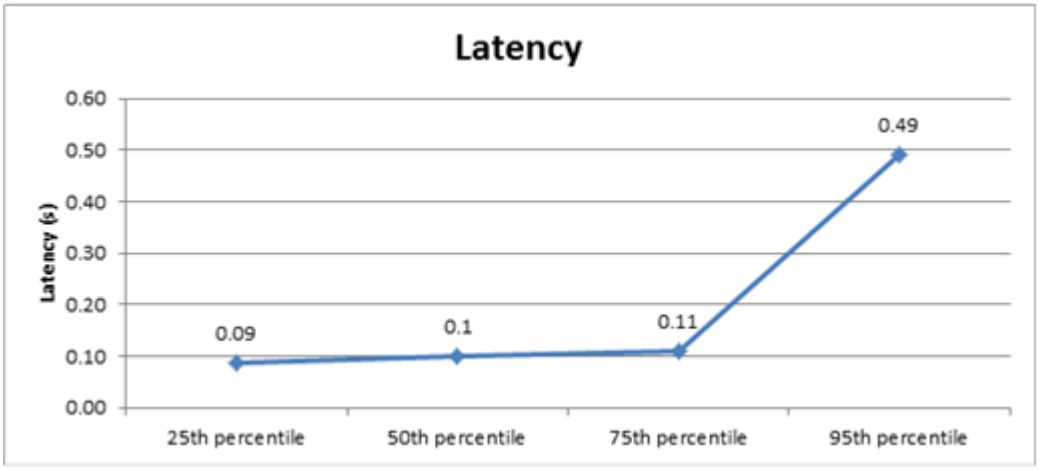
This section provides health and performance data that is specific to the Case Study environment.

General Counters

Metric	Value
Availability (uptime)	99.1%
Failure Rate	0.9%
Average memory used	3.4 GB
Maximum memory used	16.1 GB
Search Crawl % of Traffic (Search client requests / total requests)	47%
ASP.NET Requests Queued	0.00

The following charts show the average CPU utilization and latency for this environment:





In this document, latency is divided into four categories. The 50th percentile latency is typically used to measure the server's responsiveness. It means that half of the requests are served within that response time. The 95th percentile latency is typically used to measure spikes in server response times. It means that 95% of requests are served within that response time, and therefore 5% of the requests experience slower response times.

Database counters

Metric	Value
Read/Write Ratio (IO Per Database)	99.8 : 0.2
Average Disk queue length	2.3
Disk Queue Length: Reads	2
Disk Queue Length: Writes	0.3
Disk Reads/sec	131.33
Disk Writes/sec	278.33
SQL Compilations/second	9.9
SQL Re-compilations/second	0.07
SQL Locks: Average Wait Time	225 ms
SQL Locks: Lock Wait Time	507 ms
SQL Locks: Deadlocks Per Second	3.8
SQL Latches: Average Wait Time	14.3 ms

Metric	Value
SQL Server: Buffer Cache Hit Ratio	74.3%

Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Lab study

This article contains guidance on performance and capacity planning for an enterprise intranet collaboration solution that is based on Microsoft SharePoint Server 2010. It includes the following:

- Lab environment specifications, such as hardware, farm topology and configuration
- Test farm dataset
- Test results analysis which should help you determine the hardware, topology and configuration that you must have to deploy a similar environment, and optimize your environment for appropriate capacity and performance characteristics

In this article:

Introduction to this environment

This document provides guidance about scaling out and scaling up servers in a SharePoint Server 2010 enterprise intranet collaboration solution, based on a testing environment at Microsoft. Capacity planning informs decisions on purchasing hardware and making system configurations to optimize your solution.

Different scenarios have different requirements. Therefore, it is important to supplement this guidance with additional testing on your own hardware and in your own environment. If your planned design and workload resembles the environment described in this document, you can use this document to draw conclusions about scaling your environment up and out.

This document includes the following:

- **Specifications**, which include hardware, topology, and configuration
- The **workload**, which is the demand on the farm, includes the number of users, and the usage characteristics
- The **dataset**, such as database sizes
- Test results and analysis for **scaling out Web servers**
- Test results and analysis for **scaling up Web servers**
- Test results and analysis for **scaling out database servers**

- **Comparison between Microsoft Office SharePoint Server 2007 and SharePoint Server 2010** about throughput and effect on the web and database servers

The SharePoint Server 2010 environment described in this document is a lab environment that mimics a production environment at a large company. The production environment hosts very important team sites and publishing portals for internal teams for enterprise collaboration, organizations, teams, and projects. Employees use that production environment to track projects, collaborate on documents, and share information within their organization. The environment includes a large amount of small sites used for ad-hoc projects and small teams. For details about the production environment, see Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Technical case study.

Before reading this document, make sure that you understand the key concepts behind capacity management in SharePoint Server 2010. The following documentation will help you learn about the recommended approach to capacity management and provide context for helping you understand how to make effective use of the information in this document, and also define the terms used throughout this document.

- Capacity management and sizing overview for SharePoint Server 2010
- SharePoint Server 2010 capacity management: Software boundaries and limits

Also, we encourage you to read the following:

- Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

Glossary

There are some specialized terms that you will encounter in this document. Here are some key terms and their definitions.

- **RPS:** Requests per second. The number of requests received by a farm or server in one second. This is a common measurement of server and farm load.

  Note that requests differ from page loads; each page contains several components, each of which creates one or more requests when the page is loaded. Therefore, one page load creates several requests. Typically, authentication checks and events using insignificant resources are not counted in RPS measurements.
- **Green Zone:** This is the state at which the server can maintain the following set of criteria:
  - The server-side latency for at least 75% of the requests is less than 1 second.
  - All servers have a CPU Utilization of less than 50%.

    Note:
    Because this lab environment did not have an active search crawl running, the database server was kept at 40% CPU Utilization or lower, to reserve 10% for the search crawl load. This assumes Microsoft SQL Server Resource Governor is used in production to limit Search crawl load to 10% CPU.
  - Failure rate is less than 0.01%.
- **Red Zone (Max):** This is the state at which the server can maintain the following set of criteria:

    - HTTP request throttling feature is enabled, but no 503 errors (Server Busy) are returned.
    - Failure rate is less than 0. 1%.
    - The server-side latency is less than 3 seconds for at least 75% of the requests.
    - Database server CPU utilization is less than 80%, which allows for 10% to be reserved for the Search crawl load, limited by using SQL Server Resource Governor.
- **AxBxC (Graph notation):** This is the number of Web servers, application servers, and database servers respectively in a farm. For example, 8x1x2 means that this environment has 8 Web servers, 1 application server, and 2 database servers.
- **MDF and LDF:**SQL Server physical files. For more information, see Files and Filegroups Architecture.

Overview

This section provides an overview to our scaling approach, to the relationship between this lab environment and a similar case study environment, and to our test methodology.

Scaling approach

This section describes the specific order that we recommend for scaling servers in your environment, and is the same approach we took for scaling this lab environment. This approach will enable you to find the best configuration for your workload, and can be described as follows:

- First, we scaled out the Web servers. These were scaled out as far as possible under the tested workload, until the database server became the bottleneck and was unable to accommodate any more requests from the Web servers.
- Second, we scaled out the database server by moving half of the content databases to another database server. At this point, the Web servers were not creating sufficient load on the database servers. Therefore, they were scaled out additionally.
- In order to test scale up, we tried another option which is scaling up the Web servers instead of scaling them out. Scaling out the Web servers is generally preferred over scaling them up because scaling out provides better redundancy and availability.

Correlating the lab environment with a production environment

The lab environment outlined in this document is a smaller scale model of a production environment at Microsoft, and although there are significant differences between the two environments, it can be useful to examine them side by side because both are enterprise collaboration environments where the patterns observed should be similar.

The lab environment contains a subset of the data from the production environment, and also some modifications to the workload. This influences the test results with regard to Web server memory usage, because the object cache on the production environment receives a larger amount of hits on unique sites, and therefore uses more memory. The lab environment also has less data, and most of it is

cached in memory as opposed to the production environment which carries over seven terabytes of data, so that the database server on the production environment is required to perform more disk reads than the database server in the lab environment. Similarly, the hardware that was used in the lab environment is significantly different from the production environment it models, because there is less demand on those resources. The lab environment relies on more easily available hardware.

To get a better understanding of the differences between the environments, read the Specifications section in this document, and compare it to the specifications in the Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Technical case study.

Methodology and Test Notes

This document provides results from a test lab environment. Because this was a lab environment and not a production environment, we were able to control certain factors to show specific aspects of performance for this workload. In addition, certain elements of the production environment, listed here, were left out of the lab environment to simplify testing overhead. We do not recommend omitting these elements for production environments.

- Between test runs, we modified only one variable at a time, to make it easy to compare results between test runs.
- The database servers that were used in this lab environment were not part of a cluster because redundancy was not necessary for the purposes of these tests.
- Search crawl was not running during the tests, whereas it might be running in a production environment. To take this into account, we lowered the SQL Server CPU utilization in our definition of 'Green Zone' and 'Max' to accommodate the resources that a search crawl would have consumed if it were running at the same time with our tests. To learn more about this, read Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

Specifications

This section provides detailed information about the hardware, software, topology, and configuration of the lab environment.

Hardware

The following sections describe the hardware that was used in this lab environment.

Web and Application servers

There are from one to eight Web servers in the farm, plus one Application server.

Web Server	WFE1-8 and APP1
Processor(s)	2 quad-core 2.33 GHz processors

Web Server	WFE1-8 and APP1
RAM	8 GB
Operating system	Windows 2008 Server R2
Size of the SharePoint drive	80 GB
Number of network adapters	2
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Load balancer type	Windows NLB
Services running locally	WFE 1-8: Basic Federated Services. This included the following: Timer Service, Admin Service, and Trace Service. APP1: Word Automation Services, Excel Services and SandBoxed Code Services.

Database Servers

There are from two to three database servers, up to two running the default SQL Server instance housing the content databases, and one running the logging database. The logging database is not tracked in this document.

Note:

If you enable usage reporting, we recommend that you store the logging database on a separate Logical Unit Number (LUN). For large deployments and some medium deployments, a separate LUN will not be sufficient, as the demand on the server's CPU may be too high. In that case, you will need a separate database server box for the logging database. In this lab environment, the logging database was stored in a separate instance of SQL Server, and its specifications are not included in this document.

Database Server – Default Instance	DB1-2
Processor(s)	4 dual-core 3.19 GHz processors
RAM	32 GB
Operating system	Windows 2008 Server R2
Storage and geometry	Direct Attached Storage (DAS) Internal Array with 5 x 300GB 10krpm disk External Array with 15 x 450GB 15krpm disk

Database Server – Default Instance	DB1-2
	6 x Content Data (External RAID0, 2 spindles 450GB each) 2 x Content Logs (Internal RAID0, 1 spindle 300GB each) 1 x Temp Data (Internal RAID0, 2 spindles 150GB each) 1 x Temp Log (Internal RAID0, 2 spindles 150GB each) 2 x Backup drive (Internal RAID0, 1 spindle each, 300GB each)
Number of network adapters	1
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Software version	SQL Server 2008 R2 (pre-release version)

Topology

The following diagram shows the topology in this lab environment:

Collaboration Test Environment

Farm Topology

Front end

Web Servers
SharePoint Server 2010
pre-release version

WFE2-8 when scaled out

Application Servers
SharePoint Server 2010
pre-release version

Services hosted:
Sandboxed Code Services, Office Web Applications, Excel Services

Web and Application Servers

Processor	2px4c@2.33 GHz
RAM	8 GB
NIC Speed	1 GB Full

Back end

Database Servers
SQL Server 2008 R2 pre-release version

Database Servers

Processor	4px2c@3.19 GHz
RAM	32 GB
Storage	15*450 GB

Configuration

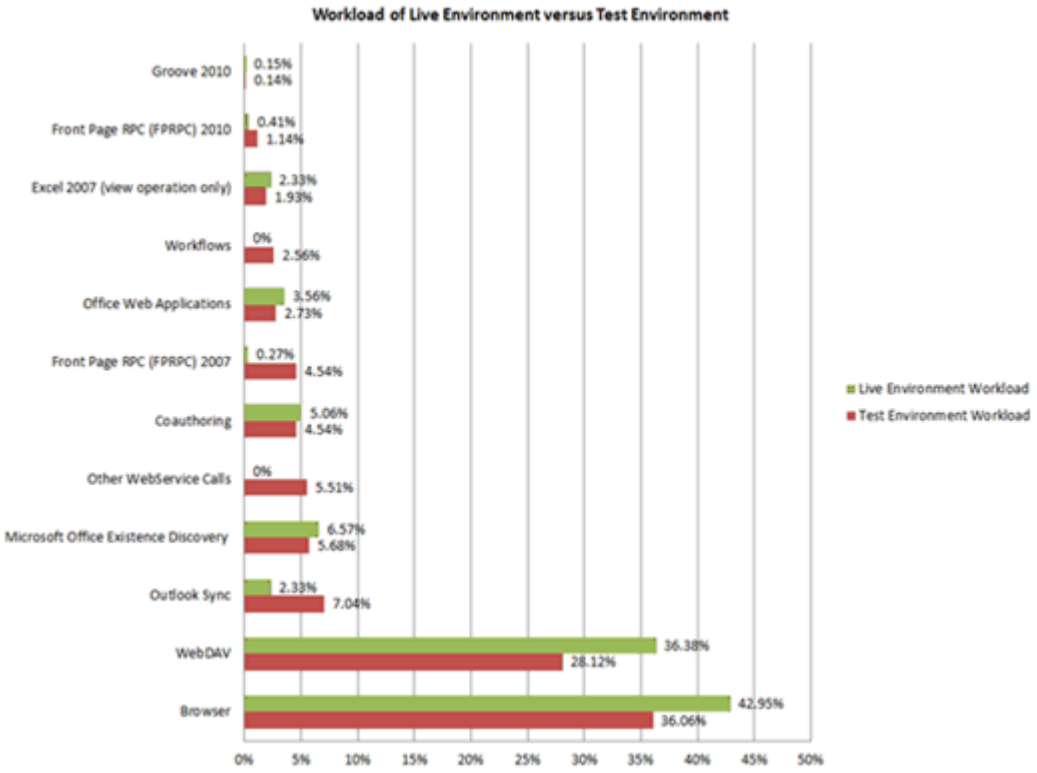
To allow for the optimal performance, the following configuration changes were made in this lab environment.

Setting	Value	Notes
Site Collection		
Blob Caching	On	The default is Off. Enabling Blob Caching improves server efficiency by reducing calls to the database server for static page resources that may be frequently requested.
Database Server – Default Instance		
Max degree of parallelism	1	The default is 0. To ensure optimal performance, we strongly recommend that you set max degree of parallelism to 1 for database servers that host SharePoint Server databases. For more information about how to set max degree of parallelism, see max degree of parallelism Option(http://go.microsoft.com/fwlink/?LinkId=189030).

Workload

The transactional mix for the lab environment described in this document resembles the workload characteristics of a production environment at Microsoft. For more information about the production environment, see Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Technical case study.

Here are the details of the mix for the lab tests run against SharePoint Server 2010 compared to the production environment. Although there are some minor differences in the workloads, both represent a typical transactional mix on an enterprise collaboration environment.



Dataset

The dataset for the lab environment described in this document is a subset of the dataset from a production environment at Microsoft. For more information about the production environment, see Microsoft SharePoint Server 2010 enterprise intranet collaboration environment: Technical case study.

Dataset Characteristics	Value
Database size (combined)	130 GB
BLOB size	108.3 GB
Number of content databases	2
Number of site collections	181
Number of Web applications	1
Number of sites	1384

Results and Analysis

The following results are ordered based on the scaling approach described in the Overview section of this document.

Web Server Scale Out

This section describes the test results that were obtained when we scaled out the number of Web servers in this lab environment.

Test methodology

- Add Web servers of the same hardware specifications, keeping the rest of the farm the same.
- Measure RPS, latency, and resource utilization.

Analysis

In our testing, we found the following:

- The environment scaled up to four Web servers per database server. However, the increase in throughput was non-linear especially on addition of the fourth Web server.
- After four Web servers, there are no additional gains to be made in throughput by adding more Web servers because the bottleneck at this point was the database server CPU Utilization.
- The average latency was almost constant throughout the whole test, unaffected by the number of Web servers and throughput.

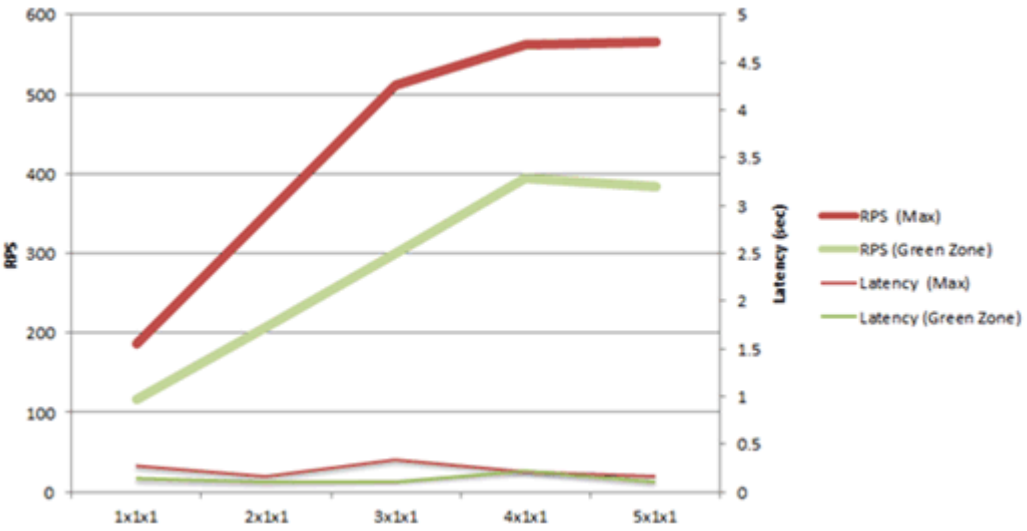
Note:
The conclusions described in this section are hardware specific, and the same throughput might have been achieved by a larger number of lower-end hardware, or a smaller number of higher-end hardware. Similarly, changing the hardware of the database server would affect the results. To get an idea on how much of a difference the hardware of the Web servers can affect these results, see the Web Server Scale Up section.

Results graphs and charts

In the following graphs, the x axis shows the change in the number of Web servers in the farm, scaling from one Web server (1x1x1) to five Web servers (5x1x1).

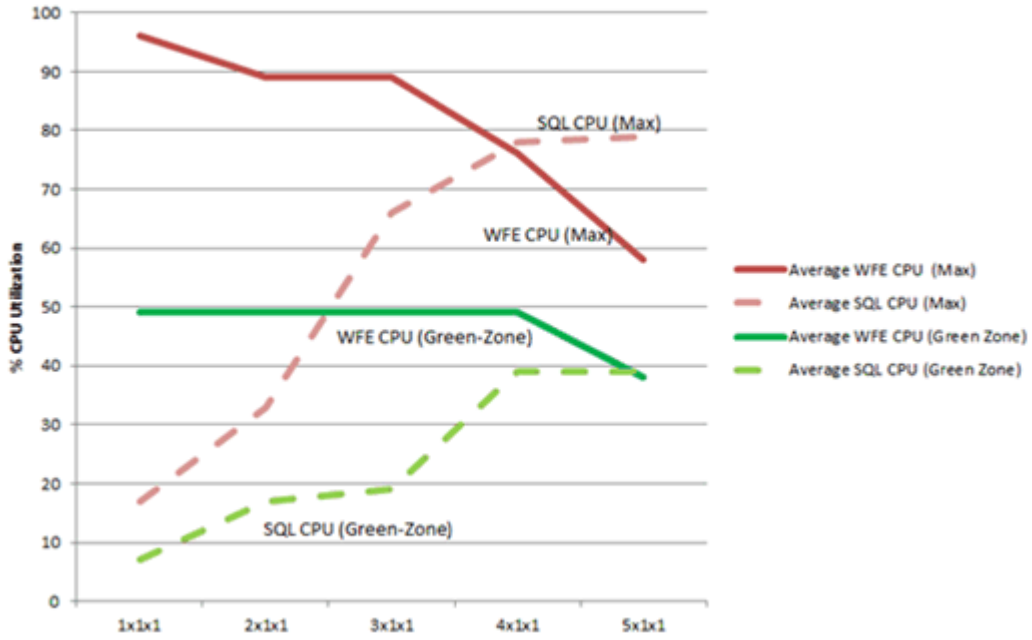
1. Latency and RPS

The following graph shows how scaling out (adding Web servers) affects latency and RPS.



2. Processor utilization

The following graph shows how scaling out the Web servers affects processor utilization on the Web server(s) and the database server.



3. SQL Server I/O operations per section (IOPs) for MDF and LDF files

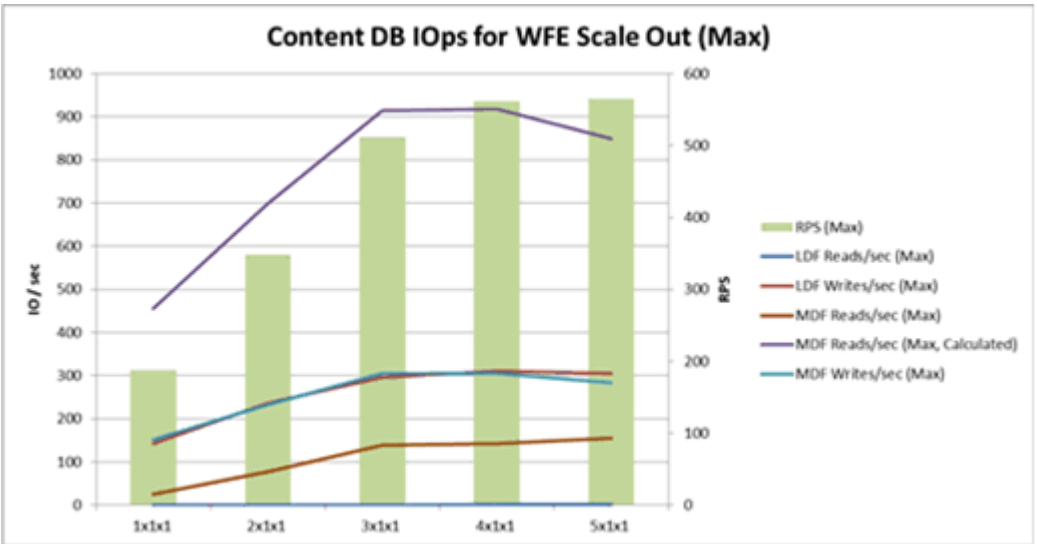
The following graphs show how the IOPs on the content databases change as the number of Web servers is scaled out. These are measured by looking at the following performance counters:

- PhysicalDisk: Disk Reads / sec

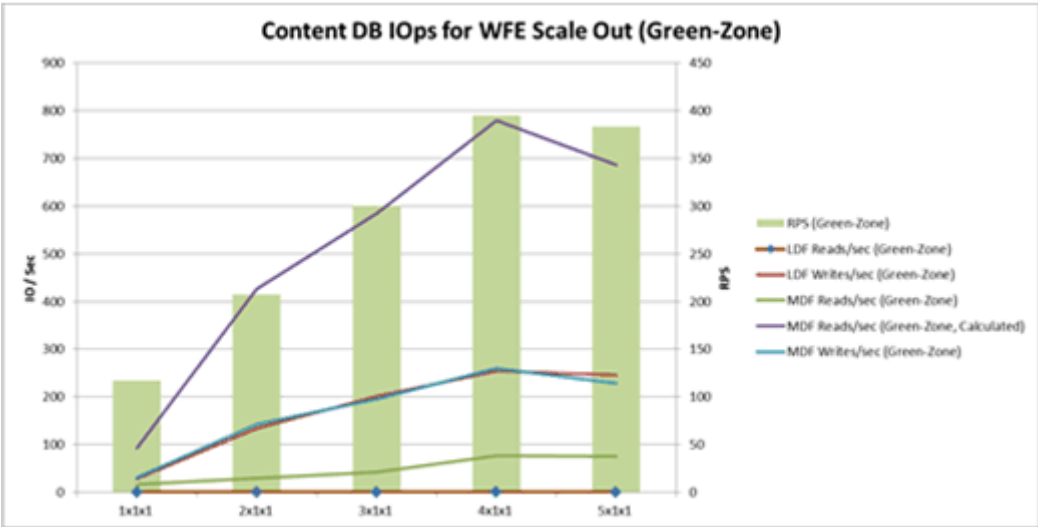
- PhysicalDisk: Disk Writes / sec

In this lab environment, we determined that our data on IOPs was not representative of a production environment because our dataset was so small that we could fit much more of it in cache than would be possible in the production environment we are modeling. We calculated projected reads by multiplying the value of the data we had from the lab for writes/second by the ratio of reads to writes in our production environment. The results in this section are averages. But there are also spikes that occur during certain operations which have to be accounted for. To learn more about how to estimate IOPs needed, see Storage and SQL Server capacity planning and configuration (SharePoint Server 2010).

Maximum:



Green Zone:



Example of how to read these graphs:

An organization with a workload similar to that described in this document that expects 300 RPS to be their green zone, could use 3x1x1 topology, and would use approximately 600 Physical Disk reads/sec on the MDF file.

Database Server Scale Out

This section describes the test results that were obtained when we scaled out the number of database servers in this lab environment.

Test methodology

- Have two content databases on one database server, and then split them to two servers to effectively double the processor cores and memory available to the database servers in the environment.
- Keep the total IOPs capacity constant even while adding a database server. This means that the number of reads/sec and writes/sec that the disks could perform for each content database did not change despite splitting the content onto two database servers instead of one.

Analysis

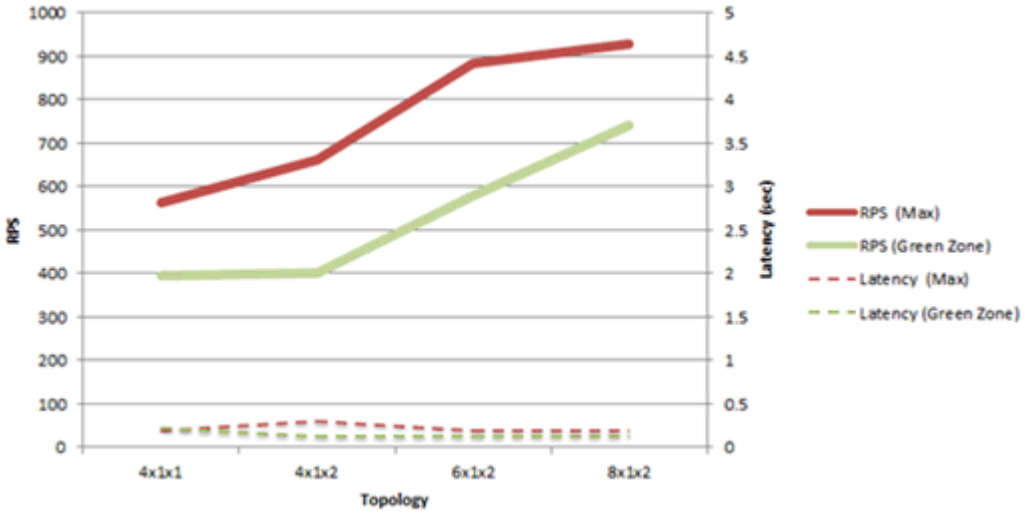
- The first bottleneck in the 4x1x2 environment was the database server CPU utilization. There was close to a linear scale when we added more processor and memory power.
- Scaling to four Web servers and 2 database servers did not provide much additional RPS because the CPU utilization on the Web servers was close to 100%.
- When we scaled out database servers (by adding one additional database server) and added four Web servers, performance scaled almost linearly. The bottleneck at that point shifted from being the database server CPU utilization to the content database disk IOPs.
- No additional tests were performed in this lab environment to scale out past 8x1x2. However we expect that additional IOPs capacity would additionally increase throughput.
- A correlation between the IOPs used and the RPS achieved by the tests was observed

Results graphs and charts

In the following graphs, the x axis is always showing four Web servers together with 1 application server and 1 database server (4x1x1) scaling out to eight Web servers together with two database servers (8x1x2). Some also show 1x1x1 or 4x1x2.

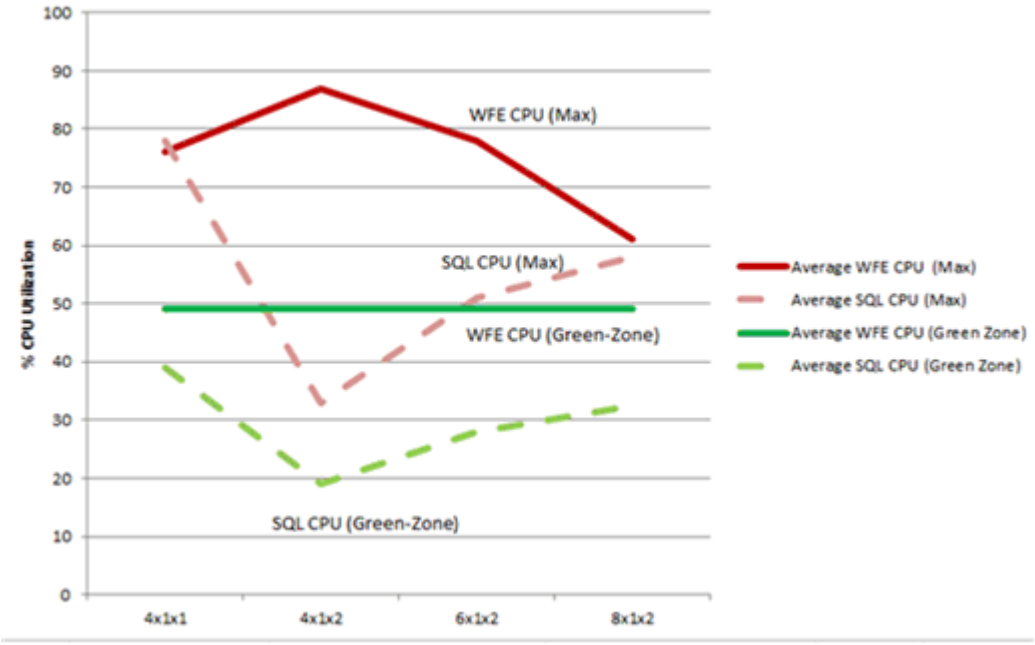
1. Latency and RPS

The following graph shows how scaling out both Web servers and database servers affects latency and RPS.



2. Processor utilization

The following graphs show how scaling out affects processor utilization.



3. Memory utilization at scale out

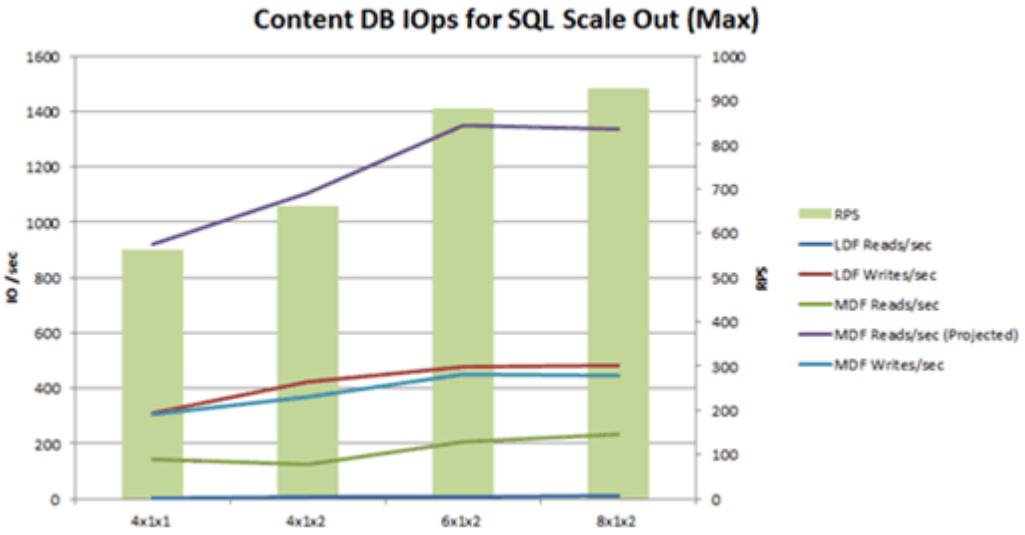
Throughout our testing, we've observed that the larger the number of site collections in an environment, the more the memory consumed. For example, in the tests here where 181 site collections were accessed, the main w3wp process used up 1.8 GB of RAM. For more examples, see the Performance and capacity technical case studies (SharePoint Server 2010). Additional content about memory

requirements for increased numbers of site collections is under development. Check back for new and updated content.

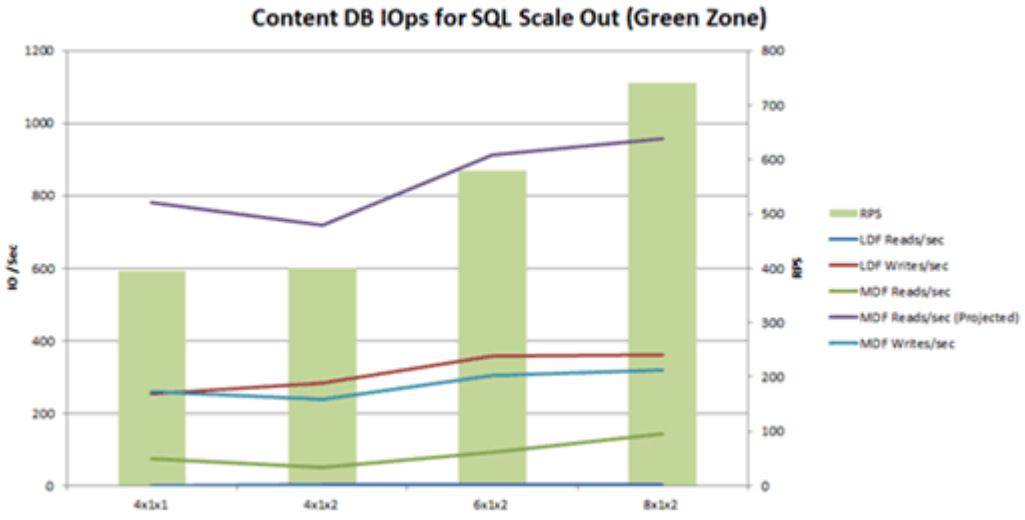
4. SQL Server I/O operations per section (IOPs) for MDF and LDF files

The following graphs show how the IOPs change as both the number of Web servers and the number of database servers is scaled out.

Maximum RPS



Green Zone RPS



Web server Scale Up

This section describes the test results that were obtained when we scaled up the Web servers in this lab environment.

Test methodology

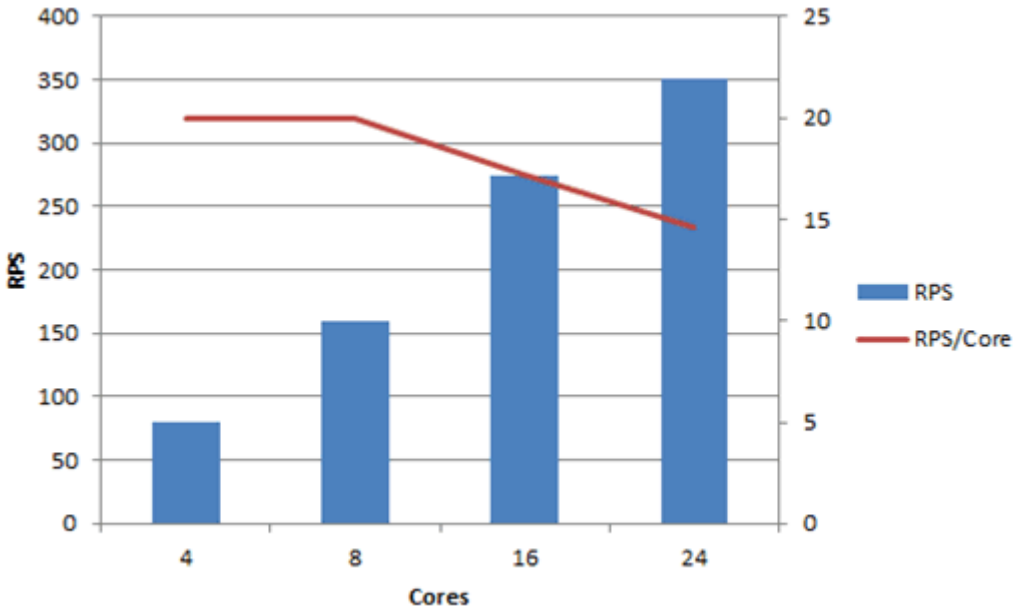
- Add more Web server processors, but keep the rest of the farm the same.

Analysis

- Scale is linear up to eight processor cores.
- Tests show that the environment can take advantage of a twenty-four core box, although there is some degradation as twenty-four cores are approached.

Results graphs and charts

In the following graph, the x axis is the number of processors and the amount of RAM on the Web server. The following graph shows how scaling up (adding processors) affects RPS on the Web server.



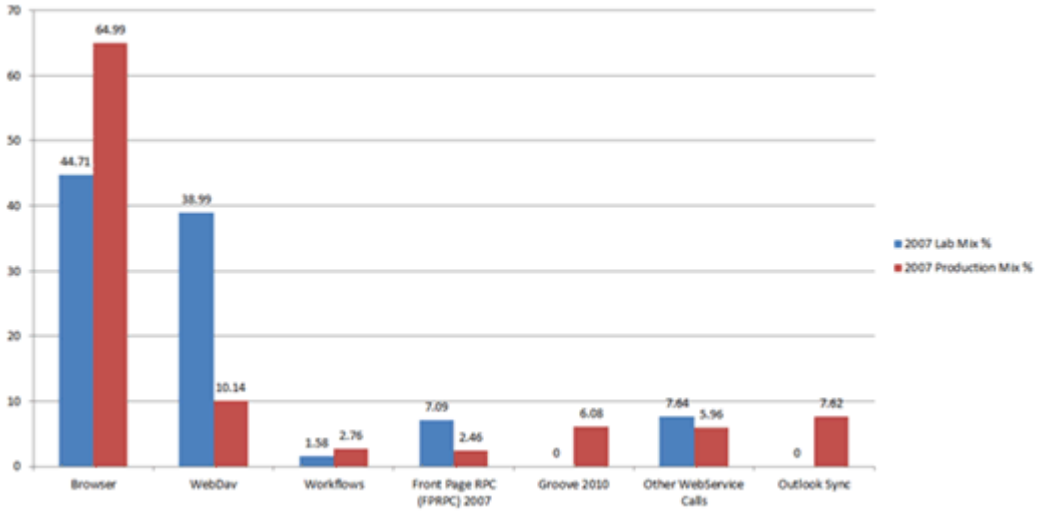
Comparing SharePoint Server 2010 and Office SharePoint Server 2007

This section provides information about how the capacity testing for this workload varied between SharePoint Server 2010 and Microsoft Office SharePoint Server 2007.

Workload

To compare SharePoint Server 2010 with Office SharePoint Server 2007, a different test mix was used than the one outlined in the Specifications section, because some SharePoint Server 2010 operations were not available in Office SharePoint Server 2007. The test mix for Office SharePoint Server 2007 is inspired by the same production environment that the SharePoint Server 2010 tests follow. However this was recorded before the upgrade to SharePoint Server 2010 on that environment.

The following graph shows the test mix for the lab and production environments for Office SharePoint Server 2007.



Test methodology

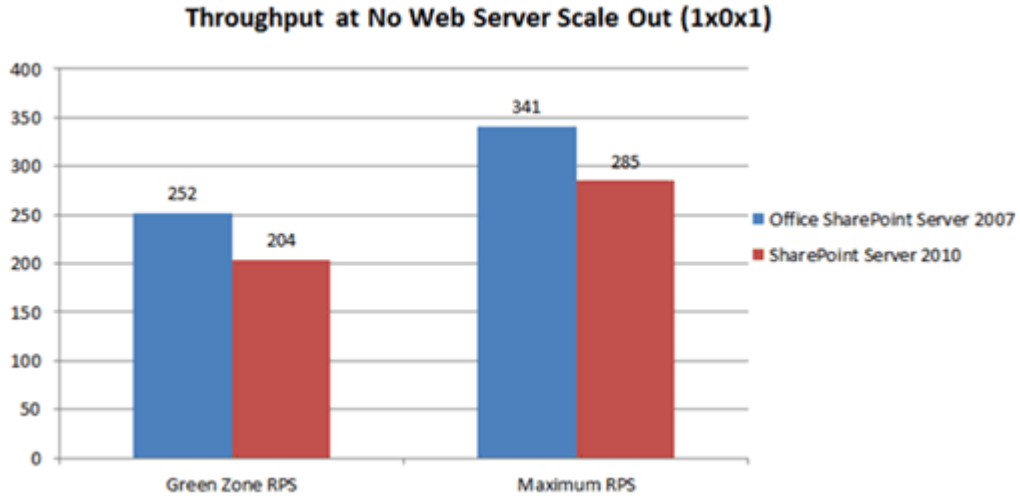
- The tests performed for this comparison were performed by creating an Office SharePoint Server 2007 environment, testing it with the workload outlined earlier in this section, and then upgrading the content databases to SharePoint Server 2010 without changing the clients using the environment, nor doing a visual upgrade. This upgraded environment was then re-tested for the SharePoint Server 2010 results with the same test mix which includes only Office SharePoint Server 2007 operations.
- The dataset was not modified after the content database upgrade for the SharePoint Server 2010 tests.
- The test mix for Office SharePoint Server 2007 excludes any new SharePoint Server 2010 specific operations, and resembles the enterprise intranet collaboration solution on the same production environment for Office SharePoint Server 2007, as described under the Workload section.

Analysis

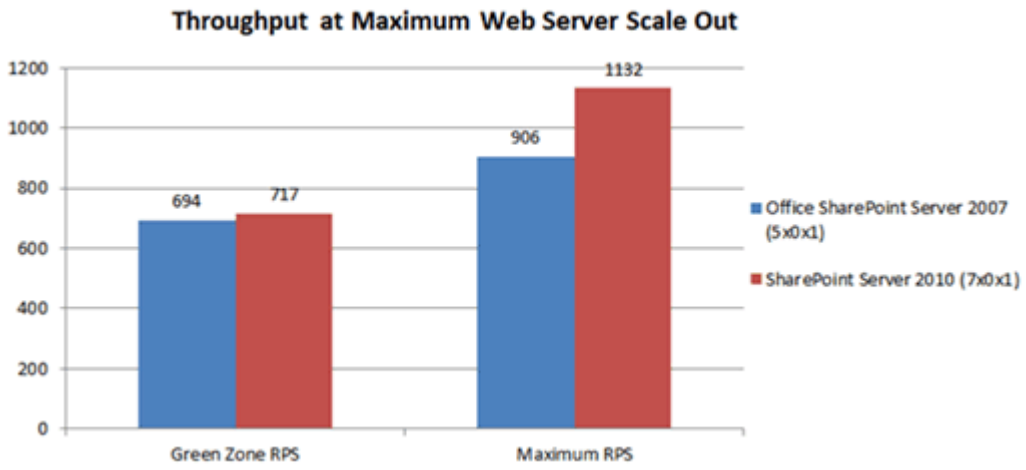
- When the same number of Web servers are stressed to their maximum throughput on SharePoint Server 2010 and Office SharePoint Server 2007, SharePoint Server 2010 achieves 20% less throughput compared to Office SharePoint Server 2007.
- When the Web servers were scaled out to maximize the database server usage, SharePoint Server 2010 was able to achieve 25% better throughput compared to Office SharePoint Server 2007. This reflects the improvements that were made in SharePoint Server 2010 to sustain larger deployments.
- When the web servers were scaled out to maximize the database server usage, SharePoint Server 2010 was SQL Server CPU Utilization bound, whereas Office SharePoint Server 2007 was Lock bound on the database tier. This means that increasing the processing power available to the database servers enables SharePoint Server 2010 to achieve better throughput than would be possible with the same hardware using Office SharePoint Server 2007. This is caused by the locking mechanisms in the database in Office SharePoint Server 2007 which are unaffected by improved hardware so that we were unable to push the database server's CPU Utilization past 80%.
- As a result of these findings outlined earlier in this section, on Office SharePoint Server 2007 the maximum throughput possible was achieved in a 5x0x1 topology whereas in SharePoint Server 2010 the maximum throughput possible with the same workload was achieved in a 7x0x1 topology, and yielded a 25% increased total RPS.

Results graphs and charts

The following graph shows the throughput without scaling out Web servers.



The following graph shows the throughput when Web servers were at maximum scale out.


Throughput at Maximum Web Server Scale Out
1200
1000
800
600
400
200
0
694
717
906
1132
Green Zone RPS
Maximum RPS
Office SharePoint Server 2007 (5x0x1)
SharePoint Server 2010 (7x0x1)

Microsoft SharePoint Server 2010 departmental collaboration environment: Technical case study:

This document describes a specific deployment of Microsoft SharePoint Server 2010. It includes the following:

- Technical case study environment specifications, such as hardware, farm topology and configuration
- The workload that includes the number, and types, of users or clients, and environment usage characteristics
- Technical case study farm dataset that includes database contents and Search indexes
- Health and performance data that is specific to the environment

In this article:

Prerequisite information

Before reading this document, make sure that you understand the key concepts behind SharePoint Server 2010 capacity management. The following documentation will help you learn about the recommended approach to capacity management and provide context for helping you understand how to make effective use of the information in this document, and also define the terms used throughout this document.

For more conceptual information about performance and capacity that you might find valuable in understanding the context of the data in this technical case study, see the following documents:

- Capacity management and sizing overview for SharePoint Server 2010
- SharePoint Server 2010 capacity management: Software boundaries and limits

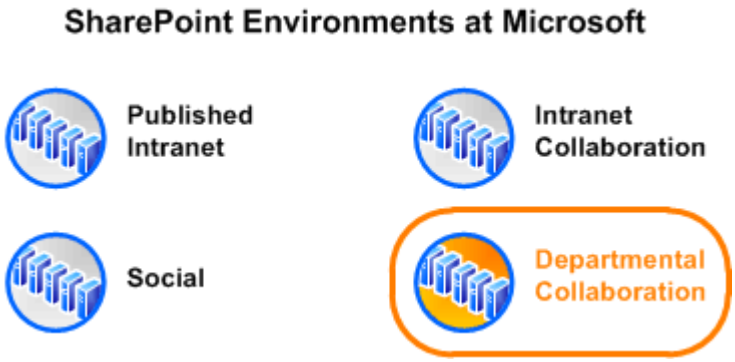
Introduction to this environment

This white paper describes an actual SharePoint Server 2010 environment at Microsoft. Use this document to compare with your planned workload and usage characteristics. If your planned design is similar, you can use the deployment described here as a starting point for your own installation.

This document includes the following:

- **Specifications**, which include hardware, topology and configuration
- **Workload**, which is the demand on the farm that includes the number of users, and the usage characteristics
- **Dataset** that includes database sizes
- **Health and performance** data that is specific to the environment

This document is part of a series of Performance and capacity technical case studies (SharePoint Server 2010) about SharePoint environments at Microsoft.



The SharePoint Server 2010 environment described in this document is a production environment at a large, geographically distributed company. Employees use this environment to track projects, collaborate on documents, and share information within their department. This environment is also used for internal testing, and is frequently upgraded to the latest SharePoint Server pre-release versions as they become available.

As many as 9,000 unique users visit the environment on a busy day, generating up to 470 requests per second (RPS) during peak hours. Because this is an intranet site, all users are authenticated.

The information that is provided in this document reflects the departmental collaboration environment on a typical day.

Specifications

This section provides detailed information about the hardware, software, topology, and configuration of the case-study environment.

Hardware

This section provides details about the server computers that were used in this environment.

Note

- This environment is scaled to accommodate pre-release builds of SharePoint Server 2010 and other products. Hence, the hardware deployed has larger capacity than necessary to serve the demand typically experienced by this environment. This hardware is described only to provide additional context for this environment and serve as a starting point for similar environments.
- It is important to conduct your own capacity management based on your planned workload and usage characteristics. For more information about the capacity management process, see Capacity management and sizing overview for SharePoint Server 2010.

Web Servers

There are four Web servers in the farm, each with identical hardware. Three serve content, and the fourth is a dedicated search crawl target.

Web Server	WFE1-2	WFE3-4
Processor(s)	2 quad core @ 2.33 GHz	2 quad core @ 2.33 GHz
RAM	32 GB	16 GB
Operating system	Windows Server 2008, 64 bit	Windows Server 2008, 64 bit
Size of the SharePoint drive	3x146GB 15K SAS (3 RAID 1 Disks) Disk 1: OS Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory	3x146GB 15K SAS (3 RAID 1 Disks) Disk 1: OS Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory
Number of network adapters	2	2
Network adapter 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	Search Query	WFE3 – No services WFE4 – Search crawl target

Application Server

There are four application servers in the farm.

Web Server	APP1-3	APP4
Processor(s)	2 quad core @ 2.33 GHz	2 quad core @ 2.33 GHz
RAM	16 GB	16 GB
Operating system	Windows Server 2008, 64 bit	Windows Server 2008, 64 bit
Size of the SharePoint drive	3x146GB 15K SAS (3 RAID 1 Disks) Disk 1: OS Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory	2x136GB 15K SAS (RAID 0) 4x60GB SSD, SATA (RAID 5) Disk 1: OS Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory
Number of network adapters	2	2
Network adapter 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	APP1 – Central Administration and all applications except for Office Web Applications APP2 – All applications (including Office Web Applications) APP3 – Office Web Applications	Search Crawler

Database Servers

There are three database servers, one running the default SQL Server instance housing the content databases, one running the Usage and Web Analytics databases, and one running the Search databases.

Database	DB1 – Default Instance	DB2	DB3
Processor(s)	4 quad core @ 3.2 GHz	2 quad core @ 3.2 GHz	2 quad core @ 3.2 GHz
RAM	32 GB	16 GB	32 GB
Operating system	Windows Server 2008 SP1, 64 bit	Windows Server 2008 SP1, 64 bit	Windows Server 2008 SP1, 64 bit
Storage and geometry	5x146GB 15K SAS + SAN Disk 1: OS (2 disk RAID 10) Disk 2: Swap (2 disk RAID 10) Disk 3: Direct Attached Storage (16 disk RAID 10, Temp DB data) SAS 146 GB 15K Disk 4: Direct Attached Storage (16 disk RAID 10, Temp DB data) SAS 146 GB 15K Disk 5-15: SAN using fiber connection. When possible, one database per two disks. Separating logs and data between LUNs. 15K drives.	6x450GB 15K SAS Directly attached 14x146GB 15K SAS Disk 1: Usage logs and OS Disk 2: Usage data	2x136GB 15K SAS (RAID 0) 6x60GB SSD, SATA (RAID 5) Disk 1: OS Disk 2: Swap and BLOB Cache Disk 3: Logs and Temp directory. Solid state drives. 6-60GB Solid state drives (RAID 5)
Number of network adapters	2	2	2
Network adapter speed	1 Gigabit	1 Gigabit	1 Gigabit
Authentication	Windows NTLM	Windows NTLM	Windows NTLM
Software version	SQL Server 2008	SQL Server 2008	SQL Server 2008 R2

Topology

The following diagram shows the topology for this farm.

Departmental Collaboration

Farm Topology

Front end

Web Servers
SharePoint Server 2010
pre-release version

WFE1 - Web plus Search Query | WFE2- Web plus Search Query | WFE3 - Web | WFE4 - Web plus Search Crawl

Application Servers
SharePoint Server 2010
pre-release version

APP1 - Central Admin | APP2 – Office Web Apps | APP3 - Office Web Apps | APP4 - Crawler

Web and Application Servers

Processor	2px4c@2.33 GHz
RAM	32 GB
NIC Speed	1 GB Full

Processor	2px4c@2.5 GHz
RAM	16 GB
NIC Speed	1 GB Full

Processor	2px4c@2.33 GHz
RAM	16 GB
NIC Speed	1 GB Full

Back end

Database Servers
SQL Server 2008

DB1 - SQL Server | DB2 - Usage Web Analytics | DB3 - Search Databases

Database servers

Server hardware specifications vary for each server.

Configuration

The following table enumerates settings that were made that affect performance or capacity in the environment.

Setting	Value	Notes
Site collection: Object Caching (On | Off) Anonymous Cache Profile (select) Anonymous Cache Profile (select) Object Cache (Off | n MB) Cross List Query Cache Changes (Every Time | Every n seconds)	 On Disabled Disabled On – 100GB 60 seconds	Enabling the output cache improves server efficiency by reducing calls to the database for data that is frequently requested.
Site collection cache profile (select)	Intranet (Collaboration Site)	"Allow writers to view cached content" is checked, bypassing the ordinary behavior of not letting people with edit permissions to have their pages cached.
Object Cache (Off | n MB)	On – 500 MB	The default is 100 MB. Increasing this setting enables additional data to be stored in the front-end Web server memory.
Usage Service: Trace Log – days to store log files (default: 14 days)	5 days	The default is 14 days. Lowering this setting can save disk space on the server where the log files are stored.
Query Logging Threshold: Microsoft SharePoint Foundation Database – configure QueryLoggingThreshold to 1 second	1 second	The default is 5 seconds. Lowering this setting can save bandwidth and CPU on the database server.
Database Server – Default Instance: Max degree of parallelism	1	The default is 0. To ensure optimal performance, we strongly recommend that you set max degree of parallelism to 1 for database servers that host SharePoint Server 2010 databases. For more information about how to set max degree of parallelism, see max degree

Setting	Value	Notes
		of parallelism Option (http://go.microsoft.com/fwlink/?LinkId=189030).

Workload

This section describes the workload, which is the demand on the farm that includes the number of users, and the usage characteristics.

Workload Characteristics	Value
Average Requests per Second (RPS)	165
Average RPS at peak time (11 AM-3 PM)	216
Total number of unique users per day	9186
Average concurrent users	189
Maximum concurrent users	322
Total # of requests per day	7,124,943

This table shows the number of requests for each user agent.

User Agent	Requests	Percentage of Total
Search (crawl)	4,373,433	67.61%
Outlook	897,183	13.87%
OneNote	456,917	7.06%
DAV	273,391	4.23%
Browser	247,303	3.82%
Word	94,465	1.46%
SharePoint Workspaces	70,651	1.09%
Office Web Applications	45,125	0.70%
Excel	8,826	0.14%
Access	1,698	0.03%

Dataset

This section describes the case study farm dataset that includes database sizes and Search indexes.

Dataset Characteristics	Value
Database size (combined)	1.8 TB
BLOB size	1.68 TB
Number of content databases	18
Total number of databases	36
Number of site collections	7,499
Number of Web applications	7
Number of sites	42,457
Search index size (number of items)	4.6 million

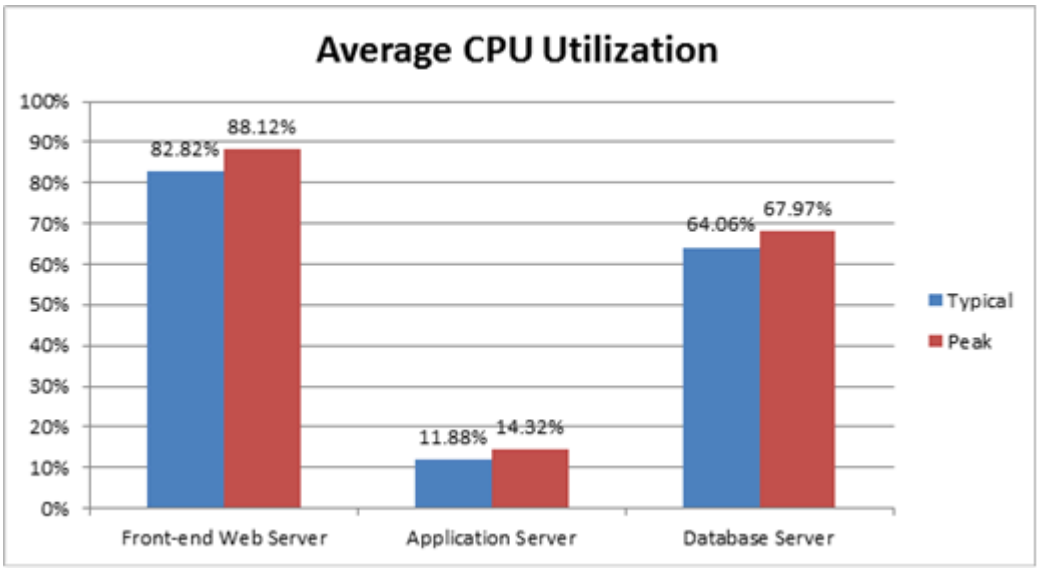
Health and Performance Data

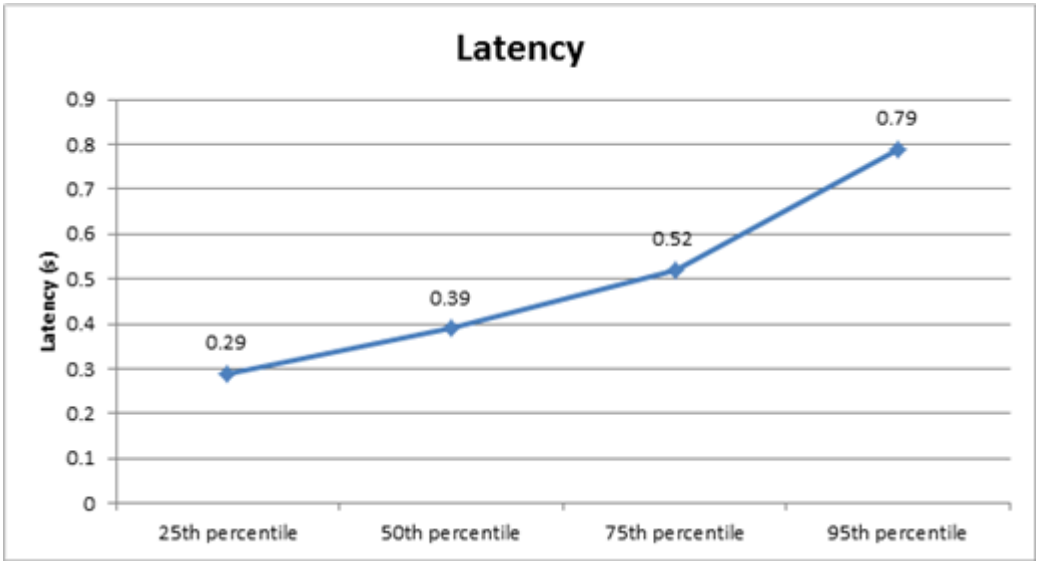
This section provides health and performance data that is specific to the case study environment.

General Counters

Metric	Value
Availability (uptime)	99.9995%
Failure Rate	0.0005%
Average memory used	0.89 GB
Maximum memory used	5.13 GB
Search Crawl % of Traffic (Search client requests / total requests)	82.5%

The following charts show the average CPU utilization and latency for this environment:





In this document, latency is divided into four categories. The 50th percentile latency is typically used to measure the server's responsiveness. It means that half of the requests are served within that response time. The 95th percentile latency is typically used to measure spikes in server response times. It means that 95% of requests are served within that response time, and therefore, 5% of the requests experience slower response times.

Database Counters

Metric	Value
Average Disk queue length	1.42
Disk Queue Length: Reads	1.38
Disk Queue Length: Writes	0.04
Disk Reads/sec	56.51
Disk Writes/sec	17.60
SQL Compilations/second	13.11
SQL Re-compilations/second	0.14
SQL Locks: Average Wait Time	294.56 ms
SQL Locks: Lock Wait Time	867.53 ms
SQL Locks: Deadlocks Per Second	1.87
SQL Latches: Average Wait Time	5.10 ms
SQL Cache Hit Ratio	99.77%

Microsoft SharePoint Server 2010 divisional portal environment: Lab study

This document provides guidance on performance and capacity planning for a divisional portal based on Microsoft SharePoint Server 2010. It includes the following:

- Test environment specifications, such as hardware, farm topology and configuration
- Test farm dataset
- Test data and recommendations for how to determine the hardware, topology and configuration that you must have to deploy a similar environment, and how to optimize your environment for appropriate capacity and performance characteristics

In this article:

Introduction to this environment

This document outlines the test methodology and results to provide guidance for capacity planning of a typical divisional portal. A divisional portal is a SharePoint Server 2010 deployment where teams mainly do collaborative activities and some content publishing. This document assumes a "division" to be an organization inside an enterprise with 1,000 to 10,000 employees.

Different scenarios will have different requirements. Therefore, it is important to supplement this guidance with additional testing on your own hardware and in your own environment. If your planned design and workload resembles the environment described in this document, you can use this document to draw conclusions about scaling your environment up and out.

When you read this document, you will understand how to do the following:

- Estimate the hardware that is required to support the scale that you need to support: number of users, load, and the features enabled.
- Design your physical and logical topology for optimal reliability and efficiency. High Availability/Disaster Recovery are not covered in this document.
- Understand the effect of ongoing search crawls on RPS for a divisional portal deployment.

The SharePoint Server 2010 environment described in this document is a lab environment that mimics a production environment at a large company. For details about the production environment, see Microsoft SharePoint Server 2010 departmental collaboration environment: Technical case study:.

Before reading this document, make sure that you understand the key concepts behind capacity management in SharePoint Server 2010. The following documentation will help you learn about the recommended approach to capacity management and provide context for helping you understand how to make effective use of the information in this document, and also define the terms used throughout this document.

- Capacity management and sizing overview for SharePoint Server 2010
- SharePoint Server 2010 capacity management: Software boundaries and limits

Also, we encourage you to read the following:

- Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

Glossary

There are some specialized terms that you will encounter in this document. Here are some key terms and their definitions.

- **RPS:** Requests per second. The number of requests received by a farm or server in one second. This is a common measurement of server and farm load.

  Note that requests differ from page loads; each page contains several components, each of which creates one or more requests when the page is loaded. Therefore, one page load creates several requests. Typically, authentication checks and events using insignificant resources are not counted in RPS measurements.
- **Green Zone:** This is the state at which the server can maintain the following set of criteria:
  - The server-side latency for at least 75% of the requests is less than .5 second.
  - All servers have a CPU Utilization of less than 50%.

  Note:
  Because this lab environment did not have an active search crawl running, the database server was kept at 40% CPU Utilization or lower, to reserve 10% for the search crawl load. This assumes Microsoft SQL Server Resource Governor is used in production to limit Search crawl load to 10% CPU.
  - Failure rate is less than 0.01%.
- **Red Zone (Max):** This is the state at which the server can maintain the following set of criteria:
  - HTTP request throttling feature is enabled, but no 503 errors (Server Busy) are returned.
  - Failure rate is less than 0. 1%.
  - The server-side latency is less than 1 second for at least 75% of the requests.
  - Database server CPU utilization is less than or equal to 75%, which allows for 10% to be reserved for the Search crawl load, limited by using SQL Server Resource Governor.
  - All Web servers have a CPU Utilization of less than or equal to 75%.

- **AxBxC (Graph notation):** This is the number of Web servers, application servers, and database servers respectively in a farm. For example, 2x1x1 means that this environment has 2 Web servers, 1 application server, and 1 database server.
- **MDF and LDF:**SQL Server physical files. For more information, see Files and Filegroups Architecture.

Overview

This section provides an overview to our assumptions and our test methodology.

Assumptions

For our testing, we made the following assumptions:

- In the scope of this testing, we did not consider disk I/O as a limiting factor. It is assumed that an infinite number of spindles are available.
- The tests model only peak time usage on a typical divisional portal. We did not consider cyclical changes in traffic seen with day-night cycles. That also means that timer jobs which generally require scheduled nightly runs are not included in the mix.
- There is no custom code running on the divisional portal deployment in this case. We cannot guarantee behavior of custom code/third-party solutions installed and running in your divisional portal.
- For the purpose of these tests, all of the services databases and the content databases were put on the same instance of Microsoft SQL Server. The usage database was maintained on a separate instance of SQL Server.
- For the purpose of these tests, BLOB cache is enabled.
- Search crawl traffic is not considered in these tests. But to factor in the effects of an ongoing search crawl, we modified definitions of a healthy farm. (Green-zone definition to be 40 percent for SQL Server to allow for 10 percent tax from Search crawls. Similarly, we used 80 percent SQL Server CPU as the criteria for max RPS.)

Test methodology

We used Visual Studio Team System for Test 2008 SP2 to perform the performance testing. The testing goal was to find the performance characteristic of green zone, max zone and various system stages in between for each topology. Detailed definitions of "max zone" and "green zone" are given in the Glossary as measured by specific values for performance counters, but in general, a farm configuration performing around "max zone" breakpoint can be considered under stress, whereas a farm configuration performing "green zone" breakpoint can be considered healthy.

The test approach was to start by using the most basic farm configuration and run a set of tests. The first test is to gradually increase the load on the system and monitor its performance characteristic. From this test we derived the throughput and latency at various user loads and also identified the

system bottleneck. After we had this data, we identified at what user load did the farm exhibit green zone and max zone characteristics. We ran separate tests at those pre-identified constant user loads for a longer time. These tests ensured that the farm configuration can provide constant green zone and max zone performance at respective user loads, over longer period of time.

Later, while doing the tests for the next configuration, we scaled out the system to eliminate bottlenecks identified in previous run. We kept iterating in this manner until we hit SQL Server CPU bottleneck.

We started off with a minimal farm configuration of 1 Web server /application server and 1 database server. Through multiple iterations, we finally ended at 3 Web servers, 1 application server, 1 database server farm configuration, where the database server CPU was maxed out. Below you will find a quick summary and charts of tests we performed on each iteration to establish green zone and max zone for that configuration. That is followed by comparison of green zone and max zone for different iterations, from which we derive our recommendations.

The SharePoint Admin Toolkit team has built a tool that is named "Load Test Toolkit (LTK)" which is publically available for customers to download and use.

Specifications

This section provides detailed information about the hardware, software, topology, and configuration of the lab environment.

Hardware

The table that follows presents hardware specs for the computers that were used in this testing. Every Web server that was added to the server farm during multiple iterations of the test complies with the same specifications.

	Web server	**Application Server**	**Database Server**
Processor(s)	2px4c@2.33GHz	2px4c@2.33GHz	4px4c@ 3.19GHz
RAM	8 GB	8 GB	32 GB
Number of network adapters	2	2	1
Network adapter speed	1 Gigabit	1 gigabit	1 Gigabit
Load balancer type	F5 - Hardware load balancer	Not applicable	Not applicable
ULS Logging level	Medium	Medium	Not applicable

Software

The table that follows explains software installed and running on the servers that were used in this testing effort.

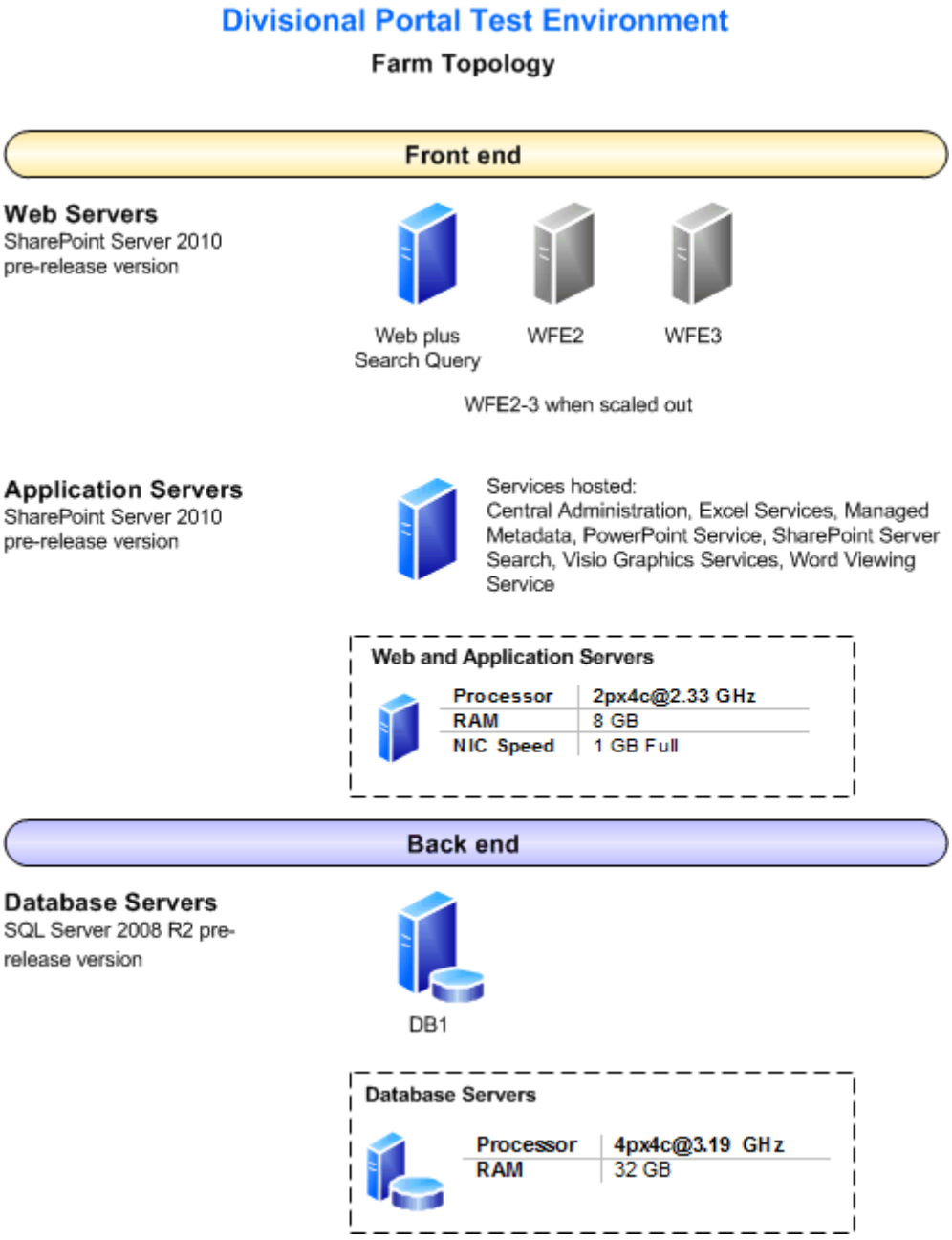
	Web Server	Application Server	Database Server
Operating System	Windows Server 2008 R2 x64	Windows Server 2008 R2 x64	Windows Server 2008 x64
Software version	SharePoint Server 2010 and Office Web Applications, pre-release versions	SharePoint Server 2010 and Office Web Applications, pre-release versions	SQL Server 2008 R2 CTP3
Authentication	Windows NTLM	Windows NTLM	Windows NTLM
Load balancer type	F5 - Hardware load balancer	Not applicable	Not applicable
ULS Logging level	Medium	Medium	Not applicable
Anti-Virus Settings	Disabled	Disabled	Disabled
Services running locally	Microsoft SharePoint Foundation Incoming E-Mail Microsoft SharePoint Foundation Web Application Microsoft SharePoint Foundation Workflow Timer Service Search Query and Site Settings Service SharePoint Server Search	Central Administration Excel Services Managed Metadata Web Service Microsoft SharePoint Foundation Incoming E-Mail Microsoft SharePoint Foundation Web Application Microsoft SharePoint Foundation Workflow Timer Service PowerPoint Services Search Query and Site Settings Service SharePoint Server Search Visio Graphics Services	Not applicable

	Web Server	Application Server	Database Server
		Word Viewing Service	

The table indicates which services are provisioned in the test environment. Other services such as the User Profile service and Web Analytics are not provisioned.

Topology and configuration

The following diagram shows the topology used for the tests. We changed the number of Web servers from 1 to 2 to 3, as we moved between iterations, but otherwise the topology remained the same.



Dataset and disk geometry

The test farm was populated with about 1.62 Terabytes of content, distributed across five different sized content databases. The following table explains this distribution:

Content database	1	2	3	4	5
Content database size	36 GB	135 GB	175 GB	1.2 terabytes	75 GB
Number of sites	44	74	9	9	222
Number of webs	1544	2308	2242	2041	1178
RAID configuration	0	0	0	0	0
Number of spindles for MDF	1	1	5	3	1
Number of spindles for LDF	1	1	1	1	1

Transactional mix

The following are important notes about the transactional mix:

- There are no My Site Web sites provisioned on the divisional portal. Also, the User Profile service, which supports My Site Web sites, is not running on the farm. The transactional mix does not include any My Site page/web service hits or traffic related to Outlook Social Connector.
- The test mix does not include any traffic generated by co-authoring on documents.
- The test mix does not include traffic from Search Crawl. However this was factored into our tests by modifying the Green-zone definition to be 40 percent SQL Server CPU usage instead of the standard 50 percent to allow for 10 percent for the search crawl. Similarly, we used 80 percent SQL Server CPU as the criteria for max RPS.

The following table describes the overall transaction mix. The percentages total 100.

Feature or Service	Operation	Read/write	Percentage of mix
ECM	Get static files	r	8.93%
	View home page	r	1.52%
Microsoft InfoPath	Display/Edit upsize list item and new forms	r	0.32%
	Download file by using "Save as"	r	1.39%
Microsoft OneNote 2010	Open Microsoft Office OneNote 2007 file	r	13.04%

Feature or Service	Operation	Read/write	Percentage of mix
Search	Search through OSSSearch.aspx or SearchCenter	r	4.11%
Workflow	Start autostart workflow	w	0.35%
Microsoft Visio	Render Visio file in PNG/XAML	r	0.90%
Office Web Applications - PowerPoint	Render Microsoft PowerPoint, scroll to 6 slides	r	0.05%
Office Web Applications - Word	Render and scroll Microsoft Word doc in PNG/Silverlight	r	0.24%
Microsoft SharePoint Foundation	List – Check out and then check in an item	w	0.83%
	List - Get list	r	0.83%
	List - Outlook sync	r	1.66%
	List - Get list item changes	r	2.49%
	List - Update list items and adding new items	w	4.34%
	Get view and view collection	r	0.22%
	Get webs	r	1.21%
	Browse to Access denied page	r	0.07%
	View Browse to list feeds	r	0.62%
	Browse to viewlists	r	0.03%
	Browse to default.aspx (home page)	r	1.70%
	Browse to Upload doc to doc lib	w	0.05%
	Browse to List/Library's default view	r	7.16%

Feature or Service	Operation	Read/write	Percentage of mix
	Delete doc in doclib using DAV	w	0.83%
	Get doc from doclib using DAV	r	6.44%
	Lock and Unlock a doc in doclib using DAV	w	3.32%
	Propfind list by using DAV	r	4.16%
	Propfind site by using DAV	r	4.16%
	List document by using FPSE	r	0.91%
	Upload doc by using FPSE	w	0.91%
	Browse to all site content page	r	0.03%
	View RSS feeds of lists or wikis	r	2.03%
Excel Services	Render small/large Excel files	r	1.56%
Workspaces	WXP - Cobalt internal protocol	r	23.00%
	Full file upload using WXP	w	0.57%

Results and analysis

This section describes the test methodology and results to provide guidance for capacity planning of a typical divisional portal.

Results from 1x1 farm configuration

Summary of results

- On a 1 Web server and 1 database server farm, in addition to Web server duties, the same computer was also acting as application server. Clearly this computer (still called Web server) was the bottleneck. As presented in the data here, the Web server CPU reached around 86% utilization when the farm was subjected to user load of 125 users by using the transactional mix described earlier in this document. At that point, the farm exhibited max RPS of 101.37.

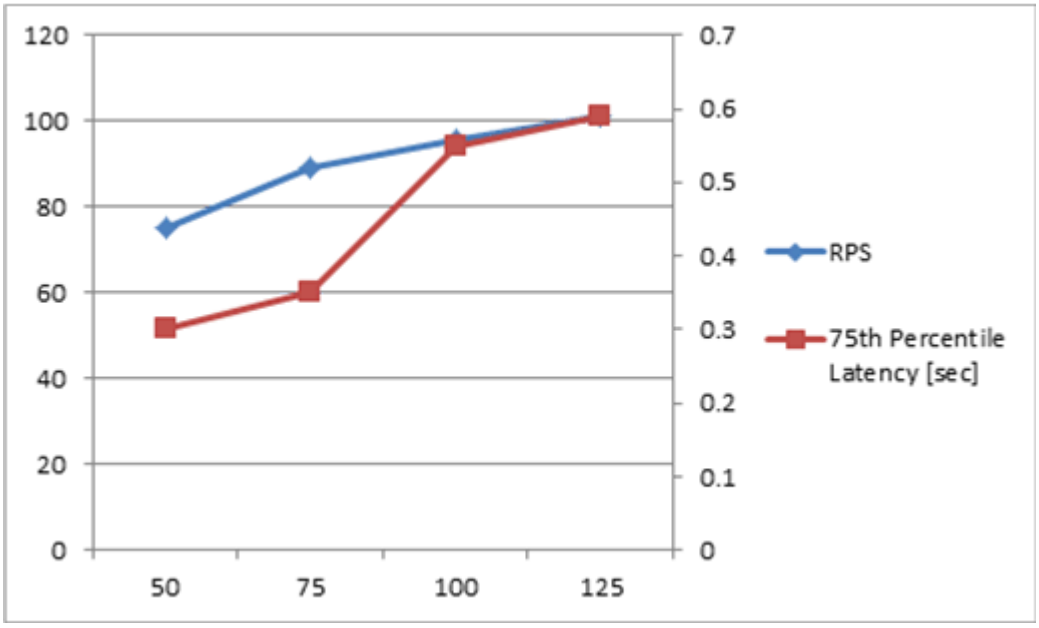
- Even at a small user load, Web server utilization was always too high to consider this farm as a healthy farm. For the workload and dataset that we used for the test, we do not recommend this configuration as a real deployment.
- Going by definition of "green zone", there is not really a "green zone" for this farm. It is always under stress, even at a small load. As for "max zone", at the smallest load, where the farm was in "max zone", the RPS was 75.
- Because the Web server was the bottleneck due to its dual role as an application server, for the next iteration, we separated out the application server role onto its own computer.

Performance counters and graphs

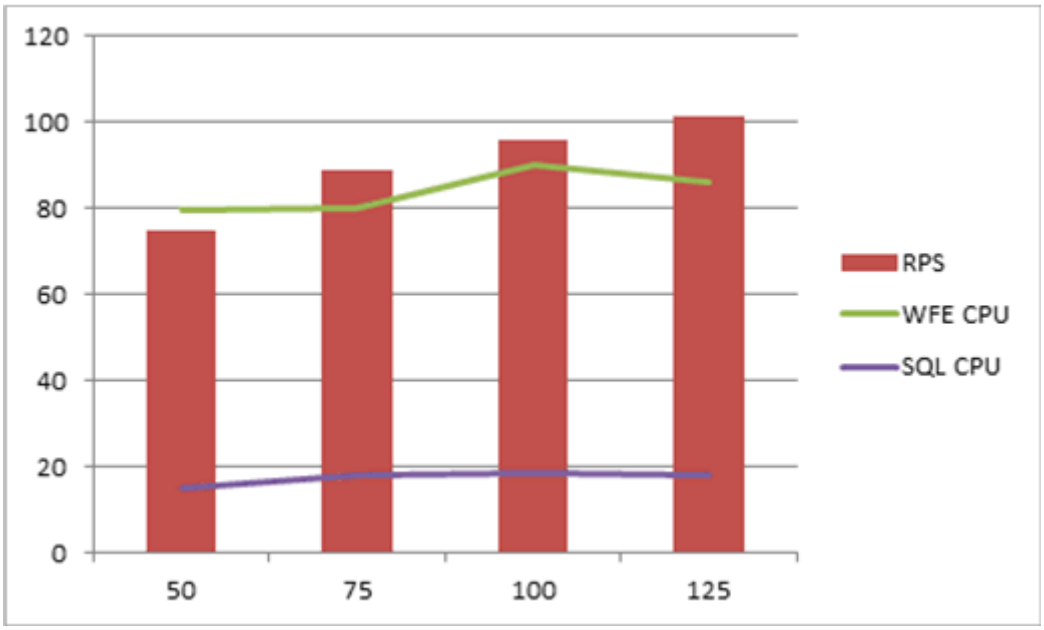
The following table presents various performance counters captured during testing a 1x1 farm at different steps in user load.

User Load	50	75	100	125
RPS	74.958	89.001	95.79	101.37
Latency	0.42	0.66	0.81	0.81
Web server CPU	79.6	80.1	89.9	86
Application server CPU	N/A	N/A	N/A	N/A
Database server CPU	15.1	18.2	18.6	18.1
75th Percentile (sec)	0.3	0.35	0.55	0.59
95th Percentile (sec)	0.71	0.77	1.03	1

The following chart shows the RPS and latency results for a 1x1 configuration.



The following chart shows performance counter data in a 1x1 configuration.



Results from 1x1x1 farm configuration

Summary of results

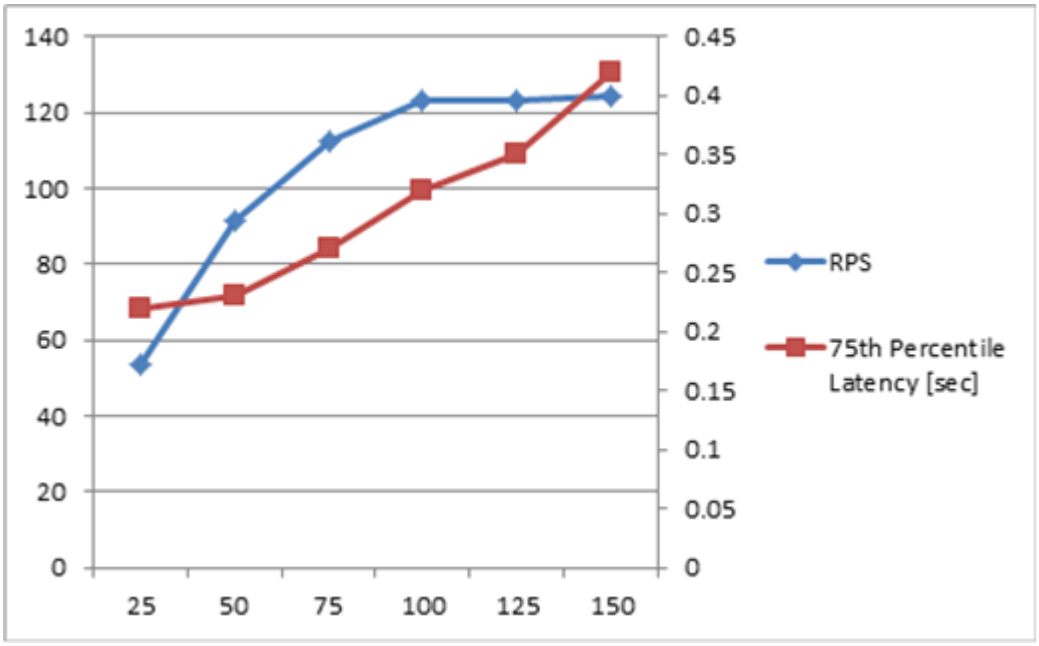
- On a 1 Web server, 1 application server and 1 database server farm, the Web server was the bottleneck. As presented in the data in this section, the Web server CPU reached around 85% utilization when the farm was subjected to user load of 150 users by using the transactional mix described earlier in this document. At that point, the farm exhibited max RPS of 124.1.
- This configuration delivered "green zone" RPS of 99, with 75th percentile latency being 0.23 sec, and the Web server CPU hovering around 56 % utilization. This indicates that this farm can healthily deliver an RPS of around 99. "Max zone" RPS delivered by this farm was 123 with latencies of 0.25 sec and the Web server CPU hovering around 85%.
- Because the Web server CPU was the bottleneck in this iteration, we relived the bottleneck by adding another the Web server for the next iteration.

Performance counters and graphs

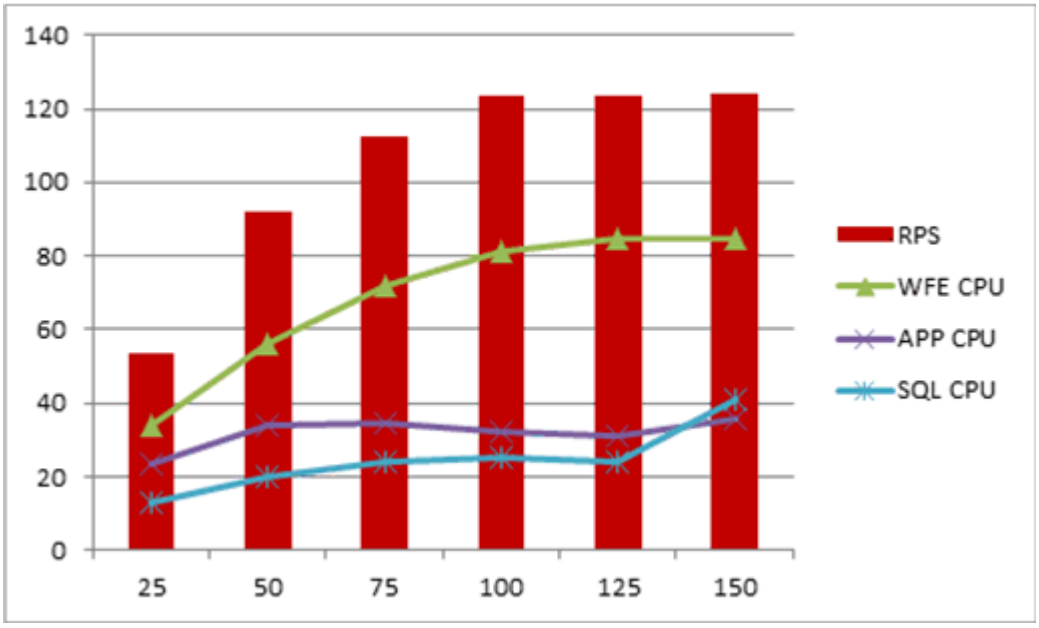
The following table presents various performance counters captured during testing a 1x1x1 farm, at different steps in user load.

User Load	25	50	75	100	125	150
RPS	53.38	91.8	112.2	123.25	123.25	124.1
Latency	34.2	56	71.7	81.5	84.5	84.9
Web server CPU	23.2	33.8	34.4	32	30.9	35.8
Application server CPU	12.9	19.7	24.1	25.2	23.8	40.9
Database server CPU	0.22	0.23	0.27	0.32	0.35	0.42
75th Percentile (sec)	0.54	0.52	0.68	0.71	0.74	0.88

The following chart shows RPS and latency results for a 1x1x1 configuration.



The following chart shows performance counter data in a 1x1x1 configuration.



Results from 2x1x1 farm configuration

Summary of results

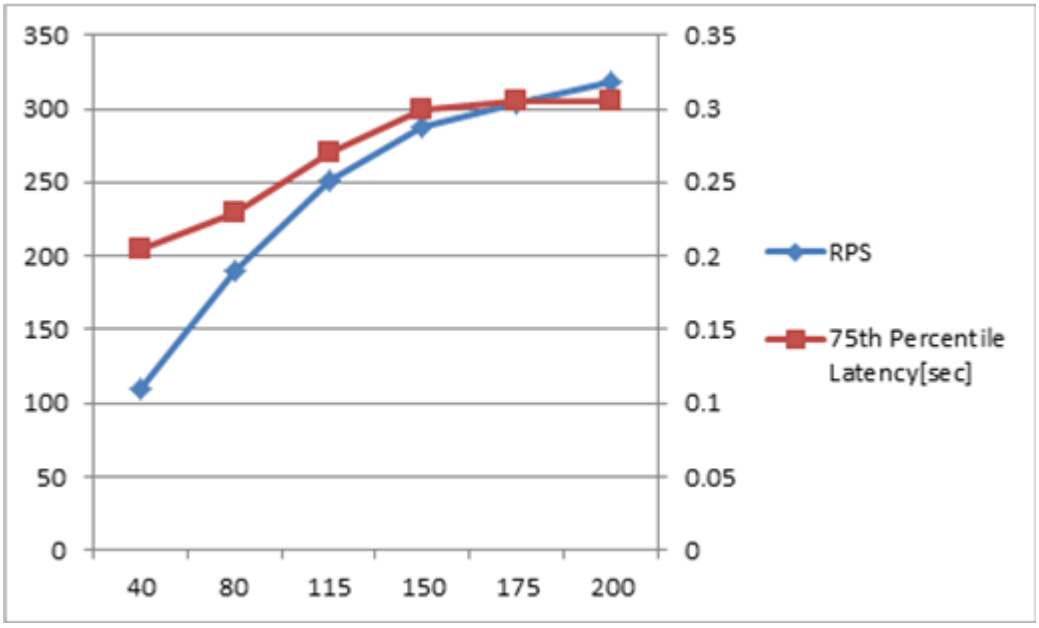
- On a 2 Web server, 1 application server and 1 database server farm, the Web server was the bottleneck. As presented in the data in this section, Web server CPU reached around 76% utilization when the farm was subjected to user load of 200 users by using the transactional mix described earlier in this document. At that point, the farm exhibited max RPS of 318.
- This configuration delivered "green zone" RPS of 191, with 75th percentile latency being 0.37 sec, and Web server CPU hovering around 47 % utilization. This indicates that this farm can healthily deliver an RPS of around 191. "Max zone" RPS delivered by this farm was 291 with latencies of 0.5 sec and Web server CPU hovering around 75%.
- Because the Web server CPU was the bottleneck in this iteration, we relived the bottleneck by adding another Web server for the next iteration.

Performance counters and graphs

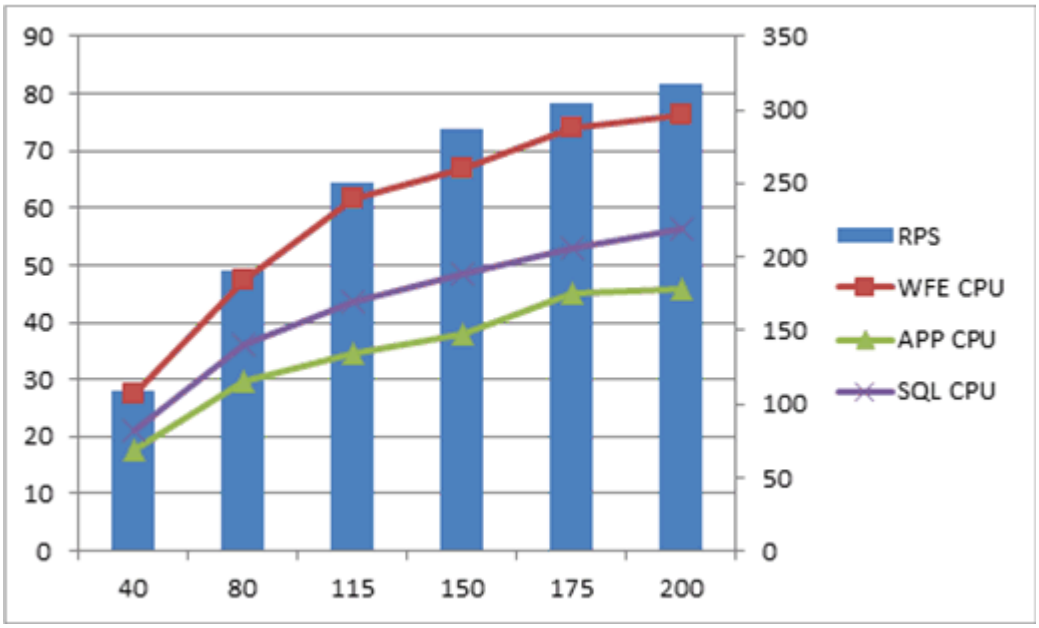
The following table presents various performance counters captured during testing a 2x1x1 farm, at different steps in user load.

User Load	40	80	115	150	175	200
RPS	109	190	251	287	304	318
Latency	0.32	0.37	0.42	0.49	0.54	0.59
Web server CPU	27.5	47.3	61.5	66.9	73.8	76.2
Application server CPU	17.6	29.7	34.7	38	45	45.9
Database server CPU	21.2	36.1	43.7	48.5	52.8	56.2
75th Percentile (sec)	0.205	0.23	0.27	0.3	0.305	0.305
95th Percentile (sec)	0.535	0.57	0.625	0.745	0.645	0.57

The following chart shows RPS and latency results for a 2x1x1 configuration.



The following chart shows performance counter data in a 2x1x1 configuration.



Results from 3x1x1 farm configuration

Summary of results

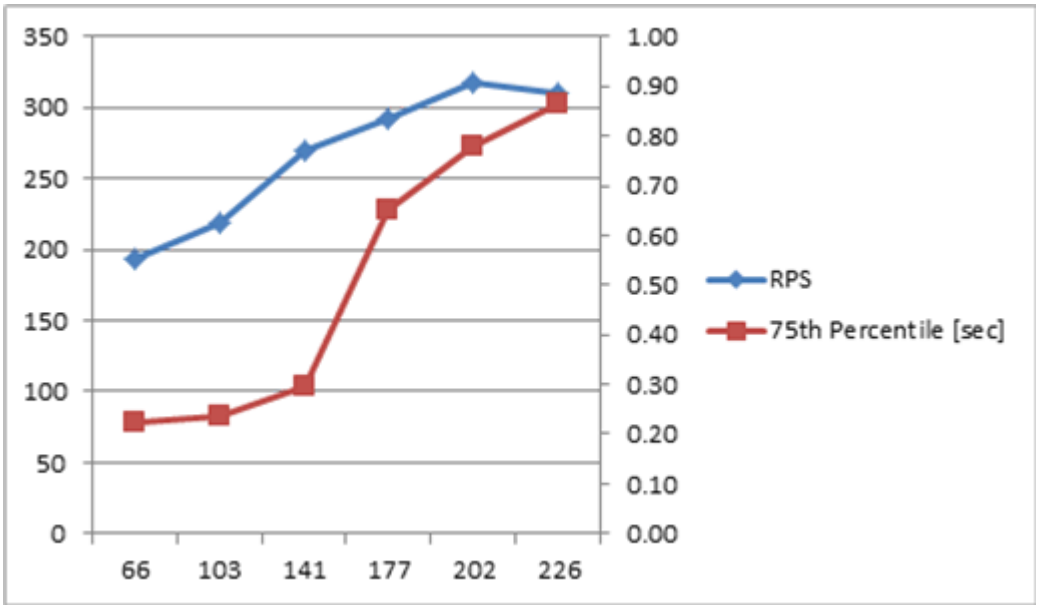
- On a 3 Web server, 1 application server and 1 database server farm, finally, the database server CPU was the bottleneck. As presented in the data in this section, database server CPU reached around 76% utilization when the farm was subjected to user load of 226 users by using the transactional mix described earlier in this document. At that point, the farm exhibited max RPS of 310.
- This configuration delivered "green zone" RPS of 242, with 75th percentile latency being 0.41 sec, and database server CPU hovering around 44% utilization. This indicates that this farm can healthily deliver an RPS of around 242. "Max zone" RPS delivered by this farm was 318 with latencies of 0.5 sec and database server CPU hovering around 75%.
- This was the last configuration in the series.

Performance counters and graphs

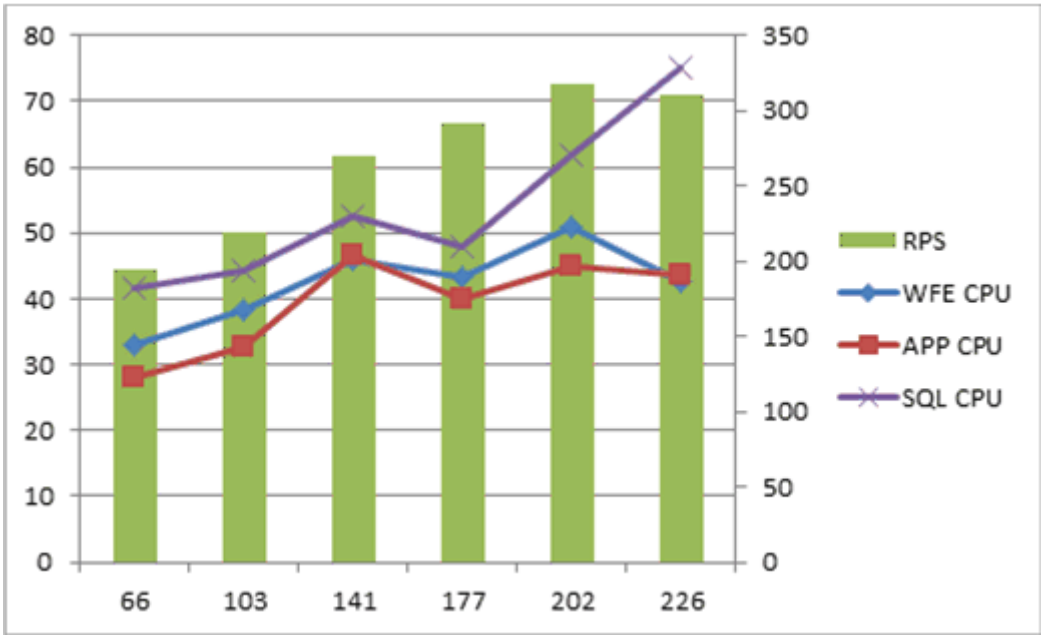
The following table presents various performance counters captured during testing a 3x1x1 farm, at different steps in user load.

User Load	66	103	141	17	202	226
RPS	193.8	218.5	269.8	275.5	318.25	310
Latency	0.3	0.41	0.47	0.58	0.54	0.78
Web server CPU	33	38.3	45.8	43.3	51	42.5
Application server CPU	28	32.6	46.5	40	45.1	43.7
Database server CPU	41.6	44.2	52.6	48	61.8	75
75th Percentile (sec)	0.22	0.24	0.30	0.65	0.78	0.87
95th Percentile (sec)	0.49	0.57	0.72	1.49	0.51	1.43

The following chart shows RPS and latency results in a 3x1x1 configuration.



The following chart shows performance counter data for a 3x1x1 configuration.



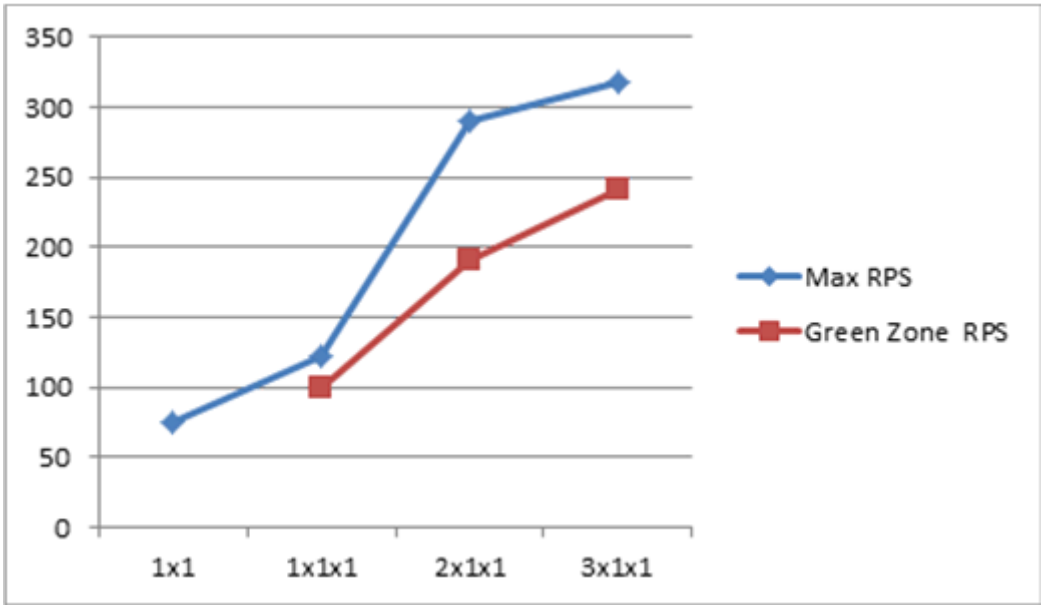
Comparison

From the iterative tests we performed, we found out the points at which a configuration enters max zone or green zone. Here's a table of those points.

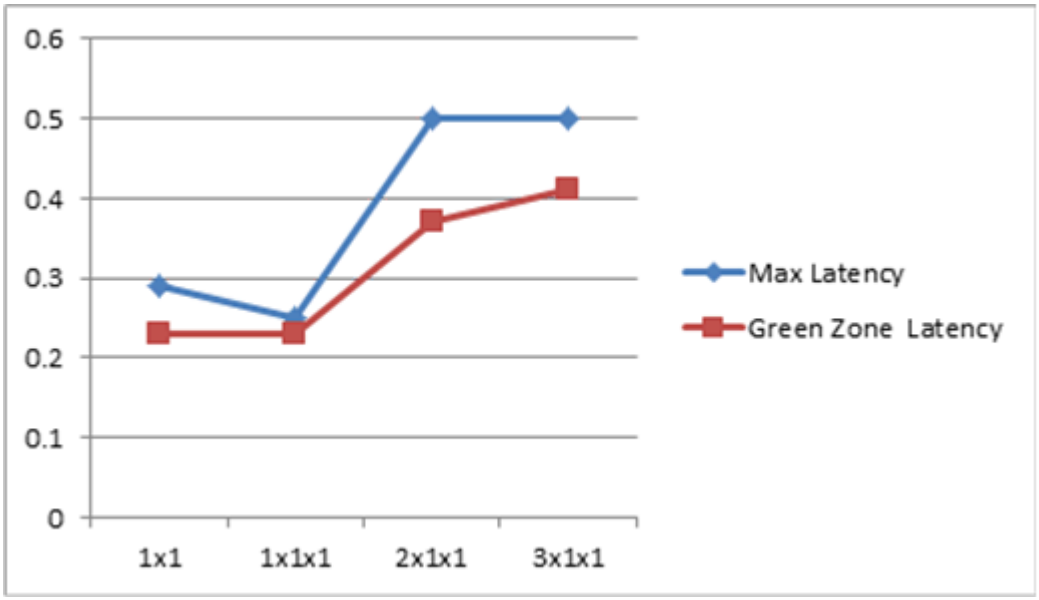
The table and charts in this section provide a summary for all the results that were presented earlier in this article.

Topology	1x1	1x1x1	2x1x1	3x1x1
Max RPS	75	123	291	318
Green Zone RPS	Not applicable	99	191	242
Max Latency	0.29	0.25	0.5	0.5
Green Zone Latency	0.23	0.23	0.37	0.41

The following chart shows a summary of RPS at different configurations.



The following chart shows a summary of latency at different configurations.



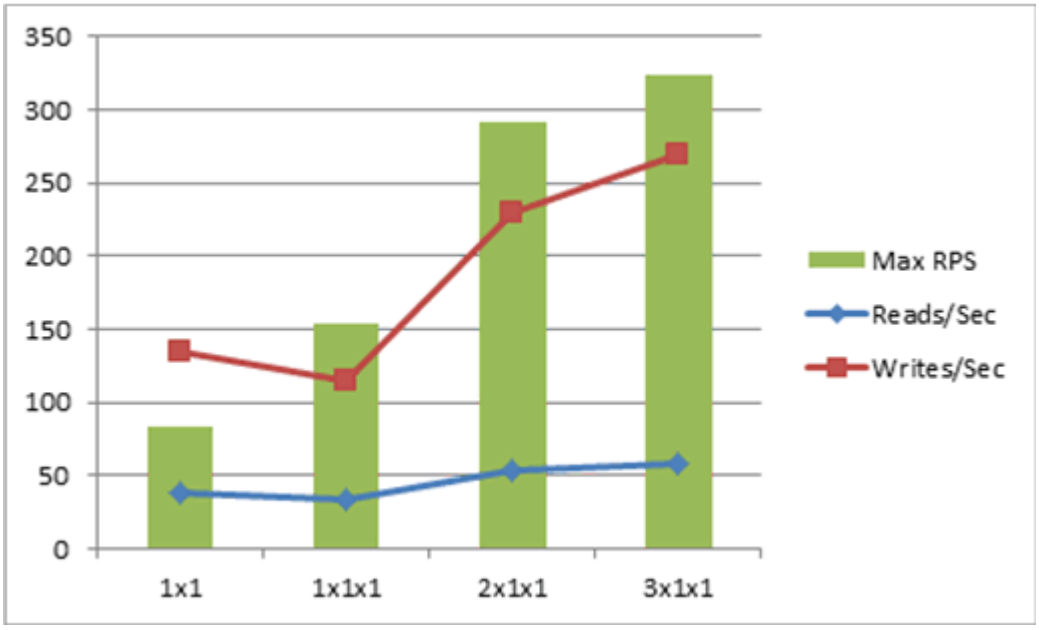
A note on disk I/O

Disk I/O based bottlenecks are not considered while prescribing recommendations in this document. However, it is still interesting to observe the trend. Here are the numbers:

Configuration	1x1	1x1x1	2x1x1	3x1x1
Max RPS	75	154	291	318
Reads/Sec	38	34	54	58
Writes/Sec	135	115	230	270

Because we ran the tests in durations of 1 hour and the test uses only a fixed set of sites/webs/document libraries and so on, SQL Server could cache all the data. Thus, our testing caused very little Read IO. We see more write I/O operations that read. It is important to be aware that this is an artifact of the test methodology, and not a good representation of real deployments. Most of the typical divisional portals would have more read operations 3 to 4 times more than write operations.

The following chart shows I/Ops at different RPS.



Tests with Search incremental crawl

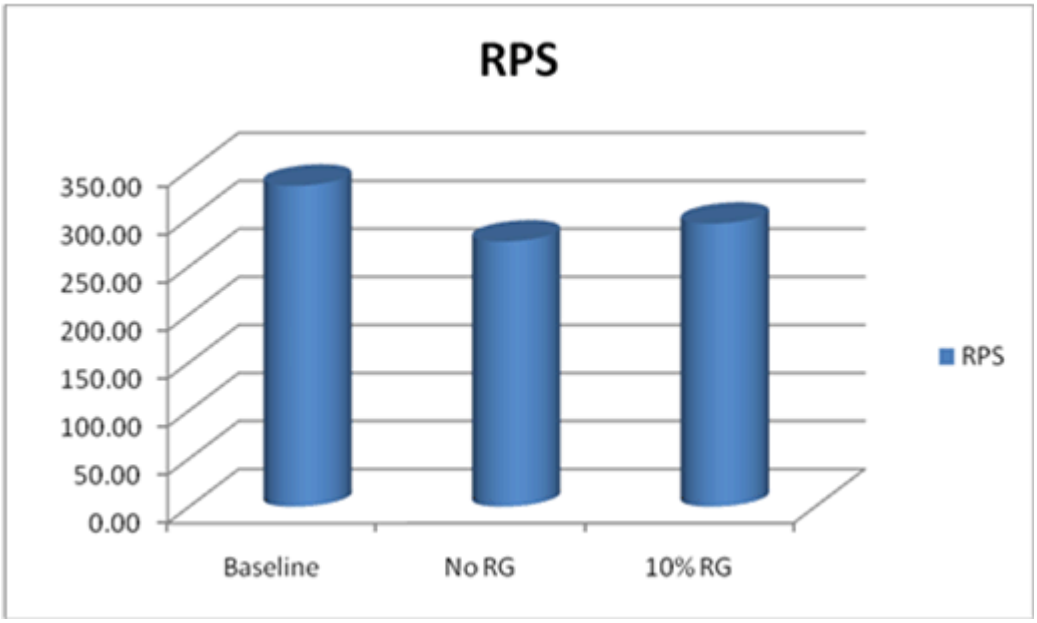
As we mentioned before, all the tests until now were run without Search crawl traffic. In order to provide information about how ongoing search crawl can affect performance of a farm, we decided to find out the max user RPS and corresponding user latencies with search crawl traffic in the mix. We added a separate Web server to 3x1x1 farm, designated as a crawl target. We saw a 17% drop in RPS compared to original RPS exhibited by 3x1x1.

In a separate test, on the same farm, we used Resource Governor to limit available resources to search crawl 10%. We saw that as Search uses lesser resources, the max RPS of the farm climbs up by 6%.

	Baseline 3x1x1	Only Incremental Crawl	No Resource Governor	10% Resource Governor
RPS	318	N/A	276	294.5
Percent RPS difference from baseline	0%	N/A	83%	88%
Database server CPU (%)	83.40	8.00	86.60	87.3
SA Database server CPU (%)	3.16	2.13	3.88	4.2

	Baseline 3x1x1	Only Incremental Crawl	No Resource Governor	10% Resource Governor
Web server CPU (%)	53.40	0.30	47.00	46.5
Application server CPU (%)	22.10	28.60	48.00	41.3
Crawl Web server CPU (%)	0.50	16.50	15.00	12.1

The following chart shows results from tests with incremental Search crawl turned on.



Important:
Here we are only talking about incremental crawl, on a farm where there are not very many changes to the content. It is important to be aware that 10% resource utilization will be insufficient for a full search crawl. It may also prove to be less if there are too many changes. It is definitely not advised to limit resource utilization to 10% if you are running a full search crawl, or your farm generally sees a high volume of content changes between crawls.

Summary of results and recommendations

To paraphrase the results from all configurations we tested:

- With the configuration, dataset and test workload we had selected for the tests, we could scale out to maximum 3 Web servers before SQL Server was bottlenecked on CPU. The absolute max RPS we could reach that point was somewhere around 318.
- With each additional Web server, increase in RPS was almost linear. We can extrapolate that as long as SQL Server is not bottlenecked, you can add more Web servers and additional increase in RPS is possible.
- Latencies are not affected much as we approached bottleneck on SQL Server.
- Incremental Search crawl directly affects RPS offered by a configuration. The effect can be minimized by using Resource Governor.

Using the results, here are few recommendations on how to achieve even larger scale if you must have more RPS from your divisional portal:

- 1x1 farm can deliver up to 75 RPS. However, it is usually stressed. It's not a recommended configuration for a divisional portal in production.
- Separate out content databases and services databases on separate instances of SQL Server: In the test workload used in tests, when SQL Server was bottlenecked on CPU, only 3% of the traffic was to the services databases. Thus this step would have achieved slightly better scale out than what we saw. But, in general, increase in scale out by separating out content databases and services databases is directly proportional to the traffic to services database in your farm.
- Separate out individual content databases on separate instances of SQL Server. In the dataset used for testing, we had 5 content databases, all located on the same instance of SQL Server. By separating them out on different computers, you will be spreading CPU utilization across multiple computers. Therefore, you will see much larger RPS numbers.
- Finally when SQL Server is bottlenecked on CPU, adding more CPU to SQL Server can increase RPS potential of the farm almost linearly.

How to translate these results into your deployment

In this article, we discussed results as measured by RPS and latency, but how do you apply these in the real world? Here's some math based on our experience from divisional portal internal to Microsoft.

A divisional portal in Microsoft which supports around 8000 employees collaborating heavily, experiences an average RPS of 110. That gives a Users to RPS ratio of ~72 (that is, 8000/110). Using the ratio, and the results discussed earlier in this article, we can estimate how many users a particular farm configuration can support healthily:

Farm configuration	"Green Zone" RPS	Approximate number of users it can support
1x1x1	99	7128
2x1x1	191	13452
3x1x1	242	17424

Of course, this is only directly applicable if your transactional mix and hardware is exactly same as the one used for these tests. Your divisional portal may have different usage pattern. Therefore, the ratio may not directly apply. However, we expect it to be approximately applicable.

About the authors

Gaurav Doshi is a Program Manager for SharePoint Server at Microsoft.

Raj Dhrolia is a Software Test Engineer for SharePoint Server at Microsoft.

Wayne Roseberry is a Principal Test Lead for SharePoint Server at Microsoft.

Microsoft SharePoint Server 2010 social environment: Technical case study

This document describes a specific deployment of Microsoft SharePoint Server 2010. It includes the following:

- Technical case study environment specifications, such as hardware, farm topology and configuration
- The workload that includes the number, and types, of users or clients, and environment usage characteristics
- Technical case study farm dataset that includes database contents and Search indexes
- Health and performance data that is specific to the environment

In this article:

Prerequisite information

Before reading this document, make sure that you understand the key concepts behind SharePoint Server 2010 capacity management. The following documentation will help you learn about the recommended approach to capacity management and provide context for helping you understand how to make effective use of the information in this document, and also define the terms used throughout this document.

For more conceptual information about performance and capacity that you might find valuable in understanding the context of the data in this technical case study, see the following documents:

- Capacity management and sizing overview for SharePoint Server 2010
- SharePoint Server 2010 capacity management: Software boundaries and limits

Introduction to this environment

This white paper describes an actual SharePoint Server 2010 environment at Microsoft. Use this document to compare with your planned workload and usage characteristics. If your planned design is similar, you can use the deployment described here as a starting point for your own installation.

This document includes the following:

- **Specifications**, which include hardware, topology and configuration
- **Workload**, which is the demand on the farm that includes the number of users, and the usage characteristics
- **Dataset** that includes database sizes
- **Health and performance data** specific to the environment

This document is part of a series of Performance and capacity technical case studies (SharePoint Server 2010) about SharePoint environments at Microsoft.

The SharePoint Server 2010 environment described in this document is a production environment at a large, geographically distributed company. This environment hosts SharePoint My Sites that connect employees with one another and the information that they need. Employees use this environment to present personal information such as areas of expertise, past projects, and colleagues to the wider organization. The environment also hosts personal sites and documents for viewing, editing, and collaboration. My Site Web sites are integrated with Active Directory Domain Services (AD DS) to provide a central location that can be accessed from the browser and various client applications.

As many as 72,000 unique users visit the environment on a busy day, generating up to 180 requests per second (RPS) during peak hours. Because this is an intranet site, all users are authenticated.

The information that is provided in this document reflects the enterprise social environment on a typical day.

Specifications

This section provides detailed information about the hardware, software, topology, and configuration of the case-study environment.

Hardware

This section provides details about the server computers that were used in this environment.

Note

- This environment is scaled to accommodate pre-release builds of SharePoint Server 2010 and other products. Hence, the hardware deployed has larger capacity than necessary to serve the demand typically experienced by this environment. This hardware is described only to provide additional context for this environment and serve as a starting point for similar environments.
- It is important to conduct your own capacity management based on your planned workload and usage characteristics. For more information about the capacity management process, see Capacity management and sizing overview for SharePoint Server 2010.

Web Servers

There are three Web servers in the farm, each with identical hardware. Two serve content, and the third is a dedicated search crawl target.

Web Server	WFE1-3
Processor(s)	2 quad core @ 2.33 GHz
RAM	16 GB
Operating system	Windows Server 2008, 64 bit
Size of the SharePoint drive	400 GB
Number of network adapters	2
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Load balancer type	Hardware load balancing
Software version	SharePoint Server 2010 (pre-release version)
Services running locally	Central Administration Microsoft SharePoint Foundation Incoming E-Mail Microsoft SharePoint Foundation Web Application Microsoft SharePoint Foundation Workflow Timer Service Search Query and Site Settings Service SharePoint Server Search
Services consumed from a federated services farm	User Profile Service Web Analytics Web Service Business Data Connectivity Service Managed Metadata Web Service

Application Server

There are two application servers in the farm, each with identical hardware.

Application Server	APP1-4
Processor(s)	2 quad core @ 2.33 GHz
RAM	16 GB
OS	Windows Server 2008, 64 bit
Size of the SharePoint drive	400 GB
Number of network adapters	1
Network adapter speed	1 Gigabit
Authentication	Windows NTLM
Load balancer type	Hardware load balancing
Software version	SharePoint Server 2010 (pre-release version)
Services running locally	Office Web Apps Excel PowerPoint Secure Store Usage and Health State Service

Database Servers

There is a SQL cluster with two database servers, each with identical hardware. One of the servers is active and the other is passive for redundancy.

Database Server	DB1-2
Processor(s)	4 quad core @ 2.4 GHz
RAM	64 GB
Operating system	Windows Server 2008, 64 bit

Database Server	DB1-2
Storage and geometry	(1.25 TB * 6) Disk 1-4: SQL Data Disk 5: Logs Disk 6: TempDB
Number of network adapters	2
Network adapter speed	1 @ 100MB, 1 @ 1GB
Authentication	Windows NTLM
Software version	SQL Server 2008

Topology

The following diagram shows the topology for this farm.

Enterprise Social Environment

Farm Topology

Front end

Web Servers
SharePoint Server 2010
pre-release version

Web plus Central Administration | Web plus Central Administration | Search Crawl

Application Servers
SharePoint Server 2010
pre-release version

Services hosted:
Office Web Apps, Excel, Secure Store, Usage, PowerPoint, State Service

Web and Application Servers

Processor	**2px4c@2.33 GHz**
RAM	16 GB
Storage	400 GB
NIC Speed	1 GB Full

Back end

Database Servers
SQL Server 2008

SQL Cluster

Database Servers

Processor	**4px6c@2.4 GHz**
RAM	64 GB
Storage	1.2 TB *6

Configuration

The following table enumerates settings that were changed that affect performance or capacity in the environment.

Setting	Value	Notes
Usage Service: Trace Log – days to store log files (default: 14 days)	5 days	The default is 14 days. Lowering this setting can save disk space on the server where the log files are stored.
QueryLoggingThreshold Microsoft SharePoint Foundation Database – configure QueryLoggingThreshold to 1 second	1 second	The default is 5 seconds. Lowering this setting can save bandwidth and CPU on the database server.
Database Server – Default Instance Max degree of parallelism	1	The default is 0. To ensure optimal performance, we strongly recommend that you set max degree of parallelism to 1 for database servers that host SharePoint Server 2010 databases. For more information about how to set max degree of parallelism, see max degree of parallelism Option(http://go.microsoft.com/fwlink/?LinkId=189030).

Workload

This section describes the workload, which is the demand on the farm that includes the number of users, and the usage characteristics.

Workload Characteristics	Value
Average Requests per Second (RPS)	64
Average RPS at peak time (11 AM-3 PM)	112
Total number of unique users per day	69,814
Average concurrent users	639
Maximum concurrent users	1186
Total # of requests per day	4,045,677

This table shows the number of requests for each user agent.

User Agent	Requests	Percentage of Total
Outlook Social Connector Browser	1,808,963	44.71%
Search (crawl)	704,569	17.42%
DAV	459,491	11.36%
OneNote	266,68	6.59%
Outlook	372,574	9.21%
Browser	85,913	2.12%
Word	38,556	0.95%
Excel	30,021	0.74%
Office Web Applications	20,314	0.50%
SharePoint Workspaces	19,017	0.47%

Dataset

This section describes the case study farm dataset that includes database sizes and Search indexes.

Dataset Characteristics	Value
Database size (combined)	1.5 TB
BLOB size	1.05 TB
Number of content databases	64
Number of Web applications	1
Number of site collections	87,264
Number of sites	119,400
Search index size (number of items)	5.5 million

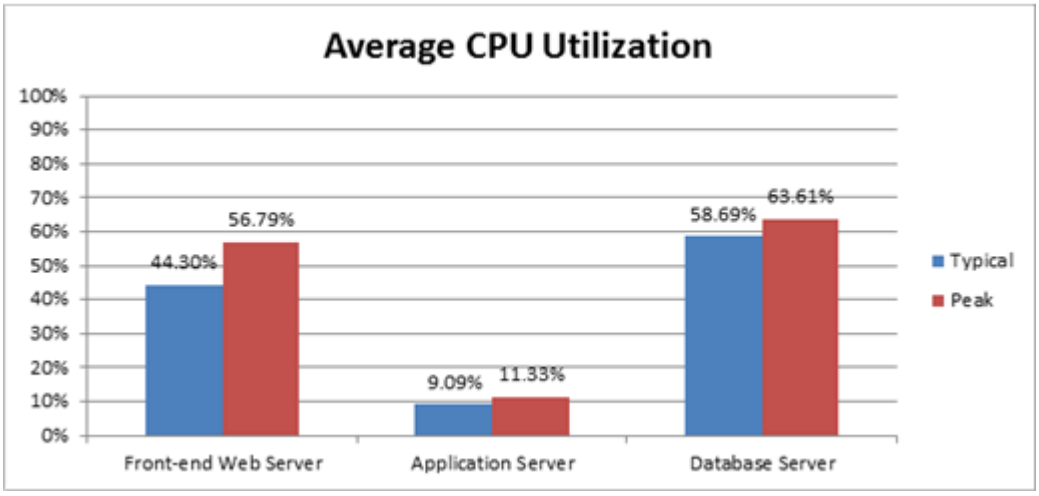
Health and Performance Data

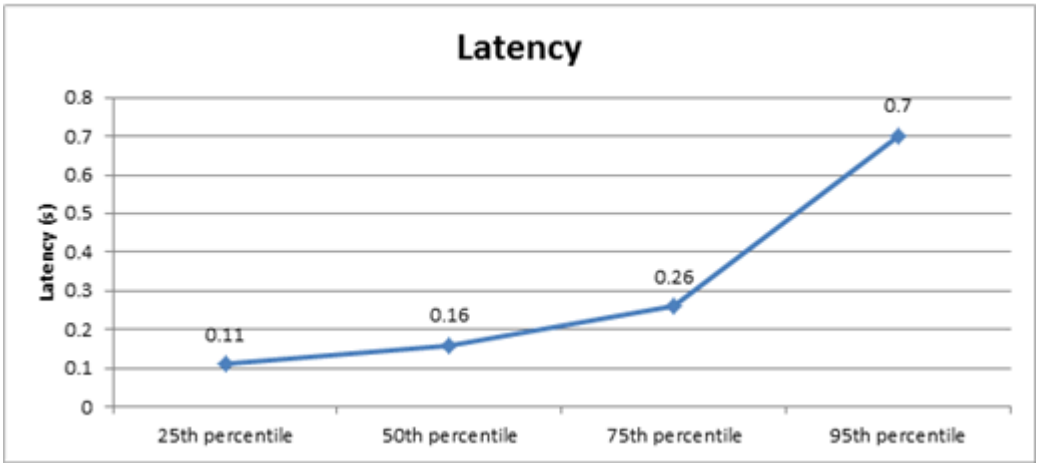
This section provides health and performance data that is specific to the case study environment.

General Counters

Metric	Value
Availability (uptime)	99.61%
Failure Rate	0.39%
Average memory used	0.79 GB
Maximum memory used	4.53 GB
Search Crawl % of Traffic (Search client requests / total requests)	17.42%

The following charts show average CPU utilization and latency for this environment.





In this document, latency is divided into four categories. The 50th percentile latency is typically used to measure the server's responsiveness. It means that half of the requests are served within that response time. The 95th percentile latency is typically used to measure spikes in server response times. It means that 95% of requests are served within that response time, and therefore, 5% of the requests experience slower response times.

Database Counters

Metric	Value
Read/Write Ratio (IO Per Database)	99.854 : 0.146
Average Disk queue length	8.702
Disk Queue Length: Reads	30.518
Disk Queue Length: Writes	4.277
Disk Reads/sec	760.886
Disk Writes/sec	180.644
SQL Compilations/second	3.129
SQL Re-compilations/second	0.032
SQL Locks: Average Wait Time	125 ms
SQL Locks: Lock Wait Time	33.322 ms
SQL Locks: Deadlocks Per Second	0
SQL Latches: Average Wait Time	0 ms

Metric	Value
SQL Cache Hit Ratio	20.1%

Performance and capacity test results and recommendations (SharePoint Server 2010)

This section contains a series of white papers and articles that describe the performance and capacity impact of specific feature sets included in Microsoft SharePoint Server 2010. These white papers and articles include information about the performance and capacity characteristics of the feature and how it was tested by Microsoft, including:

- Test farm characteristics
- Test results
- Recommendations
- Troubleshooting performance and scalability

Note:
The white papers are being updated and republished as articles. If you download a white paper from this page, there may be updated information available when it is republished as an article.

Before reading these white papers and articles, it is important that you understand the key concepts behind capacity management in SharePoint Server 2010. For more information, see Capacity management and sizing for SharePoint Server 2010.

The following table describes the available white papers and articles. If the content is available as a white paper, it is available as a Microsoft Word document from SharePoint Server 2010 performance and capacity test results and recommendations.

Subject	Description
Access Services	Provides guidance on how using Access Services impacts topologies running SharePoint Server 2010. View the article at Estimate performance and capacity requirements for Access Services in SharePoint Server 2010.
Business Connectivity Services	Provides guidance on the footprint that use of Business Connectivity Services has on topologies running SharePoint Server 2010. Download this white paper (BCSCapacityPlanningDoc.docx).
Caches overview	Provides information about how the three SharePoint Server 2010 caches help the product scale and grow to meet the demands of your business application. Download this white paper (SharePointServerCachesPerformance.docx).

Subject	Description
Excel Services in Microsoft SharePoint Server 2010	Provides planning guidance for Excel Services in Microsoft SharePoint Server 2010. View the article at Estimate performance and capacity requirements for Excel Services in SharePoint Server 2010.
InfoPath Forms Services	Provides guidance on the footprint that use of InfoPath Forms Services has on topologies running SharePoint Server 2010. Download this white paper (InfoPath2010CapacityPlanningDoc.docx).
Large lists	Provides guidance on performance of large document libraries and lists. This document is specific to SharePoint Server 2010, although the throttles and limits that are discussed also apply to Microsoft SharePoint Foundation 2010. Download this white paper (DesigningLargeListsMaximizingListPerformance.docx).
Large-scale document repositories	Provides guidance on performance of large-scale document repositories in regards to SharePoint Server 2010. Download this white paper (LargeScaleDocRepositoryCapacityPlanningDoc.docx).
My Sites and social computing	Provides guidance on the footprint that use of My Sites and other social computing features has on topologies running SharePoint Server 2010. Download this white paper (MySitesSocialComputingCapacityPlanningDoc.docx).
Office Web Apps	Provides guidance on the footprint that use of Office Web Apps has on topologies running SharePoint Server 2010. Download this white paper (OfficeWebAppsCapacityPlanningDoc.docx).
PerformancePoint Services	Provides guidance on the footprint that usage of PerformancePoint Services has on topologies running SharePoint Server 2010. View the article at Estimate performance and capacity requirements for PerformancePoint Services.
Search	Provides capacity planning information for different deployments of Search in SharePoint Server 2010, including small, medium, and large farms. Download this white paper

Subject	Description
	(SearchforSPServer2010CapacityPlanningDoc.docx). If you are using FAST Search Server 2010 for SharePoint as your enterprise search solution, see Plan for performance and capacity (FAST Search Server 2010 for SharePoint).
Visio Services	Provides guidance on the footprint that use of Visio Services has on topologies running SharePoint Server 2010. Download this white paper (VisioServicesCapacityPlanningDoc.docx).
Web Analytics	Provides guidance on the footprint that use of the Web Analytics service has on topologies running SharePoint Server 2010. View the articles at Capacity requirements for Web Analytics Shared Service in SharePoint Server 2010.
Web Content Management	Provides guidance on performance and capacity planning for a Web Content Management solution. View the article at Estimate performance and capacity requirements for Web Content Management in SharePoint Server 2010.
Word Automation Services	Provides capacity planning guidance for Word Automation Services in SharePoint Server 2010. Download this white paper (WASCapacityPlanningDoc.docx).
Workflow	Provides guidance on the footprint that usage of Workflow has on topologies running SharePoint Server 2010. Download this white paper (WorkflowCapacityPlanningDoc.docx).

Estimate performance and capacity requirements for Access Services in SharePoint Server 2010

This article provides guidance on the footprint that usage of Access Services in Microsoft SharePoint Server 2010 has on topologies that are running Microsoft SharePoint Server 2010.

In this article:

Test farm characteristics

This section describes the dataset that was used during the testing; the workloads that were placed on the product during performance gathering; the hardware that was used during the testing; and the topology for how that hardware was deployed.

Dataset

Access Services capacity and performance is highly dependent on the makeup of the applications that are hosted on the service. The size of tables and the complexity of queries often have the most effect on capacity and performance. The testing used representative sizes and complexities, but every application and dataset is different. The capacity and performance will depend on the applications that are being used, their specific complexity, and the data size.

To evaluate the capacity profile, Access Services applications were simulated on a farm dedicated to Access Services (no other SharePoint tests were running). The farm contained the following representative sites:

- 1,500 Access Services applications that have a “Small” size profile; 100 items maximum per list.
- 1,500 Access Services applications that have a “Medium” size profile; 2,000 items maximum per list.
- 1,500 Access Services applications that have a “Large” size profile; 10,000 items maximum per list.

Each application is made up of multiple lists, and the other lists are appropriately sized based on this largest list. Access Services can handle more data than 10,000 items. This number for the “Large” profile was chosen because it was expected that larger applications would not be common.

The applications were evenly distributed among the following applications:

- **Contacts** A simple contact management application, dominated by a single list.

- **Projects** A simple task and project tracking applications, dominated by two lists (projects and tasks associated with each project).
- **Orders** A simple order entry system, similar to the Northwind Traders sample of Microsoft Access, but scaled down, and included many interrelated lists (orders, order details, invoices, invoice details, purchase orders, purchase order details, and so on).

Workload

To simulate application usage, workloads were created to perform one or more of the following operations:

- Opening forms
- Paging through the forms
- Filtering and sorting data sheets
- Updating, deleting and inserting records
- Publishing application
- Render reports

Each workload includes “think time” between user actions, ranging from 5 to 20 seconds. This differs from other SharePoint capacity planning documents. Access Services is stateful; memory cursors and record sets were maintained between user interactions. It was important to simulate a full user session and not merely individual requests. For a single user workload, there is an average of two requests per second.

The following table shows the percentages used to determine which application and which size of application to use.

	Small	**Medium**	**Large**
Contacts	16%	10%	9%
Projects	18%	12%	10%
Orders	11%	8%	6%

Green and red zone definitions

For each configuration, two tests were ran to determine a “green zone” and a “red zone.” The green zone is the recommended throughput that can be sustained. The red zone is the maximum throughput that can be tolerated for a short time, but should be avoided.

The green zone was defined as a point at which the test being run consumes at most half the bottlenecking resource. In this case, the bottlenecking resource was %CPU on any of the three tiers: front-end Web server, application server (Access Data Services), or database server (SQL Server).

First, the bottleneck was identified for a particular configuration. If the bottleneck was Access Data Services CPU, we made sure that the green zone test consumed CPU on the Access Data Services computers in a range between 40 and 50 percent.

For the red zone, a point was selected at which the maximum throughput was reached. This proved to be a CPU range between 80 and 90 percent. When searching for bottleneck, we looked at %CPU, memory usage (private bytes), disk queue length, network I/O, and other resources that could result in a bottleneck.

Both the green and red zone tests were run for 1 hour at a fixed user load.

Your results might vary

The specific capacity and performance figures presented in this article will differ from figures in a real-world environment. This simulation is only an estimate of what actual users might do. The figures presented are intended to provide a starting point for the design of an appropriately scaled environment. After you have completed the initial system design, you should test the configuration to determine whether the system will support the factors in your environment.

Hardware setting and topology

Lab Hardware

To provide a high level of test-result detail, several farm configurations were used for testing. Farm configurations ranged from one to four front-end Web servers, one to four application servers (if there is Access Services or Access Data Services), and a single database server computer that is running Microsoft SQL Server 2008. In addition, testing was performed by using four client computers. All server computers were 64-bit. All client computers were 32-bit.

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

Machine role	CPU	Memory	Network	Disk
Front-end Web server	2 processor, 4 core 2.33 GHz	8 GB	1 gig	2 spindles RAID 5
Application server (Access Data Services)	2 processor, 4 core 2.33 GHz	8 GB	1 gig	2 spindles RAID 5
Database server (SQL Server)	4 processor, 4 core 2.6 GHz	32GB	1 gig	Direct Attached Storage (DAS) attached RAID 0 for each Logical Unit Number

Machine role	CPU	Memory	Network	Disk
				(LUN)

Topology

From our experience, CPU on the application sever tier, where Access Data Services is running, is an important limiting factor for throughput. We varied our topology by adding additional Access Data Services computers until it was no longer the bottleneck, and then added a front-end Web server to obtain even more throughput.

- **1x1:** One front-end Web server computer to one Access Data Services computer
- **1x2:** One front-end Web server computer to two Access Data Services computers
- **1x3:** One front-end Web server computer to three Access Data Services computers
- **1x4:** One front-end Web server computer to four Access Data Services computers
- **2x1:** Two front-end Web server computers to one Access Data Services computer
- **2x2:** Two front-end Web server computers to two Access Data Services computers
- **2x4:** Two front-end Web server computers to four Access Data Services computers

The computer running SQL Server is a relatively strong computer and at no time did it become the bottleneck (although it started to approach CPU saturation on our 2x4 test), so we did not vary this in our topologies. Depending on the queries that are a part of a real-world application mix, it is expected that the database server (SQL Server) tier could become the bottleneck.

Reporting Services was running in connected mode for all of our tests, running in the application server (Access Data Services) tier.

Test results

The following tables show the test results of Access Services. For each group of tests, only certain specific variables are changed to show the progressive impact on farm performance.

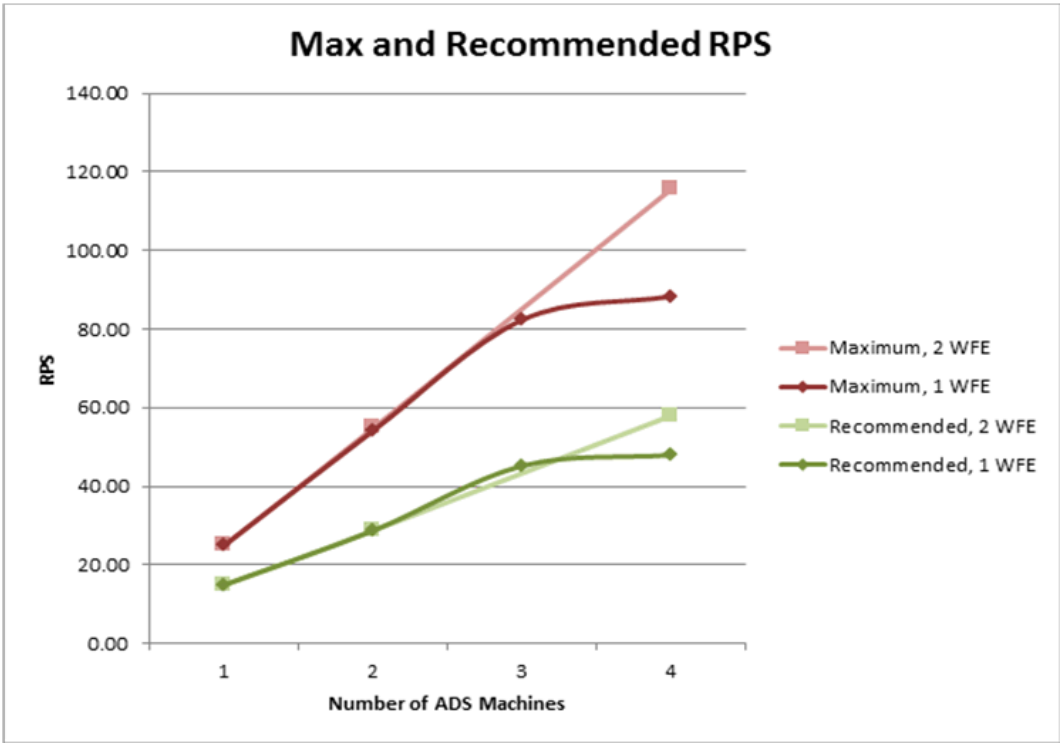
All the tests reported in this article were conducted with think time or wait time. This differs from the capacity planning results for other parts of SharePoint.

For information about bottlenecks of Access Services, see Common bottlenecks and their causes later in this article.

Overall scale

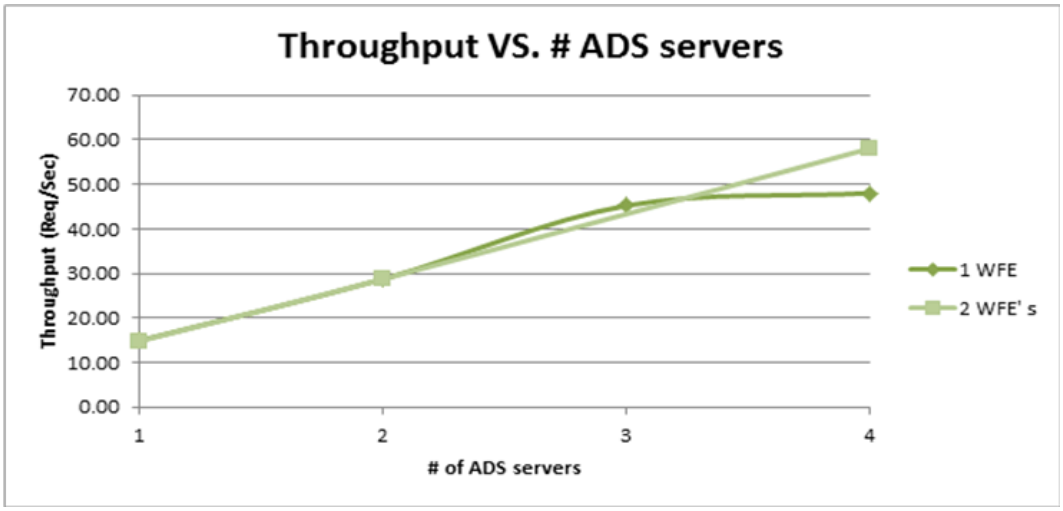
The following table and graph summarize the impact of adding additional front-end Web servers and dedicated Active Data Services computers to the farm. These throughput numbers are specifically for the Active Data Services computers. They do not reflect the impact on the overall farm.

Topology	Baseline solution maximum (RPS)	Baseline recommended (RPS)
1x1	25	15
1x2	54	29
1x3	82	45
1x4	88	48
2x1	25	15
2x2	55	29
2x4	116	58



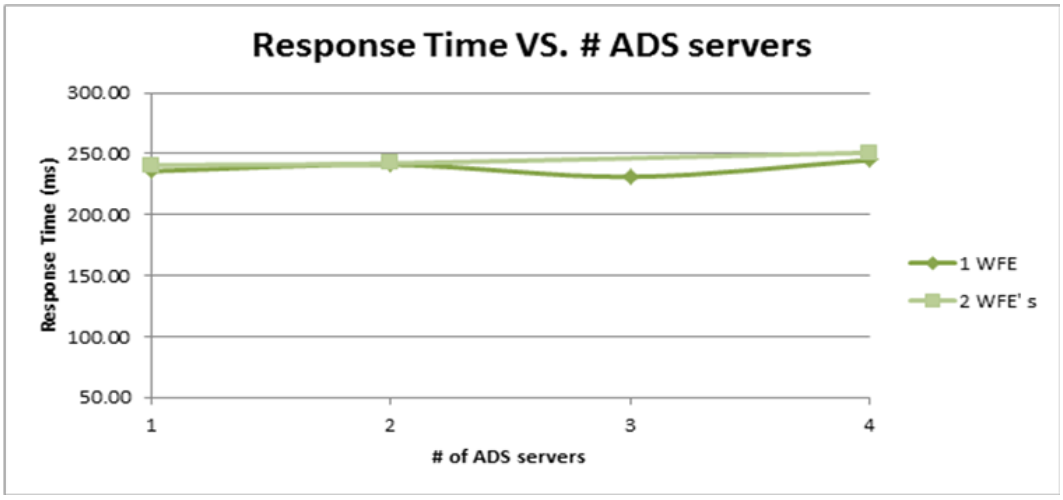
Recommended results

The following graph shows the results for recommended sustainable throughput.

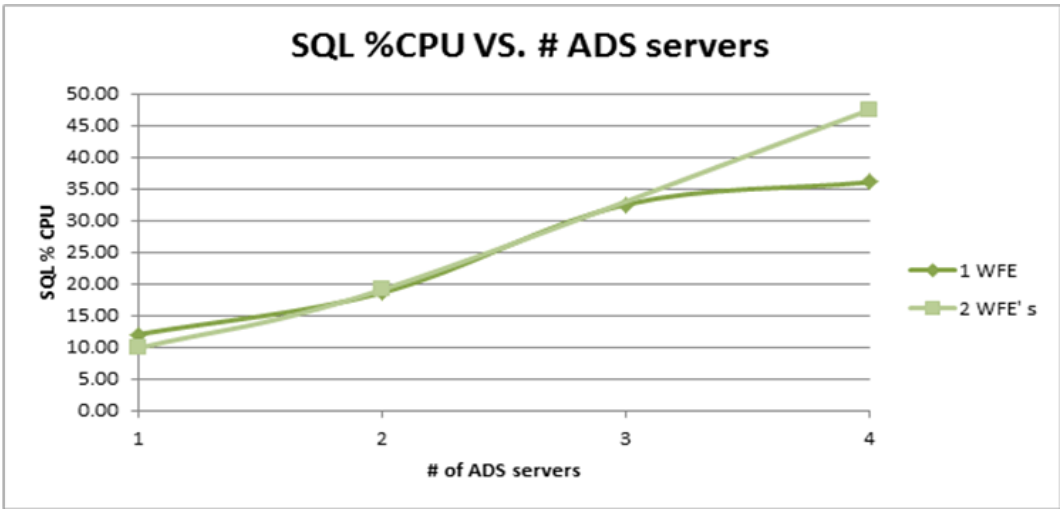


As described earlier in this article, adding the fourth Access Data Services computer shifts the bottleneck to the front-end Web server, and that adding a second front-end Web server resolves the resource constraint on the front-end Web server tier. This would imply, that 1x1, 1x2, and 1x3 are reasonable configurations. However, when the fourth Access Data Services computer is added, a front-end Web server should also be added. Because scaling is in a linear manner (straight line between from 1x1 to 1x4), it can be assumed that the addition of a seventh Access Data Services computer would also imply the addition of a third front-end Web server, and so on, to satisfy the needs of the farm.

Remember that these results are based on a simulated work load only, and that an actual deployment should be monitored to find the point at which additional front-end Web servers are needed to support additional Access Data Services computers. Also, the front-end Web servers are dedicated to Access Services, and in reality the front-end Web servers are likely shared with other SharePoint workloads. The following graph shows the results.



The following graph shows the response time at this throughput level. The response time is very fast, at less than ¼ second on average per request.

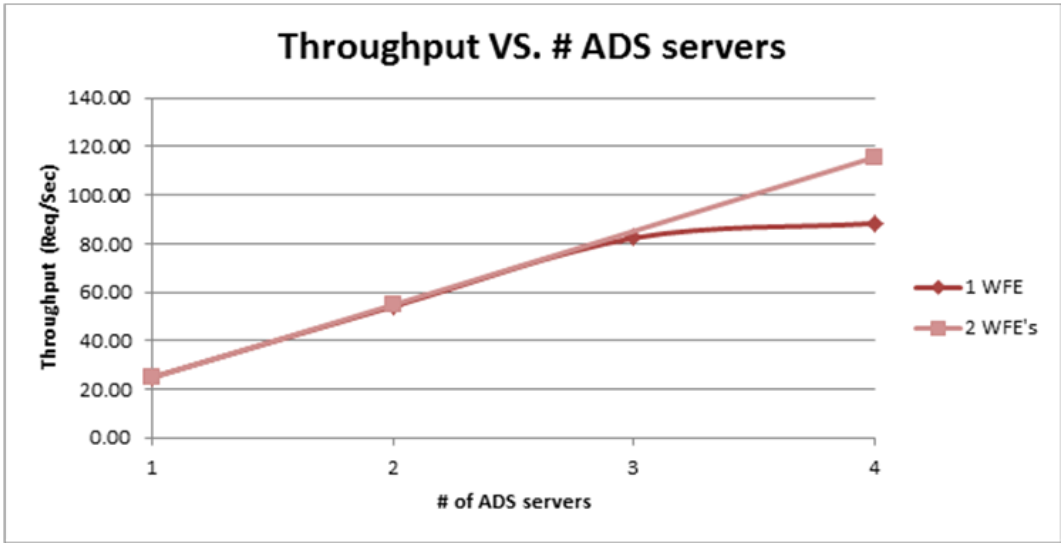


These results show that the SQL Server computer was not a bottleneck, because adding a second front-end Web server resolved the resource shortage, and the SQL Server CPU was always less than 50 percent. However, be aware that the instance of SQL Server is shared with other SharePoint services and SharePoint itself, and so the cumulative effect might drive CPU or disk I/O queue lengths to the point that they do become a bottleneck.

Maximum

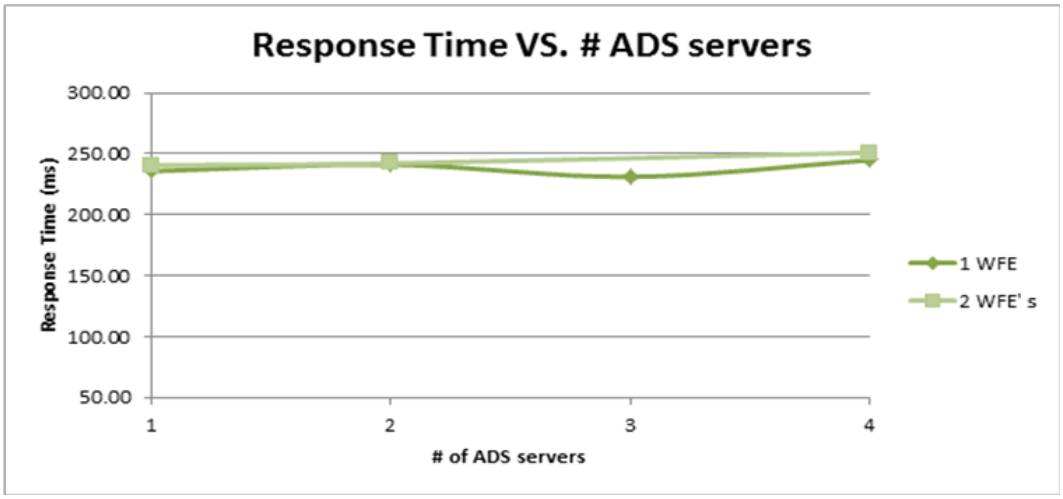
The following graph shows the results, in which throughput was pushed beyond what could be sustained.

In this graph we see that again a second front-end Web server was needed to maximum the usefulness of the fourth Access Data Services computer. Again, your results might vary, because this is highly dependent on the applications and their usage patterns.

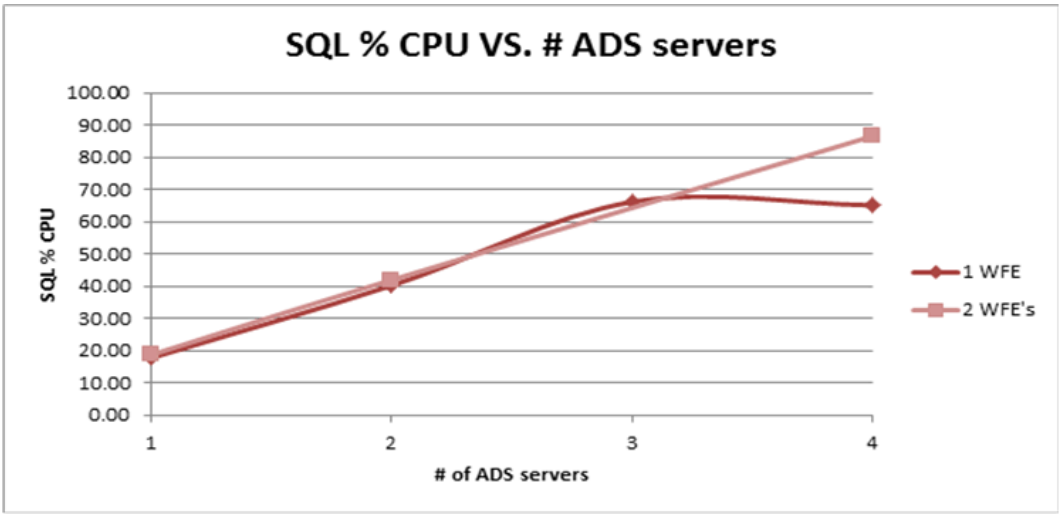


In this case, the response time is increased, as the overall system is under stress. However, these levels are still approximately one second, and acceptable to most users.

It might seem odd that with four Access Data Services computers, two front-end Web servers have an increased response time than one front-end Web server. This is because the overall throughput of the system is increased with two front-end Web servers.



SQL Server is again not a limiting factor here, because adding the second front-end Web server put us back on a linear scaling line. However, we are reaching almost 90 percent CPU usage on the instance of SQL Server. Therefore, there is very little headroom remaining. If we were to add a fifth Access Data Services computer, the SQL Server computer likely would have become the bottleneck.



Detailed results

The following tables show the detailed results for the recommended configurations.

Overall	1x1	1x2	1x3	1x4	2x1	2x2	2x4
Req/Sec	14.96	28.76	45.22	48.01	14.85	28.77	58.02
Tests/Sec	2.00	3.81	6.11	6.42	1.99	3.81	7.80
Average Latency	235.80	241.21	247.21	244.87	240.70	242.26	250.94

Average front-end Web server tier	1x1	1x2	1x3	1x4	2x1	2x2	2x4
%CPU	13.82	24.40	41.02	43.62	6.31	12.48	26.18
Max w3wp Private Bytes	9.46E+08	2.31E+08	1.49E+09	1.55E+09	8.43E+08	9.84E+08	1.19E+09

Average application server (Access Data Services) tier	1x1	1x2	1x3	1x4	2x1	2x2	2x4
%CPU	46.30	42.83	43.74	34.51	46.56	43.45	42.13
%CPU w3wp	33.61	31.15	30.71	24.29	33.48	31.64	29.72
%CPU RS	8.62	7.94	9.17	6.84	9.03	8.02	8.71
Max total Private Bytes	4.80E+09	4.89E+09	4.91E+09	4.62E+09	5.32E+09	4.82E+09	5.07E+09
Max w3wp Private Bytes	2.10E+09	1.97E+09	2.04E+09	1.86E+09	2.00E+09	2.00E+09	2.07E+09
Max RS Private Bytes	1.78E+09	2.00E+09	1.97E+09	1.86E+09	2.30E+09	1.89E+09	2.02E+09

Database server (SQL Server) tier (single computer)	1x1	1x2	1x3	1x4	2x1	2x2	2x4
%CPU	12.07	18.64	32.53	36.05	9.89	21.42	47.46
Avg Private Bytes	2.96E+10	3.22E+10	3.25E+10	3.25E+10	2.89E+10	3.22E+10	3.25E+10
Max Private Bytes	3.26E+10	3.25E+10	3.25E+10	3.25E+10	3.25E+10	3.25E+10	3.25E+10

Database server (SQL Server) tier (single computer)	1x1	1x2	1x3	1x4	2x1	2x2	2x4
Avg Disk Queue Length Total	0.74	1.18	1.64	1.77	0.67	1.24	2.18

The following tables show the detailed results for the maximum configurations.

Overall	1x1	1x2	1x3	1x4	2x1	2x2	2x4
Req/Sec	14.96	28.76	45.22	48.01	14.85	28.77	58.02
Tests/Sec	2.00	3.81	6.11	6.42	1.99	3.81	7.80
Average Latency	235.80	241.21	247.21	244.87	240.70	242.26	250.94

Average front-end Web server tier	1x1	1x2	1x3	1x4	2x1	2x2	2x4
%CPU	13.82	24.40	41.02	43.62	6.31	12.48	26.18
Max w3wp Private Bytes	9.46E+08	2.31E+08	1.49E+09	1.55E+09	8.43E+08	9.84E+08	1.19E+09

Average application server (Access Data Services) tier	1x1	1x2	1x3	1x4	2x1	2x2	2x4
%CPU	46.30	42.83	43.74	34.51	46.56	43.45	42.13
%CPU w3wp	33.61	31.15	30.71	24.29	33.48	31.64	29.72
%CPU RS	8.62	7.94	9.17	6.84	9.03	8.02	8.71
Max total Private Bytes	4.80E+09	4.89E+09	4.91E+09	4.62E+09	5.32E+09	4.82E+09	5.07E+09
Max w3wp Private Bytes	2.10E+09	1.97E+09	2.04E+09	1.86E+09	2.00E+09	2.00E+09	2.07E+09
Max RS Private Bytes	1.78E+09	2.00E+09	1.97E+09	1.86E+09	2.30E+09	1.89E+09	2.02E+09

Database server (SQL Server) tier (single computer)	1x1	1x2	1x3	1x4	2x1	2x2	2x4
%CPU	12.07	18.64	32.53	36.05	9.89	21.42	47.46
Avg Private Bytes	2.96E+10	3.22E+10	3.25E+10	3.25E+10	2.89E+10	3.22E+10	3.25E+10
Max Private Bytes	3.26E+10	3.25E+10	3.25E+10	3.25E+10	3.25E+10	3.25E+10	3.25E+10

Database server (SQL Server) tier (single computer)	1x1	1x2	1x3	1x4	2x1	2x2	2x4
Avg Disk Queue Length Total	0.74	1.18	1.64	1.77	0.67	1.24	2.18

Recommendations

This section provides general performance and capacity recommendations.

Access Services capacity and performance is highly dependent on the makeup of the applications that are hosted on the service. The size of tables and the complexity of queries often have the most effect. The testing used representative sizes and complexities, but every application and dataset is different. Therefore, the capacity and performance will depend on the applications in use, their specific complexity, and the data size.

Hardware recommendations

Access Services uses standard hardware for both front-end Web servers and application servers; no special requirements are necessary. General SharePoint Server 2010 guidelines for CPU number, speed, and memory are applicable for computers in the application server (Access Data Services) tier.

Scaled-up and scaled-out topologies

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 servers 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 Access Services scenario:

- To provide for more user load, check the CPU for the existing Access Services application servers. Add additional CPUs or cores, or both, to these servers if it is possible. Add more Access Services server computers as needed. This can be done to the point that the front-end Web server has become the bottleneck, and then add front-end Web servers as needed.
- In our tests, memory on the front-end Web server tier and application server (Access Data Services) tier was not a bottleneck. Depending on the size of the result sets, memory could become

an issue. However, we do not expect that to be the norm. Track the private bytes for the Access Data Services w3wp process, as described here.

- In our tests, SQL Server was not a bottleneck. However, our tests were run in isolation from other SharePoint Server 2010 services. SQL Server CPU and disk I/O should be monitored and additional servers or spindles added as needed.

Performance-related Access Services settings

One way to control the performance characteristics of Access Services is to limit the size and complexity of queries that can be performed. Access Services provides a set of configurable throttles for controlling queries. Each of the following queries can be set through SharePoint Central Administration. (In the **Application Management** section, click **Manage Service Applications**, and then click **Access Services**.)

In general, how much data that has to be retrieved from SharePoint to perform a query will have a significant effect on performance. This can be controlled in several ways. First, the inputs to a query can be limited:

- Maximum Sources per Query
- Maximum Records per Table

Second, the resulting size of a query can be limited:

- Maximum Columns per Query
- Maximum Rows per Query
- Allow Outer Joins

In addition to the size of the query (data size in and out), the processing complexity on the data can be controlled, to reduce the CPU load on the application server (Access Data Services) tier:

- Maximum Calculated Columns per Query
- Maximum Order by Clauses per Query

Obviously, the previous settings will affect the applications that can be run on the server. For example, if an application is written with 40 output columns from a query, and the settings are below this level, the application will throw a runtime error. A balance between user need and acceptable performance must be struck, and is highly dependent on the kind of Access applications that are expected to be run on the farm.

One additional, more extreme measure can be taken. SharePoint Server 2010 supports a set of query operations natively, which Access Services augments to cover a broader set of application scenarios. For Access Services to improve queries from SharePoint, there is the potential that a large amount of data might have to be retrieved from the SharePoint content database. Instead, Access Services can be set to stick with only query operations, which can be natively supported by SharePoint. Therefore, avoiding the data fetch required for more complex operations:

- Allow Non-Remotable Queries

Optimizations

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 table in Troubleshooting later in this article lists some common bottlenecks and describes their causes and possible resolutions.

Performance monitoring

To help you determine when you have to scale up or scale out the system, use performance counters to monitor the health of the 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.

Front-end 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	Processor(_Total)	Shows the percentage of elapsed time that this thread used the processor to execute instructions.
Private Bytes	Process(w3wp)	This value should not approach the Max Private Bytes set for w3wp processes. Iif it does, additional investigation is needed into what component is using the memory.

Access Data Services

The following table shows performance counters and processes to monitor for application servers, or Access Data Services (Access Data Services) in this case, within your farm.

Performance counter	Apply to object	Notes
% Processor Time	Processor(_Total)	Shows the percentage of elapsed time that this thread used the processor to execute instructions.
% Processor Time	Process(w3wp)	The Access Data Services runs within its own w2wp process, and it will be obvious which w2wp process this is as it will be getting the bulk of the CPU time.
Avg. Disk Queue Length	PhysicalDisk(_Total)	Watch for too much disk writing because of logging.
% Processor Time	Process(ReportingServicesService)	Reports are handled by SQL Server Reporting Services. If too many reports are being run, or if the reports are very complex, then the CPU and Private Bytes for this process will be high.
Private Bytes	Process(w3wp)	Access Services caches the results of queries in memory, until the user's session expires (the time-out for which is configurable). If a large amount of data is being processed through the Access Data Services, memory consumption for the Access Data Services' w3wp will increase.
Private Bytes	Process(ReportingSrevicesService)	Reports are handled by SQL Server Reporting Services. If too many reports are being run, or reports are very complex, the CPU and Private Bytes for this process will be high.

Database servers

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

Performance counter	Apply to object	Notes
% Processor Time	Processor(_Total)	Shows the percentage of elapsed time that this thread used the processor to execute instructions.
% Processor Time	Process(sqlservr)	Average values larger than 80 percent indicate that processor capacity on the database server is insufficient.
Private Bytes	Process(sqlservr)	Shows the average amount of memory being consumed by SQL Server.
Avg. Disk Queue Length	PhysicalDisk(_Total)	Shows the average disk queue length; the database requests waiting to be committed to disk. This is often a good indicator that the instance of SQL Server is becoming overloaded, and that possibly additional disk spindles would help distribute the load.

Troubleshooting

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

Bottleneck	Cause	Resolution
Access Data Services CPU	Access Services depends on a large amount of	Increase the number of CPUs or cores, or both, in the existing Access Data Services

Bottleneck	Cause	Resolution
	processing in the application server tier. If a 1x1, 1x2, or 1x3 configuration is used, the first bottleneck encountered will likely be the CPU on the Access Data Services servers.	computers. Add additional Access Data Services computers if possible.
Web server CPU usage	When a Web server is overloaded with user requests, average CPU usage 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 more Web servers to the farm to distribute user load, or you can scale up the Web server or servers by adding higher-speed processors.
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) contains useful information about resolving disk I/O issues.
Reporting Services CPU utilization	The Reporting Services process is using a large share of the CPU resources.	Dedicate a computer to Reporting Services, taking load from the application server (Access Data Services) tier (connected mode) or the front-end Web server tier (local mode).

Estimate performance and capacity requirements for Excel Services in SharePoint Server 2010

This article describes the effects of using Excel Services in Microsoft SharePoint Server 2010 on topologies running Microsoft SharePoint Server 2010. You can use this information to better scale your deployments based on your latency and throughput requirements.

Note:
It is important to be aware that the specific capacity and performance figures presented in this article will differ from the figures in real-world environments. The figures 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 the system will support the needs of your environment.

In this article:

For general information about how to plan and run your capacity planning for SharePoint Server 2010, see Capacity management and sizing for SharePoint Server 2010.

Test farm characteristics

This section describes the dataset, workloads, hardware settings, topology, and test definitions that were used during the performance and capacity testing of Excel Services.

Dataset

Excel Services capacity and performance is highly dependent on the makeup of the workbooks that are hosted on the service. The size of the workbook and the complexity of calculations have the most impact. Our testing used representative sizes and complexities, but every workbook is different, and your capacity and performance depends on the actual workbooks you use, and their specific size and complexity.

We simulated Excel workbooks on a farm dedicated to Excel to evaluate our capacity profile. Note that no other SharePoint Server tests were running during our capacity profile tests. Within this farm, we used three buckets of workbooks – Small, Large, and Very Large – based on workbook size and complexity:

Workbook Characteristics	Small	Large	Very Large
Sheets	1-3	1-5	1-20
Columns	10-20	10-500	10-1,000
Rows	10-40	10-10,000	100-30,000
Calculated Cells	0-20%	0-70%	0-70%
Number of Formats	1-10	1-15	1-20
Tables	0-1	0-2	0-5
Charts	0-1	0-4	0-4
Workbook Uses External Data	0%	20%	50%
Workbook Uses a Pivot Table	0%	3%	3%
Workbook Uses Conditional Formats	0%	10%	20%

This test farm included 2,000 SharePoint Server sites. Each site contained one small, one large, and one very large workbook. The distribution of the workbooks on the SharePoint Server pages was 10% small workbooks and 90% large and very large workbooks. Additionally, the test farm dataset included SharePoint Server pages that contained 1-5 Excel Web Parts.

Workload

To simulate application usage, workloads were created to perform one or more of the following operations:

Action Mix	Small Workbook	Large Workbook
View	50%	70%
Edit	35%	15%
Collaborative Viewing	10%	10%
Collaborative Editing	5%	5%

In addition, 17% of all the workbooks included external data. For large and very large workbooks that included external data, refreshes were performed 80% of the time; small workbooks do not include external data.

Each workload includes think time between user actions of 10 seconds. Think time refers to user action delays that simulate how long a user might take to perform the actions. This differs from other SharePoint Server 2010 capacity planning documents. Excel Services is stateful —the workbook is maintained in memory between user interactions — making it important to simulate a full user session and not merely individual requests. On average, there are 0.2 requests per second for a single user workload.

We randomly selected one of the 2,000 sites to run the test for each workload. We used the percentages in the following table to select application and application size, within that site.

Workbook Selection	Use Percentage
Small Workbook	30%
Large Workbook	55%
Dashboard	10%
Very Large Workbook	5%

Green and Red Zone definitions

For each configuration two zones were determined before throughput tests were performed. One zone was the green zone or recommended zone in which throughput can be sustained. The other zone was the red zone or maximum zone in which throughput can be tolerated for a short time but should be avoided.

To determine our red and green zone user loads, we first conducted a step test and then stopped when the following conditions were met:

- **Green zone** We stopped at the point when any of the computers in our farm (Web front-end, Excel Calculation Services, or Microsoft SQL Server) exceeded 50% CPU usage or the response time for the overall system exceeded 1 second.
- **Red Zone** We stopped at the point where the successful RPS for the Excel Calculation Services computers in the farm was at a maximum. Past this point, the overall throughput for the farm started to decrease and/or we would start to see failures from one of the tiers. Often the maximum private bytes in Excel Calculation Services would be exceeded when throughput was in the red zone.

After conducting the step tests, we retreated from these maximum values to run a longer constant load test of 1 hour. We stopped the green zone test when 75% of the load was used. We peaked in the red zone step test when we used 65% of the load. If the green zone test was limited by memory, and the

CPU usage percentage never exceeded 50%, we instead used 75% of the load number calculated for the red zone.

The average response time was less than .25 seconds for both green and red zones, and for both scale-out and scale-up tests.

Hardware Settings and Topology

This section describes the kinds of computer hardware we used in our lab and the farm configuration topologies that we used in our tests.

Lab Hardware

Several farm configurations were used for our testing to provide a high level of test-result detail. The farm configurations ranged from one to three Web front-end servers, one to three application servers for Excel Services and Excel Calculation Services, and a single database server computer that is running Microsoft SQL Server 2008. Additionally, our tests used four client computers. All servers were 64-bit, and the client computers were 32-bit.

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

Machine Role	CPU	Memory	Network
Web front-end server	2 proc/4 core 2.33 GHz Intel Xeon	8 GB	1 gig
Excel Calculation Services	2 proc/4 core 2.33 GHz Intel Xeon	8 GB	1 gig
SQL Server	4 proc/4 core 2.6 GHz Intel Xeon	16 GB	1 gig

Topology

Our testing experience indicates that memory on the Excel Calculation Services tier and CPU on the Web front-end server tier are the most important limiting factors for throughput. Be aware that your experience may vary. As a result, we varied the number of computer servers in both tiers for the scale-out tests.

We deployed a topology of 1:1 for the Excel Calculation Services and Web front-end servers for the scale-up tests, and then varied the number of processors and available memory in the Excel Calculation Services computers.

Excel Calculation Services is not especially demanding on the SQL Server instance running SharePoint Server 2010, as the workbook is read a binary large object (BLOB) from SharePoint Server 2010 and put in memory on the Excel Calculation Services tier (and additionally disk cached). At no time did SQL

Server become a bottleneck. For all tests, bottleneck is defined as a state in which the capacity of a particular component of a farm is reached.

Test Results

The following tables show the test results of Excel Services in Microsoft SharePoint Server 2010. For each group of tests, only certain specific variables are changed to show the progressive effect on farm performance.

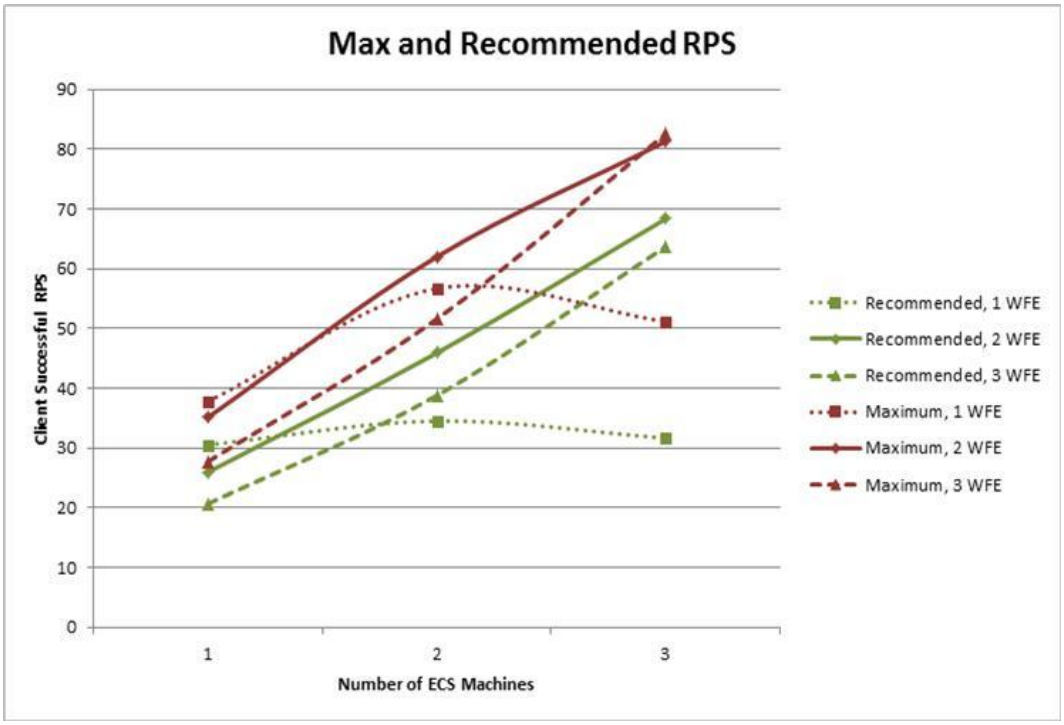
Note that all the tests reported on in this article were conducted with think or wait time (think time equals 10 seconds between user actions). This differs from the capacity planning results for other parts of SharePoint Server 2010.

For information about Excel Services bottlenecks, see the Common bottlenecks and their causes section in this article.

Overall Scale

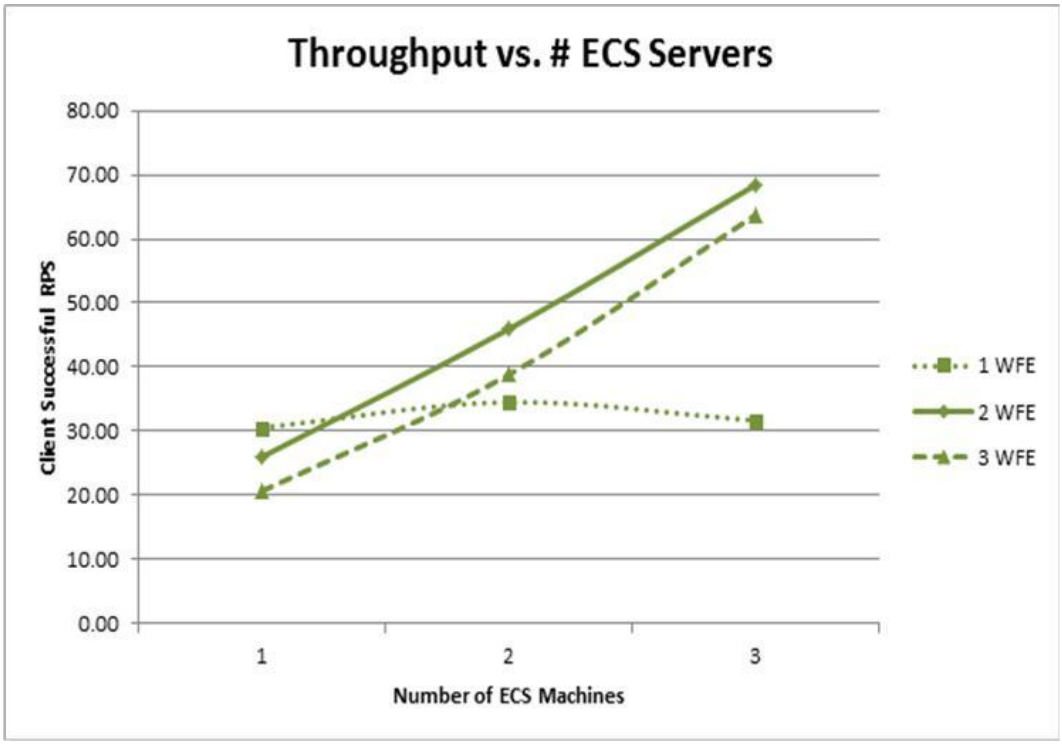
The table here summarizes the effect of adding additional Web Front-End and dedicated Excel Calculation Services computers to the farm. These throughput numbers are specifically for the Excel Calculation Services computers, and do not reflect the effect on the overall farm.

Topology	Baseline Maximum (RPS)	Baseline Recommended (RPS)
1x1	38	31
1x2	35	26
1x3	28	21
2x1	57	35
2x2	62	46
2x3	52	39
3x1	51	32
3x2	81	69
3x3	83	64

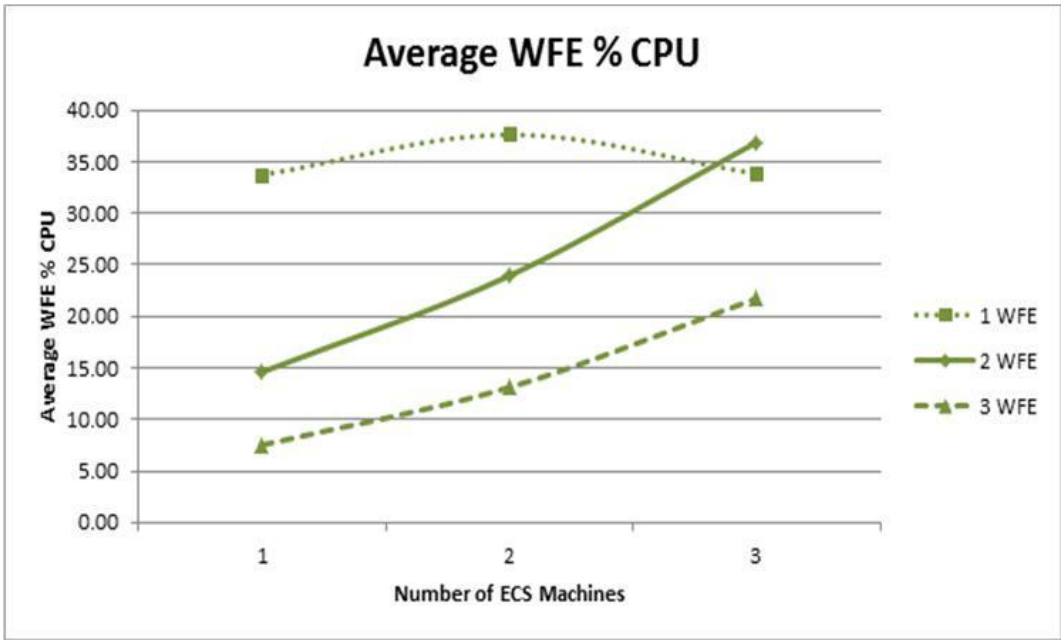


Recommended Results

The following chart shows our results for recommended sustainable throughput.



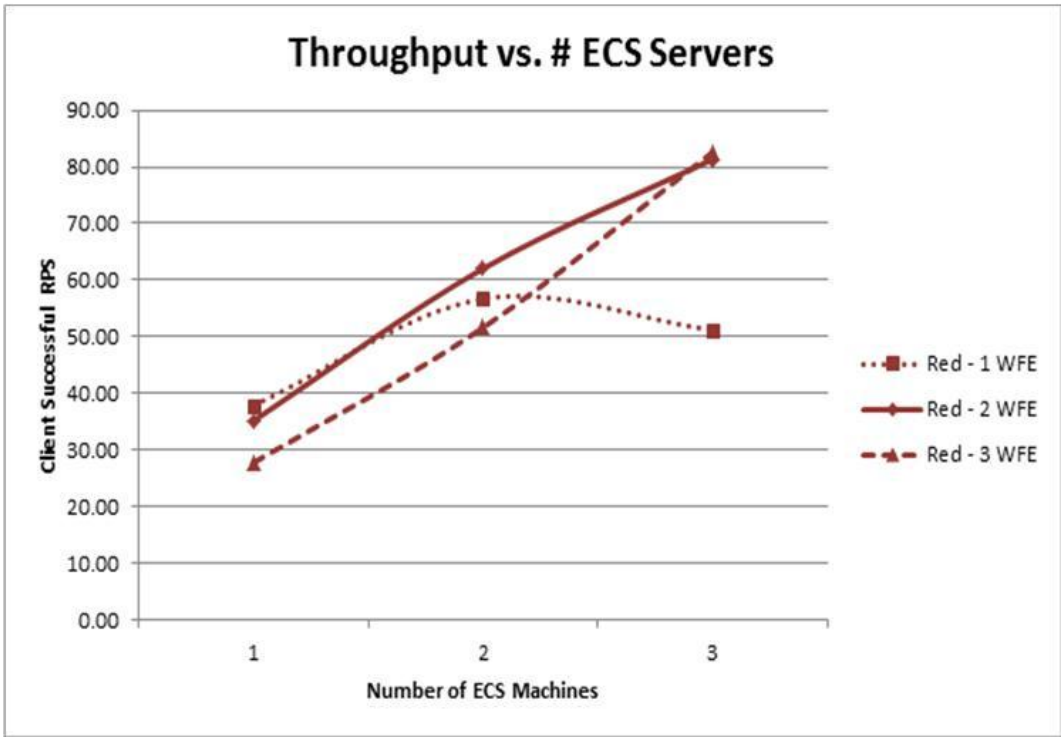
The previous chart shows that there is overhead associated with adding Web front-end computers to the farm. However, this is offset as Excel Calculation Services computers are added. A single Web front-end became the bottleneck after adding two additional Excel Calculation Services computers. This Web front-end bottleneck reversed any benefit that was gained from the additional capacity of adding a second and third Excel Calculation Services computer. Also notice that three Web front-end computers did not add any more throughput, as Excel Calculation Services became the limiting factor.



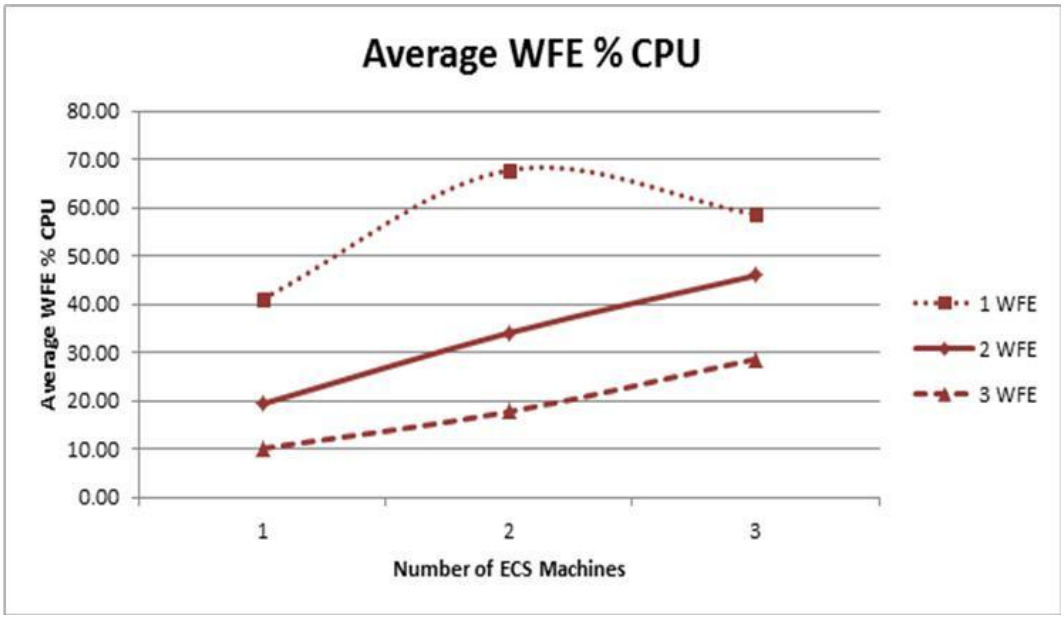
Notice in the previous chart that as Web front-end computers are added, the CPU load on each computer is reduced significantly. Note too, that with two Web front-end computers and three Excel Calculation Services computers, the CPU load is reaching the maximum seen for a single Web front-end computer. This implies that adding another Excel Calculation Services computer would make the Web front-end tier the limiting factor. Remember that these results are for the “recommended” load. This is why the CPU load is maxing out at around 35% instead of at an increased level.

Maximum Results

The following chart shows our results for maximum peak throughput.



Similar to our recommended results, we see that a single Web front-end computer is the limiting factor as we add a second and third Excel Calculation Services computer. Also notice that exactly as with the recommended results, adding a third Web front-end computer does not add to throughput as Excel Calculation Services is the limiting factor after the second Web front-end computer is added.



The results in the previous chart show that multiple Web front-end computers do not become as heavily loaded as a single Web front-end computer configuration. This indicates that the Excel Calculation Services computers are the bottleneck after the second Web front-end computer is added.

Detailed Results

This section shows details for the recommended and maximum results obtained in our tests.

Recommended Results

The following tables show the recommended results of our tests.

Overall	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
Client Successful RPS	30.56	34.55	31.67	26.03	45.94	68.37	20.71	38.82	63.70
Client Response Time (sec.)	0.22	0.18	0.19	0.16	0.19	0.20	0.15	0.15	0.17
TPS	1.58	1.77	1.61	1.40	2.38	3.54	1.08	2.03	3.25

Web Front-end Tier	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
% CPU (average over all Web Front-end computers	33.73	37.64	33.84	14.61	23.95	36.90	7.54	13.12	21.75

Excel Calculation Services Tier	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
% CPU (average over all Excel Calculation Services computers)	30.56	34.55	31.67	26.03	45.94	68.37	20.71	38.82	63.70
Peak Private Bytes (maximum over all Excel Calculation Services computers)	5.94E+09	5.82E+09	5.79E+09	5.87E+09	6.09E+09	5.92E+09	5.79E+09	5.91E+09	5.85E+09

Maximum Results

The following tables show the maximum results of our tests.

Overall	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
Client Successful RPS	37.85	56.70	51.17	35.19	62.04	81.31	27.79	51.62	82.58
Client Response Time (sec.)	0.19	0.28	0.23	0.16	0.20	0.25	0.16	0.16	0.22

Overall	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
TPS	1.92	2.96	2.59	1.81	3.21	4.60	1.41	2.72	4.30

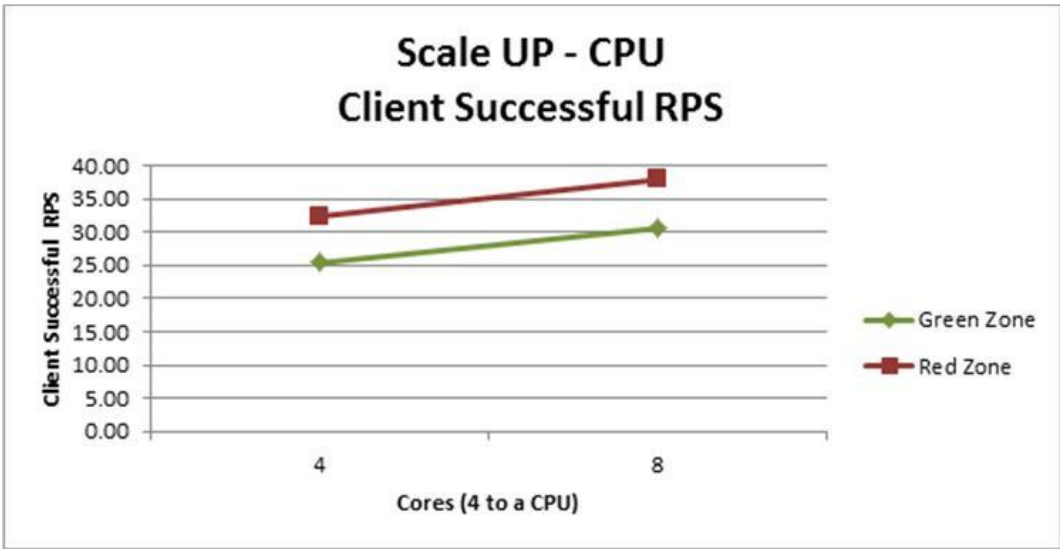
Web Front-end Tier	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
% CPU (average over all Web Front-end computers	41.08	67.78	58.59	19.44	34.11	45.97	10.19	17.79	28.69

Excel Calculation Services Tier	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
% CPU (average over all Excel Calculation Services computers)	24.99	18…44	10.96	23.57	20.56	17.77	18.97	17.04	18.10
Peak Private Bytes (maximum over all Excel Calculation Services computer	5.91E+09	5.85E+09	5.91E+09	5.88E+09	5.99E+09	6.502E+09	5.94E+09	5.94E+09	6.04E+09

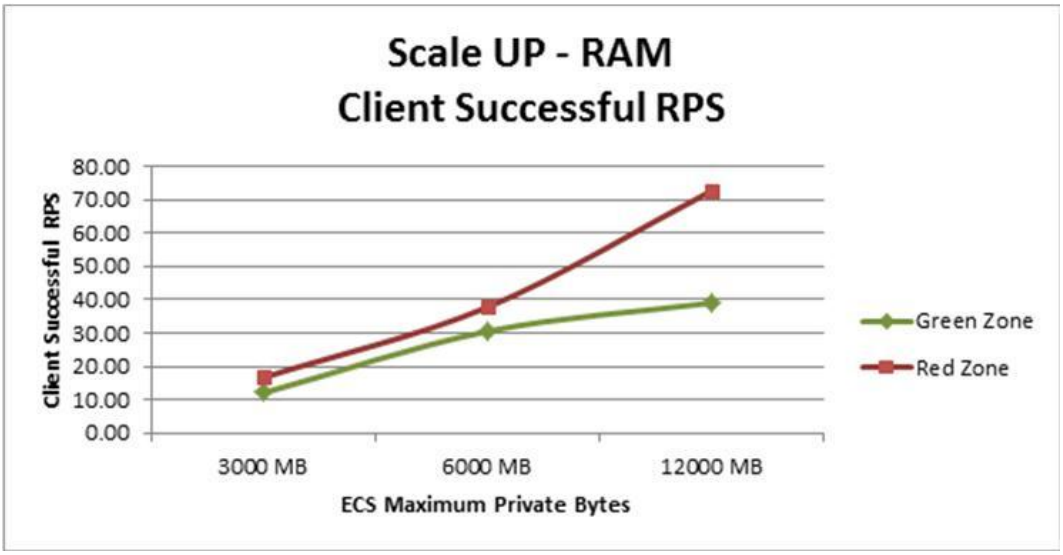
Excel Calculation Services Tier	1x1	1x2	1x3	2x1	2x2	2x3	3x1	3x2	3x3
s)									

Scale Up Test results

We also measured the effect of adding CPUs and memory to the Excel Calculation Services tier. For these tests, a 1x1 topology was used.



Our results in the previous chart show that adding additional CPUs was helpful but did not significantly affect the overall throughput.



The red zone line in the previous chart shows however, that adding memory does have a significant effect on throughput, especially at peak times. In this test, the same hardware was used throughout. However, the Maximum Private Bytes for the Excel Services process was limited. Since workbooks are kept in memory, the size of the workbooks has a significant effect on how many workbooks, and also how many users, any Excel Calculation Services computer can support.

Recommendations

This section provides general performance and capacity recommendations for hardware, Excel Services settings, common bottlenecks and troubleshooting.

Note that Excel Services capacity and performance is highly dependent on the makeup of the workbooks that are hosted on the service. The size of the workbook and the complexity of calculations have the most effect. Our testing used representative sizes and complexities, but every workbook is different, and your capacity and performance depends on the specific size and complexity of the workbooks you use.

Hardware Recommendations

Excel Services uses standard hardware for both Web front-end servers and application servers, there are no special requirements. General SharePoint Server 2010 guidelines on CPU number, speed, and memory are applicable for computers in the Excel Calculation Services tier. Note that one of the first bottlenecks an Excel Calculation Services computer is likely to encounter is memory and this may require you to add resources. Before you do, we recommend that you test with a representative set of workbooks from your organization, as the size and complexity of workbooks have a large effect on how much more capacity the addition of memory is likely to have.

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 servers 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 Excel Services scenario:

- To provide for more user load, check the CPU and memory for the existing Excel Services application servers. Add additional memory if the CPU is not a concern, or add CPUs if memory is not a concern. If both memory and CPU are reaching their upper limits, additional Excel Calculation Services computers may be necessary. Add additional Excel Calculation Services or application servers until the point that the Web front-end servers become the bottleneck, and then add Web front-end servers as needed.
- In our tests, SQL Server was not a bottleneck. Excel Services does not make large demands on the database tier, as workbooks are read and written as whole documents, and also workbooks are held in memory throughout the user's session.

Performance-Related Excel Services Settings

One of the ways to control the performance characteristics of Excel Services is to control how memory is used. Each of the global settings in the following list are set through SharePoint Server 2010 Central Administration > Application Management: Manage Service Applications > Excel Services Application > Global Settings:

- **Maximum Private Bytes** — By default, Excel Calculation Services will use up to 50% of the memory on the computer. If the computer is shared with other services, it may make sense to lower this number. If the computer is not being shared and is dedicated to Excel Calculation Services, and is indicating that memory may be a limiting factor, increasing this number may make sense. In any event, experimenting by adjusting this number can guide the administrator to making the necessary changes in order to better scale up.
- **Memory Cache Threshold** — Excel Calculation Services will cache unused objects (for example, read-only workbooks for which all sessions have timed out) in memory. By default, Excel Calculation Services will use 90% of the **Maximum Private Bytes** for this purpose. Lowering this number can improve overall performance if the server is hosting other services in addition to Excel Calculation Services. Increasing this number increases the chances that the workbook being requested will already be in memory and will not have to be reloaded from the SharePoint Server content database.
- **Maximum Unused Object Age** — By default, Excel Calculation Services will keep objects in the memory cache as long as possible. To reduce the Excel Calculation Services memory usage, in particular with other services that are running on the same computer, it may make more sense to impose a limit on how long objects are cached in memory.

There are also settings available to control the maximum size of a workbook and the lifetime of a session, which in turn control how long a workbook is held in memory. These settings are associated

with each trusted location and are not global. These settings can be set through SharePoint Server 2010 Central Administration > Application Management: Manage Service Applications > Excel Services Application > Trusted Locations, and then edit the settings for each trusted location in the Workbook Properties section on the **Edit Trusted File Location** page.

- **Maximum Workbook Size**
- **Maximum Chart or Image Size**

By default, Excel Calculation Services is limited to 10 MB or smaller workbooks and 1 MB or smaller charts/images. Obviously using larger workbooks and larger charts/images puts more strain on the available memory of the Excel Calculation Services tier computers. However, there may be users in your organization that need these settings to be increased for Excel Calculation Services to work with their particular workbooks.

- **Session Timeout** — By decreasing the session time out, memory is made available for either the unused object cache or other services faster.
- **Volatile Function Cache Lifetime** — Volatile functions are functions that can change their value with each successive recalculation of the workbook, for example date/time functions, random number generators, and so on. Because of the load this could generate on the server, Excel Calculation Services does not recalculate these values for each recalculation, instead caching the last values for a short time period. Increasing this lifetime can reduce the load on the server. However, this depends on having workbooks that use volatile functions.
- **Allow External Data** — Excel Calculation Services can draw on external data sources. However, the time that is required to draw upon the external source can be significant, with potentially a large amount of data returned. If external data is allowed, there are several additional settings that can help throttle the effect of this feature.

Common bottlenecks and their causes

During performance testing, several different common bottlenecks were revealed. Bottlenecks are defined as a state in which the capacity of a particular component 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.

Troubleshooting performance and scalability

Bottleneck	Cause	Resolution
Excel Calculation Services Memory	Excel Services holds each workbook in memory throughout the user's session. A large number of workbooks, or large workbooks, can cause Excel Calculation Services to consume	Scale Up with more memory in the Excel Calculation Services tier computers, or Scale Out with the addition of more Excel Calculation Services computers. The choice will partly depend on

Bottleneck	Cause	Resolution
	all available memory causing the actually consumed "Private Bytes" to exceed "Maximum Private Bytes."	if CPU is also reaching a maximum.
Excel Calculation Services CPU	Excel Services can depend on a large amount of processing in the application tier, depending on the number and complexity of workbooks.	Increase the number of CPUs and/or cores in the existing Excel Calculation Services computers, or add Excel Calculation Services computers.
Web server CPU usage	When a Web server is overloaded with user requests, average CPU usage 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 Web servers to the farm to distribute user load, or you can scale up the Web server or servers by adding faster processors.

Performance monitoring

To help you determine when you have to scale up or scale out the system, use performance counters to monitor the health of the 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.

Front-end Web server

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

Performance Counter	Apply to object	Notes
% Processor Time	Processor (w3wp)	Shows the percentage of elapsed time that this thread used the processor to execute instructions.
% Processor Time	Processor (_Total)	Shows the percentage of elapsed time that all threads on the server

Performance Counter	Apply to object	Notes
		computer that used the processor to execute instructions.
Private Bytes	Process (w3wp)	This value should not approach the Max Private Bytes set for w3wp processes. If it does, additional investigation is needed into what component is using the memory.

Excel Calculation Services

The following table shows performance counters and processes to monitor for application servers, or in this case Excel Calculation Services, within your farm.

Performance Counter	Apply to object	Notes
% Processor Time	Processor (_Total)	Shows the percentage of elapsed time that all threads on the server that used the processor to execute instructions.
% Processor Time	Processor (w3wp)	The Excel Calculation Services runs within its own w3wp process, and it will be obvious which w3wp process this is as it will be getting the bulk of the CPU time.
Average Disk Queue Length	PhysicalDisk(_Total)	Watch for too much disk writing because of logging.
Private Bytes	Process(w3wp)	Excel Services caches workbooks in memory, until the user's session expires (the time out for which is configurable). If a large amount of data is being processed through the Excel Calculation Services, memory

Performance Counter	Apply to object	Notes
		consumption for the Excel Calculation Services w3wp will increase.

SQL Server

As we have previously described, Excel Services is light on the SQL Server tier, as workbooks are read once into memory on the Excel Calculation Services tier during the user's session. Follow general SharePoint Server guidelines for monitoring and troubleshooting of the SQL Server tier.

Estimate performance and capacity requirements for PerformancePoint Services

This article describes the effect that use of PerformancePoint Services has on topologies running Microsoft SharePoint Server 2010.

Note:
It is important to be aware that the specific capacity and performance figures presented in this article will differ from the figures in real-world environments. The figures 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 the system will support the factors in your environment.

In this article:

For general information about how to plan and run your capacity planning for SharePoint Server 2010, see Capacity management and sizing for SharePoint Server 2010.

Test farm characteristics

Dataset

The dataset consisted of a corporate portal built by using SharePoint Server 2010 and PerformancePoint Services that contained a single, medium-sized dashboard. The dashboard contained two filters linked to one scorecard, two charts, and a grid. The dashboard was based on a single Microsoft SQL Server 2008 Analysis Services (SSAS) data source that used the AdventureWorks sample databases for SQL Server 2008 Analysis Services cube.

The table that follows describes the type and size of each element on the dashboard.

Name	Description	Size
Filter One	Member selection filter	7 dimension members
Filter Two	Member selection filter	20 dimension members
Scorecard	Scorecard	15 dimension member rows by 4 columns (2 KPIs)
Chart One	Line chart	3 series by 12 columns

Name	Description	Size
Chart Two	Stacked bar chart	37 series by 3 columns
Grid	Analytic grid	5 rows by 3 columns

The medium dashboard used the Header and Two Columns template, and the dashboard item sizes were set to either auto-size or a specific percentage of the dashboard. Each item on the dashboard was rendered with a random height and width between 400 and 500 pixels to simulate the differences in Web browser window sizes. It is important to change the height and width of each dashboard item because charts are rendered based on Web browser window sizes.

Test scenarios and processes

This section defines the test scenarios and discusses the test process that was used for each scenario. Detailed information such as test results and specific parameters are given in the "Test results" sections later in this article.

Test name	Test description
Render a dashboard and randomly change one of the two filters five times with a 15 second pause between interactions.	1. Render the dashboard. 2. Select one of the two filters and randomly select a filter value and wait until the dashboard is re-rendered. 3. Repeat four more times, randomly selecting one of the two filters and a random filter value.
Render a dashboard, select a chart, and expand and collapse it five times with a 15 second pause between interactions.	1. Render the dashboard. 2. Select a random member on a chart and expand it. 3. Select another random member on the chart and collapse it. 4. Select another random member on the chart and expand it. 5. Select another random member on the chart and collapse.
Render a dashboard, select a grid, and expand and collapse it five times with a 15 second pause between interactions.	1. Render the dashboard. Select a random member on a grid and expand the member. 2. Select another random member on the grid

Test name	Test description
	and expand it. 3. Select another random member on the grid and collapse it. 4. Select another random member on the grid and expand it.

A single test mix was used that consisted of the following percentages of tests started.

Test name	Test mix
Render a dashboard and randomly change one of the two filters five times.	80%
Render a dashboard, select a chart, and expand and collapse it five times.	10%
Render a dashboard, select a grid, and expand and collapse it five times.	10%

Microsoft Visual Studio 2008 Load Testing tools were used to create a set of Web tests and load tests that simulated users randomly changing filters and navigating on grids and charts. The tests used in this article contained a normal distribution of 15-second pauses, also known as "think times," between interactions and a think time between test iterations of 15 seconds. Load was applied to produce a two-second average response time to render a scorecard or report. The average response time was measured over a period of 15 minutes after an initial 10 minute warm-up time.

Each new test iteration select a distinct user account from a pool of five thousand accounts and a random IP address (using Visual Studio IP Switching) from a pool of approximately 2,200 addresses.

The test mix was run two times against the same medium-sized dashboard. In the first run, the data source authentication was configured to use the Unattended Service Account, which uses a common account to request the data. The data results are identical for multiple users, and PerformancePoint Services can use caching to improve performance. In the second run, the data source authentication was configured to use per-user identity, and the SQL Server Analysis Services cube was configured to use dynamic security. In this configuration, PerformancePoint Services uses the identity of the user to request the data. Because the data results could be different, no caching can be shared across users. In certain cases, caching for per-user identity can be shared if Analysis Services dynamic security is not configured and the Analysis Services roles, to which Microsoft Windows users and groups are assigned, are identical.

Hardware setting and topology

Lab hardware

To provide a high level of test-result detail, several farm configurations were used for testing. Farm configurations ranged from one to three Web servers, one to four Application servers, and a single database server that was running Microsoft SQL Server 2008. A default enterprise installation of SharePoint Server 2010 was performed.

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

	Web server	**Application server**	**Computer that is running SQL Server**	**Computer that is running Analysis Services**
Processor(s)	2px4c @ 2.66 GHz	2px4c @ 2.66 GHz	2px4c @ 2.66 GHz	4px6c @ 2.4 GHz
RAM	16 GB	32 GB	16 GB	64 GB
Operating system	Windows Server 2008 R2 Enterprise	Windows Server 2008 R2 Enterprise	Windows Server 2008 R2 Enterprise	Windows Server 2008 R2 Enterprise
NIC	1x1 gigabit	1x1 gigabit	1x1 gigabit	1x1 gigabit
Authentication	NTLM and Kerberos	NTLM and Kerberos	NTLM and Kerberos	NTLM and Kerberos

After the farm was scaled out to multiple Web servers, a hardware load balancer was used to balance the user load across multiple Web servers by using *source-address affinity*. Source-address affinity records the source IP address of incoming requests and the service host that they were load-balanced to, and it channels all future transactions to the same host.

Topology

The starting topology consisted of two physical servers, with one server acting as the Web and application server and the second server as the database server. This starting topology is considered a two-machine (2M) topology or a "1 by 0 by 1" topology where the number of dedicated Web servers is listed first, followed by dedicated application servers, and then database servers.

Web servers are also known as web front ends (WFE) later in this document. Load was applied until limiting factors were encountered. Typically the CPU on either the Web or application server was the limiting factor, and then resources were added to address that limit. The limiting factors and topologies

differed significantly based on the data source authentication configuration of either the Unattended Service Account or per-user Identity with dynamic cube security.

Test results

The test results contain three important measures to help define PerformancePoint Services capacity.

Measure	Description
User count	Total user count reported by Visual Studio.
Requests per second (RPS)	Total RPS reported by Visual Studio, which includes all requests and a static file requests such as images and style sheets.
Views per second (VPS)	Total views that PerformancePoint Services can render. A view is any filter, scorecard, grid, or chart rendered by PerformancePoint Services or any Web request to the rendering service URL that contains **RenderWebPartContent** or **CreateReportHtml**. To learn more about **CreateReportHtml** and **RenderWebPartContent**, see the PerformancePoint Services RenderingService Protocol Specification (http://go.microsoft.com/fwlink/?LinkId=200609). IIS logs can be parsed for these requests to help plan the capacity of PerformancePoint Services. Also, using this measure provides a number that is much less dependent on dashboard composition. A dashboard with two views can be compared to a dashboard with 10 views.

Tip:
When you are using a data source configured to use Unattended Service Account authentication, the rule for the ratio of dedicated servers is one Web server to every **two** application servers that are running PerformancePoint Services.

Tip:

When you are using a data source configured to use per-user authentication, the rule for the ratio of dedicated servers is one Web server to every **four** or more application servers that are running PerformancePoint Services.

At topologies larger than four application servers, it is likely that the bottleneck is the Analysis Services server. Consider monitoring the CPU and query time of your Analysis Services server to determine whether you should scale out Analysis Services to multiple servers. Any delay in query time on the Analysis Services server will significantly increase the average response time of PerformancePoint Services beyond the desired threshold of two seconds.

The tables that follow show a summary of the test results for both Unattended Service Account authentication and per-user authentication when scaling out from two to seven servers. Detailed results that include additional performance counters are included later in this document.

Unattended Service Account authentication summary

Topology (WFE x APP x SQL)	Users	Requests per second (RPS)	Views per sec (VPS)
2M (1x0x1)	360	83	50
3M (1x1x1)	540	127	75
4M (1x2x1)	840	196	117
5M (1x3x1)	950	215	129
6M (2x3x1)	1,250	292	175
7M (2x4x1)	1,500	346	205

Per-user authentication summary

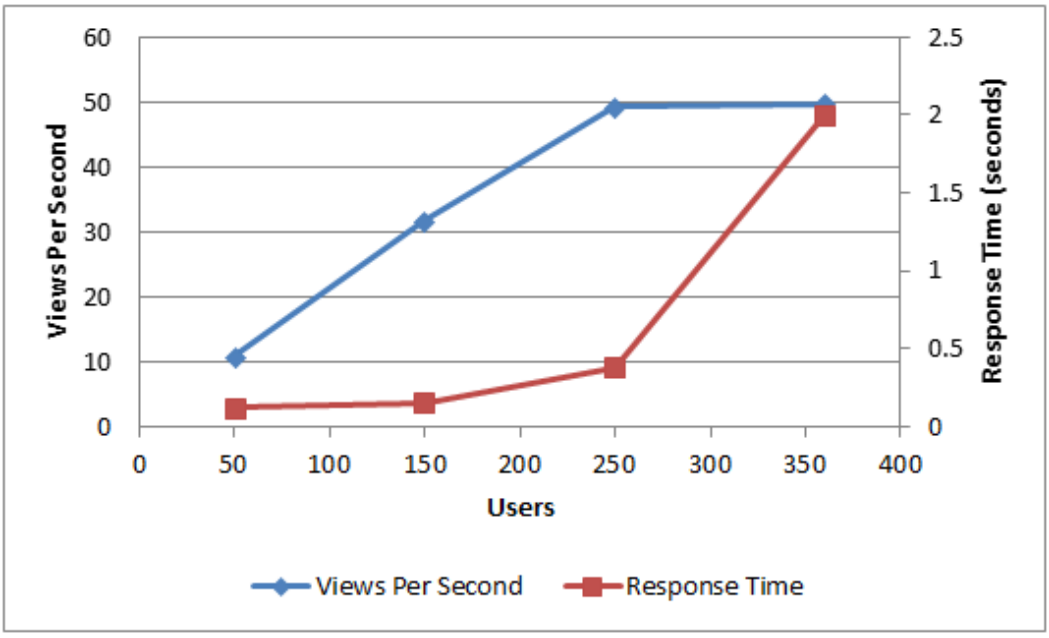
Topology (WFE x APP x SQL)	Users	Requests per second (RPS)	Views per sec (VPS)
2M (1x0x1)	200	47	27
3M (1x1x1)	240	56	33
4M (1x2x1)	300	67	40
5M (1x3x1)	325	74	44

2M and 3M topologies

To help explain the hardware cost per transaction and the response time curve, the load tests were run with four increasing user loads to the maximum user load for the 2M and 3M topologies.

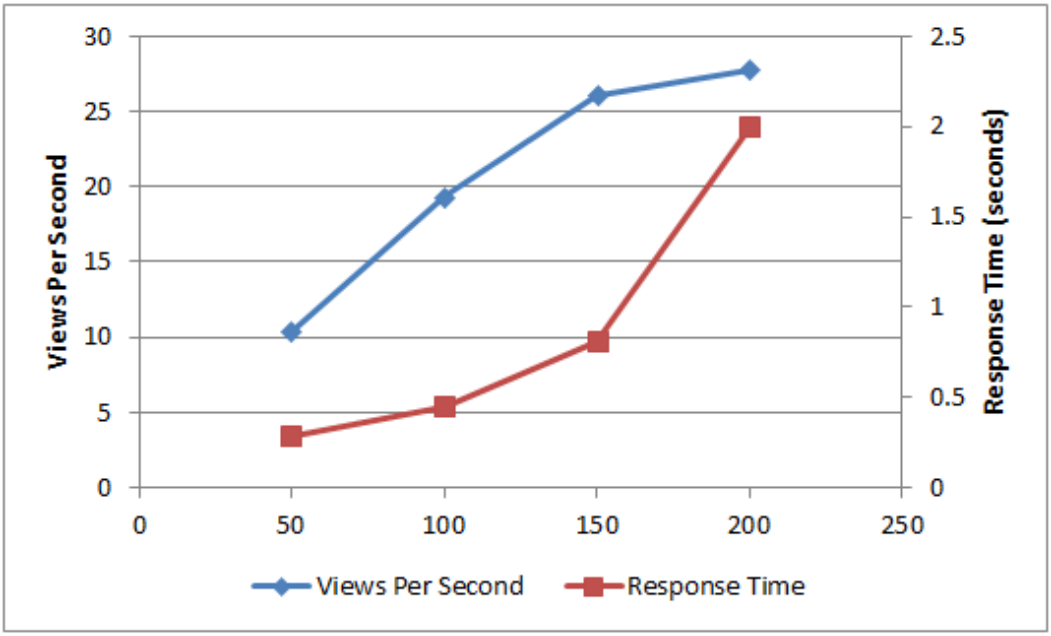
Unattended Service Account authentication

User count	50	150	250	360
Average WFE/APP CPU	19.20%	57.70%	94.00%	96.70%
RPS	18	53	83	83
Views per second	10.73	31.72	49.27	49.67
Average response time (sec)	0.12	0.15	0.38	2



Per-user authentication

User count	50	100	150	200
Average WFE/APP CPU	30.80%	61.30%	86.50%	93.30%
RPS	17	32	43	47
Views per second	10.3	19.32	26.04	27.75
Average response time (sec)	0.28	0.45	0.81	2

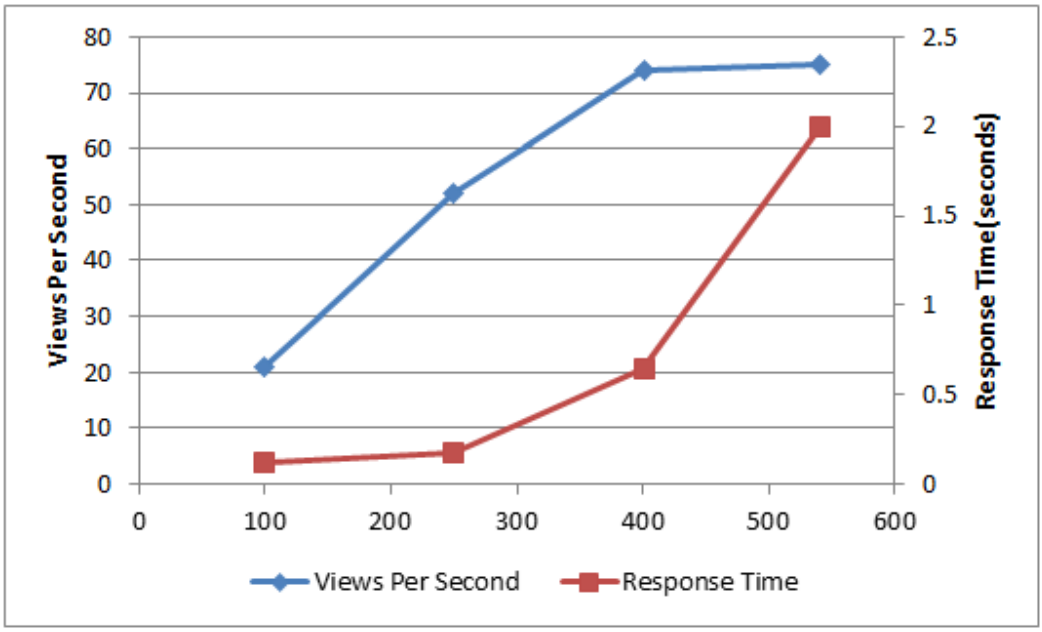


3M (1x1x1) farm results

Unattended Service Account authentication

User count	100	250	400	540
RPS	36	87	124	127
Views per second	21	52	74	75
Average response time (sec)	0.12	0.18	0.65	2

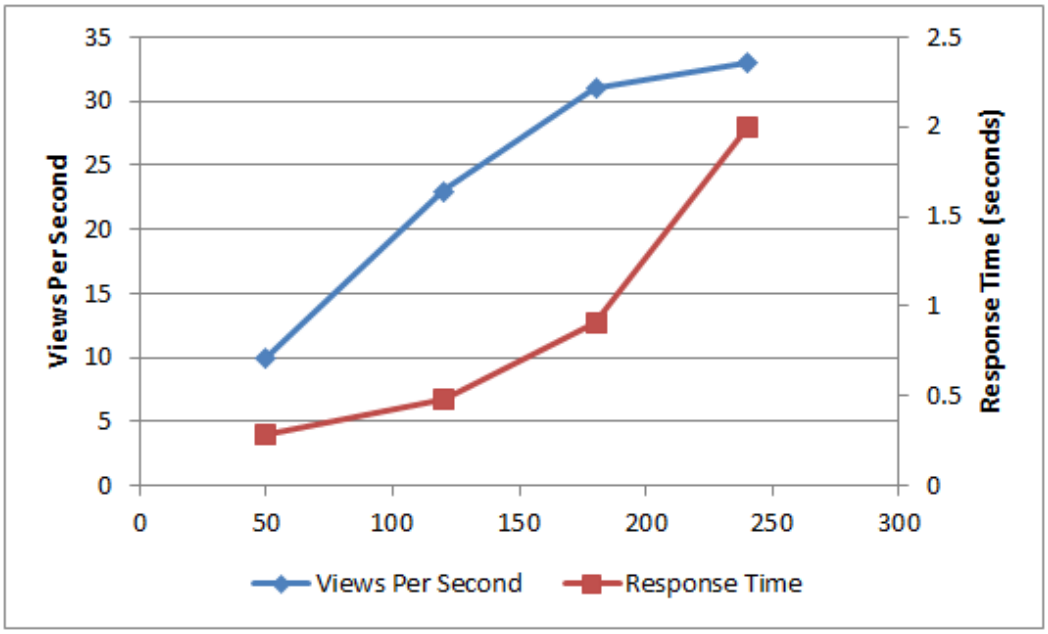
User count	100	250	400	540
Average WFE CPU	11%	28%	43%	46%
Max WFE private bytes of SharePoint Server Internet Information Services (IIS) worker process W3WP.	0.7 GB	1.4 GB	2.0 GB	2.4 GB
Average APP CPU	25%	62%	94%	95%
Max APP private bytes of PerformancePoint Services W3WP	5.9 GB10.8 GB	10.8 GB	14.1 GB	14.6 GB



Per-user authentication

User count	50	120	180	240
RPS	17	39	52	56
Views per second	10	23	31	33

User count	50	120	180	240
Average response time (sec)	0.28	0.48	0.91	2
Average WFE CPU	5%	12%	17%	19%
Max WFE private bytes of SharePoint Server W3WP	0.78 GB	1.3 GB	1.6 GB	1.9 GB
Average APP CPU	25%	57%	81%	81%
Max APP private bytes of PerformancePoint Services W3WP	19 GB	20.1 GB	20.5 GB	20.9 GB



4M+ results for Unattended Service Account authentication

Starting with a 4M topology, load was applied to produce a two-second average response time to render a scorecard or report. Next, an additional server was added to resolve the limiting factor (always CPU on the Web server or the application server) and then the test mix was re-run. This logic was repeated until a total of seven servers was reached.

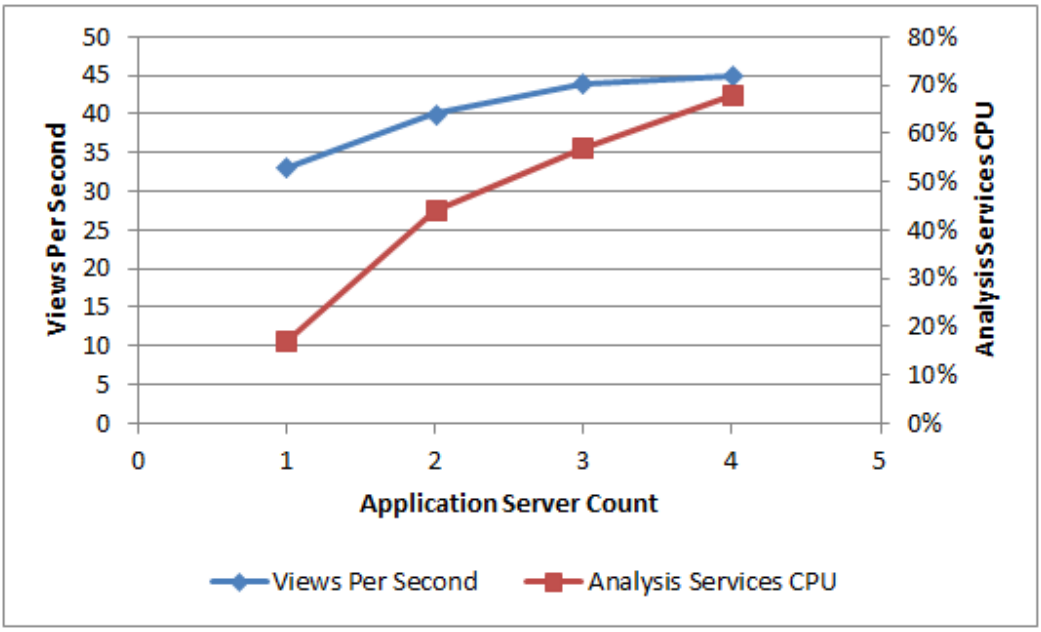
	4M (1x2x1)	5M (1x3x1)	6M (2x3x1)	7M (2x4x1)
User count	840	950	1,250	1,500
RPS	196	216	292	346
Views per second	117	131	175	206
Average. WFE CPU	77%	63%	54%	73%
Max WFE private bytes of SharePoint Server W3WP	2.1 GB	1.7 GB	2.1 GB	2.0 GB
Average APP CPU	83%	94%	88%	80%
Max APP private bytes of PerformancePoint Services W3WP	16 GB	12 GB	15 GB	15 GB

4M+ Results for per-user authentication

The same testing was repeated for a data source configured for per-user authentication. Note that adding an application server to create a four-application server topology did not increase the number of users or requests per second that could be supported by PerformancePoint Services because of the query delays that Analysis Services produced.

	3M (1x1x1)	4M (1x2x1)	5M (1x3x1)	6M (1x4x1)
User count	240	300	325	325
RPS	56	67	74	74
Views per second	33	40	44	45
Average. WFE CPU	19%	24%	26%	12%
Max WFE private bytes of SharePoint Server W3WP	2.1 GB	1.9 GB	1.9 GB	1.5 GB
Average APP CPU	89%	68%	53%	53%
Max APP private bytes of PerformancePoint	20 GB	20 GB	20 GB	20 GB

	3M (1x1x1)	4M (1x2x1)	5M (1x3x1)	6M (1x4x1)
Services W3WP				
Analysis Services CPU	17%	44%	57%	68%



Recommendations

Hardware recommendations

The memory and processor counters from the test tables should be used to determine the hardware requirements for an installation of PerformancePoint Services. For Web servers, PerformancePoint Services uses the recommended SharePoint Server 2010 hardware requirements. Application server hardware requirements may have to be changed when PerformancePoint Services consumes a large amount of memory. This happens when data sources are configured to per-user authentication or when the application server runs many dashboards with long data source timeouts.

The database server did not become a bottleneck in the tests and peaked at a maximum CPU usage of 31% under the 7M Unattended Service Account authenticated dashboard. The PerformancePoint Services content definitions such as reports, scorecards, and KPIs are stored in SharePoint lists and are cached in memory by PerformancePoint Services, reducing the load on the database server.

Memory consumption

PerformancePoint Services can consume large amounts of memory in certain configurations, and it is important to monitor memory usage of the PerformancePoint Services application pool. PerformancePoint Services caches several items in memory, including Analysis Services and other data-source query results for the data source cache lifetime (a default of 10 minutes). When you are using a data source that is configured for Unattended Service Account authentication, these query results are only stored once and shared across multiple users. However, when you are using a data source that is configured for per-user authentication and Analysis Services dynamic cube security, the query results are stored once per user per view (that is, a "per filter" combination).

The underlying cache API that PerformancePoint Services uses is the ASP.NET Cache API. The significant advantage of using this API is that ASP.NET manages the cache and removes items (also known as a trim) based on memory limits to prevent out-of-memory errors. The default memory limit is 60 percent of physical memory. After reaching these limits, PerformancePoint Services still rendered views but response times increased significantly during the short period when ASP.NET removed cached entries.

The performance counter "ASP.NET Applications \ Cache API Trims" of the application pool hosting PerformancePoint Services can be used to monitor the ASP.NET cache trims that occur because of memory pressure. If this counter is greater than zero, then review the following table for possible solutions.

Problem	**Solution**
Application server processor usage is low and other services are running on the application server.	Add more physical memory or limit the memory of the ASP.NET cache.
Application server processor usage is low and only PerformancePoint Services is running on the application server.	If acceptable, configure the ASP.NET cache settings to have the cache use more memory, or add more memory.
Application server processor usage is high.	Add another application server.

A data source configured to use per-user authentication can share query results and cache entries if the Analysis Services role membership sets of the users are identical and if dynamic cube security is not configured. This is a new feature for PerformancePoint Services in Microsoft SharePoint Server 2010. For example, if user A is in role 1 and 2, and user B is in Role 1 and 2, and user C is in Role 1 and 2 and 3, only user A and user B share cache entries. If there is dynamic cube security, users A and B and also user C do not share cache entries.

Analysis Services

When PerformancePoint Services was being tested with per-user authentication, two Analysis Services properties were changed to improve multiple-user throughput performance. The following table shows the properties that were changed and the new value of each property.

Analysis Services property	Value
Memory \ HeapTypeForObjects	0
Memory \ MemoryHeapType	2

These two memory settings configure Analysis Services to use the Windows heap instead of the Analysis Services heap. Before changing these properties and while adding user load, response times increased significantly from 0.2 seconds to over 30 seconds while the CPU on the Web, application, and Analysis Services servers remained low. To troubleshoot, query time was collected by using Analysis Services dynamic management views (DMV), which showed an increase of individual query times from 10 milliseconds to 5000 milliseconds. These results led to modifying the above memory settings.

It is important to note that while this greatly improved throughput, according to the Analysis Services team, changing these settings has a small but measurable cost on single-user queries.

Before changing any Analysis Services properties, consult the SQL Server 2008 White Paper: Analysis Services Performance Guide (http://go.microsoft.com/fwlink/?LinkID=165486) for best practices on improving multiple-user throughput performance.

Common bottlenecks and their causes

During performance testing, several 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. If high processor utilization was encountered as a bottleneck, additional servers were added to resolve the bottleneck. The following table lists some common bottlenecks and possible resolutions assuming processor utilization was low and not the bottleneck.

Possible bottleneck	Cause and what to monitor	Resolution
Analysis Services memory heap performance	By default, Analysis Services uses its own memory heap instead of the Windows heap, which provides poor multi-user	Change Analysis Services to use the Windows heap. See the "Analysis Services" section earlier in this article and the SQL Server 2008 White Paper: Analysis Services Performance Guide for instructions

Possible bottleneck	Cause and what to monitor	Resolution
	throughput performance. Review the Analysis Services query times using dynamic management views (DMV) to see if query times increase with user load and Analysis Services processor utilization is low.	(http://go.microsoft.com/fwlink/?LinkID=165486).
Analysis Services query and processing threads	By default, Analysis Services limits the number of query and processing threads for queries. Long running queries and high user loads could use all available threads. Monitor the idle threads and job queue performance counters under the MSAS 2008:Threads category.	Increase the number of threads available to query and process. See Analysis Services section and the SQL Server 2008 White Paper: Analysis Services Performance Guide for instructions (http://go.microsoft.com/fwlink/?LinkID=165486).
Application server memory	PerformancePoint Services caches the Analysis Services and other data source query results in memory for the data source cache lifetime. These items can consume a large amount of memory. Monitor the ASP.NET Applications \ Cache API Trims of the PerformancePoint Services application pool to determine whether cache removals or trims are being forced by ASP.NET because of low memory.	Add memory or increase the default ASP.NET cache memory limits. See Memory Consumption section earlier in this document for additional discussion. Also, see the ASP.NET cache element settings (http://go.microsoft.com/fwlink/?LinkId=200610) and Thomas Marquardt's blog post on Some history on the ASP.NET cache memory limits (http://go.microsoft.com/fwlink/?LinkId=200611).
WCF throttling settings	PerformancePoint Services is implemented as a WCF	If needed, change the Windows Communication Foundation (WCF) throttling behavior. See the

Possible bottleneck	Cause and what to monitor	Resolution
	service. WCF limits the maximum number of concurrent calls as a service throttling behavior. Although long-running queries could hit this bottleneck, this is an uncommon bottleneck. Monitor the WCF / Service Model performance counter calls outstanding for PerformancePoint Services and compare to the current maximum number of concurrent calls.	WCF service throttling behaviors (http://go.microsoft.com/fwlink/?LinkId=200612) and Wenlong Dong's blog post on WCF Request Throttling and Server Scalability (http://go.microsoft.com/fwlink/?LinkId=200613).

Performance monitoring

To help you determine when you have to scale up or scale out the system, use performance counters to monitor the health of the system. PerformancePoint Services is an ASP.NET WCF service and can be monitored by using the same performance counters used to monitor any other ASP.NET WCF service. In addition, use the information in the following tables to determine supplementary performance counters to monitor, and to which process the performance counters should be applied.

Performance counter	Counter Instance	Notes
ASP.NET Applications / Cache API Trims	PerformancePoint Services application pool	If the value is greater than zero, review the "Memory consumption".
MSAS 2008:Threads / Query pool idle threads	N/A	If the value is zero, review the "Analysis Services" section and SQL Server 2008 White Paper: Analysis Services Performance Guide (http://go.microsoft.com/fwlink/?LinkID=165486).
MSAS 2008:Threads / Query pool job queue length	N/A	If the value is greater than zero, review the "Analysis Services" section and SQL Server 2008 White Paper: Analysis Services Performance Guide (http://go.microsoft.com/fwlink/?LinkID=165486).

Performance counter	Counter Instance	Notes
MSAS 2008:Threads / Processing pool idle threads	N/A	If the value is greater than zero, review the "Analysis Services" section and SQL Server 2008 White Paper: Analysis Services Performance Guide (http://go.microsoft.com/fwlink/?LinkID=165486).
MSAS 2008:Threads / Processing pool job queue length	N/A	If the value is greater than zero, review the "Analysis Services" section and SQL Server 2008 White Paper: Analysis Services Performance Guide (http://go.microsoft.com/fwlink/?LinkID=165486).
WCF CountersServiceModelService 3.0.0.0(*)\Calls Outstanding	PerformancePoint Service Instance	If the value is greater than zero, see WCF Request Throttling and Server Scalability (http://go.microsoft.com/fwlink/?LinkID=200613).

See Also

Plan for PerformancePoint Services (SharePoint Server 2010)

Capacity requirements for Web Analytics Shared Service in SharePoint Server 2010

By using prerelease versions of Microsoft SharePoint Server 2010 and other applications, capacity testing was performed for a simulated midsized deployment that included 30,000 SharePoint entities. This article describes the results of the capacity testing activities and contains guidance on capacity management for the Web Analytics service application in SharePoint Server 2010.

In SharePoint Server 2010, the Web Analytics service application enables you to collect, report, and analyze the usage and effectiveness of SharePoint Server 2010 sites. Web Analytics features include reporting, Web Analytics workflow, and Web Analytics Web Part. For more information, see Reporting and usage analysis overview.

The aspects of capacity planning that are described in this article include the following:

- Description of the architecture and topology.
- Capacity planning guidelines based on the key factors such as total expected traffic and number of SharePoint components.
- Description of the other factors that affect the performance and capacity requirements.

Before you continue to read this article, make sure that you understand key concepts related to SharePoint Server 2010 capacity management. The resources that are listed in this section can help you learn about frequently used terms and get an overview of the recommended approach to capacity management. These resources can also help you use the information that is provided in this article more effectively.

For more conceptual information about performance and capacity management, see the following articles:

- Performance and capacity management (SharePoint Server 2010)
- Capacity management and sizing for SharePoint Server 2010

In this article:

Introduction

Overview

As part of SharePoint Server 2010, the Web Analytics service application is a set of features that you can use to collect, report, and analyze the usage and effectiveness of a SharePoint Server 2010 deployment. You can organize SharePoint Web Analytics reports into three main categories:

- Traffic
- Search
- Inventory

SharePoint Web Analytics reports are typically aggregated for various SharePoint entities, such as sites, site collections, and Web applications for each farm. To view an architectural overview of the Web Analytics service application in a SharePoint deployment, see Architectural overview later in this article.

The Web Analytics shared service requires resources primarily at the application server and database server level. This article does not cover the Web Server layer capacity planning, because the Web Analytics service's capacity requirements are minimal at this level.

This article contains the capacity requirements for several application servers and Microsoft SQL Server–based computers, based on the following criteria:

- Total expected site traffic (clicks, search queries, ratings).
- Number of SharePoint components (Site, Site Collection, and Web Application) for each farm.

Other less significant factors which can affect the capacity requirements are summarized in Other factors later in this article.

Architectural overview

The following diagram (Figure 1) shows the flow of the site usage data from a Web browser to the analytics databases, and then back to the Web browser as reports. The usage data is logged to the usage files on the Web servers. The usage timer job calls the Logging Web Service to submit the raw data from the usage files. The Logging Web Service writes it to the staging database, where the raw data is stored for seven days (this is not configurable). The Web Analytics components Log Batcher and User Behavior Analyzer clean and process the raw data on the staging database. The Report Consolidator runs one time every 24 hours. The Report Consolidator aggregates the raw data from the staging database on various dimensions, and then writes it to the reporting database. The aggregated data is stored in the reporting database for a default period of 25 months (this is configurable).

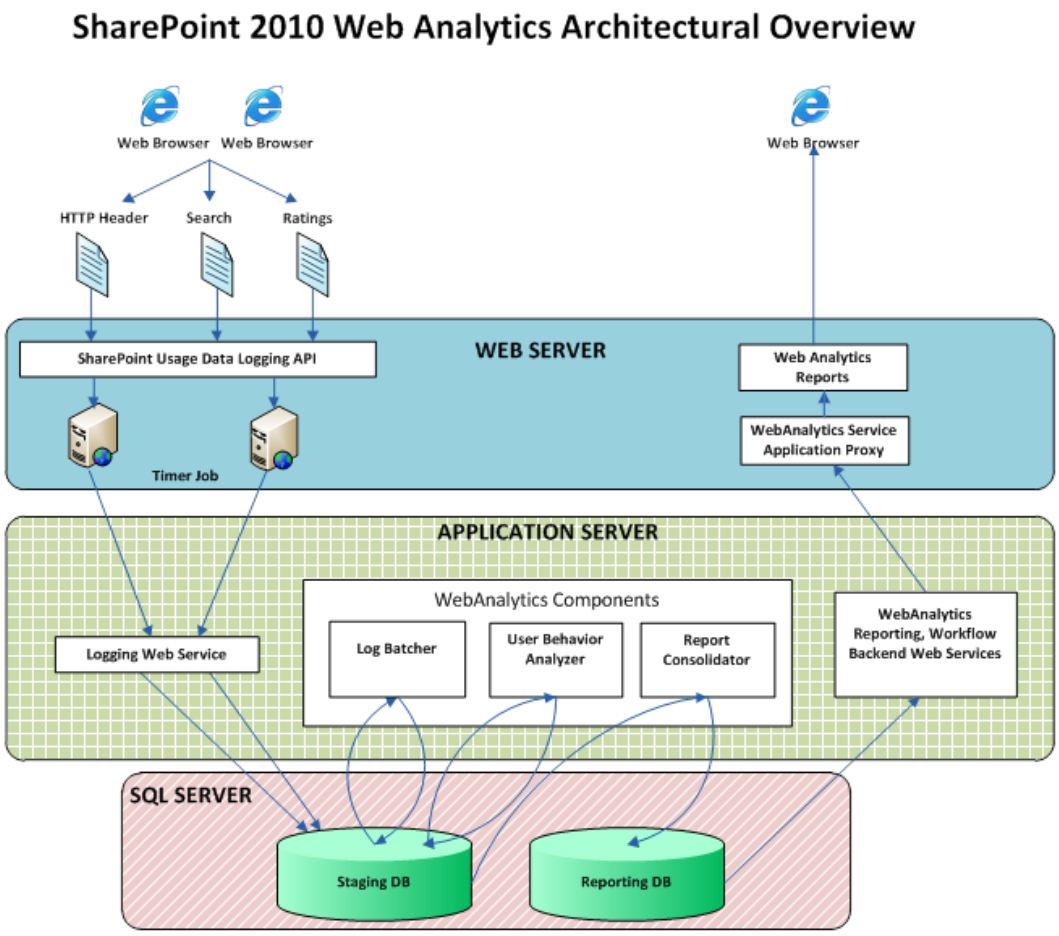


Figure 1. SharePoint Server 2010 Web Analytics architectural overview

The performance of the Logging Web Service primarily depends on the number of application servers. (Scaling out is available for the application servers.) The performance of the Log Batcher and User Behavior Analyzer depends primarily on the analytics staging database. The Read and Write activities that are performed by all the different components can cause the analytics staging database to slow down the process. (Scaling out is available for the staging database.) The performance of the Report Consolidator also primarily depends on the reporting database. (Scaling out of reporting database is not supported.)

Note:
The same server that is running SQL Server can be used to deploy both the analytics staging database and the reporting database together with the other SharePoint databases.

Hardware specifications and topology

This section provides detailed information about the hardware, software, topology, and configuration of a case study environment.

Hardware

Note:
This environment is scaled to accommodate prerelease builds of SharePoint Server 2010 and other products. Therefore, the deployed hardware has larger capacity than necessary to serve the demand typically experienced by this environment. This hardware is described only to provide additional context for this environment and serve as a starting point for similar environments. It is important to conduct your own capacity management based on your planned workload and usage characteristics. For more information about the capacity management process, see Performance and capacity management (SharePoint Server 2010).

Web servers

This article does not cover the Web server layer capacity planning, because the Web Analytic service's capacity requirements are minimal at this level.

Application servers

The following table describes the configuration of each application server. Based on the site traffic and the number of SharePoint components that are involved, users will need one or more application servers.

Application server	Minimum requirement
Processors	4 quad core @ 2.33 GHz
RAM	8 GB
Operating system	Windows Server 2008, 64 bit
Size of the SharePoint drive	300 GB
Number of network adapters	1
Network adapter speed	1 GB
Authentication	NTLM
Load balancer type	SharePoint Load Balancer
Software version	SharePoint Server 2010 (prerelease version)
Services running locally	• Central Administration • Microsoft SharePoint Foundation Incoming E-

Application server	Minimum requirement
	mail • Microsoft SharePoint Foundation Web Application • Microsoft SharePoint Foundation Workflow Timer Service • Search Query and Site Settings Service • SharePoint Server Search • Web Analytics Data Processing Service • Web Analytics Web Service

Database servers

The following table describes the configuration of each database server. Instances of SQL Server were required for both the staging and reporting databases.

Database server	Minimum requirement
Processors	4 quad core @ 2.4 GHz
RAM	32 GB
Operating system	Windows Server 2008, 64-bit
Disk size	3 terabytes
Number of network adapters	1
Network adapter speed	1 GB
Authentication	NTLM
Software version	SQL Server 2008

Topology

The following diagram (Figure 2) shows the Web Analytics topology.

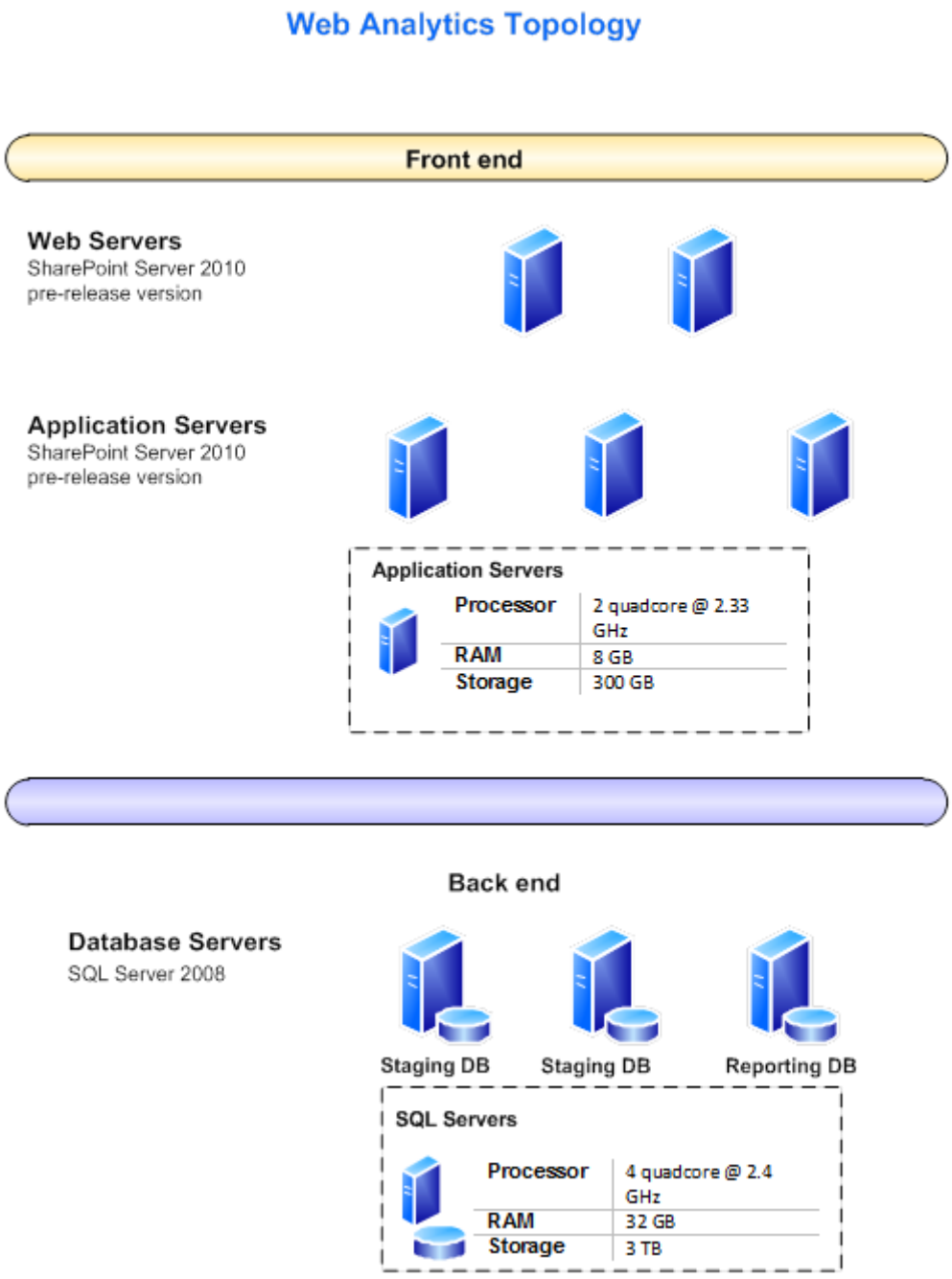


Figure 2. Web Analytics topology

Capacity requirements

Testing methodology

This section presents the capacity requirements with regard to the total amount of site traffic (this is measured by number of clicks, search queries, and ratings) per day that can be supported by different numbers of application servers and SQL Server–based computers. The numbers presented currently are for a midsize SharePoint deployment that has about 30,000 SharePoint entities. The Web Analytics shared service aggregates the data for each day. Therefore, the data volume that is presented corresponds to the total number of records (this is measured by number of clicks, search queries, and ratings) that the SharePoint farm is expected to receive each day.

This section provides diagrams that show the daily site traffic that can be supported by one, two, or three application servers (Figure 3) and the daily site traffic that can be supported that corresponds to the various database configurations (Figure 4). In the diagrams, data is shown by using two colors:

- **Green** Green values indicate the safe limit for the site traffic that can be processed for the corresponding number of application servers and SQL Server–based computers.
- **Yellow** Yellow values indicate the expected limit for the site traffic that can be processed for the corresponding number of application servers and SQL Server–based computers.

The green and yellow values are estimates that are based on two key factors:

- Total site traffic, measured by number of page view clicks, search queries, and ratings.
- Number of SharePoint entities, such as sites, site collections, and Web applications, for each farm.

The estimates also depend on other properties of the data and the data retention period in the reporting database. For testing, the other properties of the data were maintained as constant as described in Dataset description later in this section.

Also, in smaller SharePoint deployment environments, you can share the application servers and SQL Server–based computers together with other SharePoint services and databases. This article contains information about the capacity of the application servers and the SQL Server–based computers that are in a test environment so that the Web Analytics shared service is the only major service that is running on the servers. The actual performance results for environments that actively use other shared services at the same time running might vary.

To determine the capacity requirements for your environment, make sure that you estimate the expected daily site traffic and the number of components that you might use for a SharePoint deployment. Then, the number of application servers and SQL Server–based computers should be estimated independently, as shown in Figure 3 and Figure 4.

Dataset description

The dataset that was selected for the test environment is a mid-sized dataset that has approximately 30,000 SharePoint components. Other characteristics of the data that were kept constant in the environment are also listed in the following table.

Dataset characteristics	Value
Number of SharePoint components	28,967
Number of unique users	117,000
Number of unique queries	68,000
Number of unique assets	500,000
Data size in the reporting database	200 GB

The total site traffic, measured by number of clicks, search queries, and ratings, was increased as part of this case study to establish the number of records that can be supported by the corresponding topology.

Application servers

The following diagram (Figure 3) shows the daily site traffic that can be supported by one, two, or three application servers. The site traffic is represented in millions of records (each click, search query, or rating makes up a record) each day. The yellow line represents the expected number of records for the corresponding topology, whereas the green line represents the safe assumption for the number of records.

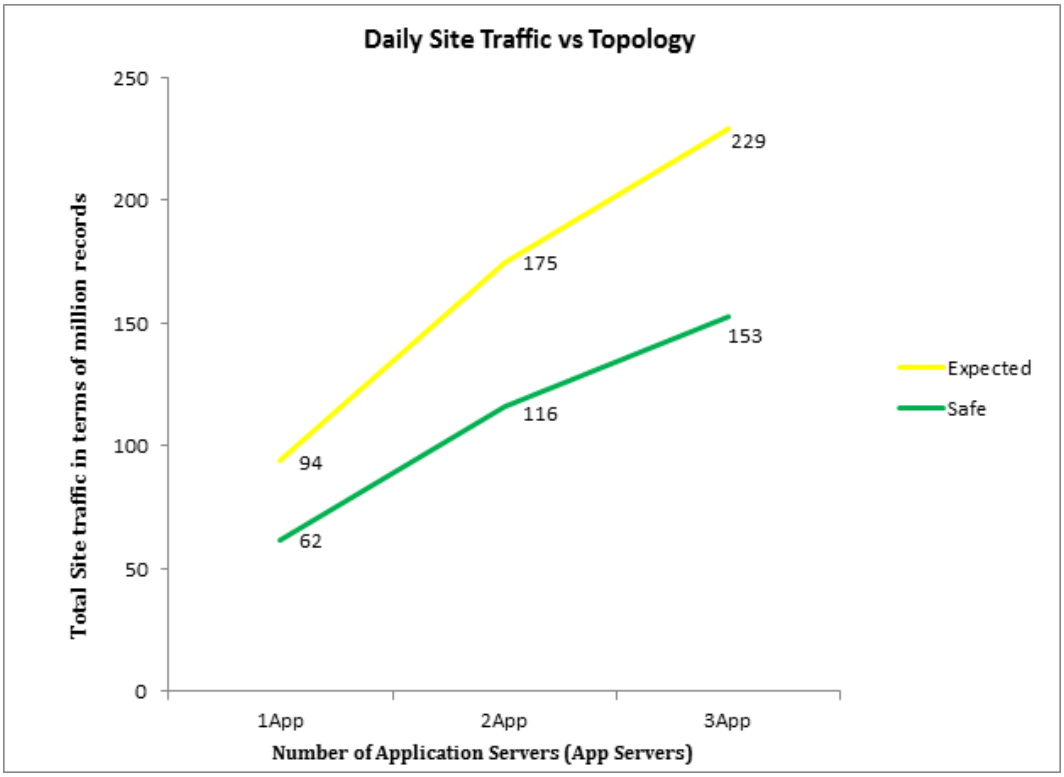


Figure 3. Daily site traffic vs. the application servers topology

The application servers are not very CPU-intensive or memory intensive. Thus, the CPU and the memory usage are not summarized for this section.

SQL Server–based computers

The following diagram (Figure 4) shows the daily site traffic that can be supported that corresponds to the following configurations:

- One instance of SQL Server for both staging and reporting databases (1S+R).
- Two instances of SQL Server, one staging database and one reporting database (1S1R).
- Three instances of SQL Server, two staging databases and one reporting database (2S1R).

The site traffic is represented in millions of records (each click, search, or rating makes up a record) each day. The yellow line represents the expected number of records for the corresponding topology, whereas the green line represents the safe assumption for the number of records.

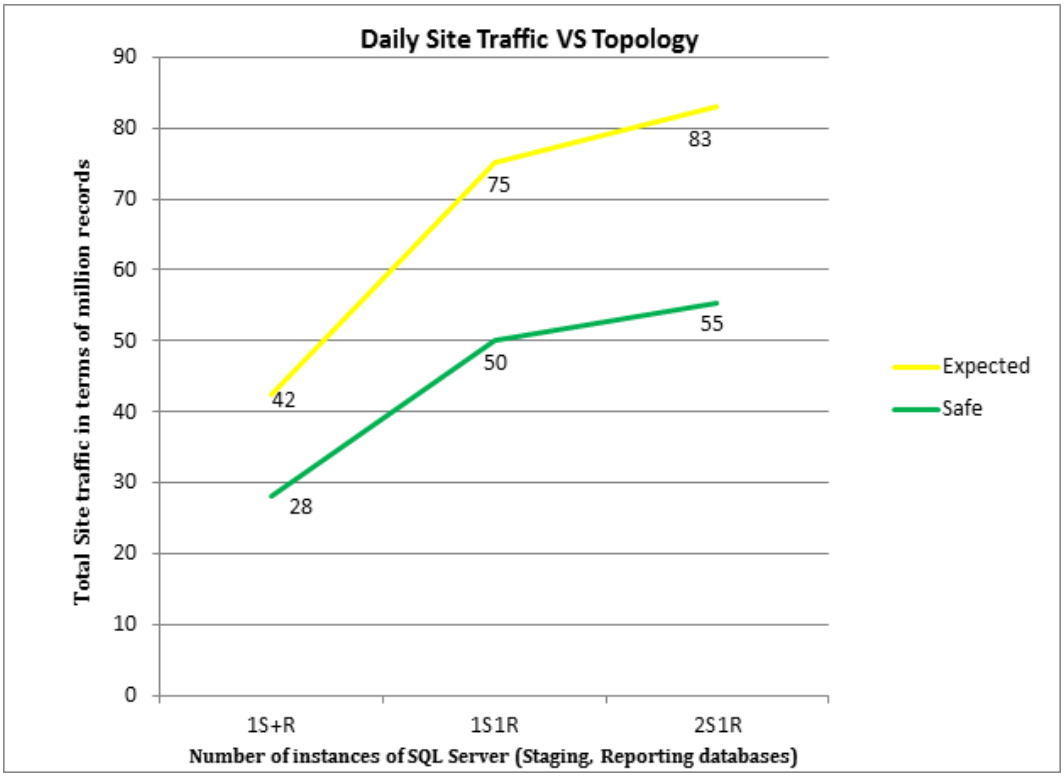


Figure 4. Daily site traffic vs. SQL Server topology

The following table summarizes the CPU and memory usage of the various components on the instances of SQL Server that are hosting the staging database and the reporting database.

Configuration	1S+R	1S1R	1S1R	2S1R	2S1R
	Staging + Reporting	Staging	Reporting	Staging	Reporting
Total sum of percentage of processor time for 8 processor computer	19	192	5.78	100	13.4
SQL Server buffer hit ratio	99	100	100	100	100
% Disk time	7,142	535	5.28	59.3	98.2
Disk queue length	357	28.6	0.26	2.97	4.91

Other factors

Many other factors can affect the performance of various analytics components and can affect the capacity planning. These factors primarily affect the performance of the Report Extractor component because they can affect the size of the data aggregated each day. The total size of the data in the reporting database also affects the performance of the Reporting Extractor, although this is not significant because the data is partitioned daily. Some of these other factors are as follows:

- Number of unique queries each day.
- Number of unique users each day.
- Total number of unique assets clicked each day.
- Existing data size in the reporting warehouse, based on the data retention in the warehouse.

The overall effect of these factors is less significant than the total data volume and the number of site entities. However, it is important to conduct your own capacity management based on your planned workload and usage characteristics. For more information about the capacity management process, see Performance and capacity management (SharePoint Server 2010).

Remaining issues

There are current known issues that significantly affect the current performance of the Web Analytics service application for deployments that have a large site hierarchy, which includes approximately 100,000 or more SharePoint components. This article might be updated with the capacity requirements for larger site hierarchies when more information is available.

See Also

Performance and capacity management (SharePoint Server 2010)

SharePoint 2010 Administration Toolkit (SharePoint Server 2010)

Estimate performance and capacity requirements for Web Content Management in SharePoint Server 2010

This article contains guidance on capacity management that is relevant to Microsoft SharePoint Server 2010 sites that have the Publishing Infrastructure enabled. This document is specific to SharePoint Server 2010, and the information that is discussed does not apply to SharePoint Foundation.

This article discusses the following scenarios:

- An Internet publishing site - a corporate presence site.

  This kind of site is published to the Internet and lets anonymous Internet users find information about a corporation. Sites such as these are branded and the content is tightly controlled.
- An intranet publishing site - an internal news site.

  This kind of site is published internally inside an organization. Its primary use is to share information with the authenticated users inside the organization. Information in the site might be managed tightly, or some areas might be less managed.
- An enterprise wiki - a knowledge repository.

  An enterprise wiki is a single-farm site that grows organically as contributors create new pages and link them to other pages that might or might not exist yet. Enterprise wikis are typically published internally inside an organization. This site enables people across a company or organization to capture and share knowledge by using a solution that is integrated into and enhanced by their SharePoint environment.

After reading this document, you will understand the following concepts:

- The key metric (throughput) that you should maximize to support lots of read operations.
- Various potential bottlenecks that are relevant to a Web Content Management SharePoint Server 2010 deployment.
- The importance of the output cache in maximizing throughput.
- The effect of write operations on the end-user read experience.

In this article:

Prerequisite information

Before you read this document, make sure that you understand the key concepts behind SharePoint Server 2010 capacity management. The following documentation will help you learn about the recommended approach to capacity management and provide context for helping you understand how to make effective use of the information in this document.

For more conceptual information about performance and capacity that you might find valuable in understanding the context of the data in this article, see the following documents:

- Capacity management and sizing for SharePoint Server 2010
- Performance and capacity technical case studies (SharePoint Server 2010)

Test details and approach

In each test, variables that might be present in the real world have been abstracted to show specific recommendations. Therefore, it is very important to test and monitor in your own environment to make sure that you have scaled correctly to meet the request volume that you expect. To learn more about capacity management concepts, you can refer to Capacity management and sizing overview for SharePoint Server 2010.

This article discusses performance with Site Collection Features, SharePoint Server Publishing Infrastructure, and Output caching. These features are available only when the SharePoint Server Publishing Infrastructure is enabled. By default, Publishing Portals have this feature enabled.

Dataset

The tests were conducted by using a dataset that shares common characteristics with actual Web Content Management deployments. Although load was constant, different pages were requested. The following table describes the dataset that was used for these tests.

Dataset

Object	Publishing site
Size of content databases	2.63 GB
Number of content databases	1
Number of site collections	1
Number of Web applications	1
Number of sites	50
Number of pages	20,000 pages, divided into 20 folders that have 1,000 pages each
Composition of pages	Article pages in basic HTML, with references to

Object	Publishing site
	two images
Page size	42 KB uncompressed; 12 KB compressed
Images	3,000 at 30 KB to 1.3 MB each

We recommend configuring Internet Information Services (IIS) to always compress files instead of the default setting to dynamically compress files. When dynamic compression is enabled, IIS compresses pages until CPU utilization exceeds a certain threshold, at which point IIS ceases to compress pages until utilization drops under the threshold. The tests in this article were conducted with compression always on.

This test dataset used only default SharePoint Server 2010 features that are included with the product. Your site probably includes customizations in addition to these basic features. Therefore, it is important to test the performance of your own solution.

Hardware

The number of Web servers in the farm varied by test. But each had identical hardware. The following table describes the Web and application server hardware that was used during these tests.

Hardware specifications for application servers and Web servers

	Web server
Processors	2 quad core at 2.33 GHz
RAM	8 GB
Operating system	Windows Server 2008, 64 bit
Size of the SharePoint drive	300 GB
Number of network adapters	2
Network adapter speed	1 gigabit
Authentication	Windows Basic
Load balancer type	Hardware load balancing
Software version	SharePoint Server 2010 (pre-release version)
Services running locally	Central Administration Microsoft SharePoint Foundation Incoming E-Mail Microsoft SharePoint Foundation Web Application Microsoft SharePoint Foundation Workflow Timer

	Web server
	Service

The following table describes the database server hardware that was used during these tests.

Hardware specifications for database servers

	Database server
Processors	4 quad core at 3.19 GHz
RAM	16 GB
Operating system	Windows Server 2008, 64 bit
Storage	15 disks of 300 GB @ 15,000 RPM
Number of network adapters	2
Network adapter speed	1 gigabit
Authentication	NTLM
Software version	Microsoft SQL Server 2008

Glossary

There are some specialized terms that you will encounter in this document. Here are some key terms and their definitions:

- **RPS** Requests per second. The number of requests received by a farm or server in one second. This is a common measurement of server and farm load.

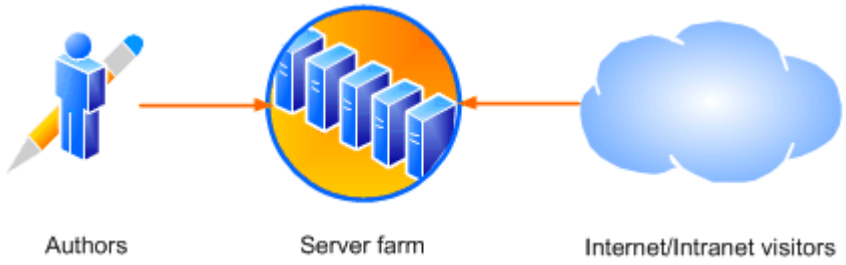
  Note that requests differ from page loads; each page contains several components, each of which creates one or more requests when the page is loaded. Therefore, one page load creates several requests. Typically, authentication checks and events that use insignificant resources are not counted in RPS measurements.
- **Green Zone** This is the state at which the server can maintain the following set of criteria:
  - The server-side latency for at least 75 percent of the requests is less than 1 second.
  - All servers have a CPU utilization of less than 50 percent.
  - Failure rate is less than 0.01 percent.

Web Content Management deployments

There are two models by which content is authored in SharePoint publishing sites that can affect your choice of server farm topology.

In the **author-in-place** model, a single site collection is shared by authors and visitors. Authors can create and update content at any time, which leads to variable distributions of read and write operations throughout a given day. This server farm typically experiences lots of reads and a moderate number of writes.

The following diagram shows how authoring-in-place works from a topology perspective.



In the **content deployment** model, multiple site collections separately and exclusively support content authoring and publishing. Content is created and updated in the authoring environment and then deployed to the publishing environment on a scheduled basis to be read by visitors. The publishing environment primarily serves read requests except when content is being deployed from the authoring environment. Unlike the author-in-place model, the server load that is exerted by content deployment can be adjusted to scheduled intervals.

The following diagram shows how content deployment works from a topology perspective.

These content authoring models are mutually exclusive.

Although Internet publishing sites and intranet publishing sites can use either the author-in-place model or the content deployment model, enterprise wikis work best with the author-in-place model. An enterprise wiki typically experiences a larger volume of write operations relative to read operations because a larger proportion of users can edit pages. Enterprise wiki pages differ from publishing article pages and exhibit different performance characteristics.

What to optimize

This section discusses information for optimizing your Web Content Management environment. Optimizing the environment includes understanding how to manage throughput, bottlenecks, and caching.

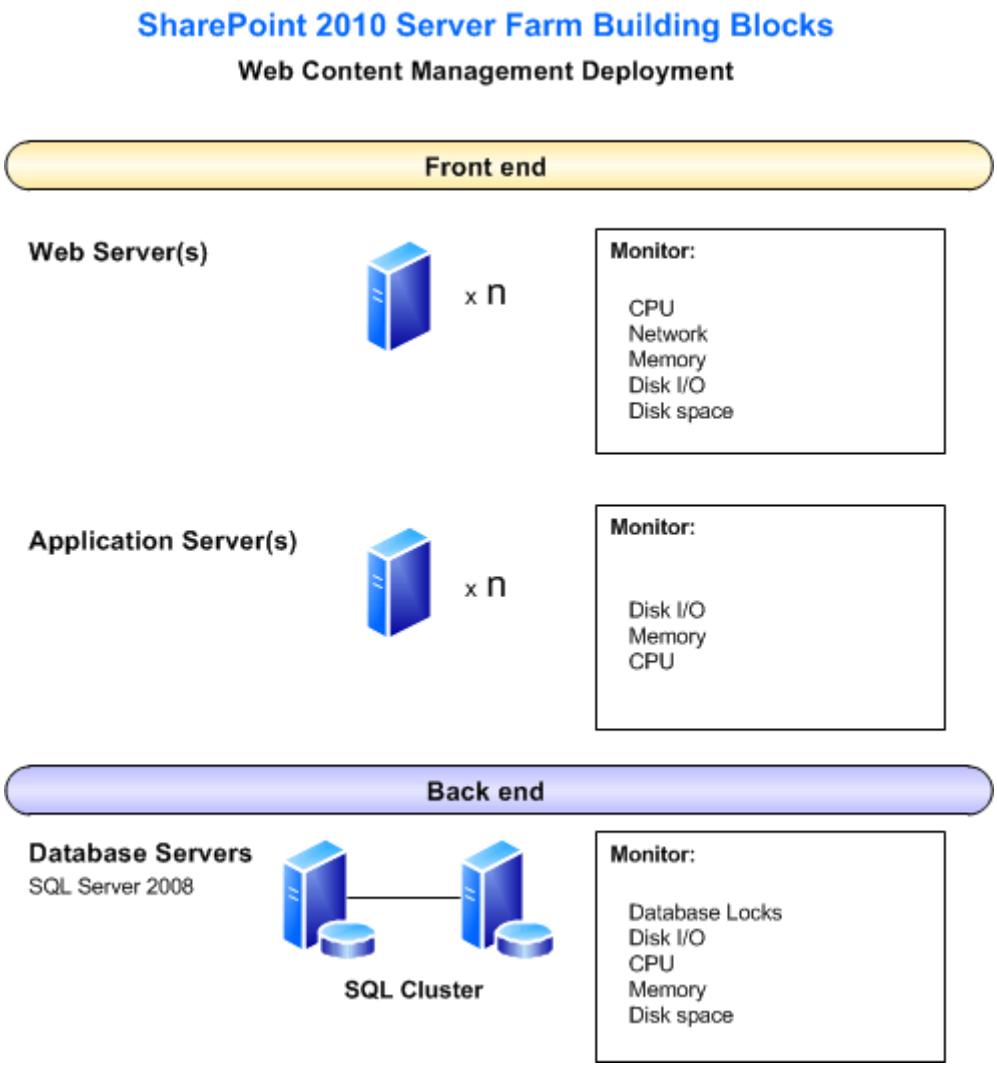
In this section:

Throughput is the key metric

Throughput and response time are the most important metrics to optimize when you conduct capacity planning for a SharePoint Server 2010 Web Content Management deployment. Throughput is the number of operations that a server farm can perform per second, measured in requests per second (RPS).

Bottlenecks and remediation

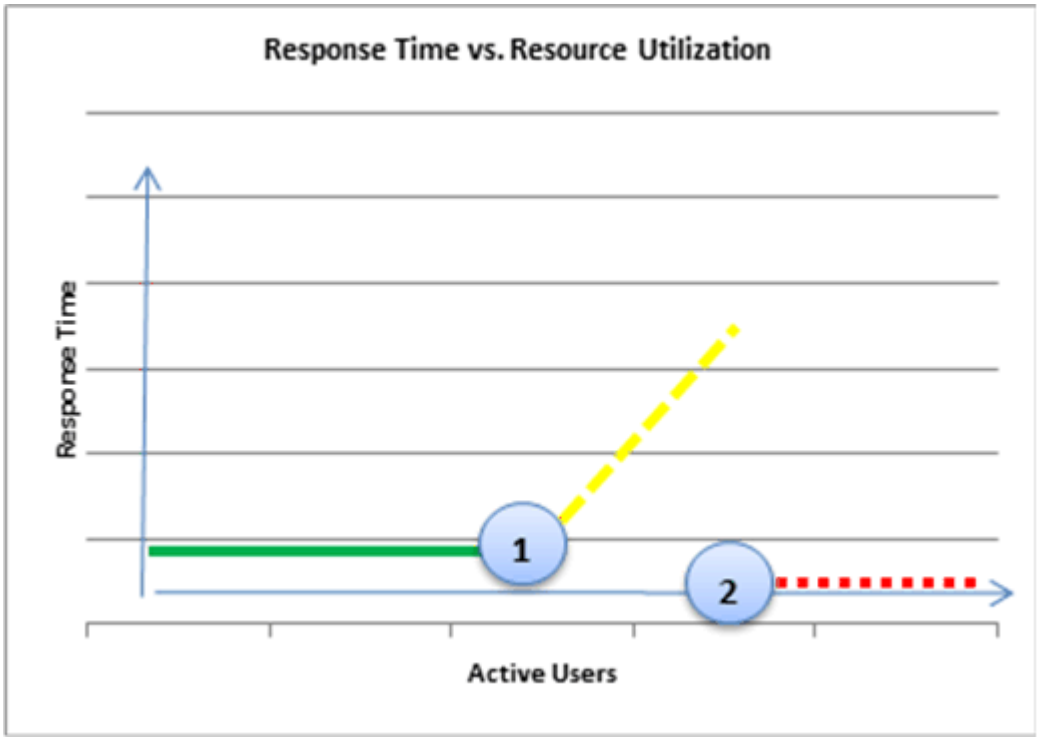
A bottleneck is the system resource that, when it is used up, prevents the server farm from serving additional requests. The following diagram shows the elements of a server farm and the resources that can become bottlenecks and that should be monitored.



Web server CPU utilization

The Web server CPU should be the bottleneck of a well-tuned topology because it is the most easily scalable component. The load balancer routes requests among Web servers and ensures that no single server is significantly more used than its peers.

Although additional users can visit the site after Web server CPU utilization is fully used, the server response time that these users experience increases. This behavior is useful for managing spikes in request volume. However, sustained load beyond a server farm's capacity eventually results in a backlog of requests that is large enough to exceed the waiting requests threshold. At this point, Web servers throttle requests and respond with HTTP error 503. In the following illustration, server response time decreases after the waiting requests threshold is met because only HTTP errors are served.



The following changes are shown in this diagram:

1. Response time increases as Web server CPU utilization approaches 100 percent.
2. After the waiting requests threshold is exceeded, additional requests are served with errors.

Other bottlenecks

If the Web server CPU is not the bottleneck, the next items to investigate for bottlenecks are the farm topology, the farm configuration, or the content of the pages being served. Some potential bottlenecks in these elements include the following:

1. **Network** In situations of high throughput, the network might be saturated either between the Web server and the database server or between the Web server and end users. To avoid this situation, we recommend that Web servers use dual gigabit network adapters.
2. **Database server CPU** If the database server CPU becomes the bottleneck, adding Web servers to the server farm cannot increase the maximum throughput that the farm can support. A bottleneck with the database CPU but not with the Web server CPUs can reflect two situations:
    a. Poor cache settings or very slow pages, especially those that are not output cached. This is characterized by high database server CPU utilization but low or medium throughput and low or medium Web server utilization.
    b. The database server might have reached capacity for the throughput required for the farm. This is characterized by high Web server and database server CPU utilization at high throughput.

Caching helps

SharePoint Server 2010 uses three kinds of caching. The common goal of these caches is to improve efficiency by reducing calls to the database for data that is frequently requested. Subsequent requests for a page can be served from the cache on the Web server, which results in significantly reduced resource utilization on the Web servers and database servers.

The three kinds of caching are as follows:

- **Output cache** This cache stores requested page content in the memory of the Web server. For more information about the output cache, see Output Caching and Cache Profiles (http://go.microsoft.com/fwlink/?LinkID=121543).
- **Object cache** This cache stores SharePoint objects, such as Web and list item metadata, in the memory of the Web server. For more information about the object cache, see Object Caching (http://go.microsoft.com/fwlink/?LinkID=123948).
- **Disk-based cache for Binary Large Objects (BLOBs)** This cache stores image, sound, video files, and other large binary files on disk. For more information about the BLOB cache, see Disk-Based Caching for Binary Large Objects (http://go.microsoft.com/fwlink/?LinkID=123947).

Each of these caches is important for sustaining high throughput. However, output caching has the largest effect and is discussed in detail throughout this article.

Test results and recommendations

This section discusses specific areas that were tested and provides recommendations that result from those tests.

In this section:

Effect of enabling the output cache

The output cache is a valuable feature to use to optimize a SharePoint Server 2010 solution for lots of read operations.

For these tests, to determine maximum RPS, the number of active users making requests on the farm was increased until CPU utilization of either the database server or the Web servers reached 100

percent and became a bottleneck. The test was conducted on 1x1 (1 Web server and 1 database server), 2x1, 4x1, and 8x1 farm topologies to demonstrate the effect of scaling out the Web servers at different output cache hit ratios.

Output cache hit ratio

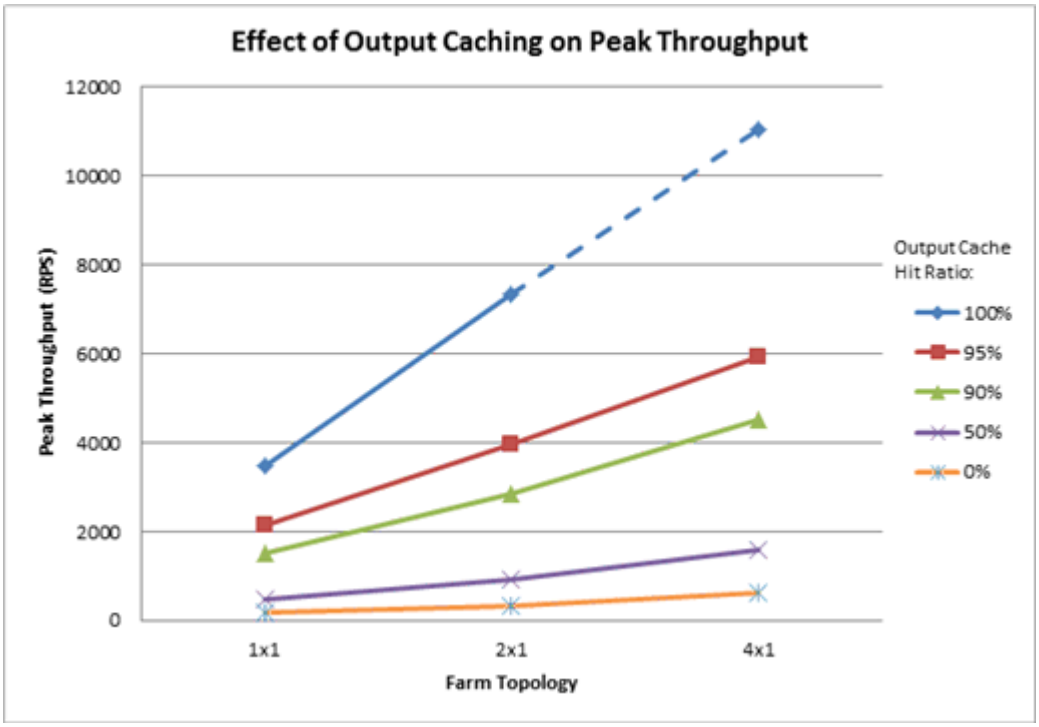
The output cache hit ratio is a measure of output cache hits to misses.

- A cache **hit** occurs when the cache receives a request for object data that is already stored in the cache.
- A cache **miss** occurs when a request is received for object data that is not stored in the cache. When a cache miss occurs, the cache will attempt to store the requested object data so that later requests for that data result in a cache hit.

There are several reasons why a page request might result in a cache miss.

- Pages that are configured not to use the output cache.
- Personalized pages, for example, pages that have data that is specific for the current user.
- First time browse per output cache variation key.
- First time browse after cached content has expired.

The following diagram shows the effect of output caching on peak throughput in farms ranging from one to four Web servers and one database server.



Note:

The data point for maximum RPS on a 4x1 server farm with a 100 percent output cache hit ratio is extrapolated and was not actually observed. The server farm request volume reached the network bandwidth limit; that is, the data transfer rate approached 1 gigabit per second. In all cases, the Web server CPU utilization is 100 percent.

The following table lists the effects of output cache hit ratios on farm topologies ranging from one to four Web servers and one database server.

Effects of output cache hit ratio on different farm topologies

Output cache hit ratio	Measure	1x1	2x1	4x1
100%	Maximum RPS	3,463	7,331	11,032
	SQL Server CPU utilization	0%	0%	0%
95%	Maximum RPS	2,137	3,945	5,937
	SQL Server CPU utilization	5.93%	12.00%	21.80%
90%	Maximum RPS	1,518	2,864	4,518
	SQL Server CPU utilization	7.12%	14.40%	28.00%
50%	Maximum RPS	459	913	1,596
	SQL Server CPU utilization	9.86%	19.50%	42.00%

Output cache hit ratio	Measure	1x1	2x1	4x1
0%	Maximum RPS	172	339	638
	SQL Server CPU utilization	9.53%	19.00%	36.30%

Conclusions and recommendations for the effect of enabling the output cache

Higher output cache hit ratios yield significant increases in maximum RPS. Therefore, we recommend enabling output caching to optimize your SharePoint Server 2010 publishing solution. You can configure the output cache on the Output Cache Settings page for the site collection. For more information, see Configure page output cache settings for a site collection (http://go.microsoft.com/fwlink/?LinkId=205058) on the Office.Microsoft.com Web site.

In tests that had output caching enabled, the first request that cached a page was excluded; that is, a certain percentage of pages are already stored in the cache. When a user first requests a page that is not cached, the page is added to the cache. If the cache has reached or is approaching capacity, the cache trims the data that was least recently requested.

The 0 percent cache hit ratio simulates the short time in an environment during which the enabled output cache is being filled after it was flushed. For example, this is observed every day in a real-world environment when the application pool recycles. It is important to scale your hardware up or out appropriately to accommodate a situation in which there is a 0 percent cache hit ratio for the brief time between the application pool recycle and the next requesting and caching of pages. The 0 percent cache hit ratio also simulates an environment in which output caching is not enabled.

Anonymous users and authenticated users

The previous test assumes that all requests to the site are made by anonymous readers. However, in your site, some or all users might be authenticated. Examples of authenticated read scenarios include a corporate intranet publishing site and members-only content on an Internet site.

With output cache profiles, you can specify output cache behavior for authenticated users that differs from the behavior for anonymous users.

Cache profiles

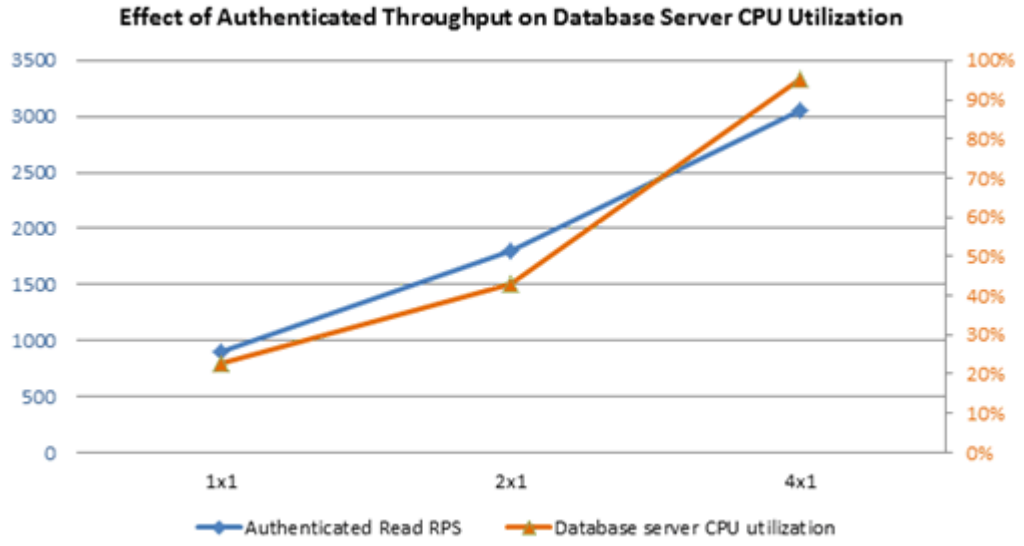
Cache profiles aggregate settings that you can apply to pages, page items, content types, and levels of scale in a site deployment. A cache profile defines the following kinds of cache behavior:

- The length of time that items should be held in the cache.
- The security trimming policy.
- The expiration of settings, such as duration and changes.
- The variations of cached content, for example, based on user permission, user rights, and other custom variables.

Any change to a cache profile immediately affects all applicable content on the site. You can set different cache profiles for anonymous users and for authenticated users.

For anonymous users, the Public Internet (Purely Anonymous) output cache profile was used and for authenticated users, the Extranet (Published Site) output cache profile was used.

The following chart shows the effects of authenticated throughput on database server CPU utilization.



The authentication model was Windows Basic authentication. Although we do not recommend that you use Windows Basic authentication for Internet sites, this authentication method was selected to demonstrate a minimum overhead that is imposed by authentication. The size of this overhead varies by your specific authentication mechanism. When you are testing your deployment, make sure that you account for the effect of your authentication mechanism. For more information about the authentication mechanisms that are supported by SharePoint Server 2010, see Plan authentication methods (SharePoint Server 2010).

Conclusions and recommendations for anonymous users and authenticated users

Authenticated users experience lower RPS and less scale-out potential because the additional work of validating credentials exerts load on the database server. As demonstrated in the test results, the maximum RPS that is observed when users are authenticated is significantly lower than that of an anonymous access farm.

Scale-out characteristics of read and write operations

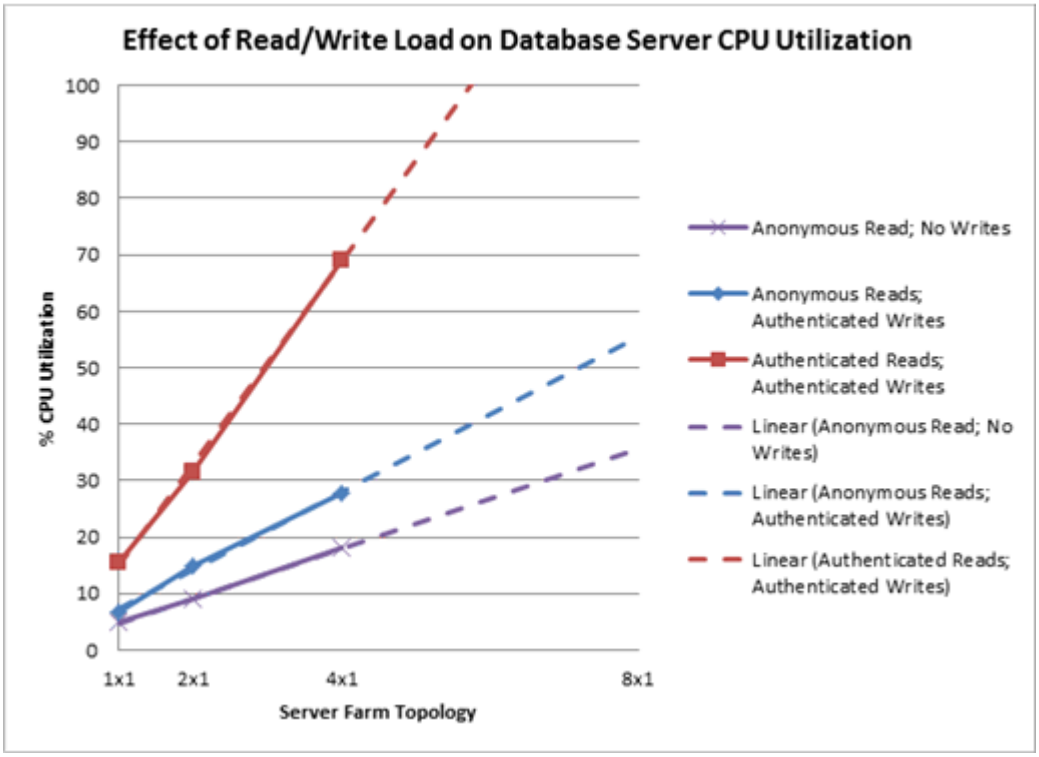
Our tests were constructed to record writes per hour. In this article, a write is defined as either the creation and check-in of a new Publishing Page or the editing and check-in of an existing Publishing Page.

For the following tests, readers were added to the system until Web server CPU utilization reached approximately 80-90 percent, and then write operations were performed in the environment by using artificial delay. The total writes per hour for the test was approximately 500. We used a 90 percent output cache hit ratio for all tests. We performed the same test on a 1x1, 2x1, and 4x1 farm and observed the Web server and SQL Server CPU usage in addition to the overall read throughput for each configuration. In addition, we tested an anonymous read-only configuration as a baseline, and we also tested a configuration with authenticated readers by using Windows Basic authentication.

Although the Web server CPU was fully utilized at 100 percent usage during the read-only scale-out tests, we held the Web server CPU between 80-90 percent for the scale-out tests with writes. This was done to leave room for additional CPU utilization when write activity was being performed.

The following chart shows the overall read RPS that were observed during each test. The read RPS scales linearly as additional Web servers are added, even with write activity. However, there is an RPS loss when writes are incorporated.

Database server CPU usage increased as the number of Web servers increased. The following chart shows the growth pattern of SQL Server CPU usage in the various configurations. As observed in the Anonymous users and authenticated users section earlier in this article, authentication affects database server CPU utilization, and this becomes more pronounced as write activity is added (which also affects database server CPU utilization).



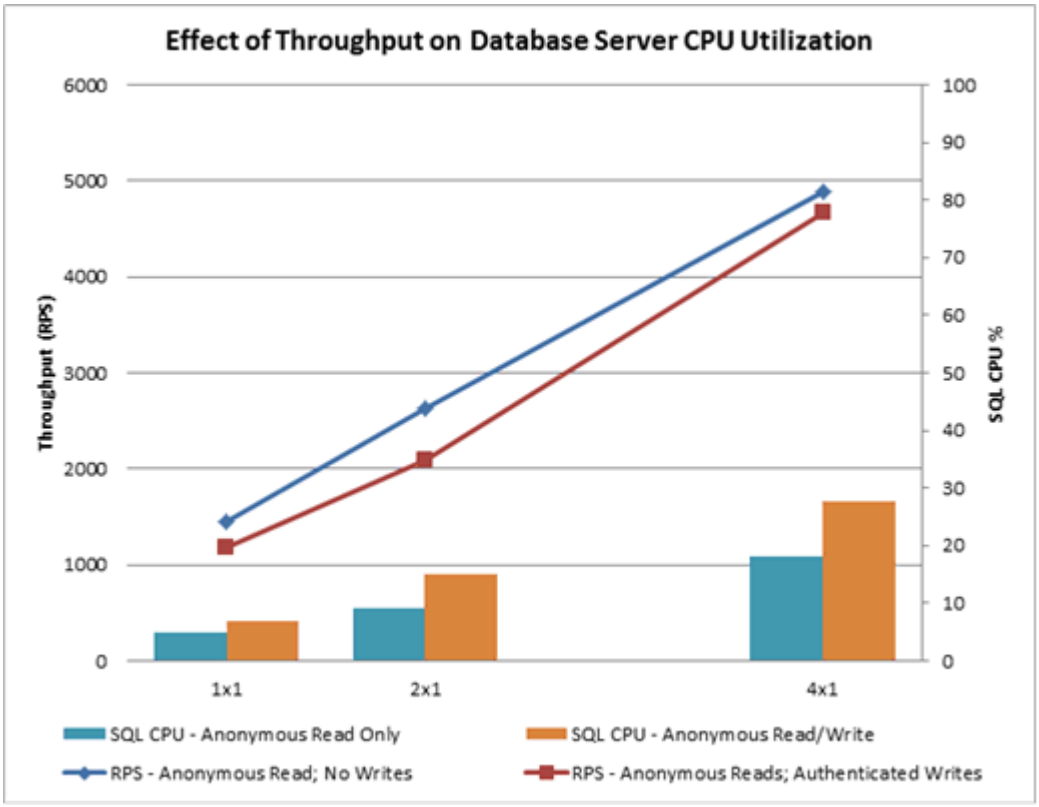
The extrapolated trend in SQL Server usage demonstrates that SQL Server will become the bottleneck with six Web servers that have authenticated read requests. However, in the anonymous read case, scaling out to a larger number of Web servers is workable.

It is important to be aware that additional factors in a typical deployment affect the load on the database server, and these factors are important to account for when you are conducting capacity planning. To learn more about how to determine a green zone for typical database server CPU utilization, see Capacity management and sizing overview for SharePoint Server 2010.

Conclusions and recommendations for scale-out characteristics of read and write operations

Our data shows that scaling out the number of Web servers is an effective strategy for increasing throughput if the database server does not become the bottleneck. On average, the anonymous read/authenticated writes test mix exerted a 52 percent increase in Web server CPU utilization compared to an anonymous read/no writes test mix. In addition, authenticated reads add a large additional SQL Server load, because each request incurs additional authentication checks, which requires a round trip to SQL Server.

The following chart shows the effect of throughput on database server CPU utilization.



Output cache caveats

If the only concern in capacity planning were to maximize RPS, these tests would suggest that the optimal cache hit ratio is 100 percent. However, it might not be workable or desirable to enable output caching of any or all pages because of data freshness requirements or memory constraints.

Data freshness

Data that is served from the output cache might not contain recent updates that have been made to the original content. In the source of content deployment or (for authenticated authors) in an author-in-production scenario, authors might want to see the most recent changes immediately after they update existing content.

This is generally eased by setting the **Duration** property in the cache profile, which specifies how long a cached page persists in the output cache before it expires. When a page expires, it is removed from the cache and a later request results in a cache miss that refreshes the page content.

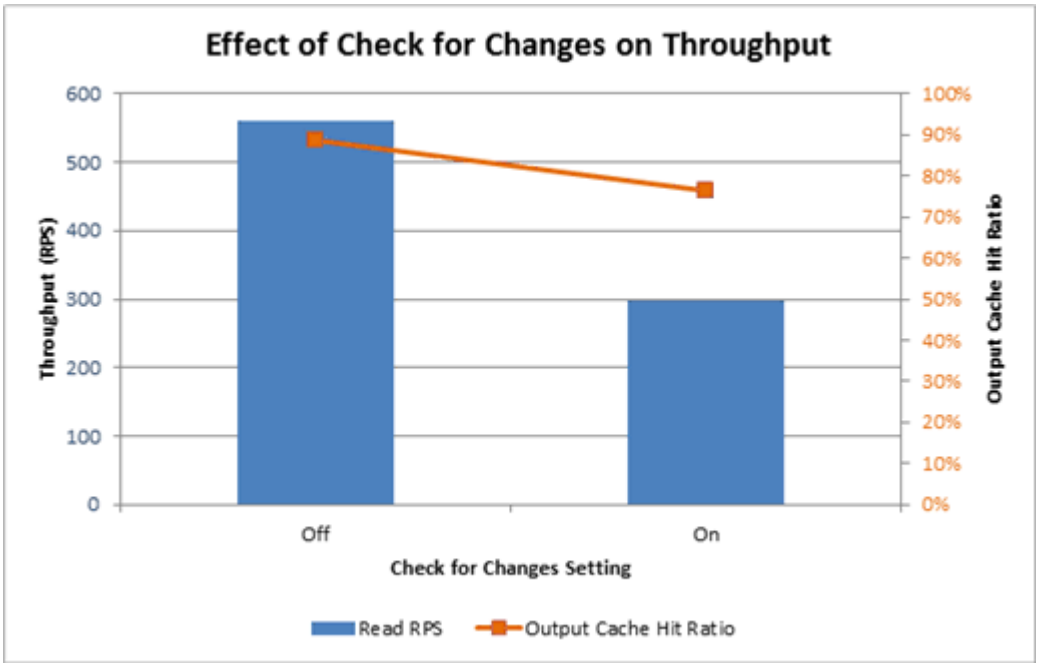
The **Check for Changes** cache profile property can also be set so that the server compares the time at which a page was cached with the time at which content was last modified in a site collection. A request for a page that has unmatched time stamps causes cache invalidation for all pages in the site collection. Because the **Check for Changes** property affects all pages in a site collection, we

recommend enabling this option only if there is an authenticated author-in-place solution that is infrequently updated and basically static. Combining this option with a long duration enables all pages to be served from the cache until an update is made to the site.

By default, the **Check for Changes** property is not enabled. This means that the Web server serves data from the output cache in response to requests for a page that has not yet expired, regardless of whether the underlying, original ASPX page was modified.

In this test, conducted on a 1x1 server farm, all variables are the same as in the tests in the Scale-out characteristics of read and write operations section except for the **Check for Changes** property. When the **Check for Changes** property is turned on, the cache is flushed more often, which results in a lower output cache hit ratio.

The following chart shows the effect of the **Check for Changes** property on throughput.



We recommend avoiding the **Check for Changes** output cache profile property except in specific cases. A site that uses the author-in-place model and experiences infrequent changes in content might benefit from this setting for authenticated users together with a long cache duration, but other kinds of sites will most likely have a degradation in RPS.

Depending on your requirements, you might want time-sensitive content to go live instantly or faster than the default settings allow for. In this case, you should decrease the duration or not enable output caching.

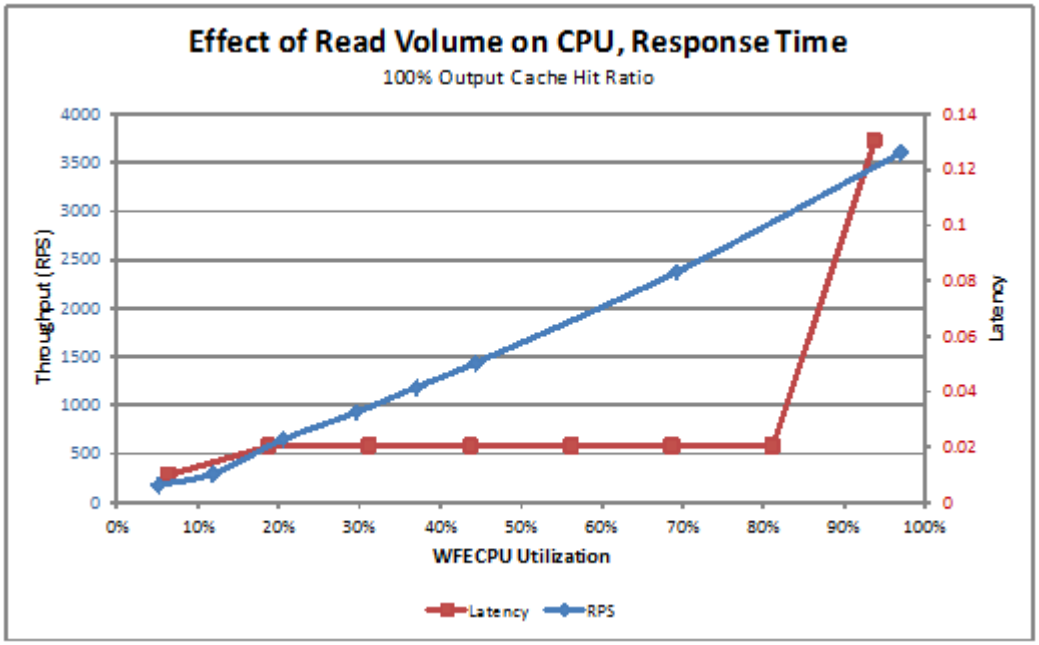
Conclusions and recommendations for output cache caveats

Output caching does not solve all the problems that are related to capacity management. There are some situations in which output caching is unsuitable, and you should consider these when you enable the output cache and configure output cache profiles.

Effect of read volume on CPU and response time

This test was conducted on a 1x1 farm with anonymous access and output caching enabled.

The following chart shows the effect of read volume on CPU and response time.



Conclusions and recommendations for effect of read volume on CPU and response time

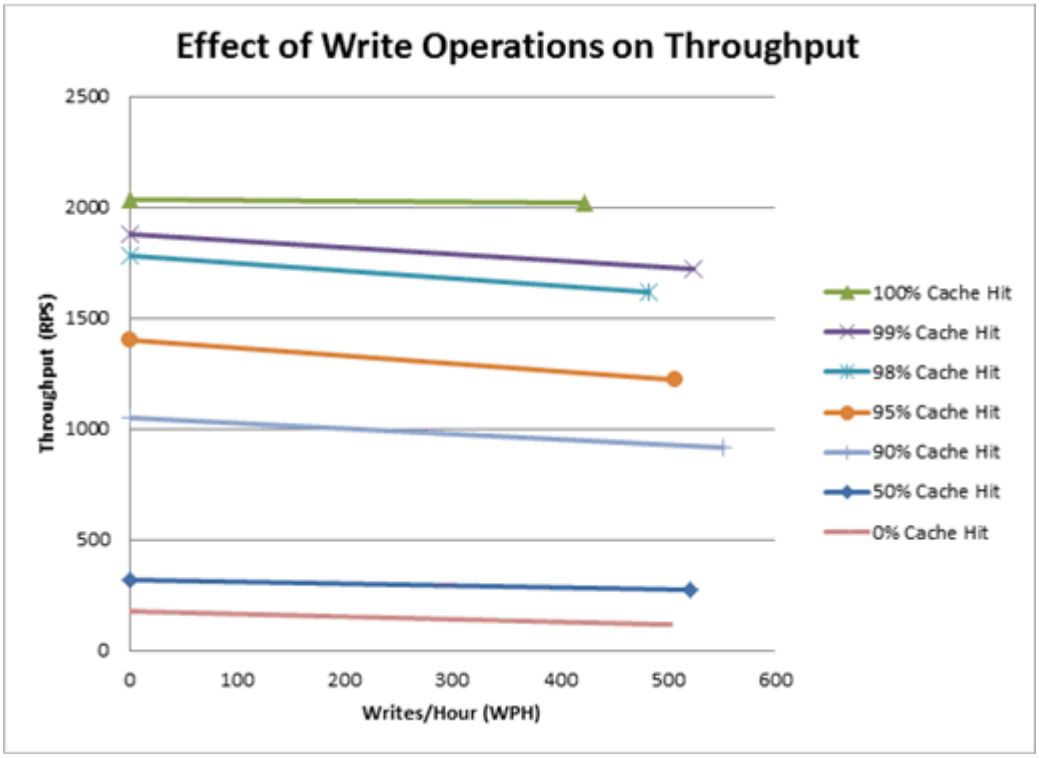
As discussed in the Bottlenecks and remediation section, server response time will stay generally constant until the Web server receives sufficient request volume to fully use its CPU. As Web server CPU is fully utilized, response time will increase significantly. However, the server farm will still be able to handle some additional request volume.

Effect of write operations on throughout

The ratio of creation to editing operations is distributed evenly through the course of the tests. Writes per hour values were tested up to approximately 500, because creating or editing more than 500 pages per hour is outside the range which most SharePoint deployments would operate. The test did not cover automated processes, such as content deployment, which is discussed in the Effect of content

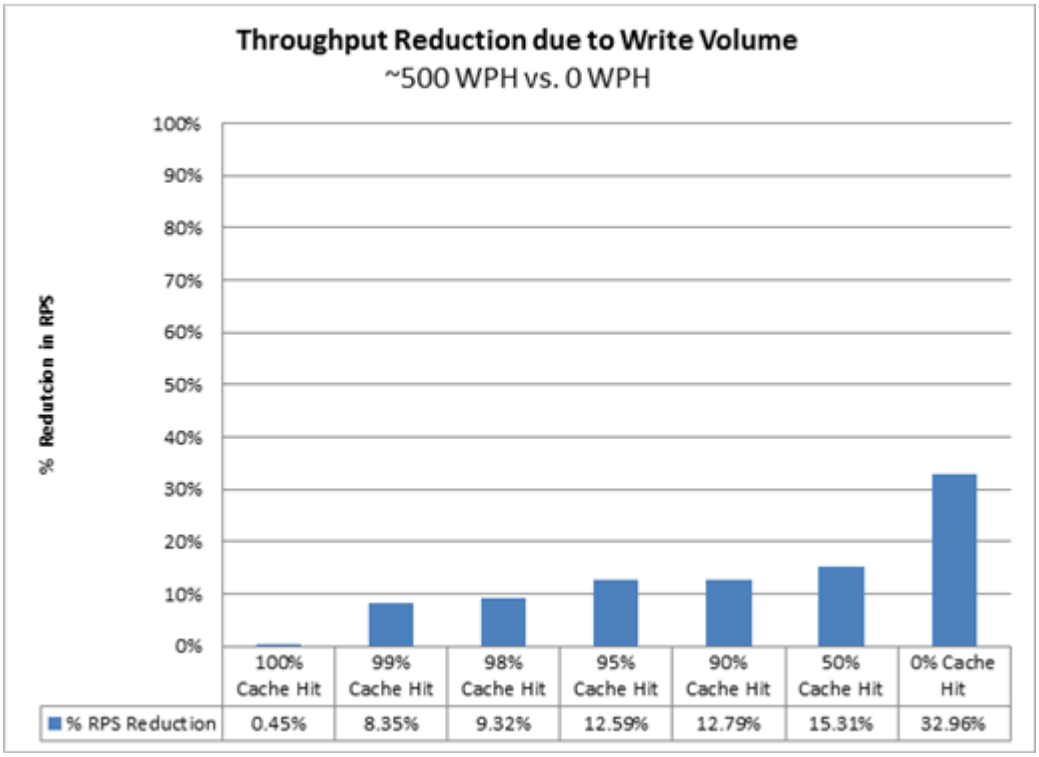
deployment section. These create and edit operations might result in multiple SQL Server operations. Therefore, it is important to be aware that the writes that are measured in these tests are not at the SQL Server level, but instead represent what a user would consider a write operation. All RPS versus writes per hour tests were conducted on a 1x1 farm.

We first added read operations to the test until Web server CPU was between 60 and 80 percent to leave a buffer for traffic spikes, and we maintained this average utilization level throughout the course of the test. We then introduced writes by using an artificial delay to control the number of write operations. However, there were spikes of increased Web server and SQL Server CPU usage while the writes were occurring. Some of these spikes for some cache hit ratios exceeded the threshold for ordinary operation as shown in the following chart, although the average stayed within the range of ordinary operation.

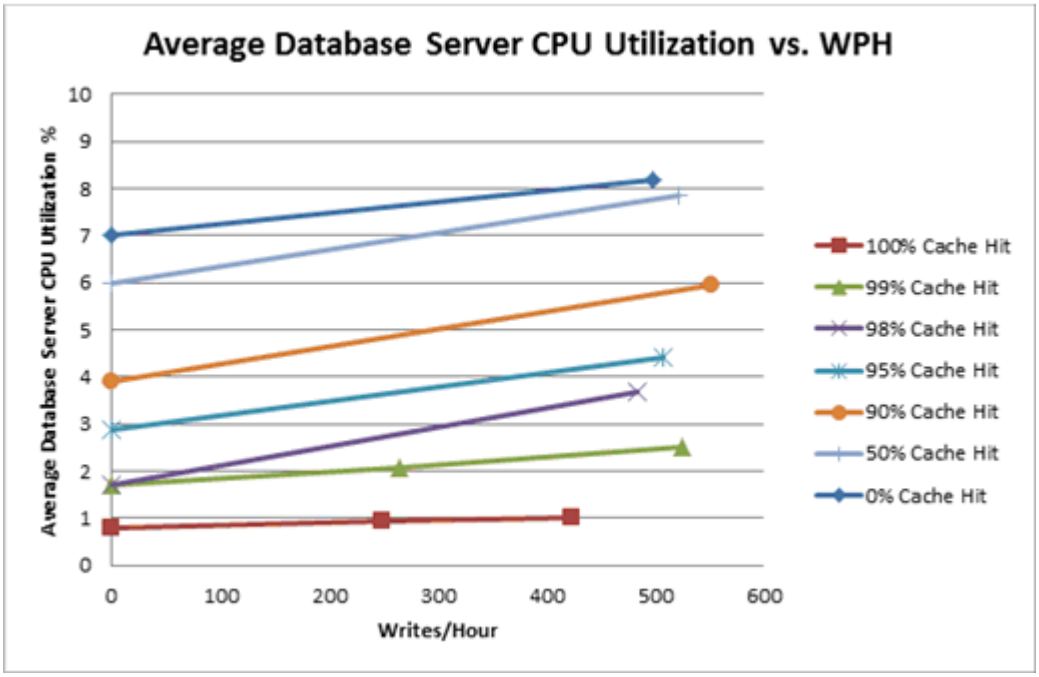


As shown in the previous chart, there is a minor reduction in throughput as writes are added to the environment. The graph demonstrates the change in throughput between a read-only scenario and read operations during approximately 500 writes per hour. Two data points were recorded for each output cache hit ratio. Therefore, the relationship between data points is not necessarily linear.

The percentage reduction is more pronounced for lower cache hit ratios, as shown in the following chart. Overall read RPS remains largely dependent on the cache hit ratio, regardless of the writes.

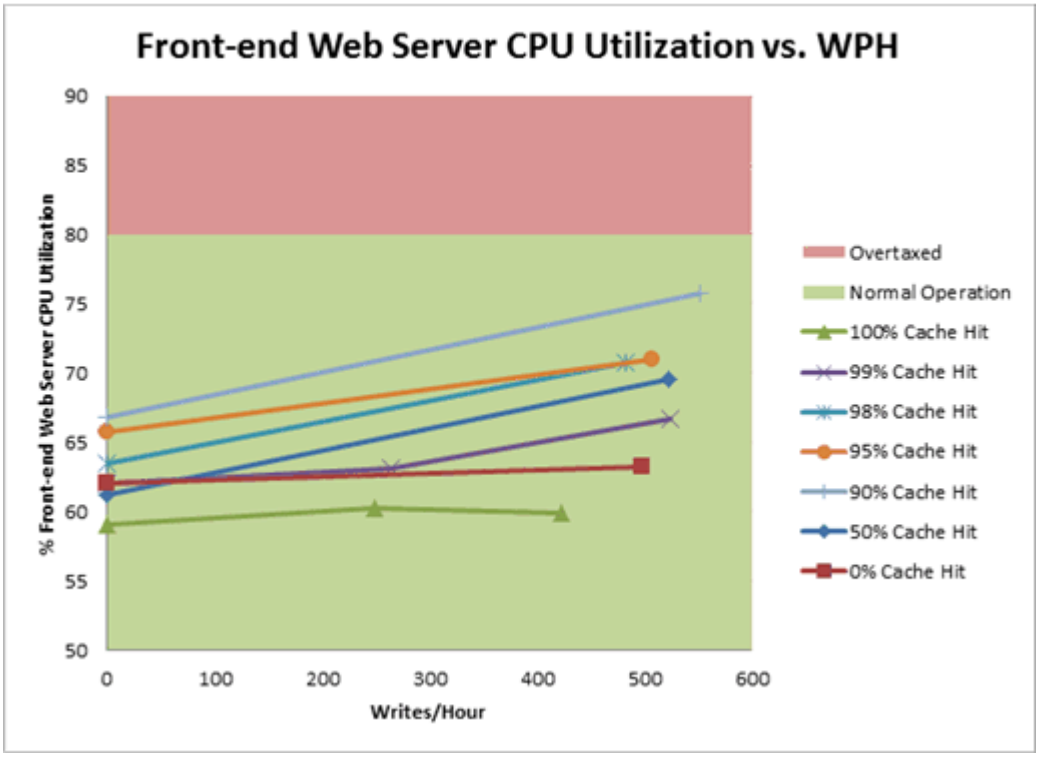


The following chart demonstrates that the database server CPU utilization did not increase appreciably when writes were added to the system. Note that the vertical scale is from 0 to10 percent of CPU capacity.

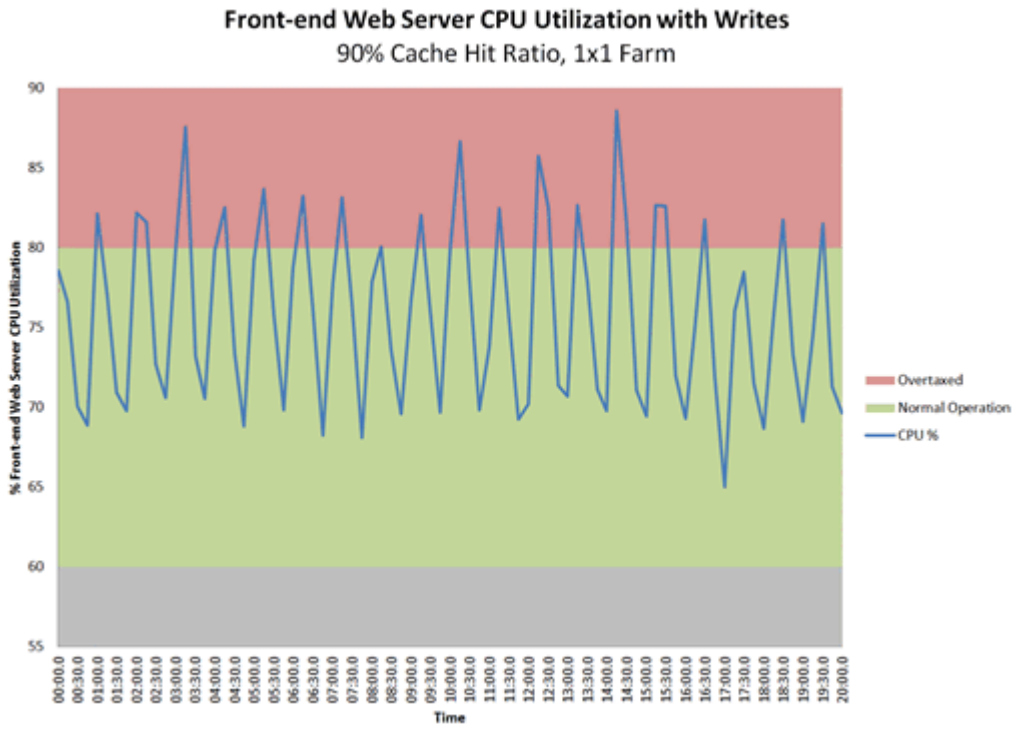


Additional SQL Server load was observed during the write operations, which is expected. However, the largest increase was an additional 2.06 percent, which is insignificant. Average database server CPU utilization stayed lower than 10 percent throughout all tests. This test was performed on a 1x1 farm. Database server CPU usage will increase as the number of Web servers is scaled out. This is discussed more in Scale-out characteristics of read and write operations.

Web server CPU utilization was also measured during the tests. The following chart demonstrates that average Web server CPU usage remained in the 60-80 percent range throughout the course of the tests, even as the writes per hour approached 500.



However, the actual measured CPU utilization spikes when the writes occur, as shown in in the following chart. In general, these CPU spikes do not represent a problem. The goal of the green zone is to provide CPU head room to absorb some spikes in CPU load. Also, as additional Web servers are added, the effect of the spikes will be distributed across these servers so that the effect on a single Web server CPU will be lessened. However, you should know that such spikes would be expected in a real deployment; write activity is not uniformly distributed, although it does generally occur in bursts.



A 90 percent cache hit ratio is very low for a typical deployment. Most SharePoint Server 2010 deployments with lots of read operations will have an output cache hit ratio of 95 percent or more.

Conclusions and recommendations for effect of write operations on throughput

The data that is presented indicates that write operations will not have a large adverse effect on the overall throughput of the system for readers. You should recognize that write activity can cause spikes in CPU usage and you should plan your typical configuration to expect these spikes. A strategy for leveling these spikes is to scale out to multiple Web servers. This has two advantages:

- It spreads out the write load to multiple Web servers, which smoothes the overall spikes.
- It provides increased read RPS, especially at high output cache hit ratios.

Effect of content deployment

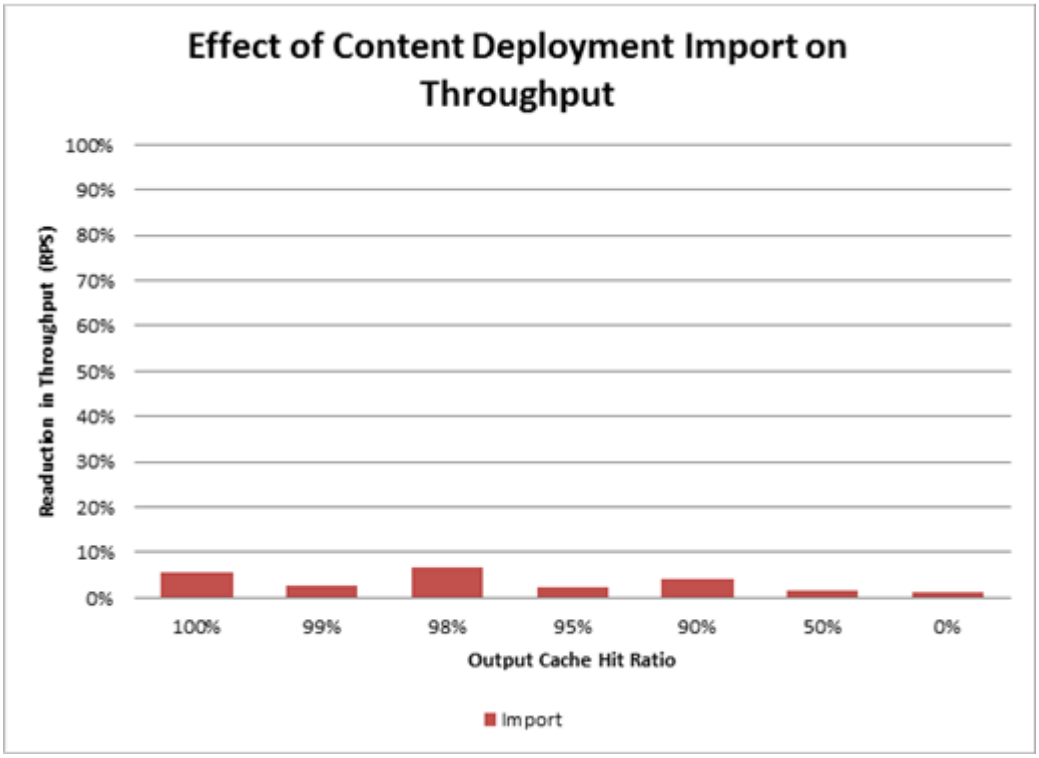
An alternative to the author-in-place model, which uses a single environment for editing and reading, is to split the single environment into two separate environments — an authoring environment and a production environment — and to use content deployment to copy content from the authoring environment to the production environment.

In this configuration, the production environment ranges from little write activity to no write activity, except when content deployment is importing content. For these tests, reads were added until the Web server CPU usage entered the 70-80 percent range. The content deployment job then exported 10 sites

that have 500 pages each from the authoring site collection as a package and imported this package into the publishing site collection. The size of the deployed package is larger than what is typically observed in real-world environments in order to sufficiently extend the duration of the content deployment job to see test results. For additional information about characteristics of the deployed content, see the Dataset section.

When export was complete, we imported the content into a separate site collection and measured the application server and SQL Server load, in addition to the throughput, while import was in progress. The import test was conducted for several different output cache hit ratios.

The key observation is that import has only a minor effect on overall read RPS, as shown in the following chart. We also observed that import had no significant effect on the Web server CPU utilization, as shown in the following tables, regardless of cache hit ratio. However, there was a more noticeable effect on SQL Server CPU, shown in the following chart. This is expected, because the database server will experience additional load while content is imported in it. However, the SQL Server CPU stayed lower than 12 percent usage at all cache hit ratios tested, even during import.



The following tables show the effect of content deployment import on Web server and database server CPU utilization.

Effect of content deployment import on Web server CPU utilization

Cache hit	100%	99%	98%	95%	90%	50%	0%
Baseline	72.32%	73.26%	71.28%	73.53%	71.79%	68.05%	63.97%
With import	70.93%	74.45%	69.56%	74.12%	70.95%	67.93%	63.94%

Effect of content deployment import on database server CPU utilization

Cache hit	100%	99%	98%	95%	90%	50%	0%
Baseline	1.19%	1.64%	2.01%	3.00%	3.73%	5.40%	6.82%
With import	6.03%	6.82%	6.97%	7.96%	8.52%	10.29%	10.56%

Conclusions and recommendations for effect of content deployment

The results from our tests show that performing content deployment operations during ordinary site operations does not pose a significant performance concern. These results show that it is not strictly necessary to deploy content during low-traffic periods to minimize the effect of the operation on overall performance and that deployment times can be driven primarily by business needs instead of performance requirements.

Effect of database snapshot during content deployment export

In SharePoint Server 2010, content deployment can be configured to create a snapshot of the source content database before exporting content from it. This effectively shields the export process from any authoring activity that might be occurring while the export happens. However, snapshots can affect the write performance of the database server, because the snapshot acts as a multiplier for the writes. For example, if you have two snapshots of a source database, and then you write to the source database, the database server copies the existing data to each snapshot, and then it writes the new data into the source database. This means that a single write to the source database incurs three actual writes: one to the source database, and an additional one for each database snapshot.

To determine the effect of a snapshot on the authoring environment, we measured the write RPS, response time, and the CPU utilization of the Web servers, database server, and application server during an export operation while write activity was also occurring. The following tables display the results.

Effect of database snapshots during content deployment

Metric	4 WPH - No snapshots	4 WPH - With snapshots
Write RPS	0.2	0.16

Metric	4 WPH - No snapshots	4 WPH - With snapshots
Response time	0.13	0.15
Web server CPU %	0.42%	0.27%
Application server CPU%	8.67%	8.98%
Database server CPU %	3.34%	2.41%

Effect of database snapshots during content deployment

Metric	8 WPH - No snapshots	8 WPH - With snapshots
Write RPS	0.44	0.44
Response time	0.13	0.13
Web server CPU %	0.61%	0.73%
Application server CPU%	14.6%	12%
Database server CPU %	7.04%	7.86%

Conclusions and recommendations for effect of database snapshot during content deployment export

The results of our tests showed no significant effect on any measured data points with database snapshots. All variance that was recorded was within the margin of error. This confirms that database snapshots can be used without strong performance considerations.

Content characteristics

The tests were conducted on a single dataset that was created to answer a specific set of questions. Your dataset will differ and will change over time. This section investigates how content characteristics, such as the number of pages in the page library and the features that are included on pages, can inform capacity management decisions.

Number of pages

Maximum RPS across many page library sizes was tested. This test was conducted on a 4x1 topology with output caching disabled and with anonymous access. All pages were 42 KB uncompressed, 12 KB compressed. The following table shows the test results.

Effect of library page count on RPS

Number of pages	3	1,000	20,000
Maximum RPS	860	801	790

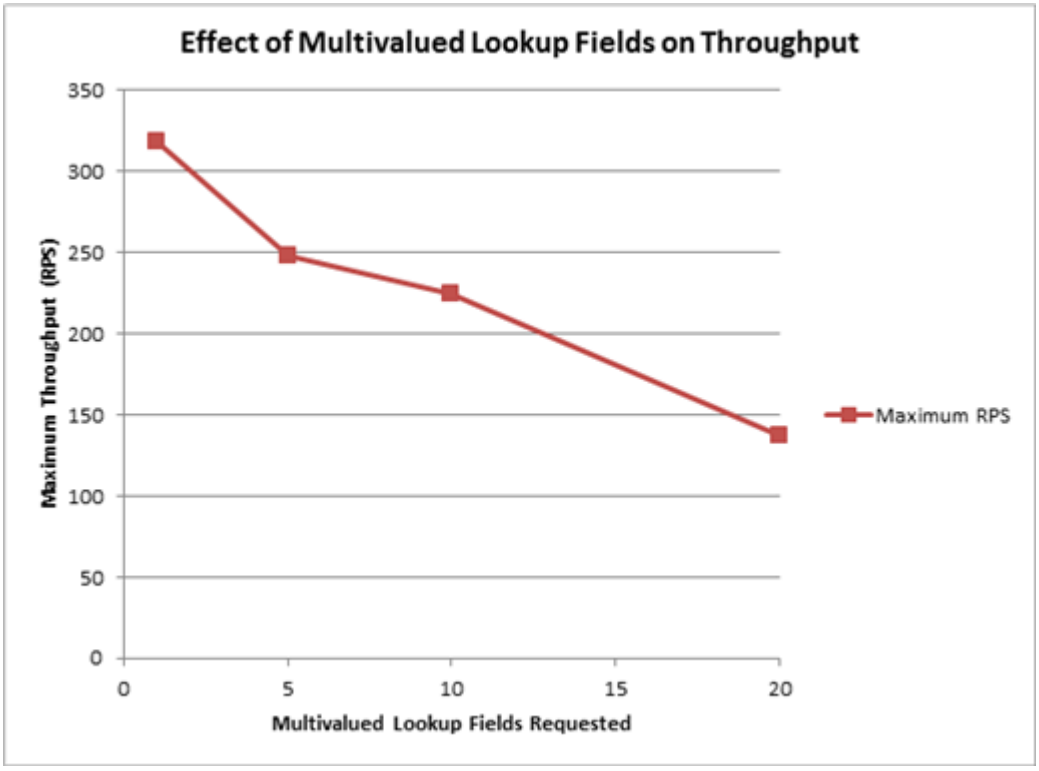
Increasing the number of pages to 20,000 did not have a significant effect on maximum RPS.

Multivalued lookup fields

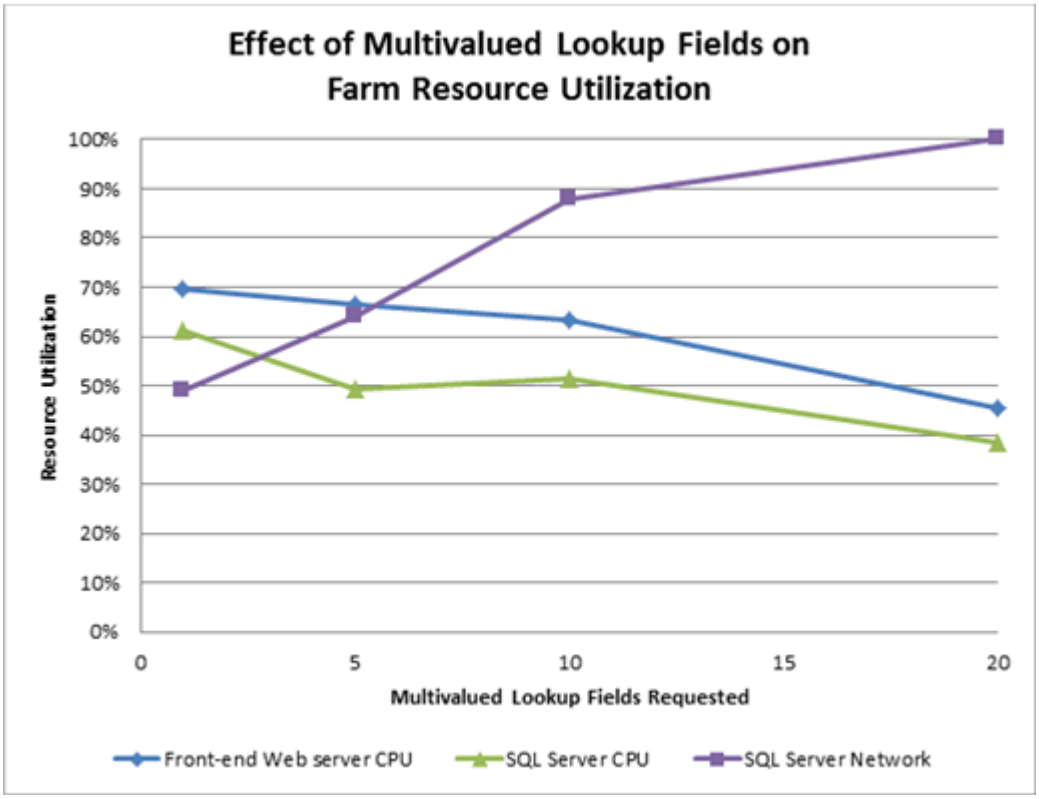
A multivalued lookup field is a column in a list that references one or more items in another list, such as columns that use enterprise managed metadata. These fields are generally used as search keywords for a page and are not necessarily rendered. In some cases, for example enterprise wikis, it makes sense to render these field values into the contents of pages. For instance, pages might be filed into categories when they are created (for example, World News, Human Interest, and Sports on a news site) and the master page includes a placeholder that will show the user which categories the page was tagged with.

Multivalued lookup fields cause more data to be fetched every time a page is requested. Therefore, having many multivalued lookup fields on a page can affect performance.

The following chart shows the effect of multivalued lookup fields on throughput.



The following chart shows the effect of multivalued lookup fields on farm resource utilization.



Maximum RPS degradation occurs as the number of multivalued lookup fields increases due to the effect on the network between the Web server and the database server.

Effect of usage reporting

Usage reporting is a service that helps administrators monitor statistics about the use of their sites. For more information about usage reporting, see Configure usage and health data collection (SharePoint Server 2010).

We tested the effect of usage reporting timer jobs on maximum RPS on a 1x1 farm. The following table describes the results.

Effect of usage logging on performance (in RPS)

	Usage DB on	Usage DB off	Difference
Output cache on	3,459	3,463	4
Output cache off	629	638	9

The results show that enabling usage logging does not significantly affect RPS in a read-only scenario.

About the authors

Joshua Stickler is a Program Manager for SharePoint Server 2010 at Microsoft.

Tyler Butler is a Program Manager for SharePoint Server 2010 at Microsoft.

Zhi Liu is a Software Development Engineer in Test for SharePoint Server 2010 at Microsoft.

Cheuk Dong is a Software Development Engineer in Test for SharePoint Server 2010 at Microsoft.

Philippe Flamm is a Software Development Engineer in Test for SharePoint Server 2010 at Microsoft.

Estimate performance and capacity planning for workflow in SharePoint Server 2010

This performance and capacity planning article provides guidance on the effect that the use of workflow has on topologies that run Microsoft SharePoint Server 2010.

For general information about capacity planning for SharePoint Server 2010, see Performance and capacity management (SharePoint Server 2010).

Contents

Test farm characteristics

The following sections describe the characteristics of the test farm:

- Dataset
- Workload
- Hardware, settings, and topology

Dataset

To get benchmarks, most tests ran on a default Team Site on a single site collection in the farm. The manual start tests started workflows by using a list that has 8,000 items.

Workload

Testing for this scenario helps develop estimates of how different farm configurations respond to changes to the following variables:

- Effect of the number of front-end servers on throughput for manually starting declarative workflows across multiple computers
- Effect of the number of front-end servers on throughput for automatically starting declarative workflows on item creation across multiple computers
- Effect of the number of front-end servers on throughput for completing tasks across multiple computers

The specific capacity and performance figures presented in this article will differ from the figures in real-world environments. The figures presented are intended to provide a starting point for the design of an

appropriately scaled environment. After you complete the initial system design, test the configuration to determine whether it will support the factors in your environment.

This section defines the test scenarios and discusses the test process that was used for each scenario. Detailed information such as test results and specific parameters are given in each test result section later in this article.

Test name	Test description
Throughput for starting workflows manually	1. Associate the included MOSS Approval workflow with a list that creates one task. 2. Populate the list with list items. 3. Call the StartWorkflow Web service method on Workflow.asmx against the items in the list for five minutes. 4. Calculate throughput by looking at the number of workflows in progress.
Throughput for starting workflows automatically when an item is created	1. Associate the included MOSS Approval workflow with a list that creates one task, set to automatically start when an item is created. 2. Create items in the list for five minutes. 3. Calculate throughput by looking at the number of workflows in progress.
Throughput for completing workflow tasks	1. Associate the included MOSS Approval workflow with a list that creates one task, set to automatically start when an item is created. 2. Create items in the list. 3. Call the AlterToDo Web service method on Workflows.asmx against the items in the task list that was created by the workflows that started. 4. Calculate throughput by looking at the number of workflows completed.

Hardware, settings, and topology

Topologies that were used for these tests use a single computer for the content database and from one to four front-end computers that have the default installation for SharePoint Server 2010. Although the workflows that were used in these tests are not available in Microsoft SharePoint Foundation 2010, the

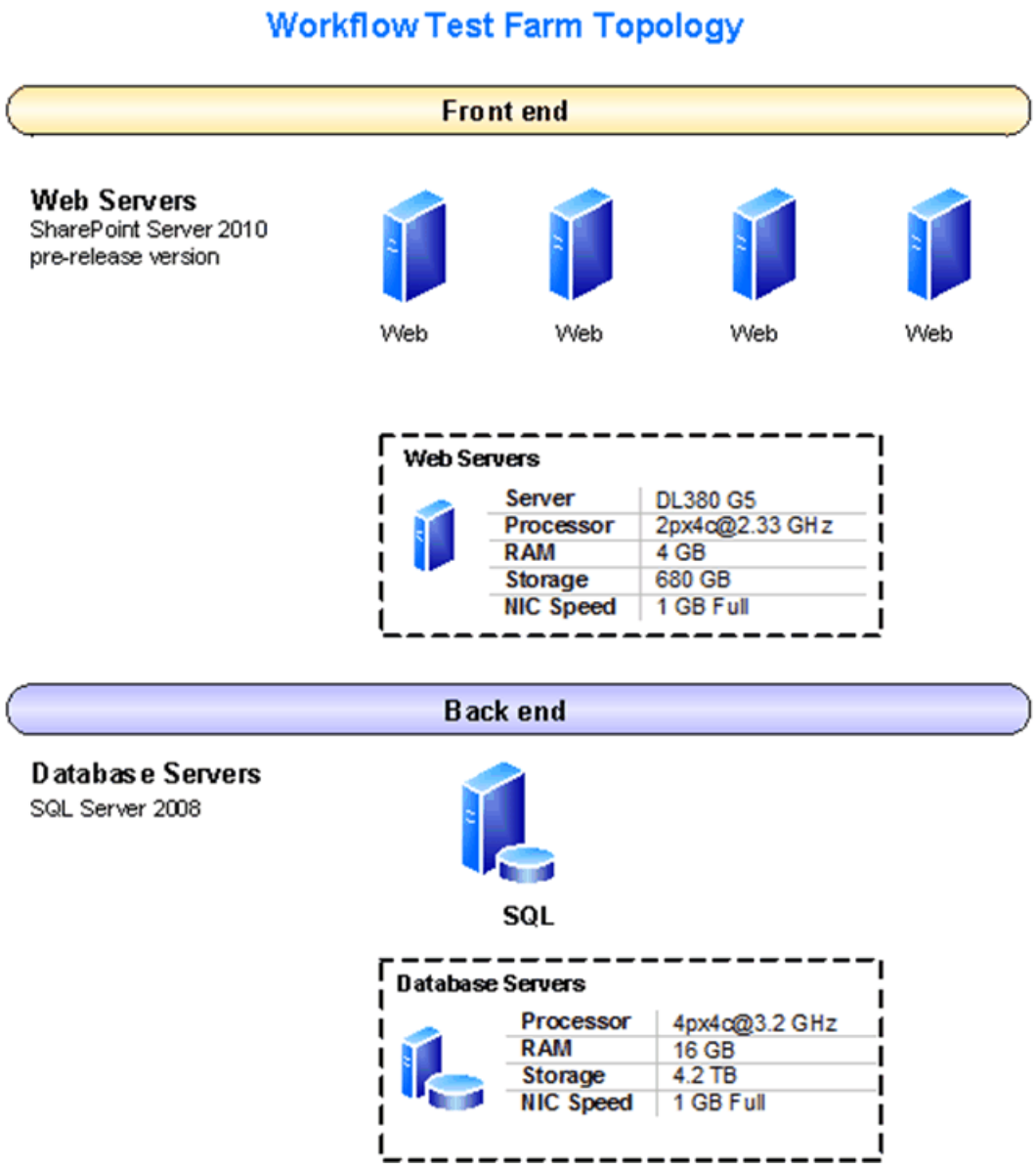
results can be used to estimate similar scenarios on those deployments. The dataset that was used for these tests contains a single site collection with a single site that is based on the Team Site template on a single content database.

To provide a high level of test-result detail, several farm configurations were used for testing. Farm configurations ranged from one to four Web servers and a single computer that is running Microsoft SQL Server 2008. Testing was performed with one client computer. The database server and all Web servers were 64-bit, and the client computer was 32-bit.

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

	Web server	**Database server**
Processor	2px4c@2.33GHz	4px4c@2.4GHz
RAM	4 GB	16 GB
Operating system	Windows Server 2008 R2 x64	Windows Server 2008 R2 x64
Storage	680 GB	4.2 terabyte
Number of network adapters	2	2
Network adapter speed	1 gigabit	1 gigabit
Authentication	NTLM	NTLM
Software version	4747	SQL Server 2008 R1
Number of SQL Server instances	1	1
Load balancer type	No load balancer	No load balancer
ULS logging level	Medium	Medium

Workflow Capacity Planning Topology



Test results

The following tables show the test results for workflow in SharePoint Server 2010. For each group of tests, only certain specific variables are changed to show the progressive effect on farm performance.

All the tests reported 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. By contrast, in the test, each operation was immediately followed by

the next operation, which resulted in a continual load on the farm. This load can cause database contention and other factors that can adversely affect performance.

Effect of scaling the Web server on throughput

The following throughput tests were run by using the Approval workflow that is included with SharePoint Server 2010. The workflow association assigns one task, and all instances are run on a single list. Each instance of this workflow creates the following in the content database:

- An entry in the Workflows table to store workflow status
- Five secondary list items (one task and four history items)
- Four event receivers to handle events on the workflow's parent item and task

Workflow Postpone Threshold was set to be very large so that workflow operations would never get queued. Each test was run five times, and each test ran for five minutes.

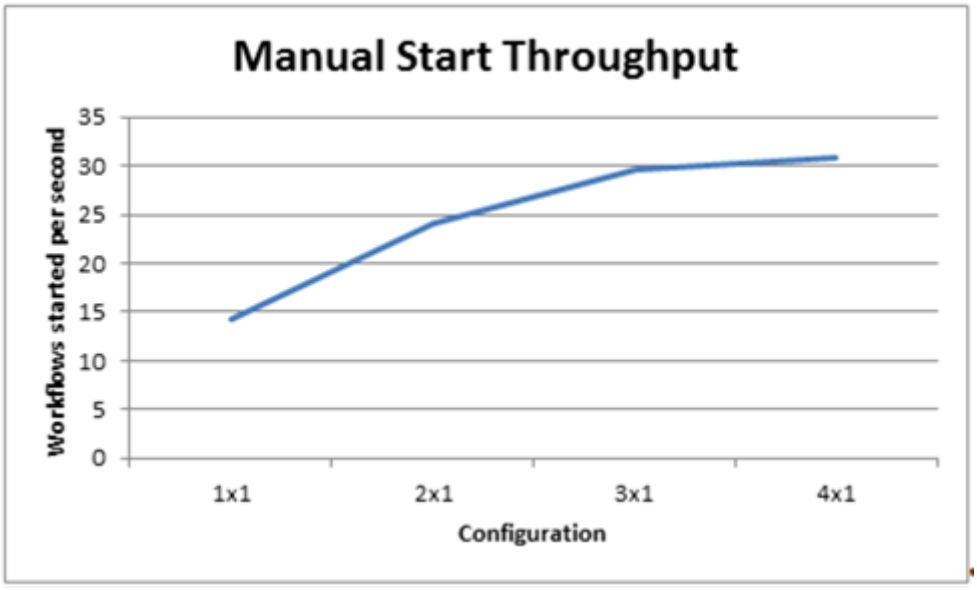
Manual start throughput

The test in the following table shows how the addition of front-end servers affects the throughput of starting workflows synchronously through the Web service. This test was run with a user load of 25 concurrent users continuously calling the StartWorkflow method on Workflow.asmx and no other load on the farm. The user load was the optimal load before dropped Web requests occurred. The list is prepopulated with up to 8,000 items.

Topology	Approval workflow maximum RPS
1x1	14.35
2x1	24.08
3x1	29.7
4x1	30.77

The following graph shows how throughput changes. The addition of front-end servers does not necessarily affect farm throughput in a linear manner but instead peaks off at around three to four front-end servers. In summary, the maximum throughput for manually starting workflows is around 30 workflows per second, and adding more than four front-end servers will likely have an insignificant effect.

Manual start throughput



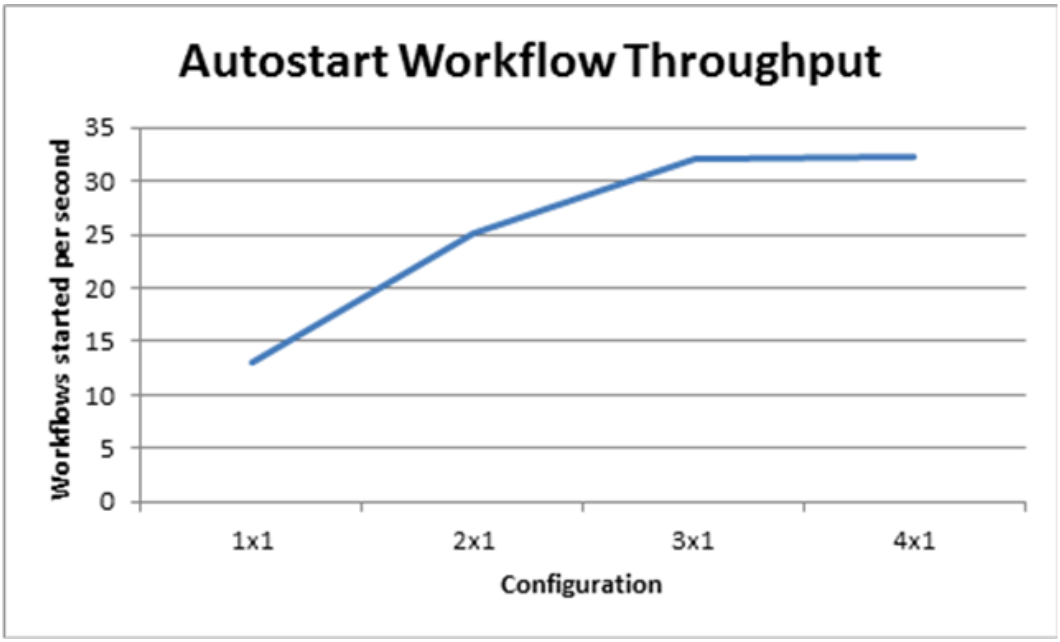
Automatically starting workflows when items are created throughput

The test in the following table shows how the addition of front-end servers affects the throughput of starting workflows automatically when items are created. This test was run with a user load of 150 concurrent users continuously calling the list Web service to create new list items in a single list and no other operations on the server. The list started as an empty list.

Topology	Approval workflow maximum RPS
1x1	13.0
2x1	25.11
3x1	32.11
4x1	32.18

The following graph shows how throughput changes. The throughput is very close to the manual start operations. Similar to the manual start test, throughput peaks at approximately three or four front-end servers at approximately 32 workflows per second maximum. Adding more than three or four front-end servers will have an insignificant effect.

Autostart workflow throughput



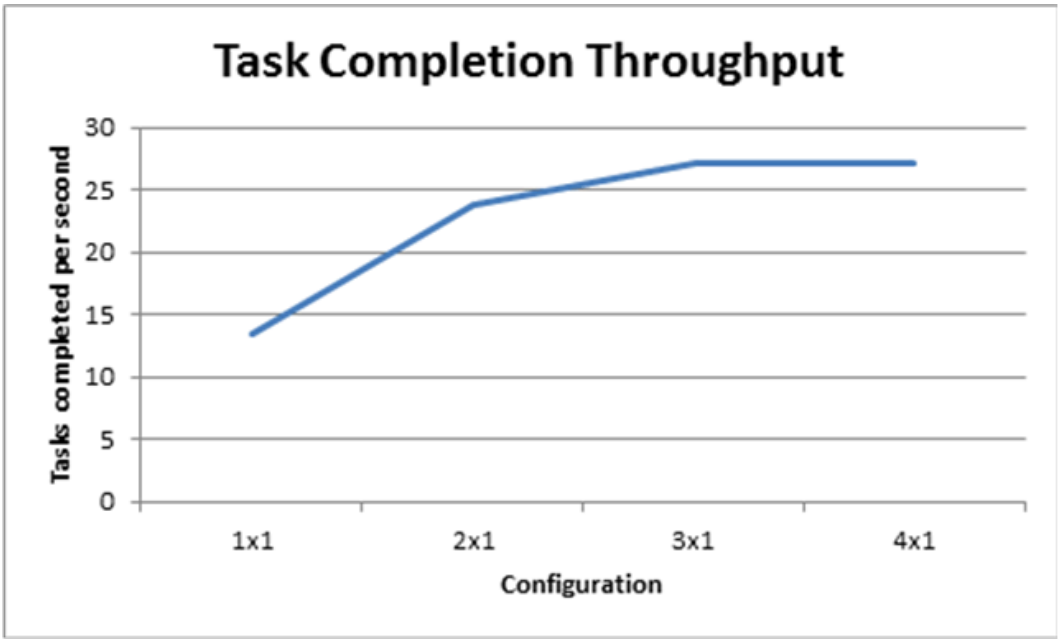
Task completion throughput

The test in the following table shows how the addition of front-end servers affects the throughput of completing workflow tasks. The task list that was used by autostart workflows in the previous test was the list that was used to complete tasks. This test was run with a user load of 25 concurrent users continuously calling the AlterToDo method on Workflow.asmx and no other operations on the server. The list started as an empty list.

Topology	Approval workflow maximum RPS
1x1	13.5
2x1	23.86
3x1	27.06
4x1	27.14

The following graph shows how throughput changes. Similar to the manual start test, throughput peaks at approximately three or four front-end servers at approximately 32 workflows per second maximum. Adding more than three front-end servers will have an insignificant effect.

Task completion throughput



Effect of list size and number of workflow instances on throughput

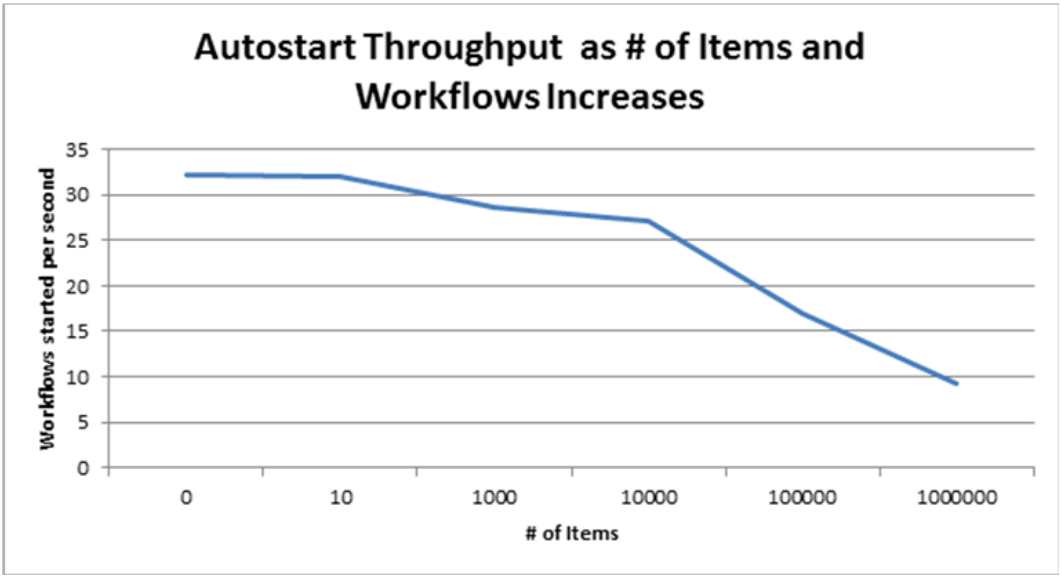
The test in the following table shows how throughput changes as list size and number of workflows increases. Data population was done by running the autostart workflow test continuously until 1 million items were created in the list, and stopping at different checkpoints throughout the test to perform throughput measurements as we did with the core throughput tests. Tests were performed on a 4x1 topology.

To maintain reliability during data population, we had to keep workflow queuing turned on to avoid reaching the maximum number of connections on the database server. If no connections are available and a workflow operation cannot connect to the content database, the operation will be unable to run. See Recommendations for more information about workflow queuing.

Number of items or workflows	Baseline solution maximum (RPS)
0	32.18
10	32
1,000	28.67
10,000	27.16
100,000	16.98

Number of items or workflows	Baseline solution maximum (RPS)
1,000,000	9.27

Autostart throughput as number of items and workflows increases



For a single list and single task and history list, throughput decreases steadily between 1,000 and 100,000 items. However, the rate of degradation reduces after that point. We attribute degradation of throughput to many factors.

One factor is the number of rows added to many tables in the content database per instance. As mentioned earlier, workflows create several list items in addition to event receivers that each workflow instance registers. As table sizes grow large in different scopes, adding rows becomes slower, and the aggregate slowdown for these additions becomes a more significant degradation than what is experienced with only list item creation.

Task list size contributes additional overhead. In comparing throughput for workflows run on new lists versus new task lists, task lists had a bigger effect on performance. This is because task lists register for more event receivers than the parent list items. The following chart describes the differences.

Throughput with different list configurations (workflows started per second)	Million item task list	Empty task list
Million item list	9.27	12
Empty item list	9.3	13

If you know that you will have to run lots of workflows against large lists and need more throughput than what your tests show you can get, consider whether your task lists can be separated between workflow associations.

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 to decide whether you have to scale out or scale up the starting topology.

For specific information about minimum and recommended system requirements, see Hardware and software requirements (SharePoint Server 2010).

Scaled-out topologies

You can increase workflow throughput by scaling out to up to four Web servers. Then, additional increase will be insignificant. Workflow throughput can be restricted by performance-related workflow settings. These settings are described in more detail in Workflow queuing and performance-related settings.

Estimating throughput targets

Many factors can affect throughput. These factors include the number of users, and the type, complexity, and frequency of user operations. More complex workflows that perform many operations against the content database or register for more events will run slower and consume more resources than other workflows.

The workflow used in this test creates several entries in the content database that are built in to the task activities. If you expect to have workflows with small numbers of tasks, you can expect similar throughput characteristics. If most workflows contain very lightweight operations, throughput may be increased. If your workflows will consist of lots of tasks or intense back-end operations or processing power, you can expect throughput to decrease.

In addition to understanding what the workflows will do, consider where the workflows will run and whether they will run against large lists, on which throughput will decrease over time.

SharePoint Server 2010 can be deployed and configured in many 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.

Workflow queuing and performance-related settings

Workflow uses a queuing system to control workflow-related stress on farm resources and the content database. By using this system, when the number of workflows executing against a database reaches

an administrator-configured threshold, successive workflow operations are added to the queue to be run by the Workflow Timer service. By default, this service receives a batch of workflow work items through timer jobs every minute.

Several farm administrator settings directly and indirectly related to the queuing mechanism affect the performance and scaling for workflow. The following sections describe what these settings do and how to adjust them to meet performance requirements.

Understanding the basic queue settings

Farm administrators can adjust the following settings to configure basic characteristics of the queuing system:

- Workflow Postpone Threshold (Set-SPFarmConfig –WorkflowPostponeThreshold <integer>)

  The maximum number of workflows that can execute against a single content database before additional requests and operations are queued. Queued workflows show a status of Starting. This is a farm-wide setting that has a default value of 15. This represents the number of workflow operations that are being processed at a time, not the maximum number of workflows that can be in progress. As workflow operations are completed, successive operations will be able to run.

- Workflow Event Delivery Batch Size (Set-SPWorkflow –BatchSize <integer>)

  The Workflow Timer service is an exception to the postpone threshold limit and will retrieve batches of items from the queue and execute them one at a time. These batches can be larger than the postpone threshold. The number of work items that the service receives per run is set by using the **BatchSize** property. The **BatchSize** property can be set one time per service instance. The default value is 100. When running on application servers that are not configured to be front-end servers, the Workflow Timer service requires workflow configuration settings in Web.config to be set in the configuration database. This must be done through a script that calls UpdateWorkflowConfigurationSettings() on the SPWebApplication object, which will copy the Web.config settings from a front-end server.

- Workflow Timer Job Frequency (Set-SPTimerJob job-workflow –schedule <string>)

  The frequency with which the Workflow Timer service runs can be adjusted through timer job settings. By default, the service is set to run every five minutes. This means that there can be a five-minute delay before the work items at the top of the queue are processed.

  Note:
  Scheduled work items such as task due date expirations are also picked up by the same timer mechanism. Therefore, they will be delayed by the same time interval.

The Workflow Timer service can be turned off on each server by using Shared Services Administration in Central Administration. By default, it will run on every front-end server in the farm. Each job will iterate through all the Web applications and content databases in the farm.

The combination of the postpone threshold, batch size, and timer frequency can be used to limit workflow operations against the database. Maximum throughput will be affected by how quickly operations get queued and processed from the queue.

For example, with the default settings, a single timer service, and a single content database, if there are 1,000 items in the queue, it will take ten timer job runs to execute them all, which will take 50 minutes to execute. However, if you set the batch size to 1,000 and set the timer job to run every minute, the operations would all begin execution after a minute. If you set the postpone threshold higher, more operations will run synchronously, reducing the number of requests that get queued and reducing the total time that is required to process those workflows.

Note:

We recommend setting the postpone threshold no larger than 200, because concurrent workflow instances run in their own threads and will each open new SQL Server connections, potentially hitting the maximum connection limits on the database server over time.

If you do not want workflows running on front-end servers and know that operations do not have to occur immediately, you can isolate the Workflow Timer service to run on select application servers, set the postpone threshold to a very low number to force workflows to usually run in the timer service, and set the batch size large so that it receives items more quickly and frequently. If you want to make sure workflows run more synchronously across the system, set the postpone threshold larger so that workflows are not postponed often and have a more immediate effect.

Modify these settings to optimize for how you want workflows to operate. We recommend experimenting with different settings and testing them to optimize them for your environments and requirements.

Adjusting settings for queuing

If the farm will sustain heavy workflow load for long periods of time or there will be many delay events queued from workflows in the system, the number of queued workflow operations will grow. In addition to the basic queue settings, you may have to tune the following settings to keep the queue in good health:

- Work Item Event Delivery Batchsize

  The table that workflow uses for queued events is a general work item table that is shared with other non-workflow features in SharePoint Server 2010. Thus, there is another timer job that dequeues non-workflow work items. Similar to the workflow event delivery batch size, the work item event delivery batch size specifies the number of non-workflow work items that are dequeued at a time.

- Workflow Failover Timer Job Frequency

  In rare circumstances, if workflow events cannot be delivered to a workflow instance, the event delivery will be scheduled on the queue as a failover work item to be retried later (starting with 5 minutes later, and then 10 minutes if it fails again, and then 20 minutes, and so on). A workflow failover timer job dequeues failover work items, and this setting adjusts the frequency at which the failover timer will run. By default, this runs every 15 minutes.

- Workflow Failover Batchsize

Similar to the workflow and work item batch size settings, this setting controls the number of failover events that each failover timer job will dequeue.

Because there are many timer jobs that operate on the same table, lots of queued items can cause database contention and reduce throughput and reliability. To reduce contention, we recommend the following:

- Balance Postpone Threshold and Workflow Batchsize so that batch size is small enough or timer job frequency high enough that a timer job can be completed before the next timer job starts in order to avoid building up too many parallel timer job runs that cannot finish.
- To avoid table locks, do not set either of the two batch size settings larger than 5,000.

Tip:
Offset the frequency of the workflow, work item, and failover timer jobs so that they are not always executing at the same times. To get a large list that has workflows, four minutes for the workflow timer job and six minutes for the failover worked well in our data population scripts.

Improving scaling for task and history lists

Workflows generate many tasks and history items per instance. By default, these lists are indexed to help with scaling, but as these lists grow, performance will always decrease. To reduce the rate of the decrease, keep separate history and task lists for different workflow associations, and periodically change these lists in the workflow association settings as lists become large.

Workflow also has a daily timer job (job-workflow-autoclean) that will automatically delete workflow instances and tasks for instances that have been finished for more than 60 days. Leave this timer job on to keep the task lists and events on the task list as clean as possible. If data must be preserved, write the data to other lists or archive locations. Workflow history items are not deleted by this timer job. If you have to clean these up, this should be done with a script or manually through the list user interface.

Other considerations

Removing columns on lists causes a database operation proportional to the number of items in the list. Removing workflow associations will remove the workflow status column from the list. This causes a large operation on large lists. If you know that a list has millions of items, set the workflow to **No New Instance** instead of removing workflows.

Troubleshooting

Bottleneck	Cause	Resolution
Database contention (locks)	Database locks prevent multiple users from	To help reduce the incidence of database locks, you can do the following:

Bottleneck	Cause	Resolution
	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 change that same set of data until the first user or process is complete, changing the data and relinquishing the lock.	• Distribute workflows to more document libraries. • Scale up the database server. • Tune the database server hard disk for read/write. Methods exist to circumvent the database locking system in SQL Server 2005, such as the NOLOCK parameter. However, we do not recommend or support use of this method because of the possibility of data corruption.
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) 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 Web servers to the farm to distribute user load, or you can scale up the Web server or servers by adding faster processors.

Web servers

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

Performance counter	Apply to object	Notes
Processor time	Total	Shows the percentage of elapsed time that this thread

Performance counter	Apply to object	Notes
		used the processor to execute instructions.
Memory utilization	Application pool	Shows the average utilization of system memory for the application pool. You must determine the correct application pool to monitor. The basic guideline is to determine peak memory utilization for a given Web application, and assign that number plus 10 to the associated application pool.

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 SharedServices.mdf	Average values larger than 1.5 per spindle indicate that the write times for that hard disk are insufficient.
Processor time	SQL Server process	Average values larger 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

Workflow Scalability and Performance in Windows SharePoint Services 3.0 (http://go.microsoft.com/fwlink/?LinkId=207353)

Storage and SQL Server capacity planning and configuration (SharePoint Server 2010)

This article describes how to plan for and configure the storage and Microsoft SQL Server database tier in a Microsoft SharePoint Server 2010 environment.

The capacity planning information in this document provides guidelines for you to use in your planning. It is based on testing performed at Microsoft on live properties. However, your results may vary based on the equipment you use and the features and functionality that you implement for your sites.

Because SharePoint Server 2010 often runs in environments in which databases are managed by separate SQL Server database administrators, this document is intended for joint use by SharePoint Server 2010 farm implementers and SQL Server database administrators. It assumes significant understanding of both SharePoint Server 2010 and SQL Server.

This article assumes that you are familiar with the concepts presented in Capacity management and sizing for SharePoint Server 2010.

Design and configuration process for SharePoint 2010 Products storage and database tier

We recommend that you break the storage and database tier design process into the following steps. Each section provides detailed information about each design step, including storage requirements and best practices:

- Gather storage and SQL Server space and I/O requirements
- Choose SQL Server version and edition
- Design storage architecture based on capacity and I/O requirements
- Estimate memory requirements
- Understand network topology requirements
- Configure SQL Server
- Validate and monitor storage and SQL Server performance

Gather storage and SQL Server space and I/O requirements

Several SharePoint Server 2010 architectural factors influence storage design. The amount of content, features and service applications used, number of farms, and availability needs are key factors.

Before you start to plan storage, you should understand the databases that SharePoint Server 2010 can use.

In this section:

- Databases used by SharePoint 2010 Products
- Understand SQL Server and IOPS
- Estimate core storage and IOPS needs
- Estimate service application storage needs and IOPS
- Determine availability needs

Databases used by SharePoint 2010 Products

The databases that are installed with SharePoint Server 2010 depend on the features that are being used in the environment. All SharePoint 2010 Products environments rely on the SQL Server system databases. This section provides a summary of the databases installed with SharePoint Server 2010. For detailed information, see Database types and descriptions (SharePoint Server 2010) and Database model (http://go.microsoft.com/fwlink/?LinkId=187968).

Product version and edition	Databases
SharePoint Foundation 2010	Configuration Central Administration content Content (one or more) Usage and Health Data Collection Business Data Connectivity Application Registry service (if upgrading from Microsoft Office SharePoint Server 2007 Business Data Catalog) Subscription Settings service (if it is enabled through Windows PowerShell)
Additional databases for SharePoint Server 2010 Standard edition	Search service application: • Search administration • Crawl (one or more) • Properties (one or more) User Profile service application: • Profile • Synchronization • Social tagging

Product version and edition	Databases
	Web analytics service application • Staging • Reporting Secure store State Managed Metadata Word Automation services
Additional databases for SharePoint Server 2010 Enterprise edition	PerformancePoint
Additional databases for Project Server 2010	Draft Published Archive Reporting
Additional database for FAST Search Server	Search administration

If you are integrating more fully with SQL Server, your environment may also include additional databases, as in the following scenarios:

- Microsoft SQL Server 2008 R2 PowerPivot for Microsoft SharePoint 2010 can be used in a SharePoint Server 2010 environment that includes SQL Server 2008 R2 Enterprise Edition and SQL Server Analysis Services. If in use, you must also plan to support the PowerPivot Application database, and the additional load on the system. For more information, see Plan a PowerPivot deployment in a SharePoint farm (http://go.microsoft.com/fwlink/?LinkID=186698).
- The Microsoft SQL Server 2008 Reporting Services (SSRS) plug-in can be used with any SharePoint 2010 Products environment. If you are using the plug-in, plan to support the two SQL Server 2008 Reporting Services databases and the additional load that is required for SQL Server 2008 Reporting Services.

Understand SQL Server and IOPS

On any server that hosts SQL Server, it is very important that the server achieve the fastest response possible from the I/O subsystem.

More and faster disks or arrays provide sufficient I/O operations per second (IOPS) while maintaining low latency and queuing on all disks.

Slow response from the I/O subsystem cannot be compensated for by adding other types of resources such as CPU or memory; however, it can influence and cause issues throughout the farm. Plan for minimal latency before deployment, and monitor your existing systems.

Before you deploy a new farm, we recommend that you benchmark the I/O subsystem by using the SQLIO disk subsystem benchmark tool. For details, see SQLIO Disk Subsystem Benchmark Tool (http://go.microsoft.com/fwlink/?LinkID=105586).

For detailed information about how to analyze IOPS requirements from a SQL Server perspective, see Analyzing I/O Characteristics and Sizing Storage Systems for SQL Server Database Applications (http://sqlcat.com/whitepapers/archive/2010/05/10/analyzing-i-o-characteristics-and-sizing-storage-systems-for-sql-server-database-applications.aspx).

Estimate core storage and IOPS needs

Configuration and content storage and IOPs are the base layer that you must plan for in every SharePoint Server 2010 deployment.

Configuration storage and IOPS

Storage requirements for the Configuration database and the Central Administration content database are not large. We recommend that you allocate 2 GB for the Configuration database and 1 GB for the Central Administration content database. Over time, the Configuration database may grow beyond 1 GB, but it does not grow quickly — it grows by approximately 40 MB for each 50,000 site collections.

Transaction logs for the Configuration database can be large, therefore we recommend that you change the recovery model for the database from full to simple.

Note:
If you want to use SQL Server database mirroring to provide availability for the Configuration database, you must use the full recovery model.

IOPS requirements for the Configuration database and Central Administration content database are minimal.

Content storage and IOPS

Estimating the storage and IOPS required for content databases is not a precise activity. In testing and explaining the following information, we intend to help you derive estimates to use for determining the initial size of your deployment. However, when your environment is running, we expect that you will revisit your capacity needs based on the data from your live environment.

For more information about our overall capacity planning methodology, see Capacity management and sizing for SharePoint Server 2010.

Estimate content database storage

The following process describes how to approximately estimate the storage required for content databases, without considering log files:

1. Calculate the expected number of documents. This value is referred to as in the formula.

   How you calculate the number of documents will be determined by the features that you are using. For example, for My Site Web sites or collaboration sites, we recommend that you calculate the expected number of documents per user and multiply by the number of users. For records management or content publishing sites, you may calculate the number of documents that are managed and generated by a process.

   If you are migrating from a current system, it may be easier to extrapolate your current growth rate and usage. If you are creating a new system, review your existing file shares or other repositories and estimate based on that usage rate.
2. Estimate the average size of the documents that you will be storing. This value is referred to as in the formula. It may be worthwhile to estimate averages for different types or groups of sites. The average file size for My Site Web sites, media repositories, and different department portals can vary significantly.
3. Estimate the number of list items in the environment. This value is referred to as in the formula.

   List items are more difficult to estimate than documents. We generally use an estimate of three times the number of documents (), but this will vary based on how you expect to use your sites.
4. Determine the approximate number of versions. Estimate the average number of versions any document in a library will have (this value will usually be much lower than the maximum allowed number of versions). This value is referred to as in the formula.

   The value of must be above zero.
5. Use the following formula to estimate the size of your content databases:

   Database size = ((×) ×) + (10 KB × (+ (×)))

   The value of 10 KB in the formula is a constant that roughly estimates the amount of metadata required by SharePoint Server 2010. If your system requires significant use of metadata, you may want to increase this constant.

As an example, if you were to use the formula to estimate the amount of storage space required for the data files for a content database in a collaboration environment with the following characteristics, you would need approximately 105 GB.

Input	**Value**
Number of documents ()	200,000 Calculated by assuming 10,000 users times 20 documents
Average size of documents ()	250 KB
List items ()	600,000
Number of non-current versions ()	2 Assuming that the maximum versions allowed is

Input	Value
	10

Database size = (((x)) ×) + ((KB × (+ (x))) = KB or GB

Features that influence the size of content databases

The use of the following SharePoint Server 2010 features can significantly affect the size of content databases:

- **Recycle bins** Until a document is fully deleted from both the first stage and second stage recycle bin, it occupies space in a content database. Calculate how many documents are deleted each month to determine the effect of recycle bins on the size of content databases. For more information, see Configure Recycle Bin settings (SharePoint Server 2010).
- **Auditing** Audit data can quickly compound and use large amounts of space in a content database, especially if view auditing is turned on. Rather than letting audit data grow without restraint, we recommend that you only enable auditing on the events that are important to meet regulatory needs or internal controls. Use the following guidelines to estimate the space you will need to reserve for auditing data:
  - Estimate the number of new auditing entries for a site, and multiply this number by 2 KB (entries generally are limited to 4 KB, with an average size of about 1 KB).
  - Based on the space that you want to allocate, determine the number of days of audit logs you want to keep.
- Office Web Apps. If Office Web Apps are being used, the Office Web Apps cache can significantly affect the size of a content database. By default, the Office Web Apps cache is configured to be 100 GB. For more information about the size of the Office Web Apps cache, see Manage the Office Web Apps cache.

Estimate content database IOPS requirements

IOPS requirements for content databases vary significantly based on how your environment is being used, and how much disk space and how many servers you have. In general, we recommend that you compare the predicted workload in your environment to one of the solutions that we tested. For more information, see Performance and capacity test results and recommendations (SharePoint Server 2010).

Important:
The testing for the content in this section is not yet complete. Check back for additional information.

Estimate service application storage needs and IOPS

After estimating content storage and IOPs needs, you must next determine the storage and IOPs required by the service applications that are being used in your environment.

SharePoint Foundation 2010 service application storage and IOPS requirements

To estimate the storage requirements for the service applications in the system, you must first be aware of the service applications and how you will use them. Service applications that are available in SharePoint Foundation 2010 that have databases are listed in the following table.

Service application database	Size estimation recommendation
Usage and Health Data Collection	The Usage database can grow very quickly and require significant IOPS. For example, in collaborative environments that use out-of-the-box settings, 1 million HTTP requests require 2 GB of storage. Use one of the following formulas to estimate the amount of IOPS required: • 115 × page hits/second • 5 × HTTP requests If you must restrict the size of the usage database, we recommend that you start by logging only page requests. You can also restrict the size of the database by setting the default interval of data to be kept to be less than two weeks. If possible, put the Usage database on its own disk or spindle.
Business Data Connectivity service	The size of the Business Data Connectivity services database is primarily affected by the number of external content types that you plan to support. Allocate 0.5 MB for each external content type. If you don't know how many external content types you might need, we recommend that you allocate 50 MB. IOPS requirements are minimal.
Application Registry service	Allocate 1 GB only if you are upgrading from the Microsoft Office SharePoint Server 2007 Business Data Catalog. IOPS requirements are minimal.
Subscription settings	Allocate 1 GB. IOPS requirements are minimal

SharePoint Server 2010 service application storage and IOPs requirements

To estimate the storage requirements for the service applications in the system, you must first be aware of the service applications and how you will use them. Service applications that are available in SharePoint Server 2010 that have databases are listed in the following table.

Service application	Size estimation recommendation
Search	Search requires three databases. Your environment may include multiple Property and Crawl databases. The Search administration database is typically small: allocate 10 GB. To estimate the required storage for your Property and Crawl databases, use the following multipliers: • Crawl: 0.046 × (sum of content databases) • Property: 0.015 × (sum of content databases) The IOPS requirements for Search are significant. • For the Crawl database, search requires from 3,500 to 7,000 IOPS. • For the Property database, search requires 2,000 IOPS. For detailed information about how to estimate capacity required for Search, see Performance and capacity test results and recommendations (SharePoint Server 2010).
User Profile	The User Profile service application is associated with three databases: Profile, Synch, and Social Tagging. To estimate the required storage for the databases, use the following information: • Profile. With out-of-the-box settings, in an environment configured to use Active Directory, the profile database requires approximately 1 MB per user profile. • Synchronization. With out-of-the-box settings, in an environment that has few groups per user, the synch database requires approximately 630 KB per user profile. 90% of the space will be used by the data file.

<table>
<tr><th>Service application</th><th>Size estimation recommendation</th></tr>
<tr><td></td><td>

Social tagging. With out-of-the-box settings, the social tagging database requires approximately 0.009 MB per tag, comment, or rating. To estimate how many tags and notes users will create, consider the following information about the site del.icio.us:

Approximately 10% of users are considered active.
Active users create 4.5 tags and 1.8 comments per month.

In a live collaboration environment with 160,000 user profiles, 5 groups, 79,000 tags, comments and ratings (2,500 comments, 76,000 tags, and 800 ratings), and out-of-the-box settings, we saw the following sizes for these databases:
<table>
<tr><th>Database name</th><th>Database size</th></tr>
<tr><td>Profile</td><td>155 GB</td></tr>
<tr><td>Synchronization</td><td>96 GB</td></tr>
<tr><td>Social tagging</td><td>0.66 GB</td></tr>
</table>
</td></tr>
<tr><td>Managed metadata</td><td>The Managed Metadata service application has one database. The size of the database is affected by the number of content types and keywords used in the system. Many environments will include multiple instances of the Managed Metadata service application. For detailed information about how to estimate the size and IOPS requirements for this database, see Performance and capacity test results and recommendations (SharePoint Server 2010).</td></tr>
<tr><td>Web Analytics</td><td>Web Analytics has two databases: Staging and Reporting. Many factors influence the size of the databases. They include retention period, the daily volume of data being tracked, and the number of</td></tr>
</table>

Service application	Size estimation recommendation
	site collections, sites, and subsites in the Web application being analyzed. For detailed information about how to estimate their sizing and IOPS requirements, see Performance and capacity test results and recommendations (SharePoint Server 2010).
Secure store	The size of the Secure Store service application database is determined by the number of credentials in the store and the number of entries in the audit table. We recommend that you allocate 5 MB for each 1,000 credentials for it. It has minimal IOPS.
State	The State service application has one database. We recommend that you allocate 1 GB for it. It has minimal IOPS.
Word Automation service	The Word Automation service application has one database. We recommend that you allocate 1 GB for it. It has minimal IOPS.
PerformancePoint	The PerformancePoint service application has one database. We recommend that you allocate 1 GB for it. It has minimal IOPS.

Determine availability needs

Availability is the degree to which a SharePoint Server 2010 environment is perceived by users to be available. An available system is a system that is resilient — that is, incidents that affect service occur infrequently, and timely and effective action is taken when they do occur.

Availability requirements can significantly increase your storage needs. For detailed information, see Plan for availability (SharePoint Server 2010).

Choose SQL Server version and edition

Although SharePoint 2010 Products can run on Microsoft SQL Server 2008 R2, SQL Server 2008, or SQL Server 2005, we strongly recommend that you consider running your environment on the Enterprise Edition of SQL Server 2008 or SQL Server 2008 R2 to take advantage of the additional performance, availability, security, and management capabilities that it provides. For more information

about the benefits of using SQL Server 2008 R2 Enterprise Edition, see SQL Server 2008 R2 and SharePoint 2010 Products: Better Together (white paper) (SharePoint Server 2010).

In particular, you should consider your need for the following features:

- **Backup compression** Backup compression can speed up any SharePoint backup, and is available in SQL Server 2008 Enterprise Edition or SQL Server 2008 R2 Standard edition. By setting the compression option in your backup script, or by configuring the server that is running SQL Server to compress by default, you can significantly reduce the size of your database backups and shipped logs. For more information, see Backup Compression (SQL Server) (http://go.microsoft.com/fwlink/?LinkId=129381&clcid=0x409).

  Note:
  SQL Server data compression is not supported for SharePoint 2010 Products.

- **Transparent data encryption** If your security requirements include the need for transparent data encryption, you must use SQL Server Enterprise Edition.
- **Web Analytics service application** If you plan to use the Web Analytics service application for significant analysis, consider SQL Server Enterprise Edition so that the system can take advantage of table partitioning.
- **Content deployment** If you plan to use the content deployment feature, consider SQL Server Enterprise Edition so that the system can take advantage of SQL Server database snapshots.
- **Remote BLOB storage** If you want to take advantage of remote BLOB storage to a database or location outside the files associated with each content database, you must use SQL Server 2008 or SQL Server 2008 R2 Enterprise Edition.
- **Resource governor** Resource Governor is a technology introduced in SQL Server 2008 that enables you to manage SQL Server workloads and resources by specifying limits on resource consumption by incoming requests. Resource Governor enables you to differentiate workloads and allocate CPU and memory as they are requested, based on the limits that you specify. It is available only in SQL Server 2008 or SQL Server 2008 R2 Enterprise edition. For more information about using Resource Governor, see Managing SQL Server Workloads with Resource Governor.

  We recommend that you use Resource Governor with SharePoint Server 2010 to:

  - Limit the amount of SQL Server resources that the Web servers targeted by the search crawl component consume. As a best practice, we recommend limiting the crawl component to 10 percent CPU when the system is under load.
  - Monitor how many resources are consumed by each database in the system — for example, you can use Resource Governor to help you determine the best placement of databases among computers that are running SQL Server.

- **PowerPivot for SharePoint 2010** Enables users to share and collaborate on user-generated data models and analysis in Excel and in the browser while automatically refreshing those analyses. It is part of SQL Server 2008 R2 Enterprise Edition Analysis Services.

Design storage architecture based on capacity and I/O requirements

The storage architecture and disk types that you select for your environment can affect system performance.

In this section:

Choose a storage architecture

Direct Attached Storage (DAS), Storage Area Network (SAN), and Network Attached Storage (NAS) storage architectures are supported with SharePoint Server 2010, although NAS is only supported for use with content databases that are configured to use remote BLOB storage. Your choice depends on factors within your business solution and your existing infrastructure.

Any storage architecture must support your availability needs and perform adequately in IOPS and latency. To be supported, the system must consistently return the first byte of data within 20 milliseconds (ms).

Direct Attached Storage (DAS)

DAS is a digital storage system that is directly attached to a server or workstation, without a storage network in between. DAS physical disk types include Serial Attached SCSI (SAS) and Serial Attached ATA (SATA).

In general, we recommend that you choose a DAS architecture when a shared storage platform cannot guarantee a response time of 20 ms and sufficient capacity for average and peak IOPs.

Storage Area Network (SAN)

SAN is an architecture to attach remote computer storage devices (such as disk arrays and tape libraries) to servers in such a way that the devices appear as locally attached to the operating system (for example, block storage).

In general, we recommend that you choose a SAN when the benefits of shared storage are important to your organization.

The benefits of shared storage include the following:

- Easier to reallocate disk storage between servers.
- Can serve multiple servers.
- No limitations on the number of disks that can be accessed.

Network Attached Storage (NAS)

A NAS unit is a self-contained computer that is connected to a network. Its sole purpose is to supply file-based data storage services to other devices on the network. The operating system and other software on the NAS unit provide the functionality of data storage, file systems, and access to files, and the management of these functionalities (for example, file storage).

Note:
NAS is only supported for use with content databases that are configured to use remote BLOB storage. Any network storage architecture must respond to a ping within 1 ms and must return the first byte of data within 20 ms. This restriction does not apply to the local SQL Server FILESTREAM provider, because it only stores data locally on the same server.

Choose disk types

The disk types that you use in the system can affect reliability and performance. All else being equal, larger drives increase mean seek time. SharePoint Server 2010 supports the following types of drives:

- Small Computer System Interface (SCSI)
- Serial Advanced Technology Attachment (SATA)
- Serial-attached SCSI (SAS)
- Fibre Channel (FC)
- Integrated Device Electronics (IDE)
- Solid State Drive (SSD) or Flash Disk

Choose RAID types

RAID (Redundant Array of Independent Disks) is often used to both improve the performance characteristics of individual disks (by striping data across several disks) and to provide protection from individual disk failures.

All RAID types are supported for SharePoint Server 2010; however, we recommend that you use RAID 10 or a vendor-specific RAID solution that has equivalent performance.

When you configure a RAID array, make sure that you align the file system to the offset that is supplied by the vendor. In the absence of vendor guidance, refer to SQL Server Predeployment I/O Best Practices (http://go.microsoft.com/fwlink/?LinkID=105583).

For more information about provisioning RAID and the SQL Server I/O subsystem, see SQL Server Best Practices Article (http://go.microsoft.com/fwlink/?LinkId=168612).

Estimate memory requirements

The memory required for SharePoint Server 2010 is directly related to the size of the content databases that you are hosting on a server that is running SQL Server.

As you add service applications and features, your requirements are likely to increase. The following table gives guidelines for the amount of memory we recommend.

Note:
Our definitions of small and medium deployments are those described in the "Reference Architectures" section of the article Capacity management and sizing for SharePoint Server 2010.

Combined size of content databases	RAM recommended for computer running SQL Server
Minimum for small production deployments	8 GB
Minimum for medium production deployments	16 GB
Recommendation for up to 2 terabytes	32 GB
Recommendation for the range of 2 terabytes to a maximum of 5 terabytes	64 GB

Note:
These values are higher than those recommended as the minimum values for SQL Server because of the distribution of data required for a SharePoint Server 2010 environment. For more information about SQL Server system requirements, see Hardware and Software Requirements for Installing SQL Server 2008 (http://go.microsoft.com/fwlink/?LinkId=129377).

Other factors that may influence the memory required include the following:

- The use of SQL Server mirroring.
- The frequent use of files larger than 15 megabytes (MB).

Understand network topology requirements

Plan the network connections within and between farms. We recommend that you use a network that has low latency.

The following list provides some best practices and recommendations:

- All servers in the farm should have LAN bandwidth and latency to the server that is running SQL Server. Latency should be no greater than 1 ms.
- We do not recommend a wide area network (WAN) topology in which a server that is running SQL Server is deployed remotely from other components of the farm over a network that has latency greater than 1 ms. This topology has not been tested.

- Plan for an adequate WAN network if you are planning to use SQL Server mirroring or log shipping to keep a remote site up-to-date.
- We recommend that Web servers and application servers have two network adapters: one network adapter to handle end user traffic and the other to handle communication with the servers running SQL Server.

Configure SQL Server

The following sections describe how to plan to configure SQL Server for SharePoint Server 2010.

In this section:

- Determine how many instances or servers are required
- Configure storage and memory
- Set SQL Server options
- Configure databases

Estimate how many servers are required

In general, SharePoint Server 2010 was designed to take advantage of SQL Server scale out — that is, SharePoint Server 2010 may perform better with a large number of medium-size servers that are running SQL Server than with only a few large servers.

Always put SQL Server on a dedicated server that is not running any other farm roles or hosting databases for any other application, unless you are deploying the system on a stand-alone server.

The following is general guidance for when to deploy an additional server that will run SQL Server:

- Add an additional database server when you have more than four Web servers that are running at full capacity.
- Add an additional database server when your content databases exceed 5 terabytes.

Note:
Microsoft supports server configurations that do not follow this guidance.

To promote secure credential storage when you are running the Secure Store service application, we recommend that the secure store database be hosted on a separate database instance where access is limited to one administrator.

Configure storage and memory

On the server that is running SQL Server 2008, we recommend that the L2 cache per CPU have a minimum of 2 MB to improve memory.

Follow vendor storage configuration recommendations

For optimal performance when you configure a physical storage array, adhere to the hardware configuration recommendations supplied by the storage vendor instead of relying on the default values of the operating system.

If you do not have guidance from your vendor, we recommend that you use the DiskPart.exe disk configuration utility to configure storage for SQL Server 2008. For more information, see Predeployment I/O Best Practices (http://go.microsoft.com/fwlink/?LinkID=105583&clcid=0x409).

Provide as many resources as possible

Ensure that the SQL Server I/O channels to the disks are not shared by other applications, such as the paging file and Internet Information Services (IIS) logs.

Provide as much bus bandwidth as possible. Greater bus bandwidth helps improve reliability and performance. Consider that the disk is not the only user of bus bandwidth — for example, you must also account for network access.

Set SQL Server options

The following SQL Server settings and options should be configured before you deploy SharePoint Server 2010.

- Do not enable auto-create statistics on a SQL Server that is supporting SharePoint Server 2010. SharePoint Server 2010 implements specific statistics, and no additional statistics are needed. Auto-create statistics can significantly change the execution plan of a query from one instance of SQL Server to another instance of SQL Server. Therefore, to provide consistent support for all customers, SharePoint Server 2010 provides coded hints for queries as needed to provide the best performance across all scenarios.
- To ensure optimal performance, we strongly recommend that you set **max degree of parallelism** to 1 for database servers that host SharePoint Server 2010 databases. For more information about how to set **max degree of parallelism**, see max degree of parallelism Option (http://go.microsoft.com/fwlink/?LinkId=189030).
- To improve ease of maintenance, configure SQL Server connection aliases for each database server in your farm. A connection alias is an alternative name that can be used to connect to an instance of SQL Server. For more information, see How to: Set a SQL Server Alias (SQL Server Management Studio) (http://go.microsoft.com/fwlink/?LinkId=132064&clcid=0x409).

Configure databases

The following guidance describes best practices to plan for as you configure each database in your environment.

Separate and prioritize your data among disks

Ideally, you should place the tempdb database, content databases, Usage database, search databases, and SQL Server 2008 transaction logs on separate physical hard disks.

The following list provides some best practices and recommendations for prioritizing data:

- When you prioritize data among faster disks, use the following ranking:
  a. Tempdb data files and transaction logs
  b. Database transaction log files
  c. Search databases, except for the Search administration database
  d. Database data files

  In a heavily read-oriented portal site, prioritize data over logs.
- Testing and customer data show that SharePoint Server 2010 farm performance can be significantly impeded by insufficient disk I/O for tempdb. To avoid this issue, allocate dedicated disks for tempdb. If a high workload is projected or monitored — that is, the average read operation or the average write operation requires more than 20 ms — you might have to ease the bottleneck by either separating the files across disks or by replacing the disks with faster disks.
- For best performance, place the tempdb on a RAID 10 array. The number of tempdb data files should equal the number of core CPUs, and the tempdb data files should be set at an equal size. Count dual core processors as two CPUs for this purpose. Count each processor that supports hyper-threading as a single CPU. For more information, see Optimizing tempdb Performance (http://go.microsoft.com/fwlink/?LinkID=148537).
- Separate database data and transaction log files across different disks. If files must share disks because the files are too small to warrant a whole disk or stripe, or you have a shortage of disk space, put files that have different usage patterns on the same disk to minimize simultaneous access requests.
- Consult your storage hardware vendor for information about how to configure all logs and the search databases for write optimization for your particular storage solution.

Use multiple data files for content databases

Follow these recommendations for best performance:

- Only create files in the primary filegroup for the database.
- Distribute the files across separate disks.
- The number of data files should be less than or equal to the number of core CPUs. Count dual core processors as two CPUs for this purpose. Count each processor that supports hyper-threading as a single CPU.
- Create data files of equal size.

Important:
Although you can use the backup and recovery tools that are built in to SharePoint Server 2010 to back up and recover multiple data files, if you overwrite in the same location, the tools cannot

restore multiple data files to a different location. For this reason, we strongly recommend that when you use multiple data files for a content database, you use SQL Server backup and recovery tools. For more information about how to back up and recover SharePoint Server 2010, see Plan for backup and recovery (SharePoint Server 2010).

For more information about how to create and manage filegroups, see Physical Database Files and Filegroups (http://go.microsoft.com/fwlink/?LinkId=117909).

Limit content database size to improve manageability

Plan for database sizing that will improve manageability, performance, and ease of upgrade for your environment.

To help ensure system performance, we strongly recommended that you limit the size of content databases to 200 GB.

A site collection should not exceed 100 GB unless it is the only site collection in the database. This limit exists so that you can use the SharePoint Server 2010 granular backup tools to move a site collection to another database if you need to.

Important:
Content database sizes up to 1 terabyte are supported only for large, single-site repositories and archives in which data remains reasonably static, such as reference document management systems and Records Center sites. Larger database sizes are supported for these scenarios because their I/O patterns and typical data structure formats have been designed for and tested at larger scales.

If your design requires a database larger than the recommended standard, follow this guidance:

- For databases that contain many large files that are stored as binary large objects (BLOBs), consider using remote BLOB storage (RBS). RBS is appropriate in the following circumstances:
  a. When you are running sites that contain large files that are infrequently accessed, such as knowledge repositories.
  b. When you have terabytes of data.
  c. For video or media files.

  For more information, see Plan for Remote BLOB Storage (RBS) (SharePoint Server 2010).
- Follow best practices for viewing data from large databases. For more information, see SharePoint Server 2010 capacity management: Software boundaries and limits.

For more information about large-scale document repositories, see "Estimate Performance and Capacity Requirements for Large Scale Document Repositories", available from Performance and capacity test results and recommendations (SharePoint Server 2010).

Proactively manage the growth of data and log files

We recommend that you proactively manage the growth of data and log files by considering the following recommendations:

- As much as possible, pre-grow all data and log files to their anticipated final size.
- We recommend that you enable autogrowth for safety reasons. Do not rely on the default autogrowth settings. Consider the following guidelines when configuring autogrowth:
  - When you plan content databases that exceed the recommended size (200 GB), set the database autogrowth value to a fixed number of megabytes instead of to a percentage. This will reduce the frequency with which SQL Server increases the size of a file. Increasing file size is a blocking operation that involves filling the new space with empty pages.
  - Set the autogrowth value for the Search service application Property Store database to 10 percent.
  - If the calculated size of the content database is not expected to reach the recommended maximum size of 200 GB within the next year, set it to the maximum size the database is predicted to reach within a year — with 20 percent additional margin for error — by using the **ALTER DATABASE MAXSIZE** property. Periodically review this setting to make sure it is still an appropriate value based on past growth rates.
- Maintain a level of at least 25 percent available space across disks to allow for growth and peak usage patterns. If you are managing growth by adding disks to a RAID array or allocating more storage, monitor disk size closely to avoid running out of space.

Validate and monitor storage and SQL Server performance

Test that your performance and backup solution on your hardware enables you to meet your service level agreements (SLAs). In particular, test the I/O subsystem of the computer that is running SQL Server to ensure that performance is satisfactory.

Test the backup solution that you are using to ensure that it can back up the system within the available maintenance window. If the backup solution cannot meet the SLAs your business requires, consider using an incremental backup solution such as System Center Data Protection Manager (DPM) 2010.

It is important to track the following resource components of a server that is running SQL Server: CPU, memory, cache/hit ratio, and I/O subsystem. When one or more of the components seems slow or overburdened, analyze the appropriate strategy based on the current and projected workload. For more information, see Troubleshooting Performance Problems in SQL Server 2008 (http://go.microsoft.com/fwlink/?LinkID=168448).

The following section lists the performance counters that we recommend that you use to monitor the performance of the SQL Server databases that are running in your SharePoint Server 2010 environment. Also listed are approximate healthy values for each counter.

For details about how to monitor performance and use performance counters, see Monitoring Performance (http://go.microsoft.com/fwlink/?LinkId=189032).

SQL Server counters to monitor

Monitor the following SQL Server counters to ensure the health of your servers:

- **General statistics** This object provides counters to monitor general server-wide activity, such as the number of current connections and the number of users connecting and disconnecting per second from computers running an instance of SQL Server. Consider monitoring the following counter:
  - **User connections** This counter shows the amount of user connections on your computer running SQL Server. If you see this number rise by 500 percent from your baseline, you may see a performance reduction.
- **Databases** This object provides counters to monitor bulk copy operations, backup and restore throughput, and transaction log activities. Monitor transactions and the transaction log to determine how much user activity is occurring in the database and how full the transaction log is becoming. The amount of user activity can determine the performance of the database and affect log size, locking, and replication. Monitoring low-level log activity to gauge user activity and resource usage can help you to identify performance bottlenecks. Consider monitoring the following counter:
  - **Transactions/sec** This counter shows the amount of transactions on a given database or on the entire server per second. This number is more for your baseline and to help you troubleshoot issues.
- **Locks** This object provides information about SQL Server locks on individual resource types. Consider monitoring the following counters:
  - **Average Wait Time (ms)** This counter shows the average amount of wait time for each lock request that resulted in a wait.
  - **Lock Wait Time (ms)** This counter shows the wait time for locks in the last second.
  - **Lock waits/sec** This counter shows the number of locks per second that could not be satisfied immediately and had to wait for resources.
  - **Number of deadlocks/sec** This counter shows the number of deadlocks on the computer running SQL Server per second. This should not rise above 0.
- **Latches** This object provides counters to monitor internal SQL Server resource locks called latches. Monitoring the latches to determine user activity and resource usage can help you to identify performance bottlenecks. Consider monitoring the following counters:
  - **Average Latch Wait Time (ms)** This counter shows the average latch wait time for latch requests that had to wait.
  - **Latch Waits/sec** This counter shows the number of latch requests that could not be granted immediately.
- **SQL Statistics** This object provides counters to monitor compilation and the type of requests sent to an instance of SQL Server. Monitoring the number of query compilations and recompilations and the number of batches received by an instance of SQL Server gives you an indication of how quickly SQL Server is processing user queries and how effectively the query optimizer is processing the queries. Consider monitoring the following counters:

    - **SQL Compilations/sec** This counter indicates the number of times the compile code path is entered per second.
    - **SQL Re-Compilations/sec** This counter indicates the number statement recompiles per second.
- **Buffer Manager** This object provides counters to monitor how SQL Server uses memory to store data pages, internal data structures, and the procedure cache, as well as counters to monitor the physical I/O as SQL Server reads and writes database pages. Consider monitoring the following counter:
    - **Buffer Cache Hit Ratio**
    - This counter shows the percentage of pages that were found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups over the last few thousand page accesses. Because reading from the cache is much less expensive than reading from disk, you want this ratio to be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory available to SQL Server.
- **Plan Cache** This object provides counters to monitor how SQL Server uses memory to store objects such as stored procedures, ad hoc and prepared Transact-SQL statements, and triggers. Consider monitoring the following counter:
    - **Cache Hit Ratio**
    - This counter indicates the ratio between cache hits and lookups for plans.

Physical server counters to monitor

Monitor the following counters to ensure the health of your computers running SQL Server:

- **Processor: % Processor Time: _Total** This counter shows the percentage of time that the processor is executing application or operating system processes other than Idle. On the computer that is running SQL Server, this counter should be kept between 50 percent and 75 percent. In case of constant overloading, investigate whether there is abnormal process activity or if the server needs additional CPUs.
- **System: Processor Queue Length** This counter shows the number of threads in the processor queue. Monitor this counter to ensure that it remains less than two times the number of core CPUs.
- **Memory: Available Mbytes** This counter shows the amount of physical memory, in megabytes, available to processes running on the computer. Monitor this counter to ensure that you maintain a level of at least 20 percent of the total available physical RAM.
- **Memory: Pages/sec** This counter shows the rate at which pages are read from or written to disk to resolve hard page faults. Monitor this counter to ensure that it remains under 100.

For more information and memory troubleshooting methods, see SQL Server 2005 Monitoring Memory Usage (http://go.microsoft.com/fwlink/?LinkID=105585).

Disk counters to monitor

Monitor the following counters to ensure the health of disks. Note that the following values represent values measured over time — not values that occur during a sudden spike and not values that are based on a single measurement.

- **Physical Disk: % Disk Time: DataDrive** This counter shows the percentage of elapsed time that the selected disk drive is busy servicing read or write requests–it is a general indicator of how busy the disk is. If the **PhysicalDisk: % Disk Time** counter is high (more than 90 percent), check the **PhysicalDisk: Current Disk Queue Length** counter to see how many system requests are waiting for disk access. The number of waiting I/O requests should be sustained at no more than 1.5 to 2 times the number of spindles that make up the physical disk.
- **Logical Disk: Disk Transfers/sec** This counter shows the rate at which read and write operations are performed on the disk. Use this counter to monitor growth trends and forecast appropriately.
- **Logical Disk: Disk Read Bytes/sec** and **Logical Disk: Disk Write Bytes/sec** These counters show the rate at which bytes are transferred from the disk during read or write operations.
- **Logical Disk: Avg. Disk Bytes/Read** This counter shows the average number of bytes transferred from the disk during read operations. This value can reflect disk latency — larger read operations can result in slightly increased latency.
- **Logical Disk: Avg. Disk Bytes/Write** This counter shows the average number of bytes transferred to the disk during write operations. This value can reflect disk latency — larger write operations can result in slightly increased latency.
- **Logical Disk: Current Disk Queue Length** This counter shows the number of requests outstanding on the disk at the time that the performance data is collected. For this counter, lower values are better. Values greater than 2 per disk may indicate a bottleneck and should be investigated. This means that a value of up to 8 may be acceptable for a logical unit (LUN) made up of 4 disks. Bottlenecks can create a backlog that can spread beyond the current server that is accessing the disk and result in long wait times for users. Possible solutions to a bottleneck are to add more disks to the RAID array, replace existing disks with faster disks, or move some data to other disks.
- **Logical Disk: Avg. Disk Queue Length** This counter shows the average number of both read and write requests that were queued for the selected disk during the sample interval. The rule is that there should be two or fewer outstanding read and write requests per spindle, but this can be difficult to measure because of storage virtualization and differences in RAID levels between configurations. Look for larger than average disk queue lengths in combination with larger than average disk latencies. This combination can indicate that the storage array cache is being overused or that spindle sharing with other applications is affecting performance.
- **Logical Disk: Avg. Disk sec/Read** and **Logical Disk: Avg. Disk sec/Write** These counters show the average time, in seconds, of a read or write operation to the disk. Monitor these counters to ensure that they remain below 85 percent of the disk capacity. Disk access time increases exponentially if read or write operations are more than 85 percent of disk capacity. To determine the specific capacity for your hardware, refer to the vendor documentation or use the SQLIO Disk

Subsystem Benchmark Tool to calculate it. For more information, see SQLIO Disk Subsystem Benchmark Tool (http://go.microsoft.com/fwlink/?LinkID=105586).

- **Logical Disk: Avg. Disk sec/Read** This counter shows the average time, in seconds, of a read operation from the disk. On a well-tuned system, ideal values are from 1 through 5 ms for logs (ideally 1 ms on a cached array), and from 4 through 20 ms for data (ideally less than 10 ms). Higher latencies can occur during peak times, but if high values occur regularly, you should investigate the cause.
- **Logical Disk: Avg. Disk sec/Write** This counter shows the average time, in seconds, of a write operation to the disk. On a well-tuned system, ideal values are from 1 through 5 ms for logs (ideally 1 ms on a cached array), and from 4 through 20 ms for data (ideally less than 10 ms). Higher latencies can occur during peak times, but if high values occur regularly, you should investigate the cause.

When you are using RAID configurations with the **Avg. Disk sec/Read** or **Avg. Disk sec/Write** counters, use the formulas listed in the following table to determine the rate of input and output on the disk.

RAID level	Formula
RAID 0	I/Os per disk = (reads + writes) / number of disks
RAID 1	I/Os per disk = [reads + (2 × writes)] / 2
RAID 5	I/Os per disk = [reads + (4 × writes)] / number of disks
RAID 10	I/Os per disk = [reads + (2 × writes)] / number of disks

For example, if you have a RAID 1 system that has two physical disks, and your counters are at the values that are shown in the following table:

Counter	Value
Avg. Disk sec/Read	80
Logical Disk: Avg. Disk sec/Write	70
Avg. Disk Queue Length	5

The I/O value per disk can be calculated as follows: (80 + (2 × 70))/2 = 110

The disk queue length can be calculated as follows: 5/2 = 2.5

In this situation, you have a borderline I/O bottleneck.

Other monitoring tools

You can also monitor disk latency and analyze trends by using the sys.dm_io_virtual_file_stats dynamic management view in SQL Server 2008. For more information, see sys.dm_io_virtual_file_stats (Transact-SQL) (http://go.microsoft.com/fwlink/?LinkID=105587).