Technical White Paper

Microsoft® IT Showcase

SharePoint Performance Optimization

How Microsoft IT Increases Availability and Decreases Rendering Time of SharePoint Sites

Technical White Paper

Published: September 2008

Microsoft®

CONTENTS

Executive Summary 3

SharePoint Environment Landscape 4

Data-Center Topology 4

Service Team Structure 5

Investigation Tools 5

Initial Performance Discoveries 7

Investigation Phases 11

Phase 1: Initial Analysis and Infrastructure Simplification 11

Phase 2: Targeted Reconfiguration 11

Phase 3: Performance Examination 11

Optimization Opportunities and Activities 12

Front End 12

Back End 15

Results 18

Future Activities 20

Best Practices 22

Conclusion 25

For More Information 26

Appendix: URL Ping Tool 27

Situation

As part of its goal to provide users with the latest available and proven technologies, Microsoft IT deployed Microsoft Office SharePoint Server 2007 in November 2006. Shortly after deploying Office SharePoint Server 2007 in the corporate production environment, Microsoft IT addressed long page-rendering times, and spikes in disk I/O and CPU utilization.

Solution

By systematically working to discover root causes of performance issues, Microsoft IT identified areas of opportunity for performance optimization, including the back-end SQL Server storage subsystem and front-end IIS servers. Microsoft IT implemented changes, monitored performance, and made configuration adjustments to improve performance.

Benefits

- More stable performance with smaller load spikes
- Decreased site rendering times
- Reduced support tickets
- Better user experience

Products & Technologies

- Microsoft Office SharePoint Server 2007 with SP1
- SQL Server 2005 with SP2
- Windows Server 2003
- IIS 6.0

EXECUTIVE SUMMARY

Microsoft Information Technology (Microsoft IT) manages a Microsoft® SharePoint® infrastructure that supports approximately 154,000 active user profiles. These profiles access more than 150,000 site collections that use more than 19 terabytes of disk space—and this increases daily. To operate the environment, Microsoft IT relies on teams, including a dedicated SharePoint Operations team and the global Helpdesk. The SharePoint Operations team has implemented the latest SharePoint technologies, culminating with Microsoft Office SharePoint Server 2007 in November 2006. The current implementation provides benefits to Microsoft users, such as self-service provisioning, content search, and integration with the 2007 Microsoft Office system. Following extensive design and deployment planning, Microsoft IT noticed performance challenges and then indentified a series of steps to optimize performance.

By using structured monitoring and operations processes according to established frameworks such as the Microsoft Operations Framework (MOF), Microsoft IT validates performance in a production and an early adopter environment while participating in product improvement. Following structured processes was especially useful in improving performance in the SharePoint environment. The SharePoint Operations team investigated root causes by considering the possible components, such as Internet Information Services (IIS) front-end servers and the Microsoft SQL Server® back-end configuration. Using common troubleshooting and diagnostic tools such as Performance Monitor and diagnostic approaches such as cause/effect diagrams, Microsoft IT identified the possible universe of root causes and implemented changes to improve performance.

The SharePoint Operations team performed key configuration changes that resolved performance challenges on both front-end and back-end subsystems. For the front-end servers, migrating to 64-bit hardware and increasing random access memory (RAM) were the most vital changes, whereas ensuring disk throughput was most vital for the back-end storage subsystem. Additionally, the SharePoint Operations team optimized the underlying network configuration by ensuring that front-end and back-end subsystems used dedicated Gigabit Ethernet network interface cards (NICs). During the performance investigations, Microsoft IT developed many best practices for SharePoint performance optimization.

This technical white paper is intended for technical decision makers who are operating an enterprise SharePoint environment. Among other topics, this paper covers the tools and processes that Microsoft IT used when investigating performance, and the resulting findings and changes made. This paper is not intended as prescriptive guidance but as an example of how to optimize a SharePoint environment.

Note: *For security reasons, the sample names of domains, internal resources, organizations, and internally developed security file names used in this paper do not represent real resource names used within Microsoft and are for illustration purposes only.*

SHAREPOINT ENVIRONMENT LANDSCAPE

At first glance, 19 terabytes of SharePoint content may not seem like much for an enterprise-level SharePoint environment. For Microsoft, however, it represents more than 30 million files stored across more than 180,000 sites. In addition to the established infrastructure required to support search functionality and document management, the environment hosts an intranet portal, many department-level portals, thousands of self-provisioned sites, and personal employee sites. The SharePoint infrastructure and the services that it provides have increasingly become mission critical for everyday business.

Data-Center Topology

The majority of data that the SharePoint infrastructure hosts comes from self-provisioned team sites and personal employee sites. Combined, they represent about 70 percent of the total data volume and 85 percent of the number of site collections. In terms of the user distribution, 60 percent of users access SharePoint sites from North America, 18 percent from Europe, 16 percent from Asia, and the remaining 6 percent from locations where Microsoft has a smaller presence, such as Africa and the Middle East. Microsoft IT distributes the traffic load across three data centers, as shown in Figure 1.

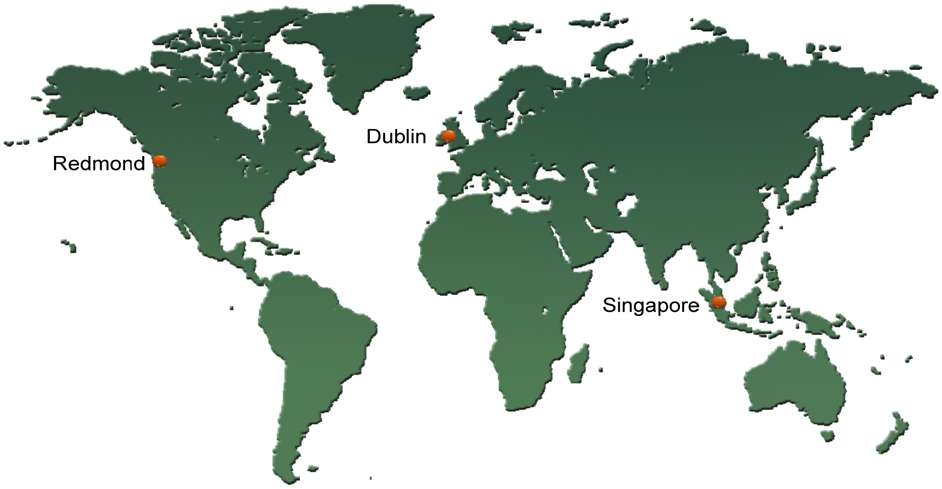


Figure 1. Data-center distribution

The Redmond location provides services primarily for users in North America and South America. It is a model deployment, with the Dublin and Singapore deployments mirroring the Redmond design on a smaller scale. Each data center implements a design where a parent farm houses Shared Services Providers (SSPs) for services such as the Business Data Catalog, Profiles, Search, Analytics, and Audiences; and where child farms consume these services. For example, the Search SSP in Redmond crawls content from all three regions and imports line-of-business data into user profiles by using Business Data Catalog functionality. The system replicates Business Data Catalog data to the regions as necessary. The Dublin and Singapore child farms crawl only local content.

Service Team Structure

Microsoft IT relies on three support tiers to handle operations for its SharePoint infrastructure, with each tier responsible for handling increasingly specific SharePoint issues. The Helpdesk is the initial contact point (Tier 1) for users and responds to tickets by using the resolution details in Microsoft IT's custom knowledge base. SharePoint specialists and support generalists contribute to this knowledge base, which specifies resolution actions to common issues. When the Helpdesk cannot resolve an issue, it escalates the ticket to Tier 2 and Tier 3 teams. Figure 2 shows the structure of the support teams.

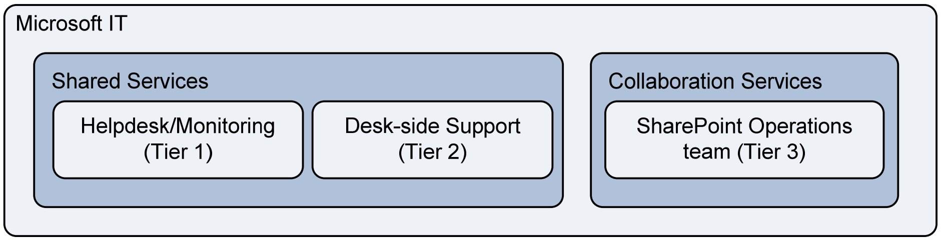

Microsoft IT
Shared Services
Helpdesk/Monitoring (Tier 1)
Desk-side Support (Tier 2)
Collaboration Services
SharePoint Operations team (Tier 3)

Figure 2. Teams involved in SharePoint optimization and support

Although all support tiers participate in optimizing the SharePoint environment, each tier plays a specific role that corresponds to the support tasks that the personnel at each tier perform. Tier 1 receives support requests from tickets and performs front-line monitoring of the infrastructure. This tier is responsible for maintenance of the server infrastructure, incident management, end-user technical support, Helpdesk training, monitoring, upgrades, patch management, hotfix management, and backup and restore services. Tier 2 responds to escalated tickets by performing more advanced diagnostics and sometimes performing in-person support to resolve issues related to end-user connectivity and productivity. Together, Tier 1 and Tier 2 resolve more than 95 percent of tickets.

Tier 3 operates the SharePoint infrastructure, including discovering the root causes of incidents and optimizing performance. This tier creates the necessary SharePoint-specific knowledge base and resolution instructions, and it provides training on general resolution and response processes.

The SharePoint Operations team handles incidents that require detailed investigations that go beyond basic monitoring and analysis of performance counters and event log entries to include more low-level analysis of dump files, SQL Server query profiling, and IIS memory usage.

Investigation Tools

The SharePoint Operations team uses the following investigation tools to gather statistics and data and to troubleshoot performance issues:

- **Event Viewer** This tool is especially useful for understanding the underlying behavior by evaluating application errors and warnings, or investigating system events that occur before, during, and after a performance incident.
- **Dump file analysis** Analyzing dump files is an advanced troubleshooting and analysis approach that provides low-level information about critical system errors and memory dumps. It enables the SharePoint Operations team to examine the data in memory and analyze the possible causes of such issues as memory leaks and invalid pointers.

- **System Monitor** The SharePoint Operations team uses tools such as Event Viewer and dump file analysis to investigate specific incidents and performance issues. The team uses System Monitor in the Windows Server® 2003 operating system (called Performance Monitor in Windows Server 2008) for establishing a performance baseline, tracking trends, and compiling data on resulting performance after making changes.
- **SQL Server Profiler** This tool is a graphical user interface to SQL Trace for monitoring an instance of SQL Server Database Engine or SQL Server Analysis Services. Microsoft IT and other teams use this tool to evaluate SQL Server performance aspects such as query times, stored procedure run times, and deadlocks. This tool is especially useful for analyzing the underlying calls to SQL Server databases that are housed on the storage area network (SAN).
- **Custom tool for client-based URL ping** The SharePoint Operations team created a custom tool that recorded the time to first byte for URLs hosted on SharePoint servers. This is one of the most useful tools because it enables the comparison of statistics before and after implementing configuration changes to the environment.

Note *The appendix in this paper includes the script for the custom URL ping tool as a reference. It is provided as is.*

- **Log Parser** The SharePoint Operations team uses logging extensively when determining root causes of issues, including SharePoint trace logs and IIS and Unified Logging Service (ULS) application and service logs. Microsoft IT uses Log Parser as one of the tools to monitor traffic, determine traffic sources distribution, and establish performance baselines. This free tool parses IIS logs, event logs, and many other kinds of structured data by using syntax similar to Structured Query Language (SQL). For more information about Log Parser, refer to the Script Center resource at http://www.microsoft.com/technet/scriptcenter/tools/logparser/default.mspx.
- **Fiddler** This tool is helpful for measuring caching, page sizes, authentication, and general performance issues. For more information, visit the Fiddler Web site at http://www.fiddler2.com/fiddler2/.

Initial Performance Discoveries

Microsoft IT follows established operations frameworks and processes, such as the Information Technology Infrastructure Library (ITIL) and MOF, for managing the SharePoint infrastructure. These frameworks provide guidance and methodologies for using people, processes, and technology to discover performance issues, determine underlying causes, and introduce changes that remedy the causes and resolve performance issues. Root cause analysis is the discipline that specifically deals with analyzing performance data and determining root causes, as shown in Figure 3.

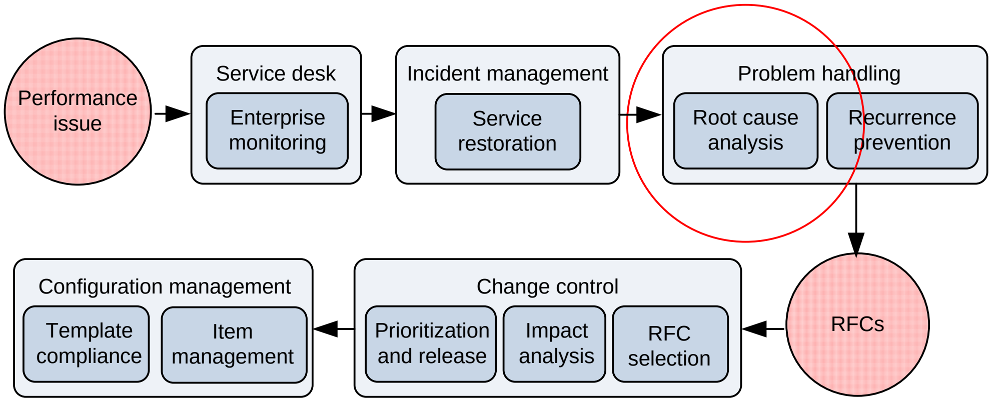


Figure 3. Root cause analysis in operations process workflow

The SharePoint Operations team uses root cause analysis within the larger scope of performance troubleshooting and optimization processes. These processes guide the SharePoint Operations team in narrowing down the possible root causes based on the symptoms and systematically implementing changes to improve performance. The SharePoint Operations team performs the following processes as part of optimization:

1. **Understand the issue and gather data** There are several ways that the SharePoint Operations team discovers that a performance issue exists, such as excessive trouble tickets from users, routine analysis of performance data and logs, and trend analysis. Regardless of the discovery source, the SharePoint Operations team seeks to obtain all possible information before taking further action. This information can come from internal analysis, such as operations staff who examine logs and performance data, or from other support tiers.

2. **Reproduce the issue if possible** By reproducing issues, the SharePoint Operations team can retrace steps and document possible causes. It is a standard practice in all troubleshooting and performance optimization scenarios.

3. **List possible root causes** The SharePoint Operations team uses common diagnostic approaches to list possible root causes before eliminating some and investigating others. These approaches include using differential diagnosis, asking five whys (asking, “why has the issue occurred” five times, with each question discovering a more detailed aspect of the issue), using cause/effect trees, and using fishbone diagrams.

4. **Implement change through the Request for Change (RFC) process** Microsoft IT has a structured RFC process that operations teams use to introduce changes to the organization. The order of introducing changes depends on factors such as time to

completion, complexity, and user impact. This SharePoint optimization effort introduced additional considerations for what changes to implement and their priority because of Microsoft IT's emphasis on product validation and suggesting improvements to the product group. For example, the SharePoint Operations team implemented changes one at a time to monitor the outcome, despite knowing that multiple root causes existed for performance issues. This enabled the SharePoint Operations team to document best practices for sharing with customers, as detailed in the "Best Practices" section later in this white paper.

5. **Monitor outcome, verify, and document** After making changes, the SharePoint Operations team verifies the resulting performance and compares the result with expectations. The team documents these findings and the specific performance issue becomes resolved or goes back for further analysis.

As part of implementing and operating any service or system, Microsoft IT gathers performance data and statistics to establish a performance baseline, forecast trends, and to catch outliers that may indicate performance issues. These statistics include information about back-end and front-end subsystem hardware, such as CPU load, disk input/output (I/O), and memory usage counters, in addition to user load data for an average number of users, and usage trends. For the SharePoint environment specifically, Microsoft IT monitored the infrastructure to develop a baseline that represents the state of a healthy system, as shown in Table 1.

Table 1. Performance Baseline Measurement After Initial Deployment

Category	Details
Front-end subsystem	
Memory	This is a key performance indicator for front-end servers. The baseline for memory usage is 50–60 percent of physical memory, as measured by the Memory/Available Bytes counter. For best performance, Microsoft IT prefers to minimize the number of virtual servers configured on a front-end server. This reduces the size of the IIS metabase in addition to the amount of server CPU and memory utilization.
CPU usage	The average CPU utilization and spikes in CPU usage provide early indication of performance issues. The baseline for this category is 30–50 percent average usage.
Disk I/O	The disk queue length is a relevant baseline measurement for Microsoft because it helps to track periods of normal and high activity.
Concurrent connections	Microsoft IT targets 150–250 average concurrent connections per server.
Back-end subsystem	
Memory	Memory is also important for back-end servers with similar considerations. A healthy utilization is 30–60 percent of physical memory.
CPU usage	The average CPU utilization is not as important for back-end servers as it is for front-end servers. Nevertheless, it affects performance. Microsoft IT established the CPU usage baseline for back-end servers at 30–50 percent average utilization.

Disk I/O	Disk queue length, as measured by the PhysicalDisk(_Total)/Current Disk Queue Length counter, in addition to disk I/O, is extremely important for back-end servers.
Concurrent connections/SQL Server blocking	For back-end servers, the number of concurrent connections is relevant, especially when they correspond to instances of SQL Server blocking. Correspondingly, Microsoft IT tracked the baseline of SQL Server blocking by using the SQLServer:Locks(_Total)\Number of Deadlocks/sec counter. An acceptable number of deadlocks is below one per second.
Percentage of database fragmentation	The baseline fragmentation is below 8 percent on back-end servers.
Other	
Site traffic	The site traffic considered healthy varies with the capacity design for each server and server farm. Microsoft IT tracks hits to SharePoint sites and reports on them. As the baseline, the environment had an average of 6 million page hits per day.

Before the SharePoint Operations team began the SharePoint optimization process, the team gathered baseline performance and reviewed statistics from the Shared Services team, which monitors the environment and reports on Helpdesk ticket categories. The SharePoint Operations team discovered opportunities to enhance optimization after reviewing the Helpdesk ticket trends.

Note *It is a Microsoft IT best practice to develop performance baseline data for all implementations. Comparing future statistics and trends with baseline statistics helps identify trends.*

The SharePoint Operations team acted upon the discovery by gathering more data to verify performance and compare it to the baseline established after the initial rollout in November 2006. To mimic user experience, the team created a custom script that pinged the URL of a server and recorded the amount of time required to receive the first byte of response data. This is more useful than traditional counters because it goes beyond basic availability to imitate a user's behavior. A server can be available and render pages, yet do it so slowly that it provides a poor user experience.

By using the custom URL ping tool, the SharePoint Operations team recorded performance data of the time to first byte for all front-end servers. This data enabled the team to discover random spikes occurring across several farms, and especially self-provisioned team sites and personal employee sites, as shown in Figure 4. At times, the average site render time was greater than six seconds.

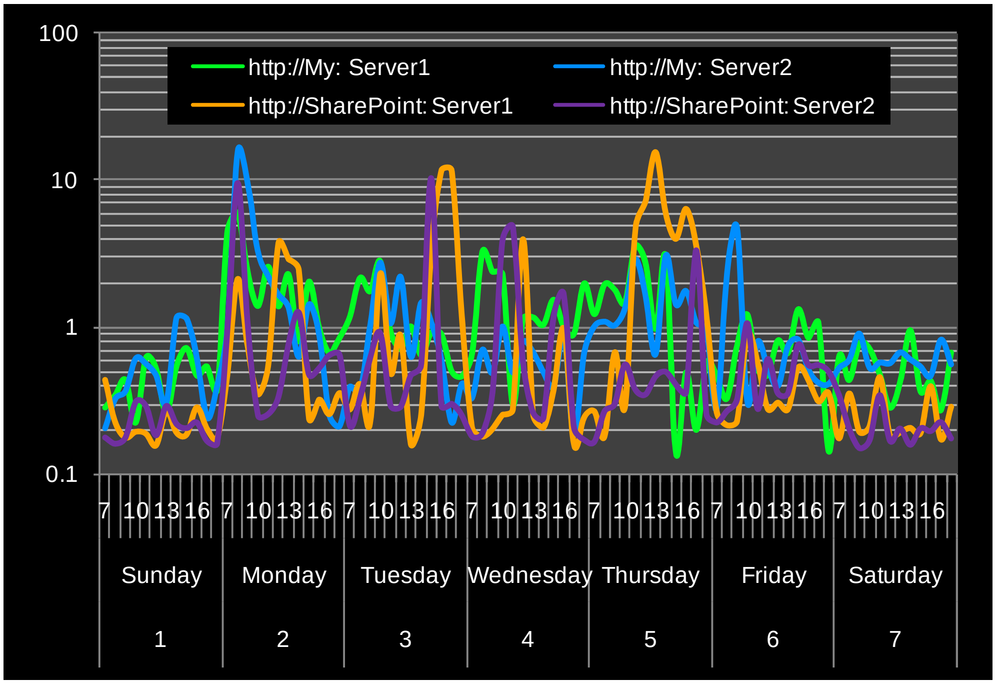


Figure 4. Spikes in site render times

After discovering the random spikes in render times, the SharePoint Operations team began investigating possible root causes. The team investigated both upstream and downstream possible causes on the front-end servers and back-end storage subsystem.

INVESTIGATION PHASES

Microsoft IT must not only operate the environment, but also verify performance to customers. Microsoft IT implemented changes and recorded the outcome of each change in phases. Microsoft IT systematically identified opportunities to optimize SharePoint performance until all possible elements, such as team processes, network, hardware, and software, were resolved. After the rollout in November 2006, three major phases occurred to optimize SharePoint performance and resolve recurring spikes:

- **Phase 1: Initial Analysis and Infrastructure Simplification** During Phase 1, the teams worked together to gather data about front-end and back-end servers, as well as underlying network infrastructure. To record more data and simplify the environment, the SharePoint Operations team separated the single farm that housed self-provisioned team sites and personal employee sites into separate farms that used a common SQL Server back end.
- **Phase 2: Targeted Reconfiguration** During Phase 2, the teams targeted specific possible root causes, determined optimal configurations, and systematically worked to resolve issues. Among other tasks, the SharePoint Operations team deployed 64-bit hardware for the front-end servers to realize the better memory-handling capabilities. The SharePoint Operations team also mitigated sites with many items in lists, as explained later.
- **Phase 3: Performance Examination** At the beginning of Phase 3, all of the teams involved gathered for a two-day examination of the performance indicators. They examined the data gathered thus far, the known issues, and the possible root causes identified or investigated. At the end, they arrived at a cause/effect analysis summary, as shown in Figure 5. Because some spikes in site render times persisted, the SharePoint Operations team eliminated the root causes that were addressed during the first two phases and focused on other root causes, such as network configuration.

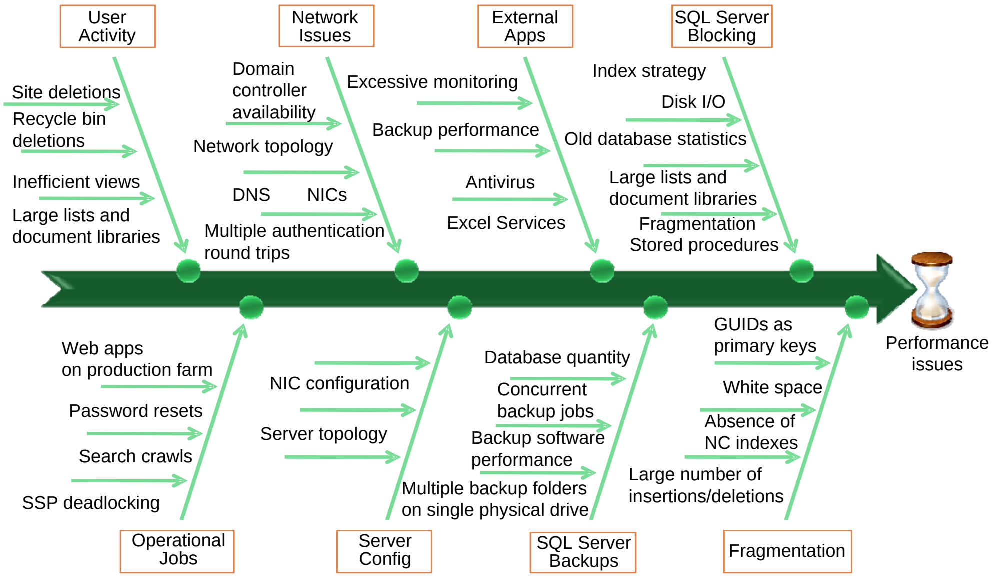


Figure 5. Cause/effect analysis summary

OPTIMIZATION OPPORTUNITIES AND ACTIVITIES

After gathering performance data, analyzing the existing configuration, and examining findings for front-end and back-end subsystems and processes, the SharePoint Operations team was ready to list all possible causes of performance issues, systematically implement changes to address the causes, and monitor resulting performance. The SharePoint Operations team prioritized the proposed changes according to criteria such as ease of implementation, time to completion, and impact to users, and implemented the more straightforward changes first.

Front End

For the front-end subsystem, the SharePoint Operations team pursued the improvement opportunities described in the following sections.

Separate Self-Provisioned Team Sites and Personal Employee Sites

The self-provisioned team sites and personal employee sites resided in the same farm and used the same front-end servers. This led to a scenario in which tracing the relationship between performance data and the underlying configuration was difficult. These sites exhibited the largest spikes in render times. The SharePoint Operations team separated the farms by using the same SQL Server cluster and SAN, as shown in Figure 6. The team also designated distinct index targets for each site.

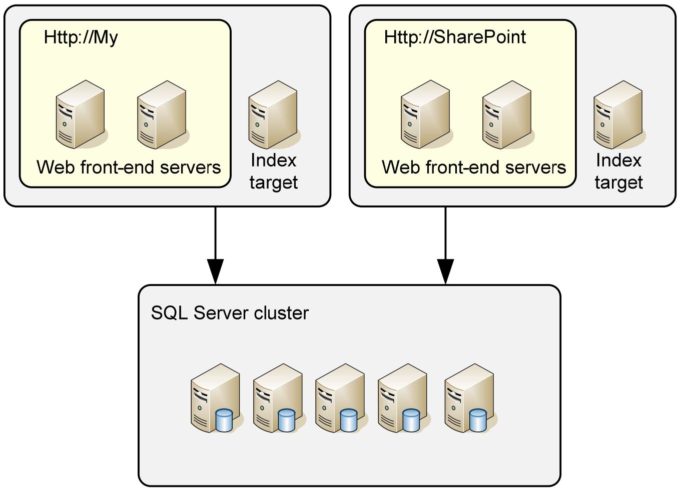


Figure 6. Farm configurations after separation

By giving each site a dedicated farm, the SharePoint Operations team created dedicated configuration databases for each site, which more evenly distributed the load. Separating the sites into individual farms also reduced the load against the **all_** tables in the content databases.

Note: *For more information about sizing front-end and back-end servers, and for general performance best practices, refer to the "Plan for Performance and Capacity" article at http://technet.microsoft.com/en-us/library/cc262971.aspx.*

Load Balance Excel Services Across Front-End Servers

Excel® Services in Office SharePoint Server 2007 sometimes consumes large amounts of server resources when processing operations. This makes processor cycles and memory unavailable to other SharePoint processes. While separating the self-provisioned team sites and personal employee sites, the SharePoint Operations team took the opportunity to load balance Excel Services across multiple front-end servers.

Migrate IIS Servers to 64-Bit Hardware

To address random IIS failures and memory issues, the SharePoint Operations team migrated servers to 64-bit hardware in the farms housed in the Redmond data center. This process occurred in two stages. First, the SharePoint Operations team migrated the farm for self-provisioned team sites. Then, it migrated the farm for personal employee sites.

The team used a standard design for all servers. Each server has a planned capacity of 400–450 simultaneous connections users, making it straightforward to increase capacity in the future by monitoring load and adding servers when necessary. The SharePoint Operations team adds servers when CPU utilization is higher than 80 percent consistently on existing servers. Table 2 lists the specifications for the original servers and the replacement servers.

Table 2. Hardware Specifications for Front-End Servers

	Original 32-bit hardware	**New 64-bit hardware (DL380-G5)**
Processor	Single dual core	Dual Intel Xeon 2.33-gigahertz (GHz) quad core
Memory	8 gigabytes (GB)	16 GB
Storage	Separate partitions for operating system, program files, utility, and index; 435 GB total capacity	Separate partitions for operating system, program files, utility, and index; 550.4 GB total capacity
Other	Redundant power supplies, redundant fans, and DVD drive	Redundant power supplies, redundant fans, and DVD drive

Note: *The SharePoint Operations team did not migrate other farms to 64-bit hardware because the existing 32-bit solution was adequate for relative user traffic and load. The farms in the Dublin and Singapore data centers kept the existing 32-bit hardware.*

In terms of Office SharePoint Server, 64-bit hardware provides advantages for caching and worker processes. Worker processes for application pools are heavy consumers of RAM and compete for available RAM with the underlying operating system and other SharePoint services. A design with 4 GB of RAM on 32-bit hardware running just two to three worker processes, in addition to mssearch.exe and Excel Services, will compete for memory with moderate load. A design with 8 GB and more on 64-bit hardware is more scalable because it can address more RAM.

When application pool recycling occurs, it uses RAM and may interfere with caching mechanisms in Office SharePoint Server and IIS. Using 64-bit hardware enables better memory utilization, which enables servers to handle more load from application pools. This means that a single server can handle more load before performance issues occur, which is especially useful for sites that have high peak loads or are especially busy.

Reinstall and Reconfigure NIC Configuration

During the course of reviewing design and implementation documentation for front-end servers, the SharePoint Operations team discovered that the existing NIC configuration did not conform to design specifications. As designed, front-end servers have two NICs of 1,000 megabits per second (Mbps), one for communication with the SQL Server back end and one for communication with the client. However, the SharePoint Operations team discovered that only one 100-Mbps NIC card existed on the server for both SQL Server and client traffic. This caused traffic congestion on front-end servers.

Consequently, the team reinstalled and reconfigured the NICs on IIS servers to conform to specifications, as shown in Figure 7. Under more typical load, such as the load in the Dublin and Singapore data centers, using 100-Mbps NICs should be adequate, but the heavy usage of self-provisioned team sites and personal employee sites made 1,000-Mbps NICs a necessity.

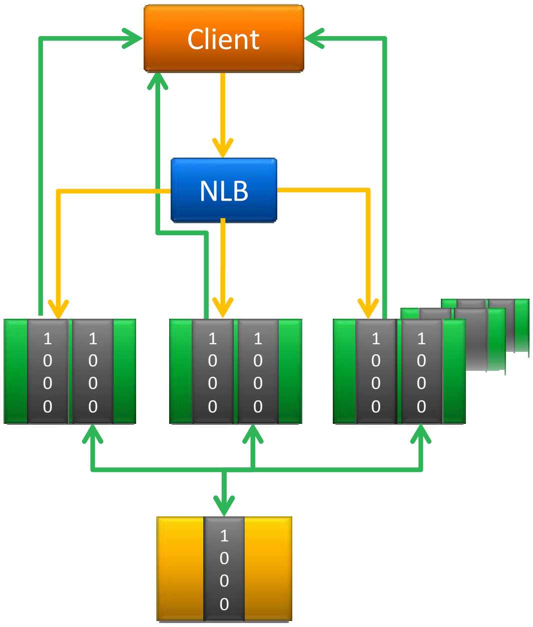


Figure 7. Revised NIC configuration

Schedule Daily Application Pool Recycling

A performance interdependency exists between caching mechanisms, memory, and application pool recycling and operations. One of the keys to the interdependency is that although caching improves performance, Office SharePoint Server disables it if the front-end server detects low-memory conditions. These low-memory conditions occur when application pools use memory during typical tasks, including recycling. Using more than one application pool worsens the issue and interferes with page output caching.

The SharePoint Operations team addressed these interdependencies by using 64-bit hardware, and by scheduling application pool recycling for once a day during non-peak hours through a garbage collection/application recycling tool. On 32-bit hardware, the SharePoint Operations team addressed the interdependencies by setting the memory limit of the application pool to 1.4 GB and scheduling application pool recycling for once a day during non-peak hours.

Enable IIS Compression

As a general best practice, the SharePoint Operations team verified that IIS compression for static content was enabled on front-end servers. Enabling static compression is especially helpful for serving content to users over slower links.

Remove Non-Production Web Applications

As general maintenance, the team reviewed existing SharePoint applications and removed non-production Web applications from the environment.

Back End

Whereas 64-bit hardware and proper network configuration proved to be the most effective at resolving front-end performance issues, ensuring disk I/O throughput proved to be the most effective at resolving back-end performance issues. For the SharePoint Operations team, ensuring available I/O is not as straightforward as checking a counter against an established baseline. It involves multiple aspects, such as comparing performance over time for specific data centers, SQL Server clusters, and SAN logical unit numbers (LUNs); auditing the configuration to enforce best practices; and mitigating SharePoint-specific issues such as large lists and configuration of index servers.

For the back-end subsystem, the SharePoint Operations team pursued the improvement opportunities described in the following sections.

Standardize File Locations

Part of the simplification process in Phase 1 included standardizing and reducing the complexity of file locations. The team isolated backup files, data and log files, and temporary database files to a non-shared drive in an effort to ensure I/O throughput.

Improve Database Distribution

The initial SAN configuration after deployment of Office SharePoint Server 2007 placed multiple databases on the same LUN in the SAN. Over time, these databases became fragmented and experienced disk I/O throughput issues. At the time of implementation, it was difficult to foresee which databases would experience the most fragmentation, and which would experience the heaviest loads.

The SharePoint Operations team distributed the load more evenly by analyzing performance data and then moving the most used databases to dedicated LUNs. Although this approach helped prevent performance issues, it did not address a root cause. Rearranging the database association with different LUNs freed disk I/O to databases due to the more even load, but it did not increase disk I/O or resolve network configuration issues.

Schedule Weekly Indexing, Defragmentation, and Crawling

The SharePoint Operations team configured defragmentation and database re-indexing to occur weekly at the object level. The team uses the change log feature of Office SharePoint Server 2007 to crawl incrementally based on date/time stamps for each site collection instead of crawling all content.

Even with incremental crawling, the crawl process places a CPU and memory-intensive load on front-end servers. The SharePoint Operations team mitigates this by using a dedicated front-end crawl target server that is also used for indexing and that is not in the load-balanced cluster. When using Network Load Balancing (NLB) and configuring server farms to use all front-end servers for crawling, the index server sends requests to each Web server in the farm. Without a dedicated front-end index, significant performance degradation would occur because of CPU and memory usage spikes from crawl operations.

For more information about recommendations for incremental crawling, refer to the article "Determine When to Crawl Your Content" at http://technet.microsoft.com/en-us/library/cc263242(TechNet.10).aspx.

Set Fill Factor Value

When an index is created or rebuilt, the fill factor value determines the percentage of space on each leaf level page to be filled with data, therefore reserving a percentage of free space for future growth. Based on past performance and index expansion rates, the SharePoint Operations team set the database fill factor to 70 percent on all content databases.

Set Growth Limit

The SharePoint Operations team set a 100-GB growth limit. The reasons for this choice are related to administration ease and SQL Server blocking.

In terms of administration, backing up and restoring databases of a moderate size is faster and less error prone than it is on larger databases. The SharePoint Operations team must sometimes restore lists on a specific database, and dealing with smaller files is easier when the team is copying over the network, mounting the databases, and reattaching them to farms.

In terms of performance and impact of user behavior, smaller database sizes also help. When users delete large lists that are stored in SharePoint databases, the SQL Server–based server must process those requests and complete the deletion. This can create performance issues through SQL Server blocking. Users sometimes experience long render times for other sites that use the same database. From a practical standpoint, smaller database sizes tend to house fewer sites, and if a user deletes a large list, fewer sites are affected if fewer sites are housed on the database.

Enable BLOB Caching and Output Caching

Office SharePoint Server includes three types of caching:

- **Output caching** at the individual page level for heavily accessed Web sites that do not need to frequently present new content
- **Object caching** of individual Web Part controls, field controls, and content levels, including cross-list query caching and navigation caching
- **Disk-based caching** for binary large objects (BLOBs) at the individual BLOB level for image, sound, and code files that are stored as BLOBs

Both BLOB caching and output caching help save traffic between front-end and back-end servers. Correspondingly, the SharePoint Operations team enabled both types. BLOB caching enables clients to cache static content, relieving load off the back-end servers, whereas output caching stores compiled Microsoft ASP.NET (.aspx) pages in the memory of front-end servers, which decreases CPU utilization.

Optimize Backup Routines

Backup operations heavily use CPU cycles and place a load on disk I/O. In the days before Office SharePoint Server 2007, the total backup volume was moderate, at five terabytes to 10 terabytes. However, due to the high adoption rate at Microsoft, the backup footprint is increasing by approximately five terabytes per year. In the past, the SharePoint Operations team could perform full backups every night during non-peak hours. However, with the increasing data volume, full backups started to exceed acceptable backup windows and placed too much load on servers.

Because of the business-critical nature of backups and the increased backup size, the SharePoint Operations team switched to using differential backups on weekdays and full backups on weekends. The team also eliminated some backed-up files, such as the system state. In the future, the team plans to deploy Microsoft System Center Data Protection Manager, which enables an organization to perform faster restores and backups, have multiple checkpoints, and save time by restoring list items directly from lists.

Identify Sites with Large Lists and Mitigate the Issue

Office SharePoint Server 2007 stores most end-user data—such as document libraries, calendars, and contacts—in lists. These lists can quickly become very large, which can affect performance in some scenarios, including backup and restore operations, and typical list tasks such as adding or deleting items. In addition, performing operations on large lists that are not indexed can cause delays of 5 seconds to 15 seconds in response times.

Knowing that large sites or long forms cause database locking, the SharePoint Operations team identified large sites and lists. It then asked the owners of the sites and lists to archive items or pursue other mitigation strategies, such as using subfolders for list items and using indexed columns to reduce performance impact.

For more information about strategies to mitigate large lists, refer to the white paper *Working with Large Lists in Office SharePoint Server 2007* at http://technet.microsoft.com/en-us/library/cc262813(TechNet.10).aspx.

RESULTS

After implementing the changes in Phase 1 (separating self-provisioned team sites and personal employee sites), the SharePoint Operations team realized many benefits. The Helpdesk call volume and tickets related to SharePoint performance decreased, and time to first byte decreased to one to two seconds on average. Fewer IIS failures occurred, and the infrastructure exhibited improved overall stability. However, as shown in Figure 8, spikes persisted.

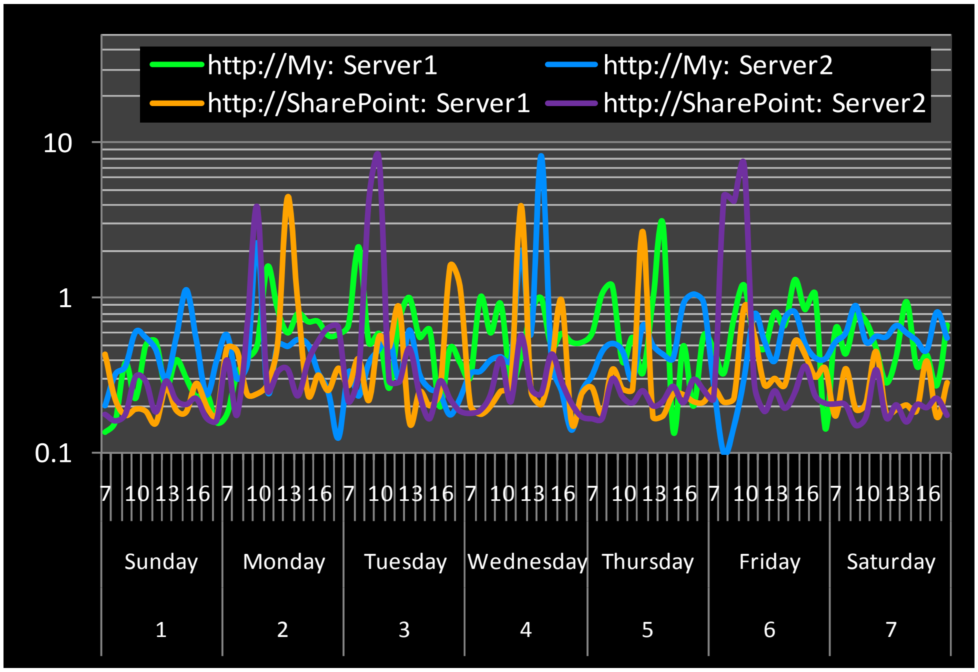


Figure 8. Performance chart after Phase 1 completion

In Phase 2, the SharePoint Operations team implemented the changes that the product group suggested—migrating to 64-bit hardware and performing additional fine-tuning, such as implementing differential backups Monday through Friday instead of full backups. The team then realized additional performance improvements, including the following:

- Average site rendering time was 1.39 seconds for self-provisioned team sites and personal employee sites.
- One-week site rendering time was 0.15 seconds at best, and 114.79 seconds at worst.
- Typical spike duration was 5 minutes to 10 minutes.

Overall, these changes stabilized the smaller spikes and revealed an underlying pattern. That is, spikes became more regular and occurred during peak usage hours from 10 A.M. until 2 P.M., as shown in Figure 9.

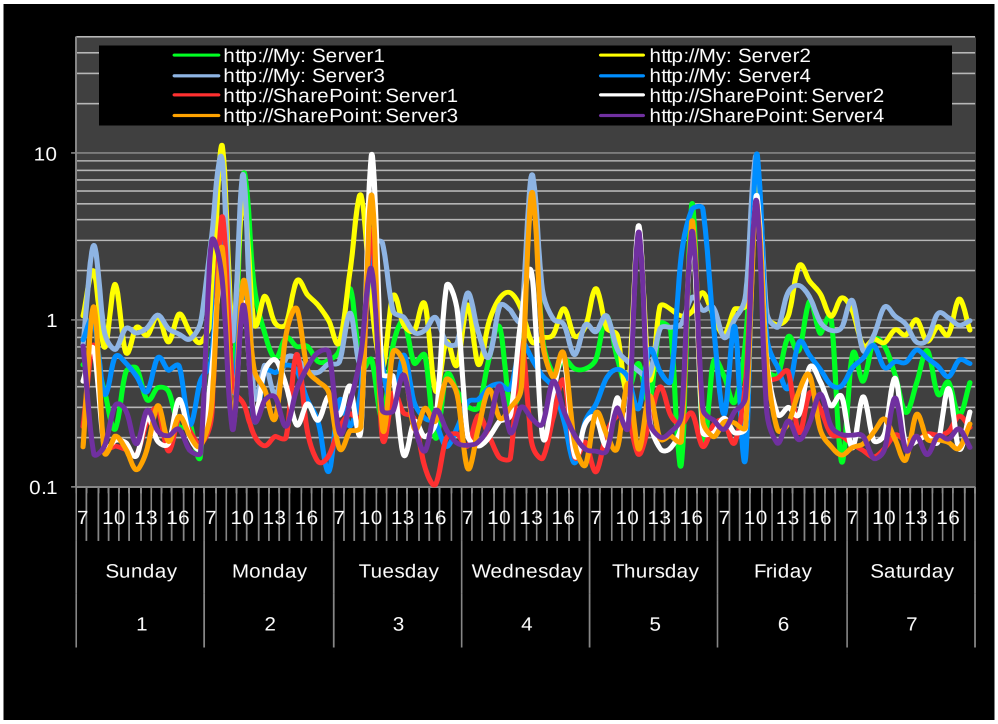
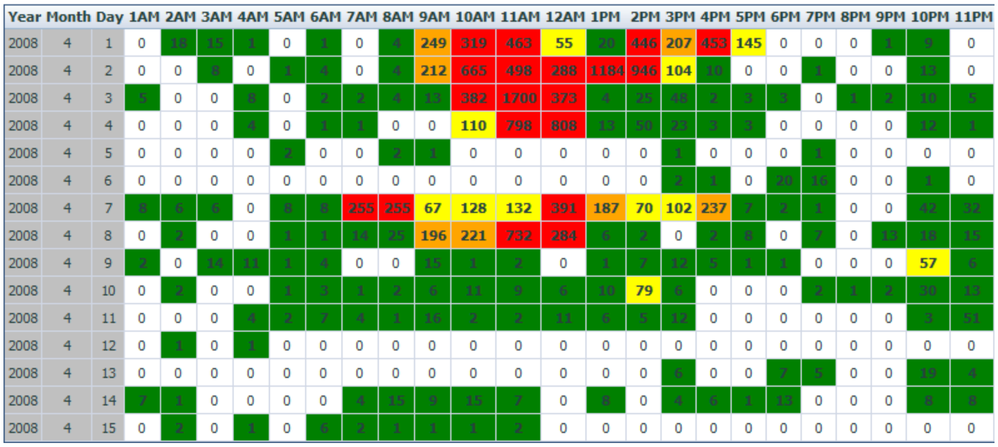


Figure 9. Performance chart after Phase 2 completion

During Phase 3, changes introduced to the network configuration resolved the persistent spikes in render times. This final resolution was possible because the activities performed during the previous phases removed “noise” (not relevant information) from the performance statistics and enabled the SharePoint Operations team to resolve the underlying issue.

One persistent occurrence was SQL Server blocking, which the SharePoint Operations team assumed was caused by the SQL Server configuration. Yet, after many configuration changes such as mitigating large sites and lists, SQL Server blocking persisted. The SharePoint Operations team investigated both upstream and downstream causes of SQL Server blocking from the beginning. Separating self-provisioned team sites and personal employee sites into individual farms helped to achieve better tracking of statistics, yet both farms shared the same SQL Server storage subsystem. By looking further upstream during Phase 3, the SharePoint Operations team realized that the SQL Server configuration was not the chief root cause—the underlying network was. Specifically, the NIC configuration prevented delivery of the SQL Server payload to clients because the single NIC card could not handle the traffic volume and created congestion.

After reconfiguring the NICs on front-end servers, the SharePoint Operations team noticed a dramatic decrease in SQL Server blocking, as shown in Figure 10.



Year	Month	Day	1AM	2AM	3AM	4AM	5AM	6AM	7AM	8AM	9AM	10AM	11AM	12AM	1PM	2PM	3PM	4PM	5PM	6PM	7PM	8PM	9PM	10PM	11PM
2008	4	1	0	18	15	1	0	1	0	4	249	319	463	55	20	446	207	453	145	0	0	0	1	9	0
2008	4	2	0	0	8	0	1	4	0	4	212	665	498	288	1184	946	104	10	0	0	1	0	0	13	0
2008	4	3	5	0	0	8	0	2	2	4	13	382	1700	373	4	25	48	2	3	3	0	1	2	10	5
2008	4	4	0	0	0	4	0	1	1	0	0	110	798	808	13	50	23	3	3	0	0	0	0	12	1
2008	4	5	0	0	0	0	2	0	0	2	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0
2008	4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	20	16	0	0	1	0
2008	4	7	8	6	6	0	8	8	255	255	67	128	132	391	187	70	102	237	7	2	1	0	0	42	32
2008	4	8	0	2	0	0	1	1	14	25	196	221	732	284	6	2	0	2	8	0	7	0	13	18	15
2008	4	9	2	0	14	11	1	4	0	0	15	1	2	0	1	7	12	5	1	1	0	0	0	57	6
2008	4	10	0	2	0	0	1	3	1	2	6	11	9	6	10	79	6	0	0	0	2	1	2	30	13
2008	4	11	0	0	0	4	2	7	4	1	16	2	2	11	6	5	12	0	0	0	0	0	0	3	51
2008	4	12	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2008	4	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	7	5	0	0	19	4
2008	4	14	7	1	0	0	0	0	4	15	9	15	7	0	8	0	4	6	1	13	0	0	0	8	8
2008	4	15	0	2	0	1	0	6	2	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10. Resolution of SQL Server blocking

Changing the NIC configuration eliminated SQL Server blocking. After this change and the other changes performed during Phase 3, the environment realized much-improved performance. The average URL rendering time was 0.8 seconds, spikes lasted fewer than 5 minutes, and the IIS server log jam cleared. Figure 11 shows the performance chart after Phase 3.

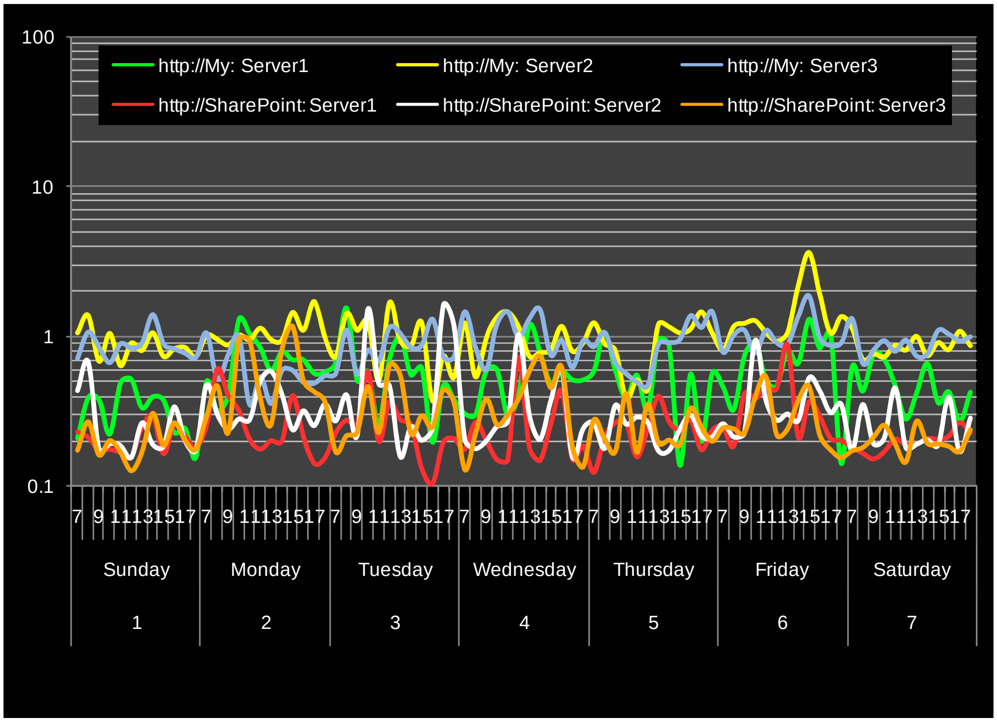


Figure 11. Performance chart after Phase 3 completion

Future Activities

The effort of optimizing the SharePoint environment did not end with the resolution of the root cause of SQL Server blocking. The SharePoint Optimization team plans to make further improvements, such as deploying new performance rollups from the product group, updating the environment to run on Windows Server 2008 and Microsoft SQL Server 2008, and sharing its findings with the product group to improve the code of future SharePoint releases. In addition, the SharePoint Operations team plans to perform the following activities:

- Continue to monitor performance and table fragmentation.
- Continue database maintenance and re-indexing.
- Replicate maintenance activity to other regions and data centers.
- Resume investigation of IIS memory usage if appropriate.
- Pursue applicable hotfix releases for SharePoint Server.

Note: *The latest performance rollup for SharePoint Server is available at http://technet.microsoft.com/en-us/office/sharepointserver/bb735839.aspx.*

BEST PRACTICES

In the course of performing investigations, gathering data, and making changes to optimize the environment, the SharePoint Services team developed best practices for front-end and back-end subsystems as well as operations processes. One of the most important best practices in troubleshooting and optimizing a SharePoint infrastructure is to establish a baseline and use structured operations and diagnostic processes, including evaluating possible causes, auditing configurations, implementing changes, and evaluating the results. Following structured processes enabled the SharePoint Operations team to narrow down the possible causes of long site render times and resolve the underlying issues. The following best practices can help other organizations optimize their SharePoint environments.

Best practices for the front end include:

- **Run IIS version 7.0 on 64-bit servers** Memory and CPU are common performance optimization factors for SharePoint Server. Using 64-bit hardware increases the amount of usable memory, which helps to maintain a healthy system state for worker processes.
- **Use a front-end and back-end NIC configuration for IIS** During peak load times, as many people access SharePoint sites, the NIC traffic increases. Using dedicated NICs for connections to the SQL Server back end and the clients provides better load distribution. Using dedicated NICs also provides more-accurate statistics and helps with troubleshooting traffic congestion issues by segregating the front-end and back-end traffic.
- **Load balance client traffic** The SharePoint Operations team uses NLB for balancing client traffic. It is a best practice to load balance incoming traffic for optimal user experience and server utilization.
- **Use IIS compression for static content** The SharePoint Operations team ensures that static compression is enabled to conserve traffic and server resources.
- **Enable caching** Page output caching on front-end servers reduces CPU utilization on front-end servers by storing compiled ASP.NET pages in RAM. Enabling this setting resulted in performance gains for the SharePoint Operations team. BLOB caching helps to relieve load on back-end servers by caching static content and not accessing databases when it is requested.

Best practices for the back end include:

- **Limit database size to enhance manageability** When databases grow, they can become less manageable for backup and restore operations, or for troubleshooting. The SharePoint Operations team uses a 100-GB limit.
- **Allocate storage for versioning and the recycle bin** When designing the environment, an organization should consider business needs, such as versioning, and ensure that adequate disk space and I/O are available to accommodate them.
- **Use quota templates to manage storage** Microsoft IT uses standardized configuration templates in all possible and practical scenarios, including quotas. Using quota templates helps preserve a standard environment, which reduces administrative overhead.
- **Manage large lists for performance** Having large lists by itself is not necessarily a performance issue. When SharePoint Server renders the many items in those lists, that can cause spikes in render times and database blocking. One way to mitigate large lists

is to use subfolders and create a hierarchical structure where each folder or subfolder has no more than 3,000 items.

- **Separate and prioritize data among disks and create disk groups for specific data** Because available disk I/O throughput is so important for optimal SQL Server performance, identifying the read/write patterns of services and dedicating SAN LUNs to them results in better performance than using many service types with the same disk group. The SharePoint Operations team takes this idea a step farther and uses dedicated partitions for data.

Best practices for processes and operations include:

- **Follow established processes, such as MOF** Microsoft IT follows industry-standard operations processes, and it uses typical diagnostic tools such as cause/effect diagrams. Following structured processes helps produce repeatable results and helps ensure that team members understand the sequence of steps necessary to evaluate root causes and resolve performance issues. Structured processes also help with other operations tasks, such as configuration auditing and change management.
- **Establish a baseline for performance comparison** Having a performance baseline is vital. After Microsoft IT designed and sized the environment, and installed SharePoint Server, the initial user experience was positive with no performance issues. Because the SharePoint Operations team recorded baseline performance statistics, the team was able to identify when performance issues started to occur and take corrective measures. Having a baseline also helps with troubleshooting and optimization because later results can be compared to the original and used as a benchmark.
- **Make data-driven decisions** Often, multiple root causes exist for performance issues. At Microsoft, back-end, front-end, and process optimizations resolved the underlying causes. This resolution was possible because the SharePoint Operations team carefully gathered data and made choices based on the analysis of that data.
- **Simplify topologies** Gathering accurate performance data in complex topologies can be difficult because services and servers compete for the same resources. This was the case initially with self-provisioned team sites and personal employee sites at Microsoft. Using the simplest topology possible eliminates statistical noise and provides operations staff with the clearest data possible.
- **Have a clear operating level agreement (OLA) with network, SQL Server, and backup teams** Because SharePoint Server depends on underlying network infrastructure and SQL Server, the team responsible for SharePoint operations should have an agreement in place that details expectations and responsibilities for all teams involved.
- **Periodically standardize environment configuration** With optimization efforts, minor changes can cause configuration inconsistencies or undocumented settings. It is a best practice to perform audits to ensure that documented and as-found configurations match.
- **Investigate upstream and downstream possible root causes** Microsoft IT examines possible root causes by considering the data flow and components involved in the transmission of data, and then analyzing these aspects according to their degree of separation. For example, a user may report an issue. Investigating downstream means considering the bandwidth and connection to a front-end server, and then any load balancing in place, and then the back-end server, and so on.

- **Routinely verify and audit system configuration** An organization should verify and audit configurations to ensure that they comply with documented specifications.
- **Assign ownership of operations responsibilities to individuals, not teams** By assigning ownership to individuals, Microsoft IT can increase accountability. It is all too easy for teams to overlook aspects in the course of operating an environment.
- **Use different teams or individuals for auditing** A specific area of ownership is configuration management and audit. Designs must be implemented correctly and checked periodically to ensure that the configuration changes only through proper documentation and approval. Having deployment staff verify designs is similar to having developers verify their own code. It is best to have another team or individual perform periodic audits to ensure conformity to specifications.
- **Monitor all aspects of the environment** Microsoft IT uses automated tools, such as Microsoft System Center Operations Manager, to monitor the environment. The SharePoint Operations team complements this by using custom tools and analyzing reports in weekly meetings to detect trends in user experience, overall system health, and health of the front-end and back-end components.
- **Maintain documentation** The SharePoint Operations team made sure that all documentation was up to date, and that multiple mirrored repositories existed in case a single source was unavailable. Part of the documentation update effort entailed ensuring that audits occurred and documenting results. The SharePoint Operations team gave ownership of specific documents to team members to maintain. This includes training and onboarding documents for new team members, in addition to revising the knowledge base for Tier 1 front-line monitoring operators.
- **Evangelize change management** Documenting the repeated changes made during the optimization effort resulted in the need to discuss change management. The SharePoint Operations team created documentation when making changes to ensure that change requests undergo the appropriate processes, are verified, and are approved.

CONCLUSION

Optimizing the performance of any environment, and especially a SharePoint environment, requires attention to the baseline performance, monitored statistics, and structured operations processes. For Microsoft IT, these processes are systematic and follow an order whereby findings from root cause analyses result in change requests, changes, and further analysis. However, real-life operations and optimization do not always follow such a structured model. In the case of this performance optimization effort, Microsoft IT worked with other teams, sometimes investigated leads that ultimately were not material to determining root causes (such as fragmentation), and acted upon configuration changes that did not always resolve the underlying issues.

Fine-tuning of optimization and performance is complex because it involves many subsystems that are interrelated and interdependent. In the case of Microsoft, the overall environment depended on front-end and back-end subsystems that had yet other dependencies in the underlying network infrastructure. By focusing on these components and investigating upstream and downstream possible causes, an organization can determine root causes through technical tools such as logs and performance data and analysis tools such as cause/effect diagrams. In the process of analyzing possible root causes, even if the organization does not immediately find the causes, systematically working upstream and downstream will still improve performance. This may make a better understanding of the underlying behavior easier to achieve.

For More Information

For more information about Microsoft products or services, call the Microsoft Sales Information Center at (800) 426-9400. In Canada, call the Microsoft Canada information Centre at (800) 563-9048. Outside the 50 United States and Canada, please contact your local Microsoft subsidiary. To access information through the World Wide Web, go to:

http://www.microsoft.com

http://www.microsoft.com/technet/itshowcase

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

APPENDIX: URL PING TOOL

```
// Create a 'WebRequest' object with the specified url.
WebRequest myWebRequest = WebRequest.Create(url);
myWebRequest.UseDefaultCredentials=true;
myWebRequest.Timeout = 90000;
// If trace enabled - set the Trace
if (Trace == "Y")
{
   WebHeaderCollection myWebCollection = new WebHeaderCollection();
   myWebCollection.Add("Trace", "1");
   myWebRequest.Headers = myWebCollection;
   // Send the 'WebRequest' and wait for response.
   DateTime startTime = DateTime.Now;       // Start time
   myWebRequest.GetResponse();              // Web Request
   DateTime stopTime = DateTime.Now;        // End Time
   TimeSpan duration = stopTime - startTime;     // Time Diff
   succededStr = "1";
   requestTime = startTime.ToString();
   elspsedSecs = duration.ToString().Substring(6);
}
catch (Exception e)
{
   details = e.Message;
}
```