Spring 2010

A Strategy for Improving Performance On a Sharepoint Social Computing Portal

Matt Adkins
Regis University

A STRATEGY FOR IMPROVING PERFORMANCE ON A SHAREPOINT SOCIAL COMPUTING PORTAL

A THESIS

SUBMITTED ON DAY OF MONTH, YEAR

TO THE DEPARTMENT OF INFORMATION SYSTEMS

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

MASTER OF SCIENCE IN

COMPUTER INFORMATION TECHNOLOGY

BY

Matt Adkins

APPROVALS

Robert G. Bowles, Thesis Advisor

James A. Lupo

Stephen D. Barnes

D.M. Likarish

Abstract

An important usability rule for any web site is the concept of speed. Failing to provide prompt pages and data will result in a negative view of the site and ultimately a lack of usership. In spite of this, many organizations implement web sites without a clear strategy regarding performance.

This project explores three database strategies to consider when deploying a Microsoft SharePoint website with a social computing usage style. Although all of the strategies do not provide significant performance gains, the study illuminates several important factors that will increase performance in sites that use other usage styles. To properly explore each database strategy, specially designed tests were executed against a medium-size SharePoint server farm. The website performance statistics were recorded and compared to measure the effect of different configurations.

The performance statistics showed a performance increase when site collections per database are limited to a specific amount. It was also discovered that large SharePoint content databases do not directly affect performance assuming three specific conditions are met. The third concept that was studied indicated that the implementation of external BLOB storage will increase performance assuming the average file size in the database is fairly large.

Table of Contents

List of Figures

List of Tables

Executive Summary

In spring 2007, a private East Coast University implemented several Microsoft Office SharePoint 2007 websites in order to meet the collaboration and content management needs within the University. Not fully understanding how the collaboration website would be used by the University, it was deployed with a “hands-off” approach by allowing the end-users to drive the direction of its usage. It wasn’t long before multiple departments realized the extensibility and ease of use Microsoft SharePoint offered for their daily operations. For these reasons along with many other factors, the University has chosen to implement a student portal using SharePoint which would service its fifty thousand student population. In comparison to the original SharePoint websites, this deployment will be much more structured and carefully planned out to ensure a smooth rollout and optimum performance.

The student portal that the University envisions will utilize a usage style that SharePoint is not commonly focused on: social computing. The aim is to provide a highly customizable, social networking student portal that will enhance collaboration between student and instructor and provide ease of access to the University’s resources. This will be accomplished using the SharePoint “My Site” feature in which each student receives a personalized SharePoint site. This would equate to over 50,000 separate SharePoint site collections which would require a different database strategy than the University’s current implementation.

Although general best practices for SharePoint web farms are widely available, the social computing usage style has several unique characteristics that affect

performance in ways that other usage styles do not. In order to produce the utmost performance for this particular usage style, this study carefully considers the optimum size of the content databases and the maximum number of site collections per database. In addition, this study compares the performance differences between database hosted binary large object (BLOB) data files and external BLOB storage.

In order to determine the optimum configuration based on those three factors, multiple tests were carried out on several SharePoint test environments and configurations. The results of these tests were closely monitored using a variety of data collection tools. Carefully following the resulting guidelines that the test results have helped to determine can significantly increase the performance of SharePoint environments that use the social computing usage style.

Chapter 1 – Introduction

The University's goal is to produce a student portal that will be a hub for common student activities such as the following: collaboration via email and instant messaging, access to information regarding housing, financial aid, and grades, and personal file storage. There are several key components that will help ensure the success of this student facing portal. Integrating the site with the University's other academic systems for example is important to making the site relevant. A product called Microsoft SharePoint was chosen to host this student portal due to the plethora of out-of the-box features it provides such as search, content management, advanced web portal functionality, and collaboration features. In addition, Microsoft SharePoint met the University's requirements regarding extensibility and unified architecture.

However, if the student portal is not adequately responsive to its users, the amount of site usage will be adversely affected. Multiple factors contribute to a highly responsive SharePoint farm such as database server and configuration, operating system version and configuration, disk sub-storage, and many other items. Several "best practices" documents have been published by Microsoft and other entities that provide direction in these areas. However, the usage style of this student portal presents several unique characteristics that have not been widely researched. This study will present three specific strategies that help to improve performance when using this specific usage style.

Before these performance strategies can be discussed, however, it is important to understand how a usage style can affect Microsoft SharePoint web farm design and performance. The majority of available research concerning SharePoint usage is focused on enterprise business solutions such as team and departmental sites. The usage style of a team site is dramatically different than that of the social computing characteristics of the planned University student portal. The book entitled "Social Computing for Microsoft SharePoint 2007" (Draper, 2009) provides excellent insight in what to expect with this form of usage. For instance, a typical SharePoint team site would consist of workflow processing, heavy in-browser Excel calculations, InfoPath forms, and large document repositories. In contrast, a social computing site often consists of four major usage types: Social Media, Social Bookmarking, Social Networks, and Social Communication (Draper, 2009). Social Media is the use of interactive media such as video and audio. The goal of this media would be to evade the classic one-to-one media distribution style (between one instructor and one student) by providing a web architecture that promotes dialogues with many students. The implementation of Social Bookmarking in the student portal will allow students to save, review, and share content with peers. The student portal will promote Social Networks which is defined as the building of groups that share common interests. The web portal will also promote Social Communication which enables the communication of ideas and presence information. Thus it is the goal of the University to provide a student web portal that unifies information from multiple sources and that promotes social interaction with this information.

This study explores three methods of providing the utmost performance in the University's Microsoft SharePoint portal that has a usage style comprised of several social computing characteristics. Contained herein are findings and recommendations regarding the optimum number of sites contained in each SharePoint content database. Recommendations are also provided regarding the optimum size of SharePoint content databases. In addition, this study also explores the subject of external SQL BLOB storage and how it can dramatically lower database size and potentially provide improved performance in a SharePoint portal.

Chapter 2 – Review of Literature and Research

2.1 Social Computing with SharePoint

In order to properly design a robust social computing portal, it is necessary to understand what kinds of activity will be present and how these activities will stress the SharePoint farm differently than that of a typical usage types. Depending on the usage of the site, the access to the database may be mostly read only or an even mixture of read/write access. As mentioned earlier, Brendon Schwartz, Matt Ranlett and Stacy Draper categorized most types of social computing activities into four major areas (Draper, 2009): media, bookmarking, networks and communication. Although many of these activities can only be accomplished through custom design, SharePoint does provide a social computing site template called "My Site" which serves as a foundation for developers to build upon.

A SharePoint My Site, as with any other SharePoint site, contains several web parts that provide out-of-the-box functionality such as announcements, lists, and calendars. However, the SharePoint My Site also contains multiple unique web parts and content rollups that are geared for a social computing usage style. The following are a few examples of these unique web parts: My Blog, Outlook Web Access, Colleague Tracker, and In Common Between Us (Sterling, 2007). The blogging functionality allows any user to create a blog and post comments on other blogs. Outlook Web Access web parts allow for integration with Microsoft Exchange email infrastructures. Colleague Tracker provides a list of the user's colleagues and their recent profile changes. The In Common Between Us web part compares two users based on Active Directory (or other directory services) information.

Although the My Site may appear to be the same as any SharePoint site, its site architecture is much different. These sites are not automatically created for all users, but are created only when users click the “My Site” link from the portal SharePoint site. The site is then created within the Shared Services Provider (a SharePoint web application that provides several core services such as site provisioning) as a separate site collection. This potential for a large number of site collections has significant impact on site architecture and thus could affect the overall performance of the portal if not designed and maintained appropriately.

The My Site architecture and site governance provides insight concerning what one can expect regarding performance. In order to illustrate the differences between various usage styles, consider the site structure of a typical team site portal in the following figure:

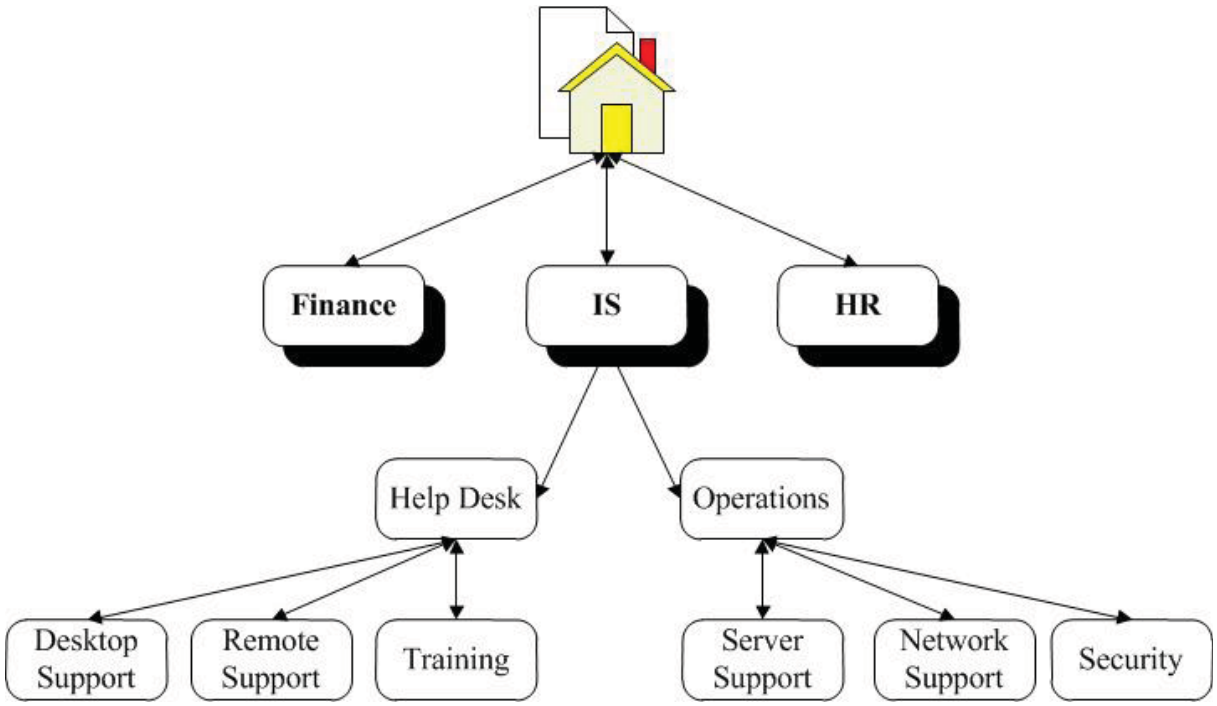

Figure 1: Typical Team Site Structure

Note that team sites typically have multiple sub-sites nested under a central top-level site. The top-level site in this figure would be called a site collection and would reside in a single database. Nesting multiple team sites in a single site collection allows for cross-site sharing of data, a shared security structure, and for a logical namespace.

On the other hand, the site structure of a My Sites portal is much more disjointed than that of a typical team site. This site template will create a site collection for every user who logs into the SharePoint portal. The following figure illustrates the site structure of a typical social computing portal:

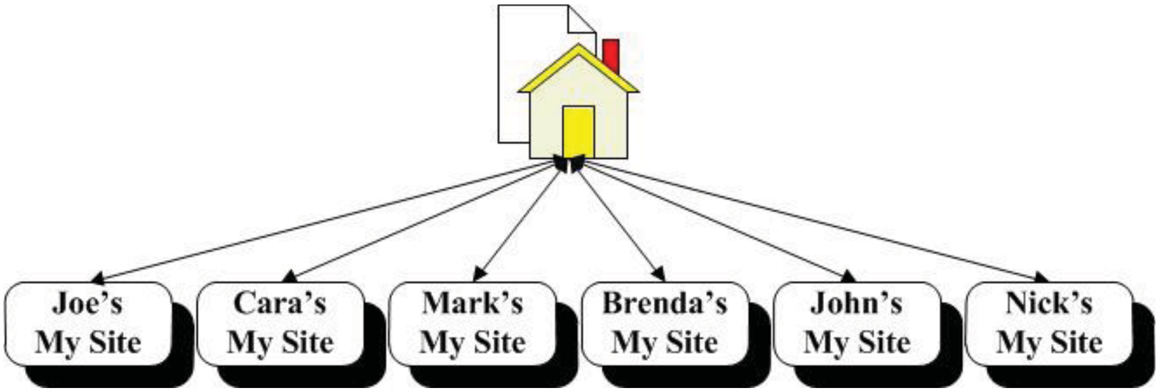

Figure 2: Social Computing Site Structure

As the figure shows, instead of a top-down site structure, social computing portals based on the My Site template span across multiple site collections with each having a separate namespace and security structure. Due to this unique site structure, a poor site governance plan could result in performance degradation. Therefore, it is important to closely examine two database related items: maximum number of site collections per content database and the optimum size of the content database.

2.2 Spreading Site Collections across Multiple Databases

An often overlooked component in a SharePoint farm is that of the database server role. SharePoint relies heavily on the database server because this is where “it stores all configuration data and content” (Chaganti, 2009). However, the configuration of the actual database server is not the only performance item to be concerned about. A wise database storage plan will tremendously affect the performance of SharePoint sites. One particular database strategy that can impact performance is limiting the number of site collections that reside in each content database.

As mentioned earlier, the nature of My Sites architecture leads to an extremely large number of site collections in the web application. By default, the first 15,000 site collections will be stored in a single content database (Callahan, 2008). The SharePoint Central Administration website provides the ability to specify multiple databases for a web application and allows the administrator to specify the maximum number of site collections that should be created on each database. For example, if the My Sites web application has three content databases specified, SharePoint will create new site collections across the three databases in a round-robin fashion.

However, estimating the appropriate number of site collections each database should contain is a difficult task. An excellent place to start researching this topic is the Microsoft Office SharePoint Server 2007 technical library located at the Microsoft TechNet website (http://technet.microsoft.com). Within this online technical library, there are multiple software boundaries, hard limits, and theoretical limits described that should be carefully considered. For instance, a theoretical limit of 50,000 site collections per content database is suggested (Office IT and Servers User Assistance Team, 2009). Although this may be a maximum number to keep in mind, it does not indicate at which point performance will decline.

2.3 Optimum Content Database Size

As previously discussed, estimating the appropriate number of site collections each database should contain is a difficult task. A good place to start is with the provided best practices regarding database sizes from Microsoft. Bill Baer published a whitepaper for Microsoft entitled “Planning and Monitoring SQL Server Storage for SharePoint:

Performance Recommendations and Best Practices" (Baer, 2008) that explains the recommended maximum database size. Although this whitepaper is targeted for single site collection portals, it does provide a basis from which to build a series of experiments. Baer discourages databases larger than 100GB and recommends that these larger databases be limited to a single site collection. The single site collection recommendation won't be possible in a My Sites social computing portal, but the maximum size recommendation can certainly be used during architecture planning.

The maximum recommended size of a SharePoint content database isn't necessarily the best size for maximum performance. To further complicate matters, an in-depth study performed by Russ Houberg from Knowledge Lake Inc. challenges Microsoft's "Golden Rule" regarding content database sizes (Houberg, 2009). This 100GB limit isn't due to a SQL server limitation since Microsoft SQL Server can accommodate much larger databases. Rather, Houberg believes the Microsoft recommendation is based on three factors: backup and restore time requirements, large list contention, and storage subsystem. An example of a backup and restore requirement would be a company that has a service level agreement with its customer to provide site restores within four hours. Large SharePoint lists can cause contention if the lists and or libraries exceed 5 million items (Office IT and Servers User Assistance Team, 2009). The third factor that Russ Houberg references is that the storage subsystem must be robust enough to handle the intensive disk I/O. Houberg describes the recommended storage architecture for SharePoint in his whitepaper entitled: Scaling SharePoint 2007: Storage Architecture (Houberg, 2009). Assuming these factors can be mitigated, Houberg has established that content databases can safely grow to 400GB before suffering from significant performance degradation. While list contention and the service level agreement requirements are not necessarily limitations in the University's proposed portal, the storage subsystem is a concern. Due to the University's current shared SQL Server disk storage architecture, it is recommended that the 100GB size limit be considered.

According to the Microsoft guidelines for acceptable performance, a web application should not exceed 100 content databases (Guinn, 2009). This recommendation creates significant concern due to the large number of site collections projected in the University's My Site portal. For example, if each site collection was 250MB there can only be a little over 400 site collections per database (to adhere to the 100GB database recommendation). Furthermore, assuming the same site collection size, the 50,000 site collection projection for the student portal would require over 100 content databases. This revelation shows the necessity of determining if the 100GB database recommendation also applies to My Site web applications.

2.4 External Storage with BLOB

One way to avoid the 100GB database guideline is to manipulate the way Microsoft SharePoint stores objects in the database. By default, all files such as spreadsheets, documents, pictures, and videos are stored as BLOBs (Binary Large Object data files) inside a single table in the database. This database architectural model was discarded by the majority of the enterprise content management industry years ago due to the lack of scalability and significant database contention it produces (Thumma, 2008). Even Microsoft has estimated that up to 80 percent of the data stored in SharePoint content databases is BLOB data objects (Cherny, 2009). This indicates that only 20 percent of the content database consists of data that Microsoft SQL Server was designed to use: relational data. However, a few months after the release of SharePoint 2007, Microsoft published a Hotfix that allows external storage of this non-relational BLOB data through the use of an API called ISPExternalBinaryProvider (Microsoft Corporation, 2007). This Hotfix would allow for the storage of these files in other mediums such as a file server share. Unfortunately, lack of Microsoft documentation and deployment support has hindered usage of this storage API in most enterprises. Since the release of this Hotfix, several companies and open source groups have released software that utilizes the BLOB API.

In addition to the large database size that results from storing BLOB objects inside Microsoft SQL Server, there are also limitations with maximum file sizes and heavy I/O operations. The maximum file size that can be uploaded to SharePoint is dictated by a Microsoft SQL Server 2005 limitation. Specifically, the VARBINARY (MAX) datatype, which allows documents to be stored directly in the database, has a hard limit of 2GB of storage (Walters, Coles, Rae, Ferracchiati, & Farmer, 2008). However, a largest drawback of SQL BLOB storage is not the maximum file size, but rather the performance hit of large files in the Microsoft SQL database. A collaborative study between researchers at University of Berkley and Microsoft Research has shown that "BLOBs greater than 1MB are more efficiently handled by a file system (Sears, Ingen, & Gray, 2006).

Furthermore, file-streaming performance is hindered since Microsoft SQL Server performs its operations at the page level and is required to perform lock management.

However, externalizing this type of data allows one to take advantage of the many benefits of dedicated server file storage and frees the SQL server to focus on relational data as illustrated in the following figure (Cherny, 2009):

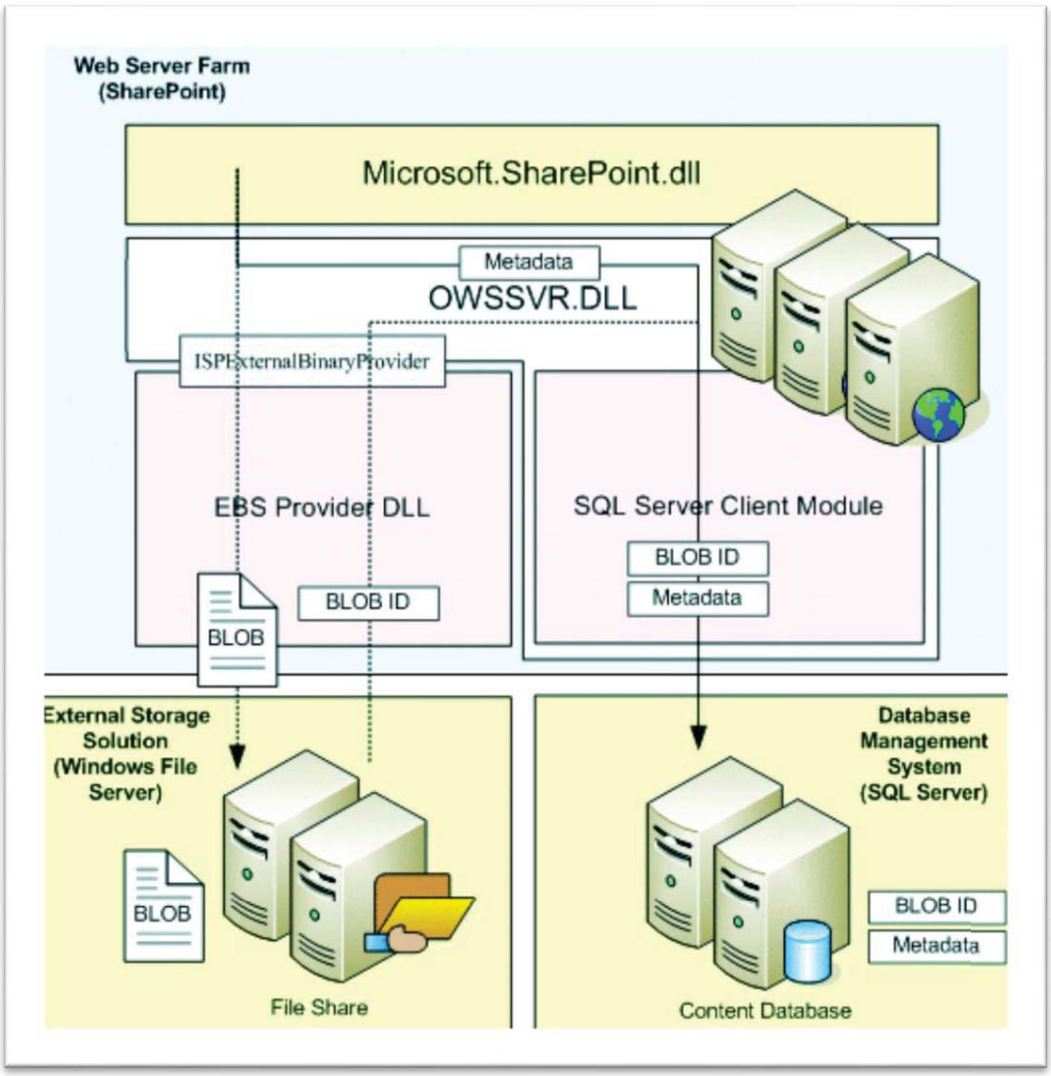

Figure 3: External storage of BLOB objects

Since content database size and number will be a significant hindrance the University's SharePoint portal, the concept of external storage of this type of data should be researched and tested. The concept of reducing content database size by 80 percent would mitigate maximum database size concerns and quite possibly improve performance.

Chapter 3 – Testing Architecture and Methodology

3.1 Architecture

All hardware and software testing was performed on a VMware ESX 3.5 and VirtualCenter 2.5 infrastructure with the exception of the Microsoft SQL 2005 database cluster. The ESX environment was hosted by three Dell PowerEdge 1950 servers which were configured with 2 x Xeon 2.99GHz quad-core processors and 32GB memory. In addition, three 500GB RAID5 LUNs were provided to the ESX host servers. This virtual server testing environment allowed for quick server builds via templates and easy reversion through snapshots.

A total of nine virtual servers were built to host the SharePoint, Active Directory, OCS, and Visual Studio testing software. The following table describes each virtual server's hardware and software role:

Server Name	OS	CPU	Memory	Role
MossWeb01	Windows Server 2003 SP2	CPU: 4 x 2.99GHz	6GB	MOSS 2007 SP2
MossWeb02	Windows Server 2003 SP2	CPU: 4 x 2.99GHz	6GB	MOSS 2007 SP2
MossWeb03	Windows Server 2003 SP2	CPU: 4 x 2.99GHz	6GB	MOSS 2007 SP2
MossIndex	Windows Server 2003 SP2	CPU: 2 x 2.99GHz	4GB	MOSS 2007 SP2
Test01	Windows Server 2003 SP2	CPU: 1 X 2.99GHz	2GB	Visual Studio 2008
Test02	Windows Server 2003 SP2	CPU: 1 X 2.99GHz	2GB	Visual Studio 2008
OCS01	Windows Server 2003 SP2	CPU: 2 X 2.99GHz	8GB	OCS / Auxiliary SQL

AD01	Windows Server 2008 R2	CPU: 1 X 2.99GHz	4GB	Active Directory 2008
AD02	Windows Server 2008 R2	CPU: 1 X 2.99GHz	4GB	Active Directory 2008

Table 1: Virtual Server Configuration

Two physical servers were used to host the Microsoft SQL 2005 server role for SharePoint. These servers were configured in an active / passive Microsoft SQL 2005 cluster with two instances. One instance hosted the My Site databases while the other instance hosted the main SharePoint portal, configuration, and indexing databases. The following table describes the Microsoft SQL and VMware ESX infrastructure:

Server Name	OS	CPU	Memory	Role
MossSQLa	Windows Server 2003 SP2	CPU: 4 x 2.99GHz	8GB	SQL 2008
MossSQLb	Windows Server 2003 SP2	CPU: 4 x 2.99GHz	8GB	SQL 2008
ESX01DEV	VMware vmkernel	CPU: 8 x 2.99GHz	32GB	ESX 3.5
ESX02DEV	VMware vmkernel	CPU: 8 x 2.99GHz	32GB	ESX 3.5
ESX03DEV	VMware vmkernel	CPU: 8 x 2.99GHz	32GB	ESX 3.5

Table 2: Physical Server Configuration

The following diagram symbolizes the farm architecture described in Table 1 and Table 2 which hosted all three test scenarios. Servers in the blue section denote SharePoint 2007 server roles. The grey section denotes auxiliary servers that were needed for testing, load, and basic infrastructure operations and the orange section represents VMware.

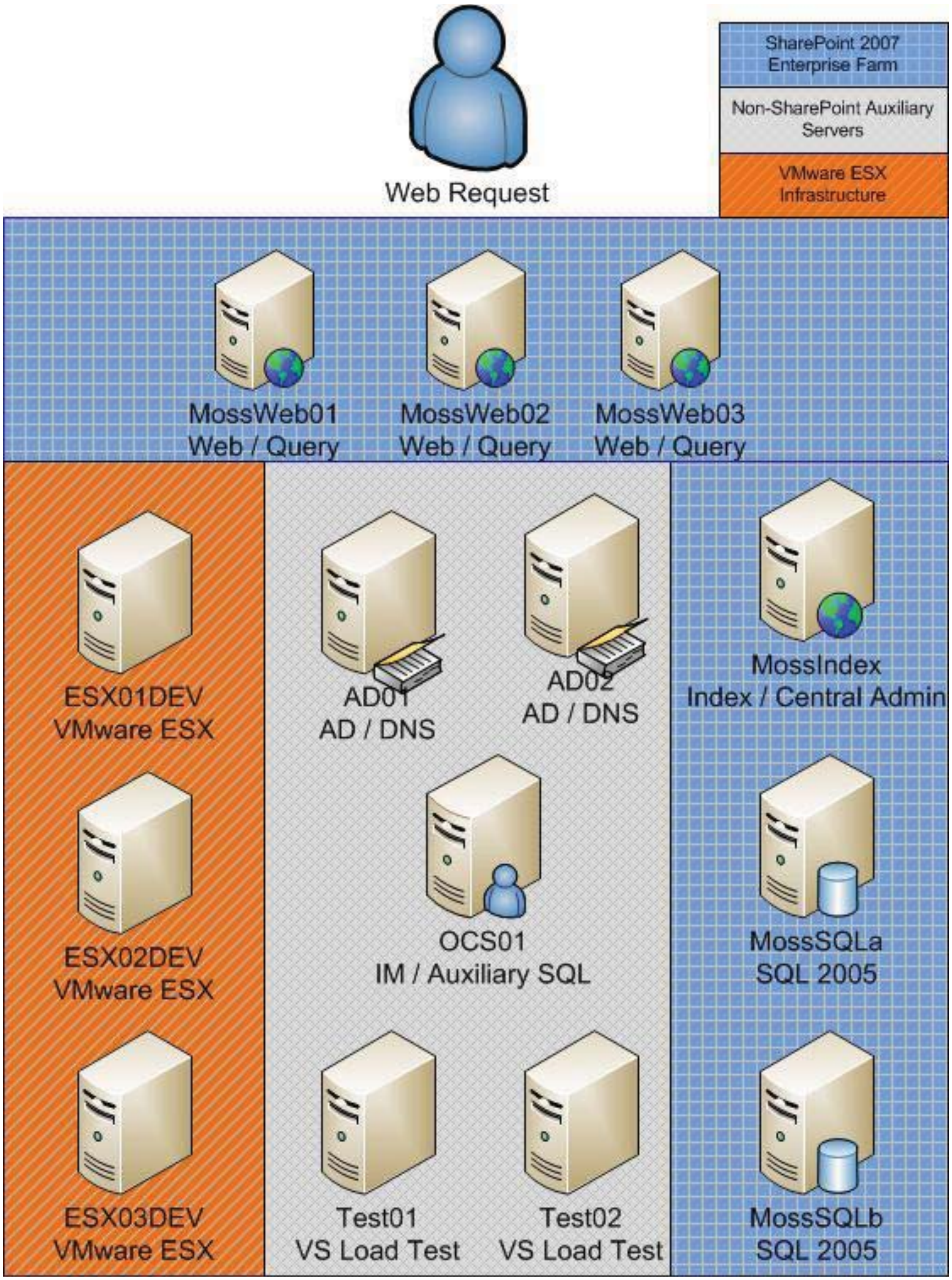

Figure 4: Virtual / Physical Farm Architecture

3.2 Special Configuration

In addition to the test farm structure, there are several configuration items that need to be pointed out which may have affected the performance results. All of these items were completed before any testing began. The three areas in which these configuration changes were made are SharePoint 2007, OCS 2007 R2, and the SQL Server 2005 instances.

Microsoft Office SharePoint Server 2007 was installed on each of the servers that hosted SharePoint web sites. Service Pack 2 for Windows SharePoint Services 3.0 and Microsoft Office Servers was installed since these updates included an important infrastructure change which tunes the performance of SharePoint. Each of the SharePoint web applications was configured to use Kerberos which has been shown to improve authentication performance over other authentication methods (Cherny, Using Kerberos for SharePoint Authentication, 2010). In addition to the above items, two forms of disk based caching were configured on each of the SharePoint web servers: BLOB Caching and Object Caching. BLOB Caching was enabled on each site in order to cache multiple file formats to disk and thus reduce repeated database access. The other form of caching that was configured was Object Caching which caches specific page items such as navigation. Both of these forms of caching have been shown to provide significant performance benefits in SharePoint 2007 (Hewlett-Packard Development Company, 2007). The last configuration items regarding the SharePoint farm is the search configuration. Queries to the search index were performed by the SharePoint web servers in order to simulate production behavior. However, search indexing was disabled during the tests.

Since instant messaging presence will be a key requirement of the University student portal, the test infrastructure included Microsoft Office Communication Server 2007 R2 (OCS 2007 R2). Enterprise edition of OCS 2007 R2 and Microsoft SQL Server 2005 was installed on a single virtual server (OCS01). Fifteen test users were provisioned instant messaging accounts in order to provide SharePoint pages the ability to pull real time presence information.

The same server that hosted OCS 2007 R2 (OCS01) also hosted a few others databases which were integral to the tests. The Visual Studio load testing databases resided on this server in order to prevent degradation on either of test servers (Test01 and Test02). Also, this server hosted a very basic database that SharePoint pulled data from using Business Data Connections (BDC). This was done in order to simulate the production activity of retrieving information from various housing and financial aid databases.

The two servers used to host the SharePoint databases were MossSQLa and MossSQLb. These servers were part of an Active / Passive cluster to mimic the University's existing SQL server architecture. All of the SQL server system databases, SharePoint databases and corresponding transaction logs were configured closely meet the recommendation outlined in Bill Baer's whitepaper entitled "Planning and Monitoring SQL Server Storage for SharePoint: Performance Recommendations and Best Practices" (Baer, 2008).

Regarding security within the test environment, all of the servers used Microsoft Forefront Antivirus for client security. However, there was not an antivirus solution in place for the SharePoint web applications. In addition, the web sites were not secured by

HTTPS and there wasn't any form of document encryption in place. Please see Chapter 8 and the Appendix regarding different methods of securing SharePoint and the corresponding performance implications one should watch out for.

3.3 Metrics for Testing

The Requests per Second (RPS) performance counter was used as the key metric in all test scenarios. A number of Active Directory users were simulated using the Microsoft Visual Studio Team System 2008 (VSTS) software which also monitored RPS and other relevant performance metrics. One RPS represented all HTTP PUTs and GETs in order to adequately test the performance of the farm. For example, Figure 5 shows a screenshot from VSTS which displays a single RPS even though the originating URL had 29 requests for information. To produce this single RPS, a simple test was configured that browsed to a specific URL on a SharePoint site. In order to display this single page, the browser had to download multiple items such as style sheets and images. As Figure 5 indicates, it took a total of .938 seconds to complete all 29 requests.

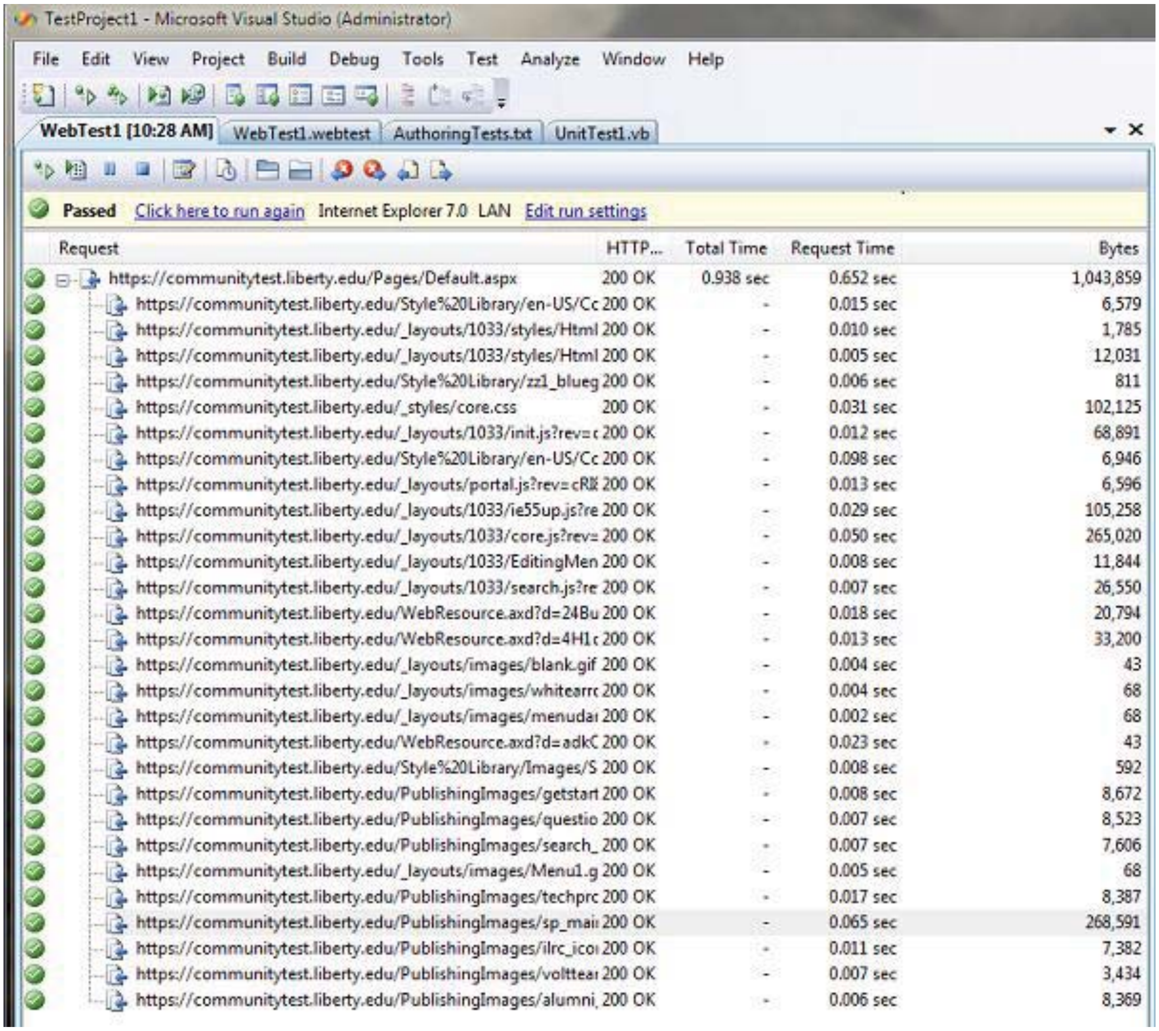

Figure 5: Example RPS

Visual Studio Team System 2008 was used to create each of the web and load tests which were then carried out by the two VSTS load agents (Test01 and Test02). Each web test was designed to stress different aspects of the SharePoint farm in order to resemble projected usage. In addition to the activity in each web and load test, other activity occurred on the portal such as Active Directory authentication, rendering OCS presence status, RSS feeds, and BDC connection data. All of these items together represent the expected production usage.

All of the tests run simultaneously for 25 minutes and were repeated three times to produce adequate sampling of test results (all tests were preceded by server reboots and a 5 minute web application warm-up). The amount of load (generated by simulated Active Directory users via the web tests) was gradually increased throughout the duration of each test. Any SharePoint document libraries or lists that were added to during the web tests were emptied after each test. This prevented degradation which would have occurred if these libraries/lists had exceeded more than 2000 items (Peschka, 2007).

The creation of My Sites and the associated content was accomplished using the "SharePoint 2007 Test Data Population Tool" (CodePlex, 2007). This CodePlex project contains multiple samples for creating effective web and load tests in a SharePoint environment. This tool allowed for creation of the necessary content and also provided a method for deleting the content between each test scenario.

3.4 Test Scenario #1: Site Collections per Database

The goal of this test was to determine the maximum number of site collections a single SharePoint content database can contain before significant degradation (measured by RPS) occurred. This test focused on the My Sites web application, although each My Site contained RSS web parts that retrieved information from the main Portal web application. The following table provides a description of the VSTS web tests that were used throughout Test Scenario #1:

Web Test	Location	% of Time	Actions
MySite_Public	My Site web	50%	View splash page of public My Site
			View another user's My Site
			Create a post on another user's My Site
			View another user's Document Library
			View JPEG in Document Library
			Perform a search for a specific user
			View one search result
MySite_Private	My Site web	30%	View splash page of private My Site
			View private Document Library
			Upload JPEG in Document Library
			View splash page of private My Site
			View private List
			Create list item
Portal_Home	Home web	20%	View splash page
			View news feed from internal site
			Perform a search for a specific group
			View one search result
			Create a post on a group site
			View a post on a group site
			View group Document Library
			View JPEG in Document Library

Table 3: VSTS Web Tests

The SharePoint 2007 Test Data Population Tool was used to clean the site collections after each test iteration in order to prevent the content database from growing larger than 100GB (doing so would have risked conflict with Test Scenario #2). The web tests were executed three times against each of the following configurations:

# My Sites	My Site Size	Content Database Size
1000	25MB	25GB
2000	25MB	50GB
3000	25MB	75GB
4000	25MB	100GB

Table 4: Test Scenario #1 Configuration

3.5 Test Scenario #2: Content Database Size

The goal of the second test was to determine how large a SharePoint My Site content database can grow before significant degradation (measured by RPS) occurred. The web test described in Table 3 was reused since the testing criteria were quite similar. However, before each iteration, the content database size was modified in order to provide an adequate performance comparison between databases smaller than 100GB and databases larger than 100GB. The following table outlines the configuration of this test scenario:

# My Sites	My Site Size	Content Database Size
500	150MB	75GB
500	200MB	100GB
500	250MB	125GB
500	300MB	150GB

Table 5: Test Scenario #2 Configuration

It is important to note that the size of this SharePoint content database was not increased by adding additional site collections. Creating additional site collections would have placed the test at risk of being tainted since it would have crossed over the software boundaries discovered in the first test scenario. Rather, the size of the content database was increased by enlarging each My Site using the SharePoint 2007 Data Population Tool. As with the first test scenario, each iteration was performed three times and content was cleared since the web tests are adding additional documents.

3.6 Test Scenario #3: External BLOB Storage

The goal of the third test was to determine if the use of external BLOB storage would increase the performance (measured by RPS) of SharePoint when applied to a My Site web application. The test configuration was designed to prevent conflict with the previous two test scenarios by only creating 500 site collections and keeping the content database under 100GB. This scenario involved six configurations as described in the following two tables:

# My Sites	My Site Size	Content Database Size	Average File Size
500	150MB	75GB	100KB
500	150MB	75GB	300KB
500	150MB	75GB	500KB

Table 6: Test Scenario #3 Configuration -- Before External BLOB

# My Sites	My Site Size	Content Database Size	Average File Size
500	150MB	11GB	100KB
500	150MB	11GB	300KB
500	150MB	11GB	500KB

Table 7: Test Scenario #3 Configuration -- After External BLOB

The three tests in Table 6 were completed without the usage of an external BLOB provider while the three in Table 7 did use a BLOB provider. Each of the six tests focused on a specific file size in order to determine the effect size would have on performance. Note that the My Site size was kept at approximately 150MB for each test. After the installation and configuration of the external BLOB solution, the database was approximately 11GB (a size reduction of 85%). The VSTS web tests described in Table 3 was used to test both of the configurations.

External BLOB storage was configured using an open source release of a library API and provider that moves data to external NTFS stores. These tools are available for download at Microsoft CodePlex Open Source Community website (CodePlex, 2008). This test was run using the August 2008 release of the provider.

Chapter 4 – Site Collections per Database

4.1 Site Collection Results and Analysis

It is important to note that all of the SharePoint servers and non-SharePoint servers were monitored closely during the tests by VSTS to ensure that other factors such as CPU, available memory, and disk I/O did not blemish the results. In fact, it was discovered during a trial run that the CPU on the web front end SharePoint server (MOSSWeb01) was consistently above 90% throughout the entire test. Since this element could dramatically affect the test results, it prompted the creation of two additional front end web servers: MOSSWeb02 and MOSSWeb03. All of the front end web servers were teamed using Windows Network Load Balancing. The addition of two more web servers into the farm considerably lowered the CPU impact on each server and increased the number of requests per second. The following table illustrates the average RPS and CPU that was recorded when testing various farm configurations:

# Web Servers	Avg. CPU	Avg. RPS
1	93%	44
2	78%	91
3	69%	110

Table 8: Performance Increase with Additional Web Servers

Although the above findings are not one of the test scenarios described in this paper, this information will prove to be very useful when designing the production web farm architecture.

In order to determine an optimum number of My Site site collections per content database, the web tests listed in Table 3 were run against the following four configurations: 1000, 2000, 3000, and 4000 site collections. Each test was completed three times and then the average of the three results was recorded. Data in the site collections was removed each iteration to ensure the results were not tainted by other known software limitations.

The number of site collection did impact performance in terms of requests per second as the data in the following table shows:

# Site Collections	Avg. RPS
1000	106
2000	110
3000	91
4000	84

Table 9: Site Collection Quantity Effect on Performance

The testing that was completed at the 2000 site collection level yielded 110 requests per second. However, between 2000 and 3000 site collections, there was a notable drop of 19 requests per second. This drop continued as the number of site collections were increased to 4000 which yielded only 84 requests per second. The performance yielded at the 4000 site collection level is approximately 24% lower than the 2000 site collection level.

A slight increase in performance was recorded when site collections were increased from 1000 to 2000 per web application. The round of tests was repeated against those two levels to confirm the initial findings. The repeated tests confirmed a 4% increase in performance at the 2000 site collection level.

4.2 Site Collection Summary

While these results do not illustrate a drastic decline in performance, they do illustrate that degradation will occur when site collections increase beyond 2000 in a content database. In fact, the findings showed a 17% drop in performance when the number of site collections were increased from 2000 to 3000 in a single database. As Figure 6 illustrates, this behavior continued again when site collection were increased from 3000 to 4000.

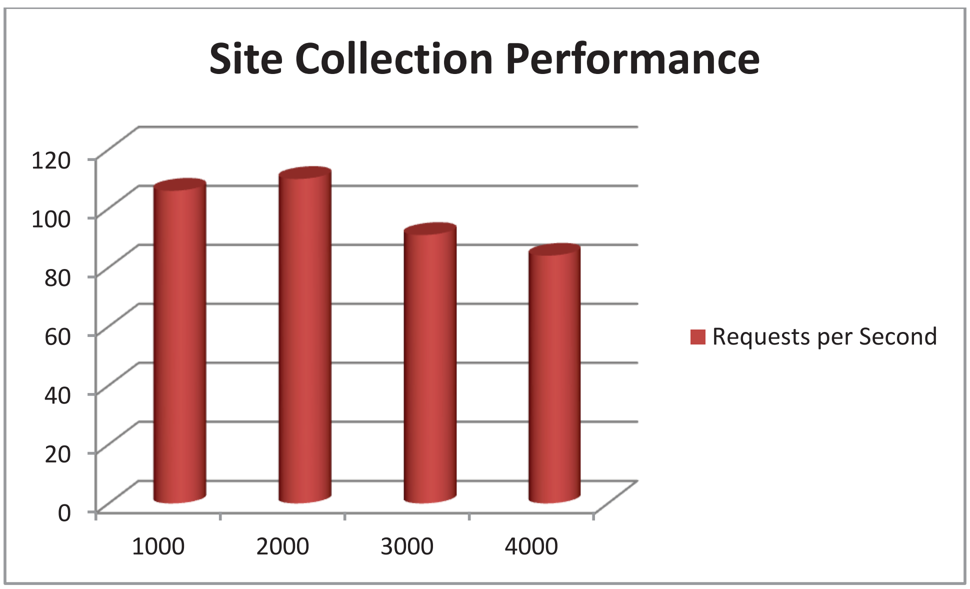

Figure 6: Site Collection Performance

This data has shown that 2000 site collections per content database is the optimum quantity. If this figure alone was used to design the production farm, it would equate to 25 content databases (grand total of 50,000 site collections). However, this does not take into account the size of each site collection which could eliminate any gains in performance. For instance, if the average site collection was 100MB and 2000 sites were created per database, then the content database would be approximately 195GB. This number is well over the Microsoft recommendation of 100GB content databases. For this reason, the results of the second testing scenario, optimum content database size, is extremely important and must be taken into consideration when planning the architecture.

Chapter 5 – Content Database Size

5.1 Content Database Size Results and Analysis

In order to determine an optimum size for content databases, the web tests listed in Table 3 were run against the following four content database sizes: 75GB, 100GB, 125GB, and 150GB. Each test was completed three times and then the average of the three results was recorded. Only 500 My Sites were created for each test since the focus of this test was primarily on size of the content databases. To reach the four database sizes, data was added to each My Site as described in Table 5.

The size of the content databases did not significantly impact performance in terms of requests per second as the data in the following table shows:

Database Size	Avg. RPS
75GB	104
100GB	100
125GB	102
150GB	106

Table 10: Content Database Size Effect on Performance

It is important to note that this test scenario had to be repeated twice because of an error during the first round of testing. During the first round tests, the My Sites were not cleaned after each iteration. This was discovered when excessive RPS performance was recorded while testing the 100GB database. Further investigation revealed SQL deadlock errors in the event logs of the front end web servers and “8sli” warnings in the SharePoint logs which refer to large list queries. It was then discovered that several of the My Site lists had grown past 2000 items and were causing significant list contention. Microsoft highly recommends that document libraries and list views not exceed 2000 items. This discovery required that the tests be repeated again with the My Sites being restored to their original size after each test. The second round of testing on the content databases showed very consistent results and no SQL deadlocks were recorded.

5.2 Content Database Size Summary

As the previous section noted, there was no significant effect on performance noted while testing any of the content database sizes. Figure 7 illustrates the findings of the tests:

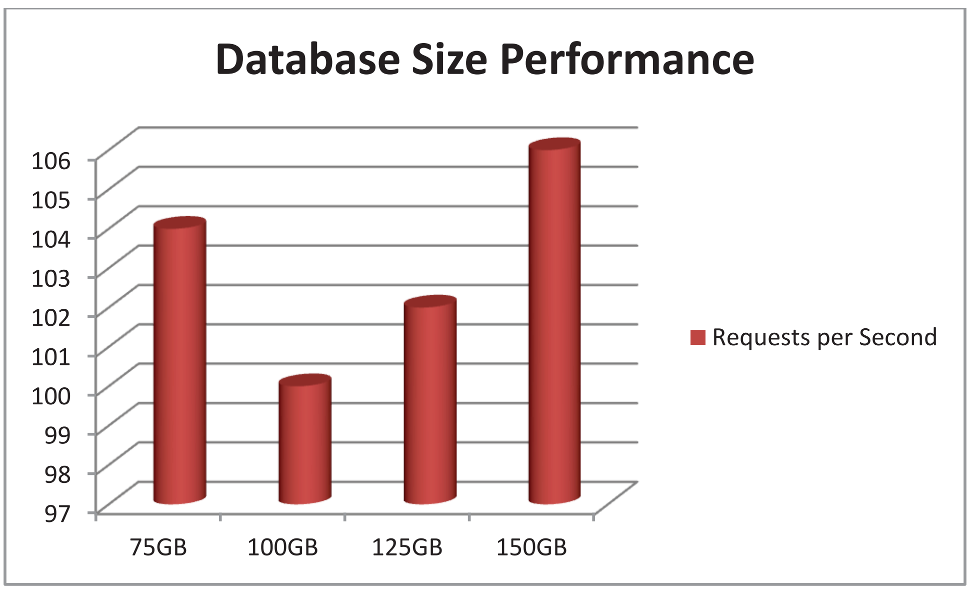

Figure 7: Database Size Performance

As noted in Chapter 2, Microsoft's recommendation regarding 100GB databases is based upon three factors: backup and restore times, contention in large and heavily accessed lists, and the SQL disk subsystem. However, these factors were not relevant in these tests since lists were kept to a minimal size and the storage subsystem was quite robust. The testing mistake noted in Section 5.1 regarding large list views confirms how important view size is in larger databases. The University's past experience with the My Site usage style has shown that large lists and document libraries are rarely an issue. However, if this SharePoint site had a different usage style with multiple business users accessing the same large views, this matter would become much more important. Nevertheless, it is imperative that document libraries and list sizes be monitored on a regular basis to ensure that the size of the views do not cause degradation.

These findings show that there is no performance degradation between 75GB to 150GB databases. This information points out that further testing is needed with even larger content database sizes to see if the trend is consistent. Backup and restores of large databases will not be an issue since the University has implemented an extremely robust backup solution. Having larger database sizes will simplify administration and allow for more room for My Site growth.

Chapter 6 – External BLOB Storage

6.1 External BLOB Storage Results and Analysis

As tables 6 and 7 illustrated, the application of an external BLOB provider reduced the size of the content database by 85% in each test. In each test, the content database was seeded with a specific file size in order to determine the effect file size would have on performance. Three tests were executed with SQL-based BLOB storage using the following three file sizes: 100KB, 300KB, and 500KB. The next three tests were performed using external BLOB storage using the same three file sizes as the previous tests. The following table describes the findings of these six tests:

Test #	BLOB Status	Avg. File Size	Avg. RPS	Avg. WFE CPU (%)
1	SQL-based	100KB	106	49
2	SQL-based	300KB	91	42
3	SQL-based	500KB	74	37
4	External	100KB	99	58
5	External	300KB	92	47
6	External	500KB	77	38

Table 11: BLOB Storage Effect on Performance

Although CPU usage patterns weren't significant in the other test scenarios, they did exhibit interesting behavior in this scenario. The tests exhibited a web front end (WFE) server CPU decrease when the file size was increased. In addition, it is interesting to note that the WFE server CPU values converged as the BLOB file size increased in both SQL-based and external BLOB storage. Potentially, files larger than

500KB may produce even lower WFE server CPU values. Alternately, if CPU utilization is a concern on the WFE servers, it will be advantageous to only externalize BLOBs larger than 500KB.

6.2 External BLOB Storage Summary

As Table 11 illustrates, the use of an external BLOB storage produces a slight increase in performance when the average file size is above 300KB. When the average file size was 300KB, a trivial 1% increase was recorded. However, when the average file size was 500KB, an increase of 9% RPS was recorded. This would be excellent findings if the purpose of this SharePoint farm was document repositories. However, in a My Site usage scenario, the benefit of external BLOB storage may not be very significant. The following chart shows the findings from the six tests:

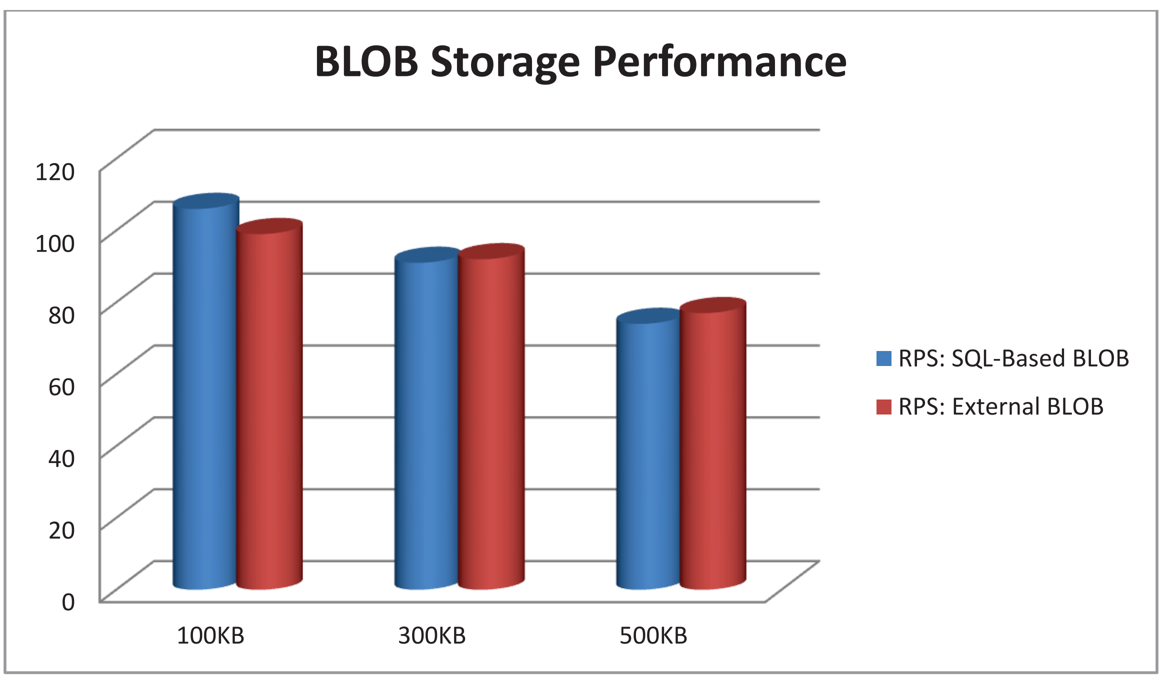

Figure 8: BLOB Storage Performance

The University's existing implementation of SharePoint has an average file size of 255KB. Assuming that file size is similar in the future student portal, external BLOB storage cannot be justified based on performance reasons alone. In addition, at the 300KB file size, the average CPU of the WFE server is higher when an external BLOB is implemented.

Chapter 7 – Conclusions

Out of the three test scenarios performed, only one has the possibility of providing substantial performance gains. It was discovered that performance degrades as the number of site collections in a database exceeds 2000. If a SharePoint farm already contains over 2000 site collections in a single content database, one can use the command-line tool called Stsadm to distribute the site to new content databases (Microsoft TechNet, 2006). Even though the other two test scenarios did not produce similar increases in performance, important data was gleaned from those tests which can assist in the planning of a SharePoint My Site farm.

Before designing and implementing a SharePoint farm using the information gleaned in this study, it is important to note that these results apply to a specific usage style. For example, there were multiple My Site web parts utilized by the web tests that would exhibit unique results. A site that focuses on workflow and document collaboration may glean differing behavior. However, an exception to this is the BLOB storage test scenario which was focused primarily on the performance effects of different file sizes in the SharePoint site.

7.1 What to Glean from the Test Results

The number of site collections per database, the optimum database size, and method of BLOB storage are all items of consideration when designing and implementing a large SharePoint My Site farm. In addition, each of these items complements one another. For instance, the number of site collections one can install in a single database may be limited based upon the results concerning optimum database size in this study.

Testing in the previous chapters has shown the importance of carefully managing the number of site collections placed upon each content database. Keeping the number of site collections less than two thousand per database will help to ensure the best performance.

Also discovered during testing is that a web application's content database size can safely exceed 100GB without a hindrance in performance. The results of this study have shown that databases can safely grow to 150GB and possibly larger without degradation. However, if one chooses to exceed 100GB in content database size, it is important to consider other factors such as backup and restore strategies, possible list contention in large lists and libraries, and the performance of the disk subsystem. Further testing is needed to determine how large a content database can safely be allowed to grow.

Effective management of the storage of BLOBs is an important subject when content database size is a concern. The tests in this study have shown that the external storage of BLOBs causes a slight degradation in performance when the average BLOB size is less than 100KB. However, as the BLOB size increases beyond 300KB, external BLOB storage exhibited a slim performance gain over SQL based BLOB storage. While this information may be integral in a large data warehouse SharePoint farm, it is not as helpful in farms where the average BLOB size is less than 300KB. However, this performance increase that is demonstrated in external based storage of large BLOBs comes at the expense of additional CPU utilization of the web front end servers.

While smaller SharePoint web farms won't benefit greatly from this study's findings, these items are of great importance in web farms that have multiple thousand My Sites and large content databases. Careful implementation of these items can help to ensure that performance will not be a negative influence in the success of a SharePoint site.

7.2 Applying this Data to the University Portal

In order to meet the original goal of excellent performance in the University's SharePoint portal, the information learned from this study must be carefully applied. Since performance was not adversely affected when the database size grew from 75GB to 150GB, this item will not be considered. Even though database size is not a large concern, the implementation of external BLOB storage is very appealing to the University since it exhibited an increase in performance when file sizes were greater than 300KB. It will be necessary to configure the BLOB API provider to only export items greater than 300KB in order to prevent degradation when working with smaller file sizes.

In addition to greater throughput when utilizing external BLOB storage with larger files, a side effect of the smaller database size is faster backups and restores. The databases used in testing were reduced by 85% after exporting the BLOBs to the file system. This smaller database size will also allow for faster database upgrade and patching times. Depending on the external BLOB provider that is chosen, this feature will also allow for greater security with BLOB encryption and archiving of older content to different storage tiers. The impact on performance that those two features might cause is examined in the Appendix.

Since database sizes will be extremely small when using external BLOB storage, it will be tempting for system administrators to allow more than 2000 site collections per database. For instance, if external BLOB storage was not utilized, 2000 site collections at 200MB each would produce a database that is greater than 400GB. However, since external BLOB storage reduces database size by 85%, this content database would only be approximately 60GB. Instead of adding additional site collections to these smaller databases, other items can be considered such as increasing the size quota for each site.

Chapter 8 – Securing SharePoint

As SharePoint becomes a mission-critical system within the University, the topic of security surges in importance. It is imperative that multiple facets of security mechanisms are planned for and implemented before its release to production. Not only do these methods cover the security of the underlying servers and SharePoint itself, but also user security and the subject of site governance. Having multiple layers of security increases the difficulty for the attacker and reduces the surface area of attack. Elements of security that will be addressed in the University's SharePoint portal are as follows: operating system security, network communication, authentication mechanism, Microsoft SQL, service accounts, Microsoft SharePoint, and end-user rights / permissions.

As required with all Windows Server operating systems at the University, each SharePoint server must be scored against the security benchmarks outlined in the Microsoft Windows Server 2008 Security Configuration Benchmark from the Center of Internet Security (Center for Internet Security, 2010). The "Enterprise" security profile should be referenced for scoring purposes. In addition to guidelines in this security benchmark, every University server must have a managed antivirus client installed, configured to query the University's Windows Update Server on a scheduled basis, and have the local firewall enabled according to the University's group policy settings. The intrusion detection system must be configured to monitor the Security event logs on each server for suspicious activity.

To ensure that network communication between the client and the SharePoint servers is secure, SSL must be used for encryption. The IIS websites should be configured to require SSL for all communication. On the SQL server side, all communication between the SQL servers and SharePoint should utilize SSL encryption. In addition, all Microsoft SharePoint and SQL servers will be internal, firewalled segment of the network. The only servers accessible by the users will be the University's existing Microsoft ISA servers which are used to publish all websites.

Although SharePoint websites can use multiple forms of authentication, it is imperative that Kerberos be used for all web applications for performance and security purposes. The most commonly implemented form of authentication in SharePoint is NTLM-based authentication which only encrypts user credentials. In contrast, Kerberos can provide stronger security by providing delegation of various network resources by means of a ticket-based system.

The Microsoft SQL servers should be configured to listen on a non-standard port. TCP port 1433 and UPD port 1434 should both be blocked by the local firewall and the SQL instance should be configured with a different port. This will require the use of SQL Server client aliases on all other servers that require a connection to the SQL Server. Additionally, one should carefully review which accounts hold the server admin role and other roles and rights that might be too excessive. Since SQL BLOBs will be externalized from the SQL database, the topic of securing the BLOB files arises. Access to these BLOB files will be restricted only to the service accounts of the SQL Server and SharePoint web applications. In addition, the University will be researching the possibility of securing the external BLOBs via encryption (see Appendix).

Multiple service accounts should be used in the SharePoint farm as follows: a separate account for each Shared Service Provider and application pool, Central

Administration, and Office Search. This diversification of service accounts will limit the area of attack in the event of a single account being compromised.

SharePoint's reliance on SQL databases produces a unique antivirus need that the client antivirus cannot meet. Each web front end server must have ForeFront for SharePoint installed and configured for real-time scans of all activity on the SharePoint web applications. Scheduled scans will be executed on a weekly basis along with spyware scans. These measures will help to ensure that the University's SharePoint portal does not become a propagation point for viruses and spyware.

Since this SharePoint portal will include custom pages and solutions to access information from other data sources, it is important that all modifications be examined by the University's security team before entering production. Any data connections to external sources need to be limited to only the users who need access. Permissions for pages, files, and data connections should be reviewed on a regular basis. SharePoint's built-in auditing policies for site collections should be enabled and actively monitored.

If the company spends the majority of its effort in designing a rock-solid security policy, but neglects to familiarize end-users with the policy, it will ultimately fail. End-user education is imperative to the success of any security program. Thus, the University will require attendance of multiple University-hosted SharePoint training courses before granting an individual ownership rights on a site. These courses will equip the users with the skills to produce effective and secure SharePoint sites for the students.

Chapter 9 – Lessons Learned

9.1 Challenges

The construction of the physical and virtual test environment was quite time consuming; however, the building and tearing down of the SharePoint content database in between each test iteration was the most intense process. Once the physical structure was established, no further changes were necessary except for a few system reboots and minor configuration changes. The VMware software allowed for the quick creation of virtual servers once a server template had been created. As mentioned earlier, the process of populating the SharePoint content databases was achieved by utilizing the "SharePoint 2007 Test Data Population Tool" (CodePlex, 2007). While this tool proved to be quite invaluable for the building of test site collections, the process was still quite tedious and time consuming.

Early in the testing process, disk space became an issue on the Microsoft SQL cluster due to the unanticipated growth of the tempdb database and the transaction logs. This issue was mitigated with the addition of extra storage and through a proper SQL database maintenance plan.

A significant challenge throughout each test scenario was to prevent the results from being skewed by infringing upon other known software boundaries. For instance, once during the testing process a list view was allowed to grow beyond 2000 items which then caused significant degradation. The test environment had to be cleaned and repopulated again before continuing with the iteration. The testing of differing numbers of site collections per content database is another example of this challenge. It was imperative to keep the content database size less than 100GB since the effect that large SharePoint databases would cause on performance was not yet known.

9.2 Further Research Needed

As with any web site that handles sensitive data, SharePoint presents a large security risk that must be assessed and a proper defense must be planned accordingly. Each of the tests performed in this study were performed with a minimal amount of security configured in order not to skew the results with external factors. However, there are several methods of securing a SharePoint web application that could affect the site performance. Before implementing a SharePoint site into a production environment, it would be advantageous to closely study the performance impact that each of these methods may cause. Please see the appendix for more information regarding this topic.

The first test scenario focused on the effects of differing numbers of site collections in a single database when a social usage style such as My Sites is used. Although beyond the context of this study, it would be interesting to compare the differences in performance between sites dedicated to business purposes and sites that focus on the social usage style.

As load testing began on the second test scenario, it quickly became obvious that performance was not affected when the database grew from 75GB to 150GB. One could potentially exceed much larger than 150GB in the content database assuming factors such as longer backup and restore times, list contention, and the extra tax on the disk subsystem can be mitigated. Additional research in this area is needed to determine at what point it is advantageous to split the content database into two or more additional databases.

Regarding the location of BLOB storage, it would be beneficial to run similar tests with other BLOB API providers. There are several software companies such as AvePoint and Blue Thread Technologies that offer solutions that utilize the external blob storage APIs. In addition, the next version of Microsoft SQL Server (Microsoft SQL Server 2008 R2) will provide an additional option regarding BLOB management called the FILESTREAM feature. This feature places the burden of BLOB management on the SQL server instead of the web front end server. Future research is needed to determine if these alternate methods of BLOB management can provide even greater performance benefits. Nevertheless, the implementation of external BLOBs allows for larger file sizes and provides greater manageability since the database is significantly smaller.

References

Baer, B. (2008). *Planning and Monitoring SQL Server Storage for SharePoint: Performance Recommendations and Best Practices.* Microsoft Online Services.

Bishop, S. (2008). *ForeFront Security for SharePoint Services.* Global Knowledge Training.

Callahan. (2008). *Mastering Windows SharePoint Services 3.0.* John Wiley and Sons.

Chaganti, R. N. (2009). *Designing and Building a Medium Server Farm for MOSS 2007.* Dell Product Group.

Cherny, P. (2009). Inside SharePoint: Creating an External Storage Solution for SharePoint. *TechNet Magazine*, June.

Cherny, P. (2010, January). Using Kerberos for SharePoint Authentication. *TechNet Magazine*.

CodePlex. (2007, September 10). *SharePoint 2007 Test Data Population Tool.* Retrieved January 21, 2010, from CodePlex: http://sptdatapop.codeplex.com/

CodePlex. (2008, August 5). *Microsoft SQL Server Remote Blob Storage Samples.* Retrieved November 30, 2009, from Codeplex: http://sqlrbs.codeplex.com/

Craytor, D., Ward, C., & Peschka, S. (2008, April 23). *Microsoft Office SharePoint Server: 2 Million User Scale Test.* Microsoft Enterprise Engineering Center.

Draper, S. R. (2009). *Social Computing with Microsoft SharePoint 2007: Implementing Applications for SharePoint to Enable Collaboration and Interaction in the Enterprise*. John Wiley and Sons.

Guinn, K. (2009). *Designing and Building a Large Farm for MOSS 2007.* Dell Product Group.

Hewlett-Packard Development Company. (2007). *Microsoft Office SharePoint Server 2007 on HP ProLiant servers – Performance Summary.* Hewlett-Packard.

Houberg, R. (2009). *Scaling SharePoint 2007: Storage Architecture.* KnowledgeLake Inc.

Microsoft Corporation. (2007, November 15). *Description of the SharePoint Server 2007 hotfix package for SharePoint Server 2007*. Retrieved March 2, 2010, from Microsoft Support: http://support.microsoft.com/kb/939077

Microsoft TechNet. (2006, 12 01). *Stsadm Command-Line Tool*. Retrieved 03 09, 2010, from Microsoft Technet: http://technet.microsoft.com/en-us/library/cc261956.aspx

Office IT and Servers User Assistance Team. (2009, April 23). *Plan for software boundaries (Office SharePoint Server).* Retrieved October 6, 2009, from Microsoft Technet: http://technet.microsoft.com/en-us/library/cc262787.aspx#section3

Peschka, S. (2007). *White Paper: Working with large lists in Office SharePoint Server 2007.* Microsoft SharePoint Products and Technologies Group.

Sears, R., Ingen, C., & Gray, J. (2006). *To BLOB or Not To BLOB: Large Object Storage in a Database or a Filesystem?* Redmond, WA: Microsoft Research.

Sterling, D. (2007). *Microsoft Office SharePoint Server 2007: The Complete Reference.* McGraw-Hill / Osborne.

Thumma, J. (2008). *SharePoint's Role as an ECM Solution: A Look behind the Curtain.* Optical Image Technology.

Walters, R., Coles, M., Rae, R., Ferracchiati, F., & Farmer, D. (2008). *Accelerated SQL Server 2008.* Appress.

Appendix

This section focuses on four important methods of securing a SharePoint farm that have the potential of impacting performance: antivirus products designed especially for SharePoint, requiring SSL on the web sites, encryption of BLOBs, and document encryption. Keep in mind that these four approaches of securing SharePoint were not implemented in the tests performed in this paper. It would be advantageous for the University to test the performance impact of each of these methods and include this data in the decision making process.

A very common risk in any SharePoint farm is that of infected documents being shared with other users within the farm. The popularity and ease of use of document libraries in SharePoint creates a potential propagation point for viruses (Bishop, 2008). Activities such saving, downloading and sharing documents in these document libraries can allow for a virus to easily propagate to other users. This is where an antivirus product especially designed for SharePoint is needed. These antivirus programs commonly use multiple engines to scan the uploading, downloading and opening of documents. However, the disadvantage of these products is that a performance impact is inevitable since all documents will be scanned in real-time. Fortunately, several of the antivirus vendors allow you to configure how many detection engines should be scanning which can mitigate the performance impact.

A very basic performance test was performed with a SharePoint antivirus product (Microsoft Forefront for SharePoint) on the test farm before it is deconstructed. A 50MB zip file containing Microsoft Word documents was uploaded to the SharePoint test farm three times while antivirus products were disabled and again when they were enabled.

The results were averaged and then compared. With antivirus software running, the upload took an average of 20 seconds. However, with all antivirus software disabled, the upload took 15 seconds to complete. While this test is by no means exhaustive or conclusive, it illustrates that security measures can impact performance and thus should be considered when developing a security plan.

The University needs to closely consider whether or not a secure connection in the portal web applications is necessary. In order to enforce a secure connection, the website must use HTTPS connections between the server and the end-user's browser. Such connections require a SSL Handshake which can be a costly overhead in terms of performance. For instance, the loading of a single page may incur multiple SSL handshakes depending on the number of external sources. With this in mind, it may be necessary to only secure the authentication pages of the University website.

If External Blob Storage (EBS) is implemented in the University's portal, the encryption of the BLOBs and the corresponding performance impact should be researched. The EBS provider that was used in the tests performed in this paper does not support BLOB encryption. However, there are several EBS vendors for SharePoint that provide 128 or even 256-bit AES encryption.

SharePoint supports a method of document encryption called Active Directory Rights Management Services (AD RMS). The service encrypts the document with a 128-bit AES key and allows management of access through SharePoint and the AD RMS server. Again, there could be degradation in performance since document approval and decryption must be obtained from the AD RMS server every time a document is opened.