



SQL Server RBS Performance with SharePoint Server 2010 and StorSimple Storage Solution

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

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

© 2011 Microsoft Corporation. All rights reserved.

SQL Server RBS Performance with SharePoint Server 2010 and StorSimple Storage Solution

Burzin Patel
StorSimple, Inc.

Peter Scharlock
Microsoft Corporation

Technical Reviewers: John Flores (StorSimple, Inc.), Srinu Acharya, Steve Howard, Shaun Tinline-Jones, Mike Weiner, Kun Cheng, Prem Mehra, Jimmy May, David Koronthaly, Bill Baer

December 2010; Revised April 2011

Applies to: SharePoint Server 2010 and SQL Server 2008 R2

Summary: Microsoft® SharePoint® technology has seen an order of magnitude increase in its usage in recent years. This increase has been a result of users storing a greater number of documents in SharePoint libraries as well as storing larger media documents, both of which result in an increase in storage costs as well as some performance and manageability challenges for SharePoint administrators. Microsoft has addressed these issues by introducing native support for the Remote BLOB Storage (RBS) feature in SharePoint Server 2010. This paper explains the RBS feature as it applies to SharePoint Server 2010 and analyzes its performance impacts on key attributes of a SharePoint farm such as database size, database backup size, transaction response times, and backup/restore time.

Contents

Introduction	4
Remote BLOB Storage	4
Why Use RBS?	5
Test Objectives	6
Test Methodology	6
Workload	7
Server Configuration	8
Hardware Configuration	9
Storage Configuration	9
Software Configuration	9
Test Results and Observations	10
1. Effects of RBS on SQL Server Database Size	10
2. Effects of RBS on Database Backup Size	13
3. Effects of RBS on Backup and Restore Times	15
4. Effects of RBS on Index Rebuild Performance	16
5. Effects of RBS on the SharePoint Transaction Response Times	18
6. Effects of RBS on Crawl Performance	20
7. Effects of RBS on File Upload Performance	21
8. Time Required to Migrate Data	23
Conclusion	24
Additional Resources	24
About StorSimple	25
About Microsoft	25

Introduction

Microsoft SharePoint Server has been growing in popularity almost exponentially in the past few years. This growth has occurred by way of more customers adopting SharePoint Server, as well as more users storing larger documents and data sets within the SharePoint farms. With the recent release of SharePoint Server 2010, the growth in usage is projected to grow even further.

SharePoint Server 2010 offers a more streamlined user interface that provides a richer user experience, making SharePoint Server the repository of choice for all types of data. This coupled with the growth of rich-media content causes the size of the SharePoint farm content to balloon which in turn results in a significant increase in the required physical storage. This growth in size often poses a challenge for SharePoint administrators who must now deal with the added burden of managing more content, larger databases, and larger backups. To address all these problems SharePoint Server 2010 introduces a new feature; Remote BLOB Storage (RBS) that helps address the issues that arise with the growth of SharePoint content.

This paper presents the benefits and operational characteristics of the RBS feature when used with Microsoft SharePoint Server 2010. It also presents the performance characteristics of a SharePoint farm configured to operate with the StorSimple storage solution as explained in the next section. Benefits such as reduction in database size, faster database backups, faster database restores, better response times for larger documents, database maintenance advantages, and less demand on the back-end storage will be discussed along with applicable performance data points. All data points presented in the paper have been generated as part of the performance tests conducted at the StorSimple, Inc. performance labs in Santa Clara in conjunction with the Microsoft SQL Server and SharePoint product teams.

Note: The test results in this whitepaper are specific to the environments described in this paper. Your results may vary.

Remote BLOB Storage

BLOB is an acronym for binary large object, and in the SharePoint application context it refers to the file object that is stored in the database. Remote BLOB Storage (RBS) is a Microsoft® SQL Server® library API set that is incorporated as an add-on feature pack for Microsoft SQL Server 2008 R2. The RBS feature enables applications to externalize the storage of binary large objects (BLOBs) to a location outside the database such as a file share, resulting in a reduction in the amount of SQL Server database storage required. An RBS store is typically a separate volume on the same network as SQL Server. SharePoint Server 2010 leverages the RBS feature to externalize BLOBs stored in the content database. SQL Server and SharePoint Server jointly manage the data integrity between the database records and the RBS external store on a per-database basis.

The SQL Server RBS feature requires a provider to be installed on each SharePoint web front end (WFE) server on which the SharePoint application is configured. The provider is comprised of a set of DLLs that implement methods for the RBS APIs and perform the actual operation of externalizing the BLOBs.

For all tests conducted in this paper, the StorSimple SharePoint Database Optimizer, which includes an RBS provider, was configured on the SharePoint Server 2010 farm. Configuration was done using the StorSimple SharePoint Database Optimizer RBS configuration manager, which is an extension to the Central Administration site, as shown in Figure (i) below.

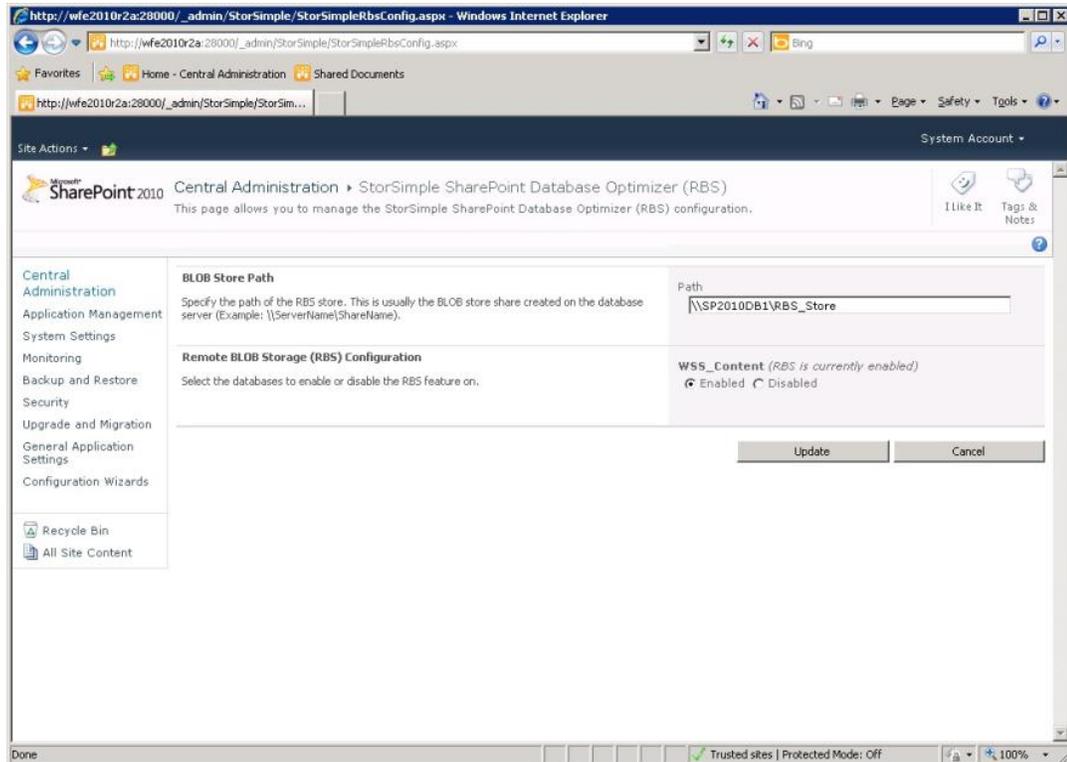


Figure (i) – StorSimple SharePoint Database Optimizer - RBS configuration

Why Use RBS?

SharePoint Server stores all its data in the database. As increasing amounts of content is stored, the size of the database can grow very rapidly. This growth is attributable to new content being uploaded into SharePoint Server, as well as revisions to existing content when SharePoint versioning is enabled; if even a single byte of a SharePoint document is changed, a new copy of the entire BLOB is stored in the database, and the previous one marked as an older version. As already seen by many SharePoint administrators, this results in an exponential growth in the size of the content.

As the size of the database grows, it becomes increasingly difficult to manage the system and ensure optimal performance. Otherwise fundamental tasks such as backup and restore and database defragmentation become increasingly more challenging to execute. This is one of the reasons Microsoft recommends that customers limit the size of their databases to a manageable size as explained in the article: "SharePoint Server 2010 capacity management: Software boundaries and limits" (<http://technet.microsoft.com/en-us/library/cc262787.aspx#ContentDB>). Adhering to this best practice recommendation can mean that the SharePoint administrator is forced to create multiple databases, which becomes costly from a management and maintainability perspective. An increased number of databases results in more backups to manage and monitor, which in turn requires more SharePoint administrators.

Using RBS, your application can store large amounts of unstructured data such as rich-media videos or audio files, leveraging the best of both the relational capabilities of SQL Server and the scalability of a Windows® file system BLOB store. In addition to this main advantage, the RBS feature offers numerous other advantages related to storage costs, maintainability, performance, and flexibility:

- Smaller database sizes, resulting in optimal use of expensive database server resources like processors, memory and disks
- Smaller database backup files
- Faster backup and restore times
- Faster database maintenance operations like defragmentation and index rebuild
- Better overall performance, especially for storing and accessing large objects.

When SharePoint Server is configured to use RBS, the transaction semantics of user operations are fully preserved and there is absolutely no change observed from an end-user perspective. The task of externalizing the BLOBs from the database is done automatically in the back end by SharePoint Server in conjunction with the RBS provider. RBS operates seamlessly when used with SQL Server failover clustering, however it does not work with SQL Server mirroring when the SharePoint content database is mirrored to a database server in another farm.

Test Objectives

The objective of our testing was to characterize the performance of a SharePoint farm configured with RBS using the StorSimple RBS provider, which is a part of the StorSimple SharePoint Database Optimizer, and to then compare that performance to the performance of a SharePoint farm without the RBS feature enabled. We also wanted to measure the effects of RBS on the following:

- SQL Server database data and transaction log file sizes
- Backup file size
- Time taken to back up and restore content database
- Time taken to rebuild content database indexes
- Effects of index rebuild operation on performance of end-user transactions
- SharePoint transaction response times
- SharePoint Server search crawl operation
- File upload performance
- Consistent performance as the scale of the content increases
- Time taken to migrate data in and out of the RBS store

The behavior of SharePoint Server 2010 for varying application workload characteristics or varying thresholds for the size of BLOBs that get externalized is outside the scope of this paper.

Test Methodology

Our goal was to conduct the tests described in the previous section against a workload that represented real-world scenarios as closely as possible. Another goal was to keep the test setup (server configuration, database settings, table schema, and so on) relatively constant across the tests so that we could compare and contrast performance of the different operations.

The tests were broadly divided into 3 categories: (1) upload test, (2) full transaction mix test, and (3) miscellaneous test.

Upload document test: This set of tests measured the performance and effects of RBS on user document upload for varying average file sizes.

Full SharePoint transaction mix test: This set of tests measured the effects of RBS on the performance of the SharePoint farm. The tests included all the commonly executed SharePoint user transactions such as

Browse, Search, Upload document, and Site creation. The main performance metric used was the average response time of the web pages.

Miscellaneous tests: These tests included operations such as database backup and restore, migrating objects in and out of the database and to the RBS store, and SharePoint Server search crawl.

Workload

The various questions we wanted to have answered via our tests forced us to use different workload data sets. Two workloads were used for the tests: (1) the upload files workload mix and (2) Full SharePoint transaction mix.

The upload file workload mix included two sets of files with an average weighted size of approximately 100 KB used to generate the 100 GB database, and 500 KB used to generate the 1 TB content database. The file size distribution for the 100 KB data set is shown in Figure (ii).

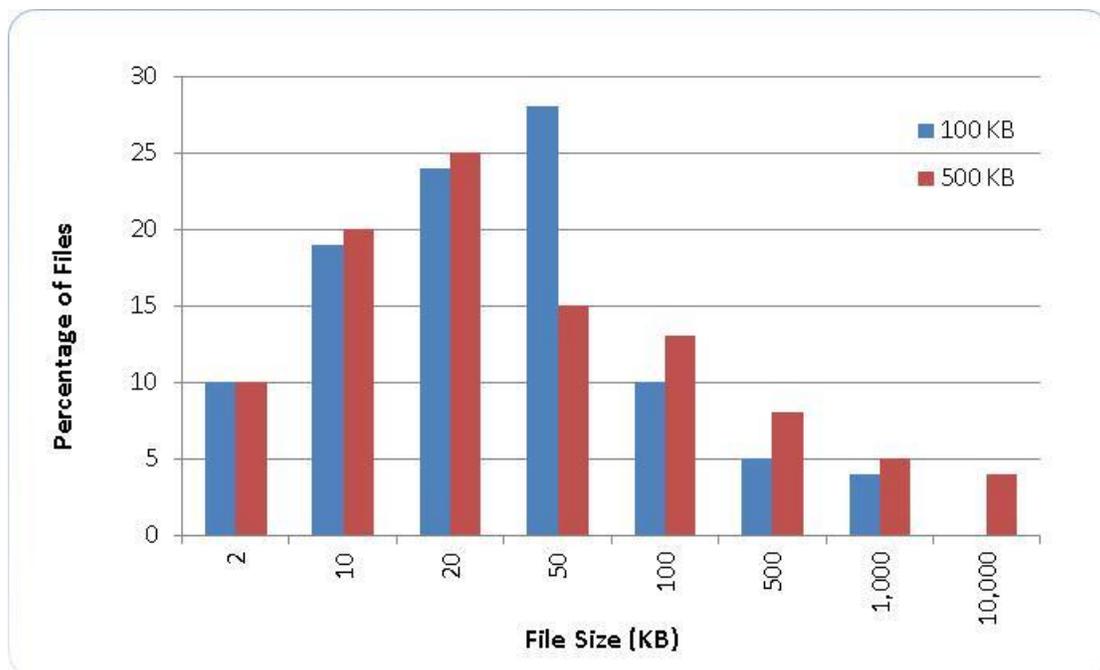


Figure (ii) – Workload file size distribution

The upload files workload mix was used primarily to measure the document upload characteristics with and without RBS.

The Full SharePoint transaction mix was used to represent a mix of typical SharePoint transactions an end-user would execute on a daily basis. Microsoft Visual Studio® Team System 2008 Team Suite was used to generate the workload using a modified version of the original Microsoft Office SharePoint Server 2007 performance toolkit shared on Codeplex. The following transactions were used for each of the tests.

Test Name	Description	Percentage
Page Workflow	Go through a page workflow: Check out, Approve and Check In	1%
Create Page	Create a new page	6%
Site Manager	Open the Site Manager view	1%
Create Publishing Site	Create a new site with the Publishing template	1%
Create Team Site	Create a new site collection using the team site template under the sites directory	1%
Homepage	Browse to the portal homepage	25%
Large Page	Browse to a variety of pages across the portal	10%
My Site Public	Browse to the My Site public page	16%
My Site Edit Profile	Edit the personal profile	7%
Search Query	Perform a Search query and view results in the Search Center page	15%
Upload Document	Upload a document (average 90 KB)	5%
Download Document	Download a document (average 90 KB)	12%
Total:		100%

Table (i) – Full SharePoint transaction mix.

Server Configuration

The SharePoint farm was configured with six Web Front End (WFE) servers, one application server that was configured to run the search crawler and one database server as shown in Figure (iii).

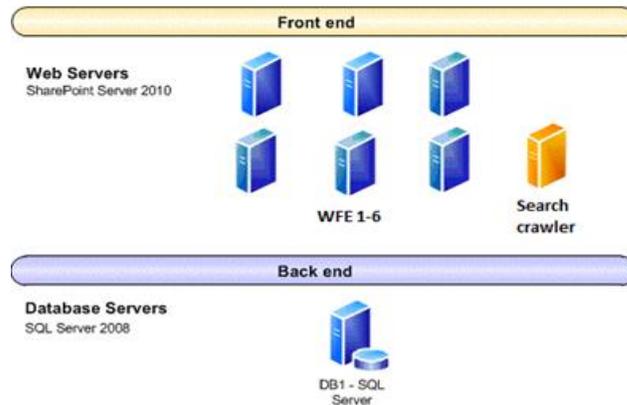


Figure (iii) - SharePoint farm topology

The WFE and application servers were configured to run on a virtual machine (VM) while the database server was run on a dedicated physical server (non-virtualized).

In addition, six VM-based load driver servers (not shown above) were used to generate the workload for the upload file transaction mix and the full SharePoint transaction mix.

Hardware Configuration

Computer role	Hardware
WFES	2 processor Intel Xeon E5504 2 GHz processors (virtualized) 8 GB RAM
Application server	2 processor Intel Xeon 2 GHz processors (virtualized) 8 GB RAM
Database server	2 quad-core Intel Xeon 2 GHz processors (not virtualized) 16 GB RAM (12 GB assigned to SQL Server)

Table (ii) – Hardware configuration

Storage Configuration

All the storage used in the benchmark test was configured on the StorSimple 1010 Storage Appliance¹. The SQL Server system databases, SharePoint databases, and BLOB store were located on separate volumes as shown in Table (iii) below.

Volume	Drive
SQL System databases	C:\
tempdb data and log files	H:\
Content database data file	P:\
Content database log file	Q:\
Search database data file	S:\
Search database log file	Q:\
BLOB store	X:\
Backups	O:\

Table (iii) – Storage configuration

Software Configuration

The software versions and settings used for the different servers were as shown in Table (iv) below.

¹ StorSimple 1010 is an application optimized storage appliance targeted to applications like Microsoft SharePoint and Microsoft Exchange. For additional information, go to <http://www.storsimple.com>.

Computer Role	Software	Additional Changes
WFEs and Application servers	Windows Server® 2008 R2 Enterprise x64 Microsoft SharePoint Server 2010 RBS.msi was installed from the SQL Server 2008 R2 Feature Pack.	All the latest Windows Server patches were applied.
Database server	Windows Server 2008 R2 Enterprise x64 SQL Server 2008 R2 Enterprise x64	Latest Windows Server patches were applied. The following changes were made to the database server settings: <ul style="list-style-type: none"> - 'Max server memory' = 12 GB - 4 <i>tempdb</i> data files were created and moved to their own volume.

Table (iv) – Software configuration

Test Results and Observations

This section summarizes the results of the tests conducted to measure the effects of using RBS to externalize the BLOB contents on the various attributes of a SharePoint Server 2010 deployment and help answer the questions listed in Table (v) below.

Test Description	
1	Effects of RBS on database size
2	Effects of RBS on database backup size
3	Effects of RBS on backup and restore time
4	Effects of RBS on index rebuild performance
5	Effects of RBS on the SharePoint transaction response times
6	Effects of RBS on crawl operation
7	Effects of RBS on file upload for varying file sizes
8	Time required to migrate data in and out of the RBS store

Table (v) – Test scenarios

1. Effects of RBS on SQL Server Database Size

As explained in the RBS section, the majority of the data in the SQL Server database corresponds to SharePoint BLOB data. In most SharePoint customer deployments, especially those using SharePoint for collaboration and records management, the BLOB data accounts for more than 95% of the database size. Depending on the size of the database, this content could easily translate to hundreds of gigabytes of database data. While this is by design, it poses many challenges and is often a limiting factor on the usage

of SharePoint Server, the scalability of the solution and the use of certain beneficial features like Recycle Bins.

In this tests whose results we summarize in this section, we measured the size of the database, data files, and transaction log file for 100 GB SharePoint content databases comprised of 100,000 objects, and a 1 TB SharePoint content database comprised of 2 million objects with and without the RBS feature. The file sizes for each of these databases are shown in Table (vi).

	Size (GB)		Reduction
	Without RBS	With RBS	
Database size (100 GB)	217.2	7.0	96.8%
Database data file size (100 GB)	106.9	3.2	97.0%
Database transaction log file size (100 GB)	111.6	3.8	96.6%
Size of RBS externalized data	--	96.2	--
Database size (1 TB)	2,292	26	98.9%
Database data file size (1 TB)	1,120	6.5	99.4%
Database transaction log file size (1 TB)	1,173	20	98.3%
Size of RBS externalized data	--	1,115	--

Table (vi) – Database and file sizes

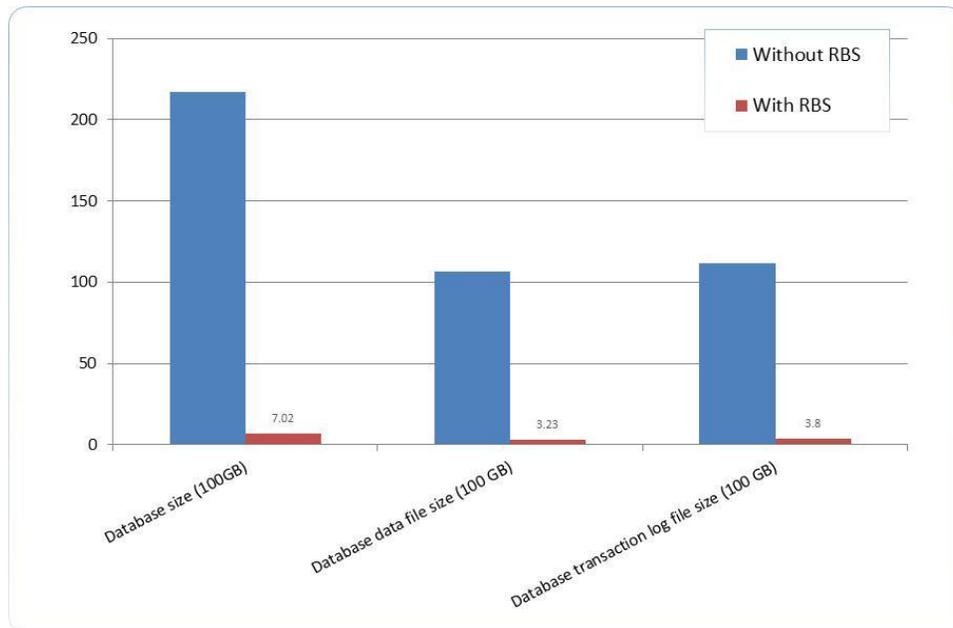


Figure (iv) – Database and file sizes

As depicted in Figure (iv) above, without RBS the overall size of the databases after 100 GB and 1 TB of SharePoint content was uploaded into it were 217.2 GB and 2.29 TB respectively. For the database with 100 GB of SharePoint content, 106.9 GB corresponded to actual database data while the other 111.6 GB corresponded to the database transaction log. Similarly for the database with 1 TB of SharePoint content, 1.12 TB corresponded to the database and 1.2 TB corresponded to the database transaction log. For a database with RBS enabled, the size of the 100 GB content database was 96.8% smaller, while the size of the 1 TB content database was 98.9% smaller. The sizes of the data and transaction log files were correspondingly smaller.

While the additional space required for storing BLOBs within the database is often apparent and well understood, a less known and even less understood negative is the challenge related to the SQL Server transaction log file growth. The reason for this growth is that SQL Server is a transactionally consistent database, offering full Atomicity, Consistency, Isolation, Durability (ACID) properties. This suggests that each transaction is guaranteed to either succeed or fail; there is no middle state. SQL Server implements the ACID properties by fully logging each operation in the database transaction log using write-through disk access before the operation is committed. ACID properties apply to all SQL Server data and data types, BLOBs included. No mechanism exists through which this can be disabled or short-circuited in any way. As can be expected, when the SharePoint BLOBs are stored within the SQL Server database the BLOBs get written twice, first to the transaction log and then to the database file as can be observed by the size of the database (2.29 TB) used for storing 1 TB of user content. This log file gets truncated when the database backup is taken and the 'truncate log' option is selected.

When RBS is used to externalize the BLOB content, the BLOB data is written to the BLOB store before the SharePoint operation is committed. Therefore the ACID properties of the operation are indirectly realized without incurring the double transaction-log-write overheads. The amount of reduction in database data and transaction log files depends on the size of data and the frequency with which you truncate the transaction log during a backup.

The externalized BLOB content is stored on a centralized file share that is accessible by all the SharePoint WFEs and application servers. This file share volume can be located on the database server or some other server. Figure (v) depicts the properties of the file share used in the benchmark tests.

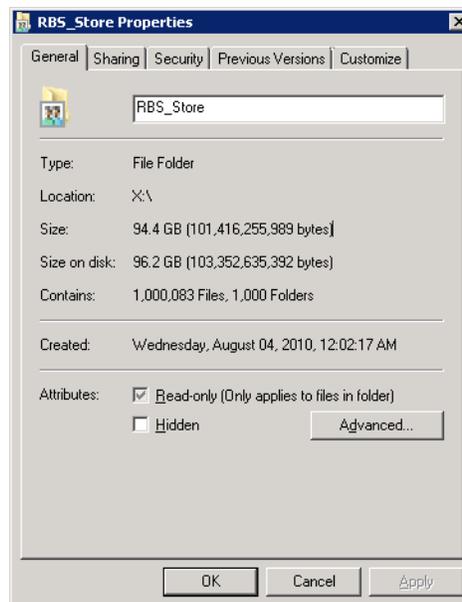


Figure (v) – RBS file share volume size

Note: RBS reduces the size of the database by moving the BLOB data to external storage therefore it is important to keep in mind that the overall disk space consumed by BLOB data is not reduced. Of course, storage vendors may assist with this using proprietary technologies such as de-duplication which could help reduce the disk space. The BLOBs are not automatically deleted from RBS store when the corresponding content is deleted from SharePoint; a separate garbage collection cycle using the built-in RBS Maintainer job is required to purge the orphaned BLOBs.

2. Effects of RBS on Database Backup Size

In the tests whose results we summarize in this section, we measured the effects of RBS on the database backup size for a 100 GB SharePoint content database comprised of 100,000 objects, and a 1 TB SharePoint content database comprised of 2 million objects. The tests and analysis did not include the RBS store. That is, techniques and durations related to the backing up and restoring of BLOB data residing on the RBS storage is not in-scope of this whitepaper.

The following Transact-SQL command was used to perform the backup.

```
BACKUP DATABASE [WSS_Content] TO DISK = N'O:\WSS_Content' WITH NOFORMAT, INIT, NAME = N'WSS_Content-Full Database Backup', SKIP, NOREWIND, NOUNLOAD;
```

Tests were also conducted to measure the effects of the SQL Server backup compression feature² to determine its effect on the size of the backup with and without RBS. The results of our tests are summarized in Table (vii) below.

	Size (GB)		Reduction
	Without RBS	With RBS	
Database data file size (100 GB)	106.9	3.2	97.0%
SQL Server backup size (100 GB)	107.0	3.3	96.9%
SQL Server backup size with compression (100 GB)	71.5	0.7	99.1%
BLOB store size (100 GB)	0	96.2	--
Database data file size (1 TB)	1120	6.5	99.4%
SQL Server backup size (1 TB)	1,119.0	6.6	99.4%
SQL Server backup size with compression (1 TB)	1,046.0	1.2	99.9%
BLOB store size (1 TB)	0	1115	--

Table (vii) – Database and backup sizes

² The ability to compress the database backups requires the use of SQL Server Enterprise. This feature is not available in SQL Server Standard or SQL Server Express.

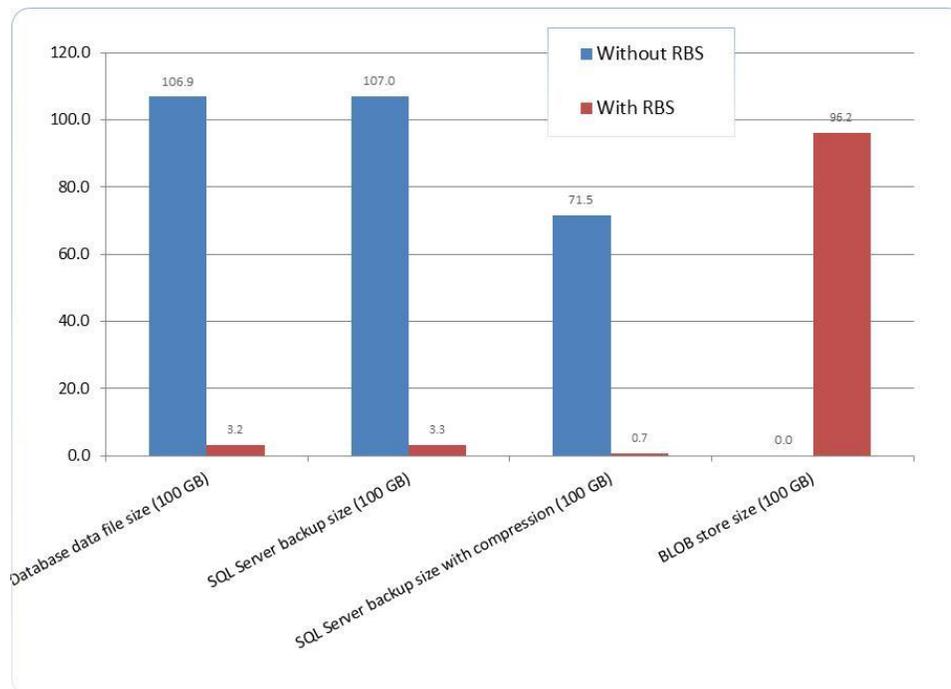


Figure (vi) – Database and backup sizes

As can be seen from the graph and table above, the size of the database backup corresponding to 100-GB content was 96.9% smaller (107 GB vs. 3.3 GB) while the database backup size of the 1-TB content was 99.4% smaller (1,119 GB vs. 6.6 GB) when RBS was enabled. For 100-GB content the size of the BLOBs that had been externalized from the database was 96.2 GB and that for the 1-TB database was 1115 GB.

When the SQL Server backup compression feature was enabled on the database, the size of the backups further reduced to 71.5 GB and 1,046 GB respectively without RBS, and 0.7 GB and 1.2 GB with RBS. Note that backup compression was effective in reducing space in the case where RBS was not used because SharePoint Server stores the BLOB data in-row with the other data (metadata). If the BLOBs were selected to be stored out of row, backup compression would not have had any effect since BLOBs stored out of row are not compressed. While this is an advantage in this case, it comes at the expense of a larger working set and degraded cache efficiency that ultimately result in reduced performance.

Since SharePoint BLOBs are immutable, meaning that once created they never change, the BLOB content can be backed up at any time following the completion of the SQL Server database backup. This provides the flexibility of doing a quick point-in-time transactionally consistent backup of the SQL Server database and then backing up the BLOB store volume at a later time. The SQL Server backup and the backup of the RBS content store together comprise a complete backup set of the SharePoint content. Once complete, the backup set can be used to restore the SharePoint database as of the time of the start of the SQL Server backup. **Note:** when planning a Backup & Restore strategy that involves RBS data storage, plan for the RBS recovery time. Until RBS is restored, SharePoint documents will be unavailable.

3. Effects of RBS on Backup and Restore Times

In the tests whose results we summarize in this section, we measured the effects of RBS on the time taken to back up and restore a database. Similar to the previous section, we used a 100 GB SharePoint content database comprised of 100,000 objects. A series of tests were conducted to measure the time required to back up and restore the databases with and without RBS enabled. The results for the tests against the 100-GB database are summarized in Table (viii) below.

Operation	Without RBS	With RBS	Reduction
Database data file size	106.9	3.2	97.0%
Time to back up database	2,490 seconds	38 seconds	98.5%
Time to restore database	1,290 seconds	28 seconds	97.8%
Time to back up database with backup compression enabled	3,160 seconds	37 seconds	98.8%
Time to restore database from compressed backup	1,330 seconds	28 seconds	97.9%
Time to backup BLOB store (snapshot)	--	14 seconds	--
Time to restore BLOB store (snapshot)	--	28 seconds	--
Time to backup BLOB store (copy command)	--	2,578 seconds	--
Time to restore BLOB store (copy command)	--	2,880 seconds	--

Table (viii) – Backup and restore times for 100 GB database

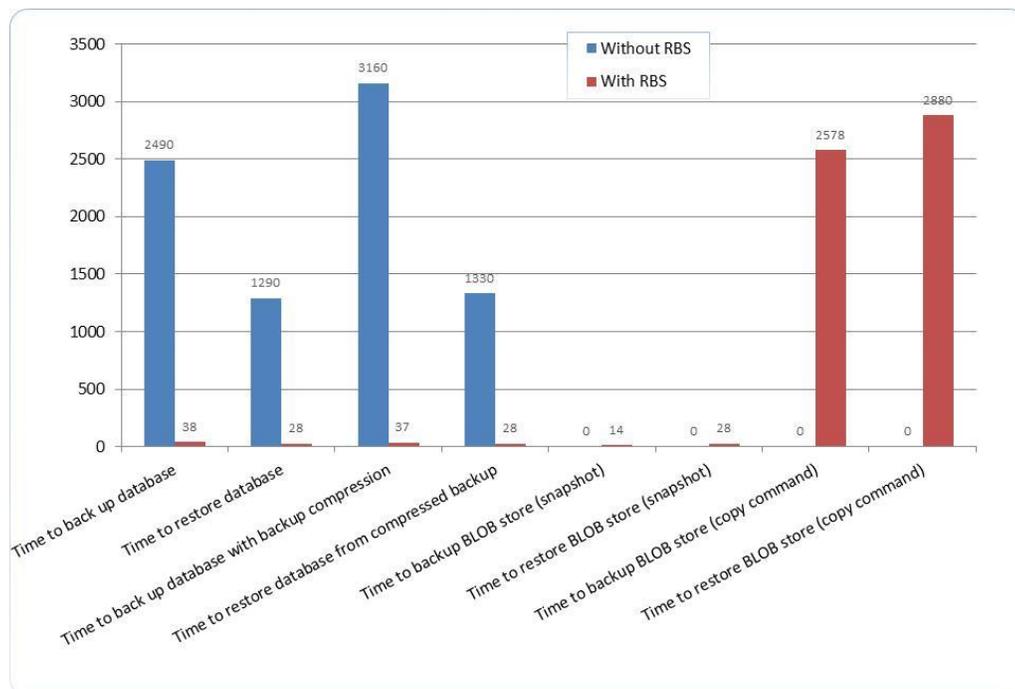


Figure (vii) – Backup and restore times for 100-GB dataset

For database backups and restores, the amount of time taken is linearly proportional to the size of the database. Given that the size of the database was significantly smaller with RBS enabled, the amount of time correspondingly lessened, as shown in Figure (vii). With RBS enabled the amount of time taken to back up the database was 98.5% less (2,490 seconds vs. 38 seconds), while the amount of time taken to restore the database was 97.7% less (1,284 seconds vs. 28 seconds). Similarly, the amount of time taken to back up the database using SQL Server backup compression was 98.8% less, while the amount of time taken to restore a backup-compressed database was 97.9% less. Backing up the database with backup compression took 27% more time and considerably more SQL Server server resources due to the added processing required to compress the data. The commands used to back up and restore the databases were as follows:

```
BACKUP DATABASE [WSS_Content] TO DISK = N'O:\WSS_Content' WITH NOFORMAT, INIT, NAME = N'WSS_Content-Full Database Backup', SKIP, NOREWIND, NOUNLOAD;
```

```
BACKUP DATABASE [WSS_Content] TO DISK = N'O:\WSS_Content' WITH COMPRESSION, NOFORMAT, INIT, NAME = N'WSS_Content-Full Database Backup', SKIP, NOREWIND, NOUNLOAD;
```

```
RESTORE DATABASE [WSS_Content] FROM DISK = N'O:\WSS_Content' WITH FILE = 1, MOVE N'WSS_Content' TO N'J:\ContentDB_Data\WSS_Content.mdf', MOVE N'WSS_Content_log' TO N'S:\ContentDB_Log\WSS_Content_log.LDF', NOUNLOAD, REPLACE;
```

When RBS is used the RBS store needs to be backed up separately. This backup can be performed asynchronously and in parallel with the database backup, the only requirement being that the backup of the RBS store needs to be initiated after the database backup has been started. Various mechanisms can be used to back up the RBS store. In our tests we measured the time taken to back up the store using a disk snapshot mechanism as well as a simple sequential directory copy. For the 100 GB of content the time taken to back up the RBS store using a disk snapshot was 14 seconds, while the time taken with the copy command was 2,578 seconds.

Note - When using the FILESTREAM provider, SharePoint 2010 will automatically backup or restore both the BLOB data and the metadata.

When restoring a database which has RBS enabled, the BLOB store needs to be restored as well. The SharePoint farm is considered fully restored and accessible only after the BLOB store has been restored. For the 100 GB of content the time taken to restore the RBS store was 28 seconds when a disk snapshot mechanism was used and 2,880 seconds when the copy command was used. It is worth mentioning that the RBS store only needs to be restored in the event that it has become corrupted or got into an inconsistent state.

4. Effects of RBS on Index Rebuild Performance

One of the characteristics of SharePoint Server is the frequent and extensive fragmentation of the back-end SQL Server database tables that store the BLOB content. This fragmentation is in many ways by-design and attributable to how the SharePoint application is architected and the access pattern of the back-end SQL Server database. When the database gets fragmented database pages that are logically contiguous do not appear physically contiguous in the data file. In addition, data pages are often not used to their full capacity causing a larger number of such low-density pages to be required to store the data. Both these factors cause the working set to be larger than necessary, which may result in degraded performance.

The good news is that SharePoint 2010 automatically reduces fragmentation by running three SharePoint Health Analyzer rules. These rules regularly check the fragmentation of the indices and run the

proc_DefragmentIndices stored procedure to defragment the indices automatically. However, one should bear in mind that this process is resource-intensive and the entire SharePoint farm becomes unavailable during the index rebuild process. The three rules are:

- Databases used by SharePoint have fragmented indices
- One or more Search crawl databases have fragmented indices
- One or more Search property databases have fragmented indices

Externalizing the BLOBs via RBS significantly helps alleviate this problem because the smaller database requires less time to rebuild the indexes.

To measure the effects of rebuilding indexes we executed a set of tests in which we forced an index rebuild operation for all the tables in the SharePoint content database. While this may not be representative of the real world deployment where indexes are rebuilt on an as needed basis, the approach was chosen to make the test deterministic and repeatable. As part of these tests we measured the time taken to rebuild indexes for 100-GB and 1-TB content databases with and without RBS enabled. We also measure the impact of an index rebuild operation on the availability and performance of the SharePoint farm.

	Without RBS	With RBS	Reduction
Index rebuild time for all tables (100 GB)	120 sec	4 sec	96.7%
Index rebuild time for all tables (1 TB)	600 sec	146 sec	75.7%

Table (x) – Database fragmentation

As can be seen in Table (x) above, the amount of time taken to rebuild indexes is 96.7% less (120 seconds vs. 4 seconds) for the 100-GB database and 75.7% (600 seconds vs. 146 seconds) for the 1-TB database when RBS is enabled. Since the SharePoint web application is unavailable for the majority of time when the indexes are being rebuilt, the reduced time directly affects the availability of the SharePoint application and allows more frequent execution of the index rebuild operation, thereby maintaining more consistent performance.

Multiple tests were performed to measure the effects of index rebuild test on a 100-GB database without RBS. Figure (viii) below plots the results for one such test where an upload document workload was simulated and the index rebuild operation was executed during the steady state.

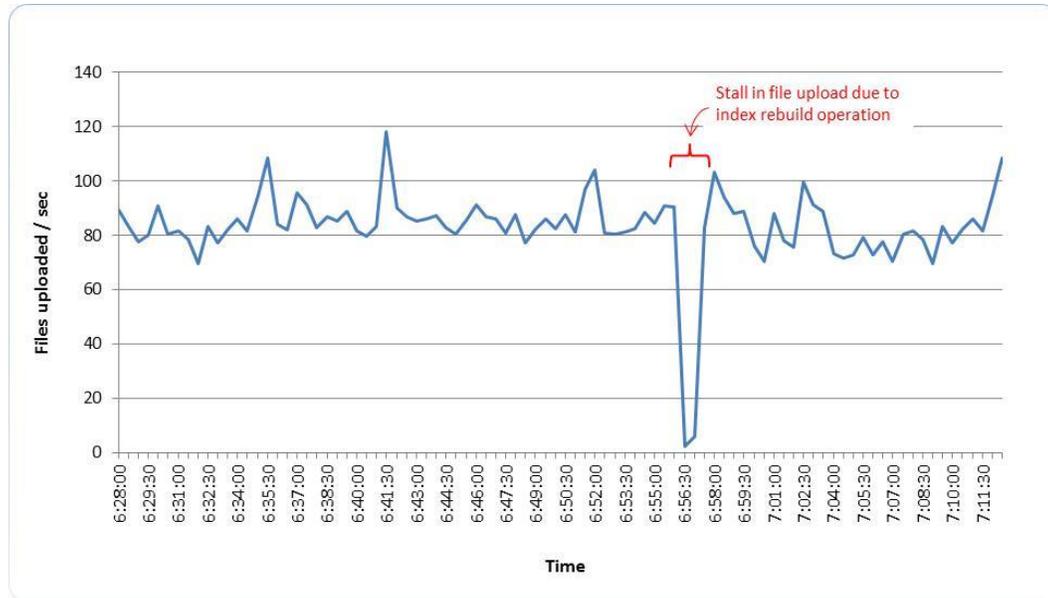


Figure (viii): Effect of index rebuild operation on performance

As can be seen during normal operation (6:28 AM to 6:56 AM) the expected file upload rate averaged about 85 files per second. At 6:56 AM an index rebuild operation was executed, which lasted for 120 seconds. During this time the file upload rate dropped nearly to zero, as can be seen in the graph. This suggests that the set of user operations executed during this period either get stalled for up to 120 seconds or worse still time out and result in an error message being displayed to the end-user.

Given that the index rebuild operation when RBS is enabled on the database is only 4 seconds, the window was so small that the overall impact was not noticeable. In fact, the dip in the performance was so minor it was hard to represent on the graph and has therefore intentionally not been plotted on the graph. While this test was conducted using a file upload as a workload, the effect on availability of the SharePoint farm is the same on all types of transactions.

5. Effects of RBS on the SharePoint Transaction Response Times

As explained in the sections above, enabling the RBS feature results in smaller SharePoint content databases that in turn require fewer resources on the SQL Server database server to execute the queries. The saved resources are freed up to process the existing queries faster and to service more queries.

In the test whose results we summarize in this section, we measured the effects of enabling RBS on the transaction response times. For this test, we used the full SharePoint transaction mix workload, which is explained in the Test Methodology section. This workload was run on 6 load drivers that simulated a user load of 100 users executing the SharePoint transaction on average every 15 seconds. Each test was ramped up for 5 minutes and then run continuously for a duration of 2 hours. The average response times were measured across the entire 2-hour steady-state performance of the test. These high-level results are shown in Table (xi) below.

Metric	Without RBS	With RBS	Reduction
Max User Load	100	100	0.0%
Requests/Sec	84	84.3	-0.4%
Requests Failed	0	0	0.0%
Avg. Response Time	28 msec	21 msec	25.0%
Tests/Sec	6.4	6.42	-0.3%
Avg. Page Time	210 msec	160 msec	23.8%

Table (xi) – Transaction response time test metrics

The average response time across all the transactions was 25% less (28 milliseconds vs. 21 milliseconds) when RBS was enabled on the content database. This suggests that when RBS was enabled on average the end-user response times of SharePoint transactions was 25% faster across the variety of transactions. Given that productivity and satisfaction of SharePoint users are often dependent on the SharePoint transaction response times, a 25% reduction would result in higher levels of productivity and satisfaction.

Table (xii) below further breaks down the response times for each of the fourteen end-user transactions.

Transaction	Transaction %	Avg Transaction Time (sec)		Reduction
		Without RBS	With RBS	
MySite Public	16.0%	0.14	0.08	42.9%
Homepage	25.0%	0.43	0.22	48.8%
Page Workflow	1.1%	109.00	109.00	0.0%
Create Page	6.0%	15.72	15.67	0.3%
Create Publishing Site	1.0%	13.00	12.70	2.3%
Create Team Site	1.0%	17.90	18.30	-2.2%
Download Document	12.2%	4.03	4.03	0.0%
MySite Edit Profile	6.9%	29.84	29.90	-0.2%
Large Page	10.1%	0.12	0.09	25.0%
Search Query	14.8%	60.00	60.10	-0.2%
Site Manager	1.0%	0.45	0.31	31.1%
Upload Documents	4.9%	30.20	30.50	-1.0%
MySite Public	16.0%	0.14	0.08	42.9%

Table (xii) – Transaction response times

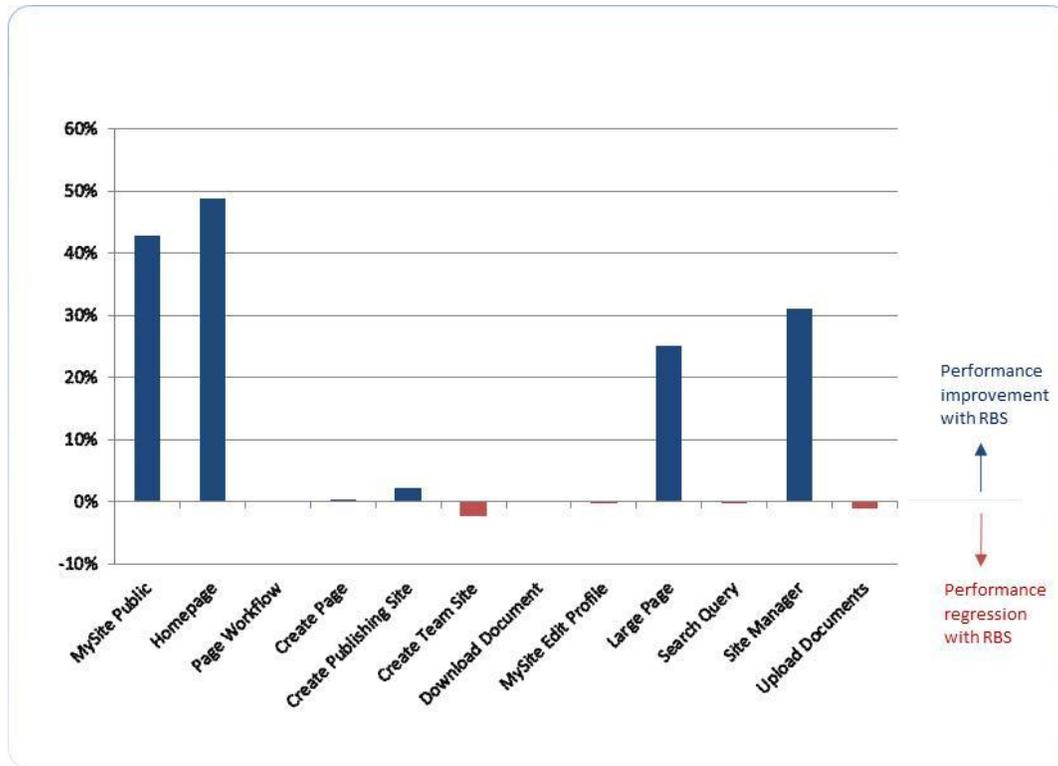


Figure (ix) – Transaction response times

As shown above the average response times for 10 of the 14 transactions were equal or better when RBS was enabled, with four of the transactions having a close to 50% improvement. For the four transactions that regressed in performance, the regression was less than 2.2%, which would most likely not be perceived by an actual user. In general when RBS is enabled one can expect to see an improvement in performance for large files, especially for IO bound systems as with IO is redirected away from the SQL Server database. For smaller files there can be a relative degradation in performance as the WFE must issue two requests over the wire as opposed to one. However the expectation is that the relative increase would not be noticeable even if the percent difference is large since the file access times are negligible to begin with.

6. Effects of RBS on Crawl Performance

Search is an integral part of most SharePoint deployments and one of the more resource-intensive SharePoint services. Many enterprise deployments have a large percentage of users who access data by navigating from the search portal as opposed to accessing the site or document directly. This behavior results in heavy use of search and it is no surprise that many customers claim that search has become their number-one resource consumer, or that it is often a bottleneck.

There are two components to SharePoint Server search, search crawl and search query. The search crawl process involves crawlers crawling the search corpus and building (or updating) the search index. The SharePoint search index consists of two parts, a search database and a flat search index file. The search queries in turn utilize the search database and index to return results for users' search queries.

In the tests results that we summarize in this section, we measured the time taken to crawl the search corpus via a single application server using the out-of-the-box search settings. The results for the time taken with and without RBS are summarized in Table (xiii) below. The search query results were summarized in the previous section and are therefore not repeated here.

Operation	No. of Objects	Without RBS	With RBS	Reduction
Search Full Crawl	503,206	150 minutes	146 minutes	2.7%

Table (xiii) – Search crawl times

Type	Name	Status	Current crawl duration	Last crawl duration	Last crawl completed	Next Full Crawl	Next Incremental Crawl	Priority		
	Local SharePoint sites	Idle		02:26:17	7/12/2010 7:55 PM	None	None	Normal		
Content Source						Successes	Warnings	Errors	Top Level Errors	Deletes
Local SharePoint sites						503,206	4	0	0	0

Figure (x) – Summary of Search full crawl

As can be seen in the results above, enabling RBS on the search corpus databases has a very negligible effect on performance, boosting performance by just 2.7%. This is in line with our expectations because the processing done in both cases is approximately the same.

7. Effects of RBS on File Upload Performance

The amount of time taken to upload large files into SharePoint Server is often an inhibitor for users uploading large amounts of content. The most common complaint is that copying a file to a Windows file share is often order of magnitudes faster than uploading the same file into SharePoint Server. One of the reasons for this is that by default, all of the contents of the file are stored in the SQL Server database, and that has an overhead associated with it. In addition, given that the SQL Server database operates in a transactionally consistent model, it is forced to log the entire BLOB in the SQL Server transaction log, in addition to storing the actual copy of it in the database, thereby resulting in twice the IO load on the system. RBS significantly boosts the performance of file uploads for large files because it externalizes the BLOB directly from the WFE and therefore the IO load on the SQL Server system is minimized.

In the tests results that we summarize in this section, we modeled a SharePoint digital asset management deployment and measured the performance of uploading large files ranging from 1 MB to 1.99 GB with and without RBS enabled. The results for the time taken to upload the files with and without RBS are shown in Table (xiv) below.

File size	Time taken to upload file (seconds)		Reduction
	Without RBS	With RBS	
1 MB	1.2	1.0	16.7%
100 MB	12.2	9.7	20.5%
500 MB	55	28.8	47.6%
1 GB	69.4	48	30.8%
1.5 GB	138	71	48.6%
1.99 GB	178	87	51.1%

Table (xiv) – File upload times

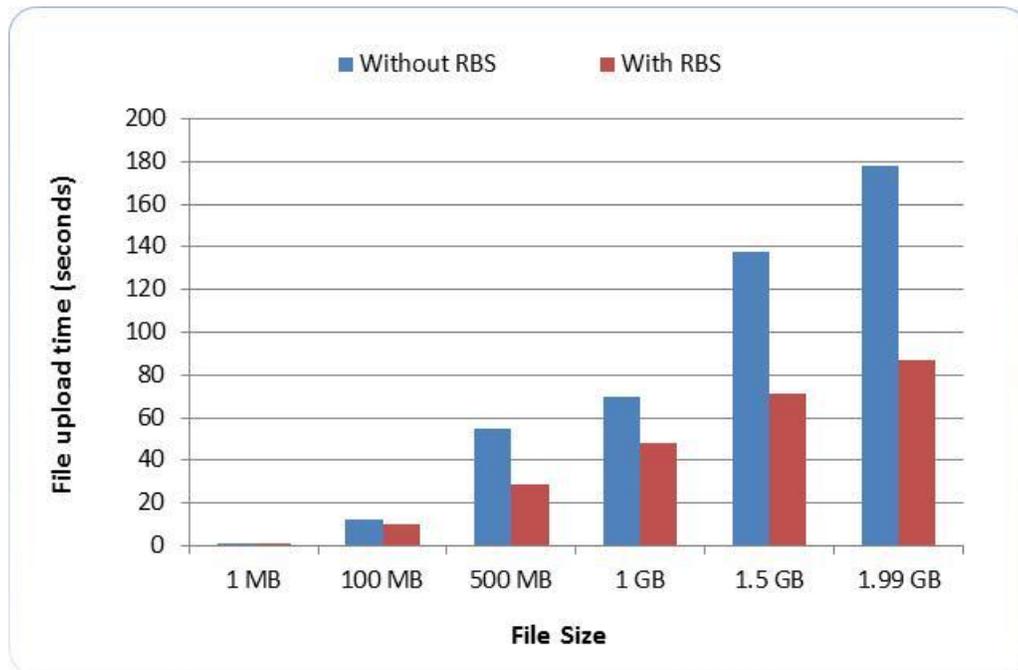


Figure (xi) – File upload times.

As the table and graph show, the time taken to upload a file with RBS is 15% - 50% faster than it was when RBS was not enabled. In absolute terms, this translates to a 1.99 GB file taking 87 seconds versus 178 seconds to upload, which is significant for record center users uploading files considering that they often simply wait in front of their web browsers for the operation to complete before continuing their activity. With hundreds of users in an organization, each doing tens of such operations, the time savings and benefits quickly add up and are particularly noticeable when there is a resource bottleneck on the server.

Similar advantages apply for file download operations as well, though with file downloads the SQL Server system and the SharePoint WFE buffer the file data, thereby causing fewer resources to be expended on the back-end storage.

8. Time Required to Migrate Data

Once RBS is enabled on a database, all files uploaded or modified are automatically externalized to the RBS BLOB store associated with the active provider. Objects that were previously stored in the database continue to remain there and be accessed from within the database; they are not automatically migrated out to the RBS store. In this configuration SharePoint facilitates seamless access to both, the files that have been externalized via RBS, as well as those that are still stored in the database.

While the above mechanism works well, over time, users may want to migrate all existing content that may be stored in the database to the external RBS store; or they may want to migrate all the externalized RBS content back into the database. Both these operations can be performed using the Windows PowerShell™ 2.0 *Migrate()* cmdlet that is provided with SharePoint Server 2010. The exact sequence of Windows PowerShell commands to execute to is shown in the script below.

```
$cdb=Get-SPContentDatabase <ContentDbName>
$rbss=$cdb.RemoteBlobStorageSettings
$rbss.GetProviderNames()
$rbss.SetActiveProviderName($rbss.GetProviderNames()[0])3
$rbss.Migrate()
```

These steps need to be executed for each database on which you want to migrate the BLOBs. Executing the Windows PowerShell script when the RBS provider is enabled results in the BLOBs being migrated out of the database and into the RBS store, while executing the Windows PowerShell script when the provider is disabled results in the BLOBs getting migrated back into the database.

Considering that a content database may have thousands or even millions of objects stored, careful consideration is required before migrating the data as it could take a long time to complete the operations. It is recommended that the *Migrate()* cmdlet be executed during off-peak hours and from a SharePoint WFE or application server that is not heavily utilized.

In our tests, we executed the above script from the application server to migrate 500,000 SharePoint objects each of an average size of 100 KB into and out of the database. The results of our tests are summarized in Table (xv) below.

³ Note: `$rbss.GetProviderNames()[0]` corresponds to the StorSimple RBS provider.

Operation	Time taken (minutes)	BLOBs migrated per second
Migrate data from content database to BLOB store (externalize data)	243	34.3
Migrate data in from BLOB store to content database (internalize data)	504	16.5

Table (xv) – RBS BLOB migration times

We attributed the additional time taken to migrate data in to the content database to additional SharePoint Server and SQL Server processing that needed to be performed in the back end. To ensure that the results were comparable and to stay in compliance with the Microsoft support requirements, no additional tuning was performed on the SQL Server database beyond that mentioned in the software configuration section.

The RBS *Migrate* method is can be restarted and it will start migrating BLOBs into or out of the database from the location where it left off on the previous invocation.

Conclusion

In this paper, we saw how the use of RBS can help reduce the effective size of the SharePoint content database and backup size by more than 95%, thereby reducing the backup time by an equivalent amount as well as providing the option of using cheaper storage to store the BLOB data. We also saw how the RBS feature helps users store large media files in SharePoint Server and get all the benefits of SharePoint Server while not bottlenecking the SQL Server database or making the solution prohibitively expensive. We also analyzed the effects of RBS on the search crawl times, performance of the index rebuild maintenance task (reduced 96%), and the end-user transaction response times (which were reduced by 30% or more for some transactions). Lastly we measured individual large media file upload performance and time taken to migrate BLOB data in and out of the database using RBS.

Overall, we found that the use of RBS helps with the maintainability of a SharePoint farm while enhancing the scalability of the solution. This in-turn results in cost savings and a better end-user experience. However when RBS is used maintenance operations like the backing up of the BLOB store, should be carefully planned and factored into the list of maintenance tasks.

Additional Resources

Overview of Remote BLOB Storage — <http://technet.microsoft.com/en-us/library/ee748649.aspx>

Migrate content into or out of RBS — <http://technet.microsoft.com/en-us/library/ff628254.aspx>

StorSimple SharePoint Database Optimizer — <http://www.storsimple.com/>

Microsoft Office SharePoint Server 2007 Performance Load Tests —

<http://sptdatapop.codeplex.com/releases/view/1214#DownloadId=6918>

Microsoft® SQL Server® 2008 R2 Feature Pack -

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=ceb4346f-657f-4d28-83f5-aae0c5c83d52>

About StorSimple

StorSimple's solution solves key storage-related issues around performance, scalability, manageability, data protection, and cost for Microsoft SharePoint Server 2010. StorSimple uniquely provides you with the ability to deploy on-premises, next-generation storage to address today's application-related challenges while providing you with the ability to leverage public or private cloud storage when you are ready. More information about StorSimple can be found at www.storsimple.com.

About Microsoft

Microsoft Corporation is a public multinational corporation headquartered in Redmond, Washington, USA that develops, manufactures, licenses, and supports a wide range of products and services predominantly related to computing through its various product divisions.