Optimize SharePoint Storage with BLOB Externalization

Dan Holme
Chief SharePoint Evangelist

Randy Williams
Enterprise Trainer & Evangelist

Jeremy Thake
Enterprise Architect

Contents

Introduction

As organizations scale the use of SharePoint as a content management system, supporting document management, records management, collaboration, and archiving, they are moving workloads traditionally supported by file shares and by email attachments to SharePoint. In the process, the storage burden for document-centric content is placed on SharePoint's data tier—the content databases hosted by SQL Server, and often located on the most highly performing and expensive Tier 1 storage resources in the enterprise.

The results include the obvious—skyrocketing storage costs—and the counterintuitive—potentially degraded SharePoint performance. Organizations struggle to architect an optimized, scalable, cost-effective storage architecture for SharePoint that supports potentially terabytes or tens of terabytes (TBs) of content.

There has been ongoing debate in the SharePoint and SQL communities among customers, between experts and MVPs, and even within Microsoft itself about content scalability, the impact of documents on storage capacity and performance, and the role of BLOB externalization, which allows you to relocate the binary large object (BLOB)—the unstructured data that represents the content of a document—to more cost-effective storage tiers.

The debate was reignited when Microsoft updated its guidance for SharePoint sizing in July 2011 to support content databases of up to 4 TB in collaborative workloads, and of unlimited size for document archives. Microsoft's new guidance contained directives toward thoughtful architecture, high-performance storage subsystems, comprehensive governance, and effective management tools beyond those provided with SharePoint to support Service Level Agreements (SLAs) for performance, backup, recovery, and availability.

This white paper will make sense of the issues as well as the varied perspectives and guidance related to BLOB externalization.

We will begin by reviewing the default configuration of SharePoint—with which SharePoint stores a document's content as a BLOB in the content database—and by examining the perhaps surprising multiplier effect of a BLOB on the storage tier through the entire lifecycle of the document. We will then detail the two options that Microsoft has provided for BLOB externalization: external BLOB store (EBS) and remote BLOB store (RBS).

The remainder of the white paper will explore in detail the potential benefits of BLOB externalization, which include:

- Reduced cost of storage
- Optimized performance
- Improved storage management and reduced storage footprint
- Efficient content restructure
- Greater scalability

And we will look at the potential downsides of BLOB externalization: increased architectural complexity and, in particular, the considerations that must be taken for backup, restore, high availability and disaster recovery.

This white paper is the first of a series of storage optimization and management resources to be authored by a team of SharePoint MVPs. The goals of this white paper are to provide a comprehensive and balanced examination of the concepts and issues, and to equip you to fully understand, to communicate with your peers and your management, and to make an informed decision about the role of BLOB externalization in your storage architecture.

Fundamentals

Let’s begin by examining the default configuration of—and the options for—document storage in SharePoint.

Documents, Databases, and BLOBs

In most organizations, SharePoint hosts myriad documents: Office file formats including Word, PowerPoint, and Excel files; PDFs; media files such as images, podcasts and videos; and other files including maps, engineering specifications, scanned documents, and more.

Users upload or save documents to one or more document libraries in a site collection, or attach documents to list items. By default, these documents are stored in the content database of the site collection containing the document library or list.

Within the content database, the metadata for a document or list item is stored in the AllUserData table. If a document is stored in a library, or as an attachment, SharePoint maintains internally used metadata about the document in the AllDocs table. The content of the document is stored as an unstructured data format called a BLOB in the AllDocStreams table. Globally unique identifiers (GUIDs) are used to link the records in these three tables. If versioning is enabled, metadata about historical versions of the document is stored in the AllDocVersions table, and historical versions of the BLOB are stored in AllDocStreams.

SQL Server is a database service that is optimized for performance of structured, relational data, where records are less than 8 kilobytes (KB) in size. Microsoft moved BLOBs to a separate table in order to optimize performance of SQL and therefore of SharePoint. While there is a small performance penalty when a document is opened or saved, because SQL must join the AllUserData table with other tables before the document binary data can be accessed, myriad other processes that rely on efficient SQL Server performance benefit tremendously. So, the split design of content database tables is a net gain for SharePoint.

The Storage Optimization Challenge

Numerous constituencies are affected by the storage infrastructure that supports an enterprise

SharePoint service. Storage architects, database administrators, SharePoint administrators, end users of SharePoint, and organizational management are each concerned, at various levels, about whether the storage infrastructure is available, recoverable, scalable, manageable, cost effective, and able to meet performance expectations.

SharePoint's out-of-box configuration places all content in the content databases of SQL Server. This presents a potential scalability problem. SQL Server is typically hosted on Tier 1 storage—the fastest, most feature rich, and most costly storage tier in an enterprise. If the entire capacity burden is placed on SQL Server, the cost of scaling Tier 1 storage as enterprise content management usage scales might be prohibitive.

The problem increases over time, because as years pass, a typical SharePoint implementation will host increasing amounts of content, but much of that content will be inactive. While total content often increases geometrically, the amount of *active* content increases at a much lower, linear rate. The result is that expensive Tier 1 storage is used as an archive for skyrocketing amounts of content that does not benefit from the performance or features of Tier 1 storage.

You might think that moving content to cheaper tiers of storage might penalize performance. Even if this is the case, there is the opportunity to optimize storage by creating a hierarchical or tiered storage architecture in which less important or less active data is moved to cheaper tiers of storage, while more important or more active data is maintained on Tier 1 storage. But, as you will see later in this white paper, it is possible in many scenarios to reduce storage cost *and* increase performance.

BLOB Externalization

There are those in the SharePoint and SQL community who argue that SharePoint should never have been designed to store documents as BLOBs in a structured relational database like SQL Server. In fact, SharePoint version 1.0 (SharePoint Portal Server 2001) used the Web Storage System, not SQL Server, for content storage. Beginning with SharePoint version 2.0 (Windows SharePoint Services and SharePoint Portal Server 2003), storage was moved to SQL Server.

In recent years, Microsoft has built into SharePoint and SQL Server options that enable moving BLOBs out of SQL Server to other storage tiers, while maintaining metadata in the content database with pointers to the associated documents.

EBS and RBS

In SharePoint 2007 Service Pack 1 (SP1), Microsoft introduced External BLOB Storage (EBS). In SQL Server 2008, Microsoft added Remote BLOB Store (RBS) to SQL Server. At a very high level, both EBS and RBS perform the same basic task: they enable SharePoint to store BLOBs outside of the SQL Server content database.

EBS is part of the SharePoint product, in SharePoint 2007 Service Pack 1 and later, and in SharePoint 2010. SharePoint 2007 supports BLOB externalization only with EBS.

In SharePoint 2010, BLOB externalization is supported using either EBS or RBS. EBS, however, is deprecated and is likely to be removed from a future version of SharePoint in favor of RBS.

RBS can be obtained by downloading the Feature Pack for SQL Server 2008 R2 at http://go.microsoft.com/fwlink/?LinkID=177388. This version (or a later version) of RBS is required by SharePoint 2010. The server components of RBS can be installed on either SQL Server 2008 R2 or SQL Server 2008 SP1. The client components are installed on all SharePoint web front end (WFE) servers.

Modules and Providers

Both EBS and RBS require additional components to manage the communication between SharePoint and the non-SQL Server location for BLOB storage, called the *BLOB store*. EBS uses a plug-in framework that requires a third-party module to implement. RBS exposes a set of Application Programming Interfaces (APIs) on top of which developers and ISVs can build providers. A provider is the interface between RBS and a specific type of BLOB store.

Microsoft created the FILESTREAM provider, which is included free-of-charge with the RBS installation files. The FILESTREAM provider allows you to externalize BLOBs to the local file system of the SQL Server. This can include direct-attached storage and iSCSI-attached SAN and NAS volumes, assuming those volumes meet SharePoint's storage performance requirements of at least .25 input/output operations per second (IOPS) per gigabyte (GB) stored, and no more than 20ms time-to-first-byte (TTFB).

While many in the industry maligned the performance of the FILESTREAM provider, experience has shown that its performance is at least better than expected. Some ISV providers outperform FILESTREAM, some underperform. An RBS provider for a cloud storage platform, for example, is likely to be slower due to the nature of the cloud storage itself. It is recommended that you test BLOB externalization providers in your environment to identify performance differences and, more importantly, to determine whether such differences have a material impact on overall SharePoint performance.

Performance aside, many organizations turn to ISV providers for their broad support for diverse storage platforms, business rules for externalization, management features, and maintenance. ISVs have built providers for BLOB stores in shared folders on a remote server, on cloud storage platforms, and on specific NAS and SAN platforms. ISV storage solutions provide various levels of rule-based management of BLOBs, so that an organization can specify what types of BLOBs are and are not externalized and thus create a hierarchical storage management platform. Installation and configuration of ISV providers is generally a significantly better experience than the FILESTREAM provider. Maintenance tasks are generally automated. And ISV providers typically deliver monitoring and reporting capabilities. Such features come at a cost, of course, which must also be considered. Later in this white paper, we will discuss the capabilities of third-party BLOB externalization solutions.

Functional Parity

It is important to note that BLOB externalization is transparent to the rest of SharePoint. SharePoint considers BLOBs part of its content repository, regardless of where the content is physically located.

Therefore, whether BLOBs are stored in a content database or are externalized, SharePoint manages SharePoint permissions to the content. SharePoint continues to index the content. All SharePoint document management features, including check-out and versioning, continue to be available. None of the business-value functionality of SharePoint changes when BLOBs are externalized.

Migrating to EBS or RBS

Generally speaking, you can enable EBS or RBS at any point in your SharePoint journey. Similarly, you can modify the rules that govern which BLOBs are externalized at any time. There are tools you can use—Windows PowerShell in the case of RBS—to scan the content database and move content out to the BLOB store.

The following sections will examine the pertinent factors—the benefits and considerations—that you should take into account as you plan for an optimized storage infrastructure for SharePoint. Due to the fact that EBS and RBS deliver conceptually similar benefits and challenges, we will refer generically to *BLOB externalization* solutions.

Benefits

BLOB externalization has the potential to deliver significant benefits in many workloads. Benefits include reduced cost, optimized performance, improved storage management, reduced storage footprint, efficient content restructure, and greater scalability.

Reduced Cost of Storage

The first, and most obvious consideration related to SharePoint storage optimization is cost, which is directly related to capacity and to the tier(s) of storage that host SharePoint content. Just how much do BLOBs impact capacity and cost? The answer might surprise you.

The 80 Percent Estimate

In a typical content database, documents—their metadata and the BLOBs that contain their content—tend to consume significant real estate. The content database becomes bloated by the BLOBs it stores.

> *Typically, as much as 80 percent of data for an enterprise-scale deployment of SharePoint Foundation consists of file-based data streams that are stored as BLOB data. These BLOB objects comprise data associated with SharePoint files.*

http://msdn.microsoft.com/en-us/library/bb802976.aspx

This “80 percent estimate” doesn’t tell the full story, however. What this 80 percent estimate fails to illuminate is the scale of the impact of BLOB storage. In a typical collaborative environment, one often finds that a document is not stored just one time—that the amount of storage required for a single document is significantly more than you would expect—in fact, many multiples of the document’s size.

Let’s examine the total impact of a single document and its BLOB storage through the entire lifecycle of a document.

Document and Metadata Storage

SharePoint supports documents up to 2 GB in size (http://technet.microsoft.com/en-us/library/cc262787.aspx#ListLibrary), a software boundary that results from a 32-bit pointer used in SQL Server. There is no way to exceed that limit and to store larger documents in the content database—with or without BLOB externalization.

SharePoint includes a file size upload limitation that's configurable per web application. The default maximum upload size is 50 megabytes (MB)—considerably smaller than the 2 GB hard limit.

This lower limit reflects practical concerns including network performance, the performance of transferring large files over HTTP, and user expectations for performance of file transfer. Many organizations retain this default upload size or raise the limit. If you choose to raise the maximum upload size, do so slowly and after careful testing.

Each document in a SharePoint library has metadata associated with it. Some metadata is user-configured, such as columns in the library.

Other metadata is used internally by SharePoint. The amount of metadata associated with a document will vary based primarily upon user-configured metadata.

It's easy to understand that scenarios with larger documents see a higher ratio of BLOB-to-metadata storage, and scenarios with smaller documents and more metadata will see lower BLOB-to-metadata ratios. The 80 percent estimate is based on an average across multiple SharePoint environments.

But here’s the rub: A document is rarely, if ever, stored only once.

Document Versions

When version history is enabled for a document library, any change to the document or its metadata results in additional storage utilization. Two points are often misunderstood and have significant impact on storage when versioning is enabled:

1. No differential compression is used within SharePoint. When a new version is saved, the amount of storage represents the entire size of the file—not just the differences or deltas between versions. Conceptually, two versions of a document with minor changes will occupy 2 x (*document size* + *metadata*) of storage.

2. A new version of the document is created if the document—or its metadata—is modified. If a document is uploaded to a library and is never changed, but the metadata associated with that document is changed five times over the course of a month, the storage occupied by that document is approximately 5 x (*document size* + *metadata*).

When versioning is enabled, the impact of a document on storage is multiplied by the number of versions of that document.

Therefore, it's critical to enforce limits to version history—unlimited version retention can lead to significant database bloat.

Recycle Bin Contents

When a document is deleted, the document and its versions are retained based on the web application's settings for the SharePoint Recycle Bin. A user can restore a document she deleted from the Recycle Bin.

When a user empties the Recycle Bin, the document and its versions continue to be retained, and can be restored by site collection administrators, from what is referred to as the *second-stage* Recycle Bin.

Each site collection has a Recycle Bin. However, the Recycle Bin has two configurable settings that are both scoped to a web application. These settings apply to all site collection Recycle Bins in the web application.

The first Recycle Bin setting specifies the total number of days that a deleted document will be retained by the Recycle Bin. This setting applies from the moment the document is deleted.

It doesn't matter whether the document is in the user Recycle Bin or the second-stage Recycle Bin. *X* days after a document was originally deleted by the user, it's deleted from the Recycle Bin and the document is removed from the content database.

The second setting applies a storage quota to the second-stage Recycle Bin. When items are moved to the second-stage Recycle Bin, they count against this quota. When the quota is reached, the oldest items in the second-stage Recycle Bin are removed to make room for newly deleted items.

The quota is configured as relative to the quota of the site collection. So if a site collection is subject to a 50 GB quota, and the second-stage Recycle Bin is limited to 50 percent of the quota, then the second-stage Recycle Bin for that site collection is effectively capped at 25 GB.

Therefore, the total storage impact of a document on a content database must take into account the fact that, until a document is purged from the second-stage Recycle Bin, the document—its BLOB and metadata—and those of the document's versions continue to impact the content database.

Auditing

Audited activities generate entries in the audit log. The amount of storage required for auditing can be significant, particularly if you are auditing *view* activities. However, audit entry size and the size of audit logs is not related to document size, or to whether BLOBs are stored in SQL or are externalized. Therefore, while you should consider auditing when estimating total storage requirements for a content database (http://technet.microsoft.com/en-us/library/cc298801.aspx#Section1), we will not examine auditing in more depth in this white paper.

Office Web Apps Cache

In order to improve performance of SharePoint when the Microsoft Word web app and Microsoft PowerPoint web app are used, the web apps create renditions of a document in a cache called the Office Web Apps cache.

When a document is rendered, it can be pulled from the cache. A document is re-rendered only if it doesn’t exist in the cache, or if the document has changed after the rendition in the cache was created. A timer job removes documents from the cache after a configurable expiration period.

If a web application is associated with the Microsoft Word or PowerPoint web apps, one content database will contain the cache for all content in the web application. In a document-heavy web application, the cache can grow quite large.

By default, the cache is capped at 100 GB. It's a best practice to configure Office Web Apps to use a separate, dedicated content database in a SharePoint web application, and to manage the size of the cache to optimize performance and storage. You can learn more about this at http://technet.microsoft.com/en-us/library/ee837422.aspx.

The size of the Office Web Apps cache doesn’t depend on whether BLOBs are stored in SQL Server or are externalized. Rather, it’s based purely upon the number and size of documents, frequency of access to those documents, and on administrator configuration.

So while the Office Web Apps cache should be considered as part of the estimate of storage required for a web application, it will not, if in a dedicated content database, affect the storage required for other content databases in a web application.

Service Databases

A document indirectly affects the storage required by service applications. For example, access to a document might be tracked by the Web Analytics service application.

Tagging, commenting and rating activities consume approximately 9 KB per entry in the social tagging database of the User Profile service application.

Such data are relatively negligible, and is neither dependent on whether a document’s BLOB is stored in SQL Server or is externalized, nor directly dependent on the document’s size.

However, the Search service application is affected directly by both the number of documents and their size. The crawl database, properties database, and index partitions each have a direct relationship to the number and size of documents.

Search capacity planning is both a science and an art, but very rough estimates from typical implementations fall around 20 percent of the total size of indexed content (the corpus).

So, if you are indexing 1 TB of typical content, you can expect approximately 200 GB of storage utilization by search-related databases and the index. For more information, see http://technet.microsoft.com/en-us/library/gg750251.aspx.

Search and other service database sizes aren't dependent on whether BLOBs are stored in SQL Server or are externalized.

Transaction Logs

SQL Server logs all activity to the transaction log for a database before committing the transaction to the data portion of the database. Transaction logs grow until a log backup, at which point space used by the log is cleared, but the file size does not shrink.

You can shrink a SharePoint transaction log manually, which can be helpful if a transaction log has grown out of control, but the best practice is to manage transaction log size by managing transaction log backups.

When a document is uploaded or modified, the document BLOB and metadata are first written to the transaction log. Then, the transaction is committed to the appropriate tables in the content database itself.

Therefore, the true impact of a document on total content database size, including the transaction log, can be approximated as *document size* x (*creation* + *modifications*) x 2 during the window between log backups.

The transaction log size is directly related to BLOB storage. If BLOBs are externalized, and aren't stored in the content database, then the BLOB is also not written to the SQL Server transaction log.

BLOB Externalization, Capacity, and Cost

As you can see, the storage required for just one document can vary greatly, based on version retention, modification of the document or its associated metadata, web application settings such as Recycle Bin, auditing settings, the use of Office Web Apps and other service applications, and even backup policies. In a typical, highly collaborative scenario, an active document may be consuming storage equivalent to many multiples of the document's actual size.

BLOB storage inside the content database can be expensive, from both a hard cost and total cost perspective. Many organizations have traditionally stored documents for business collaboration on file servers, which are relatively cheap compared to the typical, high-IOPS, Tier 1 storage that is allocated to support SQL Server for SharePoint. Moving such content to SharePoint and into Tier 1 storage—especially with the understanding that a document may require multiples of its size to support both the business and internal functionality of SharePoint throughout the lifecycle of that document—can be a costly proposition.

BLOB externalization does not by itself reduce the total storage footprint of your SharePoint infrastructure, except for reduced impact on transaction logs. But it does enable you to transfer the storage burden to more cost effective tiers. The cost savings can be tremendous. Some organizations report savings of tens of millions of dollars a year from storage optimization efforts focused on BLOB externalization. And some storage platforms have features that can, in combination with BLOB externalization, reduce the total storage footprint of SharePoint. Such features will be discussed later in this white paper.

Optimized Performance

It might seem that all BLOBs should be moved to cheaper tiers of storage in order to reduce cost of storage. But you must also consider the impact of BLOB externalization on SharePoint performance.

For example, if you use a third-party provider to externalize BLOBs to a cloud storage platform, such as Amazon or Azure, it stands to reason that reads and writes of documents will be slower than local SQL Server storage.

Interestingly, it is not true that all BLOB externalization reduces performance. In fact, it is possible to increase performance through BLOB externalization in certain scenarios and configurations.

The question is: At what point does performance of SharePoint improve with BLOB externalization, and at what point does performance degrade?

The answers to these questions require close examination of two performance-related issues:

- The performance of read and write access to a single document
- The performance of the entire SharePoint service—across all site collections in a content database and across all content hosted in SQL Server—in a real-world scenario

Most documentation in the SharePoint community focuses on only the first issue, which unfortunately clouds decision making and leads to designs that sacrifice performance in a typical, real-world production environment.

In the following sections, we will explore these two aspects of performance. It is critical to remember, however, that the only real way to know how SharePoint will perform with or without BLOB externalization in your enterprise is to test it with workloads that are realistic and representative of your environment.

In addition, you must consider whether performance degradation is noticeable to users, and whether—even with reduced performance—SharePoint can still meet user expectations and SLAs for performance. Finally, you must consider each scenario or workload that you are supporting. Users might find it acceptable for access to an archived record to be slower than access to documents around which they are actively collaborating.

Performance of Access to a Single Document

The out-of-box configuration of SharePoint stores BLOBs in SharePoint content databases. This can provide optimal performance in certain scenarios.

- **Tier 1 Storage Performance** – Many organizations dedicate high-performance, Tier 1 storage subsystems to SQL Server. The performance of the SQL Server storage platform directly impacts SharePoint performance. Documents accessed from a high-performance SAN will perform better than documents with BLOBs externalized to public cloud storage.
- **Small Files –** When a small document is read from—or written to—a content database, performance of that single activity is often optimal.
- **Frequently Read Files –** When a small document is accessed regularly for read access, SQL caching can further improve performance of access to that document. A recently accessed, cached document can be retrieved from memory, rather than from disk.

SharePoint reaches a cross-over at which point performance is better when the BLOB is externalized for a particular workload.

Reads and writes of a single small document might be faster when stored in SQL. Read performance can be further improved when a document is in memory due to SQL caching. But as files grow in size or are accessed less frequently, performance can be improved by storing BLOBs in a platform that is better suited for file storage.

Write performance crosses over more quickly than read performance because a BLOB must be written twice—first to the transaction log, and then committed to the database table. There is also no caching benefit to a new or modified document. So as a document's size increases, the performance penalty of writing the document's BLOB once, let alone twice, reduces performance of documents stored in a content database.

Fortunately, RBS enables you to specify a file size threshold, above which a document's BLOB is externalized, and below which the BLOB is stored in the content database. EBS solutions as well as custom or third-party RBS solutions can use other rules, such as file type, to determine whether or not to externalize a BLOB.

There is no formula with which to determine the file size threshold at which performance increases with BLOB externalization. There are simply too many factors in the performance equation, including the characteristics and access pattern of the document, and the performance characteristics of underlying storage platforms.

Our testing has shown that BLOBs greater than 1 MB generally perform better when externalized—assuming the BLOB store itself performs well—whereas very small files smaller than 256 KB generally perform better when stored in the content database.

Our findings align with the general consensus among consultants that documents greater than 512 KB or 1 MB should be externalized, at which point performance is improved for both reads and writes, given similar performance characteristics of the underlying storage platform; whereas access to a document smaller than 256 KB is faster with the BLOB stored in the SQL Server content database.

Between 256KB and 512KB or 1MB is the cross-over point that is heavily dependent on size and access patterns. For example, a 512KB file that is accessed frequently for reads may benefit from SQL caching, and therefore perform better when stored in the content database. The same file, archived and accessed infrequently for modification, might perform better when externalized. Write-heavy access patterns are improved by externalizing BLOBs to a greater extent than read-heavy access patterns.

Performance is also heavily dependent upon the specific EBS or RBS solution that you employ, and the characteristics of the BLOB store—the performance of the underlying storage subsystem. For example, a BLOB store on a SAN will perform better than a BLOB store hosted in the public cloud.

The threshold at which performance of file access improves depends on a variety of factors, and should be tested with your EBS or RBS storage solution as well as with content and access patterns that reflect expected workloads in your environment.

Performance of the Content Database, SQL Server and SharePoint Farm

We do not think it is helpful to get too caught up in discussions about—or testing—the performance threshold of access to a single BLOB unless your production SharePoint environment consists of one user accessing one document. In a real-world SharePoint environment, multiple users are accessing documents as well as lists, pages, workflows, services, and applications. Therefore, it is more important that you test the SharePoint environment with appropriate and complex workloads that are representative of day-to-day usage.

As multiple users access SharePoint, the impact of BLOBs on SQL Server is exacerbated, and BLOB storage in a content database can significantly impact performance of the entire content database, of the server running SQL Server and subsequently, the SharePoint farm.

As mentioned earlier, SQL Server is a database service that is optimized for performance of structured data. When you store unstructured data as BLOBs, SQL Server needs to work harder to save and retrieve the data. SQL Server is simply not optimized to be a file server.

The degradation of performance due to BLOBs is also true outside of SQL Server and SharePoint. One customer reported that they had a custom engineering application built on Oracle. When they used functionality in Oracle to remove BLOBs from the database, performance skyrocketed.

The broader performance effects of BLOBs stored in SQL Server are seen in real-world SharePoint environments, as users access documents, and other users access pages or list views.

If a content database has a large number of BLOBs (e.g. a document library with many files), and a reasonable level of activity that involves BLOB access, all other performance is degraded—for example, the performance of list views is slowed. As some users load and store BLOBs, CPU and memory utilization skyrocket, and performance suffers for other users across all content databases stored on the

SQL Server. Because SQL Server is the most common bottleneck in the request pipeline, SharePoint performance suffers.

We have observed that the mere existence of BLOBs can visibly degrade performance of content access, even if the content access you are measuring has nothing to do with a document or BLOB. This is partly due to the fact that SharePoint and SQL services and administrators perform operations in the background—including database indexing, search indexing, and management operations such as database backup—that involve BLOB access.

When you externalize BLOBs from a content database, performance of the content database and of the SQL Server—its CPU and memory utilization—can improve. This means that performance of list views and other day-to-day user and maintenance activities—even those not related to the BLOBs themselves—can be significantly faster.

So while it is interesting to discuss whether a single document of 512 KB or 1 MB should be externalized to improve the performance of access to that one document, our experience is that such a discussion detracts from the more significant impact of BLOBs—even small BLOBs—in more complex environments.

Microsoft reported a 25 percent performance increase by externalizing BLOBs, across a mix of user workloads that reflect a real-world collaborative environment. Microsoft's findings are summarized in the table below.

	SQL BLOB	RBS	Gain
Database Size - 1 TB	2292 GB	26 GB	98.9%
Database Backup Size - 100 GB	217 GB	7 GB	96.8%
Database Backup Time - 217 GB	2490 sec	38 sec	98.5%
Database Defrag Time - 100 GB	120 sec	4 sec	96.7%
Avg. SharePoint Response Time	28 msec	21 msec	25.0%
Large File Upload - 500 MB	55 seconds	29 seconds	47.6%

Source: SQL Server RBS Performance with SharePoint Server 2010 and StorSimple Storage Solution
http://www.microsoft.com/download/en/details.aspx?id=14726

We have also performed advanced testing on the impact of BLOBs on real-world, mixed workloads, and we will release detailed results of the testing at a later date.

At this time, our qualitative observation and Microsoft's quantitative results can inform your consideration of BLOB externalization: BLOB storage in SQL content databases—even for small BLOBs—can degrade performance significantly in real-world SharePoint environments.

Of course, there are a number of factors in this performance equation, as well, including the overall quantity of BLOBs and access patterns. But because of this potential database- and server-wide performance improvement, you must consider the "big picture" of data access within a content database and on an instance of SQL. For example, you might decide to externalize all BLOBs bigger than 128KB or 256KB in order to improve performance across most usage scenarios for the site collections within a content database, even though by doing so, you might degrade performance of access to very small files. In other words, opening a small 256KB file may now be slower, but everything else in the database is faster. This is one reason why you must test the performance of BLOB externalization in your environment—there is no "easy answer", but rather a threshold that is unique to your usage and access patterns.

In our experience, the potential for across-the-board performance improvement is under-documented and under-appreciated. Too many organizations make BLOB externalization threshold decisions based only on the more plentiful documentation and guidance related to the performance of BLOB access. In other words, organizations set a threshold that may be ideal for accessing the files, even though those files are not regularly accessed, at the expense of database- or server-wide performance.

Improved Storage Management and Reduced Storage Footprint

When you externalize BLOBs to a storage platform other than SQL Server, you instantly gain the feature set of the storage platform. For example, if you store BLOBs on an NTFS file system on a Windows server, you can enable NTFS encryption. Encryption of BLOBs might be a requirement of your regulatory environment.

Alternatively, you might store BLOBs on a SAN volume that supports single-instancing (de-duplication). So if users store the same document in several locations, the storage platform can recognize the duplication and store only one instance of the document.

Some storage platforms perform differential compression, where two documents that are very similar are stored as a document and a small difference file. And many storage platforms can perform compression of the BLOB itself.

Earlier in this white paper we mentioned that a document can be stored multiple times in a typical collaborative environment, particularly when versioning is enabled. These kinds of storage features can significantly reduce the total capacity required to support your SharePoint content, and can therefore significantly reduce storage cost.

Finally, some storage platforms offer valuable storage management features. For example, many storage subsystems support snapshots as a backup option. Whatever features are offered by your storage platform, those features are now available to your BLOB store.

Efficient Content Restructure

In Service Pack 1 (SP1) for SharePoint 2010, Microsoft introduced support for shallow-copy. This means that you can move content—a site collection—between content databases in a web application without touching the BLOBs in the BLOB store. Third-party tools extend the ability to restructure content between web applications or farms. Only the content in the content database—including the pointers to the BLOB store—are moved.

There are several caveats and nuances to this feature, but it is an important addition to the SharePoint toolset, as it makes it easier and significantly faster for an organization to restructure information architecture and the logical design of a SharePoint web application.

Greater Scalability

BLOB storage within the content database places immediate limitations on the scalability of SharePoint if you use only SharePoint's out-of-box tools and configuration in your implementation. Until July 2011, Microsoft's guidance for content database scalability was a content database should be limited to 200 GB; and that a site collection should be no bigger than 100 GB unless it is the only site collection in a content database.

In July 2011, Microsoft's guidance for content database scalability changed significantly—and for the better. Content databases of up to 4 TB are supported in collaboration scenarios, and content database size is unlimited for document archives.

Microsoft also introduced an additional scalability limit: A content database is supported up to 60 million items, which includes all list items and documents in all site collections in the content database. The total number of items in a content database impacts upgrade and update SLAs. If you have a very large number of rows in a content database, the performance of patches, updates, service packs, and upgrades can degrade to the point where an upgrade cannot complete within the maintenance window of an SLA or fails completely.

However, there are a number of caveats related to performance and manageability which make it difficult to scale content databases to sizes approaching or exceeding 1 TB without the following:

- High performance storage subsystems
- Third-party enhancements to the out-of-box management tools
- Externalizing BLOBs
- Leveraging storage-layer functionality to meet SLAs for recovery

It is important to note that BLOB externalization can—in combination with thoughtful architecture, comprehensive governance, and effective management tools—enable greater scalability. For example, by externalizing BLOBs to storage tier that supports snapshot backups, you can greatly reduce the backup time window for large content databases.

However, BLOB externalization does not change the supported limit of 4 TB for a collaboration database, or any other scalability limit. When you use EBS or RBS to externalize BLOBs, you must still respect the content database size limits described earlier. In other words, just because you are externalizing BLOBs does not mean you can suddenly have more total data within the scope of a content database.

It is also worth reiterating that BLOB externalization does not, on its own, allow you to store a single file that is bigger than 2 GB. This is a hard limit of the SharePoint platform—and of SQL Server specifically. Third-party solutions work around this limit, but not by using EBS or RBS functionality.

Considerations

Like any technology, EBS and RBS are not silver bullets, and BLOB externalization is not appropriate for every workload or configuration. We have already noted that certain scenarios will experience performance degradation that must be evaluated relative to SLAs and user expectations for performance. But what are the other downsides of BLOB externalization—the issues that must be considered?

Increased Architectural Complexity

When you externalize BLOBs, you add complexity to your storage architecture. Whether that architectural complexity translates to more complex manageability is dependent, in great part, on the processes and tools available for managing SharePoint.

Backup, Restore, High Availability and Disaster Recovery

The management scenario that is of greatest concern to most organizations relates to backup, restore, high availability, and disaster recovery. When BLOBs are stored in SQL Server, database maintenance and management is straightforward: Backups of a content database contain all content, including BLOBs, of site collections stored in that content database.

After you externalize BLOBs, you must consider the BLOB store in plans for backup, restore, high availability, and disaster recovery. Here, the story *can* be complex, but doesn't have to be. The bottom line is that you must understand your implementation, which includes your EBS/RBS configuration, your tools, and your procedures. You must have tested to ensure that you can meet your SLAs for backup, restore, high availability, and disaster recovery.

Out-of-the-Box Backup and Restore

Let's take RBS and the FILESTREAM provider as an example. If you externalize BLOBs with RBS and the FILESTREAM provider, you are using a SQL API and a provider to determine what is, and is not, redirected to the BLOB store.

If you use SharePoint's backup APIs—that is, if you use Central Administration or Windows PowerShell to back up SharePoint—SharePoint asks for content from SQL, and SQL provides the content, retrieving metadata from the AllUserData table and BLOBs from the BLOB store. Your backup *will* include the files from the BLOB store. Similarly, if you use SQL backup APIs—that is, if you open SQL Server Management Studio and choose to Backup Database—your backup will include BLOBs from the BLOB store. Microsoft explicitly supports farm database backup and restore using the FILESTREAM provider.

However, other RBS providers, externalization by EBS, and other backup and restore tools will perform differently. Therefore, you must consider several factors in relation to the service level agreements (SLAs) that you have defined for SharePoint.

Recovery Point Objective and Acceptable Data Loss

Your recovery SLAs should address recovery point objective, or RPO, which relates directly to the frequency of backups. If backups are taken every hour, the SLA for RPO is also one hour—the maximum amount of data loss is the time between backups.

When you externalize BLOBs, you must ensure that the backup of your SQL content database and the backup of your BLOB store are consistent within the SLAs that you have defined for SharePoint.

Conceptually, it is ideal for content database and BLOB store backups to be synchronized, to maintain data consistency. There are third-party software and hardware solutions that achieve this goal.

Some third-party software solutions back up each document as a complete unit that contains both metadata and BLOB. Such a backup ensures that the document can be recovered to the point in time of its backup.

Some storage platforms support snapshot backups, which can back up terabytes of data in mere seconds. Such solutions can be costly, but are often critical in large content scenarios. They are not always necessary, however, as long as the time window for the loss of newly introduced or modified content is acceptable within the scope of your SLAs.

It is also possible to manage backups of a content database and its associated BLOB store separately. There is a lot of hype about how complicated this can be. It is not complicated at all, as long as you understand the sequence of tasks as well as the potential for an additional window of data loss.

The process for backing up a content database and its associated BLOB store is as follows:

1. Back up the content database
2. Back up the BLOB store

The time between the start of Step 1 and the end of Step 2 represents a window of partial data loss in a restore operation. Consider that you back up the content database. After the backup completes, but before you begin the backup of the BLOB store, a user adds a document or a document version to SharePoint. The BLOB for the document goes into the BLOB store, and is backed up. The metadata and pointer to the BLOB are placed into the content database, but after the backup.

If a restore is required, you must perform the following steps:

1. Restore the BLOB store
2. Restore the content database

The BLOB store contains the newly added document, but because the content database does not, the document will not be visible to SharePoint. A process called *garbage collection* runs periodically to purge orphaned BLOBs from the BLOB store. The document that was added during the time window between the content database backup and the BLOB backup has been effectively lost.

As an interesting note, you do not need to worry as much about *deletions*. Consider that a content database is backed up at 1 a.m., and then a user permanently deletes a document from SharePoint (including the Recycle Bins) at 1:10 a.m.

When a document is deleted from SharePoint, the information is removed from the content database, but the BLOB itself is not deleted until the garbage collection process runs. So the BLOB remains in the BLOB store.

The BLOB store is then backed up at 1:15 a.m.

In a recovery scenario, the BLOB store is recovered first with the orphaned BLOB of the deleted document. The content database is recovered to a point in time before the document had been deleted. The document is effectively fully restored as it existed at 1 a.m.—the time of the content database backup.

Therefore, the real guidance related to backup is: You must test the restore of your backups to ensure that BLOBs are protected to a level that meets your SLAs for recovery point objective (RPO) and data loss. You must validate that your backup and restore performs as expected given your database and BLOB store configuration, the APIs you are using (EBS vs. RBS), and the specific tools and processes you have in place.

Backup Window and Recovery Time Objective

Another significant consideration for backup and restore SLAs is the time required to back up or restore content. If you have a large BLOB store, backup may not complete within the maintenance window of your SLA. And, if you have to restore content, it might take too long to recover the backup from tape or from another backup media to your SQL Server before you can even begin restoring content.

SharePoint's out-of-box management tools require that, to restore content, you must first copy or restore the backed up database from your tape backups or disk archives to the SQL Server, then proceed with the SharePoint restore operation.

Until recently, Microsoft supported content databases up to only 200 GB. One of the drivers behind this supportability limit was the practical reality of ensuring reasonable recovery time objective (RTO) SLAs. If a content database is several hundred gigabytes in size, copying or restoring it to the SQL Server can take a significant amount of time before content can actually be recovered.

Third-party hardware and software solutions can reduce the RTO significantly through a variety of capabilities, including snapshots and functionality to directly mount remote backups.

High Availability and Disaster Recovery

Externalized BLOB storage must also be considered for high availability and disaster recovery scenarios. To provide high availability of SharePoint's database tier, you can install SQL Server on a cluster, which ensures access to the databases in the event of a failure of the primary server in the cluster. You can use log shipping to create a warm standby of your SQL Server databases for disaster recovery purposes.

SQL Server 2008 R2 also provides SQL mirroring, which can add redundant access to the database files themselves in the event of corruption or failure of the storage platform. However, SQL mirroring is not supported for content databases that use RBS to externalize BLOBs.

However, each of these high availability and disaster recovery approaches protect only the content database. The BLOB store must also be made highly available and recoverable. The configuration that enables the "link" between document metadata in the content database and a specific BLOB in the BLOB store must also fail over correctly.

There is no single best practice or "recipe" for configuring high availability and disaster recovery with externalized BLOBs. Instead, you must architect a solution based on your business and technical requirements, and with the specific hardware platforms, software, staff, skillsets, and procedures you have in your environment.

In a subsequent article, we will examine hardware, software, and procedural options for supporting high availability, disaster recovery, and backup and restore.

Examining BLOB Externalization Options

Microsoft delivers the FILESTREAM provider for RBS free-of-charge, yet most enterprises turn to an ISV solution to support BLOB externalization as well as support backup and restore SLAs of a scaled storage environment. Third-party solutions fill the gaps between business requirements and the out-of-box SharePoint capabilities.

BLOB Externalization Features

The following tables summarize some of the features that are offered by third-party, ISV solutions that extend the capabilities of the FILESTREAM provider. For the purposes of this table, ISV solutions that leverage both EBS and RBS are included.

Support for SharePoint Versions and Storage

FEATURE	FILESTREAM	THIRD-PARTY
SharePoint 2010 (Server and Foundation)	✓	✓
SharePoint 2007 (MOSS 2007 and WSS v3)	✗	✓
All SQL versions (2000, 2005, 2008, 2008 R2)	✗	✓
All SQL editions (Express/Standard/Enterprise)	✗	✓
Externalize BLOBs to DAS, iSCSI NAS/SAN	✓	✓
Externalize BLOBs to file share, WORM	✗	✓
Externalize to the Cloud (Azure, Amazon, etc.)	✗	✓
Native compression and encryption	✗	✓
Externalize to multiple storage providers within 1 content database	✗	✓

Backup, Recovery, and Disaster Recovery

FEATURE	FILESTREAM	THIRD-PARTY
SharePoint out-of-box backup captures the BLOB and the metadata	✓	??
Synchronous, separate backups of BLOB store & SharePoint	✗	✓
Backup of content DB independent of BLOB store	✗	✓
Item level recovery	✗	✓
Platform level recovery	✗	✓
Restore without database staging	✗	✓

Content Lifecycle Support

FEATURE	FILESTREAM	THIRD-PARTY
Content restructure (shallow copy) with a Web app	✓	✓

Content restructure (shallow copy) across Web apps	✘	✓
Content replication	✘	✓
Connect to and manage file shares through SharePoint	✘	✓
Connect to and manage media shares through SharePoint	✘	✓
Business rule support (content type, metadata, access date)	✘	✓
Externalize to hardware-based HSM	✘	✓

Examining Third-Party Solutions

As you examine third-party solutions, evaluate them for the following characteristics:

- Performance
- Support for the BLOB store platform of your choice
    - File system, SAN, NAS
    - Shared folder
    - Cloud storage
- Integration with your storage platform
- Rules that can be applied to determine whether BLOBs are externalized
    - File size
    - Most-recently accessed date
    - Content type
    - Metadata
    - Other business rules
- Manageability
    - Ease of installation and configuration
    - Maintenance tasks
    - Reporting and monitoring
- Support for SLAs for backup and recovery of content with full fidelity of metadata and BLOBs
- Support for SLAs for high availability
- Support for SLAs for disaster recovery
- Management of long term retention, archiving & tiered storage
- Cost

Conclusion

The storage infrastructure for SharePoint 2010 must support a scalable, available, manageable repository that performs in line with SLAs and user expectations for access to content. As organizations scale the use of SharePoint to support enterprise content management and collaboration, document content—stored in BLOBs—proliferates. BLOBs in a content database increase the storage burden on expensive, Tier 1 storage. By externalizing BLOBs, an organization can reduce cost, optimize performance, improve manageability, reduce the storage footprint, restructure content, and scale the content repository. But, like any technology, BLOB externalization must be evaluated on a workload-by-workload basis. You must carefully consider the design, implementation, and ongoing management of a storage architecture that includes BLOB externalization.

Resources

Please refer to the following resources below, which give further guidance and background information that fortify the discussion points, recommendations, and conclusions made in this white paper.

SharePoint Server 2010 Capacity Management: Software Boundaries and Limits:
http://technet.microsoft.com/en-us/library/cc262787.aspx#Column

SharePoint 2010: Improve SharePoint 2010 Performance with RBS:
http://technet.microsoft.com/en-us/magazine/ff847521.aspx

Install and Configure RBS (SharePoint Foundation 2010):
http://technet.microsoft.com/en-us/library/ee663474.aspx

SharePoint Team Blog: Data Storage Changes in SP1:
http://sharepoint.microsoft.com/blog/Pages/BlogPost.aspx?pID=988

MSDN: External Storage of Binary Large Objects (BLOBs) in SharePoint Foundation:
http://msdn.microsoft.com/en-us/library/bb802976.aspx

White Paper: SQL Server RBS Performance with SharePoint Server 2010:
http://www.microsoft.com/download/en/details.aspx?id=14726

SQL Server RBS Performance with SharePoint Server 2010 and StorSimple Storage Solution:
http://www.microsoft.com/download/en/details.aspx?id=14726

Estimate Performance and Capacity Requirements for SharePoint Server 2010 Search:
http://technet.microsoft.com/en-us/library/gg750251.aspx

Manage the Office Web Apps Cache:
http://technet.microsoft.com/en-us/library/ee837422.aspx

Acknowledgements

The authors wish to thank George Wang, Mary Leigh Mackie, and Christopher Musico for their incredible contributions to this white paper.

Trademarks

Changes

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, AvePoint assumes no liability resulting from errors or omissions in this document or from the use of the information contained herein. AvePoint reserves the right to make changes in the product design without reservation and without notification to its users.