

OVERVIEW

This whitepaper considers why you might want to consider a solution that allows you to get specific content out of SharePoint's underlying infrastructure and in to a more 'appropriate' location. It discusses the pros and cons of the different pairing technologies that you might be considering and then give you some guidance on 'which one and when'.

WHY EXTERNALIZE THE DATA IN THE FIRST PLACE?

It is now fairly common knowledge that the decision to store Binary Large Objects (BLOBs) in SQL Server was not one of SharePoint's best architectural calls. It works exceedingly well for smaller deployments and certainly simplifies backup, DR, etc. but it creates some serious limitations when you start to expand SharePoint's footprint. Microsoft themselves state that:

"...as much as 80 percent of data for an enterprise-scale deployment of Windows SharePoint Server consists of file-based data streams that are stored as BLOB data. These BLOB objects comprise data associated with SharePoint files. However, maintaining large quantities of BLOB data in a Microsoft SQL Server database is a suboptimal use of SQL Server resources. You can achieve equal benefit at lower cost with equivalent efficiency by using an external data store to contain BLOB data."¹

So, where's a BLOB to live then? This paper considers some options including externalizing content to the file system, to an archive, to a traditional ECM system or just not putting it in SharePoint in the first place.

MOVING CONTENT OUT OF SQL SERVER – TO A FILE SYSTEM

This section discusses pairing SQL Server to a file system. When Microsoft first announced EBS and RBS² it looked like we should just all consider writing file system providers that would take the Binary Large Objects (BLOBs) from SQL Server and dump them on a local file system. I'm sure some people did just that but you run out of benefits pretty quickly. You can eke out the value by using virtualized file systems but still the challenges addressed are pretty limited. The EBS/RBS approach does give us a 100% non-invasive solution which our SharePoint end users love but it doesn't give us everything.

Let's consider four different file system-based approaches...

STORE THE UNSTRUCTURED CONTENT ON A LOCAL FILE SYSTEM.

To be fair, this does deal with the SQL Server "bloat" issues that many customers seem to be concerned about; SQL backups will become more manageable and SQL will scale to support more centralized deployments. I've seen SQL instances shrink down to 20% of their original size when the BLOBs are externalized. It also can improve performance when handling large files because it relieves the IO bottlenecks to/from SQL Server. The problem is that you don't get any other benefits from externalizing the content. It just moves from SQL Server to the local file system...oh, and your restores just became a little more complicated because the file system backup and the SQL Backup need some 'synchronicity'.

¹ http://www.devx.com/MS_TechNet/Article/41772

² [EBS vs. RBS](#)

<http://www.sharepointgovernance.org/>

STORE THE UNSTRUCTURED CONTENT ON A VIRTUALIZED FILE SYSTEM AND PERFORM HARDWARE-LEVEL DE-DUPLICATION.

Imagine that you used the same method as outline above but that the file system that you wrote the BLOBs to was actually an intelligent storage device that was capable of de-duplicating those BLOBs. This means that if you have multiple copies of the same objects in SharePoint (common with Microsoft Office documents especially with versioning switched on) you will reduce your storage requirements. Looks like a nice little bonus saving in disk space which it is but be warned that all of your BLOBs are going to one tier of storage.

STORE THE UNSTRUCTURED CONTENT ON A VIRTUALIZED FILE SYSTEM AND PERFORM SOFTWARE-LEVEL HSM

OK, this time imagine that the file system that you wrote your BLOBs to was actually a file system emulator. It looks like an NFS or CIFS file system but really it is a piece of software that can take any files and store them behind the scenes on different storage devices. Now you are starting to see some value – not just dealing with the SQL Server bloat issues, you are doing real live hierarchical storage management. Bottom line is that content can be moved from high speed, high availability and high cost storage to less expensive and less performant devices. If the physical storage device does de-duplication then there's a bonus savings there too.

The downside is that typically you don't really have enough information from simple file system attributes to do anything other than the crudest management. You are restrained to "it is more than 6 months old" or "it is a PDF rendition and less than 20K"...not exactly 21st century policy enforcement – but the ROI can be attractive.

STORE THE UNSTRUCTURED CONTENT ON A VIRTUALIZED FILE SYSTEM AND PUSH OUT TO ON/OFF PREMISE CLOUD STORAGE

So, 6 months ago I would have said that this idea was pie in the sky...but it turns out that it is actually water vapor in the sky. Imagine that in the scenario painted above some of the tiers of storage were cloud storage devices – on-premise, off-premise or a combination of both. So the content might be stored on high speed local storage for the first 3 months, then for the next 6 months it moves to your on-premise cloud and then it moves to off-premise cloud storage.

You have similar limitations as I outline in the example above – you don't have too much metadata to work from but you do have the word "cloud" both in the product description and in your resume. That's why I'd do it anyway.

CONCLUSION

Using real and/or virtualized file systems with EBS or RBS gives you a solution that is transparent to your SharePoint users, relatively inexpensive to implement with a well defined and demonstrable ROI. However, it is very limited, it really only supports HSM with very basic policy management. Companies try to milk an extra mile or so out of this technology by implementing solutions that use extended file system attributes but there's only so far you can take these solutions.

Note that considering writing BLOBs to the file system replacing the objects with shortcuts in SharePoint is the worst of both worlds because it gives you the limited benefits of just using a file system but the downsides of an invasive solution.

MOVING CONTENT OUT OF SQL SERVER – TO AN ARCHIVE

I believe that we are entering what I'll call the era of "Archive 2.0" when it comes to SharePoint content. Once SharePoint really took off we saw a flurry of archiving solutions hit the market but I class them as V1.0 type solutions. When you look at how invasive they are, the underlying technologies that they rely on and their policy management you realize that there's a lot of meat left on that bone, (lousy metaphor, I must be hungry).

Let's define an archive and look at the business problem it solves, consider how Archive 1.0 solutions address those problems and then maybe how Archive 2.0 might do a better job.

WHAT IS AN ARCHIVE AND WHY DOES SHAREPOINT CARE?

Classic archiving systems are typically focused on pulling fixed, infrequently accessed or "old" information out of production systems and managing them in a separate repository. Generally, they provide value through policy-based management of the content for storage optimization, compliance and eDiscovery. In SharePoint's case this might be content that is fixed in nature such as scanned images, equally it could be stuff that you have finished with; old versions of SharePoint content, content in abandoned SharePoint sites, etc. These content types may still be accessed after they have been archived but generally they do not represent your most active content.

The benefit of archiving is that you are leaving only active content in the production system which allows the systems to scale more effectively; it also gives you better storage and backup management. Archives are not just somewhere to let the content go to die but this is a real value too. Consider for example decommissioning a SharePoint site: if you have all of your discoverable content already stored in the archive you can decommission the sites knowing that if you have to refer back to the content you can either search for a specific item or restore the entire site from the archive.

These systems differ from the file system approach because they are interested in storing not just unstructured content on a day-forward basis. They want all of your content – structured and unstructured and they want to be able to specify exactly what they want and when. They are optimized for high rates of ingestion – typically tuned to be able to ingest the volumes of email content which typically dwarf all other data types. They will store your documents, calendars, Blogs, Wikis, site collections, site configuration and even custom metadata. As an added bonus, with many of these solutions you can often restore individual objects or even complete sites from the stored information instead of having to use SharePoint's own backup/restore tools.

Unlike using a file system, archives can store more metadata – context that allows you to understand what the objects are. This allows you to implement more intelligent policies for security, retention, classification, etc.

Many systems will support some form of basic 2-way synchronization model; for example, you might be able to run rules that dispose of content in the archive and the archive process will reach in to SharePoint and delete the corresponding objects. (Personally I find this idea scary, it reminds me of the challenges we have in automated Records Management disposition – do you really want to be the person that clicked on the 'Dispose All' button?) As scary as it seems, it is a necessity in many cases otherwise you'll just end up with thousands of sites full of thousands of unwanted documents. If eDiscovery is an issue for you then this is something that you might want to think about especially hard.

<http://www.sharepointgovernance.org/>

HOW IS IT DONE IN ARCHIVE 1.0?

OBJECTS ARE REMOVED FROM SHAREPOINT AND REPLACE WITH SHORTCUTS.

With unstructured content, these Archive 1.0 systems often **move** objects out of SharePoint and store them in the archive leaving behind shortcuts back to the archived objects. I've noticed that vendors come up with very creative names to disguise these technology travesties, you'll see them called links, stubs, pointers, proxies, placeholders, etc... These are fundamentally flawed because SharePoint does not have the concept of a native shortcut object. This means that many important SharePoint features may not survive the shortcutting process including Microsoft Office integrations, full text indexing, custom metadata, workflows...

To be fair, early on there weren't many options and shortcuts to archives are not as hideous as they are to ECM systems because archived data should not be terribly active; read-only access via a shortcut is less of a catastrophe than it would be with active content. In fact, if your policies are well defined then shortcuts are acceptable but I expect changes in SharePoint, SQL Server, CMIS and elsewhere to make this draconian approach redundant. (I'll have to chat to Dr. Pie about the CMIS one at EMC World this year.)

THE ARCHIVE IS A 'BLACK BOX'.

The archive is pretty much treated as a black box. In Archive 1.0 land this was seen as an advantage – you neither knew nor cared where the content was – it just wasn't in your production system. This seemed like a good thing but you are leaving critical business assets out of reach to anything other than maybe your eDiscovery process.

Access to the archived content is expected to be done from SharePoint not directly from the archive so you are not really able to re-use the archived content from elsewhere; to be fair, if your policies are implemented correctly then this is exactly what you want because the archived content is not going to be re-used anyway but policies are never 100% water tight. Typically you can search the archive and might be able to apply localized policies to objects to support activities like eDiscovery. You'll see that this area forms one of the biggest differentiators between storing content in an archive and storing it in your ECM system, (next posting)

HOW MIGHT ARCHIVE 2.0 IMPROVE THINGS?

It is not that Archive 2.0 is going to change the basic premise of what an archive does; it is just that it will be more sophisticated, less invasive and more integrated in to other parts of your business. This makes it more cost effective because you'll be able to leverage what it does across more systems.

NO SHORTCUTS.

With Archive 1.0 your SharePoint content is either in SharePoint or in the Archive – in the latter case it is replaced by a shortcut. In Archive 2.0 I expect to see a more sophisticated and granular approach to how and where content is managed. For example it might start in SharePoint as a native object for a while then get virtualized using RBS. Later in its lifecycle it might get replaced with a shortcut then the entire SharePoint site might be deleted leaving the objects available only to the archive application. Finally after the retention policies have expired the content might be deleted. During these phase changes the objects might be on local storage, move off premise, on to tape (real and virtual), etc. This more granular approach will provide a better end user experience, more cost effective storage utilization and a greater ability to leverage new storage paradigms as they appear.

AGGREGATION

2.0 Archives will aggregate data not just from multiple SharePoint sites but from SharePoint, email, file shares...you name it. The ability to aggregate across multiple systems does give you some real benefits, you can implement a single set of policies across the different systems' data, (when that makes sense - the rules that govern SharePoint content don't always make sense for email content). You can get object-level de-duplication across all systems which can save storage; also you can manage your centralized storage utilization better because you don't end up with multiple back-end storage solutions each with its own quota overhead. One area where aggregation does have a real cost savings is in eDiscovery – if all of your SharePoint sites and all of your email content are discoverable from within a single archive then you can realize fairly impressive gains – the single instance storage helps with this one too.

CONCLUSION

The likelihood is that at some time in the near future you will have to implement an archive for some part of your business. It might be Exchange that cracks first, you might decide to start harvesting that fine collection of crap on your network file servers or you might want to try to get control of your SharePoint content before you drown in it. You might be able to get away with using your ECM system as a back end – in some cases it will be the better option but IMHO the 80/20 rules kicks in here. It is likely that in time 80% of your content would be better off in a nice old archive and 20% should be in your ECM system (the actual number is probably 94/6 but it does not sound as good). An archive will typically be less expensive to administer and is more likely to scale over time as you grow your deployments so think hard about using an archive solution regardless of what else you are looking at.

MOVING CONTENT OUT OF SQL SERVER – TO AN ECM SYSTEM

In the context of SharePoint, it is not a trivial task to perform a low-level integration of SQL Server into an ECM system but the benefits can be significant. Let me start by saying that typically you might not want all of your SharePoint content to be stored in an ECM system but for those specific types of content where it makes sense this solution can reap significant rewards.

Let's recap what a reward is in ECM-land...a reward is something that either makes you money, saves you money or keeps you out of jail. In summary, we are looking for ways of re-purposing content created in SharePoint to make money, efficiency improvements to save money and gains in security and compliance to keep us out of jail. I'll use this model as a way of analyzing the pros of this pairing; later in the entry I'll discuss the weaknesses of this approach and how it compares to pairing to an archive.

HOW DOES IT WORK?

Technically, it works in a very similar fashion to the archiving solution described earlier – in some cases you might only be storing the unstructured content, in others you might also take the structured too (Blog entries, calendar items, tasks, etc.). The rationale for only taking the unstructured content is that ECM repositories are often just used as "Document Management" systems so just taking the documents can make sense. Also the documents constitute the majority of the storage overhead and often the most critical business assets.

SharePoint content is stored in the ECM system along with a copy of its metadata; inside the ECM system you have the document and its context from SharePoint. Once it is in the ECM system you can re-purpose it or just protect it based on the ECM's capabilities. You have to take care when re-purposing the content because SharePoint may ask for it back in the future and if you changed it or deleted it then SharePoint will fail.

<http://www.sharepointgovernance.org/>

The following illustrate the kinds of things that you might do with the content once it has been captured in the ECM system.

MAKING MONEY

The bottom line is that your SharePoint sites contain a lot of corporate assets – documents that represent the intellectual property that your clients pay for. If you are able to aggregate all of that valuable content in to a single location and then index and categorize it you can re-purpose and reuse it. If you don't have to start from scratch each time you want to deliver data to clients then you are able to decrease the cost of developing collateral and increase profits.

It should become clear that the more compelling reasons to pair SharePoint to a traditional ECM system are based on operational efficiency and compliance but I do talk to a lot of customers who view re-purpose and reuse as a compelling win for them.

SAVING MONEY

OPERATIONAL EFFICIENCY

The file system pairing and archive pairing entries both focused on how to better manage BLOBs once they have been externalized from SQL Server. Depending on your ECM system of choice, all of these efficiencies should be available to you by using an ECM system as your aggregated repository. Tiered storage, de-duplication, SQL Server scalability, etc.

INTEGRATION IN TO EXISTING SYSTEMS

SharePoint systems tend to be a little isolated in corporate environments. It is not Microsoft's strategy to integrate tightly in to the non-Windows systems within your organization but it is fair to assume that your organization has data and processes outside of the SharePoint environment. If your SharePoint content is available from your ECM system then you can make use of it from within any system that already integrates in to your ECM systems.

In this model the ECM system becomes the center of your information infrastructure and SharePoint becomes both a contributor and consumer of information just like your other systems. You can start to manage the SharePoint unstructured content in the same way that you manage other content types – directly created ECM content, scanned images, formal records, physical records, web content, etc.

DISTRIBUTED CONTENT

Most ECM systems are able to distribute content geographically and then serve it up to users from the closest available location. This can be critical when bandwidth is an issue or content sizes are large. If SharePoint content is stored in your ECM system and then pushed out to remote locations it can be consumed from the local cache...in theory anyway!

LONG TERM ARCHIVING PLATFORM

Even SharePoint content deserves dignity in its old age. If you have specific content that needs to be retained you can keep it in SharePoint and keep the entire SharePoint stack running for the life of your retention period or you

<http://www.sharepointgovernance.org/>

can make use of the existing long term archiving policies in your ECM system. In this case you retain the content in the ECM system and decommission the source SharePoint site.

Although you are probably not thinking about this yet I believe that this is one of the most significant wins – second only to tiered storage management in absolute costs savings,

MANAGEMENT EFFICIENCY

You are already managing storage, compliance policies, disposition, holds, workflows, transformations, web publishing, long term archives, etc. in your ECM system. Rather than duplicating this effort in SharePoint you can consolidate the effort in to the ECM systems at little or no extra cost.

STAYING OUT OF JAIL

Many companies view their traditional ECM system as being the ‘system of record’ and for that reason alone they want to get critical SharePoint content in to the ECM system. Once the ECM system has the content you can leverage existing retention, formal record, data protection and disposition policies. You can also have common audit reporting on that content which can greatly aid in proving that content has been adequately protected.

I’m slightly more skeptical about eDiscovery support from the ECM system alone. In theory this makes sense – you have all of the critical SharePoint information in one place why not discover against it? In reality you will probably need to mine your content directly from SharePoint if it is still active. This is for two reasons, firstly you may not have absolutely all of the unstructured content in the ECM system and until you mine the content you don’t know what you don’t have. Secondly, you may not have access to all of the context of the object – you might know its metadata but do you know what its relationship was to the rest of the SharePoint content?

SO, WHAT’S THE DIFFERENCE BETWEEN THIS AND CASE #2

This is a very common question and I think that it can be hard to draw a completely clean line between them. The truth is that there’s a lot of similarity between pairing SharePoint to an archive and to an ECM system. In fact it looks like the ECM solution fully encompasses the benefits of the archive but in reality it is more of a Venn diagram.

Let’s start with what the ECM solution can do that the Archive might not...

- Re-use – typically an archive is not built to support the re-purpose and reuse use cases. The archive is treated more like a black box with only the archive admin processes, eDiscovery and SharePoint accessing the content.
- Center of existing processes – Many companies have invested heavily in building ILM processes around their ECM systems but not around their archives. They may have some retention and disposition policies in place in the archive but it would be rare to see something like full blown workflow.
- Better security and compliance – ECM systems will provide a more comprehensive set of data protection and compliance capabilities including fully certified formal records management.
- Bridge systems – Archives tend to ingest content and then manage it in a closed environment whereas ECM systems ingest content and then make it available to other systems to utilize, this allows you to use the ECM system as a bridge between SharePoint and your other systems.

Now consider when an archive might be optimal...

<http://www.sharepointgovernance.org/>

- Typically ECM systems are not optimized for rapid ingestion of large numbers of objects or for long term archiving/disposition management. I'm not saying that they cannot do this just that archives are typically designed to scale out to a higher degree.
- ECM solutions typically store unstructured content very well but may create an unacceptable overhead when storing structured content (calendar items, tasks, Blogs, etc.) Archives should be able to store structured content in a more efficient manner.
- ECM systems are usually extremely feature rich but that also translates in to cost and complexity. Archiving systems tend to be less expensive and have lower maintenance so if you don't need all of the benefits that ECM brings then an archiving pairing might make more sense.
- In both ECM and Archiving you can see a tendency towards data bloat but this tends to be higher in ECM systems- for example the ECM system might store multiple renditions not just multiple versions.

CONCLUSION

I think that it is fair to be confused with regards to the difference between pairing an archiving system or an ECM system to SharePoint. I think that it comes down to what you are going to do with the information once you have it in your repository. I'd proffer that most archive systems do not expect you to work actively with archived content. ECM systems however are designed to hold your most important electronic assets. If you consider what an ECM system allows you to do to your content it includes business process support, transformations, lifecycles not just data protection, compliance and disposition.

Very generally, if your unstructured content is high-value inactive content or active content then an ECM system is probably a better choice. If the unstructured content is fixed or is unlikely to be changed over time then an archive is probably better.

So, how would you classify your SharePoint content – active or fixed? Well it depends on what you get and when you take it. You'd typically expect to see an archive solution taking content towards the end of its life, (perhaps the object gets versioned, moved to a different folder or has not been accessed for 6 months. Certainly it would be appropriate to decommission entire site collections and dump them in to an archive. For more active content or content that has a very high value to the organization you might expect to see that being managed in an ECM system.

Can you use both in parallel? Absolutely, using lifecycle management you could move content between backend repositories when it makes sense – financial, regulatory and performance related.

<http://www.sharepointgovernance.org/>

MOVING CONTENT OUT OF SQL SERVER – DON'T PUT IT THERE IN THE FIRST PLACE.

Bottom line, if storing specific content in SharePoint is causing you pain then you might want to consider putting it directly in to your ECM system and then accessing it from SharePoint using Web Parts. With this approach you are using SharePoint as a portal and directly accessing the ECM's content and processes. It is not without issues – specifically SharePoint does not know anything about the objects in the ECM system, you cannot use any of SharePoint's native capabilities to control or interact with these objects. I've written plenty of other entries about this and other similar pairings but included it here for completeness.

This approach is especially effective for getting access to content and processes that already exist within your existing ECM systems. Also, it can be combined with the ECM back end concept to allow you to interact directly with content that has been externalized in to the ECM system – this is more complicated that it sounds so I'll try to expand on it in the near future.

Rumor has it that SharePoint 2010 will support CMIS...using CMIS to connect a SharePoint front end to a ECM backend looks promising and I'm sure that I'll be writing plenty of other articles about that one as we learn more.

MOVING CONTENT OUT OF SQL SERVER - CONCLUSION

Not surprisingly, the conclusion is that there are a number of different options when it comes to how to externalize content from SQL Server, the more functionality you want the more complex and expensive the solutions become...no surprise there then, I guess if the cheapest solution gave you the most functionality then I'd be wasting my time writing this.

In summary, the options are...

1. Dump the unstructured content on to the local file system from SQL Server - this gets the content out of SharePoint and relieves any SQL Server bloat related issues but you do not get any additional value out of the practice.
2. Externalize all content types and store them in an archive alongside your other archived content - as well as solving the SQL bloat issues this adds significant value with common policy management, long term archiving and retention/disposition/litigation support on the archived content.
3. Store unstructured content in your ECM system alongside your existing content – fixes SQL bloat; provides policy enforcement, long term archiving and compliance management plus the ability to re-purpose and reuse the content. Also gives you a natural integration point in to other enterprise information systems
4. Leave the content in the ECM system and use SharePoint as a portal – excellent solution for accessing existing ECM content and processes but doesn't really leverage SharePoint's native capabilities.

An interesting question is, "How many of these will still exist in 5 years time?" I'll stick my neck out here and suggest that only the first will no longer be around in 2014. My rationale is that #1 solves an issue caused by a questionable architectural decision made by Microsoft, if they decided to fix that then #1 would no longer be relevant. Options 2 and 3 actually add significant value by allowing you to bring SharePoint-created content in to your enterprise's other information systems. If SharePoint is going to be the dominant solution in the space then it needs to play nicely with other solutions in your company; it becomes another ubiquitous data source/consumer. #4 will hang around simply because SharePoint is a good portal choice not because it is a good portal platform but because it is pervasive and familiar to end users.