# What We Have Learned About Secondary Storage File Migration

January, 2014

Using the file system interface to provide access to slower but cheaper secondary storage devices has been, to put it mildly, a complete disaster. Dozens of companies have tried this. Thousands of man-hours of development and testing have been poured into it. Hundreds of (former) customers tried it and were very upset. Millions of dollars of investment has been squandered. Dozens of companies have gone broke from it. It has been attempted in the same manner over and over for at least the last 30 years. Exposing slow devices as a file system simply does not work in the general case. And yet, companies are STILL tempted to invest in it, yet again, with predictable results. They seem to think that if they download LTFS, shove it on a server, and hook up a tape library, the money will come rolling in.

The list below is probably one of the most expensive lists I've ever generated. These are the "lessons learned" from personally working on these types of projects at least 6 times and watching many other companies struggle with the same nightmares.

The real answer to this tiering puzzle is that the integration of slow secondary storage needs to be part of an information asset management system. The key value that is provided by such a system is that the business context, something not currently even stored in todays computers, is used to drive what gets migrated and when.

While the above paragraph may sound dictatorial, the problem of storage tiering has been solved by businesses long ago. Secretary's desks can only hold a limited amount of paper. What do they do with the rest of it? Well, first of all they know what all the paper is, what it is for, where it is within its lifecycle, etc. They collect up papers that have similar "business context", put them in a box, and send them to the warehouse. They keep a list of what is out there, when it should be disposed of, etc. Should anyone need that information, it can be ordered from the warehouse, brought back to their desk, and then accessed as needed. This method of managing "slow secondary storage" dates back to probably the point in history where there was more papyrus scrolls than could fit on someone's rock slab. It is very important for any digital tiering solution to heed the lessons learned and follow the paper pusher's lead. This is why I know this works.

## The Realities of Secondary Storage Tiering

- Automatic restore of a migrated file when accessed via the file system is a disaster. This can lead to operating system lockup, performance degradation measured in years, etc.
- Setting the migration "policy" based on each file or directory has been a disaster. There are simply way too many decisions that someone has to make in order to do it this way.
- Using "last access time" to know what files to migrate fails. Most file systems only update this time when they are updating other file metadata so it is inaccurate anyway. There is little correlation between last access time and the probability someone will need that information in the future.
- When deciding what to migrate out to slow media, you must be 100% accurate. 99.999% correct is a dismal failure.

- Keeping the first X bytes of the file on hard disk to limit media swaps is a stop gap at best. It is not sufficient to prevent lockup. This was put in for the Unix file(1) command to keep it from swapping media. There are plenty of other commands that will cause massive performance problems.
- It is surprisingly easy to constipate the kernel requiring a pull of the power plug to recover it. When all the kernel memory buffers are marked for use by the library, anything else that needs kernel memory is blocked. Things like the memory manager, the login processes, the mouse driver, etc. all the locked up.
- When the system constipates, users initially think they could have hardware problems. These look remarkably like RAM failures. I've personally seen servers sitting in corners perfectly fine but were labeled as broken due to the library software.
- Once a customer goes through a constipation event, they generally send the product packing. Any value they get from freeing up primary storage is lost when they have to deal with upset users.
- "Performance Problems" are not just annoying. They can be devastating. We call this the "Century Problem." It is very easy to setup a situation where even a small library will take 100 years to complete some process.
- Users get very upset if they try to do something (like click on a file) that has always run quickly, and it now takes more than the attention span of a guppy. They hit the key or click the mouse over and over again. Then they get upset and call IT. Not fun to watch.
- Current solutions that are still alive require limiting access to the entire secondary storage device to a single application, and often to a single user running that single application, is insufficient to maintain the profitability of the investment.
- Changing the name of the same solution (HSM, ILM, Tape NAS, Tier 3, etc.) doesn't change the fact that the architecture has never managed to gain traction.
- When the difference in performance between the two levels of storage exceeds about 10X, all the existing caching and tiering algorithms used for faster devices(such as flash to disk) fail miserably. These are simply "guesses" that, if wrong, are not that problematic. With slow libraries, these wrong guesses have killed the industry.
- When the information owners have a "business need" to access the data on a slow device, they expect to satisfy that request with a single round trip. They get very upset if they have to continue to go back to the device in order to get what they need. This one, by far, is the most difficult problem to fix.
- Users will be patient, do the right thing, tell you what they want you to do, etc., if you tell them what is about to happen, what is happening, how long things will take, what it will take to satisfy their request, etc.
- Often the actual request or question that needs to be answered is really a request for the business context of the information. This needs to be kept on disk so these questions can be answered quickly. (e.g. "Who was the sales rep for that contact we did with ACME manufacturing back in 2007?)
- Forcing users to identify or "classify" their data is nonsense. Won't happen. In fact, the entire "classification market" has failed to gain much traction because the very thing they are looking for, the business context is not stored in the computer.
- The larger the collection of information that is migrated at once, the better. (e.g. these are all the invoices from 2007, or these are all the files related to the initial DXi release)
- Presenting the "infinite capacity" feature results in users generating infinite amounts of data. It is amazing to storage people (but not economist!) that users will find so much data. I had one guy that saved off his entire hard drive every weekend to the library and never removed any previous copies.

 

- Users need to know the cost of migrating or accessing the storage. They will do the right thing if they are presented with the information and are given choices. This is much like FedEx. They know a document might need to be at a lawyer by 10:30 the next day but the box that contains the return of a failed hardware module can go slow and cheap.)
- Users need to be able to cancel requests, schedule them later, etc. This requires sufficient knowledge about what they want to do as well as some user interface.
- The better the data is organized BEFORE being migrated to secondary storage, the better. This is the whole idea behind information asset management systems.
- It is the business context (where the data is within its lifecycle) that defines when it is safe to migrate the information out to secondary storage. (E.g. All of last years paid invoices.)
- Today's operating systems have no way to even store the business context of files. Yes, this does mean that the fundamental core architecture, design, and implementation of the file interface in today's operating systems is flawed.
- Using existing file system metadata as a guide to deciding what data to be migrated is simply not enough to be accurate enough. We were running HSM on an HP-UX system. The OS is actually in a file called hpux. It was a large file that was never accessed. Hum. Let's migrate it out! I'll let you envision what happened the next time the system was booted!
- Scanning the content of the files in some hope to extract its business context has proven time and again to be nothing more than a guess. (The big data guys have yet to learn this lesson.)
- Retention or destruction policies of the data stored on secondary storage is NOT dictated by storage policies but defined by the business requirements. (Example: Must keep patient data 7 years after the death of the patient).
- To control the thrashing of the library's robot, it is vital to control the number of concurrent data operations. Depending upon the device type, it may require that the number of concurrent threads be limited to the number of active drives in the library. This is especially important for tape media.
- Media failures are bad if that is the only copy (don't do that!), drive failures are bad if the media is stuck in the drive and that was the last available drive, library failures are disastrous since ALL the data is now inaccessible. It is important to understand this since repairing a library is not a trip down to Best Buy to get a replacement. It may take days or weeks to get parts, a technician on site, etc. Keep this in mind when suggesting a single library solution.
- If there are problems with the storage device, it is important to be able to communicate with the user as to what is happening, when it might be fixed, who is in charge, etc.
- Operating system file managers (especially Windows) do an enormous amount of file accesses in the background. Things like creating thumbnails of files can cause major performance disasters for libraries.
- Background processes that are not carefully controlled can lock up the library and often times the server's operating system. Some examples include backup directory scanners, backup itself, antivirus scanners, data loss prevention scanners, full content indexers, etc.
- A user should not be able to type in a simple command or click their mouse and, without even knowing it, lock up the library and the server.
- Processes or users must not be able to use the library as a general purpose temp storage location. (E.g. an install package uses the library as a temp location because it has the most available disk space! Disaster!)
- LTFS is NOT sufficient to control access to a tape library in a way that will turn out to be anything other than a total disaster in the long run. Something or someone must limit its access or serialize requests to keep the library from thrashing.

 

- The file system interface, so commonly used to support libraries, was never designed for slow devices. These cause all sorts of problems (as this list indicates)
- Slow response times can ripple over to other systems as well. CIFS and NFS are really not designed to handle these devices. Error messages can get generated, programs that use these directories can lock up, and all sorts of other side effects can happen when libraries are misused.

## What Is Required To Provide General Purpose Storage Tiering

So, is secondary storage tiering relegated to the "good idea that never did work" pile? No, there is a solution that can work. The following are the key requirements for a solution that can solve this problem:

- End users cannot be directly exposed to the storage file system. They can still, however, use a file system to access files that have been migrated back to primary storage.

- The system must have the business context of the information in order to correctly know what to migrate to the secondary storage device and when it is safe to do it. This indicates the information must be controlled by an information asset management system in order to pull this off.

- The system must have some interactive method to communicate with the users when it comes to requests for information stored on the device.

- The simplest implementation is to leverage the existing solutions used for paper. This is a proven solution that can be mimicked using digital content rather than paper if there is going to be any hope of creating a general purpose solution that will meet the business requirements of customers.

## Summary

The bad news is that all previous attempts to create a general purpose secondary storage tiering system have failed. The good news is that these problems have already been solved in the paper world. The storage industry simply needs to leverage their solution. However, in order to do so requires a new approach to managing unstructured data.

For more information concerning this topic, Information Asset Management Systems, or secondary storage devices, contact Bruce Thompson at brucet@expeditefile.com