Implementing a Scalable Information Lifecycle Management Solution using IBM GPFS

August 2007

By Scott Fadden

GPFS Performance Engineer, IBM

"Data volumes are growing exponentially" and "there is a growing need to manage large collections of unstructured data" are statements that have been made for many years. The time has arrived and data centers containing millions, hundreds of millions or even billions of files are a reality. With these large numbers of files, automated management of this information is essential. The process of managing the data from where it is placed when it is created, where it moves in the storage hierarchy as its value changes, where its copied for disaster recovery or document retention to its eventual archival or deletion is often referred to as information lifecycle management (ILM).

Traditionally systems of this scale create performance challenges for ILM tools. This is often due to their lack of tight integration with the underlying file system(s) containing the data. Many solutions sit on top of various file system types or in front of multiple NAS devices introducing performance and consistency issues. The IBM General Parallel File System™ (GPFS™) has addressed these issues by tightly integrating the ILM functionality into a file system that can support billions of files shared across multiple heterogeneous systems running the AIX® and Linux® operating systems. This paper reviews various approaches to ILM, describes ILM tools available in GPFS 3.1.

To better understand the benefits of the GPFS tools for ILM it is useful to take a look at how other ILM tools work and the challenges they face. This paper examines two other approaches to ILM: a layered software solution that sits on top on existing file systems and a network protocol approach that uses NFS/CIFS communications.

Layered software solutions are products that provide ILM/HSM functions by managing data in local file systems. These products execute against local file systems and enable ILM functionality with existing sets of data providing a flexible solution for many applications. This type of solution migrates data between tiers of storage by moving the files from one file system to another. In this case a pointer is maintained either in the file system using a stub or through some database facility. Though flexible, these solutions face a scalability challenge in collecting the metadata on the files using standard file system interfaces. ILM operations are usually based on file metadata like last accessed time, "move all the files to SATA Storage that has not been accessed in the past 90 days" for example. Collecting this type of file information by scanning the entire file system is a resource intensive process.

Another approach is to use an NFS or CIFS based connection often implemented as part of a Network-Attached Storage (NAS) ILM solution. There are two basic implementations of NAS based ILM tools. First there is file management within or between NAS appliances. Secondly an appliance is placed in front of multiple file servers or NAS devices and redirects I/O requests.

The first approach is often not an ILM tool as defined earlier and is limited to a combination of storage snapshots, distributed caching and replicated data. Using these tools the data is mirrored or locally cached but not actually moved between tiers of storage.

The second approach is to have a device sit in front of multiple servers or NAS devices redirecting file requests. For ILM operations this second approach has a similar metadata collection model to the local file system approach in that it requires a file system scan. This scan can be run over multiple devices at once providing some scalability though this type of solution is loosely coupled with the backend devices. This means that file operations can occur on the back end servers without knowledge of the device controlling ILM operations. This may cause inconsistency in the file metadata on the ILM device and increases the complexity of backups.

Whether using a layered software solution or a distributed NAS approach to collect metadata information on the files, last access date for example, requires a directory scan of the entire namespace. With millions of files scan times often exceed operational windows making this approach impractical.

GPFS solves the scalability issue with a unique high performance scalable metadata engine. GPFS has a highly efficient, distributed metadata storage structure that can be read quickly in parallel. This metadata access mechanism greatly reduces the time it takes to process file migration rules allowing you to manage billions of files and scale-out the solution as data volumes grow. This patented metadata and ILM policy architecture within GPFS provides extreme performance scaling for file level information lifecycle management.

GPFS is designed to help you to achieve information lifecycle management efficiencies through powerful policy-driven automation and tiered storage management. Information lifecycle management (ILM) tools in GPFS allow you to define file level policies that determine where the data is physically stored regardless of its placement in the logical directory structure. Storage pools, filesets and user-defined policies provide the ability to match the cost of your storage resources to the value of your data.

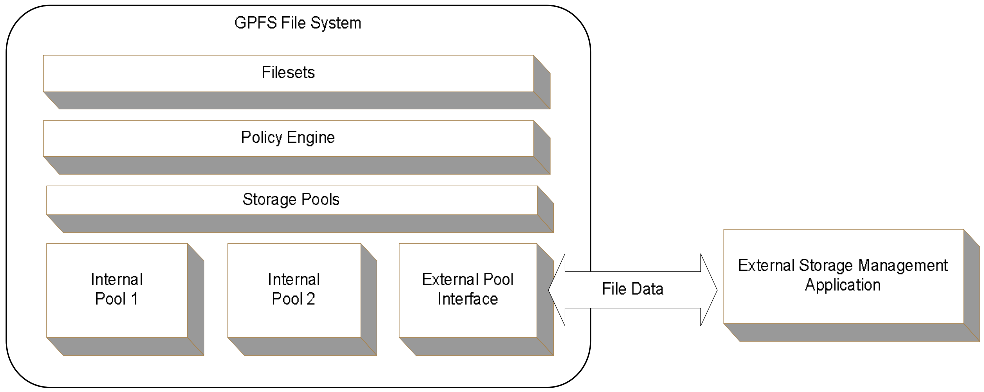

Figure 1: This picture shows the how the GPFS ILM tools interact.

Storage pools allow you to group multiple physical storage devices within a file system. Using storage pools you can create tiers of storage by grouping storage devices based on performance, locality or reliability characteristics. GPFS is supported on multiple industry standard platforms so many different types of storage devices can be added to a pool including Fibre Channel, iSCSI or InfiniBand™-attached disk and even storage internal to a node can be included. For example, one pool could be one or more enterprise class storage system(s) hosting high performance Fibre Channel disks and another pool might consist of numerous disk controllers hosting a large set of economical SATA disks. This allows you to perform complex operations (moving, mirroring, deleting) across multiple disparate physical storage devices, providing storage virtualization in a single management context.

Another ILM tool in GPFS is the fileset. A fileset is a logical sub-tree of the file system namespace and provides a way to partition the namespace into smaller, more manageable units. (i.e.: /departments/HR/records). Filesets provide an administrative boundary that can be used to set quotas and be specified in a policy to control initial data placement or data migration. Data in a single logical fileset can reside in one or across several storage pools. To which storage pool the file data is initially written and how it is migrated, mirrored, or deleted over its lifespan is based on a set of business rules in an administrator defined ILM policy.

There are two types of administrator defined ILM policies in GPFS: File placement and File management. File placement policies determine which storage pool files are initially placed in. File placement rules are determined by attributes known when a file is created such as file name, user name, group name or the fileset.

An example may include “place all files *.avi onto the platinum storage pool”, ‘place all files created by the CEO on the gold storage pool’, or ‘place all files in the fileset ‘development’ in the bronze pool’.

Policy rules syntax is based on SQL and supports multiple complex statements in a single rule allowing powerful ILM placement policies.

Once files exist in a file system, file management policies allow you to move, mirror or delete files. You can use file management policies to move data from one pool to another without changing the files location in the directory structure. They can be used to change the replication (mirroring) status at the file level, allowing fine grained control over the space used for data availability. You can use a policy that says: ‘replicate all files in \database\payroll which have the extension *.dat and are greater than 1 MB in size to storage pool #2’. In addition, file management policies allow you to prune the file system, deleting files as defined by policy rules. File management policies can use more attributes of a file than placement policies because once a file exists there is more known about the file. In addition to the file placement attributes you can now utilize attributes such as last access time, size of the file or a mix of user and file size. This may result in policies like: ‘delete all files named *.temp not accessed in 180 days’, ‘move all files that are larger than 2 GB to pool2’, or ‘migrate all files owned by Sally that are larger than 4 GB to the SATA storage pool’. Rules can include attributes not related to a single file like using the *threshold* option. You can create a rule that only moves files out of the high performance pool if it is more than 80% full. Multiple levels of rules can be applied because the complete policy rule set is evaluated for each file when the policy engine executes. This functionally can be used with any type of storage pool including the new external pools.

The GPFS ILM solution provides a single powerful interface to managing your data. The scalable architecture enables management efficiencies and the ability to virtualize the physical location of files across heterogeneous storage devices. For further details on storage pools, filesets or the policy engine in GPFS refer to the GPFS V3.1 documentation[1].

---

[1] GPFS V3.1 Documentation:
http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp

For more information on GPFS

GPFS Documentation

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp

White Paper: An Introduction to GPFS
http://www.ibm.com/systems/clusters/software/whitepapers/gpfs_intro.html

Configuration and Tuning GPFS for Digital Media Environments
http://www.redbooks.ibm.com/abstracts/sg246700.html?Open

IBM Redbooks Site:
http://www.redbooks.ibm.com

For more information on TSM

Linux device support by Tivoli® Storage Manager:

http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_Linux.html

AIX, HP, Sun, Windows device support by Tivoli Storage Manager:

http://www.ibm.com/software/sysmgmt/products/support/IBM_TSM_Supported_Devices_for_AIXHPSUNWIN.html

For more information on HPSS

http://www.hpss-collaboration.org

IBM Corporation
Marketing Communications
Systems Group
Route 100
Somers, New York 10589

CLW03004-USEN-01