Paper SAS569-2017

Shared File Systems: Determining the Best Choice for your Distributed SAS® Foundation Applications

Margaret Crevar, SAS Institute Inc., Cary, NC

ABSTRACT

If you are planning on deploying SAS® Grid Manager and SAS® Enterprise BI (or other distributed SAS® Foundation applications) with load balanced servers on multiple operating systems instances, , a shared file system is required. In order to determine the best shared file system choice for a given deployment, it is important to understand how the file system is used, the SAS® I/O workload characteristics performed on it, and the stressors that SAS Foundation applications produce on the file system. For the purposes of this paper, we use the term "shared file system" to mean both a clustered file system and shared file system, even though" shared" can denote a network file system and a distributed file system – not clustered.

INTRODUCTION

This paper examines the shared file systems that are most commonly used with SAS and reviews their strengths and weaknesses.

SAS GRID COMPUTING REQUIREMENTS FOR SHARED FILE SYSTEMS

Before we get into the reasons why a shared file system is needed for SAS® Grid Computing, let's briefly discuss the SAS I/O characteristics.

GENERAL SAS I/O CHARACTERISTICS

SAS Foundation creates a high volume of predominately large-block, sequential access I/O, generally at block sizes of 64K, 128K, or 256K, and the interactions with data storage are significantly different from typical interactive applications and RDBMSs. Here are some major points to understand (more details about the bullets below can be found in this paper):

- SAS tends to perform large sequential Reads and Writes.
- SAS does not pre-allocate storage when initializing or when performing Writes to a file.
- Reading and writing of data is done via the operating system's (OS) file cache.
- A large number of temporary files can be created during long-running SAS jobs in SAS temporary file space referred to as SAS WORK.
- SAS creates standard operating system (OS) files for its data store.
- When executing Writes, there is a single writer thread per SAS session.

There are some portions of the SAS software portfolio that can render IOPs oriented activity:

- Heavily indexed files traversed randomly
- SAS OLAP Cubes
- Data manipulation and modeling of some SAS vertical solutions

However, the above tend to be small components in most SAS shops, but they cannot be ignored and they need to be provisioned on separate physical file systems. In summary, the SAS workload can be

characterized as predominately large, sequential I/O requests with high volumes of data. It is very important to predetermine SAS usage patterns because this will guide optimal architecture and setup of the individual, underlying file systems and their respective, physical I/O provisioning.

SAS programs often read the same data multiple times. The program can perform logical reads from the local file cache much faster and more efficiently than re-reading it from the physical device. Operations that access small files tend to get a substantial performance benefit from the local file cache. Files that are larger than the size of the host system file cache cannot be cached in their entirety and might not be cached at all.

SAS does not pre-allocate storage when it initializes or when performing writes to a file. For example, SAS allocates a small amount of storage when it creates a file in an extent-based file system. As the file grows during a SAS task, SAS requests extents for the amount of storage needed.

SAS GRID COMPUTING REQUIREMENTS FOR SHARING DATA

In the SAS Grid Computing infrastructure, there might be several nodes that need access to shared data. These files all need to be accessible from any of the SAS Grid nodes via the same physical pathname. These areas include:

- Permanent SAS files – these include all SAS files (programs, catalogs, data sets, indexes, and so on) that need to persist between SAS sessions and can be shared between SAS users.
- SAS deployment and configuration files – these include all SAS binaries, configuration files, logs from SAS servers and SAS server repositories.
- SAS WORK - if the system uses SAS Check Point and Label Restart technology.

RETAINING FILE SYSTEM DATA IN MEMORY

Data read from storage (such as a solid state device or disk) is referred to as a physical read. When the data is retained in and read from the local file cache, the reads are called logical reads. Logical reads perform substantially better than physical reads (no hard faults are done to physical storage).

Most non-shared file systems will retain data in the local host system file cache until the memory is needed for some other purpose. In comparison, shared file systems implement a wide variety of rules that govern the retention of data in the local node's host system file cache. For example, some file systems allow tuning of the host system file cache, providing options for limiting either the amount of memory that can be used for file cache or the number of files kept in the local file cache.

MAINTENANCE OF FILE SYSTEM METADATA

File system metadata includes such information as lists of files in a directory, file attributes (permissions, creation date, and so on.) and other information about the physical data.

Various shared file systems differ in the maintenance of the file system metadata. Shared file systems are required to make information about file system metadata and file locking available to all systems participating in the shared file system. File system metadata is updated whenever a file is created, modified, deleted or extended, when a lock is obtained or dropped, and on some file systems, when a file is accessed.

The sizes of finished SAS files are not known as file creation begins. Therefore, storage space is not pre-allocated at the beginning of a file WRITE operation. As the file is written, additional file system blocks, or extents, are requested. Each extent request results in an update to file system metadata. In a shared file system, file system metadata changes are coordinated across systems. Because of this, more overhead is associated with shared file system metadata operations as compared to non-shared file systems.

Due to the differences in local file cache behavior and the additional overhead required to maintain file system metadata, programs that performed well with a non-shared file system might behave quite differently with a shared file system. To minimize overhead from the shared file system, a best practice is to maintain SAS WORK and UTILLOC directories in a non-shared file system and place only permanent data in a shared file system. A shared file system might be necessary for GRIDWORK with certain failover and high availability GRID architectures.

SAS WORK files are not shared and are considered to be private to the process that created the directory; there is no benefit for SAS WORK files to reside in a shared file system. SAS WORK files also tend to generate a higher number of file system metadata updates. This is because multiple copies of data segments might be created to support sort, tree, and other data manipulations by SAS PROCEDURES. This data amplification can be 3 – 4x the size of the incoming file and it can significantly increase the file creation, extension and deletion, and activity. This, in turn, increases metadata operations. In addition, a non-shared file system is more likely to retain SAS WORK files in the local file cache

IMPLICATIONS FOR PHYSICAL RESOURCES

Shared file systems require coordination between multiple host systems and they place more demand on the network and storage devices. This means that the network and storage devices must be provisioned appropriately. The storage subsystem might need to support both a higher number of I/O operations for journaling and coherency and a higher throughput rate when compared to the needs of a non-shared file system.

FILE SYSTEM CONFIGURATION

There are some file system options that are available on most UNIX or Windows systems that enable or disable recording when a file is accessed. Unless the workload specifically requires information about when a file was accessed or modified, disabling the metadata update with shared file systems reduces some of the load on the file system. On UNIX systems, mounting file systems with the option NOATIME= disables the recording of file access times. The mount option NOATIME= improves the performance of all workloads and is a recommended setting if accurate file access time is not required for correct program execution.

SHARED FILE SYSTEM CONSIDERATION

In this section, we discuss the shared file systems that we have the most experience with from either testing in-house at SAS or with a storage partner; and/or SAS customer deployments.

IBM SPECTRUM SCALE (FORMERLY CALLED GPFS™)

SAS and IBM® Spectrum Scale (version 3.5.0.17 or later) perform well together on IBM AIX 6 and higher, Red Hat® Enterprise Linux® (RHEL 6.5 and higher) and Microsoft® Windows operating systems. Both permanent and temporary SAS data files are managed by Spectrum Scale with excellent throughput and low overhead for file system metadata management.

Spectrum Scale is a very mature shared file system as it has been around since 1996. The maximum capacity limitations are in the 100s of PetaBytes. It can have up to 256 file systems per clustered file system, and there are no limitations to the number of nodes that can be attached to the clustered file system.

Notable Characteristics:

- Uses Spectrum Scale Paged Pool Space to replace Host System File Cache for SAS large block IO paging to/from storage. PPS is in host memory, managed by Spectrum Scale.
- Page management performance has been shown to be superior to local host file cache performance. It has performed better on heavier loaded systems where cache paging services were overwhelmed.
- Has separate volumes for file system metadata improves performance.
- Uses Native Encryption that is supported via IMB Security Key Lifecycle Manager (ISKLVM) version 2.5.01 or higher. There is no encryption for Windows. Encryption is data at rest.

Performance at Scale:

- Excellent when properly provisioned on storage with appropriate CPU resources and node memory for Spectrum Scale Paged Pool Space.
- Very hardy cluster. Node failures or cluster-wide issues are exceedingly rare. If found, they are generally caused by either improper tuning or poor storage resources/array-side provisioning for paged pool space.
- Our internal testing showed that Spectrum Scale 4.2.1 performed the same as XFS on the same storage with RHEL 7.2.

Lock Management:

- Token Management via metadata (one MetaNode per open file) to handle all POSIX locking requests.

Anecdotal Negative Field Experiences:

- Highly recommend IBM Support for implementation to avoid issues.
- Rare occurrences of bugs.
- Improper provisioning or tuning by customer can cause serious issues.
- Requires extra cores and memory to perform ideally.
- Expensive.

Further details about configuring IBMs Spectrum Scale on IBM Power servers for use with SAS can be found in this paper: http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102255

IBM GPFS™ File Placement Optimizer (GPFS™ FPO)

IBM introduced the File Placement Optimizer (FPO) functionality as an included feature with GPFS release 3.5 and later. GPFS FPO has the ability to aggregate storage across multiple network server devices (NSD) into one or more logical file systems across a distributed shared-nothing architecture. FPO uses all of the benefits of GPFS and also provides (1) a favorable licensing model and (2) the ability to deploy SAS Grid Manager in a shared-nothing architecture, reducing the need for expensive enterprise-class SAN infrastructure. GPFS FPO provides block-level replication at the file system level, allowing continuity of operations in light of a failed disk drive or a failed node. GPFS FPO extends the

flexibility of GPFS and has configuration options that provide fine-tune control over the level of replication. FPO also allows a level of control over how the initial and replica blocks are written across the cluster. For example, because data written to SAS WORK is read by the same node to which it's written, it has the ability to specify that the first block is written to this node, further benefitting SAS performance.

There are two additional performance-related concerns with FPO beyond those of non-FPO GPFS:

- Ensure that each node has a sufficient quantity of back-end storage devices to meet the SASrecommended IO throughput requirements, based on the number of cores and overhead needed to write replica blocks from other nodes in the environment.
- Ensure that sufficient connectivity exists so that replica blocks can be written to other nodes without throttling IO throughput.

SAS has found that, when coupled with properly provisioned hardware, GPFS FPO successfully meets the IO demands of a SAS Grid Manager implementation. For a more robust explanation of GPFS FPO functionality and SAS please see the 'SAS Grid Manager – A "Building Block" approach' white paper: http://support.sas.com/rnd/scalability/grid/SASGridManagerBuildingBlocks.pdf.

RED HAT GFS2

SAS and Red Hat GFS2 perform well together on the Red Hat® Enterprise Linux® (RHEL 6.5 and higher) operating system. Both permanent data and SAS temporary files are managed by GFS2 with excellent throughput and low overhead for file system metadata management.

GFS2 is a mature shared file system as it has been around since 1999. The theoretical maximum capacity limitations are 100 TeraBytes and 16 host nodes per clustered file system. Red Hat has no plans to increase their 16 node limitation for GFS2 shared file systems.

Notable Characteristics:

- Journaled File System
- Compatible with Red Hat multipath software
- Does not support native Red Hat encryption
- Free Red Hat support with RHEL 5.3 and later
- Reasonably priced

Performance at Scale:

- Generally, good, but can be variable based on load placed on the Distributed Lock Manager (DLM).
- DLM performance can be sensitive to a very high number of files per directory structure in a cluster, especially when the files have the extensive ACL rules applied (like Scalable Performance Data Server does). DLM performance also relies on network fabric connectivity bandwidth between host nodes and to storage.
- Many users simultaneously validating metadata-defined library structures via SAS applications (for example, populating the library panes of a SAS EG User Screen), coupled with a high

number of files (>20,000) containing extensive ACL rules, can place pressure on the DLM. This is especially notable with Scalable Performance Data Server.

- Until the spin lock issues mentioned below are fixed, the maximum number of cores in individual GRID nodes attached to a GFS2 cluster should be 8.
- Our internal testing showed that GFS2 has a 5% overhead when compared with XFS on the same storage.

Lock Management:

- Requires separate Red Hat DLM product.
- DLM requires a separate IO-based network for operation.

Anecdotal Negative Field Experiences:

- Rare occurrences of node failures causing the entire cluster to fail. This is generally caused by improper implementation of disaster recovery mechanisms (for example, live backups of SAS files in the cluster file system).
- Scalable Performance Data Server usage – High numbers of files with ACL rules can degrade DLM performance
- DLM does not scale at times. It is suspected that there might be lock request serialization issues when it gets very busy.
- A spin lock issue was recently discovered that greatly impacts the performance of GFS2. More details about this can be found in this Red Hat portal document: http://access.redhat.com/solutions/2786031

Further details about configuring Red Hat's GFS2 for use with SAS can be found in this paper: http://support.sas.com/resources/papers/proceedings11/342794_OptimizingSASonRHEL6and7.pdf.

VERITAS INFOSCALE

SAS and Veritas® Infoscale (version 7.1 or later) perform well together on IBMs AIX, HP-UX, Red Hat® Enterprise Linux®, and Oracle Solaris operating systems. Both permanent data and SAS temporary files are managed by Veritas with excellent throughput and low overhead for file system metadata management.

Veritas is a very mature shared file system as it has been around since 1991. The maximum capacity limitations are in 18 ExaBytes,

Notable Characteristics:

- Veritas can be installed as a local or a clustered file system.
- SAS testing has shown better performance with Veritas Volume Manager 256K LVM striping on FLASH, despite the underlying array preference of 64K transfers.
- Does not support transparent encryption or compression.

Performance at Scale:

- Very good on our internal test bed with a maximal 12+ GB/second IO throughput.
- Our internal testing showed that Infoscale has a 15% overhead when compared with XFS on the same storage.

Lock Management:

- Untested to date

Anecdotal Negative Field Experiences:

- None to date

Further details about configuring Veritas Infoscale for use with SAS can be found in this paper: http://support.sas.com/resources/papers/Performance_and_Tuning_Considerations_for_SAS_using_Veritas_Infoscale70.pdf.

QUANTUM STORNEXT®

SAS and StorNext® (version 4.3 or later) perform well together on a variety of operating systems, including Red Hat® Enterprise Linux®, AIX, HP-UX, Solaris, and Microsoft® Windows. StorNext is a file system and tiered data management family of products provided by Quantum Corporation.

StorNext has been around since 1999, but was migrated to Solaris/Linux in only 2006.

Notable Characteristics:

- Hardware/Software solution – StorNext File System (SNFS) and StorNext Storage Manager can be an appliance offering (latter) or software only (former).
- Clients attach to a shared-block SAN and can be heterogeneous.
- Fibre Channel Arbitrated Loop (FCAL) can be direct-attached to storage or via a Distributed LAN Controller (DLC). DLCs are reported to be faster than traditional networked-attached performance.

Performance at Scale:

- Good performance at-scale in the field.

Lock Management:

- No known issues at-scale.

Anecdotal Negative Field Experiences:

- None to date. This might be due to the lack of current SAS Grid users with this file system.

We have no paper on how to configure Quantum's StorNext for use with SAS.

INTEL® ENTERPRISE EDITION FOR LUSTRE®

SAS and Intel® Enterprise Edition for Lustre® (Version 2.0 or higher) performs well onRHEL. Intel Enterprise Edition for Lustre (we'll reference as 'Lustre' below) is a commercial version of Intel's parallel file system, Lustre.

Lustre is a shared file system that has been around since 1999, but did not become widely adopted until 2012. The maximum capacity limitation is 100s of PetaBytes per clustered file system.

Notable Characteristics:

- Journaled File System.
- Does not have the disaster recovery utilities that other clustered file systems have.
- Does not support transparent encryption.

Performance at Scale:

- Generally, good performance.

Lock Management:

- Lustre Distributed Lock Manager (LDLM).
- Cache coherent. Uses Metadata Servers (MDSs) and associated metadata targets (MDTs) of the inode that owns the file for metadata locks.
- File locks are managed by the object storage servers (OSSs) and their associated object storage targets (OSTs).

Anecdotal Negative Field Experiences:

- Support is mainly via support communities and chat rooms

Further details about configuring Intel Cloud Edition for Lustre for use with SAS can be found in this paper: http://support.sas.com/rnd/scalability/grid/SGMonAWS.pdf.

NFS

NFS is available on all operating systems for free. However, do not let the cost be the reason to consider using it. The way that NFS handles SAS typical IO makes it a less-desirable shared file system than others. In particular, the use of NFS for SAS WORK and UTILLOC is strongly discouraged due to the overwhelming number of unacceptable reliability and performance issues encountered both internally and at customer sites.

NFS is a shared file system that has been around for a very long time. The theoretical maximum capacity limitation is 18 ExaBytes per clustered file system.

Notable Characteristics:

- Cheap/free
- Easy to implement

Performance at Scale:

- Large-block READ performance is acceptable aside from network affectations.
- Large-block WRITE performance is poor due to close-to-open cache consistency issues in NFS3 and NFS4.
- Private LAN segments only. From a performance perspective, it is not recommended for WAN.
- File system performance is largely dependent on the underlying network fabric's available bandwidth.

Lock Management:

- Built-in.
- NFS, in particular NFS4, can exhibit problems with file available upon just-released file locks. Other nodes might not "see" these files immediately and this leads to an unreliable infrastructure.

Anecdotal Negative Field Experiences:

- Cache Coherency issues, metadata refresh on newly released locks, and some suspected serialization issues in IO queues render this as a negative choice for WRITE intensive SAS file systems. Details on these issues can be found in the paper: Best Practices for Configuring your IO Subsystem for SAS 9 Applications.
- DO NOT USE for SAS WORK/UTILLOC and other heavy WRITE permanent SAS data files systems.
- These comments hold true for all storage that uses NFS as their only file system. In particular, EMC Isilon, whose OneFS file system is NFSbased. If you plan to use this storage, please review this paper: http://support.sas.com/resources/papers/Advisory-Regarding-SAS-Grid-Manager-with-Isilon.pdf.

COMMON INTERNET FILE SYSTEM (CIFS)

Common Internet File System (CIFS) is the native shared file system provided with Windows operating systems. With recent patches, CIFS can be used for workloads with moderate levels of concurrency and works best for workloads that get limited benefit from the local file cache. The recommended configuration is to place SAS WORK directories on a non-CIFS file system and use CIFS to manage shared, permanent files (both SAS data files and reports/output).

With the release of the Windows 2008 operating system, many improvements were made to address performance issues and connectivity via 10-Gigabit Ethernet (GbE). These greatly improve the throughput and responsiveness of the CIFS file system. Resulting from changes made both to SAS Foundation 9.3 software and Windows Server 2008 R2 operating system, CIFS is functionally stable.

However, workload results showed that there was relatively poor retention of data in the local file cache. Workloads that reuse data from local file cache will not perform nearly as well with CIFS when compared to a local file system. The workload configuration had three systems running the Windows Server 2008 R2 operating system; one acting as a file server and two as clients – all connected via 10 GbE.

In order to function properly and perform optimally, the following registry settings, Microsoft patches, and environment variable settings for SAS® 9 software should be used:

- [HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\Executive]
- AdditionalDelayedWorkerThreads = 0x28
- AdditionalCriticalWorkerThreads = 0x28
- [HKLM\SYSTEM\CurrentControlSet\Services\lanmanworkstation\parameters] DirectoryCacheLifetime=0x0
- FileNotFoundCacheLifetime=0x0
- Hotfix referenced by http://support.microsoft.com/default.aspx?scid=kb;EN-US;974609
- Hotfix referenced by http://support.microsoft.com/kb/2646563
- Environment variable SAS_NO_RANDOM_ACCESS = 1
- Windows hotfix 2646563 created for SAS users disabled file read-ahead under some circumstances

When compared to other shared file systems that were measured on the Windows platform, GPFS had much better throughput, local file cache retention, and metadata management than CIFS. This translates to GPFS having both better scalability and the ability to serve a larger number of client systems than CIFS.

OTHER SHARED FILE SYSTEMS

There are several other shared file systems that SAS has determined should not be used with SAS Grid Manager. These include the following:

- Red Hat Gluster per Red Hat
- Red Hat CEPH per Red Hat
- Oracle CFS. We have heard of a customer using the new Oracle ACFS file system without issues.
- Parallel NFS

We will continue to review the performance of these file systems with SAS as our storage partners release newer versions.

CONCLUSION

This paper contains information gathered over several years from testing in-house, at SAS, with our storage partners and/or from SAS customer deployments using these shared file systems in their SAS infrastructure. The criteria for the testing is to see if the shared file system can achieve the minimum 125 MB/sec/core recommendation of SAS Foundation. Please note that other information on how to achieve that IO throughput suggestion can be found in the *Best Practices for Configuring IO Subsystems for SAS 9 Applications* paper and the tuning guidelines papers for storage arrays that can be found here.

We will update this paper as we work with new versions of these shared file systems and others. For notifications of these updates along with other hardware, storage and operating system tuning guidelines, please subscribe to the SAS Administration Community.

ACKNOWLEDGMENTS

Many thanks to Gretel Easter, Murali Srinivasan, Glenn Horton, Tony Brown and Jim Kuell for helping with this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Margaret Crevar
SAS Institute Inc
+1 919.623-5257
Margaret.Crevar@sas.com
http://www.sas.com