Paper SAS1929-2018

Optimizing Red Hat GFS2® on SAS® Grid

Tony Brown, SAS Institute Inc.

(Updated March 2019)

ABSTRACT

Red Hat® Global File System 2 (GFS2®) is one of the most popular shared file systems for use with SAS® Grid. This paper will serve as a "one-stop shop" for understanding GFS2 installation precepts as they relate to SAS® Grid operations and performance. It will also cover GFS2 cluster limitations, recommended LUN and file system construction and sizes, underlying hardware bandwidth and connection requirements, software placement on SAS® Grid nodes, and GFS2 tuning for Red Hat Enterprise Linux (RHEL®) operating systems, versions 6 and 7.

INTRODUCTION

Red Hat GFS2 is one of the most popular shared file systems in use for SAS Grid systems. When used within its stated limitations, and with the appropriately recommended system and hardware architectures, it is very stable and performant. Deploying a shared file system as part of a highly performant grid system requires attention to detail. This paper will provide the best-known advice, tuning, architecture, and practices to achieve high performance with Red Hat GFS2 on SAS Grid. We will begin with the necessity of an application and system architecture review, followed by operational parameters and limitations of GFS2, the operational overlay of GFS2 on SAS Grid application nodes, and finally tuning and provisioning requirements.

SAS GRID, RED HAT GFS2, AND PERFORMANCE ARCHITECTURE REVIEWS

Planning a SAS Grid involves many layers: the application, data infrastructures, host nodes, fabric interconnects, storage, and shared and non-shared file systems. Attention to detail in configuring each of these areas is vital when designing and implementing a performant system. SAS Grid Compute nodes require a minimum of 125 megabytes per second (MBps) of IO to each physical CPU core (not hyper-thread). SAS Grid Mid-Tier and Metadata nodes have a slightly lesser IO throughput minimum. They can suffice on 50 MBps per core. This can translate into a considerable IO requirement for a multi-node SAS Grid system.

Each SAS Grid implementation requires several layers of reviews. The first of these is the SAS Grid Application review that SAS conducts with the client. These meetings can span several days and cover all the application architecture aspects of installation, operation, and daily management/recovery of the SAS Grid (via the SAS Grid workshop). Please see your SAS Account Team for more information about, and to schedule, a SAS Grid Architecture Review for your SAS Grid system.

In addition to the SAS Grid Application review, you can use the following link to find further GFS2 cluster guidance and best practices from Red Hat: https://access.redhat.com/articles/2359891.

It is also possible to have a High Availability and Resilient Storage specialist review the core cluster configurations that you are seeking to implement. With the correct prerequisite information, they can offer you advice on the viability and appropriateness of your proposed approach. See the above link for more details on how they can help. Keep in mind that this is not a formalized service—but technical advice based on your documented plans.

A final performance review is highly advised after the SAS Grid Application is complete and any advice sought from Red Hat is followed. This review will use inputs from both the SAS Grid Application review as well as the Red Hat review. It will focus on the stated workload for the grid (often from the associated SAS® Enterprise Excellence Center (EEC) SAS Hardware Estimate document), examining the virtual and physical infrastructures beneath the grid to determine performance fitness. In this performance review,

every hardware aspect from host to fabric, to storage, along with any host or storage virtualization implementations, is reviewed to ensure that performance guidelines are met. The SAS Performance Lab in Platform Research and Development can assist with these performance reviews. Please work with your SAS Account Team to request this help.

GFS2 LIMITATIONS

All shared file systems have operating parameters and limitations. GFS2 is no exception. There are limits to node count in a single cluster, a cluster's physical size, and recommended file system and file sizes. In addition, there is at least one known issue with bandwidth performance for certain operations within a single node. These are all discussed in the following sections.

CLUSTER, FILE SYSTEM, AND FILE SIZE LIMITATIONS

A single Red Hat GFS2 cluster cannot span more than 16 defined machines (virtual or physical). This limitation is restricted to machines that require WRITE and UPDATE access to the cluster, and use lock management and coherency services. Machines like SAS Grid Metadata and Mid-Tier servers require shared file system space for the metadata server backup, logging, and so on. This can occur in one of two methods, but not both:

1. The SAS Grid Metadata and Mid-Tier servers might share network file space via an NFS mount (which is considered a non-performant connection). They don't need high IO performance. This method does not involve the GFS2 file system. This method will take servers that require shared space, but not high-bandwidth cluster coherency, out of the GFS2 cluster and lower the required node count. The GFS2 16-node machine count is a hard limit. Careful planning must be performed for both current grid installations and future growth to accommodate workloads that must exist in the same shared cluster.
2. The SAS Grid Metadata and Mid-Tier servers can be made into a cluster node in the GFS2 cluster. This should only be done if you can afford to count them in the cluster node count.

Each individual GFS2 file system cannot span more than 100 terabytes. You can create several file systems to accommodate capacity needs that exceed that. However, just because you can do something, doesn't always mean you should. Managing a file system that approaches 100 terabytes comes with issues of scale. These issues involve things like related inode size and traversal, and file system management (backup, recovery, replication, making the file system spread performant over large physical device pools, and so on). Because of these, we highly recommend a maximum size of around 20 terabytes for each file system. We have found easier management and better performance in this range. Just make sure that the file system size allows enough space for your largest individual jobs to run.

In addition to the file system size, we also recommend keeping individual files to smaller sizes (1 – 5 terabytes maximum, preferably a lot less). Many SAS operations will sequentially pass an entire file (for example, when indexed operations are not involved). The larger the file, the longer the intake time. By parallelizing large operations, using smaller files for each parallel segment, data intake becomes shorter and much more manageable (one of the primary precepts of SAS SPD Server).

KNOWN HOST NODE WRITE IO BANDWIDTH ISSUES

One of the first tasks after installing and configuring a SAS Grid is to conduct IO performance tests on the SAS Grid Compute Nodes to ensure that they have sufficient underlying IO bandwidth. The recommended minimum IO bandwidth is 125 MBps per host CPU core (not hyper-thread) from the SASWORK, UTILLOC, and SASDATA file systems, independently. For an average 8-core SAS Grid Node, this would result in 1 gigabyte per second throughput from each file system.

Red Hat GFS2 has a known WRITE bandwidth limitation, for a single host node running against a single file system, of approximately 800 to 900 MBps (total host node throughput). This limitation is caused by a bottleneck in the speed that a single GFS2 file system can place write requests into the page cache. Until this limitation is alleviated in future releases of GFS2, it is important to plan your SAS Grid layout to restrain core count to 8-12 cores per node. This bandwidth limitation was discovered by using an **artificial*

dd stack to flood the system with WRITE requests. Actual production SAS jobs are not likely to hit this limit as quickly as the artificial dd test. More details about this issue can be found in the following article: https://access.redhat.com/solutions/2786031.

Care must be paid to scaling up SAS Grid Compute Nodes (adding more cores to existing nodes) versus out (adding additional nodes with 8-12 cores or less). Your workload WRITE rate can help determine which of these is the better option for achieving a balance of not hitting the IO limitation while staying under a 16-node cluster.

**SASDATA permanent file systems typically READ and WRITE in ratios between 70/30 R/W to 80/20 R/W. They READ a lot but don't typically WRITE as much to the shared cluster. The dd IO test performs 100% WRITE to the cluster, tripping this GFS2 behavior in the test. Placing SASWORK, which has a 50/50 R/W mix, on local non-GFS2 file systems (for example, EXT4, XFS, and so on) can help greatly alleviate the probability of this occurrence.*

KNOWN SAS SPD SERVER EFFECTS ON GFS2 DISTRIBUTED LOCK MANAGER (DLM)

Several load issues have been discovered with shared file system lock management when using a very high number of ACL rules on a very large file set. This cropped up primarily with SAS SPD Server usage in a GFS2 shared cluster. Every time any SAS job is launched, it must perform ACL table reads to determine which files the user has permission for READ/WRITE access to. Hence, the size of the ACL table, and its corresponding performance are important. This occurs for each instantiation of a user session. When a very large set of files, with an accompanying large ACL rule table is read through the DLM, it can take a bit of time, causing slowness in the application launch. Although this effect will show up on batch jobs, it is most clearly illustrated to the SAS community using customer end-user applications like SAS® Enterprise Guide®, SAS® Studio, and so on. These applications are reading the ACL table to validate security rules before populating pick lists, library selection screens, and so on.

This has occurred in relatively extreme situations with SPD Server domains, where many more files than were recommended were placed in a single domain, with all numerous ACL rules attached. If you use SPD Server as part of your solution, or a very large number of individual files (for example, thousands in the same SPD Server domain) with ACL rules, you might want to consider using generic ACL rules wherever possible. A sample of how to do that follows. Generic ACLs have a name prefix followed by a wildcard to match to any table with the prefix. Here is an example:

```
libname foo sasspds 'public' host=<hostname> serv=<portno> user='john' ….;

/* add a generic ACL rule for all tables owned by John that begin with the name xyz */
        proc spdo lib=foo;
                add acl xyz/generic;

/* give users Bob and Fred read access to these tables */
        modify acl xyz/generic bob=(y,n,n,n) fred=(y,n,n,n);
```

Generic ACL rules can greatly reduce the number of individual ACL rules you have to create. This in turn reduces the size of the ACL table that the DLM has to process and significantly speeds up the application launch time!

There are several pointers and guides for creating performant SPD Server partition tables that are worth reading. They can be found here: https://support.sas.com/resources/papers/partitiontables.pdf.

GFS2 INSTALLATION ON SAS GRID

The following section will describe which SAS Grid nodes require which types of access to the shared GFS2 file system cluster (helping you to get an idea of node count and which nodes require cluster access). We will discuss the best practices for SASWORK and UTILLOC file system placement and conclude with the primary GFS2 services that must run on each SAS Grid Node that accesses the cluster.

GRID NODES REQUIRING ACCESS TO GFS2

There are several primary types of host nodes in a SAS Grid system, all of which can be mounted to GFS2 shared file system storage for SAS Grid operations. These include the Metadata, Mid-Tier, Control Node, GSUB, and Compute Servers. When attached to GFS2, each primary host node will be mounted for high bandwidth access to the shared storage (100 – 125 MBps per core), whether it needs high bandwidth or not (some nodes do not).

If you want to minimize the number of Grid nodes attached to GFS2 shared storage, the nodes that do not require high bandwidth performance can be attached to alternate shared storage. Table 1 below contains detailed information about shared storage requirements and access types for each type of Grid node. Nodes that can use non-GFS2 shared file space, on NFS-mounted network storage - only for low bandwidth (for example, metadata and mid-tier) are highlighted as well. Any SAS Grid host node that has file system mounts to GFS2 may also have separate NFS mounts to other shared storage. Red Hat does not recommend mounting directly into the GFS2 cluster via an NFS mount, however. The purpose of this table is to show you which nodes can use non-GFS2 shared storage in order to minimize node count against the 16-node GFS2 limitation in Grid deployments.

Node Type	Access to GFS2 required?	Shared Storage R/W Activity Required?	Access Bandwidth/Type of Shared Storage
Metadata Servers	No.	Non-GFS2 shared storage required. Activity: Reads from shared binaries, R/W to configuration and backup directories, and logging files.	Low bandwidth connection. Can be an NFS Mount to any shared network storage.
Mid-Tier Servers	No.	Non-GFS2 shared storage required, Reads and Writes for LSF configuration updates and logs, R/W to configuration and backup directories	Low bandwidth NFS connection to any shared network storage.
SAS GRID Control Nodes – Used as a Compute Node	Yes, GFS2 access is required	Reads and Writes to SAS data sets and files other than SAS files	High bandwidth connection 125 MBps per core.
SAS GRID Control Nodes – NOT used as a Compute Node	No.	Reads and Writes to grid control files	Low bandwidth NFS connection to any shared network storage.

SAS GSUB Server	No.	Reads and Writes to files shared with SAS Grid and to read shared LSF Directories (SAS M4 and below)	Low Bandwidth Connection to any shared network storage.
SAS Compute Servers	Yes, GFS2 access is required	Reads and Writes to GFS2	High Bandwidth Connection – 125 MBps per core.

Table 1**. Shared Storage Requirements and Access Types for SAS Grid Nodes**

In addition to the information in the table above, the location of the SAS binaries should also be taken into consideration. These are typically placed on a GFS2 mount in the shared file system instead of on non-shared storage like XFS. Doing this allows one copy of the binaries to be shared among all of the machines with access to that shared mount. This makes software installs, revisions, and updates much easier.

PLACEMENT OF SASWORK/UTILLOC

It is possible to place SASWORK and UTILLOC scratch working spaces on a GFS2 cluster in their own shared file system spaces. Some installations with monolithic array storage prefer to provision this way. This approach has several drawbacks:

- SASWORK and UTILLOC are not shared data repositories. Each SAS job has its own non-shared subdirectory in the file system. No distributed lock management is required. Hence, there is no requirement for a shared file system.
- Since all shared file system management has to pass through the coherent management of the GFS2 cluster, SASWORK and UTILLOC add an unnecessary burden to cluster management and services with no return benefit.
- SASWORK and UTILLOC are much more heavily used than SASDATA, adding a very large IO bandwidth burden to the storage interconnect.

The better alternative is to place SASWORK and UTILLOC on local file systems (such as EXT4 or XFS) on each SAS Grid Compute Node. This removes the cluster management overhead (shared file system overhead typically ranges from 5% to 15% in IO latency) on these file systems and alleviates added pressure on the storage adapters and the SAN/storage that houses the permanent SASDATA on the GFS2 Cluster. Neither SASWORK or nor UTILLOC require replication or backup, so they function very well on internal storage. If internal SAS Grid Compute Node storage devices cannot provide the bandwidth for SASWORK/UTILLOC, a small external storage array also works well.

DEPLOYING GFS2 SOFTWARE AND SERVICES ON SAS GRID NODES

GFS2 software and related services must be installed on each node that requires READ/WRITE access to the shared file system data. The software and service components that must be placed on each node are as follows:

- Corosync – Corosync manages cluster communications and quorum. The Corosync configuration file is set up on the Master Node to configure Corosync behavior for all nodes.
- Pacemaker – Pacemaker manages the cluster resources and configuration (for example, establishing Grid nodes, enabling the cluster, creating DLM and CLVMD resources and setting them in motion). Fencing, heartbeat, behavior, and so on, are tuned here. Pacemaker differs between RHEL6 and RHEL 7. See the Red Hat documentation for Pacemaker information: https://access.redhat.com/solutions/917813.
- DLM – Distributed Lock Manager. This service is instantiated and configured by Pacemaker on the Master Node. This runs on all nodes to manage the cluster metadata and the locking enforcement of files in the cluster.

- CLVMD – The Clustered Logical Volume Manager (LVM2) maintains a consistent view of the storage, including volume groups, logical volumes, and so on. The CLVM talks to the LVM2 and DLM to facilitate governance of the storage view and coherency.
- A host of ancillary services including GFS2 Utilities, Fencing Agents, UserID Management, LVM Cluster Pointers, and so on.

Please see the Red Hat Installation guides for instructions to deploy Red Had GFS2: https://access.redhat.com/solutions/45930.

These services have a large overhead and require ample headroom for memory and CPU in your system. Because of this, if you find the system use over 85% for extended periods of time, the host will be trying to service kernel and application memory needs, in addition to cluster management overhead. At this threshold, you will need to add additional resources. In addition, if the system is running VMWARE VSPHERE, it is advised to lower that threshold to 80% to cover cluster management, and VM management overhead.

Release used

The absolute minimum RHEL version for GFS2 installation is RHEL 6.4, but RHEL 6.8 is highly preferred. RHEL 7.1 and later is also supported for SAS Grid for RHEL 7 systems. Pacemaker on RHEL 6 is in technical preview in 6.4 and officially supported in RHEL 6.5. To note, RHEL 6.9 is the present release of RHEL 6.

Special Note about Stretch Clusters

Stretch Clusters are single clusters that span multiple physical sites. Given the geographic dispersion of IT facilities today, customers often try to implement shared file system clusters across sites. Red Hat does not support the use of CLVMD or GFS2 for Stretch Clusters! Please see the following for more information: https://access.redhat.com/articles/2906811.

Please do not attempt to implement a stretch cluster using GFS2.

RED HAT RESILIENT STORAGE – GFS2 CLUSTER ESSENTIAL NOTES AND TUNING

RED HAT GFS2 HOST NODE SETUP FOR PERFORMANCE

There is a long list of best practices for SAS running on Red Hat Linux, RHEL 6 and 7. These best practices have been tested, proven, and ratified over broad field usage. They are important to implement in physical systems as well as VM hosts (as closely as possible). The link to these best practices is http://support.sas.com/resources/papers/proceedings11/342794_OptimizingSASonRHEL6and7.pdf.

If the Red Hat Linux node is not set up according to the best practices in the above paper, serious performance issues can result. Note that some of these settings can conflict with hypervisor-set preferences when employed in virtual systems (VMware, vSAN, KVM, and so on). If you have doubts about what you are constructing, please work with your SAS Account Team to seek advice from the SAS Performance Lab in R&D, and your Red Hat vendor.

In addition to the RHEL guidance above, it is very important to set the SAS Grid host names in the /etc/hosts file using the full Cluster Interconnect Host Names. Do not use DNS name structures. This will help avoid DNS latency.

GFS2 and SAS Grid use the network interconnect between all SAS Grid Host Nodes on which they are installed for quorum, cluster communications, routing metadata, and so on. It is highly recommended that a private segment (not public) is used for this network interconnect. It should not be shared by other IT systems, and it should provide at least 10 gigabits bandwidth. SAS Grid and GFS2 will run connection checks (aka heartbeat or quorum checks) between nodes on set, short intervals, to ensure that all nodes are alive, well, and functioning. GFS2 runs its DLM and Corosync messaging and communications over

this interconnect, and it can get heavy. We have experienced slow network traffic or overloaded bandwidth interfering with these “heartbeat” and “quorum” checks, sometimes resulting in nodes being incorrectly fenced or sent into a failover situation. GFS2 DLM and Corosync can also cause lock and coherency management disruptions due to signal delay. A minimum of 10-gigabit network bandwidth is recommended to alleviate the issues caused by overloading the interconnect network fabric and switches.

The issues listed above can be very serious. In order to prevent these, it is a good idea to separate DATA IO and node communication traffic to different network interconnects.

DLM TUNING AND CONSIDERATIONS

Lock management for shared files in a GFS2 cluster is handled by the DLM. The DLM service runs on each GFS2 node. Other than increasing the size of the hash lookup tables it uses, there is no specific tuning that must be performed for the DLM. This increase is only needed in RHEL 6 and can be set in /etc/sysconfig/cman. The table setting should be raised from 1,024 to 16,384 and must be set before the cluster itself is mounted. This will help the DLM performance. Please see the following for more information: https://access.redhat.com/solutions/153163.

There is no DLM tuning, nor hash table size increase needed in RHEL 7.

STORAGE CONNECTION

There are multiple ways to provision GFS2 shared file system storage beneath a SAS Grid. Some of the primary methods are as follows:

- Host Bus Adapter attached via Fiber Channel to SAN (Direct Attached SAN).
- Network Interface Card attached via Network Switches and LAN Fabric to Network Attached Storage (Filer-based or Block-based NAS).
- Scale-Out Storage paradigms where the storage is a hardware cluster (GFS2 still sits on top) with storage devices (HDD or SSD) inside each SAS Grid Node Host (internal storage).
- Various Frankenstein builds of the above, mixed with Hyper-Converged Infrastructures (HCI), and so on. Learn from the movie—just because you can build a Frankenstein monster, doesn’t mean you should.

The minimum recommended IO throughput of 125 MBps per host CPU core will drive the number, model, type, and bandwidth rating of the cards, channels, switches, and fabric in the options above, in addition to the throughput of the back-end storage.

It is important to design storage to deliver the total bandwidth of all the SAS Grid Nodes with ample headroom. When using NAS, the Red Hat preferred IP network mode is Multicast – sending transmissions to multiple interested receivers on the network simultaneously (1:M or M:M).

CONCLUSION

Red Hat GFS2 is an excellent shared file system choice for high-performing SAS Grid systems. The performance, tuning, and architectural precepts offered in this paper can help ensure the best performance possible. Deploying high-performing underlying host, fabric, and storage hardware is crucial to success. In addition, following the RHEL 6 and 7 tuning recommendations, file system architecture, and setup, is key. It is also important to understand and stay within cluster node and size limits and recommended file size limits . Please be sure to arrange a SAS Grid Workshop as well as Red Hat and SAS Performance Lab architecture reviews before deploying a SAS Grid System with Red Hat GFS2. Closely following all of the information laid out in this paper will set you up for success.

ACKNOWLEDGMENTS

The author would like to acknowledge Bob Peterson, and Barry Marson, Shane Bradley, and Bob Peterson from Red Hat for their contributions to the GFS2 best practices for this paper. In addition, the advice and input from Guy Simpson, from the SAS SPDS team is greatly appreciated.

CONTACT INFORMATION

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

Tony Brown
SAS Institute Inc.
tony.brown@sas.com