Paper SAS1501-2015

Best Practices for Configuring Your I/O Subsystem for SAS®9 Applications

Tony Brown, Margaret Crevar, SAS Institute Inc., Cary, NC

ABSTRACT

The power of SAS® 9 applications allows information and knowledge creation from very large amounts of data. Analysis that used to consist of 10s–100s of gigabytes (GBs) of supporting data has rapidly grown into the 10s to 100s of terabytes (TBs). This data expansion has resulted in more and larger SAS® data stores. Setting up file systems to support these large volumes of data with adequate performance, as well as ensuring adequate storage space for the SAS temporary files, can be very challenging. Technology advancements in storage and system virtualization, flash storage, and hybrid storage management require continual updating of best practices to configure I/O subsystems. This paper presents updated best practices for configuring the I/O subsystem for your SAS®9 applications, ensuring adequate capacity, bandwidth, and performance for your SAS®9 workloads. We have found that very few storage systems work ideally with SAS with their "out-of-the-box" settings, so it is important to convey these general guidelines.

INTRODUCTION

Before we get into the best practices and guidelines, we need to begin with some basic tenets. This paper was written for Windows, Linux, and UNIX customers, to offer general guidelines to best set up I/O provisioning for their SAS® applications. This paper does not cover the z/OS environment or usage. It was written to help customers and their storage administrators understand how to provision well-performing file systems for SAS. The paper stresses resourcing to ensure SAS applications achieve the sustained I/O bandwidth required for timely completion of jobs.

Key to the success of configuring the storage for SAS applications is to fully understand their workload and characteristics. That understanding can assist in the configuration of the storage. By understanding the SAS workload you will be able to:

- Initially configure storage appropriately
- Ensure a healthy I/O throughput rate for your SAS applications
- Improve current SAS applications' performance
- Plan upgrades to storage before performance issues occur

GENERAL CHARACTERISTICS OF SAS SOFTWARE

Each SAS user starts their own heavy-weight SAS session for each SAS job or application they are running. With the SAS®9 Business Intelligence Architecture, there are also several SAS servers that are started to support the Java applications, but the tendency is for each active SAS user to have their own back-end independent SAS server or heavyweight process running. Each of these back-end sessions can be independently resource-intensive.

SAS data sets and associated files are built within the confines of the underlying operating system (OS) and are just "file system files." They can be managed by file management utilities that are a part of the OS (or might be a part of third-party products). This also means that file placement can be determined by the definition of directory structures within the OS.

Reading and writing of data is done via the OS file system cache. SAS does not use direct- I/O by default. NOTE: Since SAS uses the file system cache to read and write data, the maximum I/O throughput rate for a single OS instance can be restricted by how fast the system file cache can process the data.

GENERAL SAS I/O CHARACTERISTICS

The SAS I/O pattern is predominately large-block, sequential access, generally at block sizes of 64K, 128K, or 256K. There are some portions of the SAS software portfolio that can render IOPs oriented activity, such as:

- Heavily indexed files traversed randomly
- SAS OLAP Cubes
- Some SAS vertical solutions data manipulation and modeling

The above tends to be a small component in most SAS shops but cannot be ignored, and 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 since this will guide optimal architecture and setup of the individual underlying file systems and their respective physical I/O provisioning.

SAS does not preallocate storage when SAS initializes or when performing writes to a file. For example, in extent-based file systems, when SAS creates a file it allocates a small amount of storage, but as the file grows during a SAS task, SAS requests extents for the amount of storage needed.

The SAS Enterprise Excellence Center (EEC) and the SAS R&D Performance Lab recommend minimum I/O throughput metrics per SAS file system. SAS EEC sizing exercises are based on providing throughput per physical CPU core to service appropriate demand measures. These measures will range from 100 to 150 MB per second per physical core—per SAS file system type (SASDATA permanent data files, WORK temporary SAS files, and UTILLOC temporary SAS files during sorts and summarizations). Typical SAS processing (query, reporting, light analytics) is usually well sufficed by the 100 MB/sec/core rate, while advanced analytics and heavy statistical jobs might require up to 150 MB/sec/core. Please work with your account team to implement a free SAS EEC sizing to help you if you are not sure what you require.

FILE SYSTEM CONSIDERATIONS FOR SAS

This section offers general guidelines for setting up the file systems required by basic SAS applications. A specific SAS application or SAS solution might require more file systems than are listed below. Also, the exact physical configuration of the file systems will depend on the SAS usage and the underlying data model.

It is generally recommended that a minimum of three file system types be provisioned to support SAS. Depending on loads and sizes, there might need to be multiple instances of each of these. They are as follows:

- SASDATA stores persistent data for SAS exploitation and resulting SAS output files. It is heavily read from, and less heavily written back out. This file system need is typically protected with a RAID 5 or RAID 10 parity level. The parity level chosen is dictated by your corporate standards. This file system typically ranges from 80/20 READ/WRITE to 60/40 READ/WRITE. It is recommended to provide a minimum sustained I/O bandwidth of 100 MB/sec from storage to each SASDATA file system for normal SAS usage, and up to 150 for heavy statistics and analytics operations.
- WORK is the scratch working space for the SAS jobs. It is used to perform the working storage activity of single-threaded SAS procedures. Being nonpersistent space, it can be protected by as little as RAID 0 parity, but is safer with RAID 5 in case devices are lost. WORK is typically a heavily used 50/50 READ/WRITE file system. It is recommended to provide a minimum sustained I/O bandwidth of 100 MB/sec from storage to each WORK file system for normal SAS usage, and up to 150 for heavy statistics and analytics operations.
- UTILLOC is the same type of space for multi-threaded SAS procedures. UTILLOC by default is placed as a subdirectory underneath the WORK file system. We recommend splitting it out into its

own physical file system for performance. We also recommend placing it in RAID 5 parity protection. UTILLOC is typically a heavily used 50/50 READ/WRITE file system. It is recommended to provide a minimum sustained I/O bandwidth of 100 MB/sec from storage to each UTILLOC file system for normal SAS usage, and up to 150 for heavy statistics and analytics operations.

In addition to those file systems are the following:

- Root OS is the location for the OS and swap files
- SAS Software Depot could be placed on the OS file system
- Host system file swap space is recommended to be a minimum of 1.5x RAM

File extension is limited to the amount of physical space available within a file system. SAS data sets and individual partitions within a SAS Scalable Performance Data Server table do not span physical file systems!

If a given WORK or UTILLOC file system becomes overloaded and performs poorly, it is advisable to provision more resources underneath it. It might be necessary to create multiple physical file systems for WORK and UTILLOC, balancing SAS users or jobs between the different file systems for different SAS processes. This can help ensure workload balance across physical resources. More information about this subject can be found in the paper "ETL Performance Tuning Tips" starting on page 27 at: http://support.sas.com/documentation/whitepaper/technical/ETLperformance07.pdf.

We typically recommend considering provisioning additional space when the WORK or UTILLOC file systems begin to regularly reach over 80% full at peak operations.

We recommend the following file systems per host OS if your workload employs heavy sequential READ and WRITE loads:

- Solaris 10: ZFS
- AIX: JFS2
- HP-UX: JFS
- Linux RHEL: XFS
- Windows: NTFS

Where does NFS fit into the picture? To maintain file data consistency, NFS clients invalidate any data stored in the local file cache when it detects a change in a file system attribute. This significantly interrupts the performance of data operated on, in the local file cache, and is markedly pronounced on heavy, sequential WRITE activity. This NFS-specific behavior significantly punishes the SAS large-block, sequential WRITE-intensive performance. If your application workload employs heavy sequential WRITE activity (this is especially true of WORK and UTILLOC file systems), then we typically recommend that you do not employ NFS-mounted file systems. NFS also does not support file locking and can be problematic for using NFS as a shared file space for permanent SAS data.

When setting up the file systems, please make sure that READ-ahead and WRITE-behind or WRITE-through (this term differs on various hardware platforms, but what we want is for the SAS application to be given a signal that the WRITE has been committed to cache as opposed to disk) is enabled. NOTE: A word of warning with Microsoft Windows systems and processing large volumes of data—please review the following paper and SAS note:

- "Configuration and Tuning Guidelines for SAS®9 in Microsoft Windows Server 2008" at: http://support.sas.com/resources/papers/WindowsServer2008ConfigurationandTuning.pdf
- SAS note 39615 at: http://support.sas.com/kb/39/615.html

LOCAL VS CLUSTERED/SHARED FILE SYSTEMS

A local file system, including storage area network (SAN) storage, typically yields marginally better performance than a shared one. However, it is common in current enterprise data architectures to utilize multiple smaller servers and split application tasks, functions, and data across them (a scale-out approach). When a scale-out approach is employed, and common data must be shared across the server nodes, the use of a clustered file system (sometimes called a shared file system) is required. With a clustered file system, all the server nodes or OS instances have direct access to the SAS data as they would with a local file system. The clustered file system manages file locking and sharing for concurrent accesses across multiple host instances. Some examples of clustered files systems are:

- IBM General Parallel File System (GPFS)
- Red Hat Global File System 2 (GFS2)
- Quantum StorNext
- Veritas Cluster File System

It is crucial that the clustered file system provide the sustained I/O throughput required by your collective SAS applications. Again, for more details about tuning clustered or shared file systems to work best with SAS, please select the shared/clustered file systems link in SAS note 42197 at: http://support.sas.com/kb/42/197.html. OS and hardware tuning guidelines can also be found from the SAS note.

I/O PROVISIONING FOR PERFORMANCE

AGGREGATING I/O THROUGH STRIPING

For traditional spinning-disk systems, we have found that file systems striped across many smaller disks perform better with SAS than fewer larger disks. In other words, the more I/O spindles your file systems are striped across, the better. Striped file systems aggregate the throughput of each device in the stripe, yielding higher performance with each device added. Because each device has a limited throughput, and devices are getting much larger, it is not uncommon to have to provision more space than you need, to get the device bandwidth aggregation you need to meet SAS file system throughput requirements.

Flash storage can sometimes require over-provisioning in a similar capacity/bandwidth trade-off, requiring fallow cell space to avoid potential WRITE stalls due to garbage collection. Not all flash devices or device management is equal; pay attention to how your devices, array, or cluster handles preeminent cell garbage collection. Even with efficient management, it is wise to overprovision flash cell space by at least 20% above peak usage.

The primary goal in provisioning a file system is to ensure that SAS gets the sustained I/O bandwidth needed to complete the SAS jobs in the timeframe required by the SAS users. The storage type (direct-attached, network attached, appliance, spinning, disk, flash, or hybrid) does not matter, provided they yield the sustained I/O bandwidth for the core count as described above, for the SAS application or jobs.

FILE SYSTEM STRIPING AND SAS BUFSIZE

Stripe WRITE sizes are important considerations. Physical striping on disk, complemented by logical file system striping across logical unit numbers (LUNs) and logical volumes should default at a stripe size of 64K, unless dictated otherwise by your file size and demand load, or a fixed storage subsystem architecture. Some storage arrays default at 128K, others at 1MB. Ensure your physical and logical striping are consonant, or at least even multiples of each other. The most common stripe sizes employed for SAS are 64K, 128K, or 256K. For some large virtualized storage arrays it is 1 MB because that is all the array will support. If the majority of the SAS data sets consumed or produced are large (multiple gigabyte range), and the I/O access pattern is large-block, sequential access, then strongly consider setting the SAS BUFSIZE value equal to your storage stripe size or transfer size. Be aware that doing

this can slightly punish users of small files, or random-access patterns. There is a trade-off, so make this decision to benefit the largest and most important SAS workload set.

LUN CONSIDERATIONS

When executing WRITEs, SAS launches a single WRITE thread per LUN, but will start multiple READ threads for SAS threaded PROCs. Given the single WRITE thread per LUN, we typically like to see multiple LUNs support a single file system. We have generally found that employing eight or more (even numbers) allows better WRITE performance. The number and size of LUNs supporting a file system of given size is part of a complex equation involving stripe characteristics, device counts, and so forth. If at all possible, employ at least eight LUNs per file system.

ADAPTER PATHING AND MULTIPATHING

When external storage resources (e.g., SAN array) are arranged across multiple host adapters, it is imperative to employ multipathing software from your host OS or storage management system to evenly spread I/O workload across the adapters. Specific multipathing recommendations and notes are included in specific storage papers listed in the "Recommended Reading" section of this paper.

TESTING THROUGHPUT

It is wise to physically test your storage throughput to ensure it can sustain the desired I/O throughput before you install SAS. Please use suggestions in the following SAS notes:

- "Testing Throughput for your SAS®9 File Systems: UNIX and Linux Platforms" at: http://support.sas.com/kb/51/660.html
- "Testing Throughput for your SAS®9 File Systems: Microsoft Windows Platforms" at: http://support.sas.com/kb/51/659.html

The above are general guidelines and considerations to ensure adequate throughput for your SAS file systems. They are based on many years of experience across thousands of SAS customers. More specific guidelines regarding how to set up the file systems require a deeper understanding of the specific SAS applications processed, data characteristics, and collective demand load. The primary goal in provisioning I/O is to ensure that SAS gets the sustained bandwidth needed to complete the SAS jobs in the timeframe required by the SAS users. In some instances you might need to depart from the general guidelines above to best service your specific workload performance.

NEWER TECHNOLOGIES—FLASH, VIRTUALIZATION, AND CLOUD

FLASH

Flash is a broad term encompassing many solid-state forms of storage technology. It can encompass a simple USB plug-in drive, a 3.5" form-factor "disk" drive, a PCIe slotted flash card, and now a DIMM slotted flash card (fits in the DIMM memory slot of the motherboard and runs at DIMM speed). There are card models that can be used internal to the server on system boards, and cards "arrays" in a SAN arrangement. There are flash "appliances" that sit between SAN storage and the server, and act as I/O accelerators.

Underneath, are flash cells that persistently store data on charged-copper media. Their types are:

- Single-layer cells (SLC) are flash cells with a single charged copper layer arrangement for cell space. These tend to be the fastest and most expensive.
- Multi-layer cells (MLC) are flash cells with a multiple charged copper layer arrangement for cell space. These are cheaper in price, but are slightly slower than SLCs.

- Enhanced multi-layer cells (eMLC) are enhanced MLC cells that accommodate 20 – 30 thousand WRITE cycles instead of the typical 3 – 10 thousand of a typical MLC.

Above the cells, the management of I/O to and from flash cells is extremely important. Flash arrays offer varied methods to destage incoming, large, I/O blocks to flash. You must work with your flash vendor to understand their particular technology; there is some variety. Ancillary features offered or built into many flash arrays include data compression, data deduplication, and sometimes encryption. All of these features have a direct impact on performance—some very slight, others significant. For example, inline deduplication of data storage blocks has greater I/O impacts on some arrays than others. You must do your homework with your particular vendor to determine if you wish to have such services rendered on your flash array. We have tested numerous flash arrays and flash card assemblies. The results, along with OS, file system, and flash assembly tuning recommendations, can be found at: http://support.sas.com/kb/53/874.html.

In addition, flash can be mixed with traditional spinning disk devices in a hybrid storage arrangement.

Flash storage is much lauded for its extreme performance as compared to traditional spinning disks. For random READ/WRITE activities, this performance is most achieved. Large-block, sequential I/O, while performing faster than traditional storage, does not achieve the stellar numbers random I/O does. It has been our general experience testing flash storage that READs perform much faster, and WRITEs only slightly faster, than a good set of spinning disks on an optimized stripe. This holds true for many brands and types of flash storage. You cannot replace file system performance on a 100-disk stripe with optimized blocking, with a single flash drive and get better results. We have found a range of utilizing different numbers and types of flash devices to compare to a very large back-end disk array.

In multiple vender tests, we employed between 12 and 24 flash devices—based on vendor flash array models and offerings to a traditional SAN array with 140 disks striped together for optimal throughput. The flash cells were either supported “pseudo-striping” (you cannot stripe flash cells; it is a pseudo mechanism to emulate what a stripe does) to aggregate bandwidth, or depending on the type of flash array, random placement of destaged I/O blocks across flash cells (requiring no striping). In this arrangement, by far and large, the flash arrays came very close to the performance of the much larger disk array. We are essentially talking about replacing a 29 rack unit disk array with 4 – 8 rack units of flash assembly, or in device terms, 140 striped disk drives with 12 to 24 “striped” flash drives (depending on vendor offering). So that yields some idea of how flash can help a large workload.

SERVER VIRTUALIZATION

Server and computing resource virtualization has spread rapidly. Its goals to maximize hardware utilization while minimizing system administration are very attractive. Virtualization can take place within a single server chassis, across multiple rack nodes, and even across networks. Even with the best setups we often see a 3 – 7% I/O performance overhead, and a slight drop in core equivalency (physical cores to virtual cores) when running on virtualized systems. This can get worse or better depending on the physical resource allocation, and its colocation to the virtual environment. Our best experiences with server virtualization involve the following:

- Colocation of cores to associated memory is kept physically close to avoid nonuniform memory (NUMA) access. Avoid NUMA in VMware by:
- Disabling node interleaving from BIOS of vSphere host
- Using esxtop to monitor % local counter on memory screen—should be 100%
- Keeping physical memory to local socket capacity
- Not overcommitting I/O, CPU, or RAM (e.g., thinly provision resources for SAS workloads)

Please refer to our paper on server virtualization at: http://support.sas.com/resources/papers/MovingVirtuaVMware.pdf.

STORAGE VIRTUALIZATION

In addition to server virtualization, storage can be virtualized as well. This can exist within a single SAN infrastructure (e.g., EMC VMAX or VNX or IBM XIV storage) or as part of a virtualized storage assembly within a network or cloud. The goals of storage virtualization are similar to server virtualization—maximum utilization of resources, ease of management, reduced costs, and the convenience to tier data to higher or lower performing virtual pools based on performance needed. Storage virtualization is accomplished much the same as server virtualization: by uncoupling the definition and architecture of physical resources from their logical presentation. What is presented to the user is a simple set of resources. Underneath, those resources can physically exist in multiple places in shared pool arrangements, not be in the same physical place all the time, and be bits and pieces of resources instead of whole increments (like parts of a CPU).

When storage is virtualized, users see a logical file system, without necessarily knowing what is physically underneath it. In modern storage arrangement, they share space on large pools of shared storage, and can get moved around behind the scenes, possibly without physical space to back up the stated space they have in their file system. For example, your file system might have a definition of being 1 TB in size, but due to sharing space, and thin provisioning (only giving you the space when you actually use it), there might not be 1 TB of space there all the time.

In virtualized storage, you might be switched from one storage pool to the next, one type of storage to the next (e.g., slow SATA disk to faster SAS disk, or even to flash storage) without your knowledge. In addition, you might be sharing storage pools with underlying shared physical storage supporting radically different I/O patterns than what SAS engenders.

All of these issues are things to pay attention to. Below is a short list of best practices for virtualized storage and flash (it is hard to separate the two). More information can be found in the individual storage papers listed in the links in the “Recommended Reading” section.

Best practices for flash and virtualized storage:

- Do not overcommit shared or thinly provisioned storage for SAS file systems. When the usage gets high in an ad hoc SAS environment, it will result in a shortage or serious performance problems.
- Be very careful about using the automated tiering features in hybrid storage arrays (switching from faster to slower disk devices, or from disk to flash). The tiering algorithms typically make decisions too slowly for the very large files SAS uses to be migrated without negatively affecting performance. Some tiering is performed on a 24-hour cycle. Even that can cause SAN disruption with large migrations (and SAS migrations usually are).
- Do not place SAS file systems on slow rotation SATA disk drives, unless your workload has emphatically been proven to have a high random access pattern.
- If you are looking to pin something to flash, consider pinning WORK to flash. It typically has close to a 50/50 READ/WRITE ratio, so it will benefit from the much faster READ speeds.
- What about SAS permanent data? If you can afford it, it is great to have that in flash for the significant READ speeds. If you have limited flash in a hybrid array arrangement, WORK might benefit more, because highly shared data files from SASDATA pages might already be benefitting from being shared in the host system file cache.
- Be aware of automated inline data deduplication, compression, and encryption services in flash arrays; they are not all created equally, and they can have an effect on performance. If you plan on using these features, many are array-wide features, and cannot be selected or deselected for particular file systems or storage segments.
- Read the flash storage test and tuning papers in the “Recommended Reading" section of this document. They give testing results, performance information, and tuning best practices for many flash storage offerings.

CLOUD

Cloud usage by SAS customers is exploding. SAS internally hosts a very significant cloud space in our Solutions onDemand division, as one of our customer offerings. Whether you host your SAS cloud in a privately owned or subscribed space, there are things to be aware of to ensure desired performance.

Cloud spaces are an amalgamation of all of the above subjects of server and storage virtualization, advanced storage technologies, and shared file systems, along with piling on network virtualization, availability, backup and recovery, security, and a host of other layers. And it comes with all the best practices, caveats, and warnings of the same individual subjects above. Cloud resources are typically provided in "hardware cluster arrangements." Data is often shared across many virtual resources requiring shared or clustered file system arrangements. Compute and storage resources are often highly virtualized and are provisioned in defined resources clusters. Colocating physical resources to a logical cluster is extremely important—it helps in not introducing many of the issues described in virtualization above.

Meeting the primary goal of required throughput for your SAS workloads will likely cause you to make some nontraditional decisions when provisioning cloud space. You will likely engender more thick provisioning, or better said, guaranteed virtual resources—causing "thick provisioning" decisions. If you stick to your throughput requirements, and let those guide your logical and physical cloud architectural provisioning decisions, you will generally do well.

STORAGE CONSIDERATIONS FOR SAS

We have discussed the file systems used by SAS applications and how to configure them logically and physically to work best with SAS. We also have found that very few storage systems work ideally with SAS with their out-of-the-box settings. As we test new storage systems and technologies with SAS, we have put together white papers that list testing results, and host and storage tuning parameters found optimal for SAS usage. These papers can be found with the I/O subsystem and storage papers in "Recommended Reading" below.

SAS, like many workloads, can definitely benefit from the speed of flash storage. Not all flash is architected the same, created equally, or managed in the same fashion in array architectures. While prices are coming down, it is expensive enough that not all SAS shops can afford it for all SAS file systems. We have seen the best cost/performance trade-off when WORK is placed on appropriately provisioned flash, for shops that have limited budgets. The I/O subsystem and storage papers in "Recommended Reading" include the SSD and flash drives that we have tested with and the tuning guidelines for each.

CONCLUSION

It is strongly recommended to perform a detailed assessment regarding how SAS will function, the volumes of data that will be processed, analyzed, and manipulated, and the concurrent number of SAS sessions running before you implement I/O subsystems. Use this assessment to determine the I/O throughput rates needed. You should always work very closely with your storage administrator and your hardware representative to ensure your I/O subsystem can meet the I/O throughput rates required by your detailed assessment. The primary goal in provisioning a file system is to ensure that SAS gets the sustained I/O bandwidth needed to complete the SAS jobs in the timeframe required by the SAS users. It does not matter if internal drives, locally attached storage arrays, internal flash cards, external SAN or NAS arrays are used, provided they yield the sustained I/O bandwidth required by the SAS application or jobs.

In addition to this paper, which gives general information on setting up I/O subsystems, we are working with various storage vendors on additional white papers that discuss how to take these best practices and apply them to how to set up your storage arrays. The link to these papers is listed in "Recommended Reading" below.

RECOMMENDED READING

For more details on the above information, please refer to “Frequently Asked Questions Regarding Storage Configuration” at http://support.sas.com/resources/papers/proceedings10/FAQforStorageConfiguration.pdf

Recommended white papers on the following subjects can be found at http://support.sas.com/kb/42/197.html:

- General administration
- OS tuning
- I/O subsystem and storage
- Shared/clustered file systems
- Testing I/O throughput
- Performance monitoring and troubleshooting
- SAS GRID Environments

CONTACT INFORMATION

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

Tony Brown
SAS Institute Inc.
+1 (214)-977-3919 x 52155
Tony.brown@sas.com

Margaret Crevar
SAS Institute Inc.
+1 (919)-531-7095
Margaret.crevar@sas.com