Paper SAS1500-2015

Frequently Asked Questions Regarding Storage Configurations

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

ABSTRACT

The SAS® Global Forum paper "*Best Practices for Configuring Your I/O Subsystem for SAS®9 Applications*" provides general guidelines for configuring I/O subsystems for your SAS® applications. The paper reflects updated storage and virtualization technology. This companion paper ("*Frequently Asked Questions Regarding Storage Configurations*") is commensurately updated, including new storage technologies such as storage virtualization, storage tiers (including automated tier management), and flash storage. The subject matter is voluminous, so a frequently asked questions (FAQ) format is used. Our goal is to continually update this paper as additional field needs arise and technology dictates.

INTRODUCTION

This paper follows a frequently asked questions (FAQ) format for providing information, best practices, tuning, and provisioning tips for SAS application I/O performant storage provisioning. The material in this paper is similar to the material in "Best Practices for Configuring Your I/O Subsystem for SAS®9 Applications" mentioned earlier. This paper contains more in-depth information on questions about virtualization, storage tiers, and flash storage.

CHARACTERISTICS OF SAS SOFTWARE

WHAT ARE THE GENERAL CHARACTERISTICS OF SAS JOBS?

SAS users start their own heavyweight SAS session (heavyweight process) 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 his own back-end independent SAS server/heavyweight process running. Each of these back-end sessions can be independently resource intensive. These sessions operate in their own memory spaces. (SAS does not use shared memory.) They create independent scratch working spaces in the SASWORK and UTILLOC file systems. The sessions do not pre-allocate memory; they use memory incrementally up to the amount specified in the MEMSIZE option (globally or locally).

WHAT ARE THE CHARACTERISTICS OF SAS I/O?

SAS I/O pattern is predominately large-block and sequential access, generally at block sizes of 64K, 128K, or 256K. There are some portions of the SAS software portfolio that can render Input/Output Operations Per Second (IOPS) oriented activity, such as the following:

- heavily indexed files that are traversed randomly
- SAS OLAP cubes
- some SAS Vertical Solutions Data Manipulation and Modeling

These portions tend to be a small component in most SAS shops, but they cannot be ignored. 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 since this guides optimal architecture and setup of the individual underlying file systems and their respective physical I/O provisioning.

SAS does not pre-allocate storage when it initializes or when it performs 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.

Reading and writing of data is done via the operating system file system cache. SAS does not use direct I/O by default. **Note:** Because SAS uses the file system cache to read and write data, the maximum I/O throughput rate for a single operating system instance can be restricted by how fast the file system cache can process the data. There are several ways to avoid using the file cache with SAS. Starting with SAS 9.1 under Windows, you can use the SAS SGIO option. In SAS 9.2 under UNIX and Linux, commands are available on a SAS library level or a data-file level that instructs SAS to avoid using the file cache. These two methods for avoiding the file cache work very well in cases where you are working with very large files, but they do not work well when you repeatedly read the same small file. For more details about both of these methods, see the SAS Global Forum 2009 paper

WHAT ARE THE CHARACTERISTICS OF SAS FILES?

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 which are a part of the OS (or may 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.

WHAT ARE THE GENERAL PERFORMANCE REQUIREMENTS TO SUPPORT SAS?

Many software applications and databases have an IO orientation of IO’s Per Second (IOPs). Their performance is predicated on completing hundreds or thousands of small requests per second. SAS is different – its throughput orientation is large block sequential IO requests, and many of them. It is a large continuous request stream of large IO blocks.

The SAS Enterprise Excellence Center (EEC) and the SAS Research and Development 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 range from 100 to 150 megabytes per second/per physical core/per SAS file system type. The file system types are SASDATA (permanent data files), WORK (temporary SAS files), and UTILLOC (temporary SAS files during sorts and summarizations). Typical SAS processing of query/reporting/light analytics is usually satisfied 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 if you are not sure what you require.

FILE SYSTEM CONSIDERATIONS FOR SAS

WHAT SHOULD I KNOW ABOUT SAS FILE SYSTEMS?

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

- 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 is typically protected with a RAID 5 or RAID 10 parity level. The selected parity level 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 megabytes per second from storage to each SASDATA file system for normal SAS usage, and up to 150 megabytes for heavy statistics and analytics operations.
- WORK—SASWORK is the scratch working space for SAS jobs. It is used to perform the working storage activity of single-threaded SAS procedures. As a non-persistent space, it can be

protected by as little as RAID 0 parity, but it 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 megabytes per second from storage to each WORK file system for normal SAS usage, and up to 150 megabytes for heavy statistics and analytics operations.

- UTILLOC—is the same type of space for multithreaded SAS procedures. UTILLOC by default is placed as a subdirectory underneath the SASWORK 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 megabytes per second from storage to each UTILLOC file system for normal SAS usage, and up to 150 megabytes for heavy statistics and analytics operations.

In addition to the SASDATA, WORK, and UTILLOC file systems, we have the following:

- Root Operating System—location for the operating system and swap files.
- SAS Software Depot—can be placed on the operating system file system.
- Host System File Swap Space—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 an SPD Server table do not span physical file systems!

WHAT IF THE FILE SYSTEMS BECOME OVERTAXED?

If a given SASWORK or UTILLOC file system becomes overtaxed and performs poorly, it is advisable to provision more resources underneath it. It might be necessary to create multiple physical file systems for SASWORK 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. For more information about this subject, see the paper “ETL Performance Tuning Tips” at http://support.sas.com/resources/papers/ETLperformance07.pdf.

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

WHAT ARE JOURNALING FILE SYSTEMS AND WHAT SHOULD I PAY ATTENTION TO IF I USE THEM?

Journaling file systems (for example, JFS2 and EXT3ZFS) keep logs of the metadata of READ/WRITE/UPDATE/DELETE operations performed on files within a file system. This is done to protect the integrity of the data file system in case of a crash or failure. For file systems that have very heavy operations by many processes, it might be necessary to move the journaling log to a separate physical file system from the data to enhance performance.

WHAT KINDS OF FILE SYSTEMS DOES SAS SUPPORT?

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 Network File System (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 SAS' large-block, sequential WRITE-intensive performance. If your application workload employs heavy sequential WRITE activity (this is especially true of the SASWORK and UTILLOC file systems), then we typically recommend that you not employ NFS-mounted file systems. NFS also does not support file locking and can be problematic as a shared file space for permanent SASDATA.

HOW SHOULD FILE SYSTEMS BE TUNED FOR SAS?

When setting up the file systems, make sure that Read-Ahead and Write-Behind/Write-Through is enabled. 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. **Note:** A word of warning with Microsoft Windows systems and processing large volumes of data. Please review the following SAS Paper and Note:
support.sas.com/resources/papers/WindowsServer2008ConfigurationandTuning.pdf
support.sas.com/kb/39/615.html

WHAT SHARED OR CLUSTERED FILE SYSTEMS DOES SAS SUPPORT?

Here are some examples of recommended shared or clustered files systems for SAS:

IBM General Parallel File System (GPFS)

Red Hat 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. For more information about tuning shared or clustered file systems that work best with SAS, see the Shared/Clustered File Systems section at http://support.sas.com/kb/42/197.html. Operating system and hardware tuning guidelines can also be found at http://support.sas.com/kb/42/197.html.

I/O PROVISIONING FOR PERFORMANCE

WHAT ARE THE BEST PRACTICES CONCERNING TRADITIONAL DISK STORAGE?

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 (typically 20–25 megabytes per second sustained) and devices are getting much larger, it is not uncommon to provision more space than you need to get the device bandwidth aggregation to meet SAS file system throughput requirements. So you have to stripe across many drives, garnishing the ~25 megabytes per second per drive to get high throughput into the 100s of gigabytes to service today's large core hosts or collection of grid nodes.

We recommend using either Fiber Channel (FC) or Serial-Attached SCSI (SAS) drives for high spin rate and throughput performance. Slower-spinning Serial Advanced Technology Attachment (SATA) drives typically do not perform well with SAS large-block I/O throughput demands.

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. Regardless of the storage type provided (direct attached, network attached, appliance, spinning, disk, flash, or hybrid), it must yield the sustained I/O bandwidth for the core count as described earlier for the SAS application or jobs.

WHAT SIZE DOES SAS BLOCK AT, AND HOW SHOULD I STRIPE MY DISK SYSTEMS?

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 or demand load, or by a fixed storage subsystem architecture. Some storage arrays default at 128K, others at 1 MB. Ensure that your physical and logical striping is 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 can 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 and sequential access, 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.

WHAT ARE THE IMPORTANT LUN CONSIDERATIONS?

When executing WRITEs, SAS launches a single writer thread per LUN but will start multiple reader threads for SAS THREADED procedures. Given the single writer 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) LUNs allows better WRITE performance. The number and size of LUNs supporting a file system of a given size are part of a complex equation involving stripe characteristics, device counts, and so on. If at all possible, employ at least eight LUNs per file system.

Modern arrays extend LUN space without significant trouble. Older, non-virtualized storage required tearing up and rebuilding of LUNs to avoid creating new LUN space and concatenating the spaces, which lessened performance. Also be careful of LUN strips sharing underlying physical disks with NON-SAS applications, performance issues can result when differing workload patters collide at the disk /WRITE heads.

WHAT ABOUT PATHING/MULTI-PATHING FROM HOST TO STORAGE?

When external storage resources (e.g. SAN array) are arranged across multiple host adapters, it is imperative to employ multi-pathing software from your host OS or Storage Management System to evenly spread IO workload across the adapters. Specific multi-pathing recommendations and notes are included in specific storage papers listed in the Recommended Reading section of this paper.

TESTING THROUGHPUT

HOW CAN I DETERMINE WHETHER MY FILE SYSTEM DELIVERS ENOUGH THROUGHPUT PERFORMANCE FOR MY SERVERS?

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 the tools described in the following SAS Notes:

Testing Throughput for Your SAS®9 File Systems: UNIX and Linux Platforms
http://support.sas.com/kb/51/660.html

Testing Throughput for Your SAS®9 File Systems: Microsoft Windows Platforms
http://support.sas.com/kb/51/659.html

NEWER TECHNOLOGIES: FLASH, VIRTUALIZATION, AND CLOUD

WHAT IS 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, which 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 card "arrays" in a SAN arrangement. There are flash "appliances" that sit between SAN storage and the server, and act as I/O accelerators.

Flash storage can sometimes require overprovisioning 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, so pay attention to how your device/array/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.

Underneath are flash cells that persistently store data on charged-copper media:

Single Layer Cells (SLC)—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)—flash cells with a multiple charged-copper layer arrangement for cell space. These are less expensive in price but slightly slower than SLCs.

Enhanced Multi-Layer Cells (eMLC)—cells that accommodate between 20,000 and 30,000 WRITE cycles instead of the typical 3,000 to 10,000 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 de-stage 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 in in 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 whether 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.

HOW DOES FLASH PERFORM?

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, doesn't 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 can't replace file system performance on a 100-disk stripe with optimized blocking with a single flash drive and get better results. We have used a range of different numbers and types of flash devices to compare to a very large back-end disk array.

In multiple vendor 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 supported either by "pseudo-striping" to aggregate bandwidth (you can't stripe flash cells; it is a pseudo-mechanism to emulate what a stripe does) or, depending on the type of flash array, supported by random placement of de-staged I/O blocks across flash cells (requiring no striping). In this arrangement, by 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 a 4-to-8 rack unit of flash assembly or, in device terms, 140 striped disk drives with 12 to 24 "striped" flash drives (depending on the vendor offering). So that provides some idea of how flash can help a large workload.

WHAT ARE THE BEST PRACTICES FOR USING FLASH STORAGE?

With all the flash testing we have done, there are several best practices that we would like you to consider:

- If you are looking to pin something to flash, consider pinning SASWORK. It typically has close to a 50/50 READ/WRITE ratio, and it benefits from the much faster READ speeds.
- What about SAS permanent data? If you can afford it, it's great to have that in flash for the significant READ speeds. If you have limited flash in a hybrid array arrangement, SASWORK might be more beneficial 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 can affect 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. They give testing results, performance information, and tuning best practices for many flash storage offerings.
- To downsize capacity for performance, consider overprovisioning flash cell space if the brand or type of flash you use does not have exemplary preemptive garbage collection. Overprovisioning cells can create a surplus of WRITE-able cells to avoid garbage collection activities during WRITE operations. Some flash offerings do not require this as much as others.
- Consider using internal-to-the-host flash cards (for example, PCIe and DIMM) when the amount of space required fits within the card or number of card space limitations. Consider using this space for SASWORK or UTILLOC first, as neither requires invasive in-chassis backup.
- For more information, see the individual storage papers listed in the Recommended Reading section.

WHAT IS 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 between a 3% and 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.

WHAT ARE THE BEST PRACTICES FOR SERVER VIRTUALIZATION?

With all the SAS testing on a virtualized system, there are several best practices that we would like you to consider:

- Colocation of cores to associated memory is kept physically close to avoid non-uniform memory access (NUMA). Avoid NUMA in VMWARE by performing the following:

- Disable node interleaving from BIOS of vSphere Host.
- Use *esxtop* to monitor percentage of local counter on memory screen; should be 100%.
- Keep physical memory to local socket capacity.
- Do not overcommit I/O, CPU, or RAM (for example, thinly provisioned resources for SAS workloads).

For more information, see our paper on server virtualization at http://support.sas.com/resources/papers/MovingVirtualVMware.pdf

WHAT IS STORAGE VIRTUALIZATION?

In addition to server virtualization, storage can be virtualized as well. This can exist within a single SAN infrastructure (for example, EMC VMAX/VNX or IBM XIV storage) or as part of a virtualized storage assembly across 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 and not be in the same physical place all the time; they can 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 terabyte (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 and from one type of storage to the next (for example, 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 from what SAS engenders.

WHAT ARE THE BEST PRACTICES FOR VIRTUALIZED STORAGE?

With all the virtualized storage testing we have done, there are several best practices that we would like you to consider:

- Direct-attached storage is always typically the highest performing. Networked storage across local or wide area networks, or within a cloud can introduce potential performance issues if not managed carefully.

- Do not overcommit shared or thinly provisioned storage for SAS file systems. When the usage is high in an ad hoc SAS environment, it results in a shortage, serious performance problems, or both.

- Use large, thin pools across many or all of the devices available. This spreads the load across a large pool, aggregating bandwidth for throughput. Large pools must be monitored carefully for hotspots and remediation if portions of the pool become too “hot.”
- 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.

DOES SAS RUN ON THE CLOUD?

Yes, absolutely. 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 previous topics on 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 as the previously discussed topics.

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 provisioned in defined resource clusters. Co-locating physical resources to a logical cluster is extremely important; it helps in not introducing many of the issues described earlier in the virtualization sections.

Meeting the primary goal of required throughput for your SAS workloads likely causes you to make some nontraditional decisions when provisioning cloud space. You will likely engender more thick provisioning, or better said, guaranteed virtual resources that cause “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 answered key questions regarding configuring storage for performance for SAS usage. As we test new storage systems and technologies with SAS, we continually provide white papers that list testing results and host and storage tuning parameters found optimal for SAS usage. These papers can be found in I/O Subsystem and Storage and Storage listed in the Recommended Reading section.

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 and performance trade-off when SASWORK is placed on appropriately provisioned flash for shops that have limited budgets. The Recommended Reading papers include the SSD and flash drives that we have tested with and the tuning guidelines for each.

SAS can perform well on cloud and virtualized systems. Attention needs to be paid to the particulars of providing dedicated underlying resources and adequate throughput performance from those resources.

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. The information from this assessment will guide the I/O throughput rates needed and shape the storage configuration and architectures required.

Work closely with your corporate storage administrator, host, file system, and I/O subsystem vendor to iron out functional and performance requirements to provide detailed specifications, tuning, and implementation.

RECOMMENDED READING

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

SAS Administration

Operating System Tuning

I/O Subsystem Testing, Performance, and Tuning

Shared/Clustered File Systems

Testing I/O for SAS File Systems

General Performance Troubleshooting

SAS® GRID Environments

CONTACT INFORMATION

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

Tony Brown
100 SAS Campus Drive Cary,
NC 27513
SAS Institute Inc.
Tony.Brown@sas.com

Margaret Crevar
100 SAS Campus Drive Cary,
NC 27513
SAS Institute Inc.
Margaret.Crevar@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.