Paper SAS6761-2016

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

Tony Brown and Margaret Crevar, SAS Institute Inc.

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 to 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. This paper does not cover the z/OS environment or usage. It was written to offer general guidelines 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.

The I/O subsystem consists of the entire data path, which includes the following components:

- Back-end physical storage devices
- The storage subsystem
- Any storage subsystem connectivity ports
- Any network fabric or Fibre Channel attachment from the hosts to storage
- Any host connectivity cards such as network interface cards (NIC) or host bus adapters (HBA)
- The host system logical volume and physical file systems
- The host system file cache

The key to success in configuring the storage for SAS® Foundation applications is to fully understand the workload and characteristics of those applications. That understanding can assist in the configuration of the storage. By understanding the SAS workload, you will be able to do the following tasks:

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

CHARACTERISTICS OF SAS FOUNDATION SOFTWARE

Each SAS Foundation user starts a separate heavyweight SAS session for each SAS job or application the user is running. With the SAS®9 Business Intelligence Architecture, there are several SAS servers that support Java based applications. SAS user sessions on SAS Foundation compute servers each represent a back-end, independent SAS server or a 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 64KB or a multiple of 64KB (e.g. 128KB, 256KB, 512KB, 1MB). In SAS software, the following processes can render I/O per second (IOPs) oriented activity:

- Heavily indexed files traversed randomly
- SAS OLAP cubes
- Data manipulation and modeling done by some SAS vertical solutions

The above tends to be a small component in most SAS shops but it cannot be ignored and it needs 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 pre-allocate storage when it 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. 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 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 Foundation processing (query, reporting, and light analytics) usually performs well with 100 MB/sec/core I/O 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 SAS Foundation 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 is typically protected with a RAID 5 or RAID 10 parity level. The parity level chosen is specified by your corporate standards. This file system typically ranges from 80/20 READ/WRITE to 60/40 READ/WRITE. It is recommended that you 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 MB/sec 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 non-persistent 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 used to recommend splitting it out into its own physical file system for performance. For high-performing flash storage, and large, wide-striped disk pools, or “striped everything” disk storage, it is no longer necessary to split it out.   We recommend placing the UTILLOC file space in RAID 5 parity protection if spinning disk is used. UTILLOC is typically a heavily used 50/50 READ/WRITE file space. It is recommended that you provide a minimum sustained I/O bandwidth of 100 MB/sec from storage to each UTILLOC file space for normal SAS usage, and up to 150 MB/sec for heavy statistics and analytics operations.

In addition to those file systems, keep in the mind the following considerations:

- Root OS is the location for the OS and swap files.
- SAS Software Depot could be placed on the OS file system for a single host, or on a network drive for multiple host installs.
- Host system file swap space is recommended to be a minimum of 1.5x RAM.

File size extension is limited to the amount of physical space available within a file system. SAS data sets do not span physical file systems!

If a file system housing WORK or UTILLOC 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” (SAS 2006, p. 27).

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

We recommend the following local (non-clustered) file systems per host OS if your workload uses heavy sequential READ and WRITE loads:

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

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: For information about processing large volumes of data on Microsoft Windows systems, please review the following paper and SAS note:

- “Configuration and Tuning Guidelines for SAS®9 in Microsoft Windows Server 2008” (Crevar 2008)
- "Input/Output Performance in SAS® Is Degraded Due to Excessive Memory Usage on Windows" (SAS Note 39615)

NETWORK FILE SYSTEM (NFS) CACHE COHERENCY ISSUES

How do NFS based file systems fit into the picture? In order to ensure file data consistency, any file metadata in the local file cache is invalidated (for example, flushed) when an NFS client detects a change in a file system attribute. The next time the file is accessed, its metadata will be retrieved from the NFS server. This means that retention in the file cache might have much different behavior with an NFS file system when compared to other file systems. The file system storage devices and network must be provisioned to handle a larger demand when compared to either a local file system or a shared file system that uses a different strategy for cache coherency.

This “dropping” of the cached attributes causes metadata re-reads from the NFS server in order to re-obtain them. This in turn results in high process and I/O latency. In a SAS environment where there are many processes performing WRITE activity, this often introduces very noticeable application I/O slowness. This issue is persistent in both NFS3 and NFS4. It is exacerbated by frequent file metadata access in the heavily used SASWORK file system. This behavior is also more prevalent in WRITE operations than READ operations. For these reasons, we typically recommend that NFS and NFS based file systems (like EMC Isilon OneFS from EMC Corporation) be used for mostly READ predicated file systems like your permanent SAS data, and not used for heavy WRITE file systems like SASWORK or UTILLOC.

NFS FILE AVAILABILITY ON UPDATE

The NFS client maintains a cache of file and directory attributes. The default NFS settings associated with file closings and file metadata updates will not ensure that files created or modified on one system will be visible on another system within a minute of file creation or modification. The default settings might cause software to malfunction if multiple computer systems are accessing data that is created or modified on other computer systems. For example, a new SAS data set created by a SAS workspace server on System A might not be visible on System B within one minute of its creation.

In order to assure a consistent view of the file system, the file system mount option ACTIMEO= (attribute cache time-out) should be set to 0. This setting will increase the number of requests to the NFS server for directory and file attribute information. It will also ensure that the NFS client systems have a consistent view of the file system. File data modifications might not be visible on any NFS client system other than the one where the modifications are being made until an NFS commit is executed. Most NFS clients will issue a commit as part of closing a file. If multiple systems are reading files that can be modified, file system locking should be used. This is controlled by the SAS system option FILELOCKS=, the SAS library option FILELOCKS=, or the SAS LOCK statement.

Additional NFS mount options to reduce messaging traffic include:

- Noatime—disables updates to metadata timestamps on the file’s last access
- Nomtime—disables updates to metadata timestamps on the file’s last modification

These options are frequently used in SAS systems and can reduce some file system messaging traffic when used.

LOCAL VERSUS CLUSTERED/SHARED FILE SYSTEMS

A local file system, including storage area network (SAN) storage, typically yields marginally better performance than shared storage. However, it is common in current enterprise data architectures to use multiple smaller servers and split application tasks, functions, and data across them (a scale-out approach). When a scale-out approach is used, 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 the same 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. Here are some commonly used clustered files systems for SAS Foundation:

- 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 that is required by your collective SAS applications. 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. OS and hardware tuning guidelines can also be found in 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 perform with SAS. 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 device capacities are getting much larger, it is not uncommon to have to provision more physical space than you need to get the device bandwidth aggregation needed 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, arrays, or clusters handle pre-eminent cell garbage collection. Even with efficient management, it is wise to overprovision flash cell space by at least 20% above peak usage. Most modern all-flash arrays build this overage into their capacity.

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 that is 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

Block transfer sizes from host to storage are important considerations. There are several aspects to determining which SAS BUFSIZE (which governs your SAS block transfer size) you should employ. The default BUFSIZE in SAS®9 is 64KB. You have to consider the entire I/O chain from the Logical Volume Manager (LVM) and file system, down to the destination storage system. File systems to support SAS are typically built with an LVM, using underlying LUNs for construction. This LVM might be a host OS LVM, or a clustered file system LVM. The LUNs that are defined on storage, and construct the LVM, are typically striped at a 64KB multiple. This is done to generally match the underlying physical storage architecture and characteristics.

For traditional spinning disk storage, that LVM logical stripe typically matches the underlying physical RAID stripe width across the disks. This logical stripe might match an underlying physical storage RAID stripe of 64KB, or a multiple of 64KB (128KB, 256KB, 512KB, or 1MB). This creates a consonant geometry of a block-transfer size from the host to the disk storage, minimizing the number of I/O transfers (input/output operations per second [IOPs]) by matching the storage physical block size.

Since flash cells are not striped, they accept transfer sizes from a host LVM and de-stage the blocks internally. They typically take large incoming blocks such as 64KB, and de-stage them to 4KB writes to flash cells. Since there is never a true match between the common de-stage size of a 4KB block and an incoming block of 64KB to 1M, the “striping match” doesn’t apply in the same sense as a disk subsystem. Some specific brands of arrays optimally de-stage block writes to 4KB cell writes when presented with 64KB blocks from the host. Each array is different, and you must work with your array vendor to determine optimal block transfer sizes for array consumption.

We have found that some file systems and back-end storage systems perform much better with a higher LVM stripe size when working with larger data files. An example is the Veritas Cluster File System. Internal SAS lab testing has shown that the Veritas Volume Manager performs better with a 128KB logical volume stripe size than a 64KB logical volume stripe size. This held true even when it was mismatched with underneath flash storage that performed optimally de-staging 64KB block transfers. The performance gain of the logical volume striped at 128KB was more significant than the poorer performance of forcing 128KB transfers to the array storage, even when it optimized on 64KB. In such cases, the flash storage unit might experience slightly higher latency, but this was outweighed by the increased Veritas Cluster Volume Manager performance.

Different brands and types of storage arrays optimize block transfers at different sizes, ranging from 64KB to 1MB, or higher. Our prior advice to always make the logical volume stripe size the same as the underlying storage block size (for example, an LUN stripe size, physical RAID stripe size on spinning disk, or block de-stage size on flash storage), is still generally followed but has since been ameliorated. Recent discoveries have shown that it might be necessary to increase the block size of certain flash storage units above the storage unit preferred settings in order to smooth out the host or cluster file system LVM operations.

LUN CONSIDERATIONS

SAS launches a single WRITE thread per process, but starts multiple READ threads for SAS threaded procedures. Given the single WRITE thread per process, we typically like to see multiple LUNs support a single file system. This ensures that too many serial WRITE threads don't congregate on one LUN. If one LUN is used, WRITE activity can be serialized. We have generally found that striping an LVM with eight or more (even number) LUNs allows better parallel WRITE performance across many SAS processes. The number and size of LUNs supporting a file system of a given size is part of a complex equation involving stripe characteristics, device counts, and so on. If at all possible, use at least eight LUNs per file system. Eight LUNs is generally considered a sweet spot. Unless your file systems are multiple terabytes in size, you shouldn't need to go over that. Thirty-two LUNs across all SAS file systems should be considered a maximum point.

ADAPTER PATHING AND MULTI-PATHING

When external storage resources (for example, 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 I/O workload across the adapters. Specific multi-pathing recommendations and notes are included in tuning 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" (SAS Note 51660)
- "Testing Throughput for Your SAS®9 File Systems: Microsoft Windows Platforms" (SAS Note 51659)

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 card “arrays” in a SAN arrangement. There are flash “appliances” that sit between SAN storage and the server, acting as I/O accelerators.

Flash storage devices consist of flash cells that persistently store data on charged-copper media. The following types are available:

- 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 WRITE cycles of a typical MLC.
- Triple-layer cells (TLC) are high-density flash cells that, while offering a less expensive price, sport a much lower WRITE erasure endurance cycle.

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, as there is some variety. Ancillary features offered or built into many flash arrays include data compression, data de-duplication, and sometimes encryption. All of these features have a direct impact on performance—some very slight, others significant. For example, inline de-duplication of data storage blocks has greater I/O impacts on some arrays than it does on 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 in SAS note, “Troubleshooting system performance problems: I/O subsystem and storage papers” (SAS Note 53874).

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 colocation of cores to associated memory. Associated memory is kept physically close to avoid non-uniform memory (NUMA) access. Avoid NUMA in VMware by using the following methods:

- Disabling node interleaving from BIOS of vSphere host
- Using ESXTOP to monitor the percentage local counter on the memory screen—should be 100%
- Keeping physical memory to local socket capacity
- Not overcommitting I/O, CPU, or RAM (for example, thinly provisioned resources for SAS workloads)

For information about server virtualization, see “Moving SAS® Applications from a Physical to a Virtual VMware Environment” (SAS 2015).

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 or VNX or IBM XIV storage) or as part of a virtualized storage assembly within a network or cloud. It can exist as a group of storage devices managed by an umbrella interface, such as a software defined storage application (SDS). The goals of storage virtualization are similar to server virtualization—maximum utilization of resources, ease of management, reduced costs, and the convenience of tiering data to higher- or lower-performing virtual pools based on performance needed. Storage virtualization is accomplished in much the same way 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, they can be in different same physical places at times, and 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, logical file systems share space from large pools of shared storage. These logical file systems can get moved around behind the scenes, possibly without physical space to back up the stated space for their file system. For example, your file system might have a definition of being 1TB in size but, due to sharing space and thin provisioning (only giving you the space when you actually use it), there might not be 1TB of physical space available all the time.

In virtualized storage, you might be switched from one storage pool to the next or 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, your underlying shared physical storage might support radically different I/O patterns from what SAS recommends.

If using VMware to virtualize storage, pay close attention to LUNs per ESXi host maximum, at 256 LUNs per host. This might inhibit our typical eight LUNs per SAS file system recommendation. In addition, if you are using a clustered file system like IBM Spectrum Scale, please pay attention to host node limitations, as described in the article, “Enabling or Disabling Simultaneous Write Protection Provided by VMFS Using the Multi-Writer Flag” (VMware 2016).

When using virtualized storage, one must pay careful attention to all of the above issues. Below is a short list of best practices for virtualized storage and flash storage. (It is difficult to separate the two.) More information can be found in the individual storage papers listed in the links in the “Recommended Reading” section.

Apply the following 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 space 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 that SAS uses to be migrated without negatively affecting performance. Some tiering is performed on a 24-hour cycle. Even this can cause SAN disruption with large migrations (and SAS migrations are usually large).
- 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 than SASDATA because the data files from SASDATA pages might already be benefitting from being highly shared in the host system file cache.
- Be aware of automated inline data de-duplication, compression, and encryption services in flash

arrays. These are not all created equally across arrays and they can have an effect on performance. If you plan on using these features, consider that many are array-wide 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 Solutions OnDemand offers customers a very significant cloud space. Whether you host your SAS cloud in a privately owned or subscribed space, there are things to be aware of to ensure the desired performance.

Cloud spaces are an amalgamation of all of the topics this paper covers: server and storage virtualization, advanced storage technologies, shared file systems, network virtualization, availability, backup and recovery, security, and a host of other layers. The best practices, caveats, and warnings that apply to these topics, also apply to cloud spaces. 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 avoiding 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 need to engender “thick provisioning,” or better said, guaranteed virtual resources—causing thick provisioning decisions. If you stick to your throughput requirements and let those guide the 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 as well as the 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 flash storage prices are coming down, it is still expensive enough that not all SAS customer sites can afford it for all SAS file systems. 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 that you 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 that will be 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 that is required by the SAS users. It does not matter if internal drives, locally attached storage arrays, internal flash cards, or external SAN or NAS arrays are used as long as they yield the sustained I/O bandwidth that is required by the SAS application or jobs.

In addition to this paper, which gives general information about 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 setting up storage arrays. The links to these papers are listed in "Recommended Reading" below.

REFERENCES

Crevar, Margaret. "Configuration and Tuning Guidelines for SAS®9 in Microsoft Windows Server 2008." *Proceedings of the SAS Global Forum 2011 Conference*. Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/WindowsServer2008ConfigurationandTuning.pdf.

SAS Institute Inc. 2006. SAS Institute white paper. "ETL Performance Tuning Tips." Available http://support.sas.com/resources/papers/ETLperformance07.pdf.

SAS Institute Inc. 2015. SAS Institute white paper. "Moving SAS® Applications from a Physical to a Virtual VMware Environment." Available http://support.sas.com/resources/papers/MovingVirtuaVMware.pdf.

SAS Institute Inc. SAS Note 42197. "A list of papers useful for troubleshooting system performance problems." Available http://support.sas.com/kb/42/197.html.

SAS Institute Inc. SAS Note 39615. "Input/output performance in SAS® is degraded due to excessive memory usage on Windows." Available http://support.sas.com/kb/39/615.html.

SAS Institute Inc. SAS Note 51659. "Testing Throughput for your SAS®9 File Systems: Microsoft Windows Platforms." Available http://support.sas.com/kb/51/659.html.

SAS Institute Inc. SAS Note 51660. "Testing Throughput for your SAS®9 File Systems: UNIX and Linux Platforms." Available http://support.sas.com/kb/51/660.html.

SAS Institute Inc. SAS Note 53874. "Troubleshooting system performance problems: I/O subsystem and storage papers." Available http://support.sas.com/kb/53/874.html.

VMware. 2016. "Enabling or Disabling Simultaneous Write Protection Provided by VMFS Using the Multi-Writer Flag (1034165)." Available https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1034165.

RECOMMENDED READING

Recommended white papers on the following subjects can be found in SAS Note 42197, "A list of papers useful for troubleshooting system performance problems" 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 Computing environments

CONTACT INFORMATION

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

Tony Brown
SAS Institute Inc.
+1 (469) 801-7455
Tony.brown@sas.com

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