Paper SAS6201-2016

How to Maintain Happy SAS®9 Users

Margaret Crevar, SAS Institute Inc., Cary, NC

ABSTRACT

Today's SAS® environment has large numbers of concurrent SAS processes that have to process ever-growing data volumes. To help SAS users remain productive, SAS administrators must ensure that SAS applications have sufficient computer resources, properly configured and monitored often. Understanding how all of the components of SAS work and how they are used by your users is the first step. The guidance offered in this paper helps SAS administrators evaluate hardware, operating system, and infrastructure options for a SAS environment that will keep their SAS applications running at optimal performance and keep their user community happy.

INTRODUCTION

There is one common thread we have found in working with the IT administrators at a SAS customer's location with regard to maintaining happy SAS Foundation and SAS Grid users. That thread consists of ensuring that the underlying hardware is properly configured to support the SAS applications. This is not a trivial task because different SAS technologies often require hardware configurations that are different from the traditional relational database environment. Understanding how SAS is used at your site will help you evaluate options for the hardware, operating system, and infrastructure (middle tier) configurations. This is easier for existing SAS customers and more difficult with new SAS customers or new SAS applications at an existing SAS customer site.

NOTE: This paper discusses how to set up hardware for SAS Foundation applications and solutions. A separate paper is available on how to set up hardware for the new in-memory offerings from SAS such as SAS Visual Analytics and SAS High-Performance Analytics.

In this paper we will do the following:

- discuss briefly how SAS works, especially from an I/O perspective
- discuss how SAS can be used
- offer guidance on how to initially configure hardware for SAS usage
- offer guidance on how to use virtualization with SAS servers
- provide advice on how to monitor the hardware to avoid running out of a computer resource
- discuss whether you should run your SAS components under a single operating system instance or split them across multiple operating system instances (in other words, to grid or not to grid)

This paper consolidates information that has been presented in previous SAS Global Forum papers. The goal for consolidating the information into a single paper is to help the SAS and IT administrators who must support SAS better understand how SAS works and what they should do to keep their SAS customers happy.

HOW SAS WORKS

SAS differs from a traditional *relational database management system* (RDBMS). With an RDBMS, it is typical to have only a single instance of the RDBMS running at any given time. With SAS, it is common to have many SAS instances running at any given time. Generally, SAS users start their own SAS session for each SAS job and application they are running. With the SAS®9 Business Intelligence infrastructure, there are several SAS servers that are started to support the Java clients, in addition to back-end SAS servers and processes that run for each active SAS user.

SAS creates a high volume of I/O, and the interactions with data storage are significantly different from typical interactive applications and RDBMSs. Here are some major points to understand:

- **SAS tends to perform large sequential Reads and Writes**. Some of the SAS Business Intelligence applications perform random data access, but the SAS workload can be characterized as predominately large sequential I/O requests with high volumes of data.

  TIP: If there are multiple concurrent SAS sessions accessing the same data file (each SAS session is accessing the file in a sequential fashion), the access pattern for all the SAS sessions could present as a random access pattern to spinning disk storage, and the I/O subsystem might need to be tuned for random access rather than sequential access.

- **SAS does not pre-allocate storage when initializing or when performing Writes to a file**. When SAS creates a file, it allocates a small amount of storage, and as the file grows during a SAS task, SAS incrementally extends the amount of storage needed.

  TIP: File extension is limited to the amount of available space within the file system currently being used. SAS data sets and individual partitions within a SAS® Scalable Performance Data Server table do not span file systems.

- **Reading and writing of data is done via the operating system's (OS) file cache.** SAS does not use direct I/O by default.

  - Because SAS uses the OS's file cache to read and write data, the maximum I/O throughput rate is restricted by how fast the OS's file cache can process the data.
  - Discussions regarding how to enable SAS to do direct I/O are beyond the scope of this paper. For more information, see http://support.sas.com/resources/papers/proceedings09/327-2009.pdf

- **A large number of temporary files can be created during long-running SAS jobs**. These files are created and could be renamed (toward the end of the task), deleted, and potentially manipulated many times during a long-running SAS job. The size of the files might range from very small (under 1 GB) to very large (in the 100s of GBs). The location of these files is referred to as the SAS WORK area.

  TIP: You need to ensure that SAS WORK is pointing to a file system that has enough I/O throughput to support your users' I/O demands and enough disk space to support all of the temporary files that will be created during each SAS session. This file system needs to support 50% Writes, 50% Reads, and 100% destruction.

- **SAS creates standard operating system (OS) files for its data store**. These files include SAS data sets, indexes, and so on.

- **When executing Writes, there is a single writer thread per SAS session**. However, some SAS tasks that support threading do start multiple reader threads.

Configuring hardware to support the profile of your SAS applications will make a significant impact on overall performance. When your computer resource must support multiple SAS applications (especially if they have widely varying access patterns), you might need to make tradeoffs (performance versus price is the key one).

UNDERSTANDING SAS USAGE

SAS can be used in many different ways, from heavy analytics to simple query and reporting. The IT administrative staff needs to understand how SAS will be used and how SAS interfaces with the computer hardware it is running on. The better their understanding of how SAS works, the better they can configure the computer resources, especially the I/O subsystems (For more information about configuring I/O subsystems for SAS, see **Best Practices for Configuring Your I/O Subsystem for SAS®9 Applications**), file systems, and memory that best support the SAS users at their site. The SAS users will need to work with IT to review and characterize their SAS tasks and SAS interactions. The IT administrators can then use the general guidelines in this paper to set up a UNIX or Windows computer to support SAS jobs and a typical SAS® Business Intelligence deployment.

BATCH SAS JOBS

Batch SAS jobs are individual SAS processes that likely have been around for several years (spanning several releases of SAS). As the name implies, they are run as batch jobs from a command window, from a SAS remote submit, or from a scheduler on a regular basis. If you open the Task Manager on a Windows system or list all the processes running on a UNIX system using the `ps` command, you will see a separate line for each SAS session that is executing the legacy batch SAS job that is currently running.

We refer to these as legacy SAS jobs because they are often written using prior releases and versions of SAS. These legacy batch jobs read data from external sources (which can be anything from text files to relational databases) into a temporary SAS data set in the SAS WORK area. The length of these jobs could range from a few steps to hundreds or even thousands of steps. The jobs could even use the MPCONNECT technology to spawn child SAS jobs from a parent SAS job. All of these different techniques are examples of how SAS users analyze external data sources and produce the desired reports and data marts.

The steps in SAS jobs use different computer resources.

- **I/O:** The SAS DATA step and SAS procedures (known as PROCs: PROC SORT, PROC SUMMARY, and so on) that do data manipulation tend to be very I/O-intensive in nature and heavily access the SAS WORK (and underlying UTILLOC) file spaces. In addition, some steps that manipulate data (that is, analytical data model routines) create additional temporary files that might be larger than the input data files. The SQL join of multiple files within SAS (not an RDBMS) might result in many temporary intermediate join files. Furthermore, when batch SAS jobs run concurrently, the number of temporary files is multiplied because *each batch job starts its own SAS process and creates its own subdirectory in the file system that is associated with the WORK/UTILLOC area*.

  > **TIP:** To maintain happy SAS users, you will have to decide between a single, large file system for SAS WORK that all SAS users will share or multiple, smaller physical file systems (not different directories on the same file system) for SAS WORK that you round-robin your SAS users between.

- **CPU:** The analytical tasks (that is, REG, MIXED, GLM, NLMIXED, GENMOD, and HPF procedures) tend to be more CPU-intensive in nature, and it is not uncommon for a single job to consume an entire processor within your computer until it is finished with its analytical processing.

  > **TIP:** To maintain happy SAS users, you will need to both determine how many of these long-running analytical tasks will be happening simultaneously and make sure you have sufficient CPU cycles to support them along with your other SAS users.

- **MEMORY:** The matrix manipulation tasks (IML, GENMOD, and MIXED procedures) tend to be very memory- intensive in nature due to their use of formats and hash table lookups.

  > **TIP:** To maintain happy SAS users, you will need to make sure you have enough physical memory to support all of these tasks and the file cache needed to hold the data in memory of the other SAS tasks that might be running.

The following is a discussion about different types of SAS applications. Please note that, while there are many SAS applications, the applications that are discussed tend to be the most widely used.

SAS®9 BUSINESS INTELLIGENCE

SAS®9 has many ways of executing SAS. These elements should be accounted for.

SAS Clients: SAS client GUIs make it easier for business users to leverage the power of SAS without becoming a SAS programmer. Here is a list of business user-focused applications:

- SAS® Data Integration Studio (which produces Extraction, Transformation, and Loading ETL flows)
- SAS® Enterprise Guide® (which provides the front end for most of SAS)

- SAS® Web Report Studio (which produces wonderful reports)

SAS Metadata: The SAS® Metadata Server was introduced to maintain all the information about the underlying data files, stored processes, and so on, used by the preceding applications in a centralized location. The SAS Metadata Server keeps the data (referred to as a repository) registered by the SAS applications in an in-memory table. You need to ensure there is enough physical RAM in the computer to keep the repository in a memory database. If physical RAM is exceeded, paging will ensue, which degrades performance.

> **TIP:** Although the data that is accessed by the SAS Metadata Server is in memory, a copy is committed to disk. You need to monitor this copy for its total size. If you are running the SAS Metadata Server on a Windows 32-bit system, please note that the total size of the metadata repository cannot be more than 2 GB (maybe smaller). This is the limit of the amount of memory that a 32-bit version of SAS®9 can access. To get an understanding of the size of the metadata repository, you can use operating system tools to monitor the size of the corresponding files that are written to disk in the RPOSMGR directory. If you see these files getting close to 2 GB in size, then you need to clean out unwanted or unnecessary information from the metadata repository or investigate the migration of your SAS Metadata Server to a 64-bit operating system. See the *SAS Intelligence Platform: Systems Administration Guide* for more details.

SAS Servers: For the back-end SAS servers (SAS Workspace Servers, SAS® Stored Process Server, SAS® OLAP Server), you should set up the hardware for these servers the same way you set up the hardware for the legacy batch jobs (discussed above).The SAS clients are submitting SAS code to these back-end SAS servers in a similar fashion to the way the legacy batch SAS jobs submitted code. The only difference is that you might have a SAS server (SAS session) servicing multiple SAS clients instead of a single SAS user the way a legacy batch SAS job did.

Third-Party Resources. Also, with the onset of Java, several third-party applications (web applications servers, for example) are now required for the SAS®9 Business Intelligence applications to run. This means more computer resources will be needed to support these new SAS sessions. Each site should evaluate the overall system performance and determine if any of the components should be configured to run on separate instances of an operating system. This ensures the SAS users get the computer resources required to avoid performance degradations.

A multiple-tier architecture with a Java middle tier is needed to support the Java technology-based SAS®9 Business Intelligence applications. How you configure a SAS 9.3 and previous version Java Applications Server and a SAS 9.4 Web Applications server can impact SAS performance. The *SAS Intelligence Platform: Web Application Administration Guide* has some suggestions for parameters that need to be modified with regard to the various third-party middle-tier components. These generally pertain to allowing the servers to access more memory and to keep from doing unnecessary garbage collection (which happens if you use the default settings for these components).

CONSIDERATIONS FOR NEW SAS APPLICATIONS

It is fairly easy to use the above guidelines if you have a good understanding of how SAS will be used. A greater challenge lies in configuring hardware resources from scratch for a new SAS application. We will discuss a fairly straightforward process to accomplish it. This section also includes some general usage metrics that can help guide hardware capacity needs. You generally need to do the following:

- Understand what SAS applications will be simultaneously executing on the computer.
- Identify the most resource-intensive areas where these SAS applications can simultaneously overlap.
- Approximate the *demand load*.
- Use heuristics to initially determine the hardware capacity needed.

Understand which SAS applications run simultaneously. This metric entails estimating how many SAS processes are executing simultaneously and knowing the general activity and volume of data being manipulated by each.

> **TIP:** We are saying simultaneous processes, not users. There will generally be a mix of batch and ad hoc users running reports and queries, ETL processes for warehousing, SAS® Business Intelligence (BI) applications (SAS® Enterprise Guide® client queries, SAS® Enterprise Miner™ model runs, OLAP), and a myriad of possible other SAS applications.

Identify resource-intensive areas. This step requires identifying the heaviest activity in each process (for example, each job step) and building a picture of what the overlap looks like when the processes are simultaneously running. You also need to identify what would be needed for disk capacity, general throughput, CPU, and memory to get everything serviced. This will give you a *load* picture of what you are placing on the computer.

Approximate the demand load. The demand load is based on the aggregation of everything running above and the maximum amount of time the SAS user would like to see the SAS job completed. What is the widest workload in terms of megabytes per second that are demanded simultaneously? (This applies to all the file systems, permanent data, or temporary files—SAS WORK -- that you will need to use for your SAS applications.) How many CPUs would be needed by the processes? How much memory?

Estimate needed hardware capacity. Initially, you will have to guess at some of these things, but an educated guess is better than no guess. If you are planning a large system or expansion, it might be worthwhile to work with your SAS account representative to get the proper SAS resources to help in the sizing effort. You want your new system to be correctly supplied with capacity and throughput capability, so all will go well.

> **TIP:** For each SAS process running simultaneously on the computer, it is not uncommon for there to be a mixture of different types of jobs running concurrently. Good planning can help you initially configure your computer to support all the different SAS tasks running concurrently, but you will need to monitor your hardware to determine if you have any bottlenecks as the number of SAS users and the amount of data they will analyze increase.

The need to partition a large operating system into multiple workgroups or soft partitions, the use of a centralized computer to house the SAS client applications (like Citrix), or the use of virtualization software (like VMware) might be required to ensure that your SAS servers are guaranteed the computer resources they require. SAS runs in this infrastructure. However, performance might be impacted, especially with the virtualization software. A discussion about how to configure these software products is beyond the scope of this paper.

GUIDANCE ON ESTIMATING HARDWARE RESOURCES FOR SAS USAGE

Here are some guidelines that can help you determine how many resources you might need.

CPU:

SAS recommends the use of current generation processors whenever possible for all systems. Here are some general guidelines for the number of concurrent SAS sessions per core:

- For light report reading, or light query work (files less than 2 GB), you can plan on 15–20 users per core.
- For heavy processes, like jobs that involve large data sets (5+ GB), you can plan on 8–10 users per core.
- For analytical processing using SAS statistical procedures (including forecasting and modeling) and data mining, you can plan on 1–2 users per core. These tasks tend to run for a long period of time.

MEMORY:

For each tier of the environment, SAS recommends the following minimum memory guidelines:

- SAS Compute tier: A minimum of 8 GB of RAM per core
- SAS Middle tier: A minimum 24 GB **or** 8 GB of RAM per core – whichever is larger
- SAS Metadata tier: A minimum of 8 GB of RAM per core

It is also important to understand the amount of virtual memory paging space that is required in the system. SAS recommends that virtual memory be 1.5–2 times the amount of physical RAM. If, in monitoring your system, it is evident that the machine is paging a lot, then SAS recommends either adding more memory or moving the paging file to a drive with a more robust I/O throughput rate compared to the default drive. In some cases, both of these steps might be necessary.

I/O:

This is the hardest thing to "guesstimate." It depends on how large the data is that is being manipulated, how fast it must be read and written to satisfy your needs, and how many SAS sessions are executing at once. The following is a list of very general heuristics about file systems related to SAS:

- Request that your storage administrator (the person who is responsible for configuring the disks that are used in your computer) set up your file systems according to 1) required I/O throughput and 2) capacity needs. To assess capacity requirements, consider the amount of work that will be generated by multiple people using the file system at the same time. A very simple rule of thumb is to calculate the Reads and Writes, and multiply that figure by 3 for PROC SORT or other procedures that manipulate or calculate data (for example, FREQ, UNIVARIATE, MERGE, and so on). This will not be exact, but you will have a reasonable estimate. **NOTE:** This estimate for is for /saswork. An estimate for /sasdata will be based on historical data kept, data generated, and growth ratio.

- Set up a minimum of three file systems to support SAS (operating system, permanent SAS data files, and SAS temporary files – referred to as SAS WORK) and potentially a fourth file system (if your SAS jobs are doing many sorts – referred to as SAS UTILLOC). Previously, the SAS file systems were recommended to be a separate and independent set of physical disks, which provided physical separation on older SAN models. Newer SAN models are typically a "striped everything" system. Also, flash storage is physically different, so this separation is no longer feasible. Use the following SAS file system characteristics and locations as a reference, especially if the disk must be shared between SAS applications and users.

  > **TIP:** The system administrator or installer should be cautious sharing the underlying physical storage of these heavily used I/O file systems with other applications (whether these applications are performing heavy I/O or doing random access to the data) to avoid I/O conflicts between SAS and these other applications. If they must, as in the case of "striped everything" spinning storage, careful monitoring and provisioning must be made. If it is possible not to share highly random I/O with underlying SAS provisioned resources, it is wise to do so.

- Consult with your storage administrator regarding how to provide your throughput needs based on the general information below. Here are the I/O throughput requirements for the SAS Compute tier:

  - Overall I/O throughput needs to be a minimum of 100–125 MB/sec/core.
  - For SAS WORK, a minimum of 100 MB/sec/core.
  - For permanent SAS data files, a minimum of 75–100 MB/sec/core.

- Certain SAS applications might require additional file systems to hold source data, user playpens, SAS Scalable Performance Data Server tables, SAS Scalable Performance Data Server temporary tables, and other SAS output (secondary temporary files referred to as UTILLOC files, OLAP cubes, SAS Web Report Studio Query Cache, and so on).

  > **TIP:** If your SAS application requires these file systems, you need to make sure they are set up as robustly as the initial three files systems required by SAS.

- Ensure that the heaviest-used file systems (usually SAS WORK, permanent storage, UTILLOC, and so on) will be able to sustain the above I/O throughput requirements. Again, consult with your storage administrator for help.

- The initial disk space allocated for each of these file systems will depend on the SAS application. An ETL job that transforms data from one source to a SAS data warehouse format generally has many steps that create temporary files, so you will need to make sure you have enough disk space for all the temporary files that are created. The same is true with ad hoc jobs that might be run by your end users. However, some BI applications or SAS solutions might not create any temporary files.

  - So, general formula for the size of the SAS WORK area would be the number of steps in the job times the size of the input file times the number of current SAS jobs running.
  - For a better value for the number of files systems needed and their size, a detailed technical assessment is required. Please work with your SAS account team on how to accomplish this.

NOTE: The "Best Practices for Configuring your IO Subsystem for SAS®9 Applications" paper that we have referenced goes into more detail about the RAID levels and preferred tools to use to set up these file systems.

GUIDANCE ON TUNING HARDWARE RESOURCES FOR SAS USAGE

Now that we have discussed how SAS®9 applications work in general and how to do a simple sizing for the hardware, there are some general configuration tuning issues to consider for the hardware where you will be running your SAS®9 applications.

- **Moving the default location of SAS WORK directory:** By default, the `sasv9.cfg` file for each installation of SAS has the SAS WORK directory point to a known directory on all hardware platforms. This directory is the same directory where the operating system tends to write its swap files. You will notice a performance improvement if you move the SAS WORK directory to another location. See the guidelines in Best Practices for Configuring Your I/O Subsystem for SAS®9 Applications.

- **Separating SAS WORK files into a different file system from SAS UTILITY Files**: Several of the threaded procedures create temporary utility files in the SAS WORK file system while the procedure (for example, the SORT and SUMMARY procedures) runs. If your SAS application will heavily use these procedures, we recommend that you use the UTILLOC SAS parameter to point these temporary files to a separate file system from SAS WORK to avoid I/O conflict.

  NOTE: The new file system for UTILLOC needs to be just as robust as the existing SAS WORK file system.

- **Running the SAS jobs with the FULLSTIMER option turned on:** When running your batch SAS jobs, you can add the statement—OPTIONS FULLSTIMER; — to the job. When you include this statement, SAS prints to the SAS log the statistics it gathers regarding the computer hardware resources used when running the SAS application. These statistics can help you determine if you have any I/O bottlenecks with the processing, the amount of memory needed to run the application, and so on. More details about how to interpret this information can be found in A Practical Approach to Solving Performance Problems with the SAS System.

  TIP: The FULLSTIMER option does not report the same statistics when run on Windows servers as it does when run on UNIX servers.

- **Giving the SAS applications all the memory they need:** Prior to SAS 9.4, the default values for MEMSIZE and SORTSIZE for the UNIX and Windows ports of SAS were not ideal. We have found that better starting values are MEMSIZE=2 GB and SORTSIZE=1 GB, and we strongly suggest that you make these changes for general usage. You might have some jobs that require higher MEMSIZE settings, and you must set the MEMSIZE value for these jobs based on your past experience. If you want to know how much memory your SAS job is using, simply run the job with the FULLSTIMER option turned on.

  - MEMSIZE must be set at the invocation of the SAS session, so you will need to make this change in the `sasv9.cfg` file in the SAS installation directory.
  - Extensive testing by SAS Technical Support has shown the ideal setting for SORTSIZE is between 512 MB and 1 GB. There might be times when increasing SORTSIZE is justified (the main one being if you can fit the entire file being sorted into physical memory), but in general, it is best to set SORTSIZE to a value in this range.

- **Do not give SORT too much memory:** We have found that if you cannot grab enough physical RAM to put the file you are sorting into memory, it is best to use a small bucket of memory when sorting the data. The size of the bucket of memory that SORT uses is controlled by the SORTSIZE option within SAS.

  TIP: SORTSIZE can be set within the SAS job using the OPTIONS statement.

- **Balancing MEMSIZE versus physical memory:** The optimal memory configuration is the sum of the MEMSIZE values of all the simultaneous SAS sessions fit into physical memory. Determine how much memory will be needed if all your concurrent SAS sessions grab the maximum memory that they can. The total amount of RAM needs to be greater than this value. Please note this might require you have more than 8 GB of RAM per core.

- **Configuring the amount of swap space on the computer to match the amount of physical memory in the computer:** It is highly recommended that your swap space be 1.5 times the amount of physical memory in your computer.

- **Modifying the SAS applications to reduce the amount of I/O:** There are two ways to reduce the amount of I/O that a SAS session uses:

    - Modify the SAS program to reduce the number of times you have to process the data by including WHERE processing, using indexes, and so on.
    - Reduce the number of data accesses by processing more data each time the disks are accessed by increasing the values of the BUFNO, BUFSIZE, CATCACHE, and COMPRESS SAS options.

- **Configuring for large blocked sequential I/O:** For conventional spinning disk and flash storage that uses "pseudo-striping", we highly recommend that you use operating system or external storage commands to ensure that the file systems being used by SAS have a large stripe size (at least 128 KB or 256 KB, but not larger than 512 KB).

- **The default BUFSIZE value:** Prior to SAS 9.4, the default BUFSIZE value was not ideal for today's fast storage and large data transfer volumes. We strongly suggest that you match the SAS BUFSIZE value to match your storage's stripe size. Changing it to be a minimum of 64 KB is strongly suggested.

    TIP: There are times when increasing the BUFSIZE option for a SAS data file to larger than what SAS chooses will help, but when doing this, you need to consider all the ways the file will be used before you reset this value. BUFSIZE is associated with a SAS data file when it is created and will be used every time the file is accessed (both during the ETL process and the end-user exploitation).

- **Separating input file systems from output file systems for SAS steps that act as data filters:** SAS steps that act as data filters are DATA steps that produce transformations with little reduction of the incoming data, SQL statements that do similar data manipulation, and PROC APPENDs that add data to the end of another data file. If you can split where the SAS step reads from and then writes to, you will reduce the potential of I/O bottlenecks.

- **Never turn the NOTHREADS options on:** If you need for your SAS applications to run as single-threaded applications, simply set CPUCOUNT=1.

    TIP: Switching from the default of THREADS to NOTHREADS will cause several SAS procedures to use their SAS 8.2 algorithms, which in many cases are not as I/O-efficient as their SAS®9 algorithm. (The SORT procedure is a prime example).

- **Restricting the number of processors that the multi-threaded SAS procedures can use:** If you will be running SAS®9 on a computer (or computer partition) with more than four processors (cores), please set the value of CPUCOUNT to 4. The default value is all cores in the computer. This value can cause issues if you will be running multiple concurrent instances of the multi-threaded procedures.

- **Running an RDBMS on the same computer as SAS:** This is possible but, depending on what applications are using the RDBMS and the settings required to support these applications, it might not be a good idea.

    TIP: If the primary user of the RDBMS is an OLTP application that needs to have a guaranteed response time, it would not be a good idea to house both of these applications on the same operating system. However, if the primary user of the RDBMS is SAS, it does make sense to

house them together. You might have to compromise on the operating system settings so that both applications can run properly.

- **Additional Tuning Information per Operating System:** These can be found in the Operating System Tuning section of the SAS Usage Note 53873.

In addition to all of the above information, there is a checklist of what needs to be considered (along links for more details) that one should refer to prior to setting up a SAS infrastructure.

GUIDANCE ON VIRTUALIZATION FOR SAS SERVERS

Now that we have discussed how to configure hardware for SAS®9 applications, we need to address the particulars of implementing a virtual system (VMware ESX, Microsoft HyperV, Red Hat Enterprise Virtualization, and so on). In general, we need to address moving from a large physical system to several smaller virtual systems. The best practice for achieving optimal performance is to follow the same physical system tuning guidelines for the virtual host operating system and underlying hardware and I/O subsystem. Here are some additional tuning tips to follow:

- **Do not under-provision the underlying hardware in the physical frame.** By this, we mean do not overcommit your physical resources by allowing the virtual CPUs to grow beyond physical backing, memory, or I/O connections than what is thickly, physically provisioned.

- **Ensure virtual CPUs that are created are backed with dedicated physical cores.** Ideally, you should have a 1:1 ratio between the virtual cores and the physical cores. The ability to allow your virtual hypervisor to move the cores you are using randomly within the physical system can cause latency because the hypervisor will move only the processing application and not the data it is accessing in L1/L2 cache.

- **Make sure that the cores in the virtual CPU and L1/L2 cache are all on the same processor (CPU card) when creating the definition of a virtual CPU.**

- **If at all possible, connect the back-end storage with directly attached adapters to ensure optimal I/O throughput.** Anytime you introduce network traffic, additional IP traversals, and so on, you introduce the risk of encountering problems and performance degradations.

- **Consider the other virtual machines running on the same physical host and their impact on SAS applications.** For example, if you have a database running on another virtual host that consumes large volumes of I/O, this might impact the available I/O bandwidth inside the physical host that is being shared with SAS.

VIRTUALIZED STORAGE SOLUTIONS

In addition to knowing the virtual CPU virtualization used for SAS sessions and servers, virtualization can also apply to storage arrays. Storage providers are offering a wave of new and advanced storage subsystems. These offerings promise virtualized, thin-provisioned, tiered, and intelligent storage that is easy to manage and will reduce costs. For many random and mixed workload applications, the promises deliver well and with good performance. Unfortunately, the SAS I/O workload profiles tend to violate some of the primary design assumptions underlying the configuration of these new systems. The paper SAS® and the New Virtual Storage Systems addresses what needs to be considered when configuring these new storage systems for expected performance on this.

Additional information about virtualization with SAS can be found in the following papers.

Virtualization: What does it mean for SAS®?
Moving SAS® Applications from a Physical to a Virtual VMware Environment

VIRTUALIZED PRESENTATION ENVIRONMENTS

One other virtual environment to discuss is Presentation Virtualization Environments (such as Microsoft Terminal Services or Remote Desktop Services, Citrix Presentation Server, or XenApp). Most SAS clients are supported on

these environments. For a list of what versions are supported, please review this SAS Support Note.

GUIDANCE ON MONITORING HARDWARE FOR SAS

The one thing we can guarantee is that the use of SAS tends to increase at your site over time. This growth can be caused by more concurrent SAS sessions running on the computer, the size of the data being analyzed and reported getting larger, and the types of SAS applications increasing over time. Please note that the guidelines for setting up your hardware to run SAS are only guidelines, not detailed instructions. We recommend that you monitor your hardware on a regular basis to ensure you do not run out of a computer resource and therefore cause SAS sessions to perform poorly or terminate unexpectedly.

Starting with SAS 9.4, the SAS® Environment Manager has been added to SAS to help you monitor all your SAS servers and the underlying hardware. For the initial release of SAS 9.4, SAS Environment Manager focuses on serving as the operational monitoring solution with plugins for the administration, management, and monitoring of SAS technologies including the administration of the SAS® Web Application Server. Additional functionality includes agent-based, auto-discovery of resources, collection of resource consumption, monitoring of log events, alerting, and many ways to present and view the collected information. Over the 9.4 life cycle, SAS Environment Manager will be extended to provide more robust and centralized SAS administration and monitoring. Information about the new SAS Environment Manager release can be found at in this SAS Administration blog post.

More details about using this new feature can be found in these two SAS Global Forum papers:

- Monitoring 101: New Features in SAS® 9.4 for Monitoring Your SAS® Intelligence Platform
- Log entries, Events, Performance Measures, and SLAs: Understanding and Managing your SAS® Deployment by Leveraging the SAS® Environment Manager Data Mart

If you are not on SAS 9.4 yet, a SAS Global Forum paper from a few years ago talks about how to monitor your hardware using SAS and other tools with any version of SAS. For more information, see Logging 101: Leveraging the SAS Enhanced Logging Facility .

If you have been using the SAS Audit, Performance, and Measurement packagetool that is available in versions of SAS prior to SAS 9.4, this functionality is being added to SAS Environment Manager.

And for the meantime, SAS® Management Console will continue to serve as the primary interface for SAS content administration.

Please note that you might want to do snapshot monitoring periodically, or you can do continuous monitoring on a daily basis and put the results into a SAS data mart, which can be used to proactively forecast the SAS usage (capacity planning). This snapshot helps you plan for hardware upgrades before your users start experiencing performance degradations. Details of how to do this monitoring is beyond the scope of this paper.

GUIDANCE ON BENEFITS AND HARDWARE REQUIREMENTS FOR SAS GRID MANAGER

As the number of SAS sessions, the amount of data that will be manipulated, the number of different SAS applications and servers, as well as third-party components that will be used simultaneously increases, you will have to decide if running all of these SAS sessions and components under a single image of an OS will glean your SAS users the best performance or if going to SAS Grid is what you need to consider.

Please review the list of benefits that SAS Grid can give you and use it to help you decide if you want to pursue using SAS Grid:

- Your SAS applications are mission critical, and you need to set up a highly available infrastructure.

- You have many SAS users running multiple SAS applications, and you want to implement a shared SAS analytic environment that allocates resources as needed.
- You have end-of-month or end-of-quarter SAS processing needs that have very tight SLAs and need to be guaranteed the computing resources to meet these SLAs.
- You would like to establish a hardware and software infrastructure that can be scaled out to meet your ever-growing number of SAS users and the ever-growing data being analyzed by these SAS users.
- You would like to see if you can gain some performance improvements by using the new SAS High-Performance Analytics processes or by taking your existing SAS processes and converting them to a distributed processing format.

> **NOTE:** Not all SAS processes are good candidates for distributed processing. For example, processes that rely heavily on OLAP processing do not lend themselves to parallelization.

The only additional hardware infrastructure component that is required to support SAS Grid is a shared file system, which allows multiple SAS Grid nodes to easily and efficiently share data. This shared file system must be capable of sustaining the I/O bandwidth required by your SAS applications. SAS Grid environments have been tested with a variety of shared file systems, and the results can be found in **A Survey of Shared File Systems (updated October 2014).**

CONCLUSION

Managing your computer's performance is an ongoing and, sometimes, complicated task. It is essential that you establish performance baselines to help set attainable user expectations and to serve as a reference in order to compare performance. The use of tools to establish performance baselines and to monitor system and application performance is crucial to developing, improving, and maintaining high-quality operating environment performance.

ACKNOWLEDGMENTS

Many thanks to Tony Brown, Jim Kuell, Christina Harvey, and Simon Williams for their review of the content of this paper.

CONTACT INFORMATION

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

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