How to Maintain Happy SAS® Users

Margaret Crevar, SAS Institute Inc., Cary, NC

Last Update: August 1, 2008

ABSTRACT

Today's SAS® environment has high numbers of concurrent SAS processes and ever-growing data volumes. It is imperative to proactively manage system resources and performance to keep your SAS community productive and happy. We have found that ensuring your SAS applications have the proper computer resources is the best way to make sure your SAS users remain happy.

INTRODUCTION

There is one common thread we have found working with the IT Administrators at a SAS customer's location with regard to maintaining happy SAS users. That thread is 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 different hardware configurations. Understanding how SAS will be used by the end user will help you evaluate options for the hardware, operating system, and infrastructure (middle tier) configuration. This is easier for existing SAS customers and more difficult with new SAS customers or new SAS applications at an existing SAS customer site.

In this paper we:

- discuss briefly how SAS works, especially from an I/O perspective
- offer guidance on how to initially configure hardware for SAS usage
- 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

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

HOW DOES SAS WORK

SAS differs from a traditional *relational database* (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 new SAS®9 Intelligence Platform, there are also 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 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 new SAS Intelligence Platform applications perform some random data access, but the SAS workload can be characterized as predominately large sequential I/O requests with high volumes of data.

  Note: 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 likely appear random to the operating system, and the I/O subsystem needs 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, but as the file grows during a SAS task, SAS extends the amount of storage needed.

  Note: 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 OS’s 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. The maximum I/O throughput rate we have seen via file cache has been near 1GB per second, but it nominally is around 600 to 800MB per second per operating system instance.
  - Discussions regarding how to enable SAS to do direct I/O are beyond the scope of this paper, but there are ways to make this happen.

- A large number of temporary files can be created during long-running SAS jobs. These files are created, might be renamed towards the end of the task, deleted, and potentially manipulated many times within a long running SAS job. The size of the files might range from very small (under 100MB) to very large (in the 10s of GBs) on average. The location of these files is referred to as the SAS WORK area.

  Note: 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, as well as enough disk space to support all the temporary files that will be created during each SAS session.

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

- When executing writes, there is a single writer thread per SAS session. Some SAS tasks that support threading do start multiple reader threads, however.

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).

HOW WILL SAS BE USED?

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, 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 Intelligence Platform deployment.

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

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 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. These jobs might be only a couple of steps in length, there can be hundreds or thousands of steps, or they might even use the MPCONNECT technology that spawns children 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.

- The data manipulation tasks (that is, SAS DATA steps and SAS procedure steps, which are known as PROCs, such as PROC SORT, PROC SUMMARY, and so on) tend to be very I/O-intensive in nature and heavily access the SAS WORK area. 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. When doing a SQL join of multiple files within SAS (not an RDBMS), the SQL steps can 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 sub-directory in the file system that is associated with the WORK area*.

  Note: 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.

- 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.

  Note: To maintain happy SAS users, you will need to 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.

- The matrix manipulation tasks (that is, IML, GENMOD, MIXED) the use of formats and hash table lookups tend to be very memory intensive in nature.

  Note: 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 may be running.

SAS®9 INTELLIGENCE PLATFORM

SAS®9 has introduced many new ways of executing SAS, including many Java front-end applications to make it easier for business users to leverage the power of SAS without becoming a SAS programmer. The array of business user-focused applications include SAS® Data Integration Studio (which produces ETL flows), SAS® Enterprise Guide® (which provides the front end for most of SAS) and SAS® Web Report Studio (which produces wonderful reports). Also, with the onset of Java, several third-party applications (Web Applications Servers for instance) are now required for the SAS®9 Intelligence Platform 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 to ensure they get the computer resources required to avoid performance degradations. More detail is provided in a later section of this paper.

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 this in-memory database resident. If physical RAM is exceeded, paging will ensue, which degrades performance.

Note: Even though 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 2GB (maybe smaller) because this is the limit to 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 2GB in size, then you need to clean out unwanted/unneeded 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.

A multiple-tier architecture with a Java middle tier is needed to support the Java technology-based SAS®9 Intelligence Platform applications. How you configure a Java Applications Server and a Web Applications server can impact SAS performance. The SAS®9 Intelligence Platform Administration Guide has some suggestions for parameters that need to be tweaked 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).

For the back-end SAS servers (workspace, 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 because the SAS clients are just 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.

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, supported by 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 may simultaneously overlap.
- Roughly calculate the *Demand Load*.
- Use heuristics to initially determine the hardware capacity needed.

Understanding What SAS Applications Will Be Simultaneously Executing on the Computer

This entails estimating how many SAS-related processes are executing simultaneously and knowing the general activity and volume of data being manipulated by each.

Note: 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.

Identifying the Most Resource-Intensive Areas Where these SAS Applications Might Simultaneously Overlap

This means 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, as well as 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.

Roughly Calculating 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 application completed in. 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 utilize for your SAS applications)? How many CPUs would be needed by the processes? How much memory?

BEST PRACTICE GUIDELINES TO INITIALLY SIZE THE HARDWARE CAPACITY NEEDED

Initially you will have to guess at some of these things, but an educated guess is better than no guess. Following are some guidelines that can help you determine how many resources you might need:

CPU

Using the new processors that are running over 2 GHz for UNIX systems and over 3 GHz for Intel/AMD systems:

- For light report reading, or light query work (files less than 2 GB), you can plan on 15–20 users per CPU.
- For heavy processes, like jobs that involve large datasets (5+ GB), you can plan on 8–10 users per CPU.

- For analytical processing using SAS statistical procedures (including forecasting and modeling) and Data Mining, you can plan on 1–2 users per CPU. These tasks tend to run for a long period of time.

Memory

- For UNIX systems and 64-bit Windows systems, we generally recommend a minimum of 4 GB of RAM per CPU.

  Note: Additional memory is the best insurance in the case of hardware configuration uncertainty. While it can't make up for over-utilization of CPU and I/O, it can help.
- For 32-bit Windows systems, a maximum of 4 GB is ideal. This can be increased to 8 GB on a 4+ processor computer but do not exceed 8 GB because 32-bit Windows will spend more time managing the extra memory than managing the applications running on the computer.

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/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 your Storage Administrator (the person who is responsible for configuring the disks that are use in your computer) to 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 things like SORTS or other procedures that manipulate or calculate data (for example, FREQ, UNIVARIATE, MERGE, and so on). This won't be exact, but you will have a reasonable estimate.

- Consult with your Storage Administrator regarding how to provide your throughput needs based on the general information below.
  - For fewer numbers of simultaneous processes (less than 25), using smaller amounts of data (less than 5 GB per SAS session), a throughput rate of 15–25 MB per second per SAS session is normally a good place to start.
  - For more users, and larger data (greater than 5 GB per SAS session), increase this rate to 50–75 MB per second per SAS sessions.
  - For very large data and high activity, you might need greater than 120 MB per second per SAS session.

- 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). Ideally, the SAS file systems would be their own independent set of physical disks. Use the following SAS file system characteristics and locations as a reference, especially if the disk must be shared between SAS applications and users.

Note: The system administrator or installer should avoid sharing 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.

- Certain SAS applications might require additional file systems to hold source data, user playpens, SAS Scalable Performance Data Server tables and 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).

  Note: 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 physically separate (for example, on different volumes or LUNs and physical underlying disks) from each other. Again, consult with your Storage Administrator to help with this.

- 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 in this process. The same is true with ad hoc jobs that might be run by your end users. However, some BI applications 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.

Please note that the "Best Practices for Configuring your I/O Subsystem for SAS®9 Applications" paper that we have referenced goes into more details regarding the RAID levels and preferred tools to use to set up these file systems.

If you are planning a large system or expansion, it might be worthwhile to work with your SAS Account Team 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.

Note: For each SAS process running simultaneously on the computer, it is not uncommon for there to be a mixture of the 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 increases.

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) may be required to ensure your SAS servers can be guaranteed the computer resources they require. SAS runs in these scenarios; 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 CONFIGURING HARDWARE FOR SAS

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 you will be running your SAS®9 applications on.

- *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 following the guidelines in the "Best Practices for Configuring your I/O Subsystem for SAS®9 Applications" paper.

- *Separating SAS WORK files from SAS UTILITY Files:* Several of the threaded procedures create temporary utility files in the SAS WORK file system while the procedure runs (that is, the SORT and SUMMARY procedures). If your SAS application will heavily use these procedures, then we recommend that you use the UTILLOC SAS parameter to point these temporary files to a different file system from SAS WORK to avoid I/O bottlenecks.
    - The new file system for UTILLOC needs to be just as robust as the existing SAS WORK file system.
    - A SAS Technical Support note goes into more details on how to use this parameter and its ability to reference multiple directories.

- *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 and have SAS print to the SAS log the statistics it gathers behind the scenes regarding the computer hardware resources used when running the SAS application. These 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 the "A Practical Approach to Solving Performance Problems with the SAS System" paper.

Note: FULLSTIMER does not report the same details of information when run on Windows servers as it does on UNIX servers.

- *Giving the SAS applications all the memory they need:* The default values for MEMSIZE and SORTSIZE with the UNIX and Windows ports of SAS are not ideal. We have found that better values are MEMSIZE=512M and SORTSIZE=256M and strongly suggest that you make these changes. However, if you want to know how much memory your SAS job is using, simply run the job with FULLSTIMER 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 256 M and 512 M. There may be times when increasing SORTSIZE is justified, but, in general, it is best to set it between these two values.

- *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 parameter within SAS. In our testing, we have found that a good value for SORTSIZE is 256 M.

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

- *Balancing MEMSIZE versus physical memory:* The optimal memory configuration is to have to have the sum of the MEMSIZE values of all the simultaneous SAS sessions fit into physical memory.

- *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 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 parameters of the BUFNO, BUFSIZE, CATCACHE and COMPRESS SAS options.

- *Configuring for large blocked sequential I/O:* We highly recommend that you use operating system or external storage commands to specify the file systems being used by SAS have a large stripe size (at least 64 K or 128 K, but not larger than 256 K).

- *The default BUFSIZE value is best:* For the most part, the default BUFSIZE that SAS chooses for a file is best (especially when using indexes with high cardinality).

  Note: 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 APPENDs that adds 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 NOTHREADS on:* If you need for your SAS applications to run as single-threaded applications, simply set CPUCOUNT=1.

  Note: 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 algorithms (SORT is a prime example).

- *Restricting the number of processors 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 and this can cause issues if you will be running multiple concurrent instances of the multi-threaded procedures.

- *Running a 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.

  Note: If the primary user of the RDBMS is an OLTP application that needs to have a guaranteed response time, then 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, then it does make sense to house them together. You might have to compromise on the operating system settings so that both applications can run okay.

- *Additional Tuning Information per Operating System:*
    - AIX: www.sas.com/partners/directory/ibm/AIXTuningGuide.pdf
    - Solaris: www.sas.com/partners/directory/sun/sas9-on-solaris10-paper.pdf
    - HP-UX: www.sas.com/partners/directory/hp/papers.html#config
    - Windows: www2.sas.com/proceedings/sugi26/p277-26.pdf

GUIDANCE ON MONITORING HARDWARE FOR SAS

The one thing we can guarantee in this paper is that the use of SAS tends to grow at a customer site over time. This growth can either be more concurrent SAS sessions running on the computer; the size of the data being analyzed/reported gets larger; and/or the types of SAS applications can increase over time. This concept, along with the fact 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.

There have been several papers written that discuss ways to monitor the hardware:

- A SAS Global Forum 2007 paper entitled *Ensuring You Have the Proper Resources for your SAS®9 Applications* (Crevar, 2007) discusses what computer resources need to be monitored for various types of SAS applications.
- A methodology for solving performance problems with your SAS jobs or applications is documented in the SAS white paper *A Practical Approach to Solving Performance Problems with the SAS System* (Brown, 2001).
- A follow-up paper entitled *Solving SAS Performance Problems: Employing Host-Based Tools* (Brown, 2006) goes into more details about how you can solve performance problems by using standard monitoring tools that ship with most of the commonly used operating environments.

If you want to monitor an area for a few minutes at a time, there are some very simple tools that enable snapshot monitoring: on UNIX, use the commands: *top*, *prstat*, or *topas*; on Windows, use TaskManager, PerfMon, or Process Monitor from Microsoft Sysinternals. Also, there are some third-party tools that you can use; for example, HP OpenView GlancePlus and Solaris Resource Management from Sun Microsystems. These tools have very nice graphical user interfaces, and they have the capability to define “rules” that make the task of monitoring the hardware very easy. Usually, these tools run interactively and do not produce log files.

If you want to monitor an area for longer periods of time, the paper *Solving SAS Performance Problems: Employing Host-Based Tools* (Brown 2006), describes the tools that you can use. In the paper, you’ll find information about setting parameters for various tools, such as the Windows Performance Monitor (PerfMon) and the UNIX tools that use the commands: *sar*, *iostat*, and *vmstat*, to monitor system resources as well as interpret the information you have collected. There are sample scripts that can be run where you pass in the collection interval and how many collections you would like, and this information is written out to several log files. The log files are interpreted by manually scrolling through the lines of data. IBM has a nice tool for AIX called *nmon* that takes all the standard UNIX monitor data and puts it into high-quality graphs.

CONTINUOUS HARDWARE MONITORING

If you want to monitor your hardware continuously to be sure that you do not run out of a computer resource or that there are no bottlenecks affecting an important computer resource, third-party monitoring tools will make this task easier.

SAS is developing integration packages for HP OpenView Operations, IBM Tivoli Enterprise Console, and Microsoft Operations Manager. Each integration package is customized for a specific product, but the overall objectives for managing and monitoring the SAS®9 environments are consistent across the various products. The integration packages will include the following attributes:

- Packaged suite of scripts, tools, monitors, and documentation which describe the installation and integration components for SAS 9.1.3
- Facilities to start, stop, pause, and restart each of the key SAS servers: SAS Metadata Server, SAS Stored Process Server, SAS OLAP Server, SAS object spawner
- Availability management facilities to verify the status of key servers and generate events as necessary for server status
- Ability to monitor key SAS server metrics and generate events into the management environment—metrics such as processor utilization, memory allocation, and storage utilization
- Ability to analyze SAS server logs for key actions and generate events into the management environment—actions such as server initialization, user connection events, user ID authentication, and server error events

For more details on these integration packages, please reference the SAS® Global Forum 2007 SAS Presents session entitled *Monitor Your SAS®9 BI Environment with Enterprise Systems Management Applications and SAS® Management Console*.

SINGLE OPERATING SYSTEM IMAGE OR MULTIPLE OPERATING SYSTEM IMAGES?

As the number of SAS sessions; the amount of data that will be manipulated; and 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. Here are some situations to consider:

- It is very common for existing SAS customers to set up a large SMP computer with a single instance of an operating system on it to execute their legacy SAS jobs. This works great until the amount of I/O throughput required exceeds what the operating system's file cache can provide. When this happens, your SAS applications will become bottlenecked by the I/O throughput rate of the operating system's file cache. There are two ways to overcome this.

    - Divide your legacy batch jobs into groups and running each group of jobs within its own OS. This can be done on larger SMP UNIX boxes. You have to decide if you want hard partitions (LPARS, zones, containers, and so on.) or soft partitions (using the new virtual partitioning tools or WLM). There are pros and cons to both, but they exceed the purpose of this paper.
    - Determine if there is a way to do direct I/O (avoid using the OS's file cache) on the computer where you are executing your legacy SAS jobs. This can be done with the SGIO SAS option on Windows computers—with the –dio option when setting up JFS2 file systems on AIX and by implementing a Veritas file system with direct I/O enabled.

- As we have described the many new components to the SAS Intelligence Platform applications, you might have noticed that certain components are more CPU- and I/O-intensive and other components are more memory-intensive. We have found that as the SAS sessions start getting close to using up all the resources of a computer, it is best to house these components on different instances of an OS. When doing this, the preferred way to go is to use hard partitions or individual computers because the main reason you are doing the segregation of tasks is to avoid sharing computer resources.

A couple of things to mention with regard to the above are that you need to:

- Determine if the SAS sessions are sharing data before you house them on different instances of the operating system. If they are sharing data, then you will need to look into establishing a clustered or shared file system to get the maximum I/O throughput to all the instances of the OS, especially if they need to do writes to this file system.

- Understand that if you set up soft partitions on your large SMP computer, that these partitions share computer resources with the other partitions. To date, we have not found good tools to help us determine what software partition is using what computer resource (that is, memory) and this makes it very hard to diagnose what might be causing performance issues.

IN 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.

RESOURCES

Brown, Tony. 2001. "A Practical Approach to Solving Performance Problems with the SAS System". Dallas, TX: SAS Institute Inc. Available at support.sas.com/rnd/scalability/papers/solve_perf.pdf.

Brown, Tony. 2006. "Solving SAS Performance Problems: Employing Host Based Tools*".* Dallas, TX: SAS Institute Inc. Available at support.sas.com/rnd/scalability/papers/TonySUGI31_20060403.pdf.

Brown, Tony. 2008. "SAS Performance Monitoring – A Deeper Discussion*.* Dallas", TX: SAS Institute Inc. Available at www2.sas.com/proceedings/forum2008/387-2008.pdf.

Crevar, Margaret & Brown, Tony. 2007. "Best Practices for Configuring Your I/O Subsystem for SAS®9 Applications". Technical Papers and Presentations. Cary, NC: SAS Institute Inc. Available at support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf.

Crevar, Margaret & Brown, Tony. 2007. "Ensuring you Have the Proper Resources for your SAS®9 Applications". Technical Papers and Presentations. Cary, NC: SAS Institute Inc. Available at www2.sas.com/proceedings/forum2007/203-2007.pdf.

Hall, John. 2002. SAS *Application Performance Monitoring for UNIX.* Cary, NC: Hewlett Packard Company. Available at www.sas.com/partners/directory/hp/sasapp.pdf.

SAS Publishing. 2004.SAS *9.1.3 Intelligence Platform: Planning and Administration Guide.* Cary, NC: SAS Institute Inc. Available at support.sas.com/documentation/configuration/iaplanning913.pdf.

SAS Publishing. 2006.*SAS 9.1.3 Intelligence Platform: System Administration Guide.* Cary, NC: SAS Institute Inc. Available at support.sas.com/documentation/configuration/bisag.pdf.

ACKNOWLEDGEMENTS

Many thanks to Tony Brown, BJ Mulligan, John Turnley, Simon A.R. Williams, and John Maxwell for their review of the content of this paper.

CONTACT INFORMATION

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

Margaret Crevar
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Work Phone: 919-531-7095
E-mail: Margaret.Crevar@sas.com
Web: www.sas.com