How to Maintain Happy SAS® Users

Margaret Crevar, SAS Institute Inc., Cary, NC

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 when working with the IT administration staff at a SAS customer's location with regard to what they can do to maintain happy SAS users, and that is to ensure that underlying hardware is properly configured to support the SAS applications. This is not a trivial task since different SAS applications need to have the hardware configured differently and depending on where you are with your understanding of how SAS will be used will help you evaluate options for the hardware, operating, and infrastructure (mid-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 will:

- discuss briefly how SAS works, especially from an IO perspective
- give some guidance on how to initially configure hardware for SAS usage
- give some guidance on how to monitor the hardware to avoid running out of a computer resource
- discuss if you should run all your SAS components under a single operating system or split them across multiple operating system

This paper pulls together information that has been presented in recent SAS Global Forum and SUGI papers. We hope that by putting the information into a single paper it will help the IT administration staff who have to SAS at your location better understand how SAS works and what they may have to do to keep their SAS customers happy.

HOW DOES SAS WORK

SAS differs from a traditional **relational database** (RDBMS). With an RDBMS, it is common 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, each SAS user starts their own SAS session for each SAS job/application they are running. With the new SAS®9 Intelligence Platform Architecture, there are also several SAS servers that are started to support the Java clients, in addition to backend SAS server/processes that run for each active SAS user.

SAS creates a high volume of IO, 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 do some random access of data, but for the most part the SAS workload can be characterized as predominately large sequential IO requests with high volumes of data. One thing to note is 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 IO subsystem needs to be tuned for random access rather than sequential access.
- SAS does not pre-allocate storage when SAS initializes 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 datasets and individual partitions within an SPDS table do not span file systems.
- Reading and Writing of data is done via the OS's file cache. SAS does not use direct IO by default. NOTE: Since SAS uses the OS's file cache to read and write data, the maximum IO throughput rate is restricted by how fast the OS's file cache can process the data. The maximum IO throughput rate via file cache we have seen has been around 1GB/sec, but it generally is around 600MB/second per operating system instance.

- A large number of temporary files can be created during long running SAS jobs. These files are created, may be renamed towards the end of the task, deleted, and/or potentially manipulated many times within a long running SAS job. The size of the files may range from very small (under 100MB) to very large (in the 10s of GBs) on average. The location for these files is referred to as the SAS WORK area. You need to make sure that SAS WORK is pointing to a file system that has enough IO throughput to support your users' IO demands as well as enough disk space to support all the temporary files that will be created during each SAS session.
- SAS creates standard OS files, for its data store (SAS datasets, indexes, etc.).
- When executing writes, there is a single writer thread per SAS session. Some SAS tasks that support threading do start multiple reader threads though.

Configuring your hardware to support the profile of your SAS applications can make a significant impact on overall performance. When your system is computer is having to support multiple SAS applications (especially if they have a different access pattern), you may need to make tradeoffs (performance versus price is the key one).

GUIDANCE ON CONFIGURING HARDWARE FOR SAS

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 IO subsystems, file systems and memory to best supports 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 admins can then use the general guidelines in this paper to setup a UNIX or Windows computer to support SAS jobs and/or a typical SAS Intelligence Platform deployment.

There are some general configuration considerations for IO, memory and file systems that apply for all SAS environments. Let's review this before we go into specifics for legacy SAS jobs and SAS9 Intelligence Platform implementations.

- As mentioned in the previous IO section, SAS sessions create lots of temporary files and the location of these files is referenced as SAS WORK. SAS WORK needs to be moved from the default location (defined in the sasv9.cfg file) on UNIX servers to a directory that is large enough to handle all the temporary files created during the SAS sessions and this directory needs to have the IO throughput required of the SAS applications. On Windows systems, during the install process, you need to override the default location listed in the install window with a folder that has similar qualities. By default, in both cases, SAS will point to a location that is on a file system that is shared with the operating systems swap files. This file system tends to be heavily used by the operating system to write temporary files and can not support the requirements for SAS WORK.
- Because of the way SAS uses memory (both directly by the SAS process and indirectly when the operating system uses file cache as described above), we recommend that the swap file space be at least 1.5 times the size of physical RAM in the computer.
- The various file systems (often referred to as LIBREFs by SAS users) need to be configured on file systems on a computer that can sustain the overall IO throughput required by the SAS application. Some general guidelines on what the IO throughput needs to be can be what RAID level to use and what tool to use to create these files systems found in the Best Practices for Configuring your IO Subsystem for SAS9 Applications paper (support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf)

Let's talk a little more regarding some general guidelines for the two types of SAS applications mentioned above. Please note that there are lots of SAS applications, but these tend to be the most popular ones used currently.

BATCH SAS JOBS

Batch SAS jobs are individual SAS processes that tend to have been around for several years (releases of SAS). As the name implies, they are run as a 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 SAS jobs as legacy since they are written using the SAS programming language that has been around for many SAS releases; in fact many of them may have been written using earlier versions of SAS. These legacy batch jobs read data from an external source(s) (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, or

there can be hundreds of steps that are required in order to analyze external data source(s) and produce the desired reports and data marts.

The steps in SAS jobs use different computer resources.

- The data manipulation tasks (i.e. SAS DATA steps and SAS procedure steps, such as PROC SORT, PROC SUMMARY, and so on) tend to be very IO intensive in nature and heavily access the SAS WORK area. In addition, some steps that manipulate data (i.e. 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 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*. To maintain happy SAS users, you will have to decide between a single, large file system for SAS WORK that all the SAS users will share or multiple, smaller file systems (not different directories on the same file system) for SAS WORK that you round-robin your SAS users between.
- The analytical tasks (i.e. 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. 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 the CPU cycles to support them along with your other SAS users.
- The matrix manipulation tasks (i.e. IML, GENMOD, MIXED) the use of formats and hash table lookups tend to be very memory intensive in nature. 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.

SAS9 INTELLIGENCE PLATFORM

SAS9 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 new business user-focused applications include SAS® Data Integration Studio (which produces ETL flows), SAS® Enterprise Guide® (which front-ends 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 SAS9 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.

A SAS® Metadata Server was introduced to maintain all the information about the underlying data files used by the above applications in a centralized location. The SAS Metadata Server keeps the data (referred to as a repository) the SAS applications register to it in an in-memory table. You need to ensure there is enough physical RAM in the computer to keep this in-memory data base resident in physical RAM. Once paging starts because the physical RAM has been exceeded, then performance degradations start. Please note that 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 can not be more than 2GB (maybe smaller) since this is the limit to the amount of memory that a 32-bit version of SAS9 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 look into migrating your SAS Metadata Server to a 64-bit operating system.

A multi-tier architecture with a Java “mid-tier” is needed to support the Java -based SAS9 Intelligence Platform applications. How you configure a Java Applications Server and a Web Applications server can impact SAS performance. The SAS9 Intelligence Platform Administration Guide has some suggestions for parameters that need to be tweaked with regard to the various third-party mid-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 defaults settings for these components).

For the backend SAS servers (workspace, stored process, OLAP), you should setup the hardware for these servers the same way you setup the hardware for the legacy batch jobs discussed above since the SAS clients are just submitting SAS code to these backend SAS servers in a similar fashion to the way the legacy batch SAS jobs

submitted code, the only difference is you may 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 size the hardware capacity needed

Understand 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 adhoc users running reports and queries, ETL processes for warehousing, SAS Business Intelligence (BI) Applications (Enterprise Guide client queries, Enterprise Miner model runs, OLAP), and a myriad of possible other SAS applications.

Identify the Most Resource Intensive Areas where these SAS Applications may Simultaneously Overlap

This means identifying the heaviest activity in each process (e.g. each job step), and building a picture of what the overlap looks like when the processes were simultaneously running , and 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 Calculate the "Demand Load"

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. The following are some guidelines that can help you in determining how many resources you may need:

CPU – Using the new processors that are running over 2 GHz:

- For light report reading, or light query work (files < 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 and Data Mining, you can plan on 1-2 users per CPU. Please note that 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 Gigabytes of RAM per CPU. 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 IO, it can go a ways towards it.
- For 32-bit Windows systems, a maximum of 4GB is ideal. This can be increased to 8GB on a 4+ processor computer.

IO – 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 SAS-related file systems:

- It is recommended to have your Storage Administrator set your file systems up according to required IO throughput first and capacity needs second. Capacity will be determined by how many people generate how much work in the file system at once. A very simple rule of thumb is to calculate the reads and writes, and use a 3x figure for things like SORTS or other procedures that manipulate or calculate data (e.g. FREQ, UNIVARIATE, MERGE, etc.). This won't be exact but you will be in the ball park.

- For file system throughput you can at a minimum consider the following:

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

- It is generally recommended that a minimum of three file systems be setup 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 below SAS file system characteristics and locations as a reference especially if the disk must be shared between SAS applications and/or users. The system administrator or installer should avoid sharing these heavily-used IO file systems with other applications (whether these applications are performing heavy IO or doing random access to the data) to avoid IO conflicts between SAS and these other applications.

- Certain SAS application may require additional file systems to hold source data, user playpens, SPDS tables and SPDS temporary tables, and other SAS output (secondary temporary files referred to as UTILLOC, OLAP cubes, WRS Query Cache, etc.)

- It is good to make sure the heaviest-used file systems (usually SAS WORK, permanent storage, UTILLOC, etc.) will be physically separate (e.g. on different volumes/LUNs and physical underlying disks) from each other. Again, consult with your Storage Administrator to help with this.

Please note that the Best Practices for Configuring your IO Subsystem for SAS9 Applications paper (support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf) that we have referred to goes into more details regarding the RAID levels to use and preferred tools to use to setup these file systems.

If you are planning a large system or expansion, it may 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.

Please note that the above is for each SAS process running simultaneously on the computer and 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.

GUIDANCE ON CONFIGURING HARDWARE FOR SAS

Now that we have discussed how SAS9 applications work in general and to do a simple sizing for the hardware, there are some general configuration tuning issues for the hardware you will be running your SAS9 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 IO Subsystem for SAS9 Applications paper (support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf)
- Run 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 IO bottlenecks with the processing, the amount of memory needed to run the application, etc. More details on how to interprete this information can be found in the A Practical Approach to Solving Performance Problems with the SAS System paper (support.sas.com/rnd/scalability/papers/solve_perf.pdf). NOTE: The FULLSTIMER does not report the same details of information when run on Windows servers as it does on UNIX servers.
- Give the SAS applications all the memory they need: If after running the job with FULLSTIMER, you see you are using 96MB of memory the entire time (96MB is the default size for the MEMSIZE SAS parameter

on UNIX), you may see a performance improvement if you increase the amount of memory the SAS session can have by increasing the parameter on the -MEMSIZE SAS parameter. NOTE: MEMSIZE is set at the invocation of the SAS session.

- Balance MEMSIZE vs physical memory: The optimal memory configuration is to have to have the sum of the MEMSIZE values of all the simultaneous SAS session fit into physical memory.
- Configure 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.
- Modify the SAS applications to reduce the amount of IO: There are two ways to reduce the amount of IO 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, etc
    - Reduce the number of data accesses by processing more data each time the disks are accessed by upping the parameters of the BUFNO, BUFSIZE, CATCACHE and COMPRESS SAS options.
- Configure for large blocked sequential IO: For the most part, the default BUFSIZE that SAS chooses for a file is best (especially when using indexes with high cardinality). We highly recommend that you operating system or external storage commands to specify the file systems being used by SAS have a large stripe size (at least 64K, but not larger than 256K). 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. This value is set when the SAS data file is created and used every time the file is used.
- Separate input file systems form 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 to the former; 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 an IO bottlenecks.
- 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 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 data being analyzed/reported on growing 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 just that, 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. Here is a list:

- An SAS Global Forum 2007 paper entitled Ensuring you Have the Proper Resources for you SAS9 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. 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's OpenView Operations, IBM's Tivoli Enterprise Console, and Microsoft's Operations Manager. Each integration package is customized for a specific product, but the overall objectives for managing and monitoring the SAS®9 environment are consistent across the various products. The integration packages will include the following attributes:

- Packaged suite of scripts, tools, monitors, and documentation which describes 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, userID 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/servers and 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 operating system (OS) will glean your SAS users the best performance. Here are some situations to consider:

- It is very common for existing SAS customers to setup 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 IO throughput required exceeds what the OS's file cache can provide. When this happens, your SAS applications will become bottlenecked by the IO throughput rate of the OS's file cache. There are two ways to overcome this.
    - The first being dividing 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 partions (LPARS, zones, containers, etc.) 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.
    - The second being seeing if there is a way to do direct IO (avoid using the OS's file cache) on the computer you are executing your legacy SAS jobs – this can be done with the SGIO SAS parameter on Windows computers, with the –dio option when setting up JFS2 file systems on AIX and by implementing a Veritas file system with direct IO enabled.
- As we have described the many new components to the SAS Intelligence Platform applications, you may have noticed that certain components are more CPU and IO 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 since 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 is 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 IO throughput to all the instances of the OS, especially if they need to do writes to this file system.
- Understand that if you setup 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 (i.e. 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 users' 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 in developing, improving, and maintaining high-quality operating environment performance.

RESOURCES

Brown, Tony. 2001. "A Practical Approach to Solving Performance Problems with the SAS System". Cary, NC: 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". Cary, NC: SAS Institute Inc. Available at support.sas.com/rnd/scalability/papers/TonySUGI31_20060403.pdf.

Crevar, Margaret & Brown, Tony. 2007. "Best Practices for Configuring Your IO 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". Cary, NC: Hewlett Packard Company. Available at www.sas.com/partners/directory/hp/sasapp.pdf.

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