

Getting the Most Out of Your High-End UNIX and NT Server with SAS®

Margaret Crevar
Corporate Technology Center
SAS Institute

Leigh Ihnen
Numerical Architecture & Performance
SAS Institute

Gary Mehler
Technology Strategy
SAS Institute

Introduction

Getting the most out of high-end servers requires understanding the hardware and operating systems that are present on these systems. Most of this revolves around installation configurations and subsequent tuning based on these configurations. This paper will cover general tuning and installation issues as well as information on advanced tuning for operating system and hardware issues.

General Tuning Issues

Without understanding how you are going to be using SAS software, it can be difficult to tune your hardware to support SAS or other complex applications. However, there are a few basic principles that should be looked into first before more involved tuning investigations get underway. In fact, in many cases, understanding and satisfying the following issues can lead to a large improvement of overall SAS performance.

- Ensure you have as balanced a system as possible. *Balance* in this context means that the I/O subsystem is well matched to the processing capability of the overall system and has sufficient bandwidth based on the overall capacity of available storage.

An example of a **balanced** system is one that has fast disk I/O to match fast processor throughput.

An example of an **unbalanced** system is one that has very fast processing abilities (i.e., multiple processors) with a single very large single disk drive for all system I/O needs including the operating system, applications, and user data.

Another type of unbalanced system is one that has high-performance disk subsystems spinning media on multiple spindles managed by multiple disk controllers on a low throughput bus subsystem like a basic desktop PC bus.

The basic principle behind the balanced system principle is that you don't want to

spend too much for any system component that you won't be able to effectively utilize. A balanced system makes maximal and equal use of the various components in the system.

- Have the SAS application read from one set of disks and then write to another. Do not allow them to do all their processing in SASWORK, if possible. By default, SASWORK is used by many procedures, and if various steps are required, these procedures will end up successively reading and writing to the same disk area, causing unnecessary disk traffic which leads to much slower performance in many cases. A related scenario is one in which SASWORK is located on the same disk partition as the original data library, which could mean even more disk traffic.

Remember that the ideal configuration for maximum disk throughput is one in which a partition can be used for a single purpose at any given time. A read-only data library on a partition is a good thing if data are separately written to another partition elsewhere. Disk caches in hardware and in operating systems can mitigate this problem for many applications that have smaller I/O profiles, but many large SAS jobs have a tendency to overload and overwork this support. After all, it is typically geared towards smaller workloads more typical to other types of computing that isn't as demanding on hardware as high-end decision support and data warehousing.

- Move SASWORK to a separate set of disks from where the operating system's swap files are being written to. This is similar to the previous item, but may not be as obvious since swap files aren't usually a concern for many users. However, in large-scale processing like one might see with large SAS jobs, high memory use can lead to unusually heavy operating system activity to swap files, leading to greatly decreased throughput on that disk partition. Again, this

might not be a problem for smaller workloads, but for high-end I/O for very large files, the effect can be major.

- Make sure there is enough memory on the server to support all the applications running. Don't forget that the memory on the server is being used by the applications as well as by the file cache (which is generally controlled by the operating system). For typical workloads, abundant file cache memory is what leads to generally acceptable system performance to support a large number of users. A large application memory load that exhausts available memory also causes the disk cache to become ineffective.

Configuration Questionnaire

The basic set of general configuration issues now behind us, we can look into specific issues revolving around your particular data, application, and processing requirements. A basic understanding of these issues will provide the information we need to maximize throughput and return on the hardware investment you've previously made. We'll use the format of a questionnaire to address relevant concerns

General Questions: Size

Determine the size of the data files that will be analyzed

- What is the data store? Generally, the best-optimized files for processing typical in decision support applications are SAS datasets, which are optimized for streaming throughput. Other databased stores have their own behavioral characteristics when used for this type of application; these characteristics need to be understood
- What is the total size of the data? Massive amounts of data may require large amounts of processing time to process.
- Will analysis be done on one large file or multiple files? Multiple files can lead us to consider parallelization opportunities, and can tend to be less fragmented than extremely large ones. In addition, multiple files can be

spread across separate disk partitions to spread the user load across more hardware.

- What are the sizes of the files being used? Can we equally spread the user load across multiple disk partitions solely based on similar-sized data partitions?
- What is the expected growth of the data? Solving today's problems on a daily basis gets tiring. Planning for growth is always desirable if we have an understanding of the growth profile of various parts of our system.

General Questions: SAS Usage

- What SAS products are installed or being considered? Different products have different workload characteristics and data requirements.
- What types of processing will you be doing with SAS? Streaming over vast quantities of data to do simple summary statistics is a very different workload scenario than performing very complex modeling over smaller quantities of data.
- How many steps are in a typical SAS job? These steps are typically executed sequentially, so causing the number of steps to increase dramatically can have a large impact on overall runtimes.
- Will data manipulation be followed by data analysis and/or reporting? Related to the number of steps, an understanding of the data flow and data dependencies can help you understand how to organize for maximum throughput.
- Will data be extracted from one large file or will smaller files be combined for reporting? In some cases, pulling data together prior to reporting can be a costly undertaking. Particularly if prior steps can be spread across different disk partitions, combining data into a large single file can be a throughput bottleneck.
- Where will the application data reside? Being stored locally or on high throughput server storage is preferable to remote storage across an overloaded, slow local area network.

- How much historical data needs to be stored? This can have a dramatic effect on how wisely available storage can be used since historical data can greatly limit your ability to use your storage resources effectively.
- What is the total storage requirement for the system? This includes space for other applications and their data, operating system and swapfile needs. Hopefully, enough storage for various types of applications and data are available, including high throughput and archival storage.

General Questions: Server Utilization

- Will other applications be running on the server? This is an easily forgotten issue, but other applications such as databases, web servers, or email servers can have their own heavy requirements on the server hardware.
- How many SAS users are there? Depending on the expected workload, this can have a dramatic effect. Users who are high-end power users are more likely to overtax limited server resources.
- How many concurrent SAS sessions? (# of SAS users * number of sessions) Keep in mind that some complex activities may require multiple SAS processes to achieve (such as long-running batch jobs that run which high-priority interactive processing needs to occur). Also, server processes (like SAS/IntrNet[®] or SAS/SHARE[®] servers) need to be factored in as well.
- How will the client machines access the server? (via client/server SAS sessions or an X-emulation session) Client/Server sessions (such as with SAS/CONNECT[®]) might have lesser graphical user interface throughput needs, but heavy throughput requirements depending on the type of processing can also tax a server.
- What is the expected growth of the user community? Like planning for data growth, this needs to be taken into account to avoid too quickly overtaxing available server resources.

Determine Server Configuration—NT or UNIX

- Do you have a hardware vendor preference? If so, understanding the offerings from your vendor(s) will help you understand how to make the best investments to meet your application needs.
- Do you have an existing server they want to use (how is it configured)? The following items should be documented and understood, as they will come up during installation and tuning of your application software. You should be aware of your system's:
 - Model
 - Number of processors
 - Processor speed
 - Amount of physical RAM and swap file size
 - Network media and speed
 - Current I/O subsystem layout
 - Any current performance problems
 - How are the file systems created on the disks?

Sizing for Solutions

In addition to the general issues just covered, various SAS solutions have their own requirements on the underlying hardware. A few of these will be discussed to enumerate which aspects need to be considered.

For **Data Warehousing solutions**, you should consider:

- Your overall data warehouse architecture for database creation and query
- Data content and data preparation
- How data will be loaded and stored for user access
- Whether users will have read-only or update access
- Will users access shared data on the DW server or on a Data Mart server?
- Are there any problems with the current Data Warehouse SAS analysis system?
- What is the current system availability and contention?
- Run times for the longest-running data preparation jobs
- Timeliness of data and data accuracy

For **Data Mining solutions**, you should think about:

- How many Enterprise Miner® (EM) projects will be running at any given time?
- On the average, how many per day?
- Is there a peak workload period?
- How many analysts will be executing EM activities?
- What volume of data will the EM projects be accessing?
- How many variables will be used for analysis?
- How many unique values for each variable used?
- Will data preparation/cleaning jobs be running during data mining executions?
- Is there a need to store both active and inactive EM projects? How many?
- What is the expected response time of the users?

For **Web-based applications**, you need to consider:

- Report distribution, application distribution; via thin-client, HTML, SQL?
- Will users select and generate reports dynamically?
- Will they be doing graphs or static report viewing?
- What data store will the dynamic reports run against?
- What is the number of reports involved? What is the size of reports?
- How many users are involved?
- Are there peak processing times?
- Will SAS/SHARE be used?
- Will the same server host both the Web processing and the application processing?

Storage - Why does it matter?

Now that you have an understanding of how SAS is to be used, the next area we need to spend time on is how to layout the disks on the server. This is the area in which many performance issues are based, so we need to spend the most time here.

A basic principle used to give SAS the best performance is by trying to avoid I/O wait states. However, there are other factors that need to be considered too. If the data is very important, then

we may want to add redundancy to the way we lay out the disks. This may add time to the length of the SAS job, but if the data is always available, even if a disks fails, that is something you might need to take into account.

The redundancy tools used to enable disks (such as RAID) to always be available causes the total capacity of a set of disks to decrease up to 50% due to redundant storage and for recovery information. Because of this, you need to remember that the amount of disk space includes your known space needs as well as the redundancy overhead you choose to use. Needless to say, cost will generally drive what type of storage and the amount you choose. However, you get what you pay for.

To get capacity, performance, and reliability, we can use JBOD (Just a bunch of disks) or RAID, so we'll look into the trade-offs of each.

RAID is a commonly used way to meet storage needs for large enterprise applications, so we'll look into RAID, which is a:

- **Redundant**—part of the disk's storage capacity is used to store *check data* that can be used to verify data or recreate missing parts of it
- **Array**—*control software* presents disk capacity as a set of coordinated *virtual disks* **of**
- **Inexpensive and Independent**—disks are normal disks that could function independently of each other. If one fails, the others should continue working.
- **Disks**—the storage devices comprising the array are on-line storage

RAID is commonplace in server computing environments. It is typically incorporated by disk subsystems to enhance I/O performance and data availability. Software, or host-based RAID is available from several vendors, such as VERITAS. The four storage-related areas we'll need to consider are listed below. This is much more involved than just thinking about raw disk capacity.

- **Performance** - I/O wait decreases it
- **Availability** - What happens if a disk fails
- **Capacity** - Partially based on availability consideration
- **Cost** - A function of all of the above

The range of storage configurations have their own trade-offs, which are listed below.

JBOD - Just a Bunch Of Disks

- *Performance:* Can be spread across multiple disks, but hand tuning of file placement is typically required
- *Availability:* If a disk fails, only the files on that disk are lost
- *Capacity:* Additive for the number of disks involved
- *Cost:* Additive for the number of disks involved (1:1 ratio with capacity)

Striping - spreading a file over multiple disks (also known as RAID-0)

- *Performance:* Can improve throughput across more disk spindles for greater throughput
- *Availability:* if one disk goes, all data on all stripe components is lost
- *Capacity:* Additive for the number of disks involved
- *Cost:* Additive for the number of disks involved (1:1 ratio with capacity)

Mirroring – a complete second image of the data on another disk or array (RAID-1)

- *Performance:* Can be slightly faster due to choice of section to read or write
- *Availability:* If a disk fails in one part of the mirror, the other half is used
- *Capacity:* A heavy price is paid for the extra copy: 50% of the capacity is not available for general storage.
- *Cost:* A 50% overhead for the redundancy must be paid.

RAID 5 - Parity information is written to one of the disks in the stripe set.

- *Performance:* slightly worse write performance than RAID 0 due to extra overhead of writing extra parity information
- *Availability:* If a single disk fails all information can be recovered from the parity information and rebuilt with some performance impact during the rebuild process
- *Capacity:* Only one disk in the stripe set is sacrificed to store the redundancy (parity) information
- *Cost:* Only one extra disk per array needs to be factored into the cost.

RAID 1+0 - data is striped across mirrored pairs of disks

- *Performance:* Could be slightly better than RAID-0 due to availability of two disk areas

- *Availability:* both disks of a mirror pair must fail before data is lost
- *Capacity:* 50% overhead of RAID 1 for the mirrored area
- *Cost:* Like RAID 1, a 50% overhead for the redundancy must be paid.

In addition to understanding the type of redundancy, we need to consider how the storage will be attached to the server. In addition to the traditional notion of file servers attached via standard local area networks, we have other storage options available. Various vendors have offerings in of these types.

- **Host attached**, or traditional Internal scsi or fiber-channel disk drives. For maximum effect, a volume manager assigns file system storage to logical devices . An example is Veritas Volume Manager – VxVM, which supports RAID 0, 1, 1+0 and 5. For more information, see the web at:

eval.veritas.com/webfiles/docs/RAIDirectorWP.pdf OR
www.veritas.com/us/products/volumemanager/whitepaper-04.html

- **Network Attached Storage** such as a Network Appliance, which runs a file system protocol like NFS and CIFS which transmits data via the network
- **Storage Area Network**, storage devices that are attached via Fiber-channel to host systems. These have the performance benefits of host-attached storage and the separate scalability and sharability benefits of Network-attached storage.

Array-attached storage examples include:

- EMC Symmetrix [supports RAID-S(RAID 5 variant), RAID 1 – large on array cache]
- SUN T3 [RAID 1+0,5]
- HP XP series [RAID 0/1,5 - large on array cache]
- IBM ESS(Shark) [RAID5 - large on array cache]
- IBM Fibre Channel [RAID Server RAID 0, 1, 0+1, 3, 5]
- StorageWorks [0, 1, 0+1, 4, 5, and non RAID disks]

For assistance with array-attached storage, various

tools are available from their respective vendors, including:

- HP ShureStore Command View
- EMC ControlCenter Workload Analyzer, which is a performance analysis tool that collects, archives, graphs, and analyzes Symmetrix and host OS historical performance data and can provide the user with this level of detailed performance information

Using tools like EMC Control Center Resource View, administrators can

- Layout filesystems across different physical spindles, avoiding disk contention
- Obtain a front to back view of logical volumes and file systems all the way to actual physical disks.

Memory

Once storage needs are determined, the next area to look into is memory. Various methods are possible for limiting overall memory consumption, such as setting the size of system swap space, setting a process memory limit with the ulimit UNIX command, using the SAS MEMSIZE option and using the SAS SORTSIZE and SUMSIZE options. There are tradeoffs that may need to be made to minimize overall memory use since we of course want to avoid unnecessary memory swapping if at all possible. While the file cache improves access to storage, too much memory committed to the cache can cause applications to swap, leading to even more unnecessary I/O and slowing things down further.

What size server do I need?

Now that we have an understanding of storage and memory tradeoffs, let's look into the general question of how much capacity is needed to support SAS processing. This will take the form of another set of questions and their answers for the general SAS processing case

How much memory?

- plan on at least 32 Megabytes RAM per SAS session
- Exceptions:
 - MDDBs—will need enough RAM to load the largest table in the cube while building the cube and possibly while using the cube

- Heavy Statistics—will need enough RAM to load the statistical model.

How many processors?

- 8-10 SAS sessions per processor for typical workloads
- Exception is heavy statistics--will need enough RAM to load the model

How much disk space?

- Enough to hold your SAS data sets
- Enough for your SASWORK temporary directories
- The SAS System itself: ~700 MB

Tuning parameters

- Move default location of SASWORK
- Run the SAS jobs with the FULLSTIMER option turned on
- Give the SAS applications all the memory they need
- Modify the SAS applications to reduce the amount of I/O
- Monitor CPU for I/O bottlenecks
- Balance the MEMSIZE against physical memory

File system suggestions

- Configure for large blocked sequential I/O
- Separate input file systems from output file systems for SAS steps that act as data filters
- Set up SASWORK on its own file system.

Aside from the general guidelines listed above, here are details for some typical configurations for reference purposes.

Typical DW Configuration—where the data resides in the Data Warehouse and the users query the data and either do reporting or downloading the data to their workstations

- One processor for every 8-10 SAS sessions
- 32 MB of RAM per SAS session or 512 MB per processor
- Enough disk space for the Data Warehouse, the SAS System, and temporary SASWORK space
- I/O intensive application

Typical OLAP configuration—where the OLAP application resides on the server and is accessed via a SAS GUI and the data primarily resides in MDDB tables

- One processor for every 8-10 SAS sessions
- 32 MB of RAM per SAS session, unless the largest MDDB table is larger than 32 MB or 512 MB per processor
- Enough disk space for the data, the SAS System, and temporary SASWORK space
- Memory and I/O intensive application

Typical Data Mining configuration—where the data resides on the server and the users will be doing heavy analysis against these large files

- One processor for every 4-5 SAS sessions
- 128 MB of RAM per SAS session or 512 MB per processor
- Enough disk space for the data, the SAS System, and temporary SASWORK space
- CPU, memory, and I/O intensive application

NT Server Configuration.

Keep in mind that various configurations are available on Windows NT, and you'll need to understand what role will the server play in SAS processing?

- **Scenario 1:** Client/Server—the SAS application will process on both
- **Scenario 2:** File Server (NT or Novell)—SAS processes on the client, server is like an external hard drive
- **Scenario 3:** Data Server—SAS processes on the server only, data accessed from client via ODBC
- **Scenario 4:** Server Application—SAS processes on the server, data accessed from client via NT emulation or EG
- **Scenario 5:** Web Application/Data Server—SAS processes on the server, data and application results accessed from client via Web browser

Diagnosing and Correcting Performance

Problems

Assuming that the earlier information in this paper has all been taken into account, there are ways to use tools to further understand and correct

performance problems you still might be experiencing.

The SAS option FULLSTIMER gathers application data from SAS program execution. The items of most interest are the times and amount of memory used. For UNIX, the statistics are gathered from the getrusage system call, and on NT, the statistics are gathered from standard NT APIs. Below is sample FULLSTIMER output from a UNIX server:

NOTE: PROCEDURE PHREG used:

time:		memory:	
real	3:06.380	page faults	0
user cpu	2:34.728	page reclaims	0
system cpu	27.287 seconds	usage	30.99 M
block I/O operations:		context switches:	
input	1	voluntary	521
output	7	involuntary	2036

Main reasons for differences in real time and user+system time include CPU contention with other processes, paging or swapping, I/O contention with other processes, network latency, waiting for an operating system lock, or having to share other system resources with other processes. If you are seeing a large differential between the wall clock time and user+system time, there is a potential for performance improvement. The critical next step would be to identify which factors are the root cause(s) for the difference. These factors are often a combination of a hardware configuration limitation and/or application inefficiency.

For the section on context switches, voluntary context switches represent wait states on a resource, so high numbers are not necessarily bad while involuntary context switches represent a resource constraint, so a high number can be bad.

In addition to FULLSTIMER, a number of external tools are useful to assess potential performance problems. For UNIX these include vmstat, iostat, sar, top; for NT Task Manager and Perfmon are available. Third Party tools available include proctool (Solaris only), glance+ (HP-UX, Solaris, AIX), and openview (HP-UX).

Specific Issues by Platform

On Sun servers, investigate Priority Paging. With priority paging turned on, one is able to set a boundary around the file cache so that file system I/O does not cause paging of applications. You can determine your server configuration with the UNIX command:

`/usr/platform/xxx/sbin/prtdiag`, where xxx should be replaced with the output from the `uname`

-m command and `/usr/sbin/prtconf -v` | more. For more details, see www.sas.com/partners/technology/sun/performance/index.html

On HP servers, you can determine the server configuration with `sam`, the system administration tool that ships with HP-UX and with `glance+`. For more details, see

www.sas.com/partners/technology/hp/hpuxwp.pdf

On IBM RS/6000 servers, you can determine server configuration (processor, memory, I/O adapters, and disks) with `lscfg -vp`; logical volume information via `lsvg -o`, and more information on drives with `lslv xxx`, where `xxx` is the logical device name.

SAS Application coding issues

When all else fails, SAS application re-coding may be needed to improve performance. The following six areas are good areas in which to look for possible correctable situations.

Read and write data selectively. I/O is often the largest single component of program execution time.

- Use DROP/KEEP to eliminate unnecessary fields in data sets.
- Use SHAREBUFFERS on infile statements to eliminate the need for separate input and output buffers.
- Create indexes when appropriate:
 - o the data set is relatively large
 - o data set not frequently updated
 - o data frequently subset by values of the indexed variable
 - o data is uniformly distributed
 - o result sets are usually less than 30% of entire data set

Execute only the SAS datastep statements you need, in the order you need them. Examples:

- Use mutually exclusive conditions - IF-THEN/ELSE instead of IF-THEN
- Write conditions in order of descending probability


```
/* China is most likely */
select(country);
  when('China') output
    china1;
  when('India') output
    india1;
....
```

- Use IN operators rather than logical OR operators

if street in ('Maple', 'Elm', 'Willow') instead of

if street='Maple' or street='Elm' or street='Willow' then

- Use _TEMPORARY_ arrays rather than variables you DROP as they can reduce storage as well as CPU time based on their contiguous memory locations.
- Set lower bound of arrays to 0.

Take advantage of SAS procedures. Examples:

- Use PROCs if possible rather than creating your own data step.
- Use PROC DATASETS to copy datasets with indexes.
- Use WHERE conditions in procedures.
- Use SQL procedures to simplify code.

Know SAS System defaults. Examples:

- Reduce the storage space for variables. The SAS System uses 8 bytes for numeric variables. Storage can be saved if you were to shorten variables such as house number or building number.
- Use character rather than numeric variables. GENDER stored as a numeric 1 or 0 would be 8 bytes compared to a 1 byte character 'F' or 'M'.
- Avoid default type conversions - Use the PUT function to perform numeric-to-character conversions and INPUT function for character-to-numeric conversions.

Control sorting. Examples:

- Sort only when necessary.
- Sort as few observations and variables as possible.
- Use NOEQUALS if it is not necessary for observations within BY groups to have the same relative order. If you do not care about the ordering within each CITY grouping below, we specify NOEQUALS.

```
proc SORT data=big noequals;
  by city;
run;
```

Know your data and test programs. Examples:

- Examine raw data before reading them.
- Label variables and data sets.

Conclusions

This paper has listed a number of approaches for identifying and correcting configuration and performance issues with SAS application operation on high-end UNIX and NT servers. While there are a lot of good examples here, hopefully you'll also understand the methods typically used to locate and remedy general problems that can occur when doing large-scale computing on advanced systems.

Trademarks

SAS, Enterprise Miner, SAS/CONNECT, SAS/Intrnet, and SAS/SHARE are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand or product names are registered trademarks or trademarks of their respective companies.

References and Further Information

Compaq Computer Corporation: Ralston, Carl, "SAS installation guidelines for Digital UNIX", available at: www.sas.com/partners/technology/compaq/decunx1.html

EMC Corporation: "Taming the CRM Information Explosion: A collaboration between SAS and EMC for powerful customer relationship management", available at: www.sas.com/partners/technology/emc/dataexplosion.pdf

Hewlett Packard Company: "Hewlett-Packard Research and Development Update: Optimizing SAS software numerical applications for HP platforms", available at: www.sas.com/partners/technology/rd_update.pdf .

IBM Corporation: Olson, Keith and West, James, "The SAS System and GPFS – A Scalable Solution", available at: www.sas.com/partners/technology/ibm/Gpfsfina.pdf .

Intel Corporation: "The Impact of IA64 on the SAS System for Decision Support", available at: www.sas.com/partners/technology/intel/sasia64.pdf .

Sun Microsystems: Chew, Maureen, and Gilchrist, Patty, "Turbo Charging SAS Applications in Solaris Environments: Managing Highly Performing Applications in Large Multi-User Environments", available at: www.sas.com/partners/technology/sun/mgmt/index.html .

Author Information

Margaret.Crevar@sas.com
Leigh.Ihnen@sas.com
Gary.Mehler@sas.com

SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
(919) 677-8000