Paper 294-25

Tuning the SAS® System for UNIX and Tuning UNIX for the SAS System

Clarke Thacher, SAS Institute, Cary, NC

ABSTRACT

This paper will present tips and techniques for getting the best performance out of the SAS System on the UNIX operating systems. We will address changes that can be made to the SAS System as well as operating system parameters.

INTRODUCTION

Everyone wants the best performance that they can get with the least effort. We will explore some techniques that can be used to get better performance from SAS programs without changing the SAS source code.

WHAT IS PERFORMANCE?

Performance and performance problems boil down to three resources: **CPU**, **Memory**, and **I/O**. A lack in anyone of these can make any of the others look like the problem. For example: You have users complaining about slow interactive and shell performance. So you do some quick checking and you see that disk I/O is high and that a particular large batch job that you know does a lot of I/O is running. So you make the assumption that the I/O bound job is the culprit and tell your users to hang on for a little while until the job finishes. But, the job finishes and the problem does not go away. After a little further investigation you see that the I/O problem still exists but, it is actually due to paging and swapping, which indicates a memory problem not an I/O one. The real key to solving performance issues is knowing what your system looks like when it is healthy. This allows them to observe trends and to catch problems before they become severe enough to affect productivity.

CPU

The CPU determines how much work can be done by a machine in a specified amount of time. A faster processor means more work can be done in a smaller slice of time. CPU statistics describe how well utilized the CPU is. Ideally you want to see the idle time at almost zero, because then you know your machine is doing work and you are getting the money out of it that you paid for it. But, you need to be careful because a machine that has zero idle time and a high load average is going to be exhibiting poor performance. It has more work than it has CPU cycles to handle the work.

MEMORY

Memory defines how many jobs and programs can be using the CPU at once. A lack of memory can bring a machine's performance down the quickest of the three resources. Unix typically does two types of memory management: paging and swapping (there are other things like demand loading and shared executables and libraries that affect memory usage). When a Unix machine has gotten to a certain threshold of memory use as defined in the kernel it begins paging. Paging is simply taking the not recently used (NRU) memory pages and puts them on disk. Then when the machine is getting close to the point of actually running out of physical memory it starts swapping. Swapping takes whole programs out of memory and puts them on disk. So you can see that if you are trying to stuff more and more programs into memory that your machine will slow down simply due to the fact that you have more disk I/O going on.

I/O

Ideally, you want the I/O subsystem to be available when a user program asks for it and not to have to wait for any other program's I/O. Seeing a lot of jobs that are waiting for I/O can point to a problem with how the file system is organized. I/O shortages do not typically cause performance problems. I/O wait times and contention are where the I/O subsystem can cause performance problems. Memory shortages and Disk I/O Contention can have an awful domino effect on performance. The largest problem with the I/O system is the huge difference between its speed and that of the CPU. This fact alone causes longer response times and higher CPU idle percentages.
The two most common types of I/O are disk and network. In general, disk I/O will be faster than network I/O. Of course there are exceptions, there are some very fast network filesystems that can be faster than an overworked local disk drive.

DISK

Many SAS programs are very I/O intensive. They will read large files containing raw data and SAS, creating SAS datasets and reports. For example: if you have a group of users on a machine and they all run SAS and they point their WORK libraries to the same file system. Depending on the type of disk drive this could cause all of those users to see a slow down in their programs because of the contention for the use of that one device.

NETWORK

Network I/O problems can wreak havoc on interactive applications. X windows applications can suffer the most because the X windows protocol is somewhat verbose. NFS - Network File System trying to operate over an error and collision prone network is just asking for trouble.

STIMER AND FULLSTIMER

The SAS System options **STIMER** and **FULLSTIMER** provide basic information about the performance of a program. This information can be used to determine the where the performance "hotspots" may be. **FULLSTIMER** will produce a report that will give many clues about a job's performance profile. The next page includes the output from a SAS System session. We will describe each of the fields in the FULLSTIMER log and how they may relate to the overall system performance.

```
1? options fullstimer;
2? data asia;set maps.asia;run;

NOTE: There were 111532 observations read
from the data set MAPS.ASIA.
NOTE: The data set WORK.ASIA has 111532
observations and 7 variables.
NOTE: DATA statement used:
      real time           2.17 seconds
      user cpu time       0.85 seconds
      system cpu time     0.49 seconds
      Memory                              73k
      Page Faults                         2
      Page Reclaims                       0
      Page Swaps                          0
      Voluntary Context Switches          4661
      Involuntary Context Switches        23
      Block Input Operations              2
      Block Output Operations             0
```

real time shows the elapsed time that that step took.

user cpu time shows the time that the step took in The SAS System. A step that was running a very compute intensive procedure such as a logistic regression would show a high user cpu time.

system cpu time shows the time that the step spent in the operating system. A step that is very I/O intensive will show a proportionally higher system cpu time.

Memory shows the amount of memory allocated directly by the procedure running that step. This will not reflect all of the memory that is used by the entire SAS System, since there are many components that can not be attributed directly to a specific procedure. ODS is an example of one of these subsystems.

Page Faults indicate the amount of virtual memory page faults that occurred while the procedure was running. A large number of page faults indicates that the virtual memory that is being used greatly exceeds the amount of physical memory that is available. Some operating systems, notably AIX, will use the system paging mechanism to perform file I/O. Page Reclaims and Page Swaps are additional metrics on the system paging activity.

Context switches happen when the process releases control to another process. Voluntary Context Switches occur when the process releases control to another process before its timeslice has expired. This usually occurs when the process makes a call for an operating system service, such as a file I/O. Involuntary Context Switches occur when the process time slice has expired, or when a process with a higher priority takes control. Context switches tend to occur every few milliseconds on average. A very high rate of involuntary context switches is an indication that the system may be overloaded with other processes.

Block Input Operations and Block Output Operations show the number of times that the operating system had to perform an I/O operation to service a read or write request from the process. Not every read or write to a file will cause a block I/O operation. Often, a filesystem will cache a recently read file into memory, and have it available for reading without having to retrieve it from the disk. Block I/O operations are not shown in AIX, because that system includes I/O operations in the paging activity.

SYSTEM TOOLS FOR MONITORING PERFORMANCE

W OR UPTIME

Provides you with basic system status. The most important information is the load average (or length of run queue) for the past 1, 5, and 15 minutes. If the number of jobs waiting on the cpu (load average) is high then performance will be poor.
A high load average depends on the class of machine, but, for your typical desktop workstation a load average of above 3 shows an overworked CPU.

PS

Provides information about the processes currently on the system. Depending on the options you specify it can give information on the size, the state, the location (in memory), the priority, and the amount of CPU time of a process. The output and options available for the **ps** command are different from system to system.

VMSTAT

Provides information about virtual memory use, cpu status, and the number and state of the processes currently active on the machine. With the basic command with default options you get output about how much physical memory is being used, how much paging and swapping is occurring, what percentage of the cpu is in user/system/idle, and how many running, blocked, or waiting processes there are. The output and options available for the **vmstat** command differs from system to system.

IOSTAT

Provides information on the I/O subsystem statistics. This command is very different from system to system. On some systems it provides terminal I/O and CPU statistics as well as disk I/O statistics.

NETSTAT

Provides information on the status of the network. Of particular use is **netstat -i** which provides information on each configured network I/O device and it's activity and number of errors and collisions. Poor performance with an NFS drive can be seen as an unusually high rate of collisions.

SAR (SYSTEM V)

Provides a wide array of system information with a great level of detail. sar stands for System Activity Report. It has two distinct modes of use. It can either pull information from a log file or it can gather real time information. Depending on the options can display information on CPU Usage, Disk I/O, Paging information, Free Memory and swap space, Swapping Statistics and others. Usually not on by default and must be configured.

ACCT - ACCOUNTING SOFTWARE

Provides information on CPU Usage, Memory Usage, I/O statistics. When each process terminates the acct program writes to a log file describing all the resources that process used. This can be used to see if a particular program or job is causing or putting undue stress on the system.

TOP

Provides real time information on what processes are using the most CPU. It can also be used to monitor memory usage, CPU states, and the load average. It is like a continually updating ps command. All system vendors do not provide top, but, the source can usually be found on the Internet and compiled.

OTHER VENDORS TOOLS

glance by HP, proctool by SUN.

NOW YOU KNOW THE PROBLEM, WHAT NOW?

There are four basic things that you can do:
get more of the lacking resource, optimize programs, and change the way the programs get run.

BUY MORE RESOURCES

You have to decide if the cost benefit ratio is great enough to invest money in new hardware. Memory and Disk are (relatively) cheap. CPU upgrades tend to be more costly.

OPTIMIZE PROGRAMS AND DATA

Maybe your applications have not been written in the most efficient manner. Change the algorithm for solving the problem at hand. With SAS it maybe a matter of moving code segments to take advantage of the fact that you already sorted the data in a particular way. Or better yet removing sorts and taking advantage of indexes.

There are several sessions at this conference that provide tips to

improve the efficiency of your programs.

You may also want to consider using more than one cpu at a time. This topic is discussed in the paper Multiprocessing with Version 8 of The SAS System.

CHANGE WHEN PROGRAMS GET RUN

Programs can be scheduled to run at times when the server may not be as busy, either overnight or on weekends. Take advantage of ***cron***, ***at***, and ***batch***. If the jobs must run during the day select times like lunch hours to have large programs execute.

MEMSIZE

The MEMSIZE option controls the maximum amount of memory that will by allocated by a single invocation of the SAS system. If it is not properly set, it can cause performance problems. Paradoxically, performance can sometimes be improved by increasing MEMSIZE, and in other situations, decreasing MEMSIZE.

If MEMSIZE is set higher than the amount of physical memory available, page faults may increase. On the other hand, several procedures (SORT, MEANS, and SUMMARY) will run faster if they are given enough physical memory. If the amount of memory available (as determined by MEMSIZE) is not sufficient, alternate algorithms, which will increase disk IO, will be used. Many statistical procedures require a specific amount of memory that is determined by the specific problem being solved. In these cases, MEMSIZE must be increased in order to solve the problem, even if it will increase the system paging.

WORK AND OTHER LIBRARIES

The WORK option determines where the SASWORK directory will be placed. The location of the work directory is very important to performance. If work directories are placed on a slow filesystem, the overall performance of jobs will suffer. There are many performance options for UNIX filesystems that can be used to boost overall IO performance. Many operating systems allow the creation of filesystems that reside, either partially or fully, in memory.

In general, file IO can be improved by doing the following:

- Spread the load over as many spindles, controllers and paths back to the host as possible.
- Separate your data and work areas.
- Avoid using NFS for frequently accessed data.

CONCLUSION

Performance problems can only be solved if they are properly identified. We have presented many tools that are available in the UNIX environment that can pinpoint performance problems, and we have offered a few guidelines for addressing some of these performance problems.

REFERENCES

There are several excellent papers on performance written by our technology partners on the SAS Institute web. Most of these papers include information that will be relevant to all UNIX systems, so I encourage everyone to start browsing at:
http://www.sas.com/partners/techpartners.html

HP-UX - SAS Software Configuration Guide or Now that I have an HP-UX server and SAS Software, how do I configure it?
http://sasprod.unx.sas.com/partners/hp/hpuxwp.pdf

Disk I/O Considerations for SAS Processing
http://www.sas.com/partners/hp/diskio.pdf

Optimizing Systems Performance
http://www.sas.com/partners/ibm/optimize.html

Peace between SAS Users & Solaris/Unix System Administrators: Finding a Middle Ground
http://www.sas.com/partners/sun/performance/index.html

Techniques for Optimizing CPU Performance
http://www.sas.com/partners/sun/wp/optimize.html

CONTACT INFORMATION

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

Clarke Thacher
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
Work Phone: (919) 677-8000
Fax: (919) 677-4444
Email: Clarke.Thacher@sas.com