Paper 387-2008

# SAS® Performance Monitoring – A Deeper Discussion®

Tony Brown, SAS Institute Inc.

## ABSTRACT

SAS® includes a large family of products and solutions, each with varying performance paradigms and machine-resource utilization patterns. Large data scales, intensive analytic procedures, and server consolidations that place mixed workloads on shared server and storage can result in occasional performance issues.

In my previous SAS® white paper, “Solving SAS® Performance Problems: Employing Host Based Tools,” I detailed the role of basic host-provided performance monitors, used in conjunction with SAS logs, to diagnose and solve SAS performance problems. In this paper, I continue with that approach by describing the use of easy-to-use tools such as *nmon* and *perfmon*®, investigate the most common causes of performance problems, and suggest what to look for in monitors first. This paper presents two case-studies that used actual monitors and interprets their findings. The goal here is to illustrate how you can use graphical monitoring tools such as *nmon* and *perfmon*® in conjunction with SAS® FULLSTIMER log output to determine problem causes.

## INTRODUCTION

Performance logs can be lengthy and difficult to read, and it can be a challenge to remember outstanding metrics you must cross-reference from one large output set to another. In addition, it can be difficult to line up an overview of the major monitors for a brief health check when they are all on disparate pages of long text files. There are some great solutions to this issue, which involve using the GUI presentation interfaces of some excellent third-party monitoring tools. A comprehensive view is quickly gained from their graphical interfaces. They are very easy to deploy and interpret. We will look at two of them, *perfmon*® and *nmon*, and list some others.

The discussion will begin with the most frequently occurring causes of performance problems and what to initially look for, using the monitors featured in this paper. I will then discuss subsequent levels of monitor investigation, provide examples to further illustrate potential issues that you might discover, and suggest how to proceed when you encounter an issue for which there is no obvious cause. I will conclude with suggestions for extensive monitoring that could be needed to find well-hidden problems, such as file system issues, improper storage layout, lock contentions, and so on.

## TOP SAS PERFORMANCE PROBLEM CAUSES

There is no substitute for efficiently written code and good data model and process construction. These foundations have a profound impact on resource utilization and performance, but when it comes to the hardware and operating environment, there are several primary causes of performance problems when using SAS®. In the order of predominance of occurrence they typically are:

A. Insufficient Storage IO
B. Memory Exhaustion
C. CPU Binding
D. File Cache Exhaustion
E. Network IO

The most predominant environmental cause of performance issues with SAS® is insufficient IO feed from disk-storage to the CPU. This is because SAS® is used to handle very large amounts of data, and it predominantly performs large-block reads and writes to and from storage, a bandwidth-oriented operation. Most storage sub-systems are set up by default to favor IOs-per-second related processing (IOPs), which is more common in relational database management systems and application systems-of-record (for example, reading and writing small, but numerous, blocks at a time). The relevant question for small-block operations is how many IO operations can I do per second? This contrasts with the long reads and writes that SAS® applications, such as query and reporting, analysis, and BI, perform. IO performance issues account for over 90% of the performance issues that our large customer base reports to SAS Tech Support and the SAS Performance Lab.

The second most frequently encountered cause is running a server out of memory (for the same reasons as IO above); the third is completely flooding the CPUs with numeric intensive processes common to SAS®. A less common cause of performance issues is over-utilizing the system file cache, through which SAS® performs its reads and writes. SAS does not perform Direct-IO by default. Finally, the least encountered general issue on our list is poor throughput via a network segment.

We immediately look for the first three culprits, which are the most predominant and obvious occurrences. We then focus on the others as a follow-up, or if the first three possibilities are eliminated. We like to examine performance monitors in several passes. The first pass focuses on the CPUs to see if they are bound, or not getting fed enough, then memory to see if it is exhausted, and then storage IO to see if it is keeping up. CPU monitors can be good indicators of memory or IO problems. If jobs take too long, and the CPU utilization is very low, an IO or memory issue probably exists.

In the subsequent passes of monitor examination, we look more deeply at IO issues, focusing on read/write rates to physical and logical devices, IO blocking performance, IO adaptors, kernel settings, system calls, and so on. If something is found, but does not have obvious causes, we examine the problem more deeply with additional tests and monitors, which will be discussed later. If nothing is found, but a problem clearly exists, a different monitoring approach at a more process/system-invasive level is conducted. This will be discussed later as well.

## NMON – A POPULAR AIX/LINUX GRAPHICAL MONITOR

*nmon* is a popular graphical display performance monitor that was created by Nigel Griffiths, of the IBM eServer® pSeries Technical Support Advanced Technology Group. It is an unsupported monitor and is available for general use without a fee. The *nmon* monitor command is run to collect information from the host kernel monitors. Then the monitor collection can be generated into a graphical output in *Microsoft Excel* spreadsheet format using the *nmon_ analyzer.*

*nmon* is easy to use. *nmon* has a plethora of information from the normal system kernel monitors. n*mon* supports screen output, GIF output, and a .csv file that can be converted to a graphical *MICROSOFT EXCEL* spreadsheet by the *nmon_analyzer* tool (see http://www.ibm.com/developerworks/aix/library/au-analyze_aix/index.html#resources for more information about the *nmon_analyzer*.) For additional *nmon* information use the *nmon* link at the end of the paper. When you open an *nmon Microsoft Excel* spreadsheet, it has a large number of TABS. Please see the *nmon* link at the end of the paper for information about these tabs, what they contain, and how to interpret them. Always use your Systems Administrator for help with *nmon* when available.

### WHAT TO LOOK AT IN NMON

In the first pass of an *nmon* monitor output, look at the following areas and their associated *nmon* tabs:

- Server Specs (AAA)
- CPU (SYS_SUMM, CPU_ALL, CPU01- CPU0n)
- Memory (MEM, MEMNEW, MEMUSE)
- IO (SYS_ SUMM, DISK_SUMM, DISKBUSY, IOADAPT, DISKBUSY1-n)
- Processes(PROC, TOP)
- Pages(PAGE)
- Network(NET)
- System Calls (FILE)

Often, the problem set will be found in one of these areas, or at least indications of a problem that requires further investigating. In addition, the following *nmon* reports need to be examined, especially if you have indications of IO performance issues:

- ESS Reports
- JFSFILE
- JFSINODE
- DISKBSIZE1-n
- DISKREAD1-n
- DISKWRITE1-n
- DISKXFER1-n

The remaining lettered tabs (for example, BBBB, and so on) contain information about actual kernel and system settings, such as disk and volume arrangements. These tabs offer a high level of information on how the physical system is arranged and they can be examined to determine those settings and storage arrangements that *nmon* provides in its reporting.

### NMON CASE STUDY – POOR JOB PERFORMANCE FOR SAS® DAILY JOB PROCESSING

A SAS Customer had a large set of concurrent jobs that were taking longer than they should have. The jobs were characterized by reads to an MS SQL/Server database on a remote server. Data was processed and reports were generated on the local IBM Server. To address the problem, our first activity was a thorough review of the SAS logs with the FULLSTIMER option enabled. The SAS job logs all showed a long Real Time in the SAS® FULLSTIMER log output for job steps, which represented how long in wall-clock-time the jobs were taking. The jobs were using less than 1/10 of that time in total System and User CPU time. This meant that the jobs were taking a long time to do a small amount of CPU processing. This generally indicates a potential IO or Memory issue. A review of the *nmon* monitors run during the jobs execution follows.

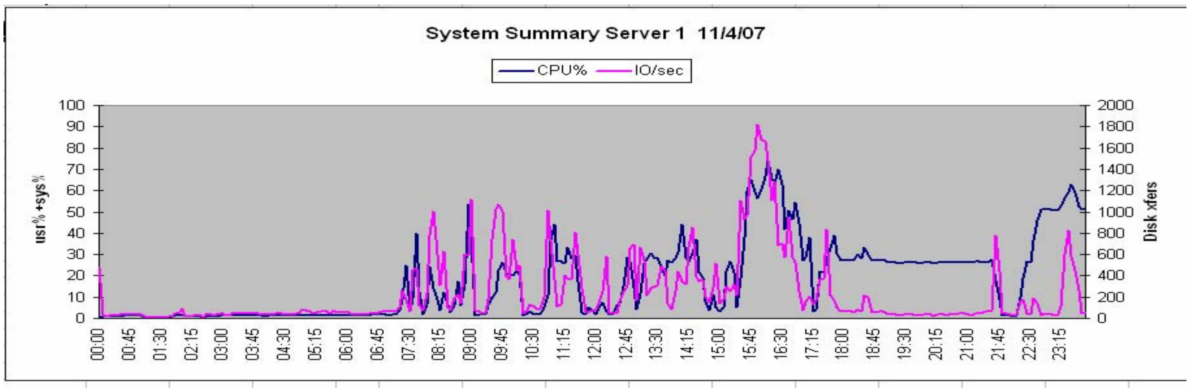
Below each *nmon* chart is an interpretation of what is seen, and what needs to be examined next. The series of charts is reviewed, the findings are explained, and recommendations are made to alleviate the IO problems experience.

| | A | B | C |
|---|---|---|---|
| 1 | **AIX** | 5.3.0.44 | |
| 2 | **build** | AIX53 | |
| 3 | **command** | /opt/CWmon/bin/nmon_aix53 -f -t -s300 -c287 | |
| 4 | **cpus** | 8 | 8 |
| 5 | **date** | 3-Jan-08 | |
| 6 | **disks_per_line** | 150 | |
| 7 | **hardware** | Architecture PowerPC Implementation POWER5 64 bit | |
| 8 | **host** | Server1 | |
| 9 | **interval** | 300 | |
| 10 | **kernel** | HW-type=CHRP=Common H/W Reference Platform Bus=PCI LPAR=Dynamic Multi-Processor 64 bit | |
| 11 | **ML** | 4 | |
| 12 | **progname** | nmon_aix53 | |
| 13 | **runname** | Server1appl | |
| 14 | **snapshots** | 287 | |
| 15 | **subversion** | @(#)nmon v11e - AIX53($Id: nmon11.c,v 1.10 2006/04/05 15:18:14 nag Exp $) | |
| 16 | **time** | 00:00.0 | |
| 17 | **user** | root | |
| 18 | **version** | v11e | |
| 19 | **analyser** | V3.2.6 | |
| 20 | **environment** | Excel 10.0 on Windows (32-bit) NT 5.01 | |
| 21 | **parms** | BATCH=0,FIRST=1,LAST=287,GRAPHS=ALL,OUTPUT=CHARTS,REORDER=True,SORTDISK=True,SVCTIMES=False,SVCXLIM=4,TOPDISKS=50 | |
| 22 | **settings** | DELIM=',',GWIDTH=683,GHEIGHT=280,LSCAPE=False,REPROC=True | |
| 23 | **elapsed** | 3.641313 seconds | |

SYS_SUMM | **AAA** | BBBB | BBBC | BBBD | BBBN | BBBP | CPU_ALL | CPU_SUMM | DISK_SUMM | DI

**Figure 1** *nmon* Case Study – *nmon* Environment and Parameter Information

The AAA Tab on the *nmon Microsoft Excel*® spreadsheet output gives you some environment information, and the version of *nmon* that is being used. We are also interested in the version of the operating system (AIX 5.3) and the number and type of CPUs (8-Power5). With this information about the host and *nmon* run interval, we move on to examine the System Summary Page, which begins our monitor review.




| Total System I/O Statistics | | CPU: | User% | Sys% | Wait% | Idle% | CPU% |
|---|---|---|---|---|---|---|---|
| Avg tps during an interval: | 241 | Avg | 14.7 | 3.1 | 4.5 | 77.8 | 17.7 |
| Max tps during an interval: | 1,824 | Max | 63.7 | 12.3 | 30.1 | 100.0 | 74.1 |
| Max tps interval time: | 15:55:54 | Max:Avg | 4.3 | 4.0 | 6.7 | 1.3 | 4.2 |
| Total number of Mbytes read: | 1,439,034 | | | | | | |
| Total number of Mbytes written: | 590,834 | | | | | | |
| Read/Write Ratio: | 2.4 | | | | | | |

**Figure 2** *nmon* Case Study – *nmon* System Summary

The first *nmon* chart we review is the System Summary log at the start of the nmon. It is very general in nature but divulges some high-level CPU and IO utilization. However, it is plain to see that the maximum CPU utilization here is only about 2/3 of what it could be. A closer look at individual CPUs is necessary to ensure that this average doesn't hide individual bound CPUs, or high wait states on CPUs. It does show a nominal amount of IO that tracks the CPU utilization closely for most of the daytime processing period. Now we look at the average CPU utilization chart.

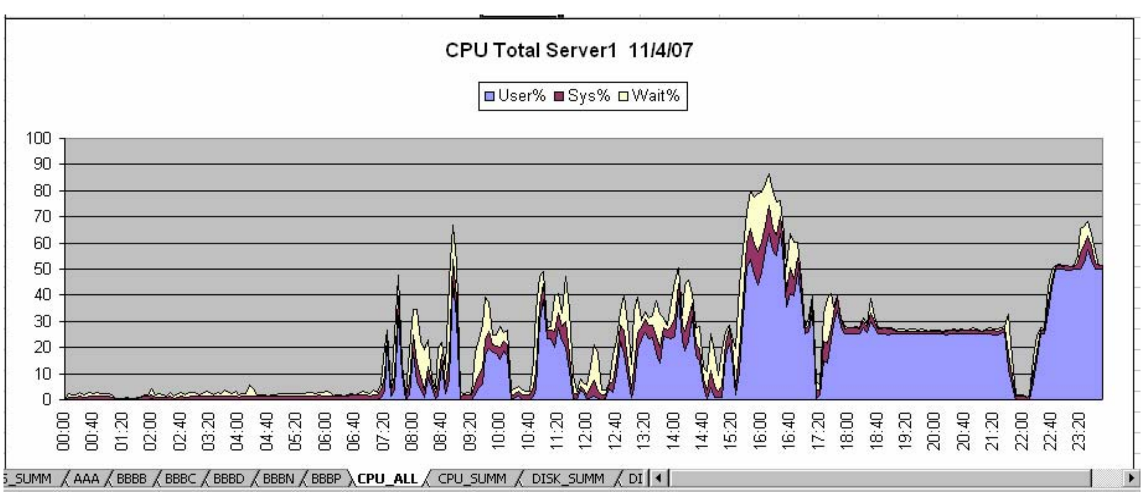



**Figure 3** – *nmon* Case Study – *nmon* All CPUs Utilization Report

This report corroborates the System Summary. It does show that the highest portion of CPU is generally spent in User time, which is good. However, it also shows a fairly significant WAIT time in daytime peak processing. It will be necessary to examine the individual CPU reports to see if they are being utilized well.

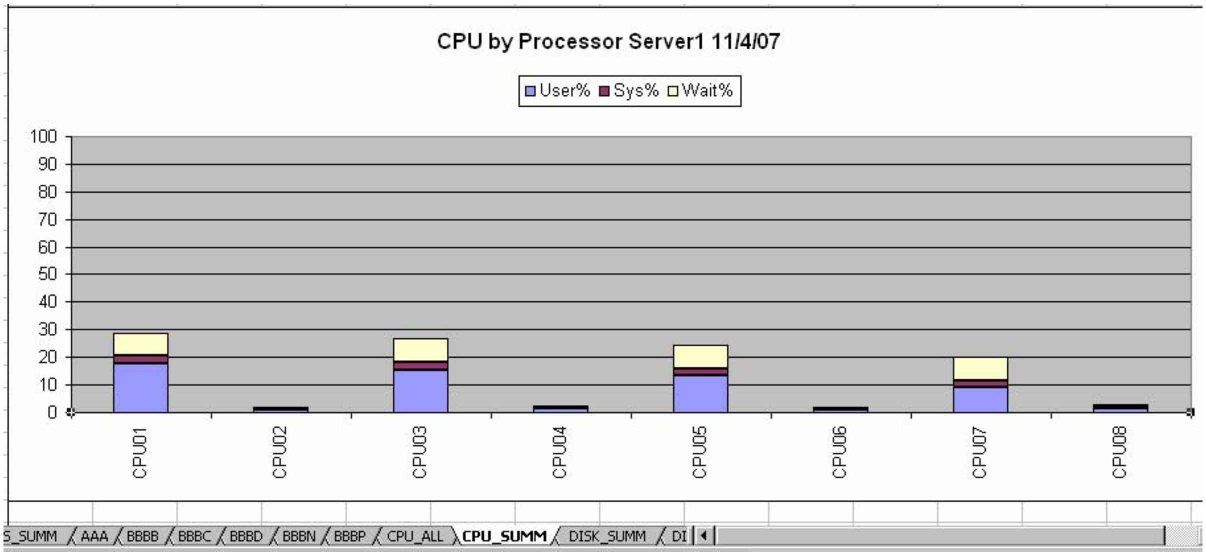



**Figure 4** *nmon* Case Study – *nmon* Individual CPU Average Utilization Report

When we turn to the individual CPU Utilization Report, it becomes apparent that half of the CPUs are readily being used, and none of them is at a high utilization rate on average. They do spend a significant amount of time waiting for IO. It's time to look at the CPUs individually to see the peaks and valleys, as the averages don't show the picture of what happens with extremes at any point in time.

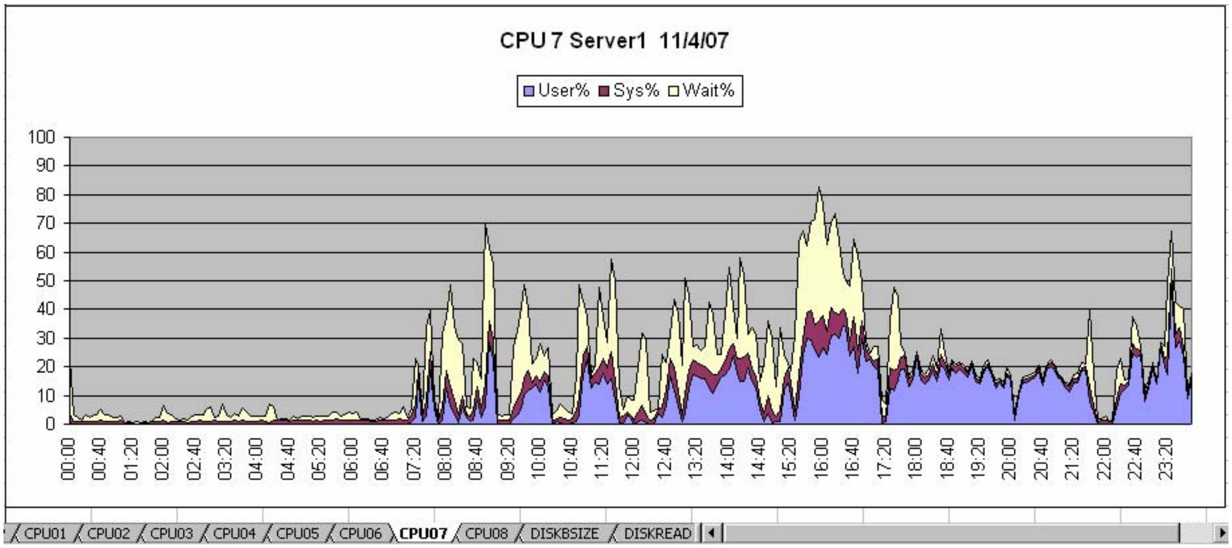



**Figure 5** *nmon* Case Study – *nmon* Individual CPU Utilization Report

Looking at an individual CPU, it becomes apparent that the CPU Wait percentages averages from the previous graph are much understated for the daytime peak processing periods. During the day, the percentage of time that the CPU spends waiting on IO or other system resources is often larger than the percentage of time that is spent in User and System processing. This indicates a serious holdup getting data to the CPU. This CPU graph was chosen as it was fairly representative of the experience of the four out of the 8 CPUs that are being heavily used by this system. It is time to begin examining IO and Memory to see what can be found.

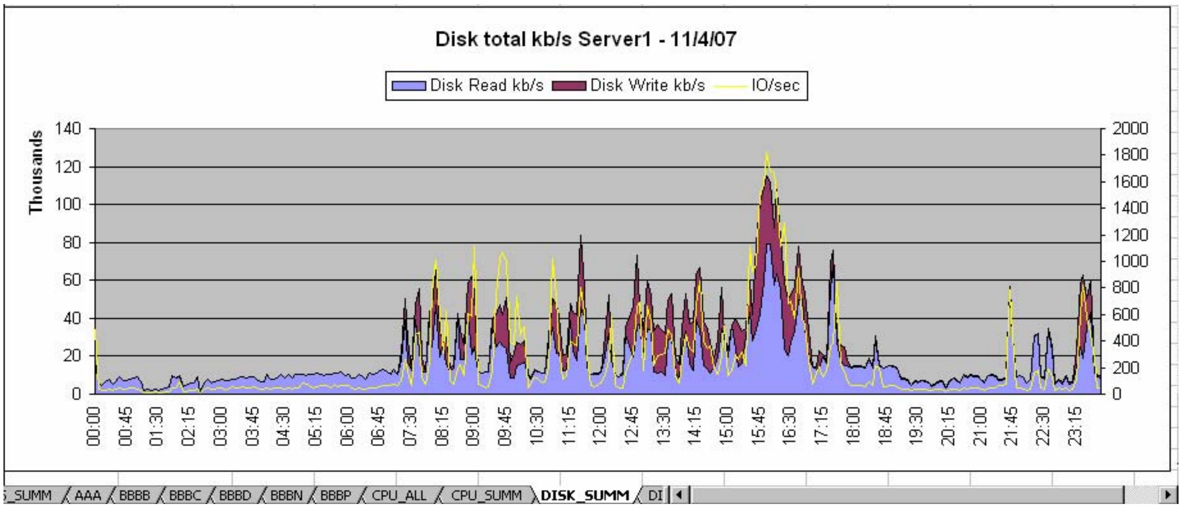



**Figure 6** *nmon* Case Study – *nmon* Disk Total Read and Write Rates Report

The read rates show an average around 20 Mb/sec during the daytime processing. Write rates are close to the same. At the highest peak rate, reads are less than 80 Mb/sec. This looks low for a system that has slow jobs and low CPU utilization, which indicates a problem somewhere. We look now at individual device reports, where an immediate problem becomes apparent.

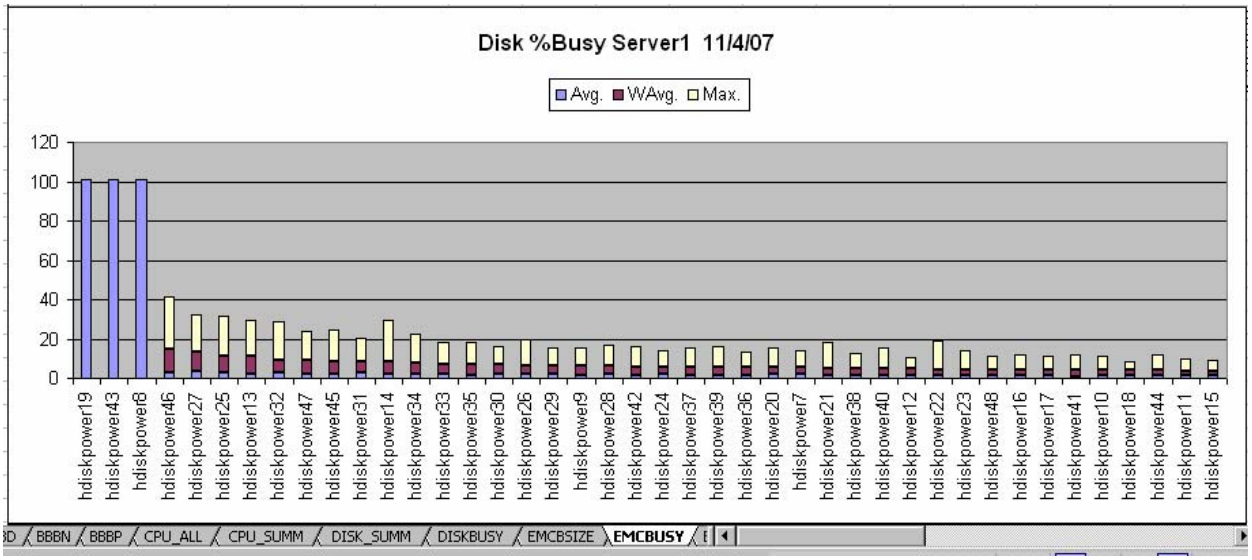



**Figure 7** *nmon* Case Study – *nmon* Device Busy Report

Hdiskpower19, 43, and 8 are all maxed out at 100% busy on average. Because all of the file systems in this system are overlaid across all the hdisks (we got this from the BBBC Report), it means that these three disks represent a serious IO bottleneck to the file systems used by SAS®. These disks are overlaid with any permanent read/write file systems, as well as SAS® WORK file systems. The question has to be raised to the Storage Administrator to determine how this SAN is configured. It appears the disks in question above could be concatenated in the stripe paradigm. This commonly happens and can represent a serious IO bottleneck for the concatenated disks if they are limiting throughput from any file system that touches that disk. Basically disk operations become serial during concatenation and stop performance for that portion of the file system overlay to one disk at a time. We examine memory next to look for any other bottleneck indications.

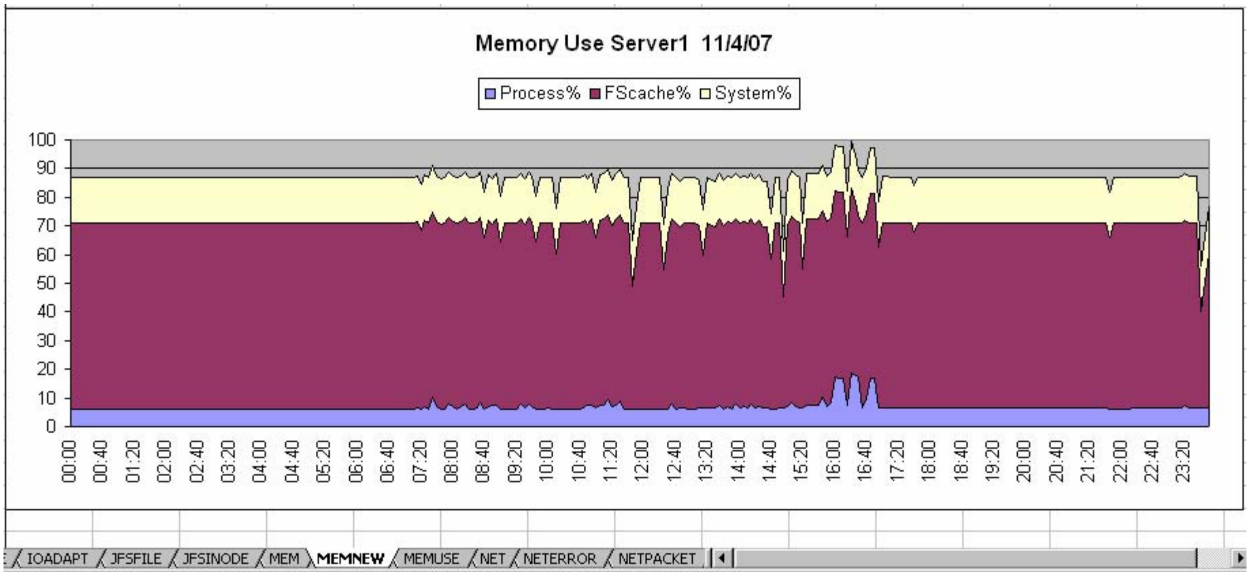



**Figure 8** *nmon* Case Study – *nmon* Memory Use Report

The memory use indicates that the System memory is much higher than the Process Memory. It is normal to see the file cache (FScache%) utilize much of the memory because SAS does all of its reads and writes via file cache. There are no indications of memory issues in this report. Virtual memory tuning on AIX is done via several *VMTune*® parameters, and a full discussion of this is outside the space and time frame allotted for this paper. See the References section below for a link to another SAS Global Forum 2008 Paper from IBM that discusses these parameters, and how to use them.

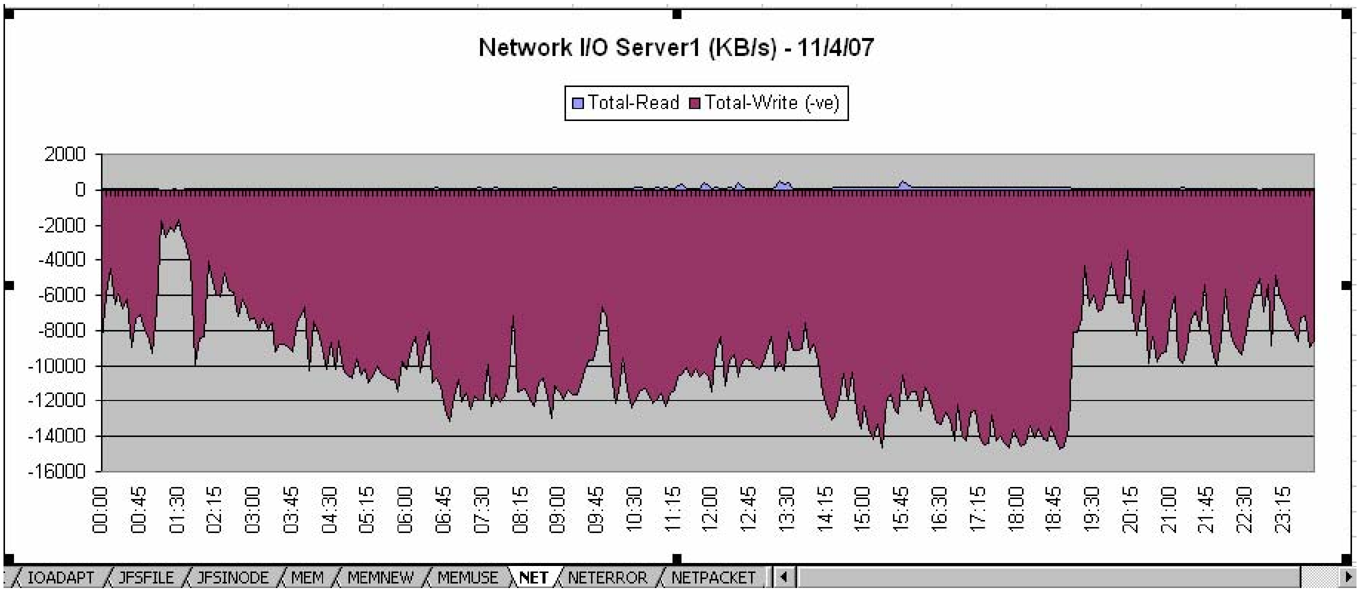


**Figure 9** *nmon* Case Study – *nmon* Network Utilization Report

The network utilization report above and the other reports on that TAB (not shown), show continuous network write activity at a low rate. Some remote reading was done to an RDMBS in some of the jobs in this mix, but the other read/write processes were all done locally. We need to see what is doing continuous network writes on this system, so we go to the CPU% By Command to see what is running on this system. We need to find out what is writing to a network interface.

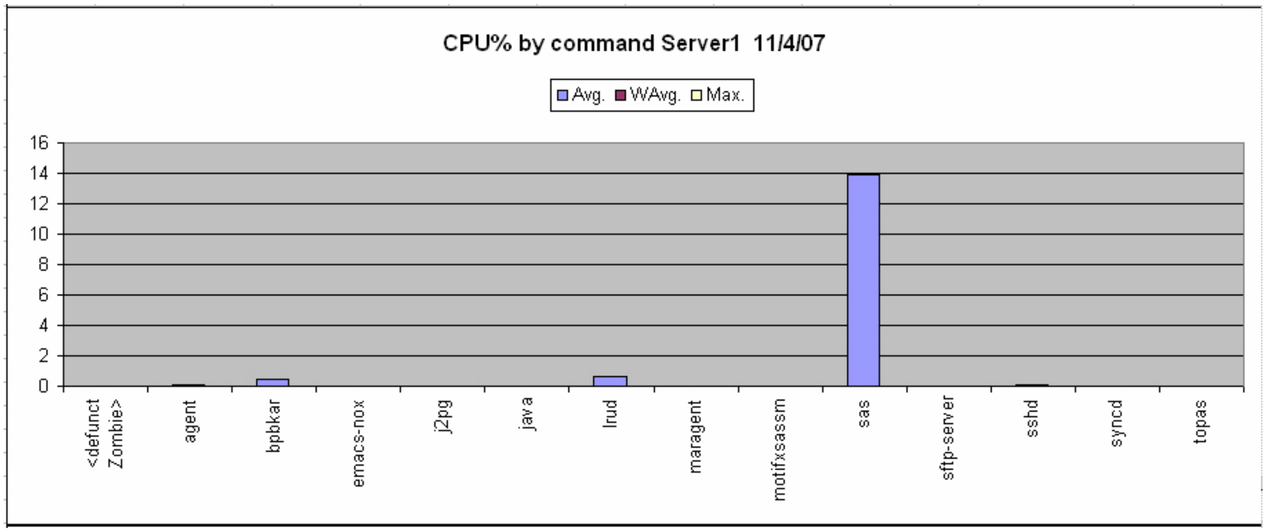



**Figure 10** *nmon* Case Study – *nmon* Top Processes by CPU Report

Looking at the top processes by CPU shows SAS® is being heavily used on this system. It also indicates the lru paging deamon (lrud) is not being over-worked, which corroborates our view that memory is not having any problems. Another process, bpbkar (a Veritas® NetBackup process), that is using some CPU is interesting. A network backup is running on this machine during the day. We go to the IO report to see if it could be contributing to the IO load of the machine. This type of process, coupled with the suspected concatenated disks above, could put a damper on the bandwidth associated with SAS IO.

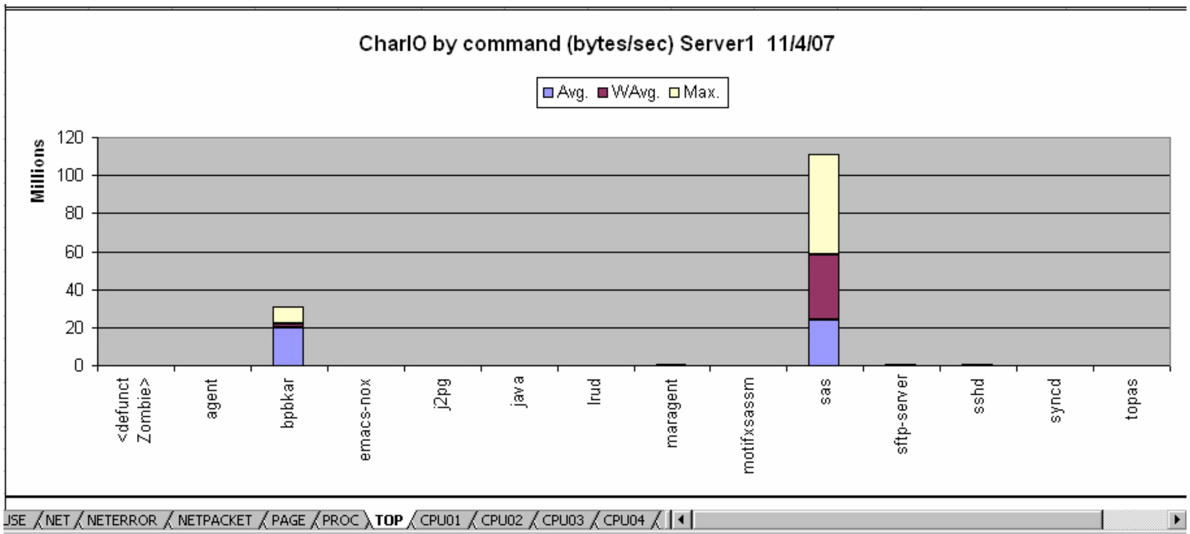



**Figure 11** *nmon* Case Study – *nmon* Top Processes by CPU Report

The Character IO by Command report indicates the bpbkar backup process is using between 13 and 30 MB/sec of IO continually. This is probably affecting the performance of the SQL/Server Reads over the network performed by the SAS jobs. It is also an interference with the local file systems during their daytime processing. Along with the concatenated disks, these are the two culprits that were causing the IO holdup for the SAS® jobs on this system.

*nmon* monitor pages that were reviewed but not shown in this case study (for example, system calls) had no effect on the case study's problem and so space was saved for the next case study.

## NMON CASE STUDY - FINDINGS AND RECOMMENDATIONS

The two obvious findings contributing to IO holdup on this system were:

- The concatenated disks, Hdiskpower19, 43, and 8
- The Veritas® Netbackup Process bpbkar running during daytime production processing

Our recommendations were the following:

- Redesign the network backup of the file systems to be more efficient (backup rates show a very low write rate over the network segment).
- Run the backup outside daily processing hours.
- Remove the concatenation between hdisk43, 19 and 8 and put them into a normal physical RAID5 stripe set so concurrent access to the disks via stripe can eliminate the single, serial disk performance exhibited on these three disks.

When these recommendations were put in place, performance improved to the customer's satisfaction.

Perfmon – a popular Windows monitor
*perfmon* is the performance monitor most widely available for Windows systems. A tutorial on its use can be found at:
[http://www.microsoft.com/technet/archive/mcis/*perfmon*.mspx](http://www.microsoft.com/technet/archive/mcis/perfmon.mspx) .

Again, we look for predominant causes first, such as IO, Memory, and CPU, followed by file cache and network performance. Primary *perfmon* statistics of interest include, but are not limited to:

| **Performance Area** | **Performance Object** | Counter |
|---|---|---|
| IO | Physical Disk and Logical Disk | **Average Disk Queue Length** |
| | | **Current Disk Queue Length** |
| | | **Disk Read Bytes/sec** |
| | | **Disk Write Bytes/sec** |
| | | **Disk Transfers/sec** |
| | | **Avg. Disk seconds/Transfer** |
| | Process | **IO Read Bytes/sec** |
| | | **IO Write Bytes/sec** |
| Memory | Memory | **Available Megabytes** |
| | | **Committed Bytes** |
| | | **Pages/Sec** |
| | | **Working Set** |
| | Paging File | **%Usage** |
| CPU | Processor | **%Processor Time** |
| | | **%Priviliged Time** |
| | Process | **%User Time** |

The definitions of these metrics, and guidelines to interpret them are included in the paper link for "Solving SAS Performance Problems: Employing Host Based Tools" referenced at the end of this paper.

The following case study illustrates the ease of using *perfmon*, and how you can use the different counters from the performance objects together to arrive at and corroborate a diagnosis.

## WHAT TO LOOK AT IN PERFMON

Beginning with IO, look at the Processor Report for %Processor Time to determine if it is underutilized. Then move on to IO counters in the Physical Disk Report, and follow up with memory. The following *perfmon* case study, which contains a primary and potential secondary performance issues for correction, illustrates this approach.

Perfmon Case Study – Poor Mixed job performance in a windows environment
While running a small set of daily SAS® jobs a SAS Customer was experiencing poor performance with the PROC IML and other large DATA STEP oriented jobs. The jobs were reading data from the local SAN. The server was a two-CPU Windows NT Server with older processors. The IML jobs were taking longer than the customer was expecting. The SAS logs showed the IML jobs had a combined CPU time (user + system) in the SAS® FULLSTIMER job statistics that was almost equal (within 99 % of the REAL TIME). This indicated the IML jobs were CPU bound. Other non-IML SAS jobs were running, too, they were not requiring as much CPU, but they were performing a large amount of IO, and showing a high Real Time versus the combined-CPU Time. The charts below describe the findings of the *perfmon* monitors run during the jobs execution.

As with our previous Case Study, we will follow our paradigm of examining CPU first to look for indications of bottlenecks, followed by memory, then IO. The highlighted blue line in the legend of each of the graphs below corresponds to the graph line emblazoned in white in the graphs below.

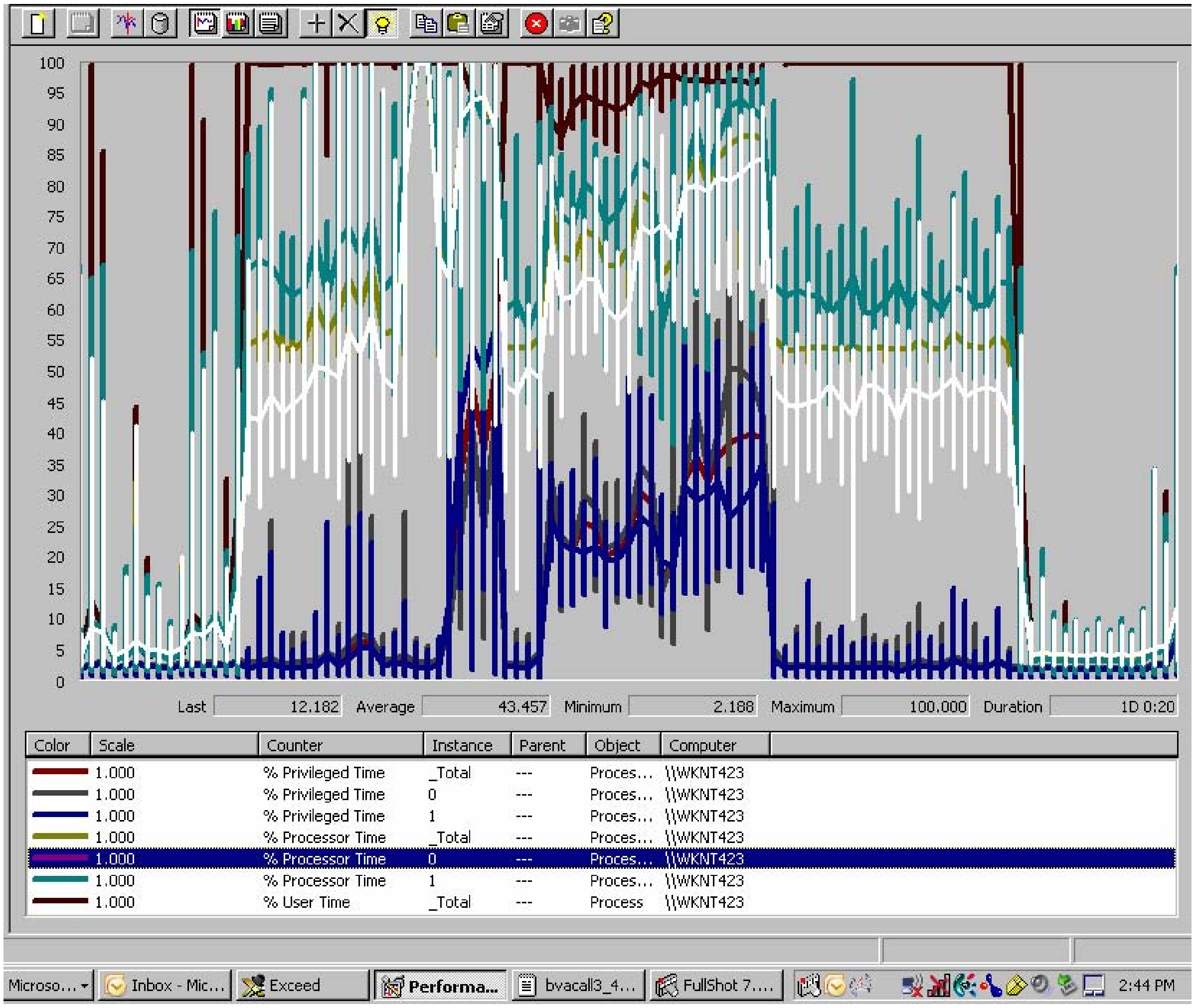



**Figure 12** *perfmon* Case Study – *perfmon* Processor Report - % Processor Time

The first *perfmon* graph is a 24-hour run that covers a typical business day, and corresponds with the SAS® logs that the customer sent to us. It started at 8:00 in the morning and ran until 8:20 the next morning. The Processor statistics illustrate how busy the CPU is. The processors all ranged from an average of 43% to a maximum of 100% busy for the two-CPU machine. The CPU spent a significant amount of time at or near 100% during the IML job run. It was substantially busy for most of the day. It is easy to see why the IML jobs showed up CPU bound during the many-job run, with all of them competing for CPU time on a two-CPU box.

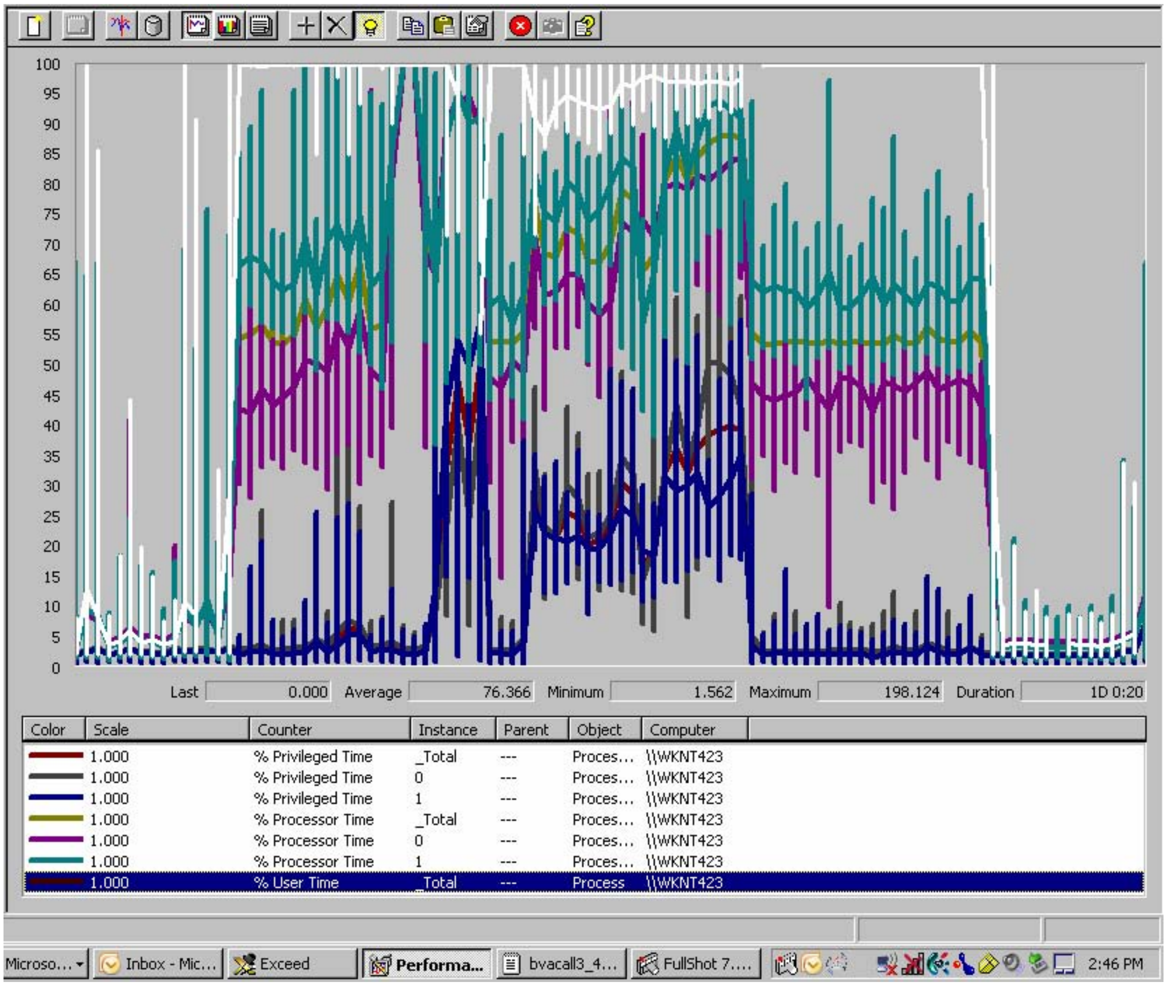



**Figure 13** *perfmon* Case Study – *perfmon* Process Report - % User Time

%User Time, a statistic from the Process set, corroborates how much time the Total Processes are using CPU. During the heart of the customer's job run times, the machine is completely CPU bound, and it stays completely or mostly bound for a significant portion of the day.

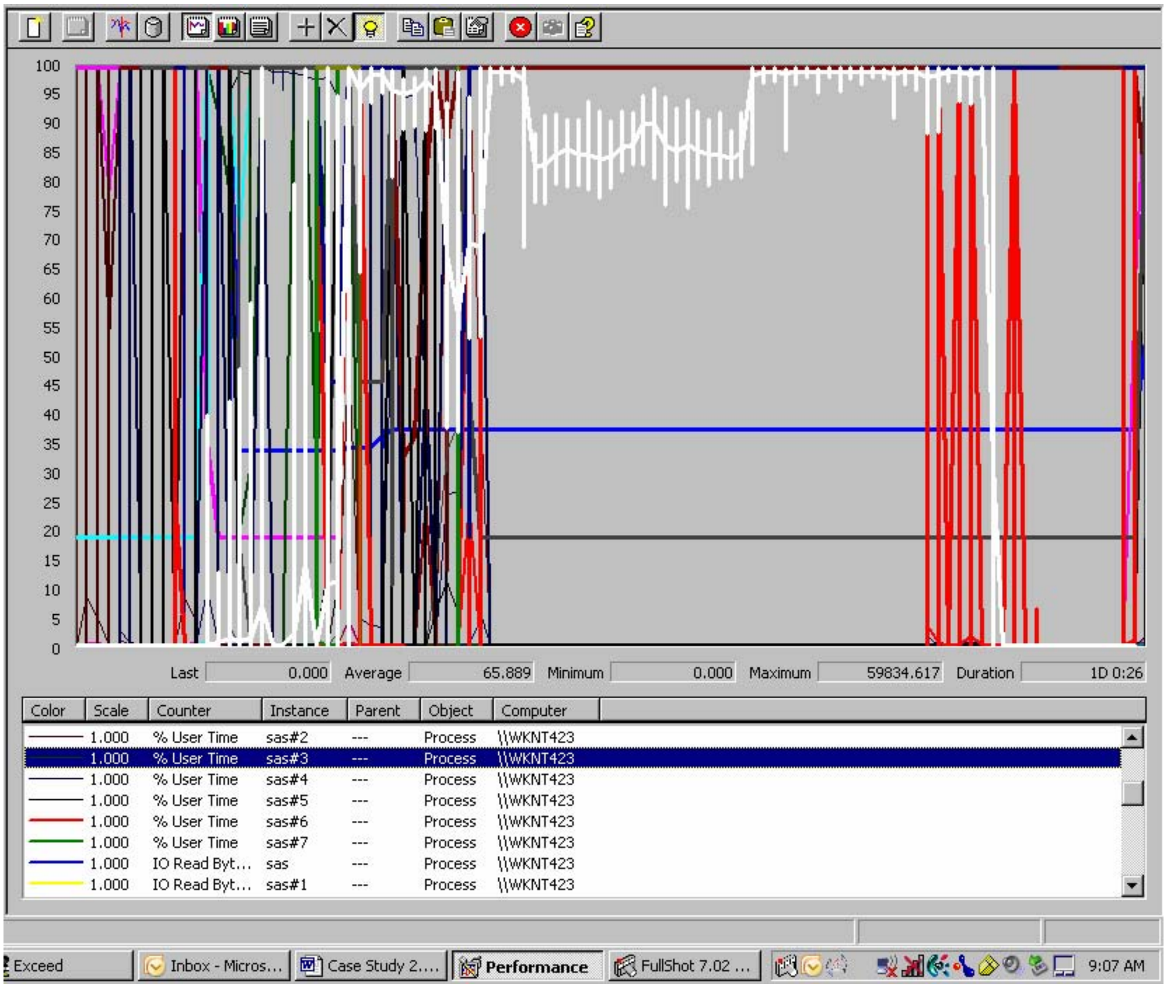



**Figure 14** *perfmon* Case Study – *perfmon* Process Report - % User Time – PROC IML

The individual SAS® logs show that one of the SAS® Processes, a PROC IML job was taking all of the CPU that was available. This job alone was consuming all of the CPU available to SAS® that was left from other server and application processes. We see that we have serious CPU resource issues with the job mix and turn our focus to IO.

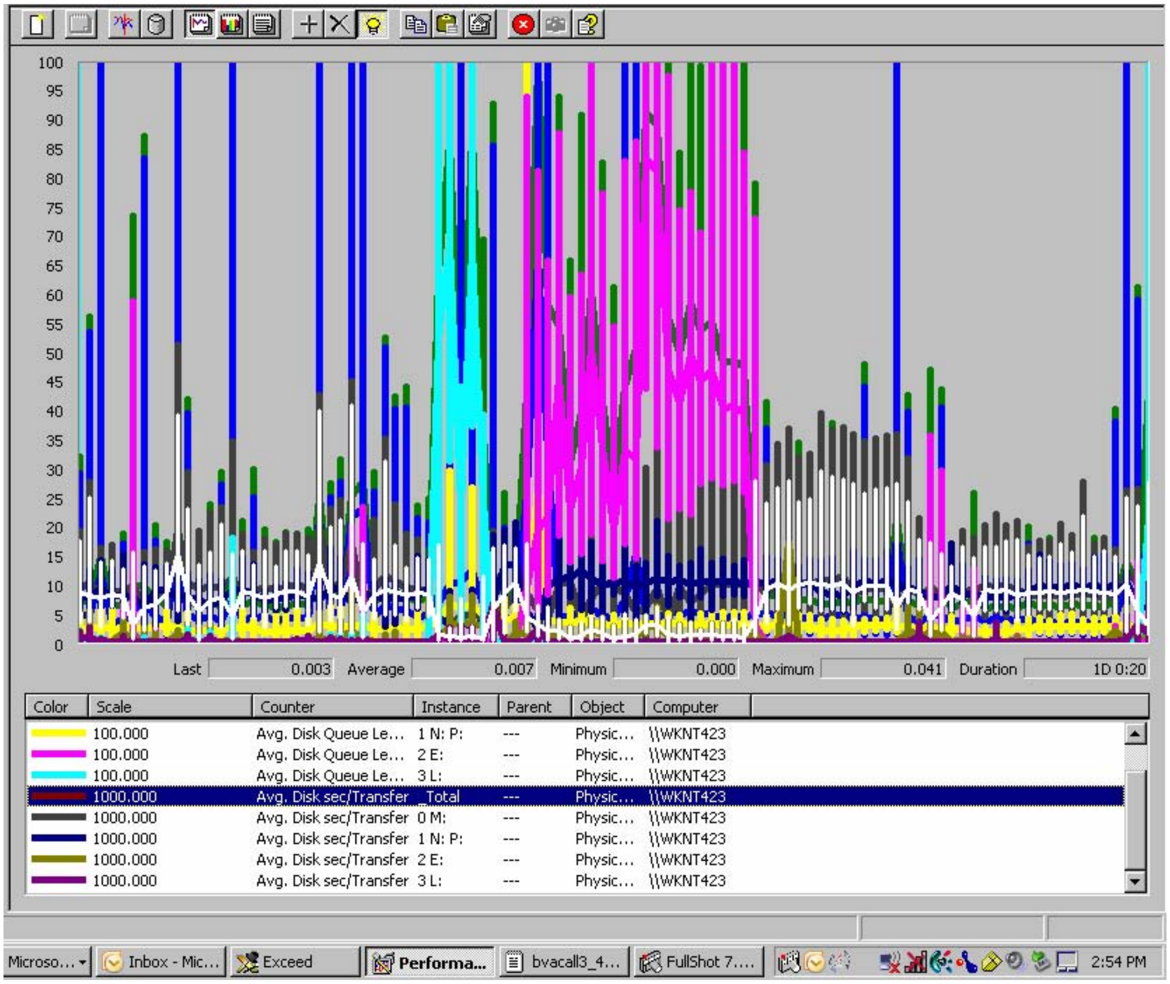



**Figure 15** *perfmon* Case Study – *perfmon* Disk Queue and Disk Seconds/Transfer Report

The first thing to look at in the *perfmon* IO report is disk queuing. The Average Disk Queue on this report was well within a reasonable range. Disk queues shouldn't go much above two for a long period of time, and they don't. The Average Disk Queue lines aren't highlighted here, because they were unremarkable and I wanted to use the paper space to show the Average Disk Seconds per Transfer, which were high. As you can see by the numbers in the maximum column above, they get to .041 seconds per transfer. Values chronically above .009 can indicate slow disk performance. The values above show that the disk array is having a hard time keeping up. While the disk queues aren't radically high, they do approach three and four for some periods of time, which is at the edge. This combined with the transfer times indicates that if more than a few jobs were running with these characteristics, the IO subsystem could be a serious bottleneck instead of a minor one for this job mix. Although the PROC IML job is CPU bound, the other jobs are exhibiting an IO holdup. A close examination of memory and file cache is warranted.

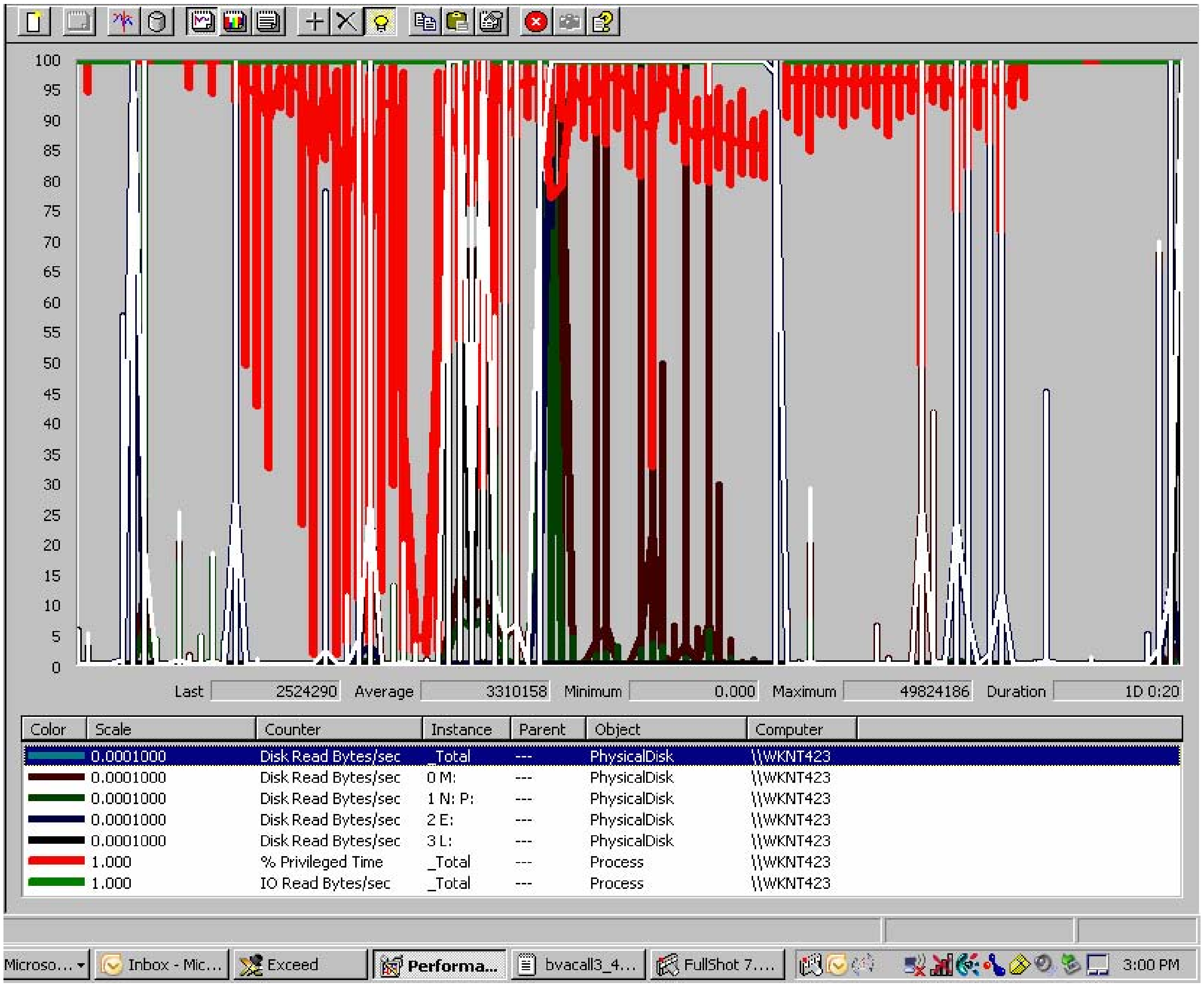



**Figure 16** *perfmon* Case Study – *perfmon* Read Performance

The Disk Read Bytes per Second (at an aggregate of 48 MB/sec here) are fairly equal to the IO Read bytes per Second, which can indicate a stressed file cache. Even though the server is not in a memory-swap state, the cache page replacement algorithm is working overtime. This file cache appears on the verge of issues with the current job load.

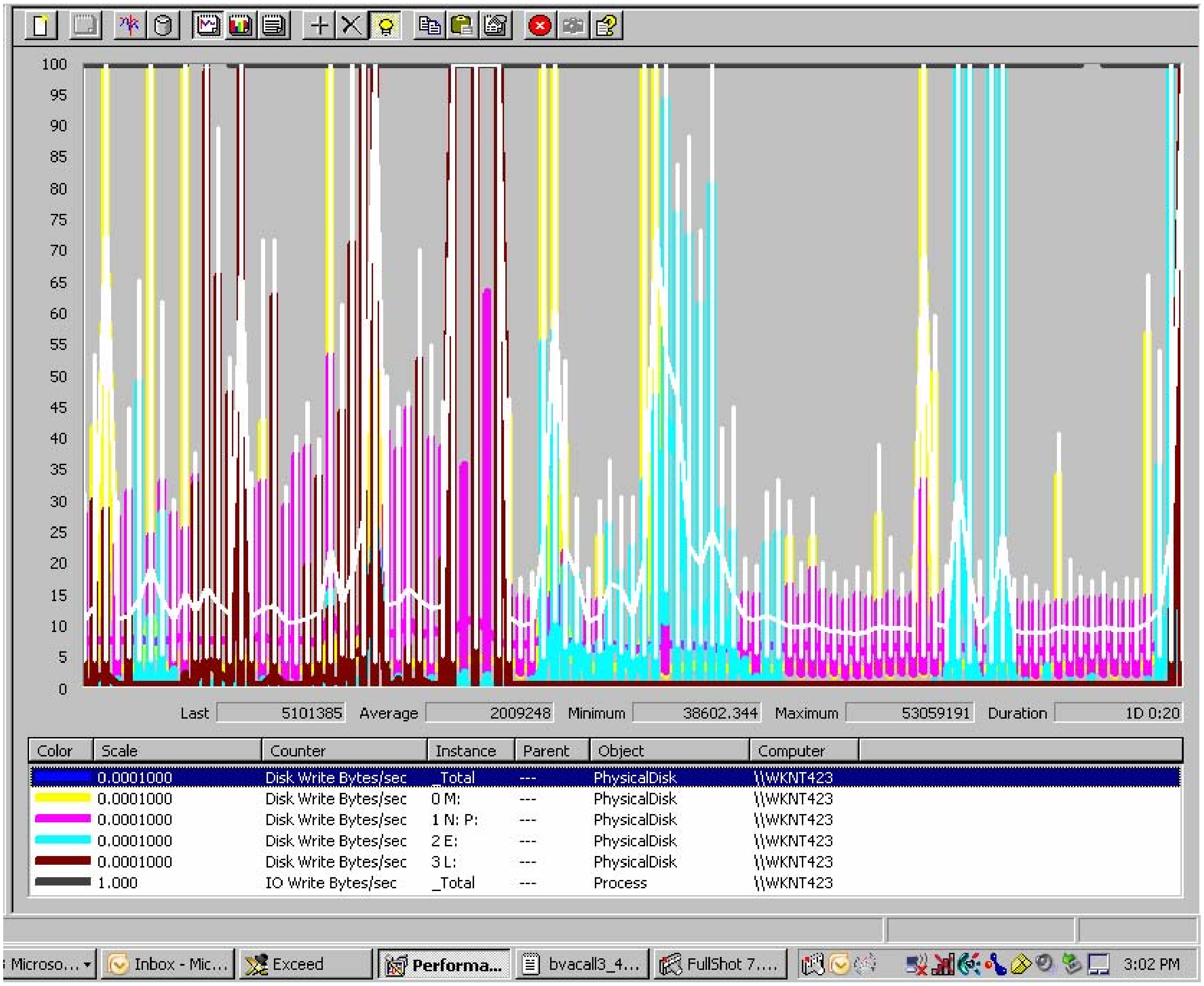



**Figure 17** *perfmon* Case Study – *perfmon* Write Performance

The same thing is true for the write performance. We can see that the maximum write output during the 24-hour period occurs during our peak processing when the SAS jobs are running, and the server is committing writes to its four file systems at an aggregate of 51 megabytes per second, most of which occurs on the L: Drive.

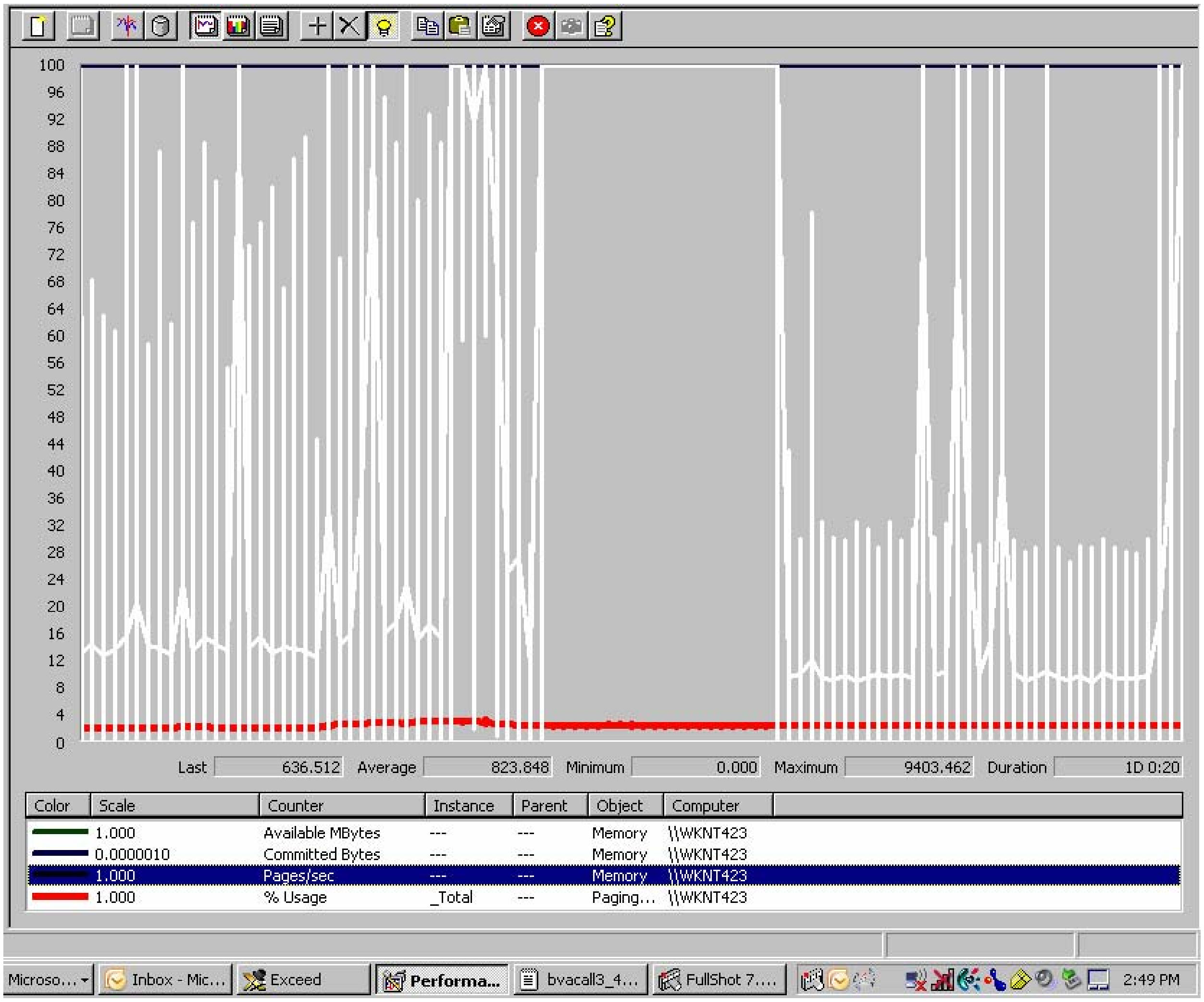



**Figure 18** *perfmon* Case Study – Memory Performance

When looking at Memory performance in the *perfmon* graphs, at first glance it appears memory is healthy, the available megabytes never go below 903, and the Committed Bytes never come too close to or exceed RAM, indicating memory is available.  However when the size of the file cache and the Committed Bytes come close to or touch the RAM size, performance with the system file cache slows down.  This machine is close to that condition.  The high Pages-per-Second figure is a corollary factor to look at here. It is very high (chronically over 100 pages per second is high) and indicates cache stress and page faulting.

## FINDINGS AND RECOMMENDATIONS

This system was running a combination of SAS® and non-SAS® jobs. While there weren't many SAS® jobs running, one was a PROC IML, which was soaking the CPU pretty well, and the others were doing a lot of IO, from a fairly slow set of file systems, and stressing the system file cache with the high throughput demand of the individual tasks.

There were only two CPUs on this machine. Given the use of PROC IML, and any other CPU intensive operations, this machine definitely needed more and faster CPUs. The CPU utilization was high over a 24-hour period, which indicates that job scheduling or working off-hours won't significantly increase the run time of these jobs.

The disk array needs to be examined to see if it can be set up to deliver better than 50 megabytes per second per file system. More memory could be allocated to the system file cache, and potentially another operating system or additional server should be considered for deployment, especially if it continues to be stressed on a continual basis.

When the Diagnosis isn't easy
The examples given in the case studies of this paper were very straightforward. Subsequent discussions with the Systems and Storage Administrators of both case studies confirmed server and storage architectural and tuning conditions. The issues were resolved with recommended changes.

Now take the following example of a performance issue that is more difficult to diagnose. For example, you have low CPU utilization, but are experiencing an obvious wait (High Real Time in the SAS Log coupled with very low CPU time). None of the disks has a wait queue, the discs aren't busy, the file cache isn't stressed, and memory is abundant. This type of scenario can be perplexing. Monitoring in such cases must go down to the kernel and process level because symptoms like these can be caused by a plethora of things as intricate as high lock wait times on file system inode traversals or by inter-socket communications blocking being off by some margin.

Monitors such as truss*, tusc, lockstat, Intel® Vtune, Sun® collect, Sun® dtrace*, and so on, can be used to see inside the process and interaction between the process and hardware and operating system resources.

When the issue cannot be attributed to easy causes, it typically becomes much more difficult to diagnose. Diagnostic efforts here can involve SAN and storage monitors at the device, RAID, file system, volume, and channel levels. In addition Server-side monitoring can include deeper kernel level monitors, process traces, lock traces, and so on. The problem often occurs in an interface between one of the entities – application software, host kernel algorithms, host memory or file system cache, host IO adaptors, volume manager, file system, or raw storage. At this point, the issue is most likely outside your expertise and possibly beyond the ability of your systems administrator to find independently. If so, you need to assemble a team of administrators, software experts, and hardware experts. This team is typically composed of your Storage and Systems Administrator, the SAS® Administrator at your site, Technical Support Engineers from your host server vendor, Technical Support Engineers from your storage vendor, Technical Support Engineers from your File Systems and Volume Management Vendors, and SAS® Support. It requires a coordinated team effort involving all of these personnel to more quickly and effectively diagnose the problem.

## Other Graphical Monitors

There are several other commonly used performance monitors that have graphical output components associated with them. They are usually easy to read and interpret. A few of the monitors are:

- *BMC Patrol*
- *HP Glance+*
- *SUN Resource Monitor*

## CONCLUSION

Graphical performance monitors such as *nmon* and *perfmon* can make short work of identifying the most common environmental performance issues you could encounter using SAS®. The monitors are easy to read, and you can quickly cross-check and corroborate information from monitor to monitor. For those issues with your SAS® applications that you cannot diagnose, work with your SAS Account Manager. Contact SAS Technical Support and involve personnel from the SAS R&D Performance Lab and the SAS Enterprise Excellence Center.

SAS Technical Support is a ready first line of defense to help review the monitors and diagnose the problem. Other SAS® Support personnel work through the Technical Support Owner to interface with the appropriate personnel identified above to engender a disciplined and scientific monitoring process to identify the problem. The SAS Performance Lab in Host Systems R&D can help with performance problems that are difficult to diagnose, and require deeper level monitoring and technical trouble-shooting interaction with your Systems, Storage, and Network administrators.

Please review the links below and discover how you can use *nmon* and *perfmon* to quickly diagnose and help solve your environmental performance issues affecting SAS.

Recommended reading

Crevar, M., Brown, T., Ihnen, L., Best Practices for Configuring your IO Subsystem for SAS®9 Applications, Cary, NC: SAS Institute Inc.
http://support.sas.com/rnd/papers/sgf07/sgf2007-iosubsystem.pdf

Tsao, H., et al. Tuning Guide for SAS®9 on AIX 5L. IBM Whitepaper. IBM Corporation.
http://www03.ibm.com/support/techdocs/atsmastr.nst/WebIndex/PRS1882

Crevar, M., How to Maintain Happy SAS Users, Cary, NC: SAS Institute Inc.
http://www.nesug.org/proceedings/nesug07/as/as04.pdf

References

For general *nmon* information, including supported os platforms and *nmon* output information, please see the following link:
http://www.ibm.com/developerworks/aix/library/au-analyze_aix/index.html - resources
http://www.ibm.com/developerworks/aix/library/au-analyze_aix/index.html#resources

A tutorial on *perfmon* and its use can be found at:
http://www.microsoft.com/technet/archive/mcis/*perfmon*.mspx .

A previously referenced paper, "Solving SAS Performance Problems: Employing Host Based Tools" can be found at:
http://support.sas.com/rnd/scalability/papers/TonySUGI31_20060403.pdf

Another paper by the same author has been updated for using SAS FULLSTIMER statistics available in SAS®9, with updated advice for modern storage array characteristics. It can be found at:
http://support.sas.com/rnd/scalability/papers/solve_perf.pdf.

ACKNOWLEDGMENTS

The author would like to acknowledge the assistance of Margaret Crevar and Leigh Ihnen of SAS Institute for their collaboration, for sharing their knowledge in many performance diagnostics experiences, and for reviewing this paper.

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

Tony Brown
SAS Institute, Inc.
15455 N. Dallas Parkway, Suite 1300
Dallas, TX 75001
Work Phone: 214-977-3916
Fax: 214-977-3921
E-mail: tony.brown@sas.com