Paper 386-2008

Using nmon to Monitor SAS Applications on AIX Servers

Alfredo Mendoza, IBM Corporation, Austin, TX

ABSTRACT

SAS is strongly recommending you monitor your SAS environment to avoid performance bottlenecks. But what tools should you use? Should you use standard monitoring commands such as vmstat and iostat? These commands produce many lines of text and require manual eyeballing or specially written scripts to peruse the output and spot potential problem causes. If you are using AIX to run your SAS environment, there is a freely downloadable tool called *nmon* that may eliminate the need for learning different tools to look at I/O, memory and CPU statistics. This paper introduces nmon as a tool for monitoring and determining the cause of performance problems that may suddenly appear or slowly creep in as a result of dynamic changes within the datacenter and the company as a whole.

INTRODUCTION

When UNIX servers that will run SAS applications are initially setup, they are configured to handle specific workloads with the provision that the servers can be scaled up to handle additional workloads in the future. Overtime, the configuration of the servers may change as more users use the SAS applications, filesystems are expanded or added, and more data files are produced. It is important that system administrators remain vigilant in monitoring their servers to continue to provide high levels of service.

	CPU	Memory	I/O Subsystem
Status Commands	vmstat, topas, iostat, ps, mpstat, lparstat, sar, time/timex, emstat/alstat	vmstat, topas, ps, lsps, ipcs	vmstat, topas, iostat, lvmstat, lsps, lsattr/lsdev, lspv/lsvg/lslv
Monitor Commands	netpmon	svmon, netpmon, filemon	fileplace, filemon
Tuning Tools	schedo, fdpr, bindprocessor, bindintcpu, nice/renice, setpri	vmo, rmss,fdpr, chps/mkps	ioo, lvmo, chdev, migratepv,chlv, reorgvg

Table 1 AIX specific commands to get status, monitor and tune the CPU, memory and I/O subsystem.

UNIX administrators are always looking for simpler ways to monitor their servers efficiently. Their mission in life is to make certain critical applications are responding well to user requests. For applications to respond well, the servers they run on must be free of performance problems that can come in the form of I/O, memory or CPU constraints. System administrators commonly use tools such as sar, iostat, vmstat, df and ps to collect server statistics. These tools are used in scripts and usually ran at certain points of the day through cron jobs. The output of these scripts is simple text and often parsed by other scripts that look for anomalies and red flags alerting administrators of potential problems. Data collected provides system administrators with simple, portable and easy to use tools that they can use and re-use on company wide servers. While simplicity has its advantages it can be limiting in determining exact causes of problems before they become more apparent. In this case, the only recourse the administrator has is to use vendor specific operating system commands to determine the cause of the problem.

For any UNIX server, there can easily be more than half a dozen vendor specific tools that a system administrator has to be proficient in using. Table 1 shows a few AIX commands to monitor CPU, memory and I/O subsystem. For those administrators lucky enough to acquire commercial quality tools, they still need to learn how to use these tools; not to mention the additional license fee that comes with it. If you happen to be using AIX to run your SAS environment, there is a downloadable tool by the name of nmon. It is free and has an easy interface to help system administrators monitor server activities. The logs collected by nmon are formatted by another free tool called nmon analyzer. The analyzer converts the collected data in a spreadsheet format. It displays data in text as well as graphs that show several server resource usage. Nmon is one of the standard tools used by IBM technical consultants to help identify performance

bottlenecks. It is simple to use and minimizes the need to know several AIX specific commands used for monitoring server performance.

The graphs in the next few pages were from a real life customer situation where nmon was used to analyze server activities in determining how resources in the server were responding during a particular SAS ETL (Extract Transform and Load) job run. After nmon data was collected and analyzed a couple of configuration changes were made to the server cutting in half the time the job took to complete after the changes were made. Let me walk you through the analysis.

ANALYZING I/O DATA

As mentioned in the paper [1], it is common to have many instances of SAS applications running at any given instance of time. Mostly, these SAS applications create a large number of temporary files that are potentially manipulated frequently for long periods of time. The operating system's file cache is utilized by SAS applications to perform high volume of read and write operations dominated by sequential rather than random access operations.

The nmon analyzer output shows how much disk activity was occurring during this particular SAS ETL run. Figure 1 shows a snapshot of total disk activity while the SAS ETL job was running.

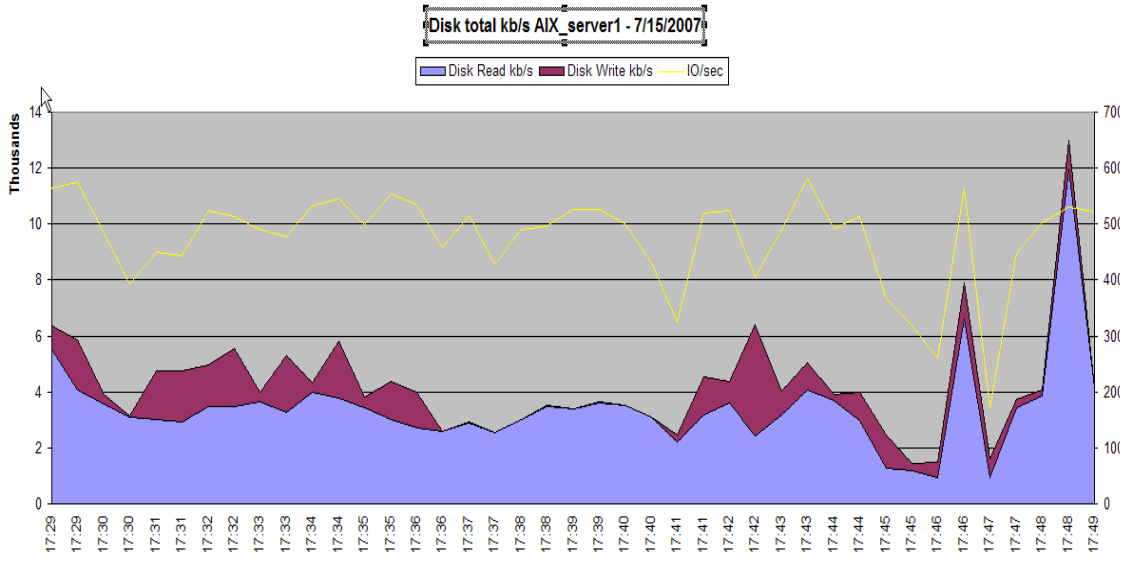

Figure 1 Total disk activity snapshot while running SAS ETL job.

As Figure 1 shows, the workload generated a fair amount of disk activity. There are almost the same amount of disk reads as there were disk writes – ranging from 1,000 kb/s to as high as 13,000 kb/s. Having confirmed that this workload is generating I/O activity, we then need to look at specific disk activities to tell us what filesystems these activities were taking place.

ANALYZING DISK ACTIVITY

Figure 2 shows a snapshot of individual disk activity for the SAS ETL workload. As Figure 2 shows, the activity was concentrated on hdisk9, hdisk13, hdisk1, hdisk8 and hdisk0. Nmon data showed that hdisk9 contains a filesystem used for temporary SAS files. It also shows that this filesystem spanned 2 disks[1] – hdisk9 and hdisk7[2]. The question then was why didn't hdisk7 have as much activity as hdisk9? In addition to analyzing disk activity, it is also worthwhile knowing what type of disks we were working with. Nmon provides data on the type of disks as well as the disk controllers that were used in this server configuration.

[1] Nmon also collects static data like filesystem layout.

[2] The disk names are logical names as seen by the AIX OS. In reality these disks were made up of a group of 8 disks in a RAID5 configuration.

It turns out that these disks were part of an IBM Enterprise Storage System (ESS). This storage unit (hdisk) can be configured in a RAID5 (7+1) configuration with stripping. So it did not make sense why nmon only showed activity on one hard disk - hdisk9.

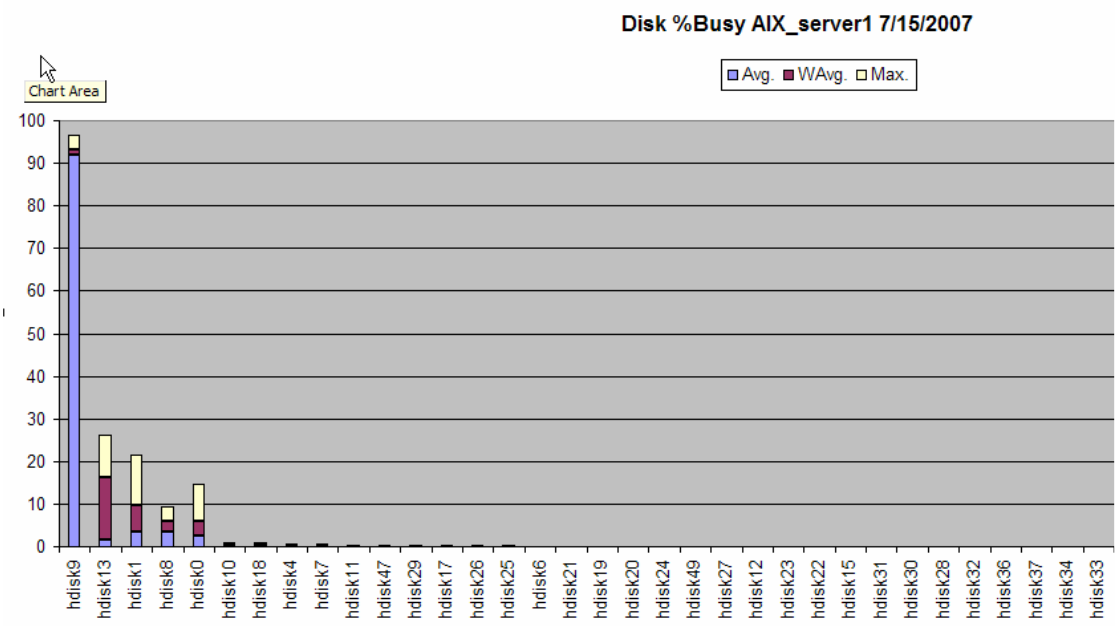

Figure 2 Individual disk activities for the SAS ETL workload.

In discussing this finding with the storage administrator, it was found that the disks were configured using concatenation instead of stripping. Several papers published on tuning SAS applications recommend striping to improve I/O bandwidth. For more information about best practices on configuring I/O subsystems for SAS applications refer to [1][2].

With a potential I/O bottleneck identified and a plan to re-configure the disks to use striping, we proceed to look at other potential areas for improvement.

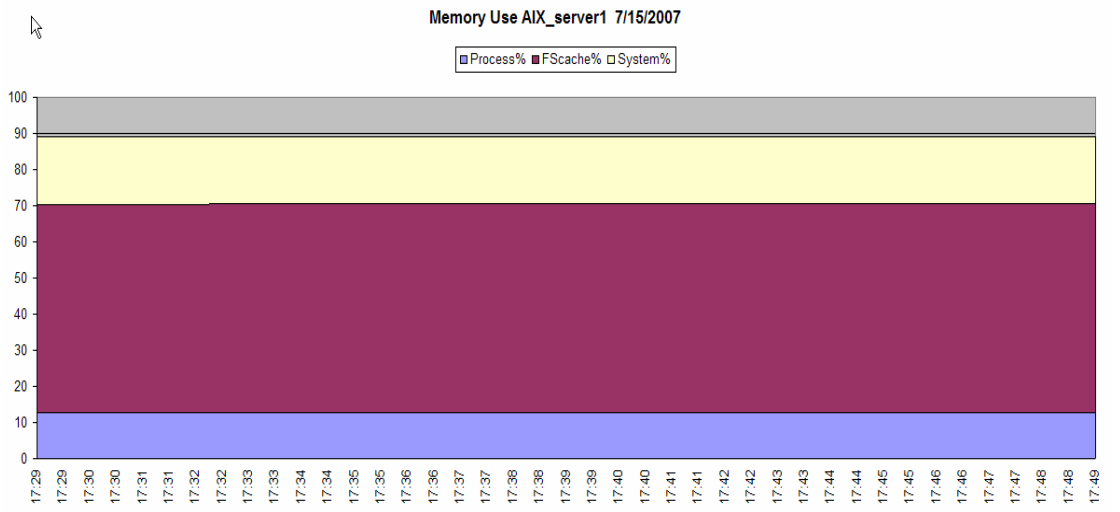

Figure 3 Visual representation of virtual memory consumed by the system, processes and file buffer cache.

MONITORING VIRTUAL MEMORY

After looking at the I/O activities, we turn our attention to virtual memory use. Nmon gathers virtual memory usage and presents it in a graphical format. Figure 3 is the visual representation of memory consumed by the system, processes and file buffer cache. In AIX, unused memory is used for buffer cache. Figure 3 shows that almost 60% (maroon color) of virtual memory is used for buffer cache, also known as *file memory*. These are usually pages from permanent data files in persistent storage. Figure 3 also shows about 20% of memory is used by processes which are known as *computational memory* or *computational pages.* These pages consist of the pages that belong to working-storage segments or program text (executable files) segments.

It is important to know that some workloads benefit by emphasizing the avoidance of file I/O while some workloads benefit more by keeping computational segment pages in memory. In our situation it is the former.

The ratio of memory used for files versus those used for computational segments is loosely controlled by the *minperm* and *maxperm* values for JFS type filesystems. The page replacement algorithm follows the following:

- If percentage of RAM occupied by the file memory rises above maxperm, page-replacement steals only file memory.
- If percentage of RAM occupied by file memory falls below minperm, page-replacement steals both file and computational memory.
- If percentage of RAM occupied by the file pages is between minperm and maxperm, page-replacement steals only file pages unless the number of file re-pages is higher than the number of computational re-pages.

Figure 4 shows a visual representation of minperm and maxperm settings as well as the percentage of file pages, represented by numperm value, and computational pages. In the figure we see that the maxperm setting, which is at 25%, is set at a lower value than the percentage of RAM currently occupied by file memory, numperm, which is at close 60%. This means that file pages have a higher possibility to be targets of page-replacement. To give the file pages a higher probability of avoiding being page-replaced, we can set the maxperm value to a higher value which is greater than numperm. In this situation, the maxperm was reconfigured to a value of 80%.

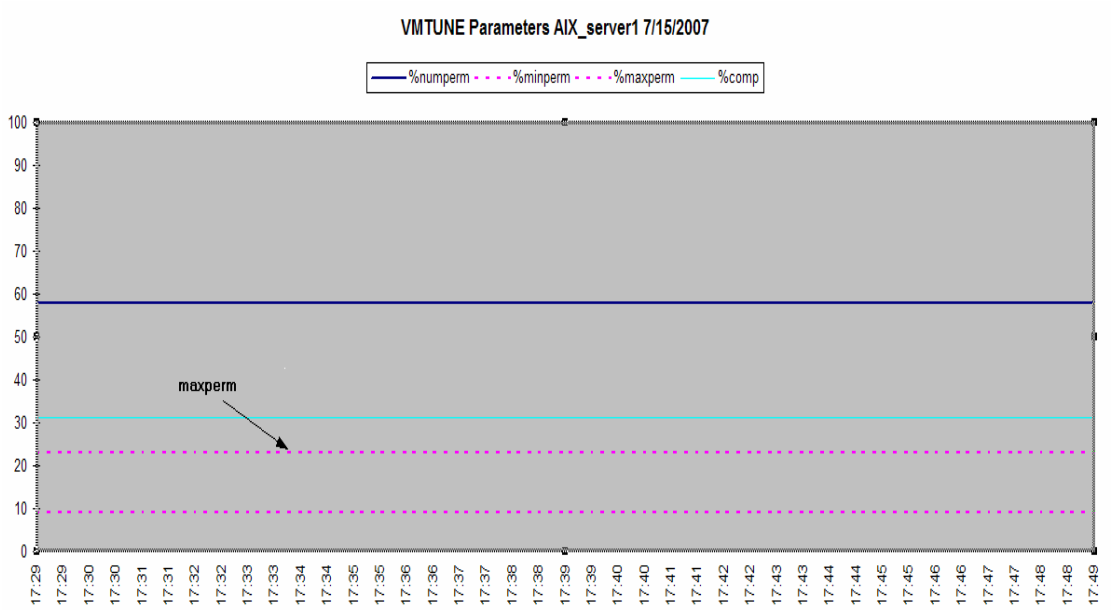

Figure 4 A graphical representation of VMM tuning parameters and computational and file memory.

For JFS2 filesystems, the tunable parameter is called *maxclient*. In our nmon report[3], we saw that maxclient is set at 25% while the numclient value, which represents the percentage of file memory currently in RAM for JFS2 files, is at 24.7%. Best practices dictate that the value of maxclient be configure to the same value as the maxperm to provide equal memory access for both JFS and JFS2 filesystems. But for situation, it was left untouched for now to minimize changes in the original configuration.

RESULTS OF CONFIGURATION CHANGES

In addition to striping hdisk7 and hdisk9, a decision was also made to move them to a faster DS8300 array[4]. After reconfiguring the disks array for striping and setting the maxperm to 80%, the workload was rerun. The end result showed that these changes improved the time to complete the workload in half (from ten hours to five hours) which made the customer extremely happy.

All of the data collected for this workload was done using one simple command – nmon. Nmon proved to be a simple but effective tool to gather data that can be displayed in both text and graphical form to help pinpoint bottlenecks caused by mis-configured disks arrays or un-tuned server settings.

MONITORING CPU UTILIZATION

While CPU utilization was not a factor in resolving the performance problem in this situation, it is still worthwhile looking at what nmon displays in terms of CPU utilization. Figure 4 shows an nmon CPU utilization graph.

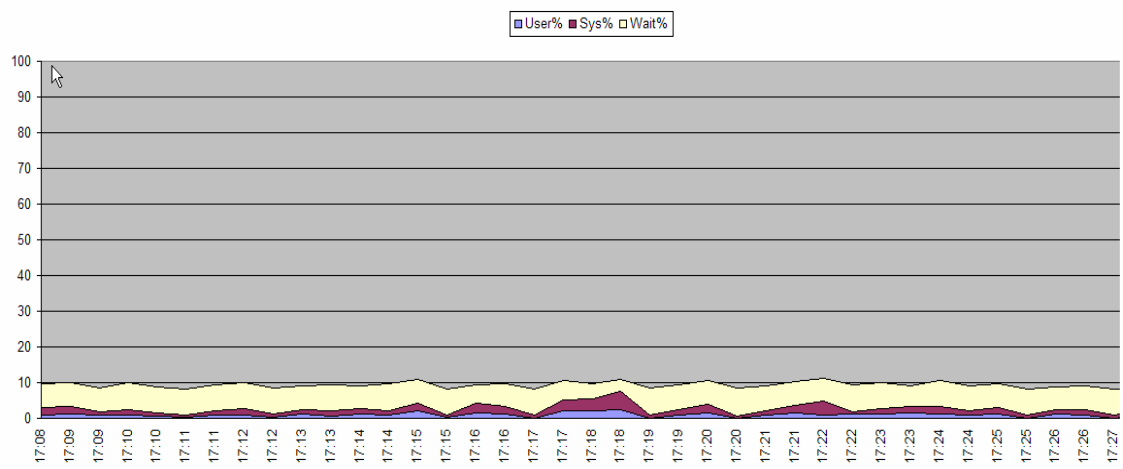

Figure 4 Nmon CPU utilization graph.

In the CPU utilization graph we see that the average CPU usage during the test run was about 10%. The utilization graphs also breaks down the usage into user, system and wait utilization. In this chart we can see that the wait time (beige color) occupy the majority of the utilization rate. Although wait times do not necessarily indicate an I/O problem in all situations, it tends to indicate some type of bottleneck specially when there is only one type of workload running on the machine.

CPU UTILIZATION ON VIRTUALIZED ENVIRONMENT

Figure 5 shows an nmon graph of the number of logical CPUs during the entire test run. The graph may not be of any significant interest because it only shows a static number of CPUs – in this case 12.

[3] The maxclient and numclient values are reported only as text.
[4] The array was already part of the original server configuration.

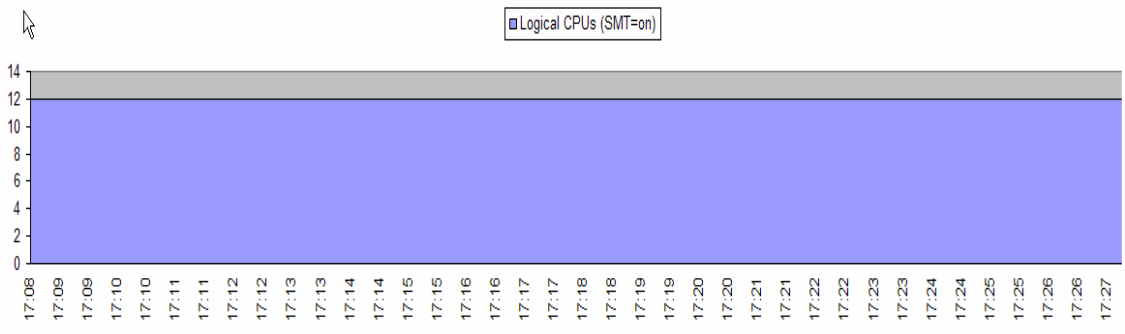

Figure 5 Number of logical CPUs

The graph becomes more relevant if the server is configured for Micro-Partitioning using minimum and maximum values for CPU usage. If that was so and CPU utilization warrants CPU resources to be allocated or de-allocated in different time frames, we would see a graph similar to that of Figure 6.

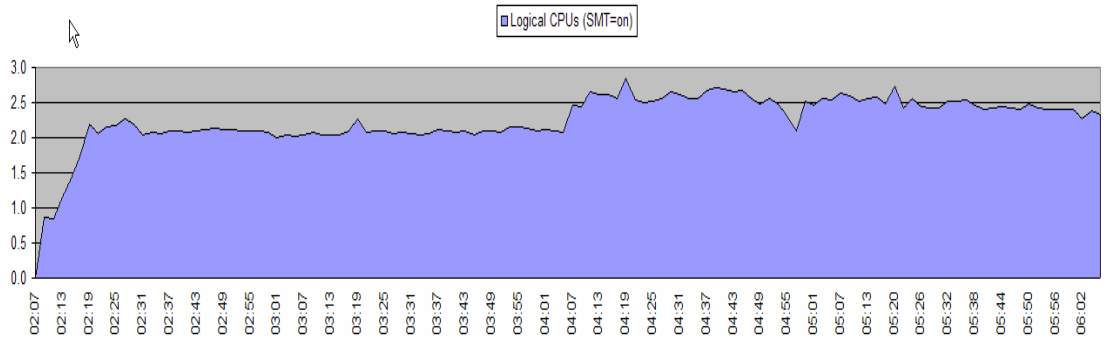

Figure 6 Nmon CPU count with virtualization turned on.

The graph on Figure 6 shows an example[5] of what nmon displays when the server is configured to use CPU virtualization using minimum (0.5 CPU unit) and maximum (3 CPU units) values. As time passes and the workload utilizes more CPU, the server allocates more CPU to satisfy the workload. In this case up to about 2.75 units of CPU.

OTHER NMON GRAPHS

Nmon also monitors paging activity and process resource utilization which includes CPU and memory utilization. Here are some examples of each.

PAGING SPACE

In AIX the virtual memory manager (VMM) is responsible for controlling memory usage and paging activity. From a performance point of view, the VMM can function to minimize disk bandwidth activity and response degradation by reducing paging activity. Application throughput can be greatly improved if paging activity can be reduced to a minimum or even totally avoided. Figure 7 shows a graph of a system where paging activity is going on.

[5] The data displayed is not part of the original customer scenario.

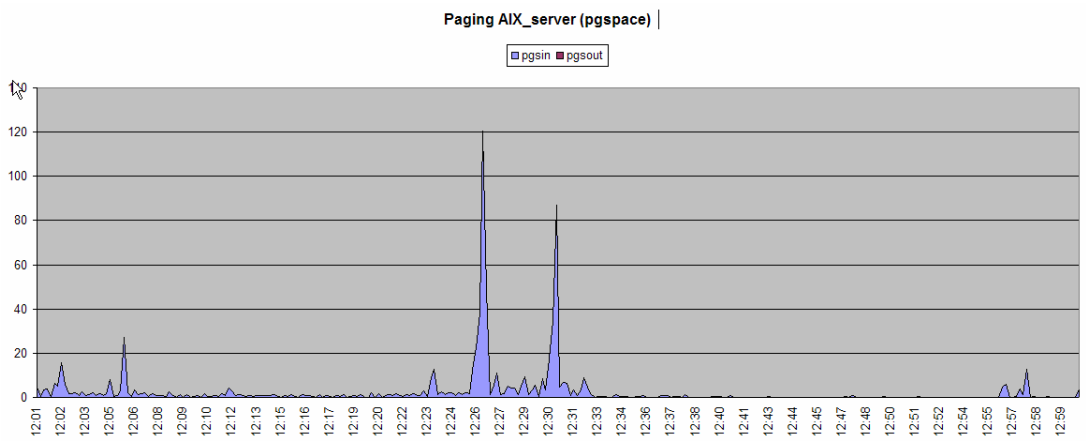

Figure 7 Paging space usage graph.

In some cases, if there is enough memory in the system and paging activity is still observed changing the VMM parameter, lru_file_replace, from 1 to 0 can eliminate the paging activity. For more information on this tuning parameter consult [4].

PROCESS RESOURCE UTILIZATION

By specifying the –T flag in the nmon arguments, nmon displays processes that uses the most CPU and memory.

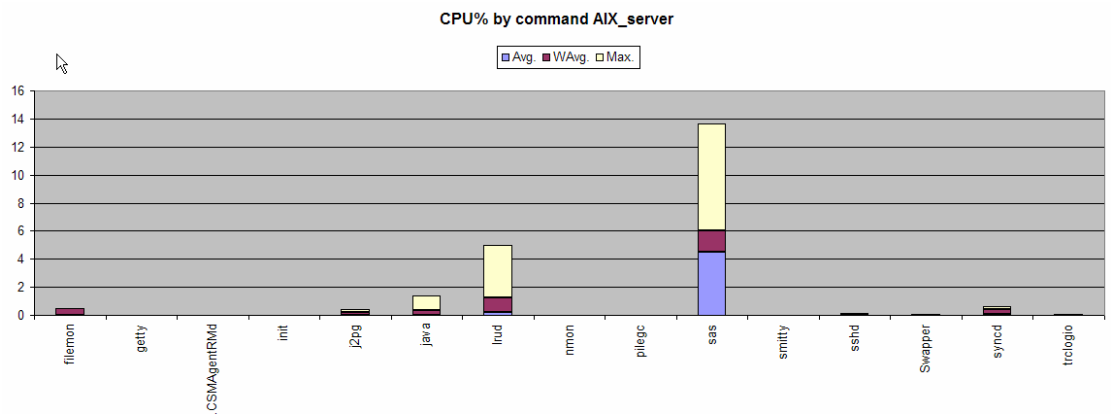

Figure 8 Top processes CPU% utilization.

Figure 8 shows the sas process utilizing close to 14% CPU as its maximum average rate. Some other processes like java and lrud (an AIX system process) using some amount of CPU. Figure 9 shows the memory usage of processes using the most memory.

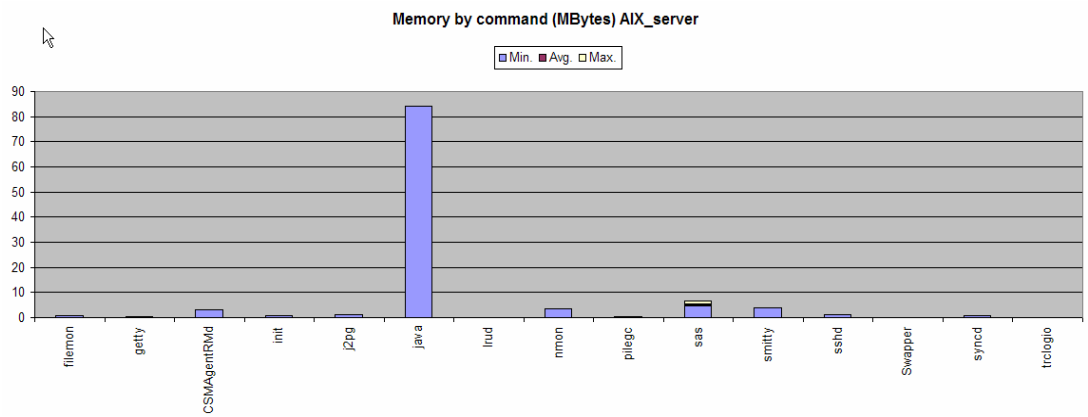

Figure 9 Top processes memory utilization.

In Figure 9, we see the java process using as much as 85Mb of memory while others use less that 10Mb. From this graphs, we are able to see right away which processes are using the most CPU and memory during the test run.

CONCLUSION

Nmon is a powerful tool that can increase the efficiency of analyzing performance on AIX servers. It only requires the knowledge of one command to start the monitoring and data collection process. Once the data is collected, it is then formatted to produce graphs by another tool called nmon analyzer. The graphs produced provide systems administrators and performance engineers an easy to view representation of disk, memory and CPU utilization at different time intervals allowing them to analyze workload bottlenecks.

DOWNLOADING AND INSTALLING NMON

Nmon can be downloaded from:
http://www-941.ibm.com/collaboration/wiki/display/ WikiPtype /nmon
A best practice would be to always download the latest version of the nmon tool. As of this writing the latest version is 11e. This version supports AIX 6.

Untar and unzip the file and start nmon. I like using the following command line arguments.

```
# nmon -fT -s 30 -c 120
```

This command runs nmon for one hour (30 * 120 = 3600 seconds).

```
-s 30  tells nmon to take snapshots every 30 seconds.
-c 120 tells nmon to capture data 120 times
-T tells nmon to capture the top processes and include their arguments
```

For this example, if your test finishes before the whole hour is up, be sure to stop nmon by using “kill –SIGUSR2” to stop it gracefully.

REFERENCES

[1] Crevar, M., Brown, T., Ihnen, L., *Best Practices for Configuring your IO Subsystem for SAS®9 Applications*, Cary, NC: SAS Institute Inc.

[2] Porter, B., et al. 2006. *Storage Best Practices: SAS® 9 with IBM System Storage™, System p5™, and System x™ Considerations for Optimal Storage Layout Version 2.0 08.* New York, NY: IBM Corporation.

[3] Lascu, O., et.al. *AIX 5L Practical Performance Tools and Tuning Guide* SG246478 IBM Redbooks

[4] Tsao, H., et al. *Tuning Guide for SAS®9 on AIX 5L.* IBM Whitepaper. IBM Corporation. http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1882

ACKNOWLEDGMENTS

I would like to thank Margaret Crevar, Tony Brown and Leigh Ihnen of SAS for suggesting and reviewing the contents of this paper.

RECOMMENDED READING

Griffiths, N. “nmon performance: A free tool to analyze AIX and Linux performance” November 2003, updated February 2006. IBM Corporation. http://www.ibm.com/developerworks/aix/library/au-analyze_aix/index.html.

SPECIAL NOTICES

The information contained in this document has not been submitted to any formal IBM test and is distributed “AS IS”. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. The use of this information or the implementation of any techniques described herein is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. Customers attempting to adapt these techniques to their own environments do so at their own risk.

IBM is not responsible for printing errors in this publication that result in pricing or information inaccuracies.

There is no guarantee these measurements will be the same on generally available systems. Actual results may vary. Users of this paper should verify the applicable data for their specific environment.

CONTACT INFORMATION

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

Author: Alfredo Mendoza
E-mail : mendoza1@us.ibm.com