A Hitchhiker's guide for performance assessment & benchmarking SAS® applications

Viraj Kumbhakarna, Cognizant Technology Solutions, Lake Hiawatha, NJ
Anurag Katare, Cognizant Technology Solutions, Lake Hiawatha, NJ

Abstract

Almost every IT department today, needs some kind of an IT infrastructure to support different business processes in the organization. For a typical IT organization the ensemble of hardware, software, networking facilities may constitute the IT infrastructure. IT infrastructure is setup in order to develop, test, deliver, monitor, control or support IT services. Sometimes multiple applications may be hosted on a common server platform. With a continual increase in the user base, ever increasing volume of data and perpetual increase in number of processes required to support the growing business need, there often arises a need to upgrade the IT infrastructure.

The paper discusses a stepwise approach to conduct a performance assessment and a performance benchmarking exercise required to assess the current state of the IT infrastructure (constituting the hardware and software) prior to an upgrade. It considers the following steps to be followed in order to proceed with a planned approach to implement process improvement.

1) Phase I: Assessment & Requirements gathering
   a) Understand ASIS process
   b) Assess AIX UNIX server configuration

2) Phase II: Performance assessment and benchmarking
   a) Server performance
      i) Server Utilization
      ii) Memory Utilization
      iii) Disk Utilization
      iv) Network traffic
      v) Resource Utilization
   b) Process Performance
      i) CPU Usage
      ii) Memory usage
      iii) Disc space

Introduction

The paper discusses a practical assessment exercise that was carried out for benchmarking performance of SAS® processes on a UNIX AIX® 64 server having the SAS® 9.2 Foundation package installed on it. The number of users on the server was close to 20, out of which 5 were power users. The steps in the performance assessment exercise were limited to the performance analysis of the processing capability on the above server. The server is used for processing relatively simple jobs which are used for processing very large volume datasets (approx dataset size ranging from 5Gbs to 150Gbs). Performance was measured for simple, medium and complex jobs which were categorized as simple, medium and complex based on the on the time required for execution, number of distinct processing steps (data or proc steps) in the jobs, processes using "Base statistics" vs. processes using "Advanced analytics".

Phase I: Assessment and requirements gathering

The assessment and requirements gathering forms the first phase of the performance assessment and benchmarking exercise. It sheds some light on answering questions like: Is the current hardware sufficient to support existing business needs? Is it scalable to support the future needs? Is the data storage capacity sufficient to store the current data? Is the storage scalable to handle future requests at the current rate of growth of the data? Are the processing speeds and memory requirements sufficient to support the advanced analytic and base statistics requirements of the users/statisticians accessing the server for data analysis?

Understand AS IS process

It is essential to analyze and benchmark the current process to be able determine whether the existing software setup is able to support the needs of the user base utilizing the server. On the sample AIX UNIX server, considered for this benchmarking exercise, multiple SAS® sessions are executed concurrently by multiple users.

The processing capability of the system is greatly dependent on the IT hardware infrastructure. Primarily, the hardware which impacts system performance is the number and speed of processors; the speed and size of available memory and the I/O speed. The available disk space also plays an important role when the processing for programs consuming huge amount of memory occurs on the disk. The paper limits the discussion to measuring and benchmarking the performance of the existing software processes, keeping the hardware infrastructure component constant. It does not discuss, or take into account any changes to the hardware during benchmarking.

In addition to the system hardware, an analysis of the existing environment is needed, including:

CPU benchmarking
The central processing unit (CPU) is the portion of a computer hardware that does the bulk of the processing of the computer programs, performs the analytical, logical and I/O operations of the system. A typical AIX UNIX server has multiple processors thereby allowing processing of multiple queries concurrently.

Certain procedures in SAS Foundation 9.2® have been modified to take advantage of multiple CPUs by threading the procedure processing. The Base SAS engine also uses threading to create indexes. The CPUCOUNT= option provides the information that is needed to make decisions about the allocation of threads.

As per the SAS 9.2® installation recommendations for UNIX, to verify that you are you using a 64-bit Power processor, enter the following command:

    $ lsconf |egrep 'Processor Type|CPU Type'

The response will be in the following format:
Processor Type: xxx
CPU Type: xxx

If the Processor Type field contains a reference to a Power CPU (for example, PowerPC_POWER4), the CPU Type must be 64-bit in order to use a Power CPU

SAS® runs on AIX using either the 32 or 64-bit kernel; however, the 64-bit kernel is recommended so that system may be able to take advantage of the full functionality of SAS® software. It is suggested to use the 32-bit kernel only if using third-party software requires it. To determine what kernel type the AIX system is running on, perform the following command:

    /usr/sbin/lsconf -k

If the kernel type is 32-bit, the output will be "Kernel Type: 32-bit". Similarly, if the kernel type is 64-bit, the output will be "Kernel Type: 64-bit".

The UNIX command lscfg displays configuration, diagnostic and vital product data (VPD) information about the system including CPU information. Processing speed of the CPU in hertz (Hz) on AIX 5.1 and subsequent releases can be found out from the following code:

    lsattr -E -l proc0 | grep "Processor Speed"

 Alternatively *pmcycles* command can be used to list the processor speed.

System memory benchmarking
Memory refers to a fast, temporary storage area within a computer system which the CPU utilizes to transfer data during processing. As per the SAS® 9.2 installation recommendations for UNIX, for general activities (for example, query and reporting, simple statistics, data manipulation), SAS® recommends 96 MB per user where as 128 MB recommended per concurrent user. More memory is needed for memory-intensive tasks such as OLAP, data mining, heavy statistical analysis, multivariate analysis, formats and any GUI-based applications.
The following command can be used to display the amount of real memory in kilobytes on the AIX UNIX server:
    lsattr -El sys0 | grep realmem

The svmon command can be used to capture a snapshot of virtual memory. It displays information about the current state of memory. The displayed information does not constitute a true snapshot of memory, because the svmon

command runs at user level with interrupts enabled. If no flag value is specified, the svmon command, by default, reports real memory at the system level. To print out global statistics, type:

svmon –G

The amount of memory that is available for a SAS® process to execute is also dependent on the value of the MEMSIZE option set in the process. MEMSIZE specifies the total amount of memory that is available to each SAS® session, and places an enforced limit on the amount of virtual memory that SAS® can dynamically allocate at any one time.

Delineation between processes using "Base statistics" and "Advanced Analytics"

SAS® processes on the server were categorized into simple, medium complex programs depending on a number of different factors such as Real Time & CPU times taken to execute the program, number of datasets used at the source for processing, number of processing steps (data or proc steps), lines of code, volume of data being processed, complexity of the programs, amount of system resources utilized in terms of CPU, Memory and I/O during the execution of the programs, etc. Base statistics constitute of SAS programs performing simple processing activities such as query and reporting, simple statistics, data manipulation, data sub setting, filtering, Extract Transform & Loading (ETL) processes etc. Where is Advanced Analytics constitutes of memory intensive processes such as OLAP, data mining, text mining, heavy statistical analysis (procedures like univariate, annova, etc.), multivariate analysis, formats, graphing and mapping and any GUI-based applications.

A small sample of SAS programs was requested from the business users who utilized the above server for statistical and analytical processing. Users were requested to provide the sample set of programs from their repertoire of analytical processes. Upon discussion with these users, the programs were analyzed based on the above parameters and they were categorized as simple, medium or complex. Following table can act as a guideline but is not necessarily a mandate to categorize the SAS processes. For this exercise, after carefully analyzing the user's programs and upon discussion with the users it was decided that the following parameters formed a good baseline to be able to categorize the programs into simple, medium and complex and be able compare and contrast the SAS programs provided by the users.

Complexity	Real Time for execution	CPU Time for	Volume of datasets in GB	# of datasets at source	# of processing steps (data or proc)	Line of code
SIMPLE	< 2 hrs	< 1 hrs	< 1 GB	< 2 datasets	< 5steps	< 50 lines
MEDIUM	2 < hrs < 10	1 < hrs < 8	1 < GB <10	2 < datasets <10	5 < steps < 20	50 < lines < 400
COMPLEX	< 10 hrs	< 8 hrs	< 10 GB	< 10 datasets	< 20 steps	< 400 lines

Fig. Sample table used for categorizing processes

Data volume transferred throughout processing of SAS sessions

The data volume in the range of terabytes was being carried out through some of the SAS processes thereby consuming an entire core of the processor within a single process for an extended period of time. Such large per session data volumes require significant I/O throughput to maintain the CPU cores in a busy state. On an average SAS process requires 50Mb/sec per core. It is required to work with the service provider to ensure this level of sustained I/O is available at minimum.

Number of concurrent users accessing the system

Performance of the SAS processes is affected by the number of users accessing the system. On the AIX 64 UNIX server considered for benchmark exercise there were approximately 20 users who were accessing the server for processing SAS jobs. There were a number of SAS processes using Base statistics or Advanced analytics that would be executing at the same time on the server. The data volume utilized within processing the SAS programs ranged from 5GB to 100GBs in some of these processes. These processes utilized a shared work space for storing and processing temporary SAS datasets. Disk space of the work environment was appox 600GB.

If a sample complex SAS program was executed multiple times on the same server, it was observed that the Real Time taken to execute the same program ranged from 12 hrs to 17 hrs where as the CPU time was constant. When the utilization of the work environment was measured, it was found that the change in real time was directly proportional to the amount of activity in terms of other user programs executing on the server. For ideal benchmarking, the independent variables should be minimized, such that performance of a program should be measured when no other programs are running on the server.

The **ps** command can be used on the AIX server to display the current state of active processes running on the server. With appropriate flag selections, the ps command examines memory or the paging area and determines what the command name and parameters were when the process was submitted. Please find below, sample commands

that can be used on the UNIX server to measure the number of different concurrent processes executing on the AIX server.

ps -ef | grep -e '-dmr -comamid tcp -device grlink –noterminal' | more

Above command will search for all the processes and the "**e"** flag in the grep command allows searching for one or more patterns. The different patterns specified in the pattern list allow for searching SAS remote sessions, SAS sessions submitted via noterminal and regular sas sessions submitted via command. You may modify the above list and include search patterns specific to the processes you intend to track.

Assess UNIX® Server configuration

An analysis of the existing server configuration is needed to account for accurately determining the optimal software configuration and updating the SAS® configuration file accordingly. This paper restricts the assessment exercise to an AIX® UNIX server with the SAS® 9.2 Foundation installation. It discusses basic steps to assess the server and SAS® Foundation software configuration. It does not offer any suggestions for upgrading the hardware or server sizing. It restricts the discussion to answer some of the questions such as: What is the current hardware of the server? What is the operating system installed on the server? What are the memory size and the number of CPU on the current server?

The lsconf command can be utilized to obtain the basic hardware and configuration details of a UNIX AIX® server. Please find following a sample output of an lsconf command:

System Model: IBM®,7040-681
Machine Serial Number: 835A7AA
Processor Type: PowerPC_POWER4
Number Of Processors: 8
Processor Clock Speed: 1100 MHz
CPU Type: 64-bit
Kernel Type: 64-bit
LPAR Info: 5 JAVADEV1 - kukicha
Memory Size: 10240 MB
Good Memory Size: 10240 MB
Platform Firmware level: 3H041021
Firmware Version: IBM®,RG041021_d78e05_s
Console Login: enable
Auto Restart: true
Full Core: true

prtconf command displays the system configuration that is available. If you run the prtconf command without any flags, it displays the system model, machine serial, processor type, number of processors, processor clock speed, cpu type, total memory size, network information, file system information, paging space information, and devices information. More details regarding prtconf can be found on the UNIX man pages. Output of a prtconf is as below:

System Model: IBM®,XXXX-YYY
Machine Serial Number: 12345ABCD
Processor Type: PowerPC_POWER6
Processor Implementation Mode: POWER 6
Processor Version: PV_6_Compat
Number Of Processors: 8
Processor Clock Speed: 4208 MHz
CPU Type: 64-bit
Kernel Type: 64-bit
LPAR Info: 3 manutd280
Memory Size: 30976 MB
Good Memory Size: 30976 MB
Platform Firmware level: EM340_041
Firmware Version: IBM®,EM340_041
Console Login: enable
Auto Restart: true
Full Core: false

Network Information
      Host Name: hostname
      IP Address: 162.88.22.111
      Sub Netmask: 255.255.255.0

Gateway: 162.86.22.1
Name Server: 162.88.335.222
Domain Name: na.manutd.net

Paging Space Information
Total Paging Space: 4096MB
Percent Used: 4%

Knowing the above configuration helps in determining the optimal configuration settings for the SAS® configuration and helps in determining whether the multithreaded procedures can be used in processing.

Phase II: Performance assessment and benchmarking

Performance measurement is a process to evaluate, using statistical tools and procedures whether the applications are performing up to the standards when compared against a benchmark. A benchmark is an established point of reference against which the processes can be measured in tests comparing the performance.

During the assessment and requirements gathering phase, the server hardware configuration in terms of CPU, Memory and I/O was assessed, applications and processes currently executing on server were identified, a set of sample processes ranging from simple to complex depending on certain predefined and commonly agreed upon parameters were identified for further benchmarking.

The performance assessment and benchmarking exercise breaks down the assessment as per the following two ways: a. Server performance and b. Process performance. The assessment exercise for benchmarking server parameters should go hand in hand with assessment of processes executing on the server. It was observed that the process performance was greatly affected due to the following independent variables affecting the performance of the server, such as, number of concurrent processes executing on the server, volume of data transferred between file systems, complexity of programs executing, resource utilization on the server.

For the sake of the benchmarking exercise, we have considered two user programs: medium and complex. These were categorized into medium and complex using the parameters.

Server performance

The system performance measurement deals with the measurement of utilization of the server resources when the processes are executing on the server. By measuring the performance of the processes in relation to the performance of the server a relationship can be derived between the two.

There are many resource and workload management tools available in the market. These tools have different ways to monitor resource utilization and server performance. These tools are primarily used for managing the workloads by distributing processes over clusters in a massively parallel environment. Such tools allow for dynamic allocation and balancing of system resources such as CPU, memory and I/O without compromising the performance of the applications.

Server Resource Management tool from IBM® Global Services was utilized for benchmarking server performance during the benchmarking exercise. Server Resource Management reports historical trends of key server resources (such as processor, memory, or disk) for all major server platforms. It provides significant insight into server utilization trends, highlights problem areas, enables planning for efficient use of existing resources, and identifies underutilized assets.

Server Utilization

Following are some of the important measures that provide meaningful information regarding performance of the server. In the examples below the benchmarking exercise was carried out on an AIX 64 server having 8 cores, therefore all measures are measured for the 8 CPUs.

On an AIX server, vmstat (virtual memory statistics) is a freely available computer system monitoring tool that collects and displays summary information about operating system memory, processes, interrupts, paging and block I/O. vmstat is usually run with flags to request how long the report run periods should be (interval), and how many periods should be run (count). Flags aren't necessary for CPU statistics only, but it's wise to turn them on if you are going to look at CPU, I/O and Memory from the same set of runs. Users of vmstat can specify a sampling interval which permits observing system activity in near-real time. The vmstat tool is available on most UNIX and Unix-like operating systems.

Vmstat CPU statistics consist of user and system time, CPU idle time, and CPU idle time waiting for pending disk I/O. The first two are most useful for CPU indications. If the user plus the system CPU time consistently stays over 80 percent (never drops below that) in a multi-user environment, the system could be CPU constrained, and further checks are warranted, corroborated with user satisfaction.

Processor utilization
The average processor utilization for each machine throughout the specified collection period.
Units: %
Source: Average of the processor utilization values for the interval selected.

Sys %
The average processor utilization for each machine that the operating system uses to service system calls invoked directly or indirectly by a running program.
Units: %
Source: The sum of the "sy" column from the vmstat command.
Where sy: Time spent running kernel code. (system time)

User %
The average processor utilization for each machine that the user program uses, including any library calls that the program makes.
Units: %
Source: The sum of the "us" column from the vmstat command for all lpars on the box.
Where us: Time spent running non-kernel code. (user time, including nice time)

% I/O Wait
The percentage of time in which the system was idle and had outstanding disk I/O requests. A consistently large value in this field might indicate a disk bottleneck, which is a source of performance problems.
Units: %
Source: The "wa" column of the vmstat command.
Where wa: Time spent waiting for IO. Prior to Linux 2.5.41, shown as zero.

Run Queue
The average number of items in the run queue over the given collection period. The run queue is the device the system uses to schedule processes for processor contention. The run queue consists of the number of processes that are in memory and runnable.

The threshold value depends on the type of machine and the number of processors it contains. Under normal operation, the run queue should be no greater than five times the number of processors. For example, a single processor run queue should be 5 or less. A system with 4 processors should have a run queue of 20 or less. A high run queue is usually a performance problem caused by either a high number of processes on the machine, high processor usage, or I/O blockage.
Units: Numeric
Source: The value of the "r" column from the vmstat command.
Where r: The number of processes waiting for run time.

AVM
The average amount of active virtual memory in use throughout the collection period.
Units: Active Virtual Memory
Source: The value of the "avm" column from the vmstat command.
Where avm are virtual pages (reported in units of 1024 bytes) are considered active if they belong to processes which are running or have run in the last 20 seconds.

Free
The size of available page frames in real memory. Available page frames can have a page of virtual memory written into them. The system maintains a list of available page frames. When the number of available frames declines, paging begins. Excessive paging causes a decline in performance.
Units: Free Memory
Source: The value of "fre" column from the vmstat command.
Where fre: size of the free list

Page Ins/Sec
The number of real memory pages per second paged in from the page space.
Units: Pages/sec
Source: The value of the "pi" column from the vmstat command.
Where pi: pages paged in

Processor > Limit
The percentage of time during collections in which the average processor utilization increased over the 'Limit.'
Units: %
Source: Calculated as the percent of time processor utilization exceeded the threshold.

Run Q > Limit
The percentage of time during statistical collections that the run queue exceeds recommended run queue threshold values.
Units: %
Source: Calculated as the percentage of time the run queue was greater than the threshold.
Page > Limit/Sec
This figure indicates the amount of time that the number of paged in (for AIX? and HP-UX) or swapped in (for Linux) or scanned (for Sun) virtual memory pages was greater than the Limit per second (or Limitond for Sun Scan Rate) threshold. This is a critical performance indicator because a great decline in performance occurs once the system passes this threshold.
Units: %
Source: Calculated as the percentage of time the paging or scanning rate was greater than the threshold.

IO Wait> Limit
This figure indicates the amount of time in which the I/O wait is greater than the Limit% threshold. When this threshold is reached or surpassed constantly, there might be a disk bottleneck. Such bottlenecks can slow response times for all users, not just those with heavy I/O operations.
Units: %
Source: Calculated as the percentage of time I/O wait is greater than the threshold.

No. Of Processors
The number of processor threads. A thread is a logically addressable processor resource. On Intel Servers, enabling hyperthreading results in two threads per core. For UNIX servers, the number represents processors available. For AIX shared processor lpars, the number is physical processors consumed.
Units: Processors
Source: Operating system commands to display processors available, varies among each OS.

Server performance benchmarking strategy
Server performance was measured by executing the sample programs obtained from the users in addition to the other traffic created by other processes executing on the server. For the sake of enumerating an example, we will consider execution of a sample program complex.sas. This program was executed for multiple runs and the server performance was measured using the SRM tool. This strategy allowed us to measure variations in the execution times of the program complex.sas over multiple runs. A comprehensive server utilization output sheet was derived by listing the above stats in a tabular format over multiple runs of the program complex.sas. Following is a sample table of the captured statistics over a single run.

Run No.	Date	Time	Obs	Processor Utilization	Sys %	User %	% I/O Wait	Run Queue	AVM	Free	Page Ins/Sec	Processor > Limit	Run Q > Limit	Page > Limit/Sec	IO Wait> Limit	No. Of Processors
Run1	7/28	1:00:00	60	14.20%	9.85%	4.35%	16.07%	1.58	1764309	265693	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	2:00:00	60	24.68%	15.80%	8.88%	20.55%	1.82	1772051	251052	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	3:00:00	60	22.08%	16.10%	5.98%	21.47%	1.8	1793958	242469	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	4:00:00	60	22.02%	13.68%	8.33%	20.18%	1.67	1778116	289545	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	5:00:00	60	11.35%	7.22%	4.13%	25.97%	1.38	1779540	264761	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	6:00:00	60	24.62%	15.75%	8.87%	21.92%	1.8	1777198	247363	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	7:00:00	60	21.85%	15.52%	6.33%	16.20%	1.62	1767254	253236	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	8:00:00	60	23.03%	12.72%	10.32%	13.50%	1.62	1763832	260900	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	9:00:00	60	35.35%	17.25%	18.10%	11.32%	2.57	1774356	278402	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	10:00:00	60	40.63%	19.03%	21.60%	8.82%	2.82	1779837	267920	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	11:00:00	60	50.72%	22.52%	28.20%	9.18%	4.43	1783648	251100	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	12:00:00	60	45.88%	19.57%	26.32%	8.15%	3.63	1782374	254594	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	13:00:00	60	41.18%	16.45%	24.73%	8.57%	3.18	1781745	281965	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	14:00:00	60	34.92%	14.08%	20.83%	3.58%	2.6	1776462	260632	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	15:00:00	60	26.68%	10.08%	16.60%	0.53%	1.75	1770074	269876	0	0.00%	0.00%	0.00%	0.00%	8
Run1	7/28	16:00:00	60	29.67%	11.92%	17.75%	1.18%	1.87	1767263	256732	0	0.00%	0.00%	0.00%	0.00%	8

Fig. Server utilization statistics over a single run for a sample complex SAS program

By studying the utilization over multiple runs and identifying patterns across multiple runs users can derive meaningful information regarding their processes and how the process interacts with the server performance. By minimizing execution of any other processes executing on the server and limiting the execution of processes to only the process executing for benchmarking, interesting patterns of effects of process execution on system performance can be obtained.

Memory Utilization

Following are some of the important measures that should be measured during measurement of the memory utilization on a server.

Processor utilization
The average processor utilization for each machine throughout the specified collection period.
Units: %
Source: Average of the processor utilization values for the interval selected.

Run Queue
The average number of items in the run queue over the given collection period. The run queue is the device the system uses to schedule processes for processor contention. The run queue consists of the number of processes that are in memory and runnable. The threshold value depends on the type of machine and the number of processors it contains. Under normal operation, the run queue should be no greater than five times the number of processors. For example, a single processor run queue should be 5 or less. A system with 4 processors should have a run queue of 20 or less. A high run queue is usually a performance problem caused by either a high number of processes on the machine, high processor usage, or I/O blockage.
Units: Numeric
Source: The value of the "r" column from the vmstat command.
Where r: The number of processes waiting for run time.

Page Ins/Sec
The number of real memory pages per second paged in from the page space.
Units: Pages/sec
Source: The value of the "pi" column from the vmstat command.
Where pi: pages paged in

Page Outs/Sec
The number of real memory pages paged-out per second from the page space.
Units: Numeric
Source: The value of the "po" column from the vmstat command.
Where po: pages paged out

Scan Rate/Sec
The number of real memory pages scanned per second.
Units: Scan Rate/Sec
Source: The value of the "sr" column from the vmstat command.
Where sr: pages scanned by clock algorithm, per-second

Pages Free'd/Sec
The number of kilobytes freed per second by the paging process.
Units: Kilobytes
Source: The value of the "fr" column from the vmstat command.
Where fr: pages freed per second

Average Scan/Freed Ratio
This column displays the average ratio of pages scanned and pages freed for samples where freed is greater than zero.
Units: Numeric
Source: A vmstat calculated monitor is used to derive the ratio for AIX, HP and Solaris.

Free Pages
The average number of available page frames in real memory. Available page frames can have a page of virtual memory written into them. The system maintains a list of available page frames. When the number of available frames declines, paging begins. Excessive paging causes a decline in performance.
Units: Free Memory
Source: The value of the "fre" column from the vmstat command.
Where fre: size of the free list

File Pages
This column displays the number of 4 KB pages currently used by the file cache.
Units: Count
Source: The File Pages value from the vmstat command.

Pages Available for Applications
This column displays the pages available for application use.
Units: Count
Source: Calculated as "MAX((free pages - minfree),0) + MAX((file pages - minperm), 0)".

%Memory Available for Applications
This column displays the percentage of memory available for applications.
Units: Count
Source: Calculated as memory available for applications divided in memory size.

Memory Average % Used
The percent of installed memory in use.
Units: %
Source: Calculated as 1 - (memory free)/(installed memory).

System Calls
The number of system calls per second.
Units: Numeric
Source: The value of the "sy" column from the vmstat command under the "faults" section.
Where sy: system calls per interval

Context Switches
The number of kernel threads context switches per second.
Units: Numeric
Source: The value of the "cs" column from the vmstat command.
Where cs: cpu context switch rate (switches/interval)

Interrupts
The number of interrupts per second.
Units: Numeric
Source: The value of the "in" column from the vmstat command.
Where in: device interrupts per interval (including clock interrupts)

No. Of Processors
The number of processor threads. A thread is a logically addressable processor resource. On Intel Servers, enabling hyperthreading results in two threads per core. For UNIX servers, the number represents processors available. For AIX shared processor lpars, the number is physical processors consumed.
Units: Processors
Source: Operating system commands to display processors available, varies among each OS.

Following is a sample table of the captured statistics over a single run.

Run No	Date	Time	Obs	Processor Util	Run Queue	Page Ins/Sec	Page Outs /Sec	Scan Rate /Sec	Pages Free'd /Sec	Average Scan/Freed Ratio	Free Pages	File Pages	Pages Available for Applications	% Memory Available for Applications	Memory Average % Used	System Calls	Context Switches	Interrupts	No. Of Processors
Run1	7/28	1:00:00	60	14.20%	1.58	0	0	66398	25667.7	2.60	265693	5705016	4452448	56.14%	96.65%	6471.9	3292.87	3412.97	8
Run1	7/28	2:00:00	60	24.68%	1.82	0	0	1E+05	44868.8	2.46	251052	5715617	4448409	56.09%	96.83%	10049.92	3857.88	3954.18	8
Run1	7/28	3:00:00	60	22.08%	1.8	0	0	1E+05	44620.6	2.38	242469	5702338	4426546	55.82%	96.94%	12616.1	6145.08	5417.35	8
Run1	7/28	4:00:00	60	22.02%	1.67	0	0	84290	36379.6	2.28	289545	5671224	4442508	56.02%	96.35%	11703.25	4141.68	4225.2	8
Run1	7/28	5:00:00	60	11.35%	1.38	0	0	51045	20545.9	2.48	264761	5694452	4440954	56.00%	96.66%	6482	1829.73	1709.57	8
Run1	7/28	6:00:00	60	24.62%	1.8	0	0	1E+05	42294.7	2.53	247363	5714242	4443344	56.03%	96.88%	12260.05	3713.67	3271.87	8
Run1	7/28	7:00:00	60	21.85%	1.62	0	0	1E+05	45556.8	2.53	253236	5718090	4453066	56.15%	96.81%	11259.38	2630.73	1917.47	8
Run1	7/28	8:00:00	60	23.03%	1.62	0	0	99237	39323.5	2.52	260900	5713856	4456496	56.19%	96.71%	9836.8	1503.97	876.38	8
Run1	7/28	9:00:00	60	35.35%	2.57	0	0	1E+05	53524.6	2.55	278402	5686076	4446218	56.06%	96.49%	15317.15	2479.23	1248.23	8
Run1	7/28	10:00:00	60	40.63%	2.82	0	0	1E+05	56626.3	2.55	267920	5691110	4440770	56.00%	96.62%	19639.87	4005.38	1278.13	8
Run1	7/28	10:00:00	60	40.63%	2.82	0	0	1E+05	56626.3	2.55	267920	5691110	4440770	56.00%	96.62%	19639.87	4005.38	1278.13	8
Run1	7/28	11:00:00	60	50.72%	4.43	0	0	1E+05	54834.4	2.55	251100	5698300	4431140	55.87%	96.83%	39821.57	30008.88	2683.53	8
Run1	7/28	12:00:00	60	45.88%	3.63	0	0	1E+05	42762.2	2.48	254594	5701994	4438328	55.96%	96.79%	39107.17	34639.93	6377.42	8
Run1	7/28	13:00:00	60	41.18%	3.18	0	0	78932	32919.7	2.39	281965	5669378	4433084	55.90%	96.44%	38221.08	31501.8	3043.02	8
Run1	7/28	14:00:00	60	34.92%	2.6	0	0	84430	35680.4	2.36	260632	5701686	4444058	56.04%	96.71%	28486.18	15115.47	1665.92	8
Run1	7/28	15:00:00	60	26.68%	1.75	0	0	70900	30435	2.32	269876	5698712	4450328	56.12%	96.60%	16016.93	742.5	667.45	8
Run1	7/28	16:00:00	60	29.67%	1.87	0	0	78930	32861.7	2.40	256732	5714758	4453230	56.15%	96.76%	10275.55	1252.85	856	8

Fig. Memory utilization statistics over a single run for a sample complex SAS program

Disk Utilization
Disk utilization measures the attributes related to amount of commonly used hard drive utilization on the server. Here are listed some of the measures which will

Hard Disk/Size (MB / GB)
The total amount of hard drive space on the server.
Units: Megabytes / Gigabytes
Source: Calculated as the sum of the disk space for all disks on the server.

Hard Disk/Used (MB / GB)
The total amount of hard drive space used.
Units: Megabytes / Gigabytes
Source: Calculated as the sum of the disk used space on the server.

Hard Disk/Free (MB / GB)
The total amount of hard drive space free.
Units: Megabytes / Gigabytes

Hard Disk/% Used
The average percentage of total disk usage for each machine throughout the specified collection period.
Units: %
Source: Calculated as total disk used divided by total space.

File System/Size (MB / GB)
The total amount of file system space measured in megabytes.
Units: Megabytes / Gigabytes
Source: The column is the value of the "1024-blocks" column from the df command.

File System/Used (MB / GB)
The amount of file system DASD used.
Units: Megabytes / Gigabytes
Source: Calculated as the file system size minus the file system free.

File System/Free (MB / GB)
The amount of file system DASD free.
Units: Megabytes / Gigabytes
Source: The file system free size from the df command.

File System/% Used
The percentage of file system DASD used.
Units: %
Source: The percentage of the file system in use from the df command.

Following is a sample table of the captured statistics over a single run.

| Run No. | Date | Status | Type | Hard Disk | | | | | | | File System | | | | | | |
				Size (MB)	Used (MB)	Free (MB)	Size (GB)	Used (GB)	Free (GB)	% Use	Size (MB)	Used (MB)	Free (MB)	Size (GB)	Used (GB)	Free (GB)	% Use
Run1	7/28		Total	1.5E+07	1.5E+07	8	14749	14516	233		1.5E+07	1.1E+07	9	14544	11197	3347	
Run1	7/28		Local	1.5E+07	1.5E+07	2E+05	14749	14516	233	98.42%	-	-	-	-	-	-	-
Run2	8/3		Total	1.5E+07	1.5E+07	8	14749	14516	233		1.5E+07	8199086	42	14544	8007	6537	
Run2	8/3		Local	1.5E+07	1.5E+07	2E+05	14749	14516	233	98.42%	-	-	-	-	-	-	-
Run3	8/3		Total	1.5E+07	1.5E+07	8	14749	14516	233		1.5E+07	8199086	42	14544	8007	6537	
Run3	8/3		Local	1.5E+07	1.5E+07	2E+05	14749	14516	233	98.42%	-	-	-	-	-	-	-
Run3	8/4		Total	1.5E+07	1.5E+07	8	14749	14516	233		1.5E+07	8255443	85	14544	8062	6482	
Run3	8/4		Local	1.5E+07	1.5E+07	2E+05	14749	14516	233	98.42%	-	-	-	-	-	-	-

Fig. Disk utilization statistics over multiple runs for a sample complex SAS program

Disk I/O Statistics
Date
The date that data was collected on the server.
Units: Date
Source: Data file name defined by the collection agent.

Obs
The number of times that data was collected from each server during the specified time period. Data in the table is an average taken over this many specific instances of data collection.

Units: Numeric
Source: Calculated.

No. of Disks on the Server
This column displays how many disks are on the server.
Units: Numeric
Source: Calculated.

No. of Disks Over Any Thresholds
The number of unique disks that were over a threshold at any time during the period.
Units: Number of disks
Source: Calculated as the number of disks that exceeded the threshold at least once during the collection interval.

Avg. No. of Disks Over Any Threshold
The average number of disks that were over a threshold during the period.
Units: Numeric
Source: Calculated as the average number of disks with % tim_act greater than the threshold.

Avg. No. of Disks Over Any Threshold Per Exception
The average number of disks that were over a threshold during the period per exception.
Units: Numeric
Source: Calculated as the average number of disks with % tim_act greater than the threshold when one or more disks were greater than the threshold.

% Time Any Disk Over Any Threshold
The percentage of time one or more disks were over a threshold during the period. For example, assume that two disks were each over a threshold 5.1% of the time. If they were both over at the same time, this column shows 5.1%. If they were over the threshold at different times, this column shows 10.2%.
Units: %
Source: Calculated from the "% tm_act" value from the iostat command.

Following is a sample table of the captured statistics over a single run.

Run No.	Status	Date	Obs	No. of Disks on the Server	No. of Disks Over Any Thresholds	Avg. No. of Disks Over Any Threshold	Avg. No. of Disks Over Any Threshold Per Exception	% Time Any Disk Over Any Threshold
Run1	Y	7/28/2011	95	139	62	3.79	7.97	100.00%
Run2	Y	8/3/2011	95	139	38	2.99	7.65	97.89%
Run3	Y	8/3/2011	95	139	38	2.99	7.65	97.89%
Run3	Y	8/4/2011	95	139	33	0.84	4.48	64.21%
Run4	Y	8/16/2011	95	143	33	1.15	3.83	90.53%
Run5	Y	8/16/2011	95	143	33	1.15	3.83	90.53%
Run5	Y	8/17/2011	95	143	34	1.95	5.59	82.11%
Run6	Y	8/21/2011	95	143	66	3.63	9.76	100.00%
Run6	Y	8/22/2011	95	143	47	2	5.95	81.05%

Fig. Disk I/O statistics over multiple runs for a sample complex SAS program

Network traffic
Date
The date that data was collected on the server.
Units: Date
Source: Data file name defined by the collection agent.

Obs
The number of times that data was collected from each server during the specified time period. Data in the table is an average taken over this many specific instances of data collection.
Units: Numeric
Source: Calculated.

In Packets
The per minute average of inbound network packets for all network cards on the server.
Units: In/Out network packets
Source: The value of the "Ipkts" column from the netstat command
Formula name: ipkts

In Bytes
The per minute average of inbound bytes for all network cards on the server.
Units: numeric
Source: AIX: netstat
Formula name: ibytes

Out Packets
The per minute average of outbound network packets for all network cards on the server.
Units: In/Out network packets
Source: The value of the "Opkts" column from the netstat command.

Out Bytes
The per minute average of outbound bytes for all network cards on the server.
Units: numeric
Source: AIX: netstat

Out Packet Errors
The per minute average of outbound network packet errors for all network cards on the server.
Units: Network packet errors
Source: The value of the "Oerrs" column from the netstat command.

Collisions
The per minute average of network packet collisions for all network cards on the server.
Units: Numeric
Source: The value of the "Coll" column from the netstat command.

Following is a sample table of the captured statistics over a single run.

Run No.	Status	Date	Obs	In Packets	In Bytes	In Packet Errors	Out Packets	Out Bytes	Out Packet Errors	Collisions
Run1	G	7/28/2011	1437	222881	20114	0.00	23565	0	0.00	0
Run2	G	8/3/2011	1438	76241	18744	0.00	9943.3	0	0.00	0
Run3	G	8/3/2011	1438	76241	18744	0.00	9943.3	0	0.00	0
Run3	G	8/4/2011	1436	63612.5	18672	0.00	5969.7	0	0.00	0
Run4	G	8/16/2011	1438	9647.09	18254	0.00	2423.6	0	0.00	0
Run5	G	8/16/2011	1438	9647.09	18254	0.00	2423.6	0	0.00	0
Run5	G	8/17/2011	1438	61344.1	18356	0.00	6285.4	0	0.00	0
Run6	G	8/21/2011	1437	555155	17449	0.00	78240	0	0.00	0
Run6	G	8/22/2011	1438	30486.5	17887	0.00	24508	0	0.00	0

Fig. Network traffic statistics over multiple runs for a sample complex SAS program

Resource Utilization
Processor Utilization
The average processor utilization for each machine throughout the specified collection period.
If the processor utilization is flagged red, this is a possible sign of a looping process. In this case, you should perform further investigations on the machine.
Units: %
Source: Average of the processor utilization values for the interval selected.

Memory Average % Used
This column displays 1 minus the ratio of free memory to AVM + free.
Units: %
Source: Calculated as used memory divided by installed memory.

Hard Disk % Utilization
The average percentage of total disk usage for each machine throughout the specified collection period.
Units: %
Source: Calculated as total disk used divided by total space.

File System % Utilization
The percentage of file system DASD used.
Units: %

Source: The percentage of the file system in use from the df command.

Following is a sample table of the captured statistics over a single run.

Run No.	Date	Processor Utilization	Memory Average % Used	Hard Disk % Utilization	File System % Utilization
Run1	7/28/2011	26.84%	96.54%	98.42%	76.98%
Run2	8/3/2011	19.72%	94.59%	98.42%	55.05%
Run3	8/3/2011	19.72%	94.59%	98.42%	55.05%
Run3	8/4/2011	5.77%	87.64%	98.42%	55.43%
Run4	8/16/2011	8.10%	90.85%	98.47%	63.94%
Run5	8/16/2011	8.10%	90.85%	98.47%	63.94%
Run5	8/17/2011	14.55%	94.21%	98.47%	64.57%
Run6	8/21/2011	13.56%	96.65%	98.47%	64.19%
Run6	8/22/2011	11.15%	95.21%	98.47%	65.65%

Fig. Resource utilization statistics over multiple runs for a sample complex SAS program

Process performance

In the context of this paper, the process performance measurement is related to measuring the performance of the SAS programs and processes identified for benchmarking. Process performance can be measured on a number of different factors such as total time required to execute the process, resources utilized by the process such as memory, I/O, network etc.

Please note, following details regarding the FULLSTIMER options in this section have been referred from the SAS® website http://support.sas.com/rnd/scalability/tools/fullstim/index.html and the paper titled "A Practical Approach To Solving Performance Problems with SAS®" by Tony Brown, SAS Performance Lab, Host Systems R&D.

The SAS System provides the FULLSTIMER option to collect performance statistics on each SAS step, and for the job as a whole and place those in the SAS log. It is important to note that the FULLSTIMER measures only give you a snapshot view of performance at the step and job level. Each SAS port yields different FULLSTIMER statistics based on the host operating system. See the SAS host specific documentation for the exact statistics offered. FULLSTIMER is invoked as a SAS option and takes effect after the option invocation.

Why start with the FULLSTIMER option for monitoring? The best reason is that it tells you what is happening with the SAS system specifically. The statistics it provides are at the job step and can help pinpoint performance problems down to the step. This is extremely helpful in narrowing troublesome activity, and relating it to what your code is telling the system to do. FULLSTIMER measures can be used to help determine if more in-depth performance monitoring with host monitoring or third party tools is indicated.

A   UNIX output for a SAS Data Step is listed below:

```
NOTE: The SAS System used:
      real time                      32:24.49
      user cpu time                  13:49.08
      system cpu time                2:10.62
      Memory                         88903k
      OS Memory                      117544k
      Timestamp                      8/21/2011  8:46:23 PM
      Page Faults                    1423
      Page Reclaims                  166788
      Page Swaps                     0
      Voluntary Context Switches     931603
      Involuntary Context Switches   71231
      Block Input Operations         0
      Block Output Operations        0
```

It is important to know how these numbers are defined and what can be derived from them.

FULLSTIMER Statistics Definition and Interpretation

Following section provides a brief overview of each of the measures from the output of the FULLSTIMER above.

Real Time
The Real Time represents the elapsed time or "wall clock" time. This is the time spent to execute a job or step. This is the time the user experiences in wait for the job/step to complete. As host system resources are heavily utilized the Real Time can go up significantly - representing a wait for various system resources to become available for the SAS job/step's usage.

User CPU Time
The time spent by the processor to execute user-written code. This is user-written from the perspective of the operating system and not the customer's language statements. That is all SAS system code that is not operating system code.

System CPU Time
The time spent by the processor to execute operating system tasks that support user-written code (all CPU tasks that were not executing user-written code). The user CPU time and system CPU time are mutually exclusive.
Starting in SAS 9, some procedures use multiple threads. On computers with multiple CPUs, the operating system can run more than one thread simultaneously. Consequently, CPU time might exceed real-time in your FULLSTIMER output.

For example, a SAS procedure could use two threads that run on two separate CPUs simultaneously. The value of CPU time would be calculated as the following:

CPU1 time + CPU2 time = total CPU time
1 second + 1 second = 2 seconds
Since CPU1 can run a thread at the same time that CPU2 runs a separate thread for the same SAS process, you can theoretically consume 2 CPU seconds in 1 second of real-time.

Memory
Memory represents the amount of memory allocated to that job/step. This does not represent the entire amount of memory that the SAS session is consuming, as it does not reflect any SAS overhead activities (SAS manager, etc.).

Page Faults
Represents the number of virtual memory page faults that occurred during the job/step. Page Faults are pages that required an I/O to retrieve (a read was done to the I/O subsystem).

Page Reclaims
Represents the number of pages retrieved from the page list awaiting re-allocation (all done in memory). These pages did not require I/O activity to obtain.

Page Swaps
The number of times a process was swapped out of main memory.

Voluntary Context Switches
Represents the number of times a process releases its CPU time-slice voluntarily before it's time-slice allocation is expired. This usually occurs when the process needs an external resource, like making an I/O call for more da ta.

Involuntary Context Switches
The number of times a process releases its CPU time-slice involuntarily. This usually happens when its CPU time-slice has expired before the task was finished, or a higher priority task takes its time-slice away.

Block Input Operations
The number of "bufsize" reads that occur. These are I/O operations to read the data into memory for usage. Not all reads have to utilize an I/O operation since the page being requested may still be cached in memory from previous reads.

Block Output Operations
This represents the number of "bufsize" writes that occur. These are the same as block input operations except that they pertain to the writes to files. As in the case of block input operations, not all block outputs will cause an I/O operation. Some files may still be cached in memory.

Identifying performance problems
Performance problems usually involve one or more of the following physical areas:
- • CPU activity
- • Memory activity
- • I/O subsystem activity (disk and file systems) Network activity

By examining FULLSTIMER statistics, and interpreting what is happening with and between the factors producing the measures, we can get a quick idea of where the system is having problems. We can then resort to host-level and third-party measuring tools to obtain a very detailed picture of problem issue. If the host-level and third-party tools give such detail why not use them first? Very simply, there are many tools to use, and each is fairly good at one or more specific areas of investigation, such as CP, Memory, and I/O. Also some require Server Root-Level access to deploy. FULLSTIMER is quick and easy (incorporated in the SAS system), requires no special privileges, you can do it yourself, and it can help quickly narrow the field of things to test next.

The following is a general list of interpretations you can make using FULLSTIMER:

<u>Real Time/CPU Time</u>
 The most valuable way to use FULLSTIMER is to compare timing information. By comparing the Real Time (elapsed time), with the total CPU time (system CPU time plus user CPU time) you can quickly determine if the problem is CPU related.

If the Real time and total CPU time are within 15 percent of each other, this usually indicates that the system is moving data well (at least during the run time of that job/step processing). This means that the ratio of CPU process time is close to that of the total job. This indicates that the system memory, disk system, and file system are getting data to the CPU quickly enough to not be a problem. If you are experiencing bad task performance, and the real and CPU time are within 15 percent of each other, it most likely means that your task is CPU bound. The only way to improve the performance will be to get a faster CPU, split the process over more CPUs (multi-threading or parallel processing), or reengineer the code to be more efficient.

If the Real time and total CPU time are routinely very disparate, (for example if there is a 50 percent margin between them), then you very likely have a problem in your system getting information to the CPU fast enough. Make a closer examination of the Memory and I/O subsystems using the host or third-party tools mentioned in the next section. Other valuable information from FULLSTIMER can be gained by looking at the other statistics:

<u>Memory</u>
 If a sizeable quantity of memory is used and your elapsed time differs greatly from your total CPU time, you may also want to take a close look at your memory using host or third-party tools that are mentioned in the next section.

<u>Involuntary Context Switches</u>
 If Involuntary Context Switches are consistently high across many steps and jobs over long time-periods, then your CPU system is under a heavy load, and you will want to examine that more closely with the tools mentioned in the next section.

<u>Page Swaps</u>
If Page Swaps are consistently high then your memory system is being stressed, and needs more examination. Other statistics like Block Input and Output operations, Page Faults and Reclaims, and Voluntary Context switches can hint at issues, but require more corroboration from the measures previously discussed to make a case for narrowing down investigation. These measures could be high in-and-of themselves without being a symptom of performance problems.

Conclusion

The paper provides the users with information on a host of tools and techniques that can be used for benchmarking system and process performance. It is left to the interpretation of the users on how to make the most of the measures obtained. By studying and identifying patterns for resource utilization over multiple runs, measuring the process performance over multiple runs, and keeping other independent variables constant, users can interpret the measures obtained and utilize the same for diagnosing performances problems with current hardware/processes, identifying exact steps causing the problems and also identifying potential areas for performance improvement.

References

A methodology for solving performance problems with your SAS jobs or applications is documented in the SAS white paper A Practical Approach to Solving Performance Problems with the SAS System (Tony Brown, SAS Performance Lab, Host Systems R & D).

Benchmarking the SAS process performance tool is documented in the SAS white paper "A Benchmarking Technique to Identify Best Practices in SAS ®" authored by Tony Pisegna, ICON Clinical Research, North Wales, PA

Acknowledgements

I would like to profusely thank Carl Feldman and James Maciaszek for being wonderful guides and mentors during my work at Novartis Pharmaceuticals Company. I would also like to thank my Cognizant managers Dhananjay Kelkar, Prakash Pothamsetti and Anuya Aher for encouraging my efforts to document my work and present a paper. I would also thank my company Cognizant Technology Solutions for providing me with an opportunity for attending the SAS Pharmasug 2011 which is when it dawned upon me to write a paper of my own. I would also like to mention the following authors: Tony Brown, Margret Crevar and Tony Pisegna who granted me their kind permissions to reference their work in my paper. Finally, I would also like to thank Neeraj Kumbhakarna for his invaluable review comments and tremendous patience with my work.

IBM® and AIX® are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Contact Information

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

```
Name            : Viraj R Kumbhakarna
Enterprise      : Cognizant Technology Solutions
Address         : 64A Van Wyk Rd.
City, State ZIP  : Lake Hiawatha, NJ, 07034
Phone:          : (973)-220-2246
E-mail:         : vkumbhakarna@gmail.com
                  Viraj.Kumbhakarna@cognizant.com
```

```
Name            : Anurag Katare
Enterprise      : Cognizant Technology Solutions
Address         : 56A Van Wyk Rd.
City, State ZIP  : Lake Hiawatha, NJ, 07034
Phone:          : (862)-579-7988
E-mail:         : anurag.katare@gmail.com
                  Anurg.Katare@cognizant.com
```

IBM® and AIX® are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

SAS® and all other SAS® Institute Inc. product or service names are registered trademarks or trademarks of SAS® Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.