

SarCheck®: Automated Analysis of AIX sar and ps data

(English text version 7.01.00)

This is an analysis of the data contained in the file sar15b. The data was collected on 15/05/2008, from 00:00:00 to 23:50:01, from the system 'aixsysx1'. There were 143 data records used to produce this analysis. The operating system used to produce the sar report was Release 5.3 of AIX. The system configuration data in the sar report indicated that 4.0 processors were configured. 2048 megabytes of memory were reported by the rmss utility.

This introductory paragraph prints the name of the sar report which was analyzed, when the data was collected, the number of records contained in the sar report, and other information about the system environment.

The date format used in this report is dd/mm/yyyy. The date format was set in the sarcheck_parms file.

Data collected by the ps -elf command on 15/05/2008 from 00:00:00 to 23:30:00, and stored in the file ps0515, will also be analyzed. This program will attempt to match the starting and ending times of the ps -elf data with those of the sar report file named sar15b. Data collected by the lspv and lsiv commands and stored in the file fs0515 will also be analyzed.

DIAGNOSTIC MESSAGE: The number of disks in SarCheck's disk table is 3 and the table is 0.067 percent full. The number of entries in SarCheck's ps -elf table is 605 and the table is 6.050 percent full. Today is 24/06/2008 (dd/mm/yyyy). The builddate is 23/06/2008 (dd/mm/yyyy).

The activation key appears to be System independent. Command line used to produce this report:

```
analyze -dtoo -ptoo -pf ps0515 -fs fs0515 -html -png -t -diag -gd /samplereport -hgd ./sar15b
```

If you choose to use the -diag switch, SarCheck will print statistics about the usage of various internal tables. It will also show the arguments used to run the analyze program to help us if we're having trouble duplicating any SarCheck output that you may have questions about.

SUMMARY

The summary lists any bottlenecks detected, changes specified in the sarcheck_parms file, and any problems which may impact the accuracy of the analysis. If anything unusual is seen in the ps -elf data, it will be summarized here too.

When the data was collected, no CPU bottleneck could be detected. **A memory bottleneck was seen.** No significant I/O bottleneck was seen. **A change to at least one tunable parameter has been recommended.** Limits to future growth have been noted in the Capacity Planning section.

Some of the defaults used by SarCheck's rules have been overridden using the sarcheck_parms file. See the Custom Settings section of the report for more information.

RECOMMENDATIONS SECTION

All recommendations contained in this report are based on the conditions which were present when the performance data was collected. It is possible that conditions which were not present at that time may cause some of these recommendations to result in worse performance. To minimize this risk, analyze data from several different days, implement only regularly occurring recommendations, and implement them one at a time or as groups of related parameters.

These are examples of hardware and parameter tuning recommendations. As a rule, SarCheck will recommend small, incremental changes to the system's tunables in order to produce gradual change. In some cases, larger changes will be recommended because this is the only way to go.

Additional memory may improve performance. The recommendation to add memory was triggered by the rate that the page stealer was scanning memory. If possible, borrow some memory for test purposes, and monitor system performance and resource utilization before and after its installation.

NOTE: The following 3 vmo changes should be made all at once.

Change the value of the lru_file_repage parameter from 1 to 0 with the command 'vmo -o lru_file_repage=0'. The -o flag changes the value of a parameter only until the next reboot. To make the change permanent, use the command 'vmo -p -o lru_file_repage=0'. The lru_file_repage parameter is used to change the algorithms used by the LRUD (page stealing daemon).

Change the value of the maxperm% parameter from 30 to 80 with the command 'vmo -o maxperm%=80'. The -o flag changes the value of a parameter only until the next reboot. To make the change permanent, use the command 'vmo -p -o maxperm%=80'.

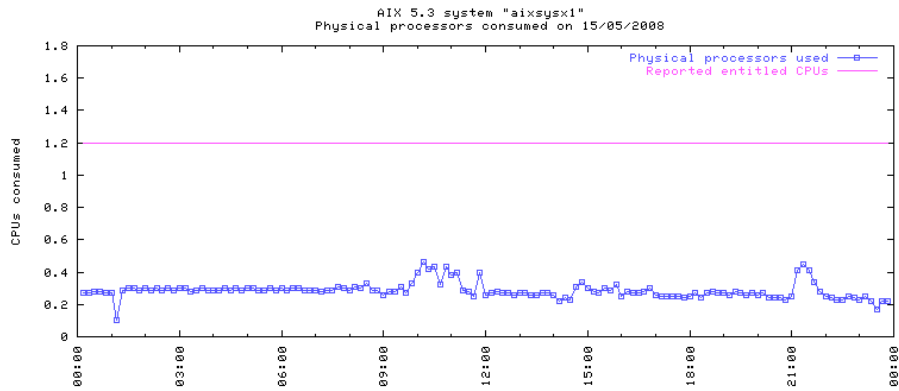
Change the value of the maxclient% parameter from 30 to 80 with the command 'vmo -o maxclient%=80'. The -o flag changes the value of a parameter only until the next reboot. To make the change permanent, use the command 'vmo -p -o maxclient%=80'.

This is the end of this set of vmo parameter changes that should be implemented together.

RESOURCE ANALYSIS SECTION

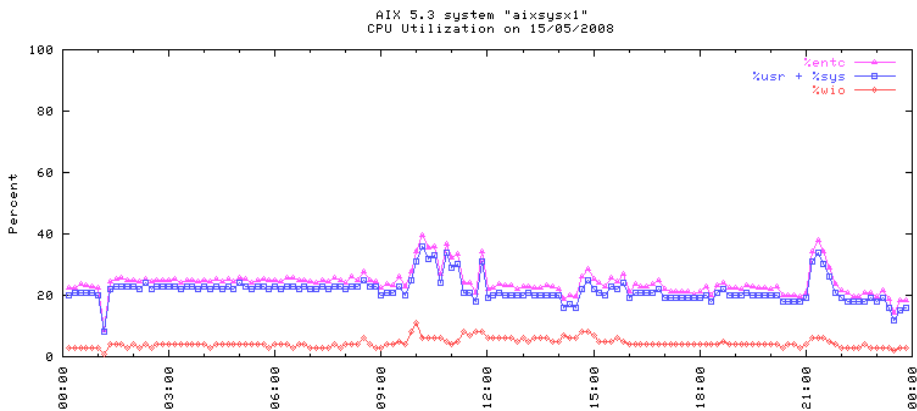
The resource analysis section is the place where various aspects of resource utilization are discussed regardless of whether a problem was seen. The first few paragraphs of this report describe the system's CPU utilization. This system was POWER5-based and was running under AIX 5.3, so much of the analysis concerns the statistics unique to a micropartitioned system. On older systems, SarCheck's output will look different.

An average of 24.16 percent of this partition's entitled CPU capacity (%entc) was used during the monitoring period. The percentage peaked at 39.40 from 10:00:01 to 10:10:01. There were 0.46 physical processors in use when the percentage of entitled CPU capacity was at its peak.



The average number of physical processors consumed by this partition (physc) was 0.29. The peak number of physical processors consumed was 0.46 from 10:00:01 to 10:10:01.

Information in this paragraph is taken from the sar -u report. This information may not be completely accurate on a micropartitioned system and is provided because people are used to seeing it. Average CPU utilization (%usr + %sys) was only 21.7 percent. This indicates that spare CPU capacity exists. If any performance problems were seen during the entire monitoring period, they were not caused by a lack of CPU power. CPU utilization peaked at 36 percent from 10:00:01 to 10:10:01. The CPU was waiting for I/O (%wio) an average of 4.4 percent of the time. The time that the system was waiting for I/O peaked at 11 percent from 09:50:01 to 10:00:01. There was a correlation between the cpu utilization and %waiting to I/O as reported by sar. This correlation is an indication of a healthy system, but the waiting for I/O statistic is not always reliable.



The preceding graph shows the relationship between %entc data and the sum of %usr and %sys. The %entc data is more accurate and should be used instead of the traditional %usr and %sys metrics. The %wio column is probably not very accurate but higher values are likely to indicate times of greater I/O activity. Because the %usr, %sys, and %wio data is not accurate on micropartitioned systems, it has not been used to calculate the percent of time that the system was idle.

Information in this paragraph is taken from the runq-sz and %runocc columns in the sar -q report and may not be completely accurate on a micropartitioned system. The run queue had an average length of 1.5 which indicates that processes were generally not bound by latent demand for CPU resources. The run queue was usually occupied, despite the lack of a significant run queue length. This condition is usually seen when the number of CPU-intensive processes is low. It is likely that the performance of these processes is closely related to CPU speed. Average run queue length (when occupied) peaked at 2.3 from 21:10:01 to 21:20:01. During that interval, the queue was occupied 88 percent of the time. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst during the period of peak CPU queuing, then a performance bottleneck may be the CPU.

The average rate of System V semaphore calls (sema/s) was 5.1 per second. System V semaphore activity (sema/s) peaked at a rate of 5.78 per second from 04:50:01 to 05:00:01. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst during the period of peak semaphore activity, then that activity may be a performance bottleneck and application or database activity related to semaphore usage should be looked at more closely. No problems have been seen, and no changes have been recommended for System V semaphore parameters. Note that SarCheck only checks these parameter's relationships to each other since semaphore usage data is not available.

The average rate of System V (msg/s) message calls was 0.060 per second. No problems have been seen, and no changes have been recommended for System V message parameters. Note that SarCheck only checks these parameter's relationships to each other since message usage data is not available.

There were no times when enforcement of the process threshold limit (kproc-ov) prevented the creation of kernel processes. This indicates that no problems were seen in this area.

The ratio of exec to fork system calls was 0.93. This indicates that PATH variables are efficient.

No buffer cache activity was seen in the sar -b data. This is normal for AIX systems, which typically do not use the traditional buffer cache.

The following paragraphs and graphs show statistics related to memory utilization. In this case, the information was used to determine that there was a shortage of memory.

There was no indication of swapped out processes in the ps -elf data. Processes which have been swapped out are usually found only on systems that have a very severe memory shortage.

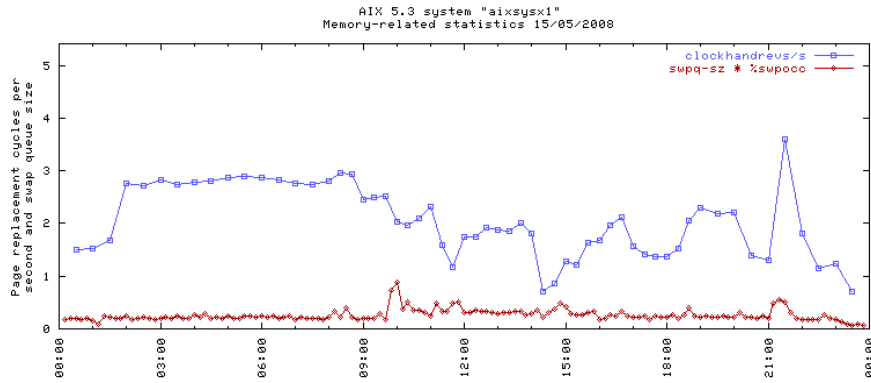
The average number of page replacement cycles per second calculated from the vmstat -s data was 2.02. The number of page replacement cycles per second (from vmstat -s) peaked at 3.61 from 21:00:00 to 21:30:00. This means that the page stealer was scanning memory at a rate of roughly 7392 mb/sec during the peak. We are collecting data on this statistic and have not yet been able to quantify when this value is high enough to indicate a problem. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst during the period of peak replacement cycle activity, then a shortage of physical memory may be a performance bottleneck.

The average number of kernel threads waiting to be paged in (swpq-sz) was 1.48. The average number of kernel threads waiting to be paged in (swpq-sz) peaked at 2.3 from 01:00:00 to 01:10:00. When the peak was reached, the swap queue was occupied 4 percent of the time. A more useful statistic is sometimes available by multiplying the swpq-sz data by the percent of time the queue was occupied. In this case, the average was 0.26 and the peak was 0.88 from

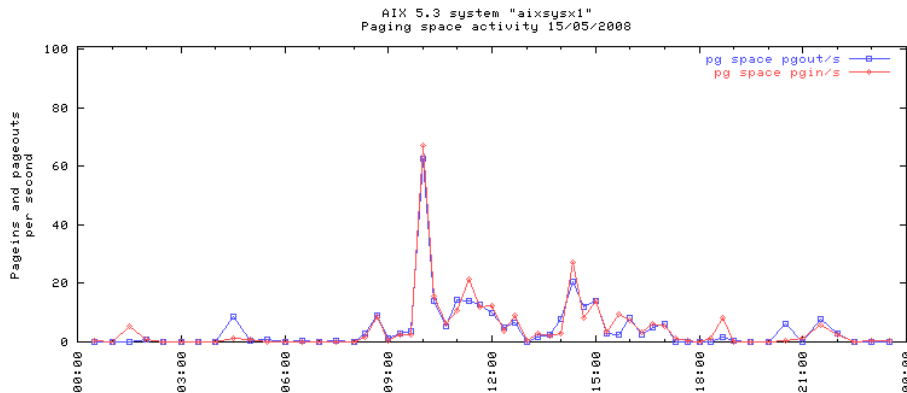
09:50:01 to 10:00:01. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst when the number of kernel threads waiting to be paged in was at its peak, then a shortage of physical memory may be a performance bottleneck.

By looking at the number of revolutions of the clock hand in vmstat -s and knowing how much memory is present, we can approximate how many megabytes of memory are being scanned per second. The granularity of this metric isn't very good unless the system is doing a lot of scanning, but it works out because it's only important when the system is scanning memory quickly. Knowing the amount of memory being scanned per second will help us establish a useful threshold that can tell us if lrubucket should be increased.

The following graph shows any significant statistics relating to page replacement cycle rate, number of kernel threads waiting to be paged in, and number of swapped processes. The page cycle replacement rate has been calculated using the "revolutions of the clock hand" field reported by vmstat -s.

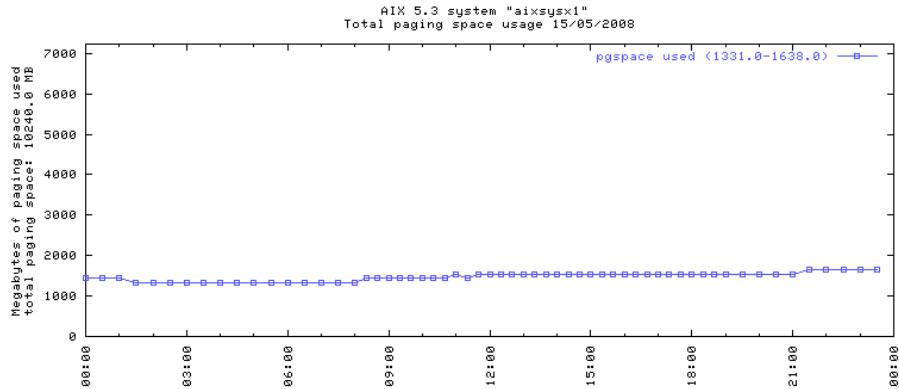


The average page out rate to the paging spaces was 4.85 per second. The paging space page out rate peaked at 62.59 from 09:40:01 to 10:00:00. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst when the paging space page out rate was at its peak, then a shortage of physical memory may be a performance bottleneck. The following graph shows the rate of paging operations to the paging spaces.

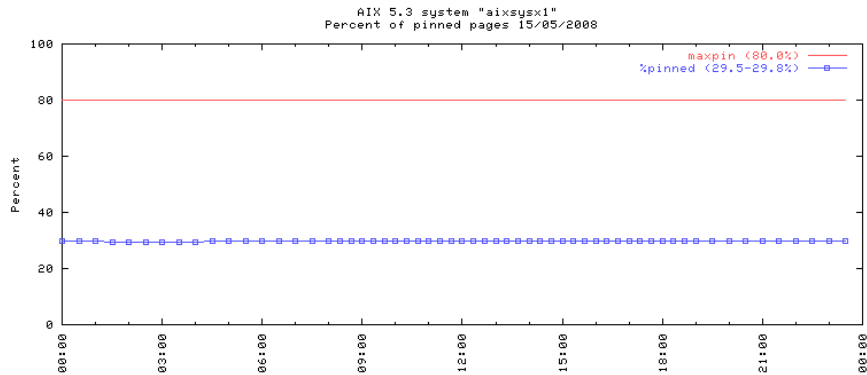


There was 1 paging space seen with the lpsps -a command. The size of the paging space was 10240 megabytes and the size of physical memory was 2048 megabytes. From 21:30:00 to

23:30:00 paging space usage peaked at approximately 1638.0 megabytes, which is about 16 percent of the page space available.



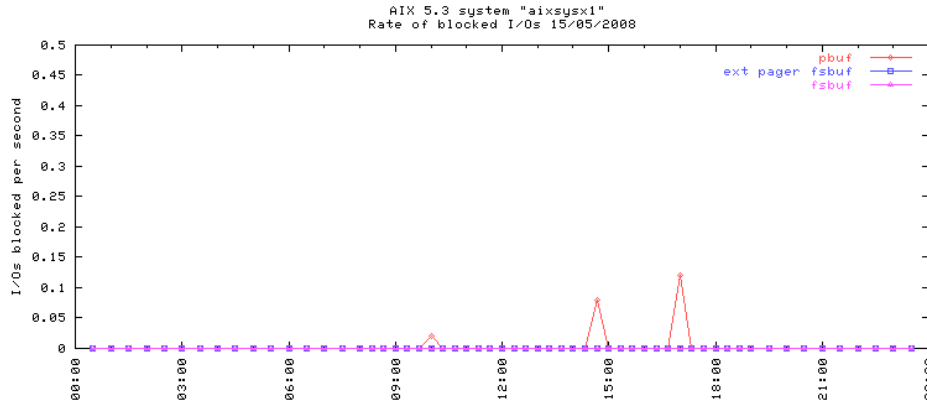
The recorded setting for maxpin% leaves 409.60 megabytes of memory unpinnable. A memory-poor environment was seen even though most of the system's memory was unpinnable.



Information on I/O blocking rates can be seen in the following paragraphs. On AIX 5.3 systems, the counters for this data can be seen with `vmstat -v`. This data is difficult to analyze because it's hard to watch the changes in the values of these counters. The following text and graph show that minimal `pbuf` and `fsbuf` blocking occurred. Because of the lack of blocking and memory-poor environment, SarCheck did not recommend an increase in the number of `pbufs`.

The average rate at which I/O was blocked because an LVM had to wait for `pbufs` was 0.00 per second. The peak rate was 0.12 per second from 16:40:00 to 17:00:00. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst when LVMs had to wait for `pbufs`, then a problem may be that the number of `pbufs` was insufficient.

The average rate at which I/O was blocked because the kernel had to wait for a free `bufstruct` (called `fsbuf` in `vmstat -v`) was 0.00 per second. The peak rate was 0.004 per second from 21:30:00 to 22:00:00. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst when the kernel had to wait for `bufstructs`, then a problem may be that `bufstructs` could not be allocated quickly enough to meet the I/O load.

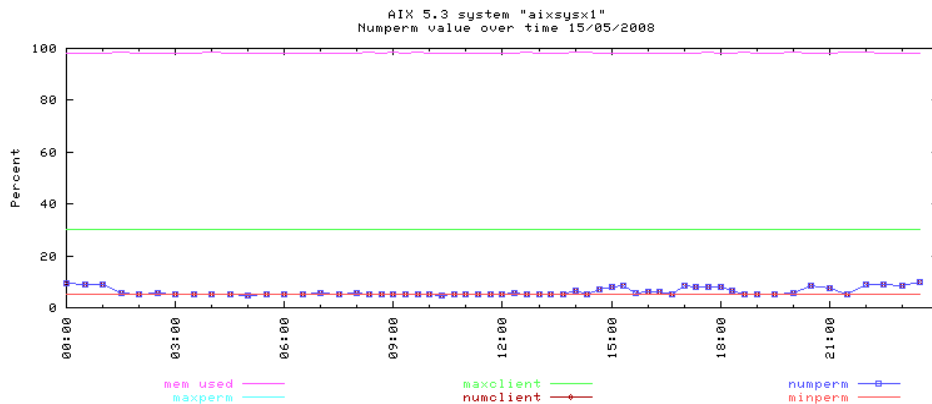


The above graph shows that the rate of I/O blocking is small. The problem does not seem to be significant enough to justify any recommendations and the graph is presented to show the small amount of blocking that was seen.

The average context switch rate (cswch/s) was 2300.03 per second. The context switch rate (cswch/s) peaked at 7464 per second from 11:10:01 to 11:20:01. Peak resource utilization statistics can be used to help understand performance problems. If performance was worst during the period of peak context switching, then a problem may be that too many processes were blocked for I/O or IPC.

The following paragraphs and graph show the changing value of numperm, relative to minperm and maxperm. These are some of the most important tunable parameters in AIX, especially on memory-poor systems.

The following graph and table show the relationship between used memory, maxperm%, maxclient%, numperm, numclient, and minperm%. Because the values of maxperm% and maxclient% are the same, only one can be seen in the graph.



VMM Statistics		
Metric	Average	Range
Memory in use: The percentage of memory being used for either file or non-file pages	98.3%	98.0 - 98.5
Non-file: IBM frequently calls this 'computational memory'.	92.0%	88.3 - 93.3
numperm: Memory which holds all cached file pages (JFS, JFS2, GPFS, NFS, etc.)	6.2%	4.8 - 9.9
numclient: Memory which holds all cached file pages except JFS.	6.2%	4.8 - 9.9
Parameter	Value	
maxperm%	30.0	
maxclient%	30.0	
minperm%	5.0	

I/O bottlenecks are identified in the following paragraphs and where applicable, the values of tunable parameters associated with I/O performance are discussed. If filesystem data is available, filesystems are analyzed as well as physical disks. The data can be presented as a combination of paragraphs of text, an HTML-based table, and JPEG or PNG graphs..

No I/O bottleneck was seen in the sar statistics, therefore no changes are recommended for maxpgahead. The value of minpgahead was set to 2. This is the kind of small value that typically works best in most environments.

No I/O bottleneck was seen in the sar statistics, therefore no changes are recommended for j2_maxPageReadAhead. The value of j2_minPageReadAhead was set to 2. This is the kind of small value that typically works best in most environments.

The value of numclust is 1. If fast disk devices, disk arrays, or striped logical volumes are in use, the performance of disk writes could be improved by increasing this value. SarCheck does not have access to enough information about the system's disk devices to make any specific recommendation for tuning numclust.

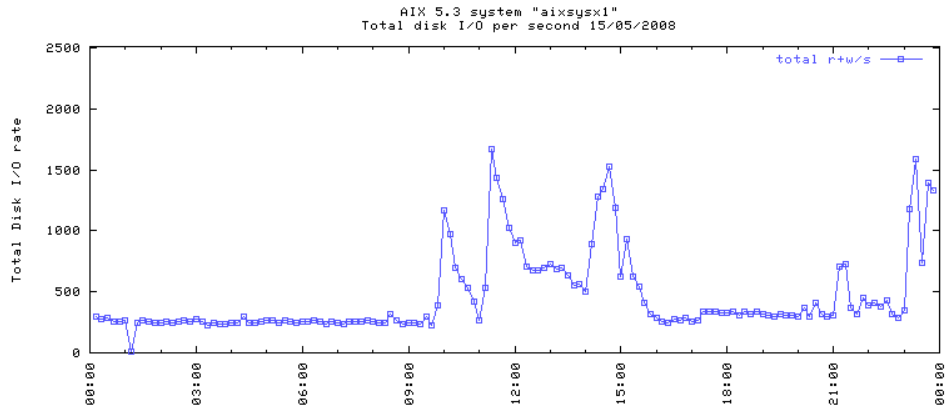
The value of maxrandwrt was 0. This value causes random JFS writes to stay in RAM until a sync operation.

The value of j2nPagesPerWriteBehindCluster was 64. This value determines the number of additional pages to be kept in RAM before scheduling them for I/O when the pattern is sequential.

The value of j2_maxRandomWrite was 0. This value causes random JFS2 writes to stay in RAM until a sync operation.

The value of j2_nRandomCluster was 0. This value determines how far apart writes need to be to be considered random by the JFS2 write behind algorithm.

The average system-wide local I/O rate as measured by the r+w/s column in the sar -d data was 455.7 per second. This I/O rate peaked at 1673 per second from 11:10:01 to 11:20:01. The average size of an I/O based on the r+w/s and kb/s columns was 98.4 kilobytes, or 24.6 pages. The iostat utility reports that 21.6 percent of disk data transferred were writes and the rest were reads. The fact that some filesystems were mounted CIO or DIO will cause the values of numperm and numclient to be lower. This is because CIO and DIO do not use the filesystem caching features of the VMM.



I/O pacing was not in use. A significant amount of fast I/O was seen to at least one disk device and the I/O rate peaked from 11:10:01 to 11:20:01. Consider turning on I/O pacing if interactive performance or keyboard response problems were seen. This is a technique to limit the amount of I/O that a process can perform, typically as a way of preventing batch jobs from hurting interactive response time when high I/O rates are present.

We've created the fsagent that will collect physical and logical volume statistics. The script /opt/sarcheck/bin/fs1 calls the program /opt/sarcheck/bin/fsaixagent. The fsaixagent program collects data from the mount, lsvg and lslv utilities. The data file will be located in the /opt/sarcheck/fs directory with the name format of yyyyymmdd.

The -dto switch has been used to format filesystem statistics into the following table.

Filesystem Statistics										
Filesystem (mounted over)	Type	FS Size	Percent Free	Free Space	Inter policy	Intra policy	MWC	Copies	DIO	CIO
hd5 (N/A)	boot				minimum	edge	on/ACTIVE	2	No	No
hd6 (N/A)	paging				minimum	middle	off	2	No	No
lg_dumplv (N/A)	sysdump				minimum	middle	on/ACTIVE	1	No	No
cbclv (/cbc)	jfs2	384.0 MB	42.47 %	163.1 - 180.9 MB	minimum	middle	on/ACTIVE	2	No	No
lvsuport (/suport)	jfs2	256.0 MB	99.21 %	254.0 MB	minimum	middle	on/ACTIVE	2	No	No
lvtech (/tech)	jfs2	128.0 MB	70.64 %	90.4 - 91.6	minimum	middle	on/ACTIVE	2	No	Yes

				MB						
hd8 (N/A)	jfs2log				minimum	center	off	2	No	No
hd4 (/)	jfs2	128.0 MB	74.41 %	95.2 MB	minimum	center	on/ACTIVE	2	No	No
hd2 (/usr)	jfs2	2688.0 MB	22.08 %	593.5 MB	minimum	center	on/ACTIVE	2	No	No
hd9var (/var)	jfs2	768.0 MB	79.72 %	612.3 - 614.9 MB	minimum	center	on/ACTIVE	2	No	No
hd3 (/tmp)	jfs2	384.0 MB	99.43 %	381.8 MB	minimum	center	on/ACTIVE	2	No	No
hd1 (/home)	jfs2	128.0 MB	93.41 %	119.6 MB	minimum	center	on/ACTIVE	2	No	No
hd10opt (/opt)	jfs2	768.0 MB	55.65 %	427.4 MB	minimum	center	on/ACTIVE	2	No	No

Filesystem [hd5](#) mounted on (N/A) was a mirrored boot filesystem. The inter policy was "minimum" which favors availability over performance. The intra policy was "edge". The edge is a good place to put mirrored filesystems or filesystems where most access is sequential, especially when there isn't a lot of activity in the center. DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume and this favors performance over availability.

Filesystem [hd6](#) mounted on (N/A) was a mirrored paging filesystem. The inter policy was "minimum". The intra policy was "middle". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

Filesystem [lg_dump1v](#) mounted on (N/A) was a sysdump filesystem. The inter policy was "minimum". The intra policy was "middle". The middle is a good place to put data when the center of the disk is full, especially if the data is accessed sequentially. DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

Filesystem [cbclv](#) mounted on (/cbc) was a mirrored jfs2 filesystem. Total size of the filesystem was 384.0 megabytes. The filesystem was 42.47 percent free and 163.1 - 180.9 megabytes of space remained. The inter policy was "minimum". The intra policy was "middle". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

Filesystem [lvsuport](#) mounted on (/suport) was a mirrored jfs2 filesystem. Total size of the filesystem was 256.0 megabytes. The filesystem was 99.21 percent free and 254.0 megabytes of space remained. The inter policy was "minimum". The intra policy was "middle". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

Filesystem [lvtech](#) mounted on (/tech) was a mirrored jfs2 filesystem. This filesystem was mounted with CIO which bypasses the filesystem cache and the VMM read-ahead and write-behind algorithms. Concurrent I/O is the same as DIO but without inode locking. Multiple threads can read and write data concurrently to the same file except if the inode itself needs to change. Total size of the filesystem was 128.0 megabytes. The filesystem was 70.64 percent free and 90.4 - 91.6 megabytes of space remained. The inter policy was "minimum". The intra policy was "middle". Write verification was disabled for this logical volume.

Filesystem hd8 mounted on (N/A) was a mirrored jfs2log filesystem. The inter policy was "minimum". The intra policy was "center". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

Filesystem hd4 mounted on (/) was a mirrored jfs2 filesystem. Total size of the filesystem was 128.0 megabytes. The filesystem was 74.41 percent free and 95.2 megabytes of space remained. The inter policy was "minimum". The intra policy was "center". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

Filesystem hd2 mounted on (/usr) was a mirrored jfs2 filesystem. Total size of the filesystem was 2688.0 megabytes. The filesystem was 22.08 percent free and 593.5 megabytes of space remained. The inter policy was "minimum". The intra policy was "center". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

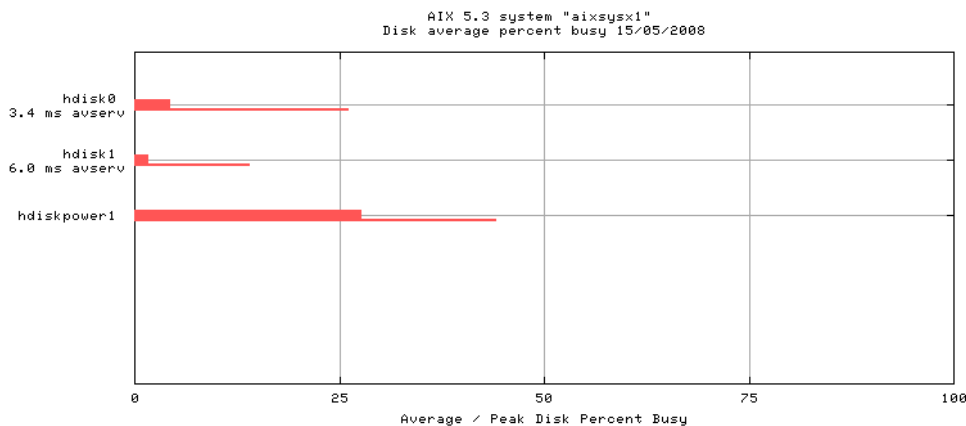
Filesystem hd9var mounted on (/var) was a mirrored jfs2 filesystem. Total size of the filesystem was 768.0 megabytes. The filesystem was 79.72 percent free and 612.3 - 614.9 megabytes of space remained. The inter policy was "minimum". The intra policy was "center". DIO and CIO were not used on this filesystem. Write verification was enabled for this logical volume and this favors availability over performance.

Filesystem hd3 mounted on (/tmp) was a mirrored jfs2 filesystem. Total size of the filesystem was 384.0 megabytes. The filesystem was 99.43 percent free and 381.8 megabytes of space remained. The inter policy was "minimum". The intra policy was "center". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

Filesystem hd1 mounted on (/home) was a mirrored jfs2 filesystem. Total size of the filesystem was 128.0 megabytes. The filesystem was 93.41 percent free and 119.6 megabytes of space remained. The inter policy was "minimum". The intra policy was "center". DIO and CIO were not used on this filesystem. Write verification was enabled for this logical volume.

Filesystem hd10opt mounted on (/opt) was a mirrored jfs2 filesystem. Total size of the filesystem was 768.0 megabytes. The filesystem was 55.65 percent free and 427.4 megabytes of space remained. The inter policy was "minimum". The intra policy was "center". DIO and CIO were not used on this filesystem. Write verification was disabled for this logical volume.

The following graph shows the average/peak percent busy and average service time for 3 disks.



The -dtoo switch has been used to format disk statistics into the following table.

Disk Device Statistics								
Disk Device (vol group)	Avg Pct Busy	Peak Pct Busy	Queue Depth	Avg Svc Time	Disk size	Percent Free	Free Space	Fragmented LVs?
hdisk0 (rootvg)	4.18	26.0	0.2	3.4	68.25 GB	75.46 %	51.50 GB	Yes
hdisk1 (rootvg)	1.47	14.0	0.3	6.0	68.25 GB	76.92 %	52.50 GB	Yes

Powerpath Disk Device Statistics				
Disk Device (vol group)	Avg Pct Busy	Peak Pct Busy	Queue Depth	Avg Svc Time
hdiskpower1 (oracle_vg1)	27.56	44.0	0.0	0.0

Please note that if RAID devices were present, %busy statistics reported for them are likely to be inaccurate and should be viewed skeptically. The presence of a RAID device is generally invisible to the operating system and therefore invisible to this program.

The disk device hdisk0 was busy an average of 4.18 percent of the time and had an average queue length of 0.2 (when occupied). This indicates that the device is not a performance bottleneck. During the peak interval from 09:40:01 to 09:50:01, the disk was 26 percent busy. The average service time reported for this device and its accompanying disk subsystem was 3.4 milliseconds. This is indicative of a very fast disk or a disk controller with cache. Service time is the delay between the time a request was sent to a device and the time that the device signaled completion of the request. The disk device was reported by lspv as being a 68.25 gigabytes disk. 51.50 gigabytes of space was reported as being free and 0.00 gigabytes have been used. This disk is part of volume group rootvg and contained 13 logical volumes. At least one logical volume occupied noncontiguous physical extents on the disk.

- Logical volume hd6 (rootvg), 14 PP (1.75 GB) gap

The disk device hdisk1 was busy an average of 1.47 percent of the time and had an average queue length of 0.3 (when occupied). This indicates that the device is not a performance bottleneck. During multiple peak intervals the disk was 14 percent busy. The average service time reported for this device and its accompanying disk subsystem was 6.0 milliseconds. This is relatively fast. The disk device was reported by lspv as being a 68.25 gigabytes disk. 52.50 gigabytes of space was reported as being free and 0.00 gigabytes have been used. This disk is part of volume group rootvg and contained 0 logical volumes. At least one logical volume occupied noncontiguous physical extents on the disk.

- Logical volume cbclv (rootvg), 65 PP (8.12 GB) gap
- Logical volume lvsuport (rootvg), 66 PP (8.25 GB) gap
- Logical volume lvtech (rootvg), 67 PP (8.38 GB) gap

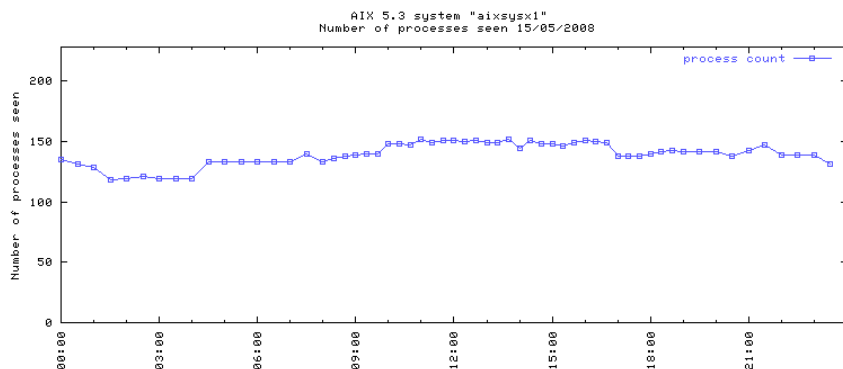
The disk device hdiskpower1 was busy an average of 27.56 percent of the time and had an average queue length of 0.0 (when occupied). This indicates that the device is not a performance

bottleneck. During the peak interval from 21:00:01 to 21:10:01, the disk was 44 percent busy. Peak disk busy statistics can be used to help understand performance problems. If performance was worst when the disk was busiest, then a performance bottleneck may be that disk. The disk device was reported by lspv as being a 33.69 gigabytes disk. This disk is part of volume group oracle_vg1 and contained 1 logical volumes. At least one logical volume occupied noncontiguous physical extents on the disk.

- Logical volume lvorad1 (oracle_vg1), 108 PP (6.75 GB) gap
- Logical volume lvorad1 (oracle_vg1), 215 PP (13.44 GB) gap
- Logical volume lvorad1 (oracle_vg1), 323 PP (20.19 GB) gap

The SarCheck report should include analyzing ps data as well as the sar report. The ps data file will give cpu usage and memory usage on processes and additional kernel tunable parameters values that are included in the analysis. If you include the -ptoo switch in the analyze command line a table and paragraph of ps statistics will be produced identifying runaway processes, memory leaks and/or large processes.

Data collected by ps -elf indicated that there were an average of 139.8 processes present. There were a peak of 152 processes during multiple time intervals. This was the largest number of processes seen with ps -elf but it is not likely to be the absolute peak because the operating system does not store the true "high-water mark" for this statistic.



No runaway processes, memory leaks, or suspiciously large processes were detected in the ps -elf data file.

No table was generated because no unusual resource utilization was seen in the ps -elf data.

Network statistics for ARP, TCP and UDP are included in the report. We'll collect and analyze data from the output of the netstat, no, and arp commands.

ARP packets were sent at an average rate of 0.0008 per second. From 01:30:01 to 02:00:00 the peak was 0.0011 per second. No ARP packets were purged during the monitoring period. There were an average of 0.00 entries per bucket and each bucket was capable of holding 7 entries.

The number of IP packets received per second averaged 32.51 and peaked at 769.62 per second from 21:00:00 to 21:30:00. No action is required because no packets were fragmented. The number of IP packets sent per second averaged 32.76 and peaked at 937.51 per second from

21:00:00 to 21:30:00. According to netstat statistics, 0.0086 percent of these packets being sent were fragmented. Only a small percentage of packets were fragmented so no action is required.

According to netstat statistics, the rate of IP fragments dropped after a timeout averaged 0.00 per second. The value of ipfragttl, the time to live for ip fragments, was 2 half-seconds.

The number of TCP packets received per second averaged 29.91 and peaked at 767.13 per second from 21:00:00 to 21:30:00. According to netstat statistics, 0.0049 percent of the packets received were duplicates. Only a small number of duplicate packets were received. This is normal and no action is required. The number of TCP packets sent per second averaged 31.81 and peaked at 936.58 per second from 21:00:00 to 21:30:00. According to netstat statistics, only 0.0001 percent of the packets were retransmitted. This is normal and no action is required.

The value of rfc1323 was 0, this is good for networks using a small MTU size.

The value of sb_max, the maximum buffer size for TCP and UDP sockets, was 1048576. This is the usual default and is recommended for most systems.

According to netstat statistics, the rate of TCP retransmit timeouts averaged 0.0000 per second and peaked at 0.00 per second from 04:30:00 to 05:00:00. The value of tcp_ttl, the time to live for TCP packets, was 60 ticks (hundredths of a minute).

The value of tcp_recvspace was 16384 and the value of tcp_sendspace was 16384. The minimum MTU size seen for this system's network connection as seen with netstat -in was 1500 bytes. The interface used to find this value was en2. Parameters are correct if this interface is a 10mb or 100 mb ethernet.

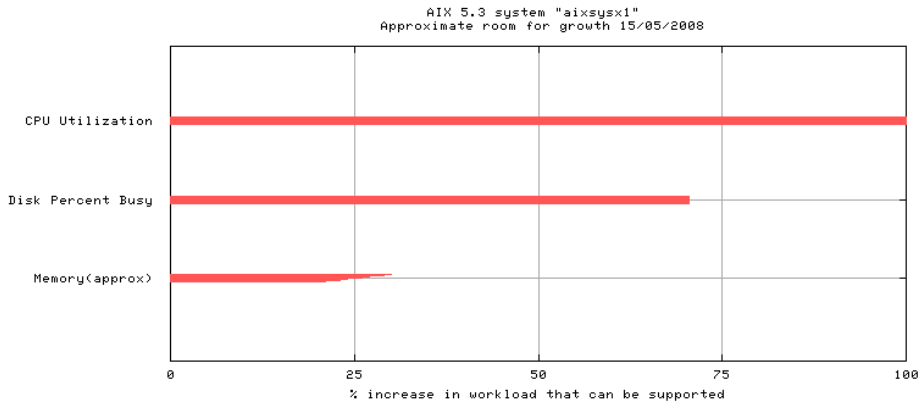
This is only our first attempt to monitor the udp_recvspace and udp_sendspace parameters and it is not complete. IBM has recommendations for these parameters that should not be overlooked and we hope to have more detailed recommendations available shortly. According to netstat statistics, no UDP socket buffer overflows were seen during the monitoring period.

CAPACITY PLANNING SECTION

The Capacity Planning section can help you to understand how much additional load your system can support. This feature is not designed to replace the features found in mainframe-type capacity planning tools, but rather to give you an approximation of how much room for growth remains. Please note the disclaimer in the paragraph above.

This section is designed to provide the user with a rudimentary linear capacity planning model and should be used for rough approximations only. These estimates assume that an increase in workload will affect the usage of all resources equally. These estimates should be used on days when the load is heaviest to determine approximately how much spare capacity remains at peak times.

Based on the data available, the system should be able to support approximately a 25 percent increase in workload at peak times before the first resource bottleneck affects performance or reliability, and that bottleneck is likely to be memory utilization. See the following paragraphs for additional information.



In the above graph, the end of the memory bar is tapered because it represents more of an approximation than the others. The CPU can support an increase in workload of at least 100 percent at peak times. Since a non-trivial level of page outs and/or swapping were detected, the amount of memory present will have trouble supporting an increase in workload of more than roughly 25 percent at peak times. The busiest disk can support a workload increase of approximately 70 percent at peak times. For more information on peak resource utilization, refer to the Resource Analysis section of this report.

CUSTOM SETTINGS SECTION

The default GRAPHDIR was changed with the -gd switch to /samplereport.

The default HSIZE value was changed in the sarcheck_parms file from 0.70 to 1.20 times the default gnuplot width. This was done with the WIDE entry in the sarcheck_parms file.

You can change SarCheck's algorithms and output format by creating a file named /opt/sarcheck/etc/sarcheck_parms. For more information about what can be changed with sarcheck_parms, see [Appendix A](#).

Please note: In no event can Aptitude Corporation be held responsible for any damages, including incidental or consequent damages, in connection with or arising out of the use or inability to use this software. All trademarks belong to their respective owners.

This software provided for the exclusive use of: Your Company. This software expires on 23/07/2008 (dd/mm/yyyy). Code version: 7.01.00. Serial number: 19201299.

(c) copyright 1995-2008 by Aptitude Corporation, Plaistow NH, USA, All Rights Reserved.
<http://www.sarcheck.com>

This is a collection of data stored in a comma separated value format for easy import into a spreadsheet. The -html switch was used, displaying this data as an HTML table. To produce data in this format at the end of the report, include the -t or -tonly options.

Statistics for system, aixsysx1				
		Start of peak interval	End of peak interval	Date of peak interval
System ID on sar report,	00CAD88G2C00			
System ID of this system,	000481674C00			
System model number is,	IBM Model 7042/7043 (ED)			
Statistics collected on,	15/05/2008			
Average phys processors consumed,	0.29			
Peak phys processors consumed,	0.46	10:00:01	10:10:01	15/05/2008
Average entitled capacity consumed,	24.16%			
Peak entitled capacity consumed,	39.4%	10:00:01	10:10:01	15/05/2008
Average CPU utilization,	21.7%			
Peak CPU utilization,	36%	10:00:01	10:10:01	15/05/2008
Average user CPU utilization,	7.1%			
Average sys CPU utilization,	14.5%			
Average waiting for I/O,	4.4%			
Average run queue length,	1.5			
Peak run queue length,	2.3	21:10:01	21:20:01	15/05/2008
Average run queue occupancy,	79.1%			
Average swap queue length,	0.26			
Peak swap queue length,	0.88	09:50:01	10:00:01	15/05/2008
Peak page replacement cycle rate,	3.76	21:10:01	21:20:01	15/05/2008
Max paging space page outs,	62.59	09:40:01	10:00:00	15/05/2008
Max paging space page ins,	67.23	09:40:01	10:00:00	15/05/2008