APPENDIX A

AIX performance tools summary

This appendix is a summary, meant as a quick reference, of the most important AIX commands that can help in monitoring and tuning RDBMS performance on AIX. Many of these commands, including the commands that update AIX parameters, require *root* permission to run. Only the most commonly used options are provided for each command. Please consult the following references for more detailed information related to monitoring AIX system performance:

- *- AIX V 4.3 Commands Reference, SBOF-1877*
- *- AIX Performance Tuning Guide, SC23-2365*
- *- Understanding IBM RS/6000 Performance and Sizing redbook, SG24-4810*
- *- RS/6000 Performance Tools in Focus redbook, SG24-4989*

AIX standard documentation can be found at:

```
http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixgen/
```

IBM Redbooks can be found at:

```
http://www.redbooks.ibm.com
```

filemon - File I/O Monitor

The `filemon` command is used to monitor the performance of the file system and report the I/O activity on behalf of files, virtual memory segments, logical volumes, and physical volumes. The global reports list the most active files, segments, logical volumes, and physical volumes during the measured interval. They are shown at the beginning of the filemon report. By default, the logical file and virtual memory reports are limited to the 20 most active files and segments, respectively, as measured by the total amount of data transferred. If the `-v` flag has been specified, activity for all files and segments is reported. All information in the reports is listed from top to bottom as most active to least active.

Syntax: `filemon -i file -o file -d -Tn -P -v _O levels`

Example: `filemon -O all -o file.out`

Start workload (in a production system workload is usually already present) and then stop trace activity with `trcstop`.

Most Active Files report

Column	Description
#MBS	Total number of megabytes transferred to/from file. The rows are sorted by this field, in decreasing order.
#opns	Number of times the file was opened during measurement period.
#rds	Number of read system calls made against the file.
#wrs	Number of write system calls made against the file.
file	Name of the file (the full path name is in the detailed report).
volume:inode	Name of the logical volume that contains the file, and the file's i-node number. This field can be used to associate a file with its corresponding persistent segment, shown in the virtual memory I/O reports. This field may be blank; for example, for temporary files that are created and deleted during execution.

Most Active Segments report

Column	Description
#MBS	Total number of megabytes transferred to/from segment. The rows are sorted by this field, in decreasing order.
#rpgs	Number of 4096-byte pages read into segment from disk (page in).
#wpgs	Number of 4096-byte pages written from segment to disk (page out).
segid	Internal ID of segment.
segtype	Type of segment: working segment, persistent segment (local file), client segment (remote file), page table segment, system segment, or special persistent segments containing file system data (log, root directory, .inode, .inodemap, .inodex, .inodexmap, .indirect, .diskmap).
volume:inode	For persistent segments, name of logical volume that contains the associated file, and the file's inode number. This field can be used to associate a persistent segment with its corresponding file, shown in the file I/O reports. This field is blank for non-persistent segments.

> **Note:** The vir tual memory analysis tool, svmon can be used to display more information about a segment, given its segment ID (segid), as follows: `svmon -S <segid>`.

Most Active Logical Volumes report

Column	**Description**
util	Utilization of the volume (fraction of time busy). The rows are sorted by this field, in decreasing order.
#rblk	Number of 512-byte blocks read from the volume.
#wblk	Number of 512-byte blocks written to the volume.
KB/sec	Total transfer throughput, in Kilobytes per second.
volume	Name of volume.
description	Contents of volume: either a file system name, or logical volume type (paging, jfslog, boot, or sysdump). Also, indicates if the file system is fragmented or compressed.

Most Active Physical Volumes report

Column	**Description**
util	Utilization of the volume (fraction of time busy). The rows are sorted by this field, in decreasing order.
#rblk	Number of 512-byte blocks read from the volume.
#wblk	Number of 512-byte blocks written to the volume.
KB/sec	Total volume throughput, in Kilobytes per second.
volume	Name of volume.
description	Type of volume, for example, 120MB disk, 355MB SCSI, or CDROM SCSI

iostat - Disk I/O Statistics

The `iostat` command is used to report CPU and I/O statistics for TTY devices, disks, and CD-ROMs. It is used to generate reports that can be used to change the system configuration to better balance the input/output load between physical disks.

Syntax: `iostat interval count`

Flag	Meaning
interval	number of second between outputs
count	number of times to output

Examples:

`iostat 10 20`	20 lines output with 10 seconds between each

Report Output:

%tm_act	Percentage of time active
Kbps	Kilobytes per second transferred
tps	Transfers per second
msps	Milliseconds per seek (if available)
Kb_read	Total Kilobytes read (likewise for write)

lsattr - List attributes

The `lsattr` command lists the attributes of AIX resources. The sys0 resource includes performance statistics.

Syntax: `lsattr -El sys0`

Examples:

`lsattr -El sys0`

outputs details of AIX parameters including minpout and maxpout
Flag Meaning

lscfg - List configuration

The `lscfg` command lists the details of the machine.

Syntax: `lscfg [-v]`

Flag	Meaning
-v	outputs the full details

Examples:

`lscfg`	outputs details of the machine
`lscfg -v`	outputs the full details, part numbers and levels

lsdev -List devices

The `lsdev` command lists the details of the devices in the machine.

Syntax: `lsdev -C`

Syntax: `lsdev -Cc class`

Flag	**Meaning**
-Cc	outputs device details for one class only

Examples:

`lsdev -C`	outputs details of all devices
`lsdev -Cc class`	output of a particular class (memory, disk, tape, ...)

lslpp - List licensed program produce

The lslpp command lists the packages, filesets and files loaded in the AIX system.

Syntax: `lslpp [-lLa <fileset>] [-f <fileset>] [-w <filename>]`

Flag	**Meaning**
-l <fileset>	outputs the most recent levels of the fileset
-La <fileset>	outputs the full details and updates of the fileset
-f <fileset>	outputs the files within a fileset
-w <file>	outputs the fileset the file belongs too

Examples:

`lslpp -l “bos.rte.*”`	outputs levels of this fileset
`lslpp -La “bos.rte*”`	outputs above plus update information
`lslpp -f “bos.rte”`	outputs the files of this fileset
`lslpp -w “/usr/bin/vi”`	outputs the fileset this file belongs too
`lslpp -w “*installp*”`	outputs the files that contain any filename that includes the directory or filename installp

lslv - List logical volume

The `lslv` command lists the details of the logical volume and their placement on the disks.

Syntax: `lslv [-l] <volume group name>`

Flag	Meaning
-l	outputs the placement on the disks
Examples:	
`lslv lv00`	outputs details of the logical volume
`lslv -l lv00`	outputs the placement of the logical volume on disks (physical volumes)

lsps - List Paging Space

The `lsps` command displays the characteristics of paging spaces, such as the paging space name, physical volume name, volume group name, size, percentage of the paging space used and whether the space is active or inactive.

Syntax: `lsps -a -s [paging space]`

Flag	Meaning
-a	displays all paging spaces
-s	displays summary of all paging spaces
Examples:	
`lsps -a`	lists the characteristics of all paging spaces

lspv - List physical volume

The `lspv` command lists the details and contents of physical volumes (disks).

Syntax: `lspv [-p] [-l] <hdisk name>`

Flag	Meaning
-p	outputs contents and placement
-l	outputs contents of the physical volume
Examples:	
`lspv hdisk0`	outputs volume group names only
`lspv -l hdisk22`	outputs details about disk hdisk22
`lspv -p hdisk22`	outputs more details about disk hdisk22

lsvg - List volume group

The `lsvg` command lists the names of the volume group, their contents and their details.

Syntax: `lsvg [-i] [-l] <volume group name>`

Flag	Meaning
-i	takes volume group names from standard input
-l	outputs the details of a volume group

Examples:

`lsvg`	outputs volume group names only
`lsvg rootvg`	outputs the details of the volume group called `rootvg`
`lsvg -l rootvg`	outputs details of volume group called `rootvg`

ncheck - Inode Check

The `ncheck` command is used to display the i-node numbers and path names for file system files.

Syntax: `ncheck [-a][-i inodenumber...] [-s] [filesystem]`

Flag	Meaning
-a	lists all file systems including those starting with '.' and '..'
-i inode	finds the file(s) with these inode numbers
-s	lists special and set UID files

Examples:

`ncheck -a /`	lists all files in the '/' file system
`ncheck -i 2194 /tmp`	finds the name for inode 2194 in /tmp

netpmon - Network Monitor

The `netpmon` command is used to monitor and report on network activities and network related CPU usage. It uses the AIX system trace to gather information.

Syntax: `netpmon -o file -Tn -P -v -O report-type`

Flag	Meaning
-o outputfile	puts the output to a file, not stdout
-T n	sets the output buffer size (default 64000)
-P	forces the monitor process into pinned memory
-v	Verbose (default only top 20 processes)

-O	Allows the selection of one of the following options: cpu, dd(device driver), so(socket), nfs, all

Example: `netpmon -O all -o net.out`

Start workload (see filemon) and then stop trace activity with `trcstop`.

nfsstat - Network File System statistics

The `nfsstat` command lists the NFS details.

Syntax: `nfsstat`

Examples:

`nfsstat`	outputs all NFS statistics

nmon - online monitor

The nmon command is used to display all the AIX statistics on one screen and updates them every 2 seconds. When running hit h for further help on the options or hit q to quit. An alternative mode saves the same data to a file that can be loaded into a spreadsheet.

Syntax: `nmon [-?][-fdt]`

Flag	**Meaning**
-?	output help information on running nmon
-fdt	run in file output mode, including disk and top process statistics

Note: this tool is not supported by IBM and no warranty is given or implied by including this tool in this redbook. It is available to IBM at:

`http://w3.aixncc.uk.ibm.com`

and to IBM Business Partners via PartnerInfo.

no - Network options

The no command lists the details of the network options.

Syntax: `no -a`

Flag	**Meaning**
-a	outputs all options

Examples:

`no -a`	outputs all network options

ps - Process State

The `ps` command is used to display the status of currently active processes.

Syntax: `ps -a -e -f -l -p plist -u user -x .`

Flag	Meaning
-a	writes information about all processes, except the session leaders and processes not associated with a terminal to standard output
-e	lists every user’s process
-f	full listing
-l	long listing
-p	pid lists the process number N
-u	user lists the specified user's processes (-u fred)

Examples:

`ps -fu jim`	Lists user jim’s processes in full
`ps -lt 04`	List all processes on terminal tty04
`ps -fe`	List all processes

Report column headings	Meaning
PID/PPID	Process IDentity & Parent Process IDentity
S	State= Running, Sleeping, Waiting, Zombie, Terminating, Kernel, Intermediate
UID/USER	User IDentity/User name
C	CPU recent use value (part of priority)
STIME	Start time of process
PRI	Priority (higher means less priority)
NI	NIce value (part of priority) default 20
ADDR	ADDRess, of stack (segment number)
SZ	SiZe of process in 1K pages
CMD	Command the user typed (-f to display more)

WCHAN	Event awaited for (kernel address)
TTY	Terminal processes connected to (- = none)
TIME	Minutes and Seconds of CPU time consumed by the process
SSIZ	Size of kernel stack
PGIN	# of pages paged in
SIZE	Virtual size of data section in 1K's
RSS	Real memory (resident set) size of process 1K's
LIM	Soft limit on memory xx=none
TSIZ	Size of text (shared text program) image
TRS	Size of resident set (real memory)
%CPU	Percentage of CPU used since started
%MEM	Percentage of real memory used

rmss - Reduced Memory System Simulator

The `rmss` command is used to simulate a system with various sizes of real memory that are smaller than the actual amount of physical memory installed on the machine.

Syntax: `rmss -p -c M -r`

Flag	**Meaning**
-p	print the current value
-c M	changes to size M (in Mbytes)
-r	restores all memory to use

Examples:

`rmss -c 32`	Change available memory to 32 Mbytes
`rmss -r`	Undo the above

sar - System Activity Reporter

The `sar` command is a standard UNIX command, used to gather statistical data about the system.

Syntax: `sar -A -o savefile -f savefile -i secs -s HH[:MM[:SS]] -e HH[:MM[:SS]] -P ALL interval number`

Flag	**Meaning**
-A	All stats to be collected/reported
-o savefile	Collect stats to binary file
-f savefile	Report stats from binary file
-i secs	Report at secs interval from binary file
-s and -e	Report stats only between these times
-P ALL	Report on all CPU stats
CPU related output	
%usr %sys	Percent of time in user / kernel mode
%wio %idle	Percent of time waiting for disk io/idle
Buffer Cache related output	
bread/s bwrit/s	Block I/O per second
lread/s lwrit/s	Logical I/O per second
pread/s pwrit/s	Raw disk I/O (not buffer cached)
%rcache %wcache	Percentage hit on cache
Kernel related Output	
exec/s fork/s sread/s swrite/s	Calls of these system calls per second
rchar/s wchars/s scall/s	exec and fork are used for process creation sread/ swrite system calls (files, raw, tty or network). rchar/ wchar the numbers of characters transferred scall is the total number of system calls per second
msg/s sema/s	Inter-process communication (IPC) for messages and semaphores
kexit/s ksched/s kproc-ov/s	Process exits, process switches and process overload (hit proc thresholds)
runq-sz	Average process on run queue
%runocc	Percentage of time with process on queue
swap-sz	Avgerage process waiting for page in
%swap-occ	Percentage of time with process on queue
cycles/s	# of page replace search of all pages
faults/s	# of page faults

slots	# of free pages on paging spaces
odio/s	# of non-paging disk I/O per second
file-ov, proc-ov	# of times these tables overflow per second
file-sz inode-sz proc-sz	Entries in the tables
pswch/s	Process switches per second
canch/s outch/s rawch/s	Characters per second on terminal lines
rcvin/s xmtin/s	Receive and transmit interrupts per second

Examples:

sar 10 100	Reports now at 10 second intervals
sar -A -o fred 10 6 >/dev/null	Collects data into fred
sar -A -f fred	Reports on the data
sar -A -f fred -s 10:30 -e 10:45	Reports for 15 minutes starting at 10:30 a.m.
sar -A -f fred -i60	Reports on a 1 minute interval rather that 10 seconds as collected
sar -P ALL 1 10	Reports on each CPU or the next 10 seconds

schedtune - Process Scheduling Tuning

The `schedtune` command is used to set the parameters for the CPU scheduler and Virtual Memory Manager processing.

Syntax: `schedtune -h sys -p proc -w wait -m multi -e grace -f ticks -t time_slice -D (default)`

Flag	Meaning
-h 6	Sets system wide criteria for when process suspension begins and ends (thrashing)
-p 4	Sets per-process criteria for determining process suspension begins and end
-w 1	Seconds to wait before trashing ended
-e 2	Seconds exempt after suspension
-f 10	Clock tick waited after fork failure

-t 0	Clock tick interrupts before dispatcher called
-D	Restore default values

Examples:

`schedtune -t5`	set time slice to 50 ticks
`schedtune`	Report current settings

svmon - System Virtual Memory Monitor

The `svmon` command is used to capture and analyze a snapshot of virtual memory.

Syntax: `svmon -G -Pnsa pid... -Pnsa[upg][count] -S sid... -i secs count`

Flag	**Meaning**
-G	Global report
-P[nsa] pid..	Process report n=non-sys s-system a=both
-S[nsa][upg][x]	Segment report nsa as above plus
	u = real memory, p = pinned,
	g = paging, x = top x items
-S sid...	Segment report on particular segments
-i secs count	Repeats report at interval second & count times
-D sid...	Detailed report

Detailed Report Output:

Report column headings	**Description**
size	in pages (4096)
inuse	in-use
free	not inuse included rmss pages
pin	pined (locked by application)
work	pages in working segments
pers	pages in persistent segments
clnt	pages in client segments
pg space	paging space

Note: pages can be in more than one process

Examples:

`svmon -G`	Global / General statistics
`svmon -Pa 215`	Processes report for process 215
`svmon -Ssu 10`	Top ten system segments in real-mem order
`svmon -D 340d`	Detailed report on a particular segment

vmstat - Virtual Memory Management Statistics

The `vmstat` command is used to report statistics about kernel threads in the run and wait queues, memory, paging, disks, interrupts, system calls, context switches, and CPU activity. If the vmstat command is used without any options or only with the interval and optionally, the count parameter, like vmstat 2, then the first line of numbers is an average since system reboot.

Syntax: `vmstat interval count`

Flag	**Meaning**
interval	number of seconds between outputs
count	number of times to output

Report column headings	**Description**
r	# of processes on run queue per second
b	# of processes awaiting paging in per second
avm	active virtual memory pages in paging space
fre	real memory pages on the free list
re	Page reclaims, free but claimed before reused
pi	paged in (per second)
po	paged out (per second)
fr	pages freed (page replacement per second)
sr	pages per second scanned for replacement
cy	complete scans of page table
in	device interrupts per second
sy	system calls per second
cs	CPU context switches per second
us	User CPU time percentage

sys	System CPU time percentage
id	CPU idle percentage (nothing to do)
wa	CPU waiting for pending local Disk I/O

Examples:

`vmstat 10 20`	20 lines output with 10 seconds between each

Special Considerations about vmstat on AIX V4.3.2 and earlier versions

AIX 4.3.3 contains an enhancement to the method used to compute the percentage of CPU time spent waiting on disk I/O (wio time). The method used in AIX 4.3.2 and earlier versions of AIX can give an inflated view of wio time on SMPs in some circumstances. The wio time is reported by the commands sar (%wio), vmstat (wa) and iostat (%iowait).

Method used in AIX 4.3.2 and earlier AIX versions

At each clock interrupt on each processor (100 times a second in AIX), a determination is made as to which of four categories (usr/sys/wio/idle) to place the last 10 ms of time. If the CPU was busy in usr mode at the time of the clock interrupt, then usr gets the clock tick added into its category. If the CPU was busy in kernel mode at the time of the clock interrupt, then the sys category gets the tick. If the CPU was NOT busy, then a check is made to see if ANY I/O to disk is in progress. If any disk I/O is in progress, then the wio category is incremented. If NO disk I/O is in progress and the CPU is not busy, then the idl category gets the tick. The inflated view of wio time results from all idle CPUs being categorized as wio regardless of the number of threads waiting on I/O. For example, RS/6000 with just one thread doing I/O could report over 90 percent wio time regardless of the number of CPUs it has.

Method used in AIX 4.3.3

The change in AIX 4.3.3 is to only mark an idle CPU as wio if an outstanding I/O was started on that CPU. This method can report much lower wio times when just a few threads are doing I/O and the system is otherwise idle. For example, an RS/6000 with four CPUs and one thread doing I/O will report a maximum of 25 percent wio time. An RS/6000 with 12 CPUs and one threadvdoing I/O will report a maximum of 8.3 percent 'wio' time.

vmtune - Virtual Memory Tuning

The `vmtune` command is used to modify the AIX Virtual Memory Manager (VMM) parameters for the purpose of changing the behavior of the memory management subsystem.

Syntax: `vmtune -p min -P max -f min -F max -r min -R max`

Flag	Meaning
-p min	min percentage of memory reserved for file pages (default 20 percent)
-P max	max percentage of memory reserved for file pages (default 80 percent)
-f min	number of pages on free list, below which page stealing starts (default 120)
-F max	number of pages on free list, above which page stealing stops (default 128)
-r min	min number of pages to be read ahead after sequential access is detected
-R max	max number of pages to be read ahead after sequential access is detected