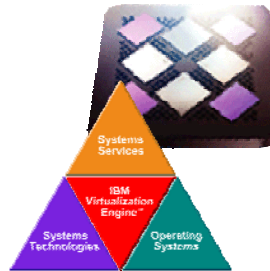


# AIX Performance Tuning



Jaqui Lynch

Senior Systems Engineer

Mainline Information Systems

<http://www.circle4.com/papers/a09perf.pdf>

**Mainline:** solutions you need  
from people you trust

## Agenda

- Filesystem Types
- DIO and CIO
- AIX Performance Tunables
- Oracle Specifics
- Commands
- References

**Mainline:** solutions you need  
from people you trust

## Filesystem Types

- **JFS**

- 2gb file max unless BF
- Can use with DIO
- Optimized for 32 bit
- Runs on 32 bit or 64 bit
- Better for lots of small file creates and deletes

- **GPFS**

Clustered filesystem

Use for RAC

Similar to CIO – noncached, nonblocking I/O

- **JFS2**

- Optimized for 64 bit
- Required for CIO
- Can use DIO
- Allows larger file sizes
- Runs on 32 bit or 64 bit
- Better for large files and filesystems

**Mainline:** solutions you need  
from people you trust

## DIO and CIO

- **DIO**

- Direct I/O
- Around since AIX v5.1
- Used with JFS
- CIO is built on it
- Effectively bypasses filesystem caching to bring data directly into application buffers
- Does not like compressed JFS or BF (lfe) filesystems
  - Performance will suffer due to requirement for 128kb I/O
- Reduces CPU and eliminates overhead copying data twice
- Reads are synchronous
- Bypasses filesystem readahead
- Inode locks still used
- Benefits heavily random access workloads

**Mainline:** solutions you need  
from people you trust

## DIO and CIO

- CIO
  - Concurrent I/O
  - Only available in JFS2
  - Allows performance close to raw devices
  - Use for Oracle dbf and control files, and online redo logs, **not for binaries**
  - Redo logs must be in separate filesystem with 512 blocksize
  - No system buffer caching
  - **Designed for apps (such as RDBs) that enforce write serialization at the app**
  - Allows non-use of inode locks
  - Implies DIO as well
  - Benefits heavy update workloads
  - **Not all apps benefit from CIO and DIO – some are better with filesystem caching and some are safer that way**

Mainline: solutions you need  
from people you trust

## Performance Tuning

- CPU
  - vmstat, ps, nmon
- Network
  - netstat, nfsstat, no, nfso
- I/O
  - iostat, filemon, ioo, lvmo
- Memory
  - lsps, svmon, vmstat, vmo, ioo

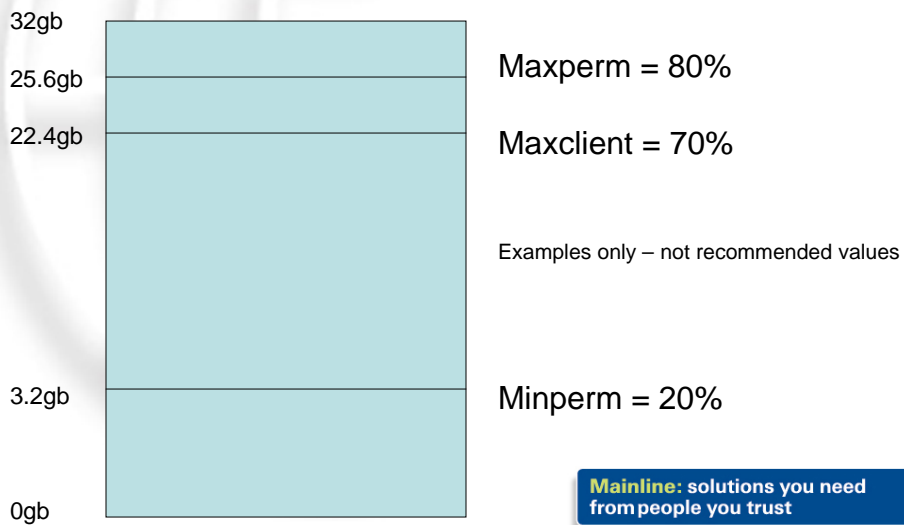
Mainline: solutions you need  
from people you trust

## New tunables Method

- Old way
  - Create rc.tune and add to inittab
- New way
  - /etc/tunables
    - lastboot
    - lastboot.log
    - Nextboot
  - Use `-p -o` options
    - ioo            `-p -o` options
    - vmo           `-p -o` options
    - no             `-p -o` options
    - nfso          `-p -o` options
    - schedo       `-p -o` options

**Mainline:** solutions you need  
from people you trust

## minperm, maxperm, maxclient



**Mainline:** solutions you need  
from people you trust

## 5.3 Tuneables 1/4

- vmo minperm%
  - Value below which we steal from computational pages - default is 20%
  - We lower this to something like 3 or 5%, depending on workload
- vmo maxperm%
  - default is 80%
  - This is a soft limit and affects ALL file pages (including those in maxclient)
  - Value above which we always steal from persistent, affects maxclient
  - We no longer tune this – we use lru\_file\_repage instead
  - Reducing maxperm stops file caching affecting programs that are running
- vmo maxclient
  - default is 80%
  - Must be less than or equal to maxperm, hard limit by default
  - Affects NFS, GPFS and JFS2
  - We no longer tune this – we use lru\_file\_repage instead
- numperm
  - This is what percent of real memory is currently being used for caching ALL file pages
- numclient
  - This is what percent of real memory is currently being used for caching GPFS, JFS2 and NFS

**Mainline:** solutions you need from people you trust

## 5.3 Tuneables 2/4

- vmo strict\_maxperm
    - Set to a soft limit by default – leave as is
  - vmo strict\_maxclient
    - Available at AIX 5.2 ML4
    - By default it is set to a hard limit
    - We used to change to a soft limit – now we do not
- The following two should be done separately to the others
- ioo maxrandwrt
    - Random write behind
    - Default is 0 – try 32
    - Helps flush writes from memory before syncd runs
      - syncd runs every 60 seconds but that can be changed
    - When threshold reached all new page writes are flushed to disk
    - Old pages remain till syncd runs
  - ioo j2\_maxRandomWrite
    - Random write behind for JFS2
    - On a per file basis
    - Default is 0 – try 32
- Use these with care

**Mainline:** solutions you need from people you trust

## 5.3 Tuneables 3/4

- `ioo minpgahead, maxpgahead, J2_minPageReadAhead & J2_maxPageReadAhead`
  - Default min =2 max = 8
  - `maxfree - minfree >= maxpgahead`
- `ioo lvm_bufcnt`
  - Buffers for raw I/O. Default is 9 (9 x 128kb I/Os)
  - Increase if doing large raw I/Os (no jfs)
- `ioo j2_nBufferPerPagerDevice`
  - Minimum filesystem bufstructs for JFS2 – default 512, effective at filesystem mount
- `ioo numfsbufs`
  - Helps write performance for large write sizes
  - Filesystem buffers
- `ioo pv_min_pbuf`
  - Pinned buffers to hold JFS I/O requests
  - Increase if large sequential I/Os to stop I/Os bottlenecking at the LVM
  - One pbuf is used per sequential I/O request regardless of the number of pages
  - With AIX v5.3 each VG gets its own set of pbufs
  - Prior to AIX 5.3 it was a system wide setting

**Mainline:** solutions you need  
from people you trust

## 5.3 Tuneables 4/4

- `vmo mempools`
  - 1 LRUD per pool, default pools is 1 per 8 processors
  - Do not set this parameter – instead use `cpu_scale_memp`
- `vmo cpu_scale_memp`
  - Defaults to 8, Processors per pool
- `vmo minfree and maxfree`
  - Used to set the values between which AIX will steal pages
  - `maxfree` is the number of frames on the free list at which stealing stops (must be  $\geq \text{minfree} + 8$ )
  - `minfree` is the number used to determine when VMM starts stealing pages to replenish the free list
  - On a memory pool basis so if 4 pools and `minfree=1000` then stealing starts at 4000 pages
  - 1 LRUD per pool, default pools is 1 per 8 processors
- `vmo lru_file_repage`
  - Default is 1 – set to 0
  - Available on  $\geq$  AIX v5.2 ML5 and v5.3
  - Means LRUD steals persistent pages unless `numperm < minperm`
  - **HOWEVER if you leave `maxclient` down at 30 then you still limit how much memory filesystems can use**
- `vmo lru_poll_interval`
  - Improves responsiveness of the LRUD when it is running
  - Set to 10 – now the default

**Mainline:** solutions you need  
from people you trust

## minfree/maxfree

- On a memory pool basis so if 4 pools and minfree=1000 then stealing starts at 4000 pages
- 1 LRUD per pool
- Default pools is 1 per 8 processors
- cpu\_scale\_memp can be used to change memory pools (vmo)
- Try to keep distance between minfree and maxfree <=1000
- Obviously this may differ

**Mainline:** solutions you need from people you trust

## vmstat -v

- 26279936 memory pages
  - 25220934 lruable pages
  - 7508669 free pages
  - 4 memory pools
  - 3829840 pinned pages
  - 80.0 maxpin percentage
  - 20.0 minperm percentage
  - 80.0 maxperm percentage
  - 0.3 numperm percentage
  - 89337 file pages
  - 0.0 compressed percentage
  - 0 compressed pages
  - 0.1 numclient percentage
  - 80.0 maxclient percentage
  - 28905 client pages
  - 0 remote pageouts scheduled
  - 280354 pending disk I/Os blocked with no pbuf
  - 0 paging space I/Os blocked with no psbuf
  - 2938 filesystem I/Os blocked with no fsbuf
  - 7911578 client filesystem I/Os blocked with no fsbuf
  - 0 external pager filesystem I/Os blocked with no fsbuf
  - Totals since boot so look at 2 snapshots 60 seconds apart
  - pbufs, psbufs and fsbufs are all pinned
- All filesystem buffers
- Client filesystem buffers only
- LVM – pv\_min\_pbuf  
VMM – fixed per page dev  
numfsbufs - JFS  
NFS & VxFS  
nfs\_v3\_pdots or nfs\_v3\_vm\_bufs or v4 ones  
JFS2 - j2\_nBufferPerPagerDevice  
and/or J2\_dynamicBufferPreallocation

**Mainline:** solutions you need from people you trust

## Starter Set of tunables for 5.3

```
no -p -o rfc1323=1
no -p -o sb_max=1310720
no -p -o tcp_sendspace=262144
no -p -o tcp_recvspace=262144
no -p -o udp_sendspace=65536
no -p -o udp_recvspace=655360
nfs -p -o nfs_rfc1323=1
nfs -p -o nfs_socketsize=600000
nfs -p -o nfs_tcp_socketsize=600000

vmo -p -o minperm%=5
vmo -p -o minfree=960
vmo -p -o maxfree=1088
vmo -p -o lru_file_repage=0
vmo -p -o lru_poll_interval=10

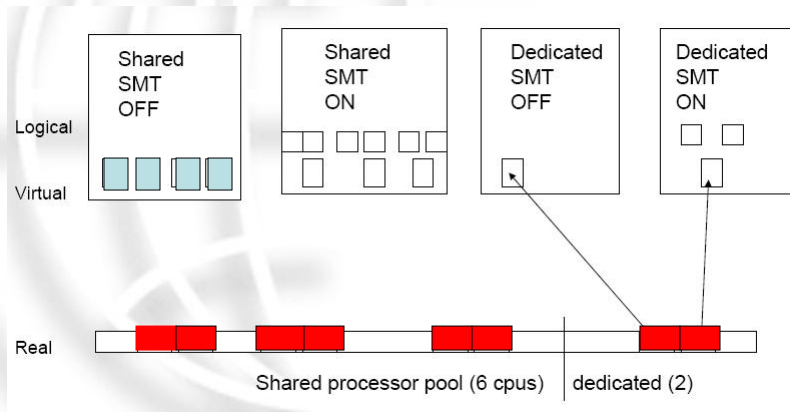
ioo -p -o j2_maxPageReadAhead=128
ioo -p -o maxpgahead=16
ioo -p -o j2_nBufferPerPagerDevice=1024
ioo -p -o pv_min_pbuf=1024
ioo -p -o numfsbufs=2048
ioo -p -o j2_nPagesPerWriteBehindCluster=32
```

NB please test these before putting into production

Increase the following if using raw LVMs (default is 9)  
ioo -p -o lvm\_bufvnt=12

**Mainline: solutions you need from people you trust**

## Logical, Virtual or Real?



In shared world there is no one to one relationship between real and virtual processors  
The dispatch unit becomes the VP

**Mainline: solutions you need from people you trust**

# vmstat -l

**IGNORE FIRST LINE - average since boot**  
Run vmstat over an interval (i.e. vmstat 2 30)

vmstat -l output

System configuration: lcpu=8 mem=1024MB ent=0.50

kthr	memory	page	faults	cpu																
r	b	p	avm	fre	fi	fo	pi	po	fr	sr	in	sy	cs	us	sy	id	wa	pc	ec	
1	1	0	170334	968	96	163	0	0	190	511	11	556	662	1	4	90	5	0.03	6.8	
1	1	0	170334	1013	53	85	0	0	107	216	7	268	418	0	2	92	5	0.02	4.4	

fi/fo is filesystem paging

pi/po is real paging

pc = physical processor units consumed – if using SPP

ec = %entitled capacity consumed – if using SPP

Fre may well be between minfree and maxfree

fr:sr ratio 1783:2949 means that for every 1783 pages freed 2949 pages had to be examined.

ROT was 1:4 – may need adjusting

To get a 60 second average try: vmstat 60 2

Need to know if shared & SMT to figure out VPs, etc

**Mainline:** solutions you need  
from people you trust

## Memory and I/O problems

- iostat
  - Look for overloaded disks and adapters
- vmstat
- vmo and ioo (replace vmtune)
- sar
- Check placement of JFS and JFS2 filesystems and potentially the logs
- Check placement of Oracle or database logs
- fileplace and filemon
- Asynchronous I/O
- Paging
- svmon
  - svmon -G >filename
- nmon
- Check error logs

**Mainline:** solutions you need  
from people you trust

# iostat

**IGNORE FIRST LINE - average since boot**  
**Run iostat over an interval (i.e. iostat 2 30)**

```
tty:  tin      tout  avg-cpu: % user % sys % idle % iowait physc % entc
      0.0    1406.0          93.1  6.9  0.0    0.0  12.0  100.0

Disks:      % tm_act      Kbps    tps      Kb_read  Kb_wrtn
hdisk1      1.0              1.5     3.0      0         3
hdisk0      6.5             385.5   19.5     0         771
hdisk14     40.5            13004.0 3098.5   12744    13264
hdisk7      21.0            6926.0  271.0    440      13412
hdisk15     50.5            14486.0 3441.5   13936    15036
hdisk17     0.0              0.0     0.0      0         0
```

**Mainline: solutions you need  
from people you trust**

# iostat -a Adapters

**System configuration: lcpu=16 drives=15**

```
tty:  tin      tout  avg-cpu: % user % sys % idle % iowait
      0.4    195.3          21.4  3.3  64.7  10.6
```

```
Adapter:      Kbps    tps  Kb_read  Kb_wrtn
fscsi1      5048.8  516.9 1044720428 167866596
```

```
Disks:      % tm_act  Kbps    tps  Kb_read  Kb_wrtn
hdisk6      23.4    1846.1  195.2 381485286 61892408
hdisk9      13.9    1695.9  163.3 373163554 34143700
hdisk8      14.4    1373.3  144.6 283786186 46044360
hdisk7      1.1     133.5   13.8  6285402  25786128
```

```
Adapter:      Kbps    tps  Kb_read  Kb_wrtn
fscsi0      4438.6  467.6 980384452 85642468
```

```
Disks:      % tm_act  Kbps    tps  Kb_read  Kb_wrtn
hdisk5      15.2    1387.4  143.8 304880506 28324064
hdisk2      15.5    1364.4  148.1 302734898 24950680
hdisk3      0.5     81.4    6.8   3515294  16043840
hdisk4      15.8    1605.4  168.8 369253754 16323884
```

**Mainline: solutions you need  
from people you trust**

# iostat -D

## Extended Drive Report

Also check out the -aD option

```

hdisk3   xfer: %tm_act  bps  tps  bread  bwrtn
          0.5  29.7K  6.8   15.0K  14.8K
read:    rps  avgserv  minserv  maxserv  timeouts  fails
          29.3  0.1   0.1    784.5   0          0
write:   wps  avgserv  minserv  maxserv  timeouts  fails
          133.6  0.0   0.3    2.1S   0          0
wait:    avgtime  mintage  maxtime  avgqsz   sqfull
          0.0   0.0     0.2     0.0     0
    
```

tps Transactions per second – transfers per second to the adapter  
 avgserv Average service time  
 Avgtime Average time in the wait queue  
 avgwqsz Average wait queue size  
 If regularly >0 increase queue-depth  
 avgsqsz Average service queue size (waiting to be sent to disk)  
 Can't be larger than queue-depth for the disk  
 sqfull Number times the service queue was full  
 Look at iostat -aD for adapter queues  
 If avgwqsz > 0 or sqfull high then increase queue\_depth  
 Per IBM Average IO sizes:

read = bread/rps  
 write = bwrtn/wps

Mainline: solutions you need from people you trust

# iostat Other

## iostat -A async IO

System configuration: lcpu=16 drives=15

```

aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait
      150  0  5652  0 12288          21.4  3.3  64.7  10.6
    
```

```

Disks:  % tm_act  Kbps  tps  Kb_read  Kb_wrtn
hdisk6  23.4  1846.1  195.2  381485298  61892856
hdisk5  15.2  1387.4  143.8  304880506  28324064
hdisk9  13.9  1695.9  163.3  373163558  34144512
    
```

If maxg close to maxr or maxservers then increase maxreqs or maxservers  
 (from A06 Grover's presentatin

### iostat -m paths

```

System configuration: lcpu=16 drives=15
tty:  tin  tout  avg-cpu: % user % sys % idle % iowait
      0.4  196.3          21.4  3.3  64.7  10.6

Disks:  % tm_act  Kbps  tps  Kb_read  Kb_wrtn
hdisk0  1.6  17.0  3.7  1190873  2893501

Paths:  % tm_act  Kbps  tps  Kb_read  Kb_wrtn
Path0   1.6  17.0  3.7  1190873  2893501
    
```

Mainline: solutions you need from people you trust

## lvmo

- lvmo output
- 
- **vgname** = rootvg (default but you can change with -v)
- **pv\_pbuf\_count** = 256
  - Pbufs to add when a new disk is added to this VG
- **total\_vg\_pbufs** = 512
  - Current total number of pbufs available for the volume group.
- **max\_vg\_pbuf\_count** = 8192
  - Max pbufs that can be allocated to this VG
- **pervg\_blocked\_io\_count** = 0
  - No. I/O's blocked due to lack of free pbufs for this VG
- **global\_pbuf\_count** = 512
  - Minimum pbufs to add when a new disk is added to a VG
- **global\_blocked\_io\_count** = 46
  - No. I/O's blocked due to lack of free pbufs for all VGs
- If global blocked IOs then run lvmo against the other volume groups

**Mainline:** solutions you need  
from people you trust

## lsps output

```
lsps -a
Page Space Physical Volume Volume Group Size %Used Active Auto Type
paging00    hdisk5          vgpaging    2048MB    1   yes   yes   lv
hd6         hdisk0          rootvg      2048MB    1   yes   yes   lv
```

```
lsps -s
Total Paging Space Percent Used
4096MB             1%
```

Should be balanced  
Make hd6 the same size as the others in a mixed environment like this

**Best practice**  
More than one page volume  
All the same size including hd6

**Mainline:** solutions you need  
from people you trust

## SVMON Terminology

- *persistent*
  - Segments used to manipulate files and directories
- *working*
  - Segments used to implement the data areas of processes and shared memory segments
- *client*
  - Segments used to implement some virtual file systems like Network File System (NFS) and the CD-ROM file system
- <http://publib.boulder.ibm.com/infocenter/pseries/topic/com.ibm.aix.doc/cmds/aixcmds5/svmon.htm>

**Mainline:** solutions you need  
from people you trust

## svmon -G

	size	inuse	free	pin	virtual
memory	26279936	18778708	7501792	3830899	18669057
pg space	7995392	53026			

	work	pers	clnt	lpage
pin	3830890	0	0	0
in use	18669611	80204	28893	0

In GB Equates to:

	size	inuse	free	pin	virtual
memory	100.25	71.64	28.62	14.61	71.22
pg space	30.50	0.20			

	work	pers	clnt	lpage
pin	14.61	0	0	0
in use	71.22	0.31	0.15	0

**Mainline:** solutions you need  
from people you trust

## svmon -G (page sizes)

	size	inuse	free	pin	virtual
memory	33554432	3332994	30221438	2620698	3265499
pg space	8388608	9132			
	work	pers	clnt		
pin	2620698	0	0		
in use	3265506	0	67488		
PageSize	PoolSize	inuse	pgsp	pin	virtual
s 4 KB	-	3150482	9132	2494058	3082987
m 64 KB	-	11407	0	7915	11407

**Mainline:** solutions you need  
from people you trust

## General Recommendations

- Different hot LVs on separate physical volumes
- Stripe hot LV across disks to parallelize
- Mirror read intensive data
- Ensure LVs are contiguous
  - Use lslv and look at in-band % and distrib
  - reorgvg if needed to reorg LVs
- minpgahead=2, maxpgahead=16 for 64kb stripe size
- Increase maxfree if you adjust maxpgahead
- Tweak minperm, maxperm and maxrandwrt
- Tweak lvm\_bufcnt if doing a lot of large raw I/Os
- If JFS2 tweak j2 versions of above fields
- Clean out inittab and rc.tcpip and inetd.conf, etc for things that should not start
  - Make sure you don't do it partially
  - i.e. portmap is in rc.tcpip and rc.nfs

**Mainline:** solutions you need  
from people you trust

## Oracle Specifics

- Use JFS2 with external JFS2 logs
  - (if high write otherwise internal logs are fine)
  - External JFS2 logs allow you to mirror the log to another disk live if hot spot
  - Give the log more than 1 PP
  - Do not share logs between busy filesystems
- Use CIO where it will benefit you
  - Do not use for Oracle binaries
  - Ensure redo logs are in their own filesystem with the correct (512) blocksize
  - I give each instance its own filesystem and their redo logs are also separate
- Leave DISK\_ASYNC\_IO=TRUE in Oracle
- Tweak the maxservers AIO settings
  
- If using JFS
  - Do not allocate JFS with BF (LFE)
  - It increases DIO transfer size from 4k to 128k
  - 2gb is largest file size
  - Do not use compressed JFS – defeats DIO

**Mainline:** solutions you need  
from people you trust

## Tools

- vmstat – for processor and memory
- nmon
  - <http://www-941.ibm.com/collaboration/wiki/display/WikiPtype/nmon>
  - To get a 2 hour snapshot (240 x 30 seconds)
  - `nmon -fT -c 240 -s 30`
  - Creates a file in the directory that ends .nmon
- nmon analyzer
  - <http://www-941.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmonanalyser>
  - Windows tool so need to copy the .nmon file over
  - Opens as an excel spreadsheet and then analyses the data
  - Also look at nmon consolidator
- sar
  - `sar -A -o filename 2 30 >/dev/null`
  - Creates a snapshot to a file – in this case 30 snaps 2 seconds apart
- ioo, vmo, schedo, vmstat -v
- lvmo
- lparstat, mpstat
- lostat
- Check out Alphaworks for the Graphical LPAR tool
- Ganglia
  - <http://ganglia.info>
- Many many more

**Mainline:** solutions you need  
from people you trust

## Other tools

- filemon
  - filemon -v -o filename -O all
  - sleep 30
  - trcstop
- pstat to check async I/O
  - pstat -a | grep aio | wc -l
- perfpmr to build performance info for IBM if reporting a PMR
  - /usr/bin/perfpmr.sh 300

Mainline: solutions you need  
from people you trust

## filemon Logical Volume Stats

-----  
Detailed Logical Volume Stats (512 byte blocks)  
-----

VOLUME: /dev/d10 description: /llocal  
reads: 18463 (0 errs)  
read sizes (blks): avg 16.0 min 8 max 16 sdev 0.1  
read times (msec): avg 36.531 min 0.175 max 12618.067 sdev 223.008  
read sequences: 18434  
read seq. lengths: avg 16.0 min 8 max 80 sdev 0.8  
  
writes: 186 (0 errs)  
write sizes (blks): avg 29.1 min 8 max 256 sdev 31.4  
write times (msec): avg 1452.762 min 0.355 max 12755.286 sdev 3230.891  
write sequences: 186  
write seq. lengths: avg 29.1 min 8 max 256 sdev 31.4  
  
seeks: 18620 (99.8%)  
seek dist (blks): init 189844552,  
avg 73513927.0 min 16 max 670226672 sdev 145561022.8  
time to next req(msec): avg 1.609 min 0.000 max 74.103 sdev 5.353  
throughput: 5011.0 KB/sec  
utilization: 1.00

Avg read times should be <20 msecs  
Avg write times if cache should be < 2 msecs

Mainline: solutions you need  
from people you trust

# filemon Physical Volume Stats

Detailed Physical Volume Stats (512 byte blocks)

Note block size

VOLUME: /dev/hdisk115 description: IBM FC 1750  
 reads: 5639 (0 errs)  
 read sizes (blks): avg 16.0 min 16 max 16 sdev 0.0  
 read times (msec): avg 40.022 min 0.169 max 9065.663 sdev 246.877  
 read sequences: 5632  
 read seq. lengths: avg 16.0 min 16 max 32 sdev 0.6  
 writes: 70 (0 errs)  
 write sizes (blks): avg 33.8 min 16 max 160 sdev 30.5  
 write times (msec): avg 1604.654 min 0.366 max 12617.016 sdev 3654.443  
 write sequences: 70  
 write seq. lengths: avg 33.8 min 16 max 160 sdev 30.5  
 seeks: 5702 (99.9%)  
 seek dist (blks): init 51567944,  
 avg 24166627.0 min 16 max 68424352 sdev 22062225.9  
 seek dist (%tot blks):init 74.51367,  
 avg 34.91984 min 0.00002 max 98.87053 sdev 31.87906  
 time to next req(msec): avg 5.258 min 0.001 max 183.632 sdev 10.657  
 throughput: 1542.5 KB/sec  
 utilization: 1.00

Avg read size x 512 provides average IO size for reads  
 Avg write size x 512 provides average IO size for writes

**Mainline: solutions you need  
 from people you trust**

# lparstat -h

## lparstat -h 30 2 (Busy database)

System configuration: type=Dedicated mode=Capped smt=On lcpu=8 mem=16384

%user	%sys	%wait	%idle	%hypv	hcalls
67.9	4.6	0.4	27.1	11.8	2767450
76.7	4.6	0.3	18.4	9.8	1858344

## lparstat -h 30 2 output

System configuration: type=Shared mode=Uncapped smt=On lcpu=4 mem=16384 psize=16 ent=2.00

%user	%sys	%wait	%idle	physc	%entc	lbusy	vcswh	phint	%hypv	hcalls
2.4	1.8	1.2	94.6	0.10	4.8	2.3	700	2	3.2	4463
1.3	1.2	0.4	97.1	0.06	2.9	1.2	659	2	1.8	2795

Physc physical processors consumed  
 If capped then will not exceed entitled capacity  
 For uncapped it can match up to the number of processors in the pool depending on on-line VPs  
 %entc percent of entitled capacity  
 Will not exceed 100% if capped  
 Lbusy logical processor utilization for system and user  
 If this gets close to 100% you may bneed more VPs  
 Vcswh Virtual context switches  
 Phint phantom interrupts (Interrupts that belong to another shared partition)  
 %hypv %time in the hypervisor for this lpar – weird numbers on an idle system may be seen  
 Hcalls Number of hypervisor calls

<http://publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.aix.doc/cmts/aixcmds3/lparstat.htm>

**Mainline: solutions you need  
 from people you trust**

# lparstat -i

lparstat -i output	DEDICATED-SMT	lparstat -i output	SHARED-SMT
Node Name	: abcdef	Node Name	: xyw
Partition Name	: lpar abcdef	Partition Name	: lpar xyw
Partition Number	: 3	Partition Number	: 7
Type	: Dedicated-SMT	Type	: Shared-SMT
Mode	: Capped	Mode	: Uncapped
Entitled Capacity	: 4.00	Entitled Capacity	: 2.00
Partition Group-ID	: 32771	Partition Group-ID	: 32775
Shared Pool ID	: -	Shared Pool ID	: 0
Online Virtual CPUs	: 4	Online Virtual CPUs	: 2
Maximum Virtual CPUs	: 16	Maximum Virtual CPUs	: 8
Minimum Virtual CPUs	: 1	Minimum Virtual CPUs	: 1
Online Memory	: 16384 MB	Online Memory	: 16384 MB
Maximum Memory	: 65536 MB	Maximum Memory	: 24576 MB
Minimum Memory	: 512 MB	Minimum Memory	: 8192 MB
Variable Capacity Weight	: -	Variable Capacity Weight	: 128
Minimum Capacity	: 1.00	Minimum Capacity	: 0.20
Maximum Capacity	: 16.00	Maximum Capacity	: 8.00
Capacity Increment	: 1.00	Capacity Increment	: 0.01
Maximum Physical CPUs in system	: 16	Maximum Physical CPUs in sys	: 16
Active Physical CPUs in system	: 16	Active Physical CPUs in sys	: 16
Active CPUs in Pool	: -	Active CPUs in Pool	: 16
Unallocated Capacity	: -	Unallocated Capacity	: 0.00
Physical CPU Percentage	: 100.00%	Physical CPU Percentage	: 100.00%

**Mainline: solutions you need  
from people you trust**

# mpstat

mpstat -s shows how processor is distributed using SMT

System configuration: lcpu=4 ent=0.5

Proc1		Proc0		Physical or Virtual
cpu0	cpu2	cpu1	cpu3	Logical or relative SMT split
0.17%	0.10%	3.14%	46.49%	

System configuration: lcpu=8

Proc0		Proc2		Proc4		Proc6	
cpu0	cpu1	cpu2	cpu3	cpu4	cpu5	cpu6	cpu7
53.30%	46.70%	52.91%	47.09%	52.83%	47.21%	52.66%	47.28%

---

Proc0		Proc2		Proc4		Proc6	
cpu0	cpu1	cpu2	cpu3	cpu4	cpu5	cpu6	cpu7
53.16%	46.73%	52.88%	47.06%	52.77%	47.23%	0.00%	0.00%

**Mainline: solutions you need  
from people you trust**

## mpstat 2 2

# mpstat 2 2

System configuration: lcpu=4 ent=0.5 mode=Uncapped

cpu	min	maj	mpc	int	cs	ics	rq	mig	lpa	syc	us	sy	wa	id	pc	%ec	lcs
0	8	0	0	168	168	72	0	0	100	145	18	63	0	19	0.00	1.0	127
1	0	0	0	11	0	0	0	0	-	0	0	3	0	97	0.00	0.3	127
2	0	0	0	10	0	0	0	0	-	0	0	25	0	75	0.00	0.0	20
3	0	0	0	10	0	0	0	0	-	0	0	24	0	76	0.00	0.0	20
U	-	-	-	-	-	-	-	-	-	-	-	-	0	99	0.49	98.7	-
ALL	8	0	0	199	168	72	0	0	100	145	0	1	0	99	0.01	1.3	294

MIN minor page fault – no disk I/O  
MAJ major page fault – real disk I/O  
RQ total processes on the runQ  
PC fraction of physical processor consumed (if shared mode and/or SMT)  
%EC Percentage of entitled capacity consumed

US,SY,WA & ID should be interpreted relative to PC

**Mainline: solutions you need  
from people you trust**

## Async I/O

### Total number of AIOs in use

pstat -a | grep aios | wc -l  
Maximum AIOservers started since boot

Or new way for Posix AIOs is:

ps -k | grep aio | wc -l  
4205

### AIO max possible requests

lsattr -El aio0 -a maxreqs  
maxreqs 4096 Maximum number of REQUESTS True

### AIO maxservers

lsattr -El aio0 -a maxservers  
maxservers 320 MAXIMUM number of servers per cpu True

NB – maxservers is a per processor setting in AIX 5.3

Look at using fastpath  
Fastpath can now be enabled with DIO/CIO

Also iostat -A  
THIS ALL CHANGES IN AIX V6 – SETTINGS WILL BE UNDER IOO THERE

**Mainline: solutions you need  
from people you trust**

## netstat

- netstat -i
  - Shows input and output packets and errors for each adapter
  - Also shows collisions
- netstat -ss
  - Shows summary info such as udp packets dropped due to no socket
- netstat -m
  - Memory information
  - Look for failed calls
- netstat -v
  - Statistical information on all adapters

Mainline: solutions you need from people you trust

## netstat -s

Some fields to look at

udp:

1087 datagrams received  
0 bad checksums  
76 dropped due to no socket  
122 broadcast/multicast datagrams dropped due to no socket  
0 socket buffer overflows  
889 delivered  
960 datagrams output

ip:

0 output packets dropped due to no bufs, etc.  
0 IP Multicast packets dropped due to no receiver  
0 ipintrq overflows  
0 packets dropped due to the full socket receive buffer

tcp:

0 discarded due to listener's queue full  
0 packets dropped due to memory allocation failure

Mainline: solutions you need from people you trust

## AIX v6 Changes

- Some parameters on vmo, ioo become restricted and have new defaults
- You will need to use a -F on the command to even see them
- Some examples include:
  - vmo
    - cpu\_scale\_memp=8
    - lru\_file\_repage=0
    - lru\_poll\_interval=10
    - maxclient% and maxperm% both now 90%
  - ioo
    - j2\_nBufferPerPagerDevice=512
    - numfsbufs=196
    - pv\_num\_pbufs=512
    - aio (posix and other) settings are now here

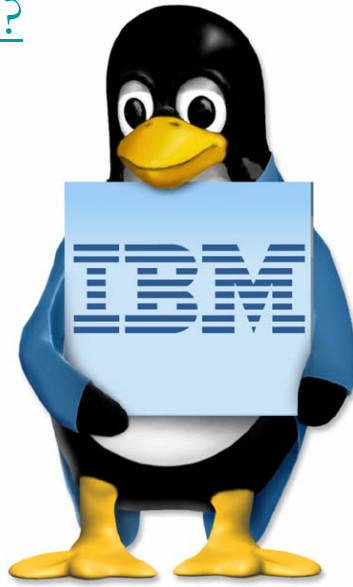
**Mainline:** solutions you need  
from people you trust

## Useful Links

- **1. Ganglia**
  - [ganglia.info](http://ganglia.info)
- **2. Lparmon**
  - [www.alphaworks.ibm.com/tech/lparmon](http://www.alphaworks.ibm.com/tech/lparmon)
- **3. Nmon**
  - [www.ibm.com/collaboration/wiki/display/WikiPtype/nmon](http://www.ibm.com/collaboration/wiki/display/WikiPtype/nmon)
- **4. Nmon Analyser**
  - [www.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmonanalyser](http://www.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmonanalyser)
- **5. Jaqui's AIX\* Blog**
  - Has a base set of performance tunables for AIX 5.3 - [www.circle4.com/blosxomj1.cgi/](http://www.circle4.com/blosxomj1.cgi/)
- **6. vmo command**
  - [publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds6/vmo.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds6/vmo.htm)
- **7. ioo command**
  - [publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/ioo.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/ioo.htm)
- **8. vmstat command**
  - [publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/ioo.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/ioo.htm)
- **9. Ivmo command**
  - [publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/ioo.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/ioo.htm)
- **10. eServer Magazine and AiXtra**
  - <http://www.eservercomputing.com/>
    - Search on Jaqui AND Lynch
    - Articles on Tuning and Virtualization
- **11. Find more on Mainline at:**
  - <http://www.mainline.com>

**Mainline:** solutions you need  
from people you trust

Questions?



**Mainline:** solutions you need  
from people you trust