

IBM TRAINING



## A26

### AIX Performance Tuning

Jaqui Lynch

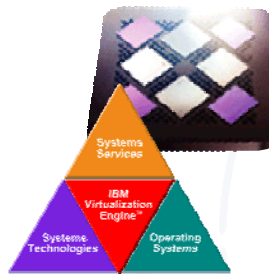
**IBM SYSTEM p, AIX 5L  
and LINUX TECHNICAL  
UNIVERSITY**  
**Sept 11 - 15, 2006**

Las Vegas, NV

# AIX Performance Tuning

Updated Presentation will be at:  
<http://www.circle4.com/papers/pseries-a26-aug06.pdf>

**Jaqui Lynch**  
**Senior Systems Engineer**  
**Mainline Information Systems**



## Agenda

- AIX v5.2 versus AIX v5.3
- 32 bit versus 64 bit
- Filesystem Types
- DIO and CIO
- AIX Performance Tunables
- Oracle Specifics
- Commands
- References



## New in AIX 5.2

- P5 support
- JFS2
- Large Page support (16mb)
- Dynamic LPAR
- Small Memory Mode
  - Better granularity in assignment of memory to LPARs
- CuOD
- xProfiler
- New Performance commands
  - vmo, ioo, schedo replace schedtune and vmtune
- AIX 5.1 Status
  - Will not run on p5 hardware
  - Withdrawn from marketing end April 2005
  - Support withdrawn April 2006



## AIX 5.3

- New in 5.3
  - With Power5 hardware
    - SMT
    - Virtual Ethernet
  - With APV
    - Shared Ethernet
    - Virtual SCSI Adapter
    - Micropartitioning
    - PLM



## AIX 5.3

- New in 5.3
  - JFS2 Updates
    - Improved journaling
    - Extent based allocation
    - 1tb filesystems and files with potential of 4PB
    - Advanced Accounting
    - Filesystem shrink for JFS2
    - Striped Columns
      - Can extend striped LV if a disk fills up
    - 1024 disk scalable volume group
      - 1024 PVs, 4096 LVs, 2M pps/vg
    - Quotas
    - Each VG now has its own tunable pbuf pool
      - Use lvmo command



## AIX 5.3

- New in 5.3
  - NFSv4 Changes
    - ACLs
  - NIM enhancements
    - Security
    - Highly available NIM
    - Post install configuration of Etherchannel and Virtual IP
  - SUMA patch tool
  - Last version to support 32 bit kernel
  - MP kernel even on a UP
  - Most commands changed to support LPAR stats
  - Forced move from vmtune to ioo and vmo
  - Page space scrubbing
  - Plus lots and lots of other things



## 32 bit versus 64 bit

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• 32 Bit</li></ul>  | <ul style="list-style-type: none"><li>• 64 bit</li></ul>   |
| <ul style="list-style-type: none"><li>• Up to 96GB memory</li><li>• Uses JFS for rootvg</li><li>• Runs on 32 or 64 bit hardware</li><li>• Hardware all defaults to 32 bit</li><li>• JFS is optimized for 32 bit</li><li>• 5.3 is last version of AIX with 32 bit kernel</li></ul> | <ul style="list-style-type: none"><li>• Allows &gt; 96GB memory</li><li>• Current max is 256GB (arch is 16TB) except 590/595 (1TB &amp; 2TB)</li><li>• Uses JFS2 for rootvg</li><li>• Supports 32 and 64 bit apps</li><li>• JFS2 is optimized for 64 bit</li></ul> |



## 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

- **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

- **GPFS**

Clustered filesystem  
Use for RAC  
Similar to CIO – noncached, nonblocking I/O



## 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



## 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**
  - 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**



## Performance Tuning

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



## New tunables

- 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



## Tuneables 1/3

- minperm%
  - Value below which we steal from computational pages - default is 20%
  - We lower this to something like 5%, depending on workload
- 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
  - Be careful as this also affects maxclient
  - We no longer tune this – we use lru\_file\_repage instead
  - Reducing maxperm stops file caching affecting programs that are running
- maxclient
  - default is 80%
  - Must be less than or equal to maxperm
  - Affects NFS, GPFS and JFS2
  - Hard limit by default
  - 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
- strict\_maxperm
  - Set to a soft limit by default – leave as is
- 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



## Tuneables 2/3

- 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
- Numclust
  - Sequential write behind
  - Number of 16k clusters processed by write behind
- J2\_maxRandomWrite
  - Random write behind for JFS2
  - On a per file basis
  - Default is 0 – try 32
- J2\_nPagesPerWriteBehindCluster
  - Default is 32
  - Number of pages per cluster for writebehind
- J2\_nRandomCluster
  - JFS2 sequential write behind
  - Distance apart before random is detected
- J2\_nBufferPerPagerDevice
  - Minimum filesystem bufstructs for JFS2 – default 512, effective at fs mount



## Tuneables 3/3

- minpgahead, maxpgahead, J2\_minPageReadAhead & J2\_maxPageReadAhead
  - Default min = 2 max = 8
  - Maxfree – minfree >= maxpgahead
- lvm\_bufcmt
  - Buffers for raw I/O. Default is 9
  - Increase if doing large raw I/Os (no jfs)
- numfsbufs
  - Helps write performance for large write sizes
  - Filesystem buffers
- 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
- sync\_release\_ilock
  - Allow sync to flush all I/O to a file without holding the i-node lock, and then use the i-node lock to do the commit.
  - Be very careful – this is an advanced parameter
- 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 >= 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
- lru\_file\_repage
  - Default is 1 – set to 0
  - Available on >= AIX v5.2 ML5 and v5.3
  - Means LRUD steals persistent pages unless numperm < minperm
- lru\_poll\_interval
  - Set to 10
  - Improves responsiveness of the LRUD when it is running



NEW

# 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
- Try to keep distance between minfree and maxfree  $\leq 1000$
- Obviously this may differ



# 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
- j2\_nBufferPerPagerDevice



## Starter Set of tunables

NB please test these before putting into production

```
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=60000
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_maxRandomWrite=32
ioo -p -o maxrandwrt=32
ioo -p -o j2_nBufferPerPagerDevice=1024
ioo -p -o pv_min_pbuf=1024
ioo -p -o numfsbufs=2048
ioo -p -o j2_nPagesPerWriteBehindCluster=32
```

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



## vmstat -l

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

System configuration: lcpu=24 mem=102656MB ent=0

kthr	memory	page	faults	cpu
r	b	avm	fre	re pi po fr sr cy in sy cs us sy id wa pc ec
56	1	18637043	7533530	0 0 0 0 0 0 0 4298 24564 9866 98 2 0 0 12.00 100.0
57	1	18643753	7526811	0 0 0 0 0 0 0 3867 25124 9130 98 2 0 0 12.00 100.0

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	

Pc = physical processors 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



## 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



## ioo Output

- lvm\_bufcnt = 9
- minpgahead = 2
- maxpgahead = 8
- maxrandwrt = 32 (default is 0)
- numclust = 1
- numfsbufs = 186
- sync\_release\_ilock = 0
- pd\_npages = 65536
- pv\_min\_pbuf = 512
  
- j2\_minPageReadAhead = 2
- j2\_maxPageReadAhead = 8
- j2\_nBufferPerPagerDevice = 512
- j2\_nPagesPerWriteBehindCluster = 32
- j2\_maxRandomWrite = 0
- j2\_nRandomCluster = 0



# vmo Output

## DEFAULTS

maxfree = 128  
minfree = 120  
minperm% = 20  
maxperm% = 80  
maxpin% = 80  
maxclient% = 80  
strict\_maxclient = 1  
strict\_maxperm = 0

## OFTEN SEEN

maxfree = 1088  
minfree = 960  
minperm% = 10  
maxperm% = 30  
maxpin% = 80  
Maxclient% = 30  
strict\_maxclient = 0  
strict\_maxperm = 0

numclient and numperm are both 29.9

So numclient-numperm=0 above

Means filecaching use is probably all JFS2/NFS/GPFS

Remember to switch to new method using `lru_file_repage`



# 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



## 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
```

## iostat -D

### Extended Drive Report

```
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:    avctime  mintime  maxtime  avgqsz  qfull
          0.0  0.0  0.2  0.0  0
```



# iostat Other

## iostat -A async IO

System configuration: lcpu=16 drives=15

```
aiostat: 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

## iostat -m paths

System configuration: lcpu=16 drives=15

```
ttystat: tin tout avg-cpu: % user % sys % idle % iowait
0.4 195.3 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



# 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



# lsps -a (similar to pstat)

- Ensure all page datasets the same size although hd6 can be bigger - ensure more page space than memory
  - Especially if not all page datasets are in rootvg
  - Rootvg page datasets must be big enough to hold the kernel
- Only includes pages allocated (default)
- Use lsps -s to get all pages (includes reserved via early allocation (PSALLOC=early))
- Use multiple page datasets on multiple disks
  - Parallelism



# lsps output

```
lsps -a
```

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
paging05	hdisk9	pagvg01	2072MB	1	yes	yes	lv
paging04	hdisk5	vgpaging01	504MB	1	yes	yes	lv
paging02	hdisk4	vgpaging02	168MB	1	yes	yes	lv
paging01	hdisk3	vgpagine03	168MB	1	yes	yes	lv
paging00	hdisk2	vgpaging04	168MB	1	yes	yes	lv
hd6	hdisk0	rootvg	512MB	1	yes	yes	lv

```
lsps -s
```

Total Paging Space	Percent Used
3592MB	1%

Bad Layout above  
Should be balanced  
Make hd6 the biggest by one lp or the same size as the others in a mixed environment like this



# 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/topi/c/com.ibm.aix.doc/cmds/aixcmds5/svmon.htm>



# 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



# 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 lsv and look at in-band % and distrib
  - reorgvg if needed to reorg LVs
- Writeverify=no
- 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



# Oracle Specifics

- Use JFS2 with external JFS2 logs (if high write otherwise internal logs are fine)
- Use CIO where it will benefit you
  - Do not use for Oracle binaries
- 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



## 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 30 -s 240`
  - Creates a file in the directory that ends .nmon
- nmon analyzer
  - <http://www-941.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmonanalyzer>
  - Windows tool so need to copy the .nmon file over
  - Opens as an excel spreadsheet and then analyses the data
- 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
- Many many more



## 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`



# lparstat

lparstat -h

System Configuration: type=shared mode=Uncapped smt=On lcpu=4 mem=512 ent=5.0

%user	%sys	%wait	%idle	phisc	%entc	lbusy	app	vcs	phint	%hypv	hcalls
0.0	0.5	0.0	99.5	0.00	1.0	0.0	-	1524	0	0.5	1542
16.0	76.3	0.0	7.7	0.30	100.0	90.5	-	321	1	0.9	259

Phisc – physical processors consumed

%entc – percent of entitled capacity

Lbusy – logical processor utilization for system and user

Vcs – Virtual context switches

Phint – phantom interrupts to other partitions

%hypv - %time in the hypervisor for this lpar – weird numbers on an idle system may be seen

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



# mpstat

mpstat -s

System configuration: lcpu=4 ent=0.5

	Proc1	Proc0		
	0.27%	49.63%		
cpu0	cpu2	cpu1	cpu3	
0.17%	0.10%	3.14%	46.49%	

Above shows how processor is distributed using SMT



# Async I/O

## Total number of AIOs in use

```
pstat -a | grep aios | wc -l  
Or new way 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

See Session A23 by Grover Davidson for a lot more info on Async I/O



# I/O Pacing

- Useful to turn on during backups (streaming I/Os)
- Set high value to multiple of  $(4*n)+1$
- Limits the number of outstanding I/Os against an individual file
- minpout – minimum
- maxpout – maximum
- If process reaches maxpout then it is suspended from creating I/O until outstanding requests reach minpout



# Network

- no -a & nfs -a to find what values are set to now
- Buffers
  - Mbufs
    - Network kernel buffers
    - thewall is max memory for mbufs
    - Can use maxmbuf tuneable to limit this or increase it
      - Uses chdev
      - Determines real memory used by communications
      - If 0 (default) then thewall is used
      - Leave it alone
  - TCP and UDP receive and send buffers
  - Ethernet adapter attributes
    - If change send and receive above then also set it here
  - no and nfs commands
  - nfsstat
  - rfc1323 and nfs\_rfc1323



# 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
- netstat -v
  - Statistical information on all adapters



## Network tuneables

- no -a
- Using no
  - rfc1323 = 1
  - sb\_max=1310720 ( $\geq 1\text{MB}$ )
  - tcp\_sendspace=262144
  - tcp\_recvspace=262144
  - udp\_sendspace=65536 (at a minimum)
  - udp\_recvspace=65536
    - Must be less than sb\_max
- Using nfso
  - nfso -a
  - nfs\_rfc1323=1
  - nfs\_socketsize=60000
  - nfs\_tcp\_socketsize=600000
- Do a web search on "nagle effect"
- netstat -s | grep "socket buffer overflow"



## nfsstat

- Client and Server NFS Info
- nfsstat -cn or -r or -s
  - Retransmissions due to errors
    - Retrans $>5\%$  is bad
  - Badcalls
  - Timeouts
  - Waits
  - Reads

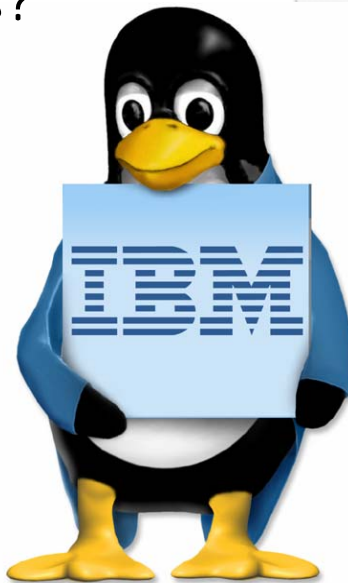


## Useful Links

- **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/blosxomjl.cgi/](http://www.circle4.com/blosxomjl.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/loo.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/loo.htm)
- **8. vmstat command**
  - [publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/loo.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/loo.htm)
- **9. lvmo command**
  - [publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/loo.htm](http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/loo.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://mainline.com/ebrochure>



## Questions?



# Supplementary Slides



## Disk Technologies

- Arbitrated
  - SCSI 20 or 40 mb/sec
  - FC-AL 100mb/sec
  - Devices arbitrate for exclusive control
  - SCSI priority based on address
- Non-Arbitrated
  - SSA 80 or 160mb/sec
  - Devices on loop all treated equally
  - Devices drop packets of data on loop



## Adapter Throughput - SCSI

	100% mby/s	70% mby/s	Bits Bus	Max Devs Width
• SCSI-1	5	3.5	8	8
• Fast SCSI	10	7	8	8
• FW SCSI	20	14	16	16
• Ultra SCSI	20	14	8	8
• Wide Ultra SCSI	40	28	16	8
• Ultra2 SCSI	40	28	8	8
• Wide Ultra2 SCSI	80	56	16	16
• Ultra3 SCSI	160	112	16	16
• Ultra320 SCSI	320	224	16	16
• Ultra640 SCSI	640	448	16	16

- Watch for saturated adapters



Courtesy of <http://www.scsita.org/terms/scsiterms.html>



## Adapter Throughput - Fibre

	100% mbit/s	70% mbit/s
• 133		93
• 266		186
• 530		371
• 1 gbit		717
• 2 gbit		1434

- SSA comes in 80 and 160 mb/sec



## RAID Levels

- Raid-0
  - Disks combined into single volume stripeset
  - Data striped across the disks
- Raid-1
  - Every disk mirrored to another
  - Full redundancy of data but needs extra disks
  - At least 2 I/Os per random write
- Raid-0+1
  - Striped mirroring
  - Combines redundancy and performance



## RAID Levels

- RAID-5
  - Data striped across a set of disks
  - 1 more disk used for parity bits
  - Parity may be striped across the disks also
  - At least 4 I/Os per random write (read/write to data and read/write to parity)
  - Uses hot spare technology

