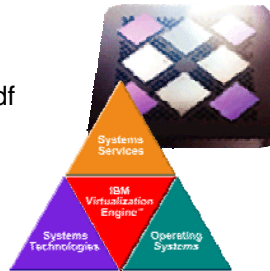


AIX Performance Tuning

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

Jaqui Lynch
Senior Systems Engineer
Mainline Information Systems



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**
 - 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
 - lspas, svmon, vmstat, vmo, ioo

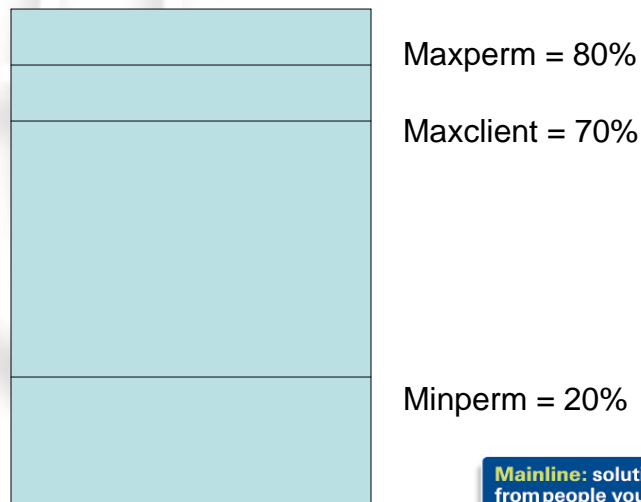
Mainline: solutions you need
from people you trust

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`

Mainline: solutions you need
from people you trust

minperm, maxperm, maxclient



Mainline: solutions you need
from people you trust

Tuneables 1/4

- vmo minperm%
 - Value below which we steal from computational pages - default is 20%
 - We lower this to something like 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
 - 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
- vmo 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
- 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

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

Tuneables 2/4

- 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 numclust
 - Sequential write behind
 - Number of 16k clusters processed by write behind
- ioo j2_maxRandomWrite
 - Random write behind for JFS2
 - On a per file basis
 - Default is 0 – try 32
- ioo j2_nPagesPerWriteBehindCluster
 - Default is 32
 - Number of pages per cluster for writebehind
- ioo j2_nRandomCluster
 - JFS2 sequential write behind
 - Distance apart before random is detected
- ioo j2_nBufferPerPagerDevice
 - Minimum filesystem bufrstructs for JFS2 – default 512, effective at filesystem mount

Use these with care

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

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 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
- `ioo 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

Mainline: solutions you need
from people you trust

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`
- `vmo lru_poll_interval`
 - Set to 10
 - Improves responsiveness of the LRUD when it is running

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
- 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 lrutable 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

Mainline: solutions you need from people you trust

Starter Set of tunables

```
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
```

NB please test these before putting into production

```
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
```

Use caution

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

Mainline: solutions you need from people you trust

vmstat -I

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

System configuration: lcpu=24 mem=102656MB ent=12.0

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 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 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

ioo Output

- lvm_bufcnt = 9
- minpgahead = 2
- maxpgahead = 8
- maxrandwrt = 32 (default is 0)
- numclust = 1
- numfsbufs = 186
- sync_release_iloc = 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

Mainline: solutions you need
from people you trust

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`

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

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	mintime	maxtime	avgqsz	qfull	
	0.0	0.0	0.2	0.0	0	

Mainline: solutions you need
from people you trust

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

```
tiostat: 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

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

Mainline: solutions you need
from people you trust

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

Mainline: solutions you need
from people you trust

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

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

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
- 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.tep and inetd.conf, etc for things that should not start
 - Make sure you don't do it partially
 - i.e. portmap is in rc.tep 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)
- Use CIO where it will benefit you
 - Do not use for Oracle binaries
- Leave DISK_ASYNCH_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
- Iostat
- 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

lparstat

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

%entc – percent of entitled capacity

Lbusy – logical processor utilization for system and user

Vcswh – 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>

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

lparstat -i

lparstat -i output

Node Name	: abcdef
Partition Name	: lpar abcdef
Partition Number	: 3
Type	: Dedicated-SMT
Mode	: Capped
Entitled Capacity	: 4.00
Partition Group-ID	: 32771
Shared Pool ID	: -
Online Virtual CPUs	: 4
Maximum Virtual CPUs	: 16
Minimum Virtual CPUs	: 1
Online Memory	: 16384 MB
Maximum Memory	: 65536 MB
Minimum Memory	: 512 MB
Variable Capacity Weight	: -
Minimum Capacity	: 1.00
Maximum Capacity	: 16.00
Capacity Increment	: 1.00
Maximum Physical CPUs in system	: 16
Active Physical CPUs in system	: 16
Active CPUs in Pool	: -
Unallocated Capacity	: -
Physical CPU Percentage	: 100.00%

lparstat -i output

Node Name	: xyw
Partition Name	: lpar xyw
Partition Number	: 7
Type	: Shared-SMT
Mode	: Uncapped
Entitled Capacity	: 2.00
Partition Group-ID	: 32775
Shared Pool ID	: 0
Online Virtual CPUs	: 2
Maximum Virtual CPUs	: 8
Minimum Virtual CPUs	: 1
Online Memory	: 16384 MB
Maximum Memory	: 24576 MB
Minimum Memory	: 8192 MB
Variable Capacity Weight	: 128
Minimum Capacity	: 0.20
Maximum Capacity	: 8.00
Capacity Increment	: 0.01
Maximum Physical CPUs in system	: 16
Active Physical CPUs in system	: 16
Active CPUs in Pool	: 16
Unallocated Capacity	: 0.00
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		
	0.27%	49.63%		
cpu0	cpu2	cpu1	cpu3	
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%	52.73%	47.28%

Mainline: solutions you need
from people you trust

Async I/O

Total number of AIOs in use

```
pstat -a | grep aios | wc -l
```

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

Mainline: solutions you need
from people you trust

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
- Often seen in HACMP configurations as well

Mainline: solutions you need from people you trust

Network

- `no -a` & `nfso -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 `nfso` commands
 - `nfsstat`
 - `rfc1323` and `nfs_rfc1323`

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 -i

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
en0	1500	link#2	0.9.6b.dd.28.d6	545573	0	429166	4	0
en0	1500	172.26.157	v157s08.rsb.iic.i	545573	0	429166	4	0

Mainline: solutions you need
from people you trust

netstat -m

Kernel malloc statistics:

***** CPU 0 *****

By size	inuse	calls	failed	delayed	free	hiwat	freed
32	82	88	0	0	46	10484	0
64	89	641	0	3	103	10484	0
128	124	370	0	2	68	5242	0
256	27	499	0	2	21	10484	0
512	48	157898	0	1	16	13105	0
1024	86	663	0	38	66	5242	0
2048	91	561	0	66	49	7863	0
4096	4	19	0	0	25	2621	0
8192	4	4	0	1	13	1310	0
16384	2	2	0	1	6	655	0
32768	1	1	0	1	3	327	0
131072	1	1	0	0	9	22	0

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

Network tuneables

- no -a
- Using no
 - rfc1323 = 1
 - sb_max=1310720 (\geq 1MB)
 - 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=600000
 - nfs_tcp_socketsize=600000
- Do a web search on “nagle effect”
- netstat -s | grep “socket buffer overflow”

Mainline: solutions you need
from people you trust

nfsstat

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

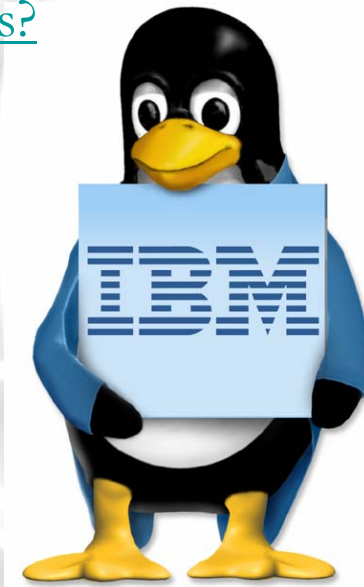
Mainline: solutions you need
from people you trust

Useful Links

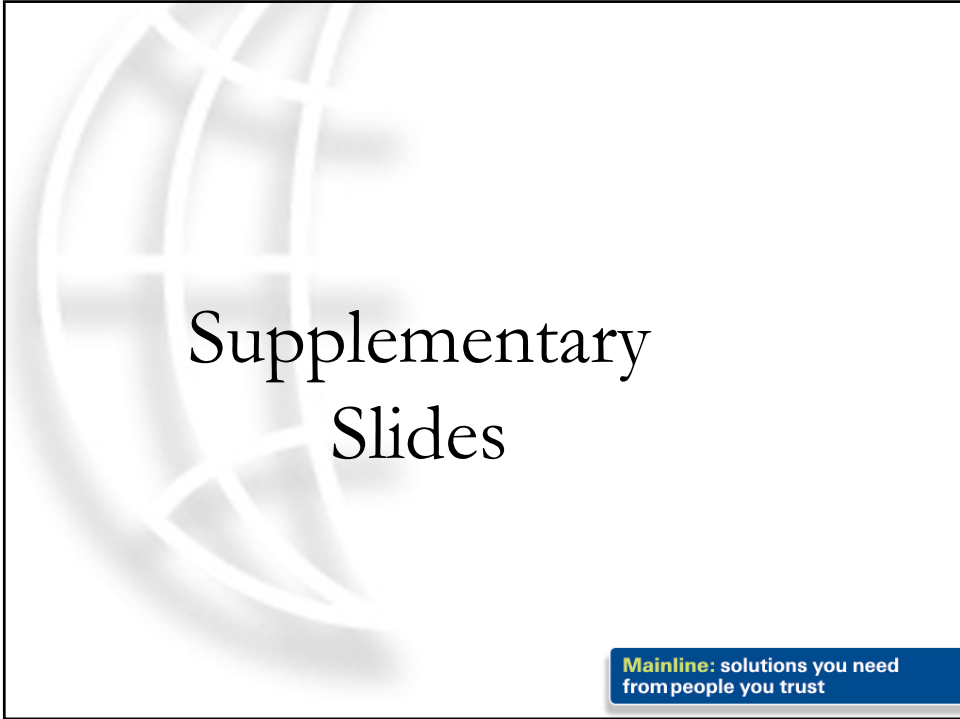
- **1. Ganglia**
 - ganglia.info
- **2. Lparmon**
 - www.alphaworks.ibm.com/tech/lparmon
- **3. Nmon**
 - www.ibm.com/collaboration/wiki/display/WikiPtype/nmon
- **4. Nmon Analyser**
 - www.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmonanalyser
- **5. Jaqui's AIX* Blog**
 - Has a base set of performance tunables for AIX 5.3 - www.circle4.com/blossomjl.cgi/
- **6. vmo command**
 - 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
- **8. vmstat command**
 - publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.doc/cmds/aixcmds3/ioo.htm
- **9. lvm command**
 - 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://mainline.com/ebrochure>

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

Questions?



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



Supplementary Slides

Mainline: solutions you need
from people you trust



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

Mainline: solutions you need
from people you trust

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>

Mainline: solutions you need
from people you trust

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

Mainline: solutions you need
from people you trust

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

Mainline: solutions you need
from people you trust

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

Mainline: solutions you need
from people you trust