AIX Performance Tuning

Jaqui Lynch  lynchj@forsythe.com

Presentation current as of 7/9/2012

Agenda

• Part 1
  o How to avoid performance crit sits!
  o CPU
  o Starter Set of Tunables

• Part 2
  o Memory tuning

• Part 3
  o Network

• Part 4
  o Volume groups and filesystems

• Part 5
  o Performance Tools

• Part 6
  o AIO and CIO for Oracle (if time permits)
Terms to understand – 1/2

• Process
  o A process is an activity within the system that is started with a command, a shell script, or another process.

• Run Queue
  o Each CPU has a dedicated run queue. A run queue is a list of runnable threads, sorted by thread priority value. There are 256 thread priorities (zero to 255). There is also an additional global run queue where new threads are placed.

• Time Slice
  o The CPUs on the system are shared among all of the threads by giving each thread a certain slice of time to run. The default time slice of one clock tick is 10 ms

Terms to understand – 2/2

• Cache Coherency
  o All processors work with the same virtual and real address space and share the same real memory. However, each processor may have its own cache, holding a small subset of system memory. To guarantee cache coherency the processors use a snooping logic. Each time a word in the cache of a processor is changed, this processor sends a broadcast message over the bus. The processors are “snooping” on the bus, and if they receive a broadcast message about a modified word in the cache of another processor, they need to verify if they hold this changed address in their cache. If they do, they invalidate this entry in their cache.

• Processor Affinity
  o If a thread is running on a CPU and gets interrupted and redispached, the thread is placed back on the same CPU (if possible) because the processor’s cache may still have lines that belong to the thread. If it is dispatched to a different CPU, the thread may have to get its information from main memory. Alternatively, it can wait until the CPU where it was previously running is available, which may result in a long delay.
  o AIX automatically tries to encourage processor affinity by having one run queue per CPU. Processor affinity can also be forced by binding a thread to a processor with the bindprocessor command. CPUs in the system.
Tips to keep out of trouble

- Check the performance apars have all been installed
  - Yes this means you need to stay current
  - See slides under backup

- Keep firmware up to date
  - In particular look at the firmware history for your server to see if there are performance problems fixed

- Information on the firmware updates can be found at:

- Firmware history including release dates can be found at:
  - Power7 Midrange
  - Power7 High end

IO Wait and why it is not necessarily useful

- System has 3 threads blocked (red threads)
- SMT is turned on
- There are 4 threads ready to run so they get dispatched and each is using 80% user and 20% system

Metrics would show:
- %user = .8 * 4 / 4 = 80%
- %sys = .2 * 4 / 4 = 20%
- Idle will be 0% as no core is waiting to run threads
- IO Wait will be 0% as no core is idle waiting for IO to complete as something else got dispatched to that core

**SO we have IO wait**
**BUT we don’t see it**
## Adapter Priorities affect Performance

### Power 770 Layout

<table>
<thead>
<tr>
<th>Slot</th>
<th>Desc</th>
<th>Pri</th>
<th>Alloc</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>B2-BP Flox</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>B2-BP Flox</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>B2-BP Flox</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>B2</td>
<td>B2-BP Flox</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>B2-BP Flox</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>B2-BP Flox</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>B2-BP Flox</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>B2-BP Flox</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>B2-BP Flox</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>E2</td>
<td>B2-BP Flox</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

### CPU

- **Top 1224/MC**: 16x slots
- **Bottom 1224/MC**: 8x slots
- **4x slot**: 2x slots
- **Adapter Priority**: 1-10
Monitoring CPU

- User, system, wait and idle are fine for dedicated LPARs
- They are not fine for SPLPAR or dedicated donating LPARs
- You need to measure and charge back based on used CPU cycles
- Moral of the story – use Physc (Physical consumed)
- Iparstat
  - Use with no flags to view partition configuration and processor usage

 Logical Processesors

Logical Processors represent SMT threads

<table>
<thead>
<tr>
<th>LPAR 1</th>
<th>LPAR 2</th>
<th>LPAR 3</th>
<th>LPAR 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMT on</td>
<td>SMT off</td>
<td>SMT on</td>
<td>SMT off</td>
</tr>
<tr>
<td>vmstat - lcpu=4</td>
<td>lcpu=2</td>
<td>lcpu=4</td>
<td>lcpu=2</td>
</tr>
<tr>
<td>L L L L</td>
<td>L L L L</td>
<td>L L L L</td>
<td>L L L L</td>
</tr>
</tbody>
</table>

2 Cores Dedicated

2 Cores Dedicated

V=0.6  V=0.6  V=0.4  V=0.4

PU=1.2  Weight=128  PU=0.8  Weight=192

Hypervisor

Core  Core  Core  Core  Core  Core

Logical  Virtual  Physical
Applications and SPLPARs

• Applications do not need to be aware of Micro-Partitioning

• Not all applications benefit from SPLPARs

• Applications that may not benefit from Micro-Partitioning:
  o Applications with a strong response time requirements for transactions may find Micro-Partitioning detrimental:
    – Because virtual processors can be dispatched at various times during a timeslice
    – May result in longer response time with too many virtual processors:
      • Each virtual processor with a small entitled capacity is in effect a slower CPU
      – Compensate with more entitled capacity (2-5% PUs over plan)
  o Applications with polling behavior
  o CPU intensive application examples: DSS, HPC, SAS

• Applications that are good candidates for Micro-Partitioning:
  o Ones with low average CPU utilization, with high peaks:
    – Examples: OLTP, web applications, mail server, directory servers

• In general Oracle databases are fine in the shared processor pool

• For licensing reasons you may want to use a separate pool for databases

Useful processor Commands

• lsdev –Cc processor
• lsattr –EL proc0
• bindprocessor –q
• sar –P ALL
• topas, nmon
• lparstat
• vmstat (use –l or -v)
• iostat
• mpstat –s
• lssrad -av
Using `sar -P ALL (Power7 & SMT4)`

AIX bpicnim 1 7 00F6934B4C00 10/05/11 (1 core and 2 VPs)

System configuration: lcpu=8 ent=1.00 mode=Uncapped

```
19:40:49 cpu  %usr %sys %wio %idle physc %entc
19:40:50  0    7    88    0    5  0.01   1.4
  1    0    0   100   0.00  0.3
  2    0   100    0   0.00  0.3
  3    0    0   100   0.00  0.3
  7    0   59    0   0.00  0.0
-    0   98    0  98.97  97.5
  -    0    1    0   99.00  2.5
```

In the above cpu4-6 are missing as they are 0 so sar did not print them to save space.

`mpstat -s 1 1`

System configuration: lcpu=8 ent=1.0 mode=Uncapped

```
Proc0          Proc4
  2.26%          0.01%
  1.33%          0.00%
  0.31%          0.00%
  0.31%          0.00%
  0.31%          0.00%
  0.00%          0.01%
  0.01%          0.00%
```

Using `sar -P ALL – AIX 5.3 or SMT2`

SMT2 Example

```
sar -P ALL 1 1
AIX sys01a 3 5 00CDAF6F4C00 ent=0.80
System Configuration: lcpu=4 ent=0.80
```

```
12:18:01 cpu  %usr %sys %wio %idle physc %entc
12:18:01  0    0    7    0   93  0.03   3.3
  1    0  100    0   0.00  0.37  46.8
  2  100    0   0.00  0.38  46.9
  3    0    1  99    0  0.02  3.1
-    94    0   0   6.00  100
```

System is clearly busy – now map this to the `mpstat` command.

```
mpstat -s 1 1
System configuration: lcpu=4 ent=0.80
```

```
Proc0         Proc1
  39.99%       39.76%
  2.55%       37.45%
  37.57%       2.19%
```

Oracle tends to really like SMT and to take advantage of it.
**Starter set of tunables 1**

**For AIX v5.3**
No need to set memory Affinity=0 after 5.3 tl05

**MEMORY**
- vmo -p -o minperm%=3
- vmo -p -o maxperm%=90
- vmo -p -o maxclient%=90
- vmo -p -o minfree=960
- vmo -p -o maxfree=1088
- vmo -p -o lru_file_repage=0
- vmo -p -o lru_poll_interval=10
- vmo -p -o page_steal_method=1

**For AIX v6 or v7**
Memory defaults are already correctly except minfree and maxfree.
If you upgrade from a previous version of AIX using migration then you need to check the settings after

---

**Starter set of tunables 2**

The parameters below should be reviewed and changed (see vmstat -v and lvmo -a later)

**PBUFS**
Use the new way (coming up)

**JFS2**
- ioo -p -o j2_maxPageReadAhead=128
  (default above may need to be changed for sequential)
- j2_dynamicBufferPreallocation=16
  Default that may need tuning
  Replaces tuning j2_nBufferPerPagerDevice

**Network changes in later slide**
MEMORY

Memory Types

- Persistent
  - Backed by filesystems

- Working storage
  - Dynamic
    - Includes executables and their work areas
    - Backed by page space

- Prefer to steal from persistent as it is cheap

- minperm, maxperm, maxclient, lru_file_repage and page_steal_method all impact these decisions
Correcting Paging

11173706 paging space I/Os blocked with no psbuf

lsps output on above system that was paging before changes were made to tunables

<table>
<thead>
<tr>
<th>lsps -a</th>
<th>Page Space</th>
<th>Physical Volume</th>
<th>Volume Group</th>
<th>Size</th>
<th>%Used</th>
<th>Active</th>
<th>Auto</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>paging01</td>
<td>hdisk3</td>
<td>pagingvg</td>
<td>16384MB</td>
<td>25</td>
<td>yes</td>
<td>yes</td>
<td>lv</td>
<td>lv</td>
</tr>
<tr>
<td>paging00</td>
<td>hdisk2</td>
<td>pagingvg</td>
<td>16384MB</td>
<td>25</td>
<td>yes</td>
<td>yes</td>
<td>lv</td>
<td>lv</td>
</tr>
<tr>
<td>hd6</td>
<td>hdisk0</td>
<td>rootvg</td>
<td>16384MB</td>
<td>25</td>
<td>yes</td>
<td>yes</td>
<td>lv</td>
<td>lv</td>
</tr>
</tbody>
</table>

What you want to see

lsps -a

<table>
<thead>
<tr>
<th>Page Space</th>
<th>Physical Volume</th>
<th>Volume Group</th>
<th>Size</th>
<th>%Used</th>
<th>Active</th>
<th>Auto</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>paging01</td>
<td>hdisk3</td>
<td>pagingvg</td>
<td>16384MB</td>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>lv</td>
</tr>
<tr>
<td>paging00</td>
<td>hdisk2</td>
<td>pagingvg</td>
<td>16384MB</td>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>lv</td>
</tr>
<tr>
<td>hd6</td>
<td>hdisk0</td>
<td>rootvg</td>
<td>16384MB</td>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>lv</td>
</tr>
</tbody>
</table>

lsps -s

<table>
<thead>
<tr>
<th>Total Paging Space</th>
<th>Percent Used</th>
<th>Can also use vmstat -l and vmstat -s</th>
</tr>
</thead>
<tbody>
<tr>
<td>49152MB</td>
<td>1%</td>
<td></td>
</tr>
</tbody>
</table>

Should be balanced – NOTE VIO Server comes with 2 different sized page datasets on one hdisk (at least until FP24)

Default Page Space Calculation

- AIX Client default
  - hd6 must be >= 64MB, others must be >= 16MB
  - Page space can use no more than 20% disk
  - If real <256MB then page space = 2 x real
  - If real >= 256MB then page space = 256MB

- VIO Server
  - 1 x 512MB and 1 x 1024MB page space both on the same disk
  - Supposedly fixed if installing FP24 but not if upgrade

- On my VIO:
  - # lsps -a

<table>
<thead>
<tr>
<th>lsps -a</th>
<th>Page Space</th>
<th>Physical Volume</th>
<th>Volume Group</th>
<th>Size</th>
<th>%Used</th>
<th>Active</th>
<th>Auto</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>hd6</td>
<td>hdisk0</td>
<td>rootvg</td>
<td>4096MB</td>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>lv</td>
<td>0</td>
</tr>
</tbody>
</table>
PAGE SPACE BEST PRACTICE

- More than one page volume
- All the same size including hd6
- Page spaces must be on different disks to each other
- Do not put on hot disks
- Mirror all page spaces that are on internal or non-raided disk

Memory with lru_file_repage=0

- minperm=3
  - Always try to steal from filesystems if filesystems are using more than 3% of memory
- maxperm=90
  - Soft cap on the amount of memory that filesystems or network can use
  - Superset so includes things covered in maxclient as well
- maxclient=90
  - Hard cap on amount of memory that JFS2 or NFS can use – SUBSET of maxperm
page_steal_method

- Default in 5.3 is 0, in 6 and 7 it is 1
- What does 1 mean?
- lru_file_repage=0 tells LRUD to try and steal from filesystems
- Memory split across mempools
- LRUD manages a mempool and scans to free pages
- 0 – scan all pages
- 1 – scan only filesystem pages

page_steal_method Example

- 500GB memory
- 50% used by file systems (250GB)
- 50% used by working storage (250GB)
- mempools = 5
- So we have at least 5 LRUDs each controlling about 100GB memory
- Set to 0
  - Scans all 100GB of memory in each pool
- Set to 1
  - Scans only the 50GB in each pool used by filesystems
- Reduces cpu used by scanning
- When combined with CIO this can make a significant difference
Rough Anatomy of an I/O

- LVM requests a PBUF
  - Pinned memory buffer to hold I/O request in LVM layer

- Then placed into an FSBUF
  - 3 types
    - These are also pinned
    - Filesystem: JFS
    - Client: NFS and VxFS
    - External Pager: JFS2

- If paging also need PSBUFs (also pinned)
  - Used for I/O requests to and from page space

- Then queue I/O to hdisk (queue_depth)

- Then queue it to adapter (num_cmd elems)

- Adapter queues it to the disk subsystem

lvmo –a Output

2725270 pending disk I/Os blocked with no pbuf

vgname = rootvg
pv_pbuf_count = 512
total_vg_pbufs = 1024
max_vg_pbuf_count = 16384
pervg_blocked_io_count = 0 this is rootvg
pv_min_pbuf = 512
Max_vg_pbuf_count = 0
global_blocked_io_count = 2725270 this is the others

Use lvmo –v xxxxvg -a
For other VGs we see the following in pervg_blocked_io_count

<table>
<thead>
<tr>
<th>vg</th>
<th>blocked</th>
<th>total_vg_pbufs</th>
</tr>
</thead>
<tbody>
<tr>
<td>nimvg</td>
<td>29</td>
<td>512</td>
</tr>
<tr>
<td>sasvg</td>
<td>2719199</td>
<td>1024</td>
</tr>
<tr>
<td>backupvg</td>
<td>6042</td>
<td>4608</td>
</tr>
</tbody>
</table>

lvmo –v sasvg –o pv_pbuf_count=2048
vmstat –v Output

3.0 minperm percentage
90.0 maxperm percentage
45.1 numperm percentage
45.1 numclient percentage
90.0 maxclient percentage

1468217 pending disk I/Os blocked with no pbuf
11173706 paging space I/Os blocked with no psbuf
2048 file system I/Os blocked with no fsbuf
238 client file system I/Os blocked with no fsbuf
39943187 external pager file system I/Os blocked with no fsbuf

numclient=numperm so most likely the I/O being done is JFS2 or NFS or VxFS
Based on the blocked I/Os it is clearly a system using JFS2
It is also having paging problems
pbufs also need reviewing

Memory Pools and fre column

- fre column in vmstat is a count of all the free pages across all the memory pools
- When you look at fre you need to divide by memory pools
- Then compare it to maxfree and minfree
- This will help you determine if you are happy, page stealing or thrashing

- You can see high values in fre but still be paging
- In below if maxfree=2000 and we have 10 memory pools then we only have 990 pages free in each pool on average. With minfree=960 we are page stealing and close to thrashing.

<table>
<thead>
<tr>
<th>kthr</th>
<th>memory</th>
<th>page</th>
<th>faults</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>b</td>
<td>p</td>
<td>avm</td>
<td>fre</td>
</tr>
<tr>
<td>70</td>
<td>309</td>
<td>0</td>
<td>8552080</td>
<td>9902</td>
</tr>
</tbody>
</table>

9902/10 = 990.2
Calculating minfree and maxfree

```
vmstat -v | grep memory
  3 memory pools

You may need to look using kdb as mempools seems to have disappeared in some levels of 6.1
kdb
memp *
Quit

vmo -a | grep free
  maxfree = 1088
  minfree = 960

Calculation is:
minfree = (max (960,(120 * lcpus) / memory pools))
maxfree = minfree + (Max(maxpgahead,j2_maxPageReadahead) * lcpus) / memory pools

So if I have the following:
Memory pools = 3 (from vmo –a or kdb)
J2_maxPageReadahead = 128
CPUS = 6 and SMT on so lcpu = 12
So minfree = (max(960,(120 * 12)/3)) = 1440 / 3 = 480 or 960 whichever is larger
And maxfree = minfree + (128 * 12) / 3 = 960 + 512 = 1472
I would probably bump this to 1536 rather than using 1472 (nice power of 2)
```

If you over allocate these values it is possible that you will see high values in the “fre” column of a vmstat and yet you will be paging.

NETWORK
Typically we set the following for both versions:

```
NETWORK
no   -p -o rfc1323=1
no   -p -o tcp_sendspace=262144
no   -p -o tcp_recvspace=262144
no   -p -o udp_sendspace=65536
no   -p -o udp_recvspace=655360
```

Also check the actual NIC interfaces and make sure they are set to at least these values.

Check sb_max is at least 1040000 – increase as needed.

```
ifconfig

ifconfig -a output

en0:  flags=1e080863,480<UP,BROADCAST,NOTRAILERS, RUNNING,SIMPLEX,MULTICAST,T,GROUPRT,64BIT, CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
       inet 10.2.0.37 netmask 0xfffffe00 broadcast 10.2.1.255
tcp_sendspace 65536 tcp_recvspace 65536 rfc1323 0

lo0:  flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
       inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
inet6 ::1/0
tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1

These override no, so they will need to be set at the adapter.
Additionally you will want to ensure you set the adapter to the correct setting if it runs at less than GB, rather than allowing auto-negotiate.

Stop inetd and use chdev to reset adapter (i.e. en0)
Or use chdev with the –P and the changes will come in at the next reboot
```
```bash
# ifconfig -a
en6:
  flags=1e080863,580<UP,BROADCAST,NOTRAILERS,Running,SIMPLEX,MULTICAST,GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
      inet 192.168.2.5   netmask 0xffffffff broadcast 192.168.2.255
tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1

lo0:
  flags=e08084b,1c0<UP,BROADCAST,LOOPBACK,Running,SIMPLEX,MULTICAST,GROUPRT,64BIT,LARGESEND,CHAIN>
      inet 127.0.0.1 netmask 0xffffffff broadcast 127.255.255.255
      inet6 ::1%1/0
tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

## Network

<table>
<thead>
<tr>
<th>Interface</th>
<th>Speed</th>
<th>MTU</th>
<th>tcp_sendspace</th>
<th>tcp_recvspace</th>
<th>rfc1323</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo0</td>
<td>N/A</td>
<td>16896</td>
<td>131072</td>
<td>131072</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet</td>
<td>10/100 mb</td>
<td>1500</td>
<td>262144</td>
<td>262144</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet</td>
<td>1000 (Gb)</td>
<td>9000</td>
<td>131072</td>
<td>165536</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet</td>
<td>1000 (Gb)</td>
<td>9000</td>
<td>262144</td>
<td>262144</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet</td>
<td>1000 (Gb)</td>
<td>9000</td>
<td>262144</td>
<td>262144</td>
<td>1</td>
</tr>
<tr>
<td>Ethernet</td>
<td>1000 (Gb)</td>
<td>9000</td>
<td>131072</td>
<td>131072</td>
<td>1</td>
</tr>
<tr>
<td>Virtual Ethernet</td>
<td>N/A</td>
<td>any</td>
<td>262144</td>
<td>262144</td>
<td>1</td>
</tr>
<tr>
<td>InfiniBand</td>
<td>N/A</td>
<td>2044</td>
<td>131072</td>
<td>131072</td>
<td>1</td>
</tr>
</tbody>
</table>

Check up to date information at:

AIX v6.1
Other Network

• netstat –v
  o Look for overflows and memory allocation failures
    Max Packets on S/W Transmit Queue: 884
    S/W Transmit Queue Overflow: 9522
  o “Software Xmit Q overflows” or “packets dropped due to memory allocation
    failure”
    – Increase adapter xmit queue
    – Use lsattr –EL ent? To see setting
  o Look for receive errors or transmit errors
  o dma underruns or overruns
  o mbuf errors

• tcp_nodelay (or tcp_nodelayack)
  o Disabled by default
  o 200ms delay by default as it waits to piggy back acks on packets

• Also check errpt – people often forget this

Volume groups and file systems
Basics

• Data layout will have more impact than most tunables
• Plan in advance

• Large hdisks are evil
  • I/O performance is about bandwidth and reduced queuing, not size
  • 10 x 50gb or 5 x 100gb hdisk are better than 1 x 500gb
  • Also larger LUN sizes may mean larger PP sizes which is not great for lots of little filesystems
  • Need to separate different kinds of data i.e. logs versus data

• The issue is queue_depth
  • In process and wait queues for hdisks
  • In process queue contains up to queue_depth I/Os
  • hdisk driver submits I/Os to the adapter driver
  • Adapter driver also has in process and wait queues
  • SDD and some other multi-path drivers will not submit more than queue_depth I/Os
  to an hdisk which can affect performance
  • Adapter driver submits I/Os to disk subsystem
  • Default client qdepth for vSCSI is 3
    • chdev –i hdisk? –a queue_depth=20 (or some good value)
  • Default client qdepth for NPIV is set by the Multipath driver in the client

The AIX IO stack

From: PE23 Disk I/O Tuning in AIX v6.1 – Dan Braden and Steven Nasypany, October 2010
### sar -d

- **sar –d 2 6 shows:**

  - **avque**
    - Average IOs in the wait queue
    - Waiting to get sent to the disk (the disk’s queue is full)
    - Values > 0 indicate increasing queue_depth may help performance
    - Used to mean number of IOs in the disk queue
  - **avwait**
    - Time waiting in the wait queue (ms)
  - **avserv**
    - I/O service time when sent to disk (ms)

<table>
<thead>
<tr>
<th>device</th>
<th>%busy</th>
<th>avque</th>
<th>r+w/s</th>
<th>Kbs/s</th>
<th>avwait</th>
<th>avserv</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdisk7</td>
<td>0</td>
<td>0.0</td>
<td>2</td>
<td>160</td>
<td>0.0</td>
<td>1.9</td>
</tr>
<tr>
<td>hdisk8</td>
<td>19</td>
<td>0.3</td>
<td>568</td>
<td>14337</td>
<td>23.5</td>
<td>2.3</td>
</tr>
<tr>
<td>hdisk9</td>
<td>2</td>
<td>0.0</td>
<td>31</td>
<td>149</td>
<td>0.0</td>
<td>0.9</td>
</tr>
</tbody>
</table>

### iostat -D

**Extended Drive Report**

- Also check out the –aD option

<table>
<thead>
<tr>
<th>device</th>
<th>xfer:</th>
<th>%tm_act</th>
<th>bps</th>
<th>tps</th>
<th>bread</th>
<th>bwrtn</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdisk18</td>
<td></td>
<td>8.8</td>
<td>3.4M</td>
<td>148.4</td>
<td>409.9</td>
<td>3.4M</td>
</tr>
<tr>
<td>read</td>
<td>rps</td>
<td>avgser</td>
<td>minser</td>
<td>maxser</td>
<td>timeouts</td>
<td>fails</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>write</td>
<td>wps</td>
<td>avgser</td>
<td>minser</td>
<td>maxser</td>
<td>timeouts</td>
<td>fails</td>
</tr>
<tr>
<td></td>
<td>148.3</td>
<td>3.6</td>
<td>0.2</td>
<td>632.2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>queue</td>
<td>avgtime</td>
<td>mintime</td>
<td>maxtime</td>
<td>avgwqsz</td>
<td>avgsqsz</td>
<td>sqfull</td>
</tr>
<tr>
<td></td>
<td>24.5</td>
<td>0.0</td>
<td>631.7</td>
<td>4.0</td>
<td>0.0</td>
<td>83.2</td>
</tr>
</tbody>
</table>

**tps**  
Transactions per second – transfers per second to the adapter

**avgserv**  
Average service time

**Avgtme**  
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. Also look at avgsqsz.

**Per IBM**  
Average IO sizes:

<table>
<thead>
<tr>
<th></th>
<th>read = bread/rps</th>
<th>write = bwrtn/wps</th>
</tr>
</thead>
</table>
Interactive nmon D

```
#topas nmonperlNFSqqqqqqqqqqqqqqqHost=1750violqqqqqqqqRefresh=2 secqqq21:56.53qq

Disk - Service times and Queues

x Name Read Service Write Service Wait ServQ WaitQ ServQ
x milli-seconds milli-seconds Size Size Full
xhdisks1 0.0 0.0 0.0 0.0 0.0 0.0
xhdisks2 0.0 0.0 0.0 0.0 0.0 0.0
xhdisks0 0.0 0.0 0.0 0.0 0.0 0.0
xhdisks3 0.0 0.0 0.0 0.0 0.0 0.0
xhdisks5 0.0 0.0 0.0 0.0 0.0 0.0
xcd0 0.0 0.0 0.0 0.0 0.0 0.0
xhdisks4 0.0 0.0 0.0 0.0 0.0 0.0
xTotals(MB/s) Read=0.0 Write=0.0 Size(GB)=273 Free(GB)=64
```

Adapter Queue Problems

- Look at BBBF Tab in NMON Analyzer or run fcstat command
- Adapter device drivers use DMA for IO
- From fcstat on each fcs
- NOTE these are since boot

**FC SCSI Adapter Driver Information**

- No DMA Resource Count: 0
- No Adapter Elements Count: 2567
- No Command Resource Count: 34114051

- No DMA resource – adjust max_xfer_size
- No adapter elements – adjust num_cmd elems
- No command resource – adjust num_cmd elems

- If using NPIV make changes to VIO and client, not just VIO
### Adapter Tuning

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
<th>Default Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>bus_intr_lvl</td>
<td>115</td>
<td>Bus interrupt level</td>
<td>False</td>
</tr>
<tr>
<td>bus_io_addr</td>
<td>0xdfc00</td>
<td>Bus I/O address</td>
<td>False</td>
</tr>
<tr>
<td>bus_mem_addr</td>
<td>0xe8040000</td>
<td>Bus memory address</td>
<td>False</td>
</tr>
<tr>
<td>init_link</td>
<td>al</td>
<td>INIT Link flags</td>
<td>True</td>
</tr>
<tr>
<td>intr_priority</td>
<td>3</td>
<td>Interrupt priority</td>
<td>False</td>
</tr>
<tr>
<td>lg_term_dma</td>
<td>0x800000</td>
<td>Long term DMA</td>
<td>True</td>
</tr>
<tr>
<td>max_xfer_size</td>
<td>0x100000</td>
<td>Maximum Transfer Size</td>
<td>True (16MB DMA)</td>
</tr>
<tr>
<td>num_cmd elems</td>
<td>200</td>
<td>Maximum number of COMMANDS to queue to the adapter</td>
<td>True</td>
</tr>
<tr>
<td>pref_alpa</td>
<td>0x1</td>
<td>Preferred AL_PA</td>
<td>True</td>
</tr>
<tr>
<td>sw_fc_class</td>
<td>2</td>
<td>FC Class for Fabric</td>
<td>True</td>
</tr>
</tbody>
</table>

Changes I often make (test first):

- max_xfer_size = 0x200000 Maximum Transfer Size  True  **128MB DMA area for data I/O**
- num_cmd elems = 1024 Maximum number of COMMANDS to queue to the adapter True
- lg_term_dma is the DMA area for control I/O

Check these are ok with your disk vendor!!!

chdev -l fcs0 -a max_xfer_size=0x200000 -a num_cmd elems=1024 -P
chdev -l fcs1 -a max_xfer_size=0x200000 -a num_cmd elems=1024 -P

Remember make changes too both VIO servers and client LPARs if using NPIV
VIO server setting must be at least as large as the client setting

### My VIO Server and NPIV Client Adapter Settings

#### VIO SERVER

```
lsattr -El fcs0
lg_term_dma  0x800000  Long term DMA          True
max_xfer_size 0x200000 Maximum Transfer Size  True
num_cmd elems  1024     Maximum number of COMMANDS to queue to the adapter True
```

#### NPIV Client (running at defaults before changes)

```
lsattr -El fcs0
lg_term_dma  0x800000  Long term DMA          True
max_xfer_size 0x200000 Maximum Transfer Size  True
num_cmd elems  512     Maximum Number of COMMAND Elements True
```
Parameter Settings - Summary

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DEFAULTS</th>
<th>AIXv5.3</th>
<th>AIXv6</th>
<th>AIXv7</th>
<th>NEW SET ALL TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETWORK (no)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rfc1323</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>tcp_sendspace</td>
<td>16384</td>
<td>16384</td>
<td>16384</td>
<td>262144 (1Gb)</td>
<td></td>
</tr>
<tr>
<td>tcp_recvspace</td>
<td>16384</td>
<td>16384</td>
<td>16384</td>
<td>262144 (1Gb)</td>
<td></td>
</tr>
<tr>
<td>udp_sendspace</td>
<td>9216</td>
<td>9216</td>
<td>9216</td>
<td>65536</td>
<td></td>
</tr>
<tr>
<td>udp_recvspace</td>
<td>42080</td>
<td>42080</td>
<td>42080</td>
<td>655360</td>
<td></td>
</tr>
<tr>
<td>MEMORY (vmo)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>minperm%</td>
<td>20</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>JFS, NFS, VxFS, JFS2</td>
</tr>
<tr>
<td>maxperm%</td>
<td>80</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>JFS2, NFS</td>
</tr>
<tr>
<td>lru_file_repage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>lru_poll_interval</td>
<td>?</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Minfree</td>
<td>960</td>
<td>960</td>
<td>960</td>
<td>calculation</td>
<td></td>
</tr>
<tr>
<td>Maxfree</td>
<td>1088</td>
<td>1088</td>
<td>1088</td>
<td>calculation</td>
<td></td>
</tr>
<tr>
<td>page_steal_method</td>
<td>0</td>
<td>0 /1 (TL)</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>JFS2 (ioo)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j2_maxPageReadAhead</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>as needed</td>
<td></td>
</tr>
<tr>
<td>j2_dynamicBufferPreallocation</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>as needed</td>
<td></td>
</tr>
</tbody>
</table>

Other Interesting Tunables

- noatime
  - Why write a record every time you read or touch a file?
  - mount command option
  - Use for redo and archive logs

- Release behind (or throw data out of file system cache)
  - rbr – release behind on read
  - rbw – release behind on write
  - rbrw – both

- Read the various AIX Difference Guides:
Performance Tools

Useful Links

- AIX Wiki
  - https://www.ibm.com/developerworks/wikis/display/WikiPtype/AIX

- Workload Estimator
  - http://ibm.com/systems/support/tools/estimator

- Performance Tools Wiki

- Performance Monitoring
  - https://www.ibm.com/developerworks/wikis/display/WikiPtype/Performance+Monitoring+Documentation

- Other Performance Tools
  - https://www.ibm.com/developerworks/wikis/display/WikiPtype/Other+Performance+Tools
    - Includes new advisors for Java, VIOS, Virtualization

- VIOS Advisor
• topas
  o New –L flag for LPAR view

• nmon analyzer
  o Windows tool so need to copy the .nmon file over in ascii mode
  o Opens as an excel spreadsheet and then analyses the data
  o Also look at nmon consolidator

• sar
  o sar -A -o filename 2 30 >/dev/null
  o Creates a snapshot to a file – in this case 30 snaps 2 seconds apart
  o Must be post processed on same level of system

• errpt
• Check for changes from defaults

• filemon
  o filemon -v -o filename -O all
  o sleep 30
  o trcstop

• pstat to check async I/O in 5.3
  o pstat -a | grep aio | wc –l

• perfpmr to build performance info for IBM if reporting a PMR
  o /usr/bin/perfpmr.sh 300
- `nmon -ft -A -s 15 -c 120`
  - Grabs a 30 minute nmon snapshot with async I/O

- `nmon -ft -A -M -L -^ -s 15 -c 120`
  - Same as above but includes large pages and some other features

- Must be running nmon12e or higher
- Nmon comes with AIX at 5.3 tl09 or 6.1 tl01 and higher BUT on 5.3 I download the latest version from the web so I get the latest v12 for sure
- Creates a file in the working directory that ends .nmon
- This file can be transferred to your PC and interpreted using nmon analyser or other tools
- `nmon -f -O` -- now gets seastats for VIO server
- `nmon -f -K` - dump libperfstat structures


**Courtesy Nigel Griffiths - IBM**
nmon on POWER6 & AIX6 + - New Features for V12

- Disk Service Times
- Selecting Particular Disks
- Time Drift
- Multiple Page Sizes
- Timestamps in UTC & no. of digits
- More Kernel & Hypervisor Stats *
- High Priority nmon
  - Advanced, POWER6 and AIX6 items
    - Virtual I/O Server SEA
    - Partition Mobility (POWER6)
    - WPAR & Application Mobility (AIX6)
    - Dedicated Donating (POWER6)
    - Folded CPU count (SPLPAR)
    - Multiple Shared Pools (POWER6)
    - Fibre Channel stats via entstat
VIOS Advisor

• https://www.ibm.com/developerworks/wikis/display/WikiPtype/VIOS+Advisor

• Application that collects performance metrics and does a health check

• Following slides run on a production VIO during a regular production day

The ratings and recommendations in the table below were chosen with the following information:

<table>
<thead>
<tr>
<th>Hostname</th>
<th>vio1.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartitionID</td>
<td>2</td>
</tr>
<tr>
<td>Monitoring Start Time</td>
<td>03/09 11:15:19</td>
</tr>
<tr>
<td>Monitoring Stop Time</td>
<td>03/09 13:49:39</td>
</tr>
<tr>
<td>Duration</td>
<td>120 min</td>
</tr>
</tbody>
</table>

**SYSTEM - CONFIGURATION**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Family</td>
<td>POWER6</td>
</tr>
<tr>
<td>Server Model</td>
<td>IBM,9117-MMA</td>
</tr>
<tr>
<td>Server Frequency</td>
<td>4.208 GHz</td>
</tr>
<tr>
<td>Server - Online CPUs</td>
<td>10 cores</td>
</tr>
<tr>
<td>Server - Maximum Supported CPUs</td>
<td>16 cores</td>
</tr>
<tr>
<td>VIOS Level</td>
<td>2.2.0.13-FP24 SP-03</td>
</tr>
<tr>
<td>VIOS Advisor Release</td>
<td>121211B</td>
</tr>
</tbody>
</table>
### VIOS - CPU

<table>
<thead>
<tr>
<th>Name</th>
<th>Measured Value</th>
<th>Recommended Value</th>
<th>First Observed</th>
<th>Last Observed</th>
<th>Risk 1=lowest 5=highest</th>
<th>Impact 1=lowest 5=highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Capacity</td>
<td>1.0 ent</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>CPU Consumption</td>
<td>avg:5.4% (cores:0.1) high:40.2% (cores:0.5)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Processing Mode</td>
<td>Shared CPU, (Uncapped)</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Variable Capacity</td>
<td>200</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Virtual Processors</td>
<td>2 vCPUs</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>SMT Mode</td>
<td>SMT2</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### SYSTEM - SHARED PROCESSING POOL

<table>
<thead>
<tr>
<th>Name</th>
<th>Measured Value</th>
<th>Recommended Value</th>
<th>First Observed</th>
<th>Last Observed</th>
<th>Risk 1=lowest 5=highest</th>
<th>Impact 1=lowest 5=highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Pool Monitoring</td>
<td>enabled</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Shared Processing Pool Capacity</td>
<td>10.0 ent.</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Free CPU Capacity</td>
<td>avg_free:9.4 ent. lowest_free:7.7 ent.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
## VIOS - I/O Activity

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk I/O Activity</td>
<td>avg: 229 iops @ 32KB peak: 1916 iops @ 137KB</td>
</tr>
<tr>
<td>Network I/O Activity</td>
<td>[ avgSend: 0 iops 0.0MBps, avgRcv: 0 iops 0.0MBps ] [ peakSend: 0 iops 0.0MBps, peakRcv: 0 iops 0.0MBps ]</td>
</tr>
</tbody>
</table>

## VIOS - Disk Adapters

<table>
<thead>
<tr>
<th>Name</th>
<th>Measured Value</th>
<th>Recommended Value</th>
<th>First Observed</th>
<th>Last Observed</th>
<th>Risk 1=lowest 5=highest</th>
<th>Impact 1=lowest 5=highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC Adapter Count</td>
<td>3</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>FC Avg I/Os</td>
<td>avg: 77 iops @ 32KB</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>03/09 13:45:19</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>FC Adapter Utilization</td>
<td>optimal</td>
<td>-</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>FC Port Speeds</td>
<td>running at speed</td>
<td>-</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
### VIOS - DISK DRIVES

<table>
<thead>
<tr>
<th>Name</th>
<th>Measured Value</th>
<th>Recommended Value</th>
<th>First Observed</th>
<th>Last Observed</th>
<th>Risk 1=lowest 5=highest</th>
<th>Impact 1=lowest 5=highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Drive Count</td>
<td>93</td>
<td>-</td>
<td>03/09 11:45:19</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>I/Os Blocked</td>
<td>optimal</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Long I/O Latency (hdisk3)</td>
<td>avg:9.7ms (9.7 + 0.0) high:11.5ms (11.5 + 0.0)</td>
<td>Range: 8-12ms</td>
<td>03/09 12:35:58</td>
<td>03/09 13:44:02</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### VIOS - MEMORY

<table>
<thead>
<tr>
<th>Name</th>
<th>Measured Value</th>
<th>Recommended Value</th>
<th>First Observed</th>
<th>Last Observed</th>
<th>Risk 1=lowest 5=highest</th>
<th>Impact 1=lowest 5=highest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Memory</td>
<td>4.000 GB</td>
<td>7.000 GB</td>
<td>03/09 11:45:19</td>
<td>n/a</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Available Memory</td>
<td>0.346 GB</td>
<td>1.5 GB Avail.</td>
<td>03/09 11:45:39</td>
<td>03/09 13:45:05</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Paging Rate</td>
<td>0.2 MB/s pg rate</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Paging Space Size</td>
<td>8.000 GB</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Free Paging Space</td>
<td>7.923 GB free</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Pinned Memory</td>
<td>1.262 GB pinned</td>
<td>-</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
ORACLE
Asynchronous I/O and Concurrent I/O

Async I/O - v5.3

Total number of AIOs in use
psstat -a | grep aios | wc -l

AIO maxservers
Isattr -El aio0 -a maxservers

Maximum AIOservers started since boot maxservers
psattr -El aio0 | wc -l

per cpu True

NB – maxservers is a per processor setting in AIX 5.3

Or new way for Posix AIOs is:
ps -k | grep aio | wc -l

4205

At AIX v5.3 tl05 this is controlled by aioo command

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

Isattr -El aio0
autoconfig defined STATE to be configured at system restart True

fastpath enable State of fast path True

kprocprio 39 Server PRIORITY True

maxreqs 4096 Maximum number of REQUESTS True

maxservers 10 MAXIMUM number of servers per cpu True

minservers 1 MINIMUM number of servers True

AIO is used to improve performance for I/O to raw LVs as well as filesystems.
iostat -A

iostat -A async IO

System configuration: lcpu=16 drives=15
aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait

<table>
<thead>
<tr>
<th></th>
<th>% tm_act</th>
<th>Kbps</th>
<th>tps</th>
<th>Kb_read</th>
<th>Kb_wrt</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>hdisk6</td>
<td>23.4</td>
<td>1846.1</td>
<td>195.2</td>
<td>381485298</td>
<td>61892856</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hdisk5</td>
<td>15.2</td>
<td>1387.4</td>
<td>143.8</td>
<td>304880506</td>
<td>28324064</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hdisk9</td>
<td>13.9</td>
<td>1695.9</td>
<td>163.3</td>
<td>373163558</td>
<td>34144512</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If maxg close to maxr or maxservers then increase maxreqs or maxservers

Old calculation – no longer recommended

minservers = active number of CPUs or 10 whichever is the smaller number
maxservers = number of disks times 10 divided by the active number of CPUs
maxreqs = 4 times the number of disks times the queue depth

***Reboot anytime the AIO Server parameters are changed

Async I/O – AIX v6 and v7

No more smit panels and no AIO servers start at boot
Kernel extensions loaded at boot
AIO servers go away if no activity for 300 seconds
Only need to tune maxreqs normally

llo -a -F | more
aio_active = 0
aio_maxreqs = 65536
aio_maxservers = 30
aio_minservers = 3
aio_server_inactivity = 300
posix_aio_active = 0
posix_aio_maxreqs = 65536
posix_aio_maxservers = 30
posix_aio_minservers = 3
posix_aio_server_inactivity = 300

##Restricted tunables
aio_fastpath = 1
aio_fsfastpath = 1
aio_kprocprio = 39
aio_multidssus = 1
aio_sample_rate = 5
aio_samples_per_cycle = 6
posix_aio_fastpath = 1
posix_aio_fsfastpath = 1
posix_aio_kprocprio = 39
posix_aio_sample_rate = 5
posix_aio_samples_per_cycle = 6

pstat -a | grep aio
22 a 1608e 1 1608e 0 0 1 aioPpool
24 a 1804a 1 1804a 0 0 1 aioLpool
Oracle now recommending the following as starting points

<table>
<thead>
<tr>
<th></th>
<th>5.3</th>
<th>6.1 or 7 (non CIO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>minservers</td>
<td>100</td>
<td>3 - default</td>
</tr>
<tr>
<td>maxservers</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>maxreqs</td>
<td>16384</td>
<td>65536 – default</td>
</tr>
</tbody>
</table>

These are per CPU
So for lcpu=10 and maxservers=100 you get 1000 aioservers

AIO applies to both raw I/O and file systems

Grow maxservers as you need to

• DIO
  - Direct I/O
  - Around since AIX v5.1, also in Linux
  - 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 (after 4MB)
  - Reduces CPU and eliminates overhead copying data twice
  - Reads are asynchronous
  - No filesystem readahead
  - No lrud or syncd overhead
  - No double buffering of data
  - Inode locks still used
  - Benefits heavily random access workloads
DIO and CIO

- **CIO**
  - Concurrent I/O – AIX only, not in Linux
  - Only available in JFS2
  - Allows performance close to raw devices
  - 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
  - Speeds up writes significantly
  - Saves memory and CPU for double copies
  - No filesystem readahead
  - No lrud or syncd overhead
  - No double buffering of data
  - **Not all apps benefit from CIO and DIO** – some are better with filesystem caching and some are safer that way

- **When to use it**
  - Database DBF files, redo logs and control files and flashback log files.
  - Not for Oracle binaries or archive log files

DIO/CIO Oracle Specifics

- **Use CIO where it will benefit you**
  - Do not use for Oracle binaries
  - Ensure redo logs and control files are in their own filesystems with the correct (512) blocksize
    - Use lsfs –q to check 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**
- **Remember CIO uses DIO under the covers**

- **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
Telling Oracle to use CIO and AIO

If your Oracle version (10g/11g) supports it then configure it this way:

There is no default set in Oracle 10g do you need to set it

Configure Oracle Instance to use CIO and AIO in the init.ora (PFILE/SPFILE)

- `disk_async_io = true` (init.ora)
- `filesystemio_options = setall` (init.ora)

Note if you do backups using system commands while the database is up then you will need to use the 9i method below for v10 or v11

If not (i.e. 9i) then you will have to set the filesystem to use CIO in the /etc filesystems

- `options = cio` (/etc/filesystems)
- `disk_async_io = true` (init.ora)

Do not put anything in the filesystem that the Database does not manage – remember there is no inode lock on writes

Or you can use ASM and let it manage all the disk automatically

Also read Metalink Notes #257338.1, #360287.1

See Metalink Note 960055.1 for recommendations

Do not set it in both places (config file and /etc/filesystems)

Demoted I/O in Oracle

- CIO write fails because IO is not aligned to FS blocksize
  - i.e. app writing 512 byte blocks but FS has 4096

- Ends up getting redone
  - Demoted I/O consumes more kernel CPU
  - And more physical I/O

- To find demoted I/O (if JFS2)

  ```bash
  trace –aj 59B,59C ; sleep 2 ; trcstop ; trcrpt –o directio.trcrpt
grep –i demoted directio.trcrpt
  ```

Look in the report for:
Thank you for your time

If you have questions please email me at: lynchj@forsythe.com