AIX Performance Tuning
CPU and Memory

Jaqui Lynch
lynchj@forsythe.com

Handout at:

Agenda

• Part 1
  • CPU
  • Memory tuning
  • Starter Set of Tunables

• Part 2
  • I/O
  • Volume Groups and File systems
  • AIO and CIO for Oracle

• Part 3
  • Network
  • Performance Tools
Starting Point

• Baseline
  • A baseline should be taken regularly but at least prior to and after any kind of changes
• Baseline can be a number of things
  • I use a combination of nmon, my own scripts and IBM’s perfpmr
• PerfPMR is downloadable from a public website:
  • Choose appropriate version based on the AIX release
Logical Processors

Logical Processors represent SMT threads

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

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

2 Cores Dedicated VPs under the covers

Hypervisor

Core Core Core Core Core

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

Dispatching in shared pool

- VP gets dispatched to a core
  - First time this becomes the home node
  - All SMT threads for the VP go with the VP
- VP runs to the end of its entitlement
  - If it has more work to do and noone else wants the core it gets more
  - If it has more work to do but other VPs want the core then it gets context switched and put on the home node runQ
  - If it can’t get serviced in a timely manner it goes to the global runQ and ends up running somewhere else but its data may still be in the memory on the home node core
Understand SMT4

- SMT
  - Threads dispatch via a Virtual Processor (VP)
  - Overall more work gets done (throughput)
  - Individual threads run a little slower
    - SMT1: Largest unit of execution work
    - SMT2: Smaller unit of work, but provides greater amount of execution work per cycle
    - SMT4: Smallest unit of work, but provides the maximum amount of execution work per cycle
  - On POWER7, a single thread cannot exceed 65% utilization
  - On POWER6 or POWER5, a single thread can consume 100%
  - Understand thread dispatch order

POWER5/6 vs POWER7/8 - SMT Utilization

POWER7 SMT=2 70% & SMT=4 63% tries to show potential spare capacity
- Escaped most peoples attention
- VM goes 100% busy at entitlement & 100% from there on up to 10 x more CPU
- SMT4 100% busy 1st CPU now reported as 63% busy
- 2nd, 3rd and 4th LCPUs each report 12% idle time which is approximate

POWER8 Notes
  - Uplift from SMT2 to SMT4 is about 30%
  - Uplift from SMT4 to SMT8 is about 7%
  - Check published rPerf Numbers

Diagram courtesy of IBM

POWER5/6 vs POWER7/8 Virtual Processor Unfolding

- Virtual Processor is activated at different utilization threshold for P5/P6 and P7
- P5/P6 loads the 1st and 2nd SMT threads to about 80% utilization and then unfolds a VP
- P7 loads first thread on the VP to about 50% then unfolds a VP
  - Once all VPs unfolded then 2nd SMT threads are used
  - Once 2nd threads are loaded then tertaries are used
  - This is called raw throughput mode

Why?
Raw Throughput provides the highest per-thread throughput and best response times at the expense of activating more physical cores

- Both systems report same physical consumption
- This is why some people see more cores being used in P7 than in P6/P5, especially if they did not reduce VPs when they moved the workload across.
- HOWEVER, idle time will most likely be higher
- I call P5/P6 method “stack and spread” and P7 “spread and stack”

**BASICALLY:** POWER7/POWER8 will activate more cores at lower utilization levels than earlier architectures when excess VP’s are present

Scaled Throughput

- P7 and higher with AIX v6.1 TL08 and AIX v7.1 TL02
- Dispatches more SMT threads to a VP core before unfolding additional VPs
- Tries to make it behave a bit more like P6

- **Raw** provides the highest per-thread throughput
  and best response times at the expense of activating more physical core

- **Scaled** provides the highest core throughput at the expense of per-thread response times
  and throughput.
  It also provides the highest system-wide throughput per VP because tertiary thread capacity is “not left on the table.”

- `schedo -p -o vpm_throughput_mode=`
  0 Legacy Raw mode (default)
  1 “Enhanced Raw” mode with a higher threshold than legacy
  2 Scaled mode, use primary and secondary SMT threads
  4 Scaled mode, use all four SMT threads
  8 Scaled mode, use eight SMT threads (POWER8, AIX v7.1 required)
  Dynamic Tunable

- SMT unfriendly workloads could see an enormous per thread performance degradation
Checking SMT

# smtctl

This system is SMT capable.
This system supports up to 8 SMT threads per processor.
SMT is currently enabled.
SMT boot mode is set to enabled.
SMT threads are bound to the same virtual processor.

proc0 has 4 SMT threads.
Bind processor 0 is bound with proc0
Bind processor 1 is bound with proc0
Bind processor 2 is bound with proc0
Bind processor 3 is bound with proc0

proc8 has 4 SMT threads.
Bind processor 4 is bound with proc8
Bind processor 5 is bound with proc8
Bind processor 6 is bound with proc8
Bind processor 7 is bound with proc8

Show VP Status on POWER8

echo vpm | kdb

<table>
<thead>
<tr>
<th>VSD Thread State.</th>
<th>CPU CPFVR VP_STATE FLAGS SLEEP_STATE PROD_TIME: SECS NSECS CEDE_LAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 ACTIVE 0 AWAKE</td>
<td>000000000000000000 00000000 00</td>
</tr>
<tr>
<td>1 255 ACTIVE 0 AWAKE</td>
<td>00000000000005540A09 38EB345F 00</td>
</tr>
<tr>
<td>2 255 ACTIVE 0 AWAKE</td>
<td>00000000000005540A09 38E7B944 00</td>
</tr>
<tr>
<td>3 255 ACTIVE 0 AWAKE</td>
<td>00000000000005540A09 38E7C250 00</td>
</tr>
<tr>
<td>4 0 DISABLED 0 AWAKE</td>
<td>000000000000000000 00000000 00</td>
</tr>
<tr>
<td>5 0 DISABLED 0 AWAKE</td>
<td>000000000000000000 00000000 00</td>
</tr>
<tr>
<td>6 0 DISABLED 0 AWAKE</td>
<td>000000000000000000 00000000 00</td>
</tr>
<tr>
<td>7 0 DISABLED 0 AWAKE</td>
<td>000000000000000000 00000000 00</td>
</tr>
<tr>
<td>8 0 DISABLED 0 AWAKE</td>
<td>000000000000000000 00000000 00</td>
</tr>
<tr>
<td>9 11 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB1B8A 02</td>
</tr>
<tr>
<td>10 11 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB1A8A 02</td>
</tr>
<tr>
<td>11 11 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB1886 02</td>
</tr>
<tr>
<td>12 11 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB16B6 02</td>
</tr>
<tr>
<td>13 11 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB15D6 02</td>
</tr>
<tr>
<td>14 11 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB14B6 02</td>
</tr>
<tr>
<td>15 11 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB11AB 02</td>
</tr>
<tr>
<td>16 0 DISABLED 0 AWAKE</td>
<td>000000000000000000 00000000 00</td>
</tr>
<tr>
<td>17 1 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB81E5 02</td>
</tr>
<tr>
<td>18 1 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB84C0 02</td>
</tr>
<tr>
<td>19 1 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB85F0 02</td>
</tr>
<tr>
<td>20 1 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB84D0 02</td>
</tr>
<tr>
<td>21 1 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB85F0 02</td>
</tr>
<tr>
<td>22 1 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB84F9 02</td>
</tr>
<tr>
<td>23 1 DISABLED 0 SLEEPING</td>
<td>00000000000005540A09 33BB852F 02</td>
</tr>
</tbody>
</table>

System is SMT8 so CPU0-7 are a VP, CPU8-15 are a VP and so on
More on Dispatching

How dispatching works
Example - 1 core with 6 VMs assigned to it

VPs for the VMs on the core get dispatched (consecutively) and their threads run
As each VM runs the cache is cleared for the new VM
When entitlement reached or run out of work CPU is yielded to the next VM
Once all VMs are done then system determines if there is time left
Assume our 6 VMs take 6MS so 4MS is left
Remaining time is assigned to still running VMs according to weights
VMs run again and so on

Problem - if entitlement too low then dispatch window for the VM can be too low
If VM runs multiple times in a 10ms window then it does not run full speed as cache has to be warmed up
If entitlement higher then dispatch window is longer and cache stays warm longer - fewer cache misses

Entitlement and VPs

- Utilization calculation for CPU is different between POWER5, 6 and POWER7
- VPs are also unfolded sooner (at lower utilization levels than on P6 and P5)
- May also see high VCSW in lparstat

- This means that in POWER7 you need to pay more attention to VPs
  - You may see more cores activated a lower utilization levels
  - But you will see higher idle
  - If only primary SMT threads in use then you have excess VPs

- Try to avoid this issue by:
  - Reducing VP counts
  - Use realistic entitlement to VP ratios
    - 10x or 20x is not a good idea
    - Try setting entitlement to .6 or .7 of VPs
  - Ensure workloads never run consistently above 100% entitlement
  - Too little entitlement means too many VPs will be contending for the cores
  - NOTE – VIO server entitlement is critical – SEAs scale by entitlement not VPs

- All VPs have to be dispatched before one can be redispached

- Performance may (in most cases, will) degrade when the number of Virtual Processors in an LPAR exceeds the number of physical processors
- The same applies with VPs in a shared pool LPAR – these should exceed the cores in the pool
Iparstat 30 2

Iparstat 30 2 output
System configuration: type=Shared mode=Uncapped smt=4 lcpu=72 mem=319488MB psize=17 ent=12.00

%user %sys %wait %idle physc %entc lbusy app vcs w phint
46.8 11.6 0.5 41.1 11.01 91.8 16.3 4.80 28646 738
48.8 10.8 0.4 40.0 11.08 92.3 16.9 4.88 26484 763

lcpu=72 and smt=4 means I have 72/4=18 VPs but pool is only 17 cores - BAD
psize = processors in shared pool
lbusy = %occupation of the LCPUs at the system and user level
app = Available physical processors in the pool
vcs w = Virtual context switches (virtual processor preemptions)

High VCSW may mean too many VPs or entitlement too low
phint = phantom interrupts received by the LPAR

interrupts targeted to another partition that shares the same physical processor
i.e. LPAR does an I/O so cedes the core, when I/O completes the interrupt is sent to the
core but different LPAR running so it says "not for me"

NOTE – Must set “Allow performance information collection” on the LPARs to see good values for app, etc
Required for shared pool monitoring

mpstat -s

mpstat –s 1 1
System configuration: lcpu=64 ent=10.0 mode=Uncapped

 shows breakdown across the VPs (proc*) and smt threads (cpu*)
### lparstat & mpstat –s  POWER8 Mode Example

**b814aix1: lparstat –s**

```
System configuration: type=Shared mode=Uncapped smt=8 lcpu=48 mem=32768MB psize=2 ent=0.50
%user %sys %wait %idle physc %entc lbusy app vcsw phint
----- ----- ------ ------ ----- ----- ------ --- ----- -----
0.0   0.1  0.0   99.9  0.00   0.8    2.3   1.96  244   0
0.0   0.2  0.0   99.8  0.00   1.0    2.3   1.96  257   0
```

**b814aix1: mpstat –s**

```
System configuration: lcpu=48 ent=0.5 mode=Uncapped

Proc0

0.00%

cpu0 cpu1 cpu2 cpu3 cpu4 cpu5 cpu6 cpu7
0.00% 0.00% 0.00% 0.00% 0.00% 0.00% 0.00% 0.00%

Proc8

0.00%

cpu8 cpu9 cpu10 cpu11 cpu12 cpu13 cpu14 cpu15
0.00% 0.00% 0.00% 0.00% 0.00% 0.00% 0.00% 0.00%
```

---

### mpstat –v  - Use this to monitor VP activity

```
# mpstat –v

System configuration: lcpu=24 ent=0.5 mode=Uncapped

<table>
<thead>
<tr>
<th>vcpu</th>
<th>lcpu</th>
<th>us</th>
<th>sy</th>
<th>wa</th>
<th>id</th>
<th>pbusy</th>
<th>pc</th>
<th>VTB(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>27.45</td>
<td>25.76</td>
<td>0.20</td>
<td>46.59</td>
<td>0.00</td>
<td>0.00</td>
<td>53.2%</td>
<td>0.00</td>
</tr>
<tr>
<td>1</td>
<td>0.08</td>
<td>0.26</td>
<td>0.00</td>
<td>14.04</td>
<td>0.00</td>
<td>0.00</td>
<td>56.6%</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.15</td>
<td>0.02</td>
<td>14.29</td>
<td>0.00</td>
<td>0.00</td>
<td>57.8%</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.15</td>
<td>0.00</td>
<td>14.51</td>
<td>0.00</td>
<td>0.00</td>
<td>57.8%</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>30.43</td>
<td>26.46</td>
<td>0.06</td>
<td>43.04</td>
<td>0.00</td>
<td>0.00</td>
<td>58.0%</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.08</td>
<td>0.00</td>
<td>13.87</td>
<td>0.00</td>
<td>0.00</td>
<td>58.0%</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>0.08</td>
<td>0.00</td>
<td>13.95</td>
<td>0.00</td>
<td>0.00</td>
<td>58.0%</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>0.00</td>
<td>0.08</td>
<td>0.00</td>
<td>13.99</td>
<td>0.00</td>
<td>0.00</td>
<td>58.0%</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>35.79</td>
<td>22.03</td>
<td>0.04</td>
<td>42.14</td>
<td>0.00</td>
<td>0.00</td>
<td>57.8%</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>35.79</td>
<td>21.11</td>
<td>0.04</td>
<td>0.88</td>
<td>0.00</td>
<td>0.00</td>
<td>57.8%</td>
<td>0.00</td>
</tr>
<tr>
<td>10</td>
<td>0.00</td>
<td>0.31</td>
<td>0.00</td>
<td>13.73</td>
<td>0.00</td>
<td>0.00</td>
<td>57.8%</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>0.31</td>
<td>0.00</td>
<td>13.76</td>
<td>0.00</td>
<td>0.00</td>
<td>57.8%</td>
<td>0.00</td>
</tr>
</tbody>
</table>
```

shows VP and SMT thread usage
Avoiding Problems

- Stay current
- Known memory issues with 6.1 tl9 sp1 and 7.1 tl3 sp1
- Java 7.1 SR1 is the preferred Java for POWER7 and POWER8
- Java 6 SR7 is minimal on POWER7 but you should go to Java 7
- WAS 8.5.2.2
- Refer to Section 8.3 of the Performance Optimization and Tuning Techniques Redbook SG24-8171
- HMC v8 required for POWER8 – does not support servers prior to POWER6
- Remember not all workloads run well in the shared processor pool – some are better dedicated
  - Apps with polling behavior, CPU intensive apps (SAS, HPC), latency sensitive apps (think trading systems)

vmstat -IW

bnim: vmstat -IW 2 2

```
vmstat -IW 60 2
```

System configuration: lcpu=12 mem=24832MB ent=2.00

<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>w</td>
<td>avm</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2708633</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2831669</td>
</tr>
</tbody>
</table>

Note pc=2.42 is 120.0% of entitlement

- I shows I/O oriented view and adds in the p column
- p column is number of threads waiting for I/O messages to raw devices.

- W adds the w column (only valid with –I as well)
  - w column is the number of threads waiting for filesystem direct I/O (DIO) and concurrent I/O (CIO)

- r column is average number of runnable threads (ready but waiting to run + those running)
  - This is the global run queue – use mpstat and look at the rq field to get the run queue for each logical CPU

- b column is average number of threads placed in the VMM wait queue (awaiting resources or I/O)
### System Configuration: lcpu=48 mem=32768MB ent=0.50

<table>
<thead>
<tr>
<th>kthr</th>
<th>memory</th>
<th>page</th>
<th>faults</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### SMT4

<table>
<thead>
<tr>
<th>kthr</th>
<th>memory</th>
<th>page</th>
<th>faults</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### System Configuration: lcpu=24 mem=32768MB ent=0.50

<table>
<thead>
<tr>
<th>kthr</th>
<th>memory</th>
<th>page</th>
<th>faults</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### SMT4

<table>
<thead>
<tr>
<th>kthr</th>
<th>memory</th>
<th>page</th>
<th>faults</th>
<th>cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### AIX Performance Tuning Part 1
mpstat -h

pc and context switches with stolen and donation statistics

# mpstat -h 1 1

System configuration: lcpu=24 ent=0.5 mode=Uncapped

<table>
<thead>
<tr>
<th>cpu</th>
<th>pc</th>
<th>ilcs</th>
<th>vlcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00</td>
<td>1</td>
<td>234</td>
</tr>
<tr>
<td>1</td>
<td>0.00</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>0.00</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>ALL</td>
<td>0.01</td>
<td>1</td>
<td>299</td>
</tr>
</tbody>
</table>

Detailed Cpu Statistics
lparstat -d

# lparstat -d 2 2

System configuration: type=Shared mode=Uncapped smt=4 lcpu=24 mem=32768MB psize=8 ent=0.50

<table>
<thead>
<tr>
<th>%user</th>
<th>%sys</th>
<th>%wait</th>
<th>%idle</th>
<th>physc</th>
<th>%entc</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.3</td>
<td>0.0</td>
<td>99.7</td>
<td>0.01</td>
<td>1.1</td>
</tr>
<tr>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>99.8</td>
<td>0.00</td>
<td>0.8</td>
</tr>
</tbody>
</table>
### Summary Hypervisor Statistics
**lparstat -h**

<table>
<thead>
<tr>
<th>%user</th>
<th>%sys</th>
<th>%wait</th>
<th>%idle</th>
<th>phyc</th>
<th>%entc</th>
<th>lbusy</th>
<th>app</th>
<th>vcsw</th>
<th>phint</th>
<th>%hypv</th>
<th>hcalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
<td>99.5</td>
<td>0.01</td>
<td>1.4</td>
<td>4.0</td>
<td>8.00</td>
<td>201</td>
<td>0</td>
<td>1.1</td>
<td>222</td>
</tr>
<tr>
<td>0.1</td>
<td>0.3</td>
<td>0.0</td>
<td>99.6</td>
<td>0.01</td>
<td>1.1</td>
<td>1.9</td>
<td>7.97</td>
<td>218</td>
<td>0</td>
<td>1.0</td>
<td>305</td>
</tr>
<tr>
<td>0.0</td>
<td>0.2</td>
<td>0.0</td>
<td>99.8</td>
<td>0.00</td>
<td>0.9</td>
<td>2.0</td>
<td>7.98</td>
<td>205</td>
<td>0</td>
<td>0.8</td>
<td>241</td>
</tr>
</tbody>
</table>

**Using sar –mu -P ALL (Power7 & SMT4)**

AIX (ent=10 and 16 VPs) so per VP phyc entitled is about .63

<table>
<thead>
<tr>
<th>Average</th>
<th>cpu</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
<th>physc</th>
<th>%entc</th>
<th>.9 physc</th>
<th>.86 physc</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:24:31</td>
<td>0</td>
<td>77</td>
<td>22</td>
<td>0</td>
<td>1</td>
<td>52</td>
<td>5.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>37</td>
<td>14</td>
<td>1</td>
<td>48</td>
<td>0.18</td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>0.10</td>
<td>1.0</td>
<td>.9 physc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>0.10</td>
<td>1.0</td>
<td>.86 physc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>84</td>
<td>14</td>
<td>0</td>
<td>1</td>
<td>0.49</td>
<td>4.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>42</td>
<td>7</td>
<td>1</td>
<td>50</td>
<td>0.17</td>
<td>1.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>0.10</td>
<td>1.0</td>
<td>.9 physc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>0.10</td>
<td>1.0</td>
<td>.86 physc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>88</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0.51</td>
<td>5.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>11</td>
<td>1</td>
<td>48</td>
<td>0.18</td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>..........</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>0.11</td>
<td>1.1</td>
<td>12.71</td>
<td>127.1</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>55</td>
<td>11</td>
<td>0</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

So we see we are using 12.71 cores which is 127.1% of our entitlement
This is the sum of all the physc lines – cpu0-3 = proc0 = VP0

May see a U line if in SPP and is unused LPAR capacity (compared against entitlement)
nmon Summary

[Image: System Summary Gandalf 4/22/2013]

<table>
<thead>
<tr>
<th>Total I/O Statistics</th>
<th>VP_CPU</th>
<th>User%</th>
<th>sys%</th>
<th>Wait%</th>
<th>Mem%</th>
<th>CPU%</th>
<th>PhysCPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg tps during an interval</td>
<td>8,694</td>
<td>65.7</td>
<td>7.8</td>
<td>0.2</td>
<td>25.6</td>
<td>77.3</td>
<td>12.6</td>
</tr>
<tr>
<td>Max tps during an interval</td>
<td>7,489</td>
<td>98.3</td>
<td>9.0</td>
<td>0.6</td>
<td>28.4</td>
<td>87.1</td>
<td>13.9</td>
</tr>
<tr>
<td>Max tps interval time</td>
<td>10:09:54</td>
<td>4.0</td>
<td>1.3</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Total number of I/O requests</td>
<td>137,837</td>
<td>4.0</td>
<td>1.3</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Total number of bytes written</td>
<td>37,085</td>
<td>4.0</td>
<td>1.3</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Bytes/Second Passed</td>
<td>57.7</td>
<td>4.0</td>
<td>1.3</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

lparstat – bbbl tab in nmon

- lparno: 3
- lparname: gandalf
- CPU in sys: 24
- Virtual CPU: 16
- Logical CPU: 64
- smt threads: 4
- capped: 0
- min Virtual: 8
- max Virtual: 20
- min Logical: 8
- max Logical: 80
- min Capacity: 8
- max Capacity: 16
- Entitled Capacity: 10
- min Memory MB: 131072
- max Memory MB: 327680
- online Memory: 303104
- Pool CPU: 16
- Weight: 150
- pool id: 2

Compare VPs to poolsize
LPAR should not have more VPs than the poolsize
Entitlement and vps from lpar tab in nmon

Physical CPU vs Entitlement - gandalf 4/22/2013

LPAR always above entitlement – increase entitlement

Cpu by thread from cpu_summ tab in nmon

Note mostly primary thread used and some secondary – we should possibly reduce VPs
Different levels of analyzer may show some threads missing – 33g is good but higher levels may not list all threads
Shared Processor Pool Monitoring

Turn on “Allow performance information collection” on the LPAR properties
This is a dynamic change

topas -C

Most important value is app – available pool processors
This represents the current number of free physical cores in the pool

nmon option p for pool monitoring
To the right of PoolCPUs there is an unused column which is the number of free pool cores

nmon analyser LPAR Tab

lparstat

Shows the app column and poolsize

topas -C

<table>
<thead>
<tr>
<th>Topas CEC Monitor</th>
<th>Interval:</th>
<th>Thu Feb 27 08:53:05 2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitions Memory (GB)</td>
<td>Processes</td>
<td></td>
</tr>
<tr>
<td>Shr: 5</td>
<td>Mon: 86.0</td>
<td>InUse: 23.0</td>
</tr>
<tr>
<td>Ded: 0</td>
<td>Avl: -</td>
<td></td>
</tr>
<tr>
<td>Shr: 6</td>
<td>FSz: 16</td>
<td>Don: 0.0</td>
</tr>
<tr>
<td>Ded: 0</td>
<td>APP: 16.0</td>
<td>Stl: 0.0</td>
</tr>
<tr>
<td>Host</td>
<td>OS</td>
<td>Mod</td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>b740nl1</td>
<td>A71</td>
<td>Ud</td>
</tr>
<tr>
<td>b740vib2</td>
<td>A61</td>
<td>U-d</td>
</tr>
<tr>
<td>b740ft2</td>
<td>A71</td>
<td>Ud</td>
</tr>
<tr>
<td></td>
<td>A61</td>
<td>U-d</td>
</tr>
<tr>
<td>b740nl1</td>
<td>A71</td>
<td>Ud</td>
</tr>
<tr>
<td>Host</td>
<td>OS</td>
<td>Mod</td>
</tr>
<tr>
<td>-------</td>
<td>-----</td>
<td>-----</td>
</tr>
</tbody>
</table>

Shows pool size of 16 with all 16 available
Monitor VCSW as potential sign of insufficient entitlement
nmon -p

Shared-CPU-Logical-Partition

PartitionNumber=0 "b750n1" flag:LPAR DRable SMF Shared Uncapped PoolAuth Migratable Not-Donating AMSable
Summary: Entitled= 0.50 Used 1.31 (261.3%) 16.3% of CPUs in System

PoolCPUs= 8 Unused 6.62 16.3% of CPUs in Pool

CPU-Stats Capacity ID-Memory

xmax Phys in sys 6 Cap. Processor Min 0.10 SMLPAR Group:Pool 32773:0
xPhys CPU in sys 6 Cap. Processor Max 8.00 Memory(MB) Min:Max 8192:65536
xVirtual Online 6 Cap. Increment 0.01 Memory(MB) Online 32760
xLogical Online 24 Cap. Unallocated 0.00 Memory Region LMB 256MB min
xPhysical pool 6 Cap. Entitled 0.50 Time-------------------Seconds
xSMT threads/CPU 4 MinKernVirtualCPU 0.05 Time Dispatch Wheel 0.0100
xCPU-------Min-Max Weight------------------------ MaxDispatch Latency 0.0184
xVirtual 1 8 Weight Variable 128 Time Pool Idle 6.6259
xLogical 1 64 Weight Unallocated 0 Time Total Dispatch 1.3067

Shared_Pools MaxPoolCapacity= 8.00 MyPoolMax = 8.00 SharedCPU-Total= 8.00
SharedCPU=6 EntPoolCapacity= 7.50 MyPoolBusy= 1.35 SharedCPU-Busy = 1.35

nmon Analyser LPAR Tab

Shared Pool Utilisation - b750n1 10/4/2013

- PhysicalCPU
- OtherLPARs
- PoolIdle
Memory Types

- **Persistent**
  - Backed by filesystems

- **Working storage**
  - Dynamic
  - Includes executables and their work areas
  - Backed by page space
  - Shows as avm in `vmstat -l` (multiply by 4096 to get bytes instead of pages) or as %comp in nmon analyser or as a percentage of memory used for computational pages in `vmstat -v`
  - ALSO NOTE – if %comp is near or >97% then you will be paging and need more memory

- **Prefer to steal from persistent as it is cheap**

- **minperm, maxperm, maxclient, lru_file_repage and page_steal_method all impact these decisions**
Checking tunables

• Look at /etc/tunables/nextboot
  /etc/tunables/nextboot
  vmo:
    maxfree = "2000"
    minfree = "1000"

  no:
    udp_recvspace = "655360"
    udp_sendspace = "65536"
    tcp_recvspace = "262144"
    tcp_sendspace = "262144"
    rfc1323 = "1"

  Also run commands like “vmo –a –F”

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
  • lru_file_repage goes away in v7 later TLs
    • It is still there but you can no longer change it

All AIX systems post AIX v5.3 (tl04 I think) should have these 3 set
On v6.1 and v7 they are set by default
Check /etc/tunables/nextboot to make sure they are not overridden from
defaults on v6.1 and v7
**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
Correcting Paging

From `vmstat -v`
```
11173706 paging space I/Os blocked with no psbuf
```

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

```
lsps
Page Space    Physical Volume   Volume Group    Size        %Used  Active   Auto  Type
paging01   hdisk3                  pagingvg     16384MB     25    yes       yes      lv
paging00   hdisk2                  pagingvg     16384MB     25    yes       yes      lv
hd6          hdisk0                   rootvg 16384MB     25    yes        yes lv
```

`lsps -s`
```
Total Paging Space   Percent Used Can also use vmstat -l and vmstat -s
49152MB               25%
```

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

**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
If you can't make hd6 as big as the others then swap it off after boot

All real paging is bad

---

Looking for Problems

- `lssrad -av`
- `mpstat -d`
- `topas -M`
- `svmon`
  - Try `-G -O unit=auto,timestamp=on,pgsz=on,affinity=detail` options
  - Look at Domain affinity section of the report
- Etc etc
Memory Problems

- Look at computational memory use
  - Shows as avm in a vmstat –I (multiply by 4096 to get bytes instead of pages)
    
    System configuration: lcpu=48 mem=32768MB ent=0.50
    
    r b p w avm fre fi fo pi po fr sr in sy cs us sy id wa pc ec
    
    0 0 0 0 807668 7546118 0 0 0 0 0 1 159 161 0 0 99 0 0.01 1.3
    
    AVM above is about 3.08GB which is about 9% of the 32GB in the LPAR
    
    - or as %comp in nmon analyser
    
    - or as a percentage of memory used for computational pages in vmstat –v
      
      - NOTE – if %comp is near or >97% then you will be paging and need more memory

- Try svmon –P –Osor=pgsp –Ounit=MB | more
  - This shows processes using the most pagespace in MB
  - You can also try the following:
    
    svmon –P –Ofiltercat=exclusive –Ofiltertype=working –Ounit=MB | more

svmon

```
# svmon -G -O unit=auto -i 2 2

Unit: auto

----------------------------------------
size  inuse  free  pin  virtual  available  mmode
memory 16.0G 8.26G 7.74G 5.50G 10.3G 7.74G  Ded
pg space 12.0G 2.43G

     work  pers  cint other
pin  5.01G 0K  4.11M  497.44M
in use 8.06G 0K  202.29M

Unit: auto

----------------------------------------
size  inuse  free  pin  virtual  available  mmode
memory 16.0G 8.26G 7.74G 5.50G 10.3G 7.74G  Ded
pg space 12.0G 2.43G

     work  pers  cint other
pin  5.01G 0K  4.11M  497.44M
in use 8.06G 0K  202.29M
```

Keep an eye on memory breakdown especially pinned memory. High values mean someone has pinned something.
svmon

```bash
# svmon -G -O unit=auto,timestamp=on,pgsz=on,affinity=detail -i 2

Unit: auto                                   Timestamp: 16:27:26

<table>
<thead>
<tr>
<th>size</th>
<th>inuse</th>
<th>free</th>
<th>pin</th>
<th>virtual</th>
<th>available</th>
<th>mmode</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory</td>
<td>8.00G</td>
<td>3.14G</td>
<td>4.86G</td>
<td>2.20G</td>
<td>2.57G</td>
<td>5.18G</td>
</tr>
<tr>
<td>pg space</td>
<td>4.00G</td>
<td></td>
<td>10.4M</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

work pers clnt other
pin 1.43G OK OK 794.95M
in use 2.57G OK 589.16M

Domain affinity free used total filecache lcpus
0 4.86G 2.37G 7.22G 567.50M 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

svmon pgsp

```bash

# svmon -P -Osrtseg=pgsp -Ounit=MB | more

Unit: MB

```
<table>
<thead>
<tr>
<th>Pid</th>
<th>Command</th>
<th>Inuse</th>
<th>Pin</th>
<th>Pgsp</th>
<th>Virtual</th>
</tr>
</thead>
<tbody>
<tr>
<td>19660878</td>
<td>java</td>
<td>340.78</td>
<td>58.7</td>
<td>0</td>
<td>230.12</td>
</tr>
<tr>
<td>7798966</td>
<td>java</td>
<td>249.05</td>
<td>58.8</td>
<td>0</td>
<td>216.24</td>
</tr>
<tr>
<td>6750304</td>
<td>cimserver</td>
<td>173.93</td>
<td>58.5</td>
<td>0</td>
<td>173.86</td>
</tr>
<tr>
<td>7864386</td>
<td>rmcd</td>
<td>159.07</td>
<td>72.0</td>
<td>0</td>
<td>155.81</td>
</tr>
<tr>
<td>7209186</td>
<td>cimprovagt</td>
<td>155.44</td>
<td>58.4</td>
<td>0</td>
<td>155.34</td>
</tr>
<tr>
<td>8978494</td>
<td>smitty</td>
<td>154.95</td>
<td>58.4</td>
<td>0</td>
<td>153.55</td>
</tr>
<tr>
<td>7733488</td>
<td>cimlistener</td>
<td>152.21</td>
<td>58.4</td>
<td>0</td>
<td>152.14</td>
</tr>
<tr>
<td>6095040</td>
<td>dirsnmpd</td>
<td>151.87</td>
<td>58.4</td>
<td>0</td>
<td>151.80</td>
</tr>
<tr>
<td>19136714</td>
<td>IBM.MgmtDomai</td>
<td>150.91</td>
<td>64.8</td>
<td>0</td>
<td>148.96</td>
</tr>
<tr>
<td>4849820</td>
<td>tier1slp</td>
<td>149.68</td>
<td>58.4</td>
<td>0</td>
<td>149.65</td>
</tr>
<tr>
<td>18939966</td>
<td>IBM.HostRMd</td>
<td>145.30</td>
<td>58.4</td>
<td>0</td>
<td>144.59</td>
</tr>
<tr>
<td>7929856</td>
<td>IBM.DRMd</td>
<td>145.02</td>
<td>58.5</td>
<td>0</td>
<td>144.85</td>
</tr>
<tr>
<td>19333204</td>
<td>IBM.ServiceRM</td>
<td>144.89</td>
<td>58.5</td>
<td>0</td>
<td>144.70</td>
</tr>
<tr>
<td>6422730</td>
<td>clcomd</td>
<td>144.86</td>
<td>58.4</td>
<td>0</td>
<td>144.57</td>
</tr>
<tr>
<td>5767408</td>
<td>rpc.statd</td>
<td>144.60</td>
<td>58.4</td>
<td>0</td>
<td>144.52</td>
</tr>
</tbody>
</table>
```
Affinity

- **LOCAL** SRAD, within the same chip, shows as s3
- **NEARSRAD**, within the same node – intra-node, shows as s4
- **FAR** SRAD, on another node – inter-node, shows as s5

- Command is lssrad –av or can look at mpstat –d
- Topas M option shows them as Localdisp%, Neardisp%, Fardisp%
- The further the distance the longer the latency
- Problems you may see
  - SRAD has CPUs but no memory or vice-versa
  - CPU or memory unbalanced
- Note – on single node systems far dispatches are not as concerning
- To correct look at new firmware, entitlements and LPAR memory sizing
- Can also look at Dynamic Platform Optimizer (DPO)

Memory Tips

Avoid having chips without DIMMs.
Attempt to fill every chip’s DIMM slots, activating as needed.
Hypervisor tends to avoid activating cores without “local” memory.

Diagram courtesy of IBM
mpstat –d Example from POWER8

b814aix1: mpstat -d

System configuration: lcpu=48 ent=0.5 mode=Uncapped

<table>
<thead>
<tr>
<th>cpu</th>
<th>cs</th>
<th>ics</th>
<th>bound</th>
<th>rq</th>
<th>push</th>
<th>S3pull</th>
<th>S3grd</th>
<th>S0rd</th>
<th>S1rd</th>
<th>S2rd</th>
<th>S3rd</th>
<th>S4rd</th>
<th>S5rd</th>
<th>ilcs</th>
<th>vlcs</th>
<th>S3hrd</th>
<th>S4hrd</th>
<th>S5hrd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>82340</td>
<td>11449</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>98.9</td>
<td>0.0</td>
<td>0.0</td>
<td>1.1</td>
<td>0.0</td>
<td>0.0</td>
<td>23694</td>
<td>120742</td>
<td>1.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>81</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9488</td>
<td>9541</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>81</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9501</td>
<td>9533</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>82</td>
<td>82</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.2</td>
<td>98.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9515</td>
<td>9876</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>81</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9515</td>
<td>9525</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>81</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9522</td>
<td>9527</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>81</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9522</td>
<td>9518</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>82</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9526</td>
<td>9511</td>
<td>0.0</td>
<td>100.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The above is for a single socket system (S814) so I would expect to see everything local (s3hrd)
On a multi socket or multimode pay attention to the numbers under near and far

# lssrad -av

<table>
<thead>
<tr>
<th>REF1</th>
<th>SRAD</th>
<th>MEM</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>31288.19</td>
<td>0-23</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>229.69</td>
<td></td>
</tr>
</tbody>
</table>
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  We will calculate these
vmo -p -o maxfree=1088  We will calculate these
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

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  pbufs
11173706 paging space I/Os blocked with no psbuf  pagespace
2048 file system I/Os blocked with no fsbuf  JFS
238 client file system I/Os blocked with no fsbuf  NFS/VxFS
39943187 external pager file system I/Os blocked with no fsbuf  JFS2

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
vmstat -v Output

uptime
02:03PM up 39 days, 3:06, 2 users, load average: 17.02, 15.35, 14.27

9 memory pools
3.0 minperm percentage
90.0 maxperm percentage
14.9 numperm percentage
14.9 numclient percentage
90.0 maxclient percentage

66 pending disk I/Os blocked with no pbuf
0 paging space I/Os blocked with no psbuf
1972 filesystem I/Os blocked with no fsbuf
527 client filesystem I/Os blocked with no fsbuf
613 external pager filesystem 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
This is a fairly healthy system as it has been up 39 days with few blockages

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
- You have to divide the fre column by mempools
- 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.
Calculating minfree and maxfree

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

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.
```

nmon Monitoring

- nmon -ft -AOPV^dMLW -s 15 -c 120
  - Grabs a 30 minute nmon snapshot
  - A is async IO
  - M is mempages
  - t is top processes
  - L is large pages
  - O is SEA on the VIO
  - P is paging space
  - V is disk volume group
  - d is disk service times
  - ^ is fibre adapter stats
  - W is workload manager statistics if you have WLM enabled

If you want a 24 hour nmon use:
```
nmon -ft -AOPV^dMLW -s 150 -c 576
```

May need to enable accounting on the SEA first – this is done on the VIO
```
chdev -dev ent* -attr accounting=enabled
```

Can use entstat/seastat or topas/nmon to monitor – this is done on the vios
topas -E
```
nmon -O
```

VIOS performance advisor also reports on the SEAs
Running nmon from cron

In cron I put:
59 23 * * * /usr/local/bin/runnmon.sh >/dev/null 2>&1

SCRIPT is:
# cat runnmon.sh

#!/bin/ksh
#
cd /usr/local/perf
/usr/bin/nmon -ft AOPV^dMLW -s 150 -c 576
#
#A is async IO
#M is mempages
#t is top processes
#L is large pages
#O is SEA on the VIO
#P is paging space
#V is disk volume group
#d is disk service times
#^ is fibre adapter stats
#W is fibre adapter stats

Using ps –gv to find memory leak

To find memory leak - run the following several times and monitor the size column

```
# ps vg | head -n 1; ps vg | egrep -v "SIZE" | sort +5 -r | head -n 3

PID   TTY STAT  TIME  PGIN SIZE   RSS  LIM  TSIZ  TRS  %CPU  %MEM COMMAND
7471110 - A     0:05  2493  74768  74860 xx   92  0.0  2.0     /var/op
5832806 - A     0:03  2820  55424  55552 xx   128 0.0  2.0     /usr/ja
5243020 - A     0:01  488   31408 31480 xx   47  72  0.0  1.0     [cimserve]
```

sort +5 -r says to sort by column 5 (SIZE) putting largest at the top
Memory Planning

http://www.circle4.com/ptechu/memoryplan.xlsx

Note div 64 is 128 for p7+ and p8

<table>
<thead>
<tr>
<th>Memory Planning Worksheet</th>
<th><a href="http://www.circle4.com/ptechu/memoryplan.xlsx">http://www.circle4.com/ptechu/memoryplan.xlsx</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent 770</td>
<td>Premium 900%</td>
</tr>
<tr>
<td>Max RAM Capacity</td>
<td>RAM Installed</td>
</tr>
<tr>
<td>1 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Change the VM size on this line to match NED on HMC</td>
<td></td>
</tr>
<tr>
<td>1 GB RAM - (MB below in KB)</td>
<td></td>
</tr>
<tr>
<td>- Need to show worst possible</td>
<td></td>
</tr>
<tr>
<td>8 NIMPS (VIO2 per VIO)</td>
<td>16 NIMPS (VIO4 per VIO)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR NAME</th>
<th>Decoded Memory (MB)</th>
<th>Maximum Memory (MB)</th>
<th>Unused RAM (MB)</th>
<th>Round Up (IU)</th>
<th>Actual Memory (MB)</th>
<th>Extra high if NVG (MB)</th>
<th>Port ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIOS1</td>
<td>3172</td>
<td>4096</td>
<td>92</td>
<td>1.25</td>
<td>1</td>
<td>256</td>
<td>8994</td>
</tr>
<tr>
<td>VIOS2</td>
<td>3172</td>
<td>4096</td>
<td>92</td>
<td>1.25</td>
<td>1</td>
<td>256</td>
<td>8994</td>
</tr>
<tr>
<td>LPAR1</td>
<td>120932</td>
<td>28368</td>
<td>256</td>
<td>1.00</td>
<td>1</td>
<td>256</td>
<td>8994</td>
</tr>
<tr>
<td>LPAR2</td>
<td>200234</td>
<td>254570</td>
<td>256</td>
<td>1.00</td>
<td>2</td>
<td>512</td>
<td>175053</td>
</tr>
<tr>
<td>LPAR3</td>
<td>141508</td>
<td>184562</td>
<td>256</td>
<td>1.00</td>
<td>1</td>
<td>256</td>
<td>8994</td>
</tr>
<tr>
<td>LPAR4</td>
<td>175132</td>
<td>175132</td>
<td>256</td>
<td>1.05</td>
<td>3</td>
<td>513</td>
<td>8994</td>
</tr>
<tr>
<td>LPAR5</td>
<td>55564</td>
<td>71680</td>
<td>1120</td>
<td>4.38</td>
<td>5</td>
<td>1280</td>
<td>8994</td>
</tr>
<tr>
<td>LPAR6</td>
<td>107888</td>
<td>107888</td>
<td>576</td>
<td>2.98</td>
<td>6</td>
<td>368</td>
<td>8994</td>
</tr>
</tbody>
</table>

VIOs Monitoring

$ part -?

usage: part [-i INTERVAL] [-f FILENAME] [-t LEVEL] [-h | -?]

- i <minutes> interval can range between 10-60
- f <file> any nmon recording
- t <level> 1 - Basic logging, 2 - Detailed logging
- h or -? usage message

$ part -i 10

part: Reports are successfully generated in b814vio1_150713_13_07_37.tar

Does a 10 minute snap
Creates files in /home/padmin

../performance\part-output\b814vio1_150713_13_07_37\vios_advisor_report.xml
DEMOS if time

Part output
HMC Performance reporting

Thank you for your time

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

Also check out:
http://www.circle4.com/forsythetalks.html
http://www.circle4.com/movies/

Handout at:
Useful Links

- Charlie Cler Articles
- Jaqui Lynch Articles
- Jay Krumke Twitter – chromeaix
  - [https://twitter.com/chromeaix](https://twitter.com/chromeaix)
- Nigel Griffiths Twitter – mr_nmon
  - [https://twitter.com/mr_nmon](https://twitter.com/mr_nmon)
- Gareth Coates Twitter – power_gaz
  - [https://twitter.com/power_gaz](https://twitter.com/power_gaz)
- Jaqui’s Upcoming Talks and Movies
  - Upcoming Talks
  - Movie replays
    - [http://www.circle4.com/movies](http://www.circle4.com/movies)
- IBM US Virtual User Group
  - [http://www.tinyurl.com/ibmaixvug](http://www.tinyurl.com/ibmaixvug)
- Power Systems UK User Group
- AIX Wiki
  - [https://www.ibm.com/developerworks/wikis/display/WikiPtype/AIX](https://www.ibm.com/developerworks/wikis/display/WikiPtype/AIX)
- HMC Scanner
- Workload Estimator
- Performance Tools Wiki
- Performance Monitoring
  - [https://www.ibm.com/developerworks/wikis/display/WikiPtype/Performance+Monitoring+Documentation](https://www.ibm.com/developerworks/wikis/display/WikiPtype/Performance+Monitoring+Documentation)
- Other Performance Tools
  - Includes new advisors for Java, VIOS, Virtualization
- VIOS Advisor
  - [https://www.ibm.com/developerworks/wikis/display/WikiPtype/Other+Performance+Tools#OtherPerformanceTools-VIOSPA](https://www.ibm.com/developerworks/wikis/display/WikiPtype/Other+Performance+Tools#OtherPerformanceTools-VIOSPA)
References

- Simultaneous Multi-Threading on POWER7 Processors by Mark Funk
- Processor Utilization in AIX by Saravanan Devendran
- SG24-7940 - PowerVM Virtualization - Introduction and Configuration
- SG24-7590 – PowerVM Virtualization – Managing and Monitoring
- SG24-8171 – Power Systems Performance Optimization
- Redbook Tip on Maximizing the Value of P7 and P7+ through Tuning and Optimization
Starter set of tunables 2

*Explanations for these will be covered in the IO presentation*

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

**PBUFS**
Use the new way

**JFS2**

ioo -p -o j2_maxPageReadAhead=128
  (default above may need to be changed for sequential) – dynamic
  Difference between minfree and maxfree should be > that this value

j2_dynamicBufferPreallocation=16
  Max is 256. 16 means 16 x 16k slabs or 256k
  Default that may need tuning but is dynamic
  Replaces tuning j2_nBufferPerPagerDevice until at max.

*Network changes in later slide*