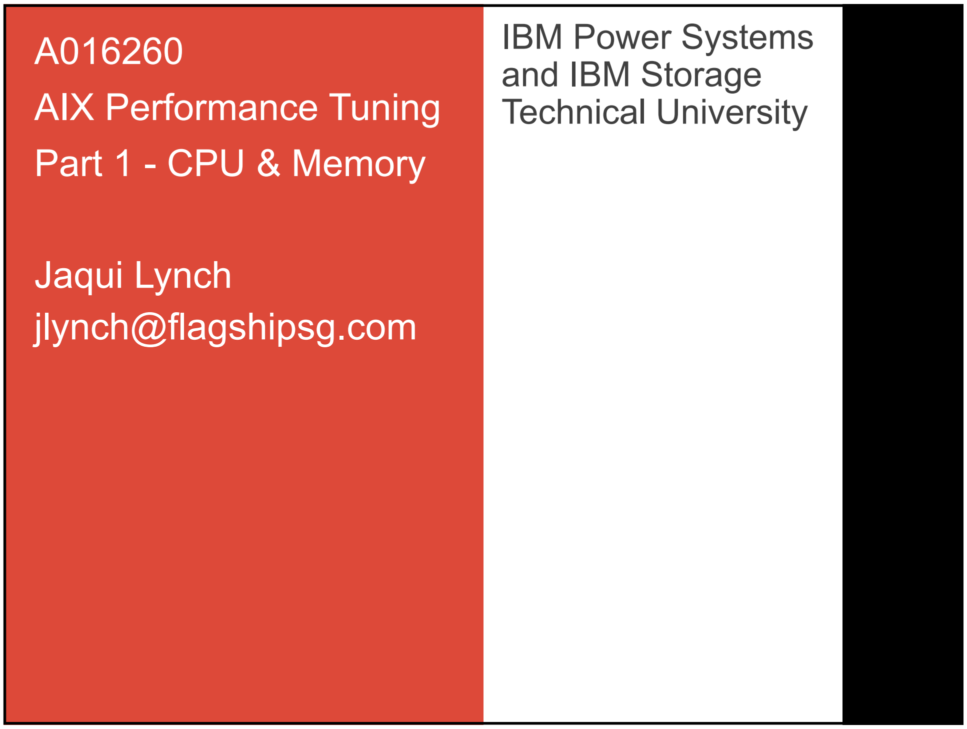
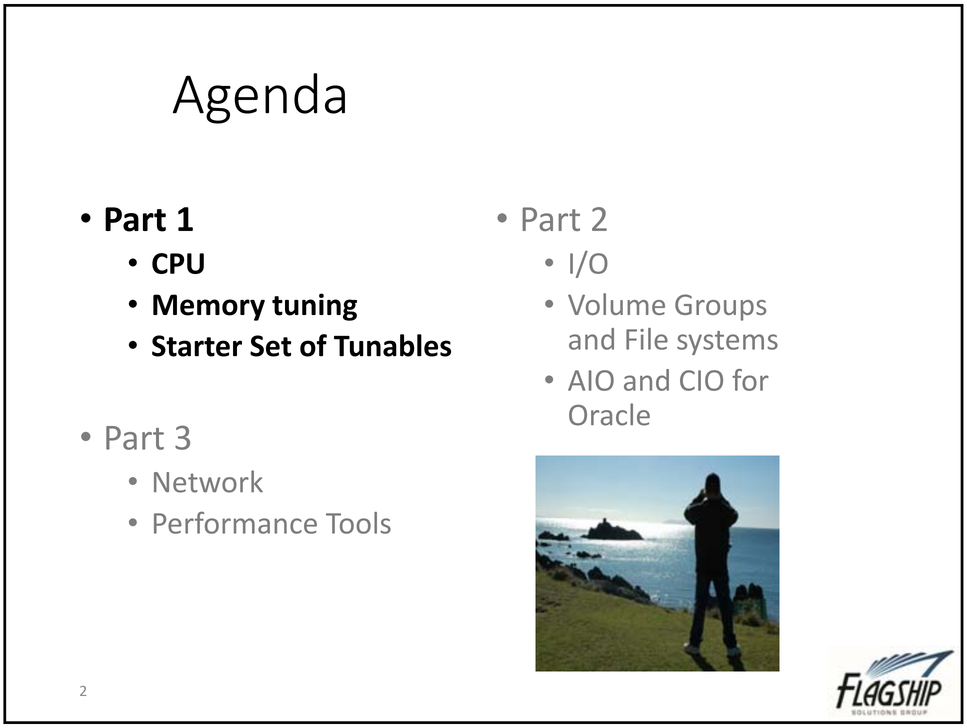


A016260
AIX Performance Tuning
Part 1 - CPU & Memory

Jaqui Lynch
jlynch@flagshipsg.com

IBM Power Systems and IBM Storage Technical University



Agenda

- **Part 1**
  - **CPU**
  - **Memory tuning**
  - **Starter Set of Tunables**
- Part 3
  - Network
  - Performance Tools
- Part 2
  - I/O
  - Volume Groups and File systems
  - AIO and CIO for Oracle

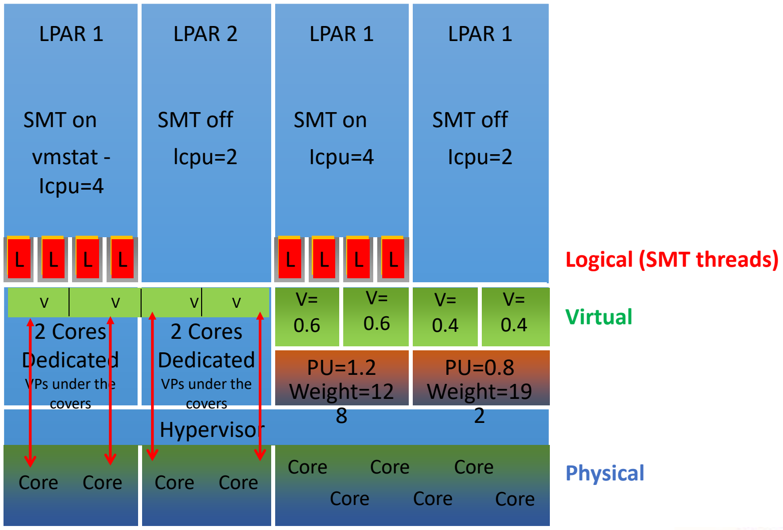
2

FLAGSHIP
SOLUTIONS GROUP

CPU

Logical Processors

Logical Processors represent SMT threads



LPAR 1	LPAR 2	LPAR 1	LPAR 1
SMT on vmstat - lcpu=4	SMT off lcpu=2	SMT on lcpu=4	SMT off lcpu=2
L L L L		L L L L	
V V	V V	V=0.6 V=0.6	V=0.4 V=0.4

2 Cores Dedicated
VPs under the covers
	2 Cores Dedicated
	VPs under the covers
		PU=1.2
		Weight=128
			PU=0.8
			Weight=192

Hypervisor

Core Core	Core Core	Core Core Core Core Core Core

Logical (SMT threads)

Virtual

Physical

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

5

Understand SMT

- 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

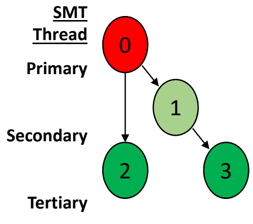
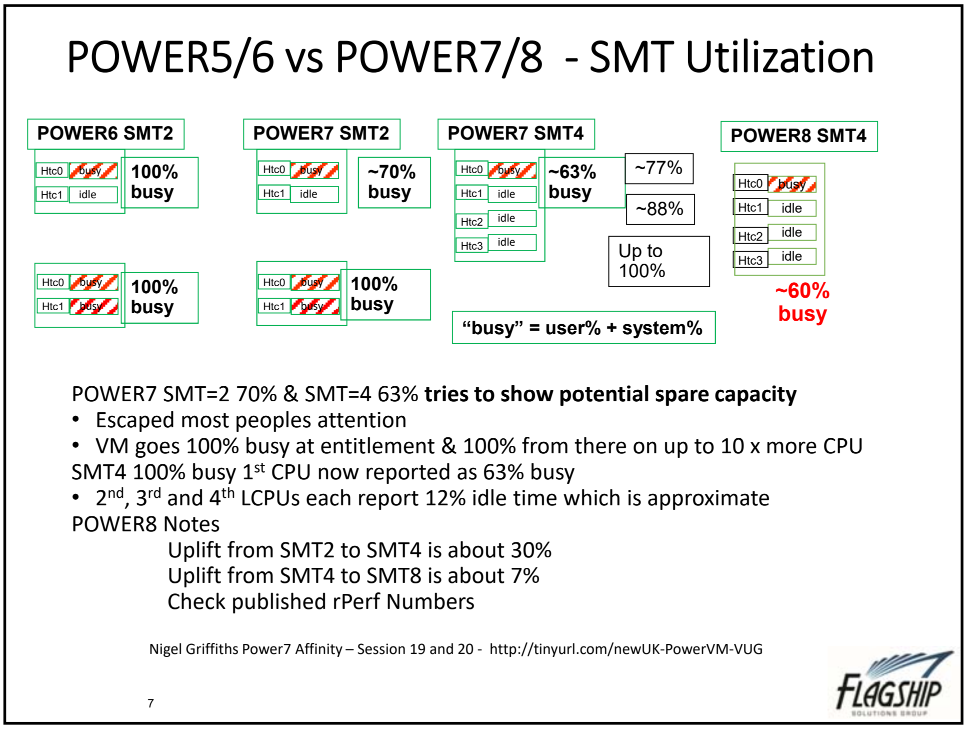


Diagram courtesy of IBM

6



POWER5/6 vs POWER7/8 - SMT Utilization

POWER6 SMT2
Htc0 busy
Htc1 idle
100% busy
Htc0 busy
Htc1 busy
100% busy
POWER7 SMT2
Htc0 busy
Htc1 idle
~70% busy
Htc0 busy
Htc1 busy
100% busy
POWER7 SMT4
Htc0 busy
Htc1 idle
Htc2 idle
Htc3 idle
~63% busy
~77%
~88%
Up to 100%
"busy" = user% + system%
POWER8 SMT4
Htc0 busy
Htc1 idle
Htc2 idle
Htc3 idle
~60% busy

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

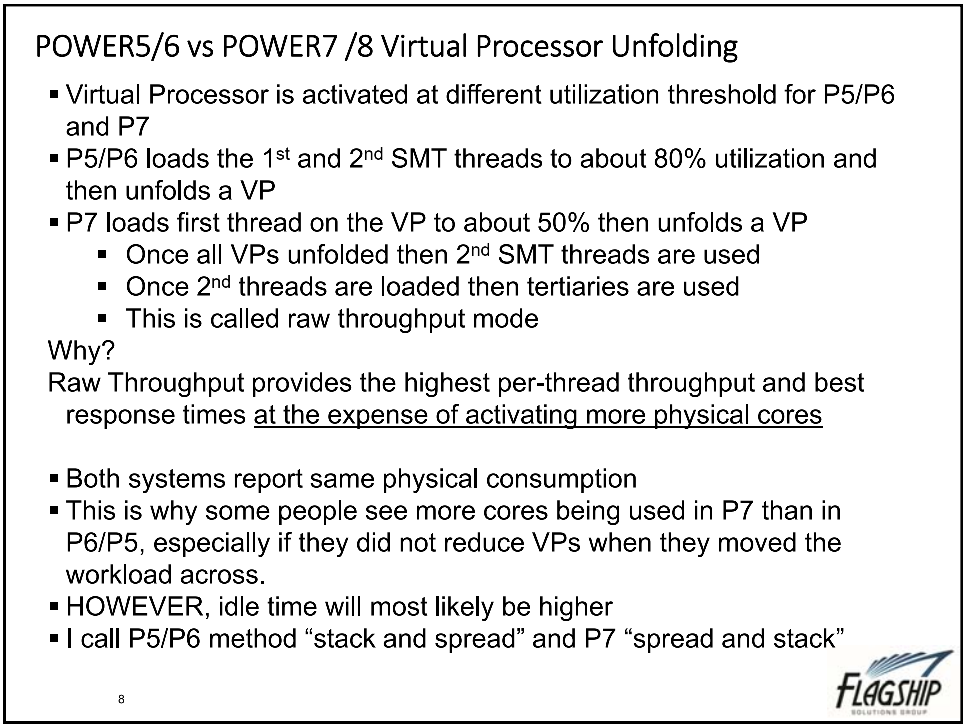
- 2nd, 3rd and 4th LCPUs each report 12% idle time which is approximate

POWER8 Notes

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

Nigel Griffiths Power7 Affinity – Session 19 and 20 - http://tinyurl.com/newUK-PowerVM-VUG

7



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 tertiaries 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"

8

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

Show VP Status on POWER8

```
echo vpm | kdb

VSD Thread State.
 CPU  CPPR VP_STATE FLAGS SLEEP_STATE  PROD_TIME: SECS
NSECS     CEDE_LAT

  0    0  ACTIVE     0 AWAKE       0000000000000000  00000000  00
  1   255  ACTIVE     0 AWAKE       00000000554BA05B  38E6945F  00
  2   255  ACTIVE     0 AWAKE       00000000554BA05B  38E72B44  00
  3   255  ACTIVE     0 AWAKE       00000000554BA05B  38E7C250  00
  4    0  DISABLED    0 AWAKE       0000000000000000  00000000  00
  5    0  DISABLED    0 AWAKE       0000000000000000  00000000  02
  6    0  DISABLED    0 AWAKE       0000000000000000  00000000  02
  7    0  DISABLED    0 AWAKE       0000000000000000  00000000  02
  8    0  DISABLED    0 AWAKE       0000000000000000  00000000  00
  9   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB1B4A  02
 10   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB16A8  02
 11   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB1CEC  02
 12   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB1806  02
 13   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB1ED6  02
 14   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB164B  02
 15   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB1ABF  02
 16    0  DISABLED    0 AWAKE       0000000000000000  00000000  02
 17   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB51EA  02
 18   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB4C01  02
 19   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB52F0  02
 20   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB4DCA  02
 21   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB5765  02
 22   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB4F79  02
 23   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB521F  02
```

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

```
CPU  CPPR VP_STATE FLAGS SLEEP_STATE  PROD_TIME: SECS   NSECS
CEDE_LAT

 24   11  ACTIVE     0 SLEEPING    00000000554BA0A9  33BB6FB9  00
 25   11  ACTIVE     0 SLEEPING    00000000554BA0A9  33BB7209  00
 26   11  ACTIVE     0 SLEEPING    00000000554BA0A9  33BB744B  00
 27   11  ACTIVE     0 SLEEPING    00000000554BA0A9  33BB75A3  00
 28   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB75BC  02
 29   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB78EB  02
 30   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB6C3D  02
 31   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB6CD3  02
 32   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBB1C3  02
 33   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBB44E  02
 34   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBB53E  02
 35   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBB746  02
 36   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBAA43  02
 37   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBAA13  02
 38   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBAD66  02
 39   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BBAFC2  02
 40   11  DISABLED    0 SLEEPING    00000000554BA0A7  2DC515C8  02
 41   11  DISABLED    0 SLEEPING    00000000554BA0A7  2DC51557  02
 42   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB28B2  02
 43   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB2A48  02
 44   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB21FB  02
 45   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB23B2  02
 46   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB2E61  02
 47   11  DISABLED    0 SLEEPING    00000000554BA0A9  33BB371D  02
```

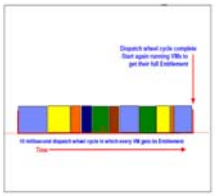
More on Dispatching

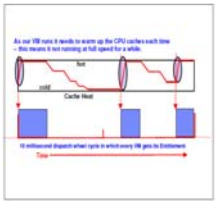
How dispatching works

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


Dispatch wheel cycle complete
Start again running VMs to
get their full Entitlement
10 millisecond dispatch wheel cycle in which every VM gets its Entitlement
Time


As our VM runs it needs to warm up the CPU caches each time
– this means it not running at full speed for a while.
hot
cold
Cache Heat
10 millisecond dispatch wheel cycle in which every VM gets its Entitlement
Time

Nigel Griffiths Power7 Affinity – Session 19 and 20 - http://tinyurl.com/newUK-PowerVM-VUG

Entitlement and VPs

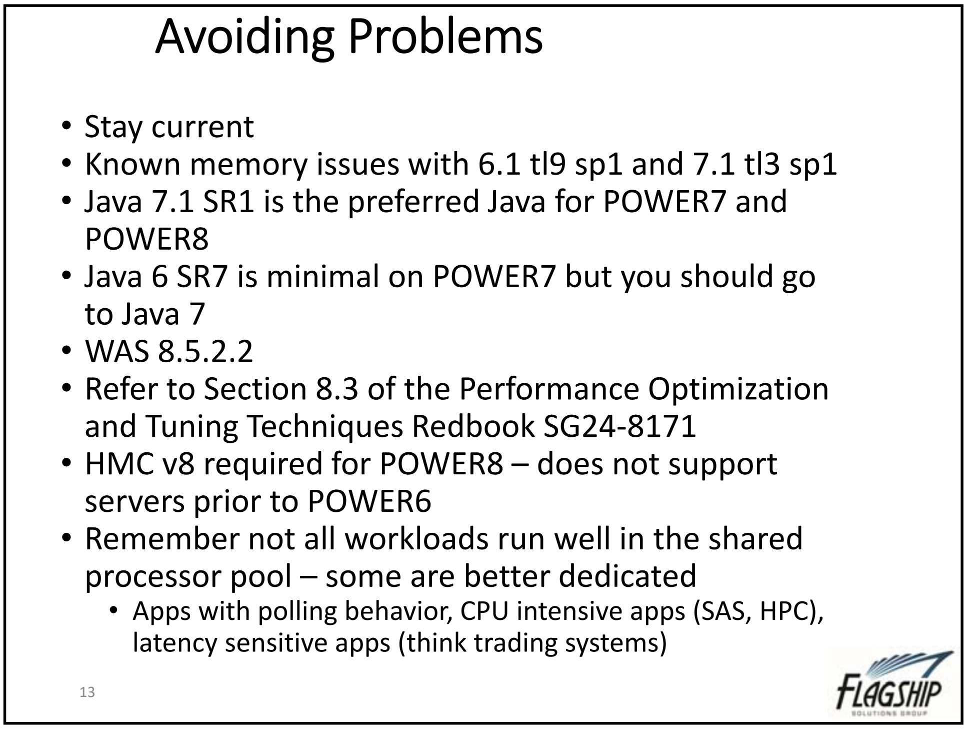
- Utilization calculation for CPU is different between POWER5, 6 and POWER7
- VPs are also unfolded sooner (at lower utilization levels than on P6 and P5)
- May also see high VCSW in lparstat

- 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**

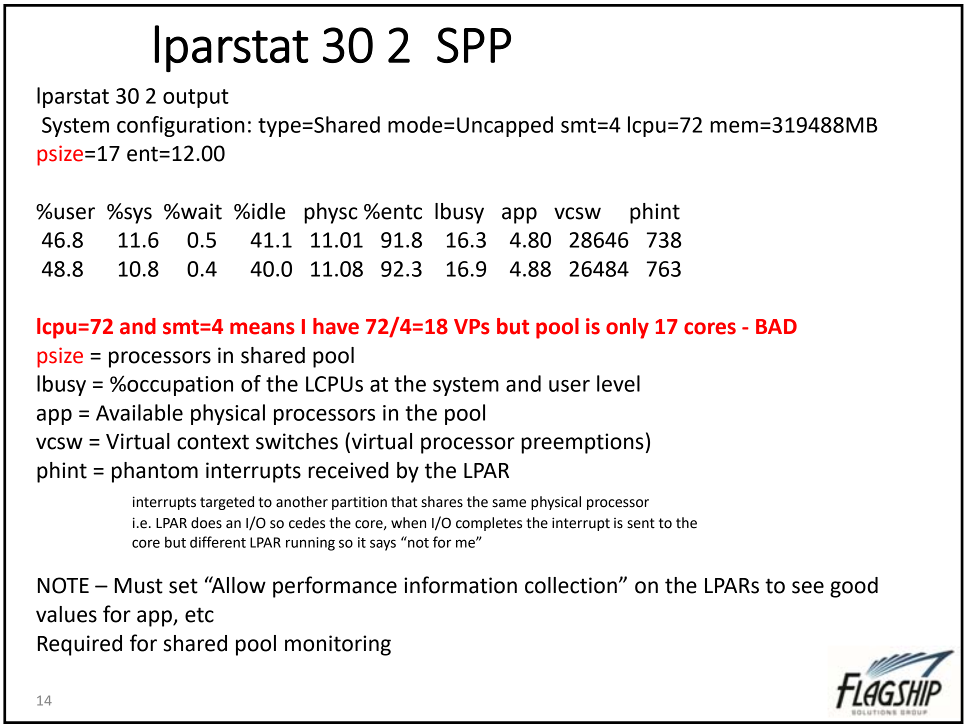
  - All VPs have to be dispatched before one can be redispatched

  - **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**



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)



lparstat 30 2 SPP

```
lparstat 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   vcsw    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

vcsw = Virtual context switches (virtual processor preemptions)

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

lparstat 30 2 Dedicated

```
lparstat 30 2 output
System configuration: type=Dedicated mode=Capped smt=4 lcpu=80 mem=524288MB

%user    %sys    %wait    %idle
-----    -----   ------   ------
 16.8    28.7    6.4      48.1
 17.0    29.3    5.8      48.0
```

lcpu=4 and smt=80 means I have 80/4=20 cores

lbusy = %occupation of the LCPUs at the system and user level

```
lparstat -h 30 2 output
System configuration: type=Dedicated mode=Capped smt=4 lcpu=80 mem=524288MB

%user    %sys    %wait    %idle    %hypv    hcalls
-----    -----   ------   ------   ------   ------
  16.8   29.8     5.4     48.0     61.3     2222545
 16.8    30.1     5.1     48.0     61.1     2258600
```

%hypv
  Indicates the percentage of physical processor consumption spent making hypervisor calls.
hcalls
  Indicates the average number of hypervisor calls that were started.

15

Using sar –mu -P ALL (Power7 & SMT4)

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

```
System configuration: lcpu=64 ent=10.00 mode=Uncapped
14:24:31         cpu   %usr   %sys   %wio   %idle   physc   %entc
Average           0     77     22      0       1    0.52     5.2
                  1     37     14      1      48    0.18     1.8
                  2      0      1      0      99    0.10     1.0
                  3      0      1      0      99    0.10     1.0          .9 physc

                  4     84     14      0       1    0.49     4.9
                  5     42      7      1      50    0.17     1.7
                  6      0      1      0      99    0.10     1.0
                  7      0      1      0      99    0.10     1.0          .86 physc

                  8     88     11      0       1    0.51     5.1
                  9     40     11      1      48    0.18     1.8
                  ............. Lines for 10-62 were here
                 63      0      1      0      99    0.11     1.1
                  -     55     11      0      33   12.71   127.1     Above entitlement on average
– increase entitlement?
```

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)

16

mpstat -s

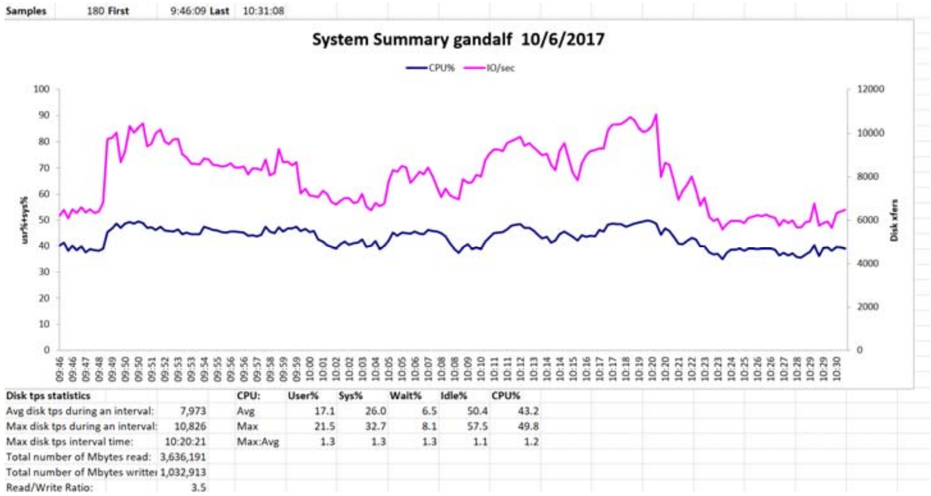
```
mpstat –s 1 1
System configuration: lcpu=64 ent=10.0 mode=Uncapped

          Proc0                          Proc4                       Proc8
          89.06%                         84.01%                      81.42%
 cpu0    cpu1      cpu2    cpu3      cpu4    cpu5     cpu6     cpu7      cpu8    cpu9      cpu10   cpu11
 41.51%  31.69%   7.93%   7.93%      42.41%  24.97%   8.31%   8.32%      39.47%  25.97%   7.99%   7.99%

          Proc12                         Proc16                      Proc20
          82.30%                         38.16%                      86.04%
 cpu12   cpu13     cpu14   cpu15     cpu16   cpu17    cpu18   cpu19      cpu20   cpu21    cpu22   cpu23
 43.34%  22.75%   8.11%   8.11%      23.30%  4.97%    4.95%   4.94%      42.01%  27.66%   8.18%   8.19%

……………………………….

          Proc60
          99.11%
 cpu60   cpu61   cpu62   cpu63
 62.63%  13.22%  11.63%  11.63%
```

shows breakdown across the VPs (proc*) and smt threads (cpu*)

Proc* are the virtual CPUs
CPU* are the logical CPUs (SMT threads)

17

nmon Summary



Samples	180 First	9:46:09 Last	10:31:08

Disk tps statistics		CPU:	User%	Sys%	Wait%	Idle%	CPU%
Avg disk tps during an interval:	7,973	Avg	17.1	26.0	6.5	50.4	43.2
Max disk tps during an interval:	10,826	Max	21.5	32.7	8.1	57.5	49.8
Max disk tps interval time:	10:20:21	Max:Avg	1.3	1.3	1.3	1.1	1.2
Total number of Mbytes read:	3,636,191						
Total number of Mbytes writter	1,032,913						
Read/Write Ratio:	3.5						

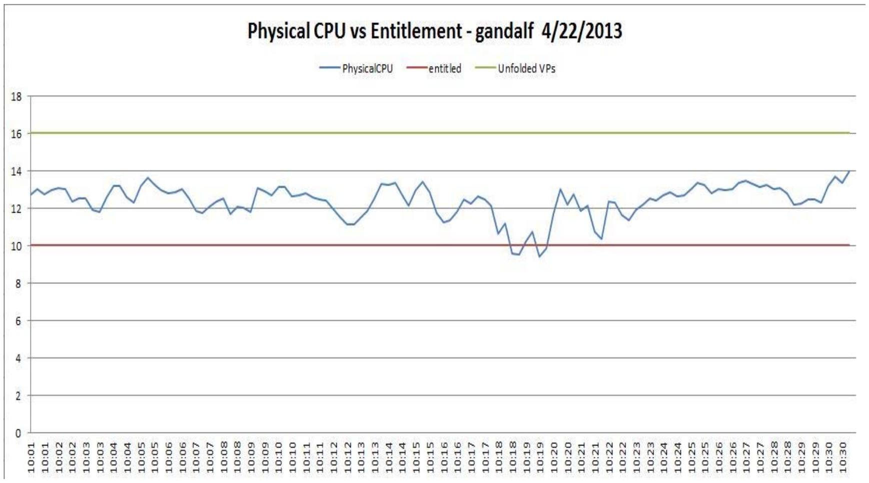
18

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

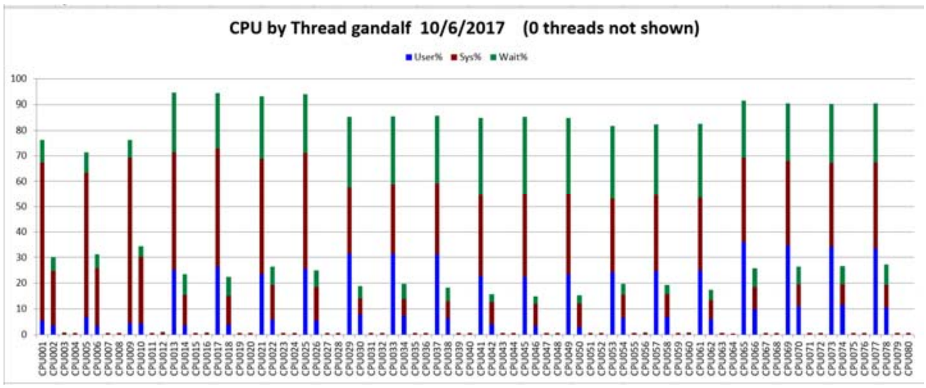
Compare VPs to poolsize
LPAR should not have more VPs than the poolsize

Entitlement and vps from lpar tab in nmon



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 cores/VPs

vmstat -IW

bnim: vmstat -IW 2 2

vmstat –IW 60 2

```
System configuration: lcpu=12 mem=24832MB ent=2.00
   kthr       memory             page              faults           cpu
----------- ----------- ------------------------ ------------ -----------------------
 r  b  p  w   avm       fre         fi  fo  pi  po  fr  sr  in         sy      cs     us  sy  id  wa   pc     ec
 3  1  0  2 2708633 2554878    0  46   0   0   0    0   3920 143515 10131   26  44  30  0    2.24  112.2
 6  1  0  4 2831669 2414985 348  28   0   0   0    0   2983 188837   8316   38  39  22  0    2.42  120.9
```

Note pc=2.42 is 120.0% of entitlement

When looking at system time to user time ratios – remember on a VIO server that high system time is most likely normal as the VIO handles all the I/O and network and really has little normal user type work

-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)

vmstat –IW

	lcpu	**72**	**Mem**	**319488MB**	**Ent**	**12**															
	r	**b**	**p**	**w**	**avm**	**fre**	**fi**	**fo**	**pi**	**po**	**fr**	**sr**	**in**	**sy**	**cs**	**us**	**sy**	**id**	**wa**	**pc**	**ec**
	17	0	0	2	65781580	2081314	2	5	0	0	0	0	11231	146029	22172	52	13	36	0	11.91	99.2
	11	0	0	5	65774009	2088879	2	136	0	0	0	0	11203	126677	20428	47	13	40	0	11.29	94.1
	14	0	0	4	65780238	2082649	2	88	0	0	0	0	11463	220780	22024	50	14	36	0	12.62	105.2
	20	0	0	9	65802196	2060643	51	114	0	0	0	0	9434	243080	21883	55	15	29	1	12.98	108.2
	26	1	0	7	65810733	2052105	9	162	0	0	0	0	9283	293158	22779	60	16	23	1	13.18	109.8
	18	1	0	5	65814822	2048011	0	0	0	0	0	0	11506	155344	27308	53	13	33	0	13.51	112.6
	18	0	0	5	65798165	2064666	0	0	0	0	0	0	10868	200123	25143	53	14	32	0	13.27	110.6
	17	0	0	5	65810136	2052662	4	0	0	0	0	0	12394	230802	29167	51	17	32	1	14.14	117.8
	15	0	0	7	65796659	2066142	0	0	0	0	0	0	12301	217839	27142	48	16	35	0	13.42	111.9
	13	0	0	8	65798332	2064469	1	20	0	0	0	0	14001	160576	30871	44	16	39	0	11.41	95.1
	15	1	0	4	65795292	2067486	7	212	0	0	0	0	14263	215226	31856	51	14	35	0	13.33	111.1
	19	0	0	7	65807317	2055454	0	0	0	0	0	0	11887	306416	26162	52	13	34	0	13.85	115.4
	13	0	0	5	65807079	2055689	0	0	0	0	0	0	11459	196036	26782	49	14	36	0	12.49	104.1
	19	0	0	5	65810475	2052293	0	0	0	0	0	0	13187	292694	28050	52	13	35	0	14	116.7
	17	0	0	11	65819751	2043008	4	0	0	0	0	0	12218	225829	27516	51	14	35	0	13.14	109.5
	26	0	0	10	65825374	2037373	1	35	0	0	0	0	11447	220479	23273	55	13	32	0	13.91	115.9
	26	0	0	6	65820723	2042005	6	182	0	0	0	0	11652	331234	26888	63	11	25	1	14.3	119.2
	18	1	0	6	65816444	2046275	4	0	0	0	0	0	11628	184413	25634	51	14	35	0	13.19	110
	20	0	0	8	65820880	2041819	0	13	0	0	0	0	12332	190716	28370	51	14	36	0	13.37	111.4
	17	0	0	6	65822872	2039836	0	0	0	0	0	0	13269	128353	30880	47	14	39	0	11.46	95.5
	15	0	0	5	65832214	2030493	0	0	0	0	0	0	12079	207403	26319	51	13	35	0	13.24	110.4
	14	0	0	8	65827065	2035639	17	14	0	0	0	0	14060	117935	32407	48	15	36	0	12.06	100.5
	15	0	0	4	65824658	2037996	10	212	0	0	0	0	12690	137533	27678	44	20	36	0	13.53	112.8
	18	0	0	10	65817327	2045339	0	0	0	0	0	0	12665	161261	28010	50	14	36	0	12.69	105.8
	17	0	0	8	65820348	2042321	0	0	0	0	0	0	14047	228897	28475	53	13	34	0	14.44	120.4
	16	0	0	6	65817053	2045609	0	0	0	0	0	0	12953	160629	26652	50	14	35	0	12.83	106.9
	18	0	0	12	65813683	2048949	0	0	0	0	0	0	11766	198577	26593	53	13	33	0	13.54	112.9
	18	0	0	13	65808798	2053853	18	23	0	0	0	0	12195	209122	27152	53	14	33	0	13.86	115.5
	12	1	0	14	65800471	2062164	6	218	0	0	0	0	12429	182117	27787	55	13	31	1	13.38	111.5
	18	2	0	8	65805624	2056998	6	72	0	0	0	0	12134	209260	25250	54	13	32	0	13.73	114.4
	r	**b**	**p**	**w**	**avm**	**fre**	**fi**	**fo**	**pi**	**po**	**fr**	**sr**	**in**	**sy**	**cs**	**us**	**sy**	**id**	**wa**	**pc**	**ec**
Average	17.33	0.23	0.00	7.10	65809344	2053404	5.00	50	0.00	0.00	0.00	0.00	12134.80	203285	26688	51.53	14.10	33.93	0.17	13.14	109.48
Max	26.00	2.00	0.00	14.00	65832214	2088879	51.00	218	0.00	0.00	0.00	0.00	14263.00	331234	32407	63.00	20.00	40.00	1.00	14.44	120.40
Min	11.00	0.00	0.00	2.00	65774009	2030493	0.00	0	0.00	0.00	0.00	0.00	9283.00	117935	20428	44.00	11.00	23.00	0.00	11.29	94.10

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

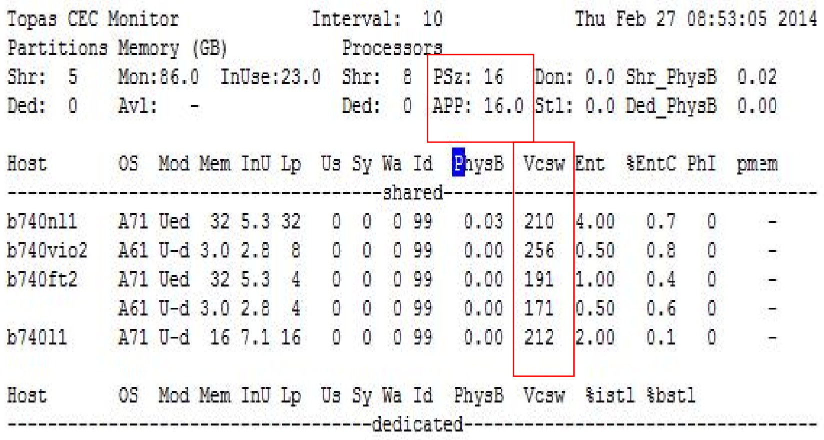
To the right of PoolCPUs there is an unused column which is the number of free pool cores

nmon analyser LPAR Tab

lparstat

Shows the app column and poolsize

topas -C

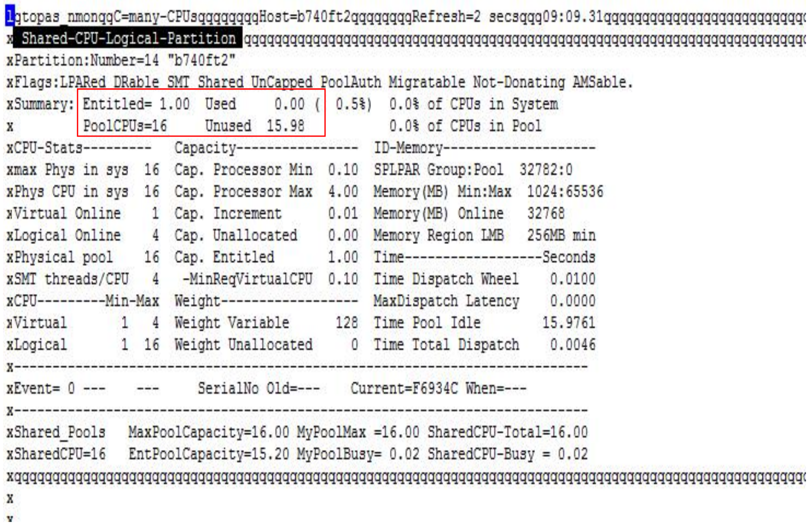


```
Topas CEC Monitor              Interval:  10            Thu Feb 27 08:53:05 2014
Partitions Memory (GB)            Processors
Shr:   5     Mon:86.0  InUse:23.0  Shr:   8  PSz: 16     Don: 0.0 Shr_PhysB  0.02
Ded:   0     Avl:    -             Ded:   0  APP: 16.0  Stl: 0.0 Ded_PhysB  0.00

Host       OS  Mod Mem InU Lp  Us Sy Wa Id  PhysB  Vcsw Ent  %EntC PhI  pmem
-------------------------------------shared-------------------------------------
b740nl1    A71 Ued  32 5.3 32   0  0  0 99   0.03  210 4.00   0.7   0     -
b740vio2   A61 U-d 3.0 2.8  8   0  0  0 99   0.00  256 0.50   0.8   0     -
b740ft2    A71 Ued  32 5.3  4   0  0  0 99   0.00  191 1.00   0.4   0     -
           A61 U-d 3.0 2.8  4   0  0  0 99   0.00  171 0.50   0.6   0     -
b740l1     A71 U-d  16 7.1 16   0  0  0 99   0.00  212 2.00   0.1   0     -

Host       OS  Mod Mem InU Lp  Us Sy Wa Id  PhysB  Vcsw  %istl %bstl
------------------------------------dedicated-----------------------------------
```

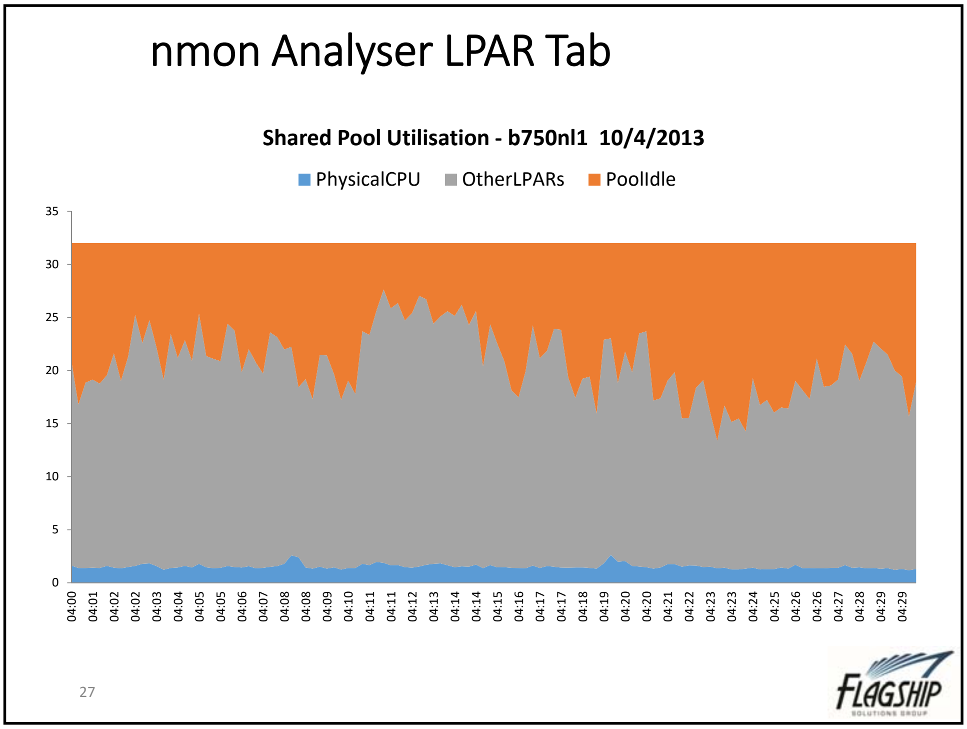
Shows pool size of 16 with all 16 available

Monitor VCSW as potential sign of insufficient entitlement

nmon -p



```
lqtopas_nmonqqC=many-CPUsqqqqqqqqHost=b740ft2qqqqqqqqRefresh=2 secsqqq09:09.31qqqqqqqqqqqqqqqqqqqqqqqqqqq
x Shared-CPU-Logical-Partition qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
xPartition:Number=14 "b740ft2"
xFlags:LPARed DRable SMT Shared UnCapped PoolAuth Migratable Not-Donating AMSable.
xSummary: Entitled= 1.00  Used     0.00 (  0.5%)  0.0% of CPUs in System
x         PoolCPUs=16     Unused  15.98           0.0% of CPUs in Pool
xCPU-Stats---------   Capacity----------------  ID-Memory--------------------
xmax Phys in sys  16  Cap. Processor Min  0.10  SPLPAR Group:Pool  32782:0
xPhys CPU in sys  16  Cap. Processor Max  4.00  Memory(MB) Min:Max  1024:65536
xVirtual Online    1  Cap. Increment      0.01  Memory(MB) Online   32768
xLogical Online    4  Cap. Unallocated    0.00  Memory Region LMB   256MB min
xPhysical pool    16  Cap. Entitled       1.00  Time------------------Seconds
xSMT threads/CPU   4   -MinReqVirtualCPU  0.10  Time Dispatch Wheel    0.0100
xCPU---------Min-Max  Weight------------------  MaxDispatch Latency    0.0000
xVirtual       1   4  Weight Variable      128  Time Pool Idle        15.9761
xLogical       1  16  Weight Unallocated     0  Time Total Dispatch    0.0046
x-------------------------------------------------------------------------
xEvent= 0 ---    ---     SerialNo Old=---    Current=F6934C When=---
x-------------------------------------------------------------------------
xShared_Pools   MaxPoolCapacity=16.00 MyPoolMax =16.00 SharedCPU-Total=16.00
xSharedCPU=16   EntPoolCapacity=15.20 MyPoolBusy= 0.02 SharedCPU-Busy = 0.02
xqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
x
x
```

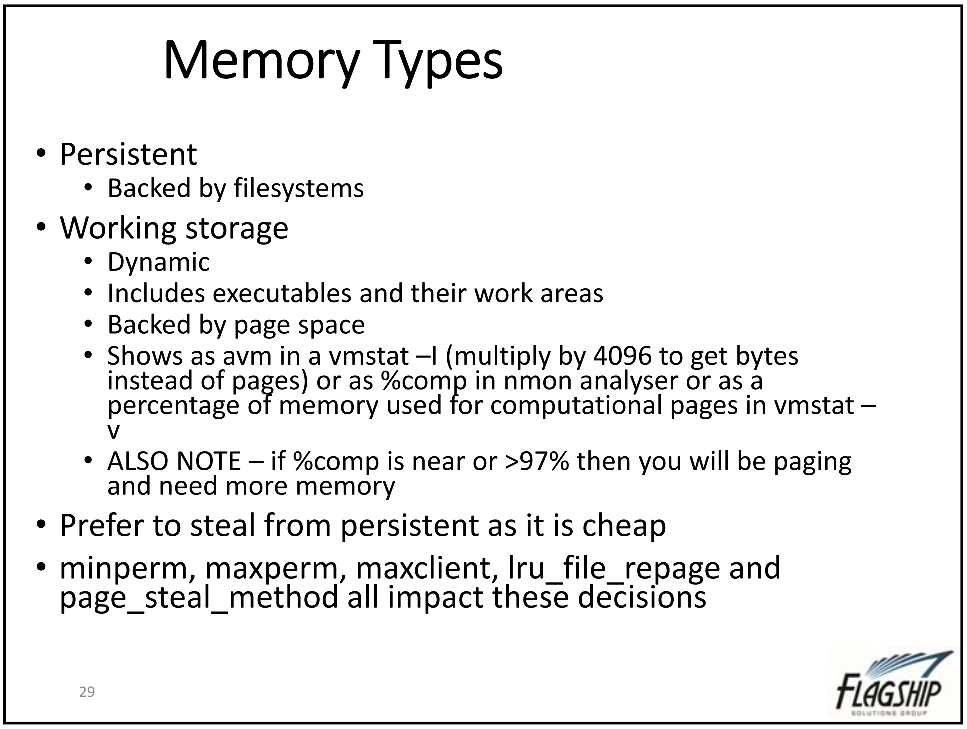


nmon Analyser LPAR Tab

27

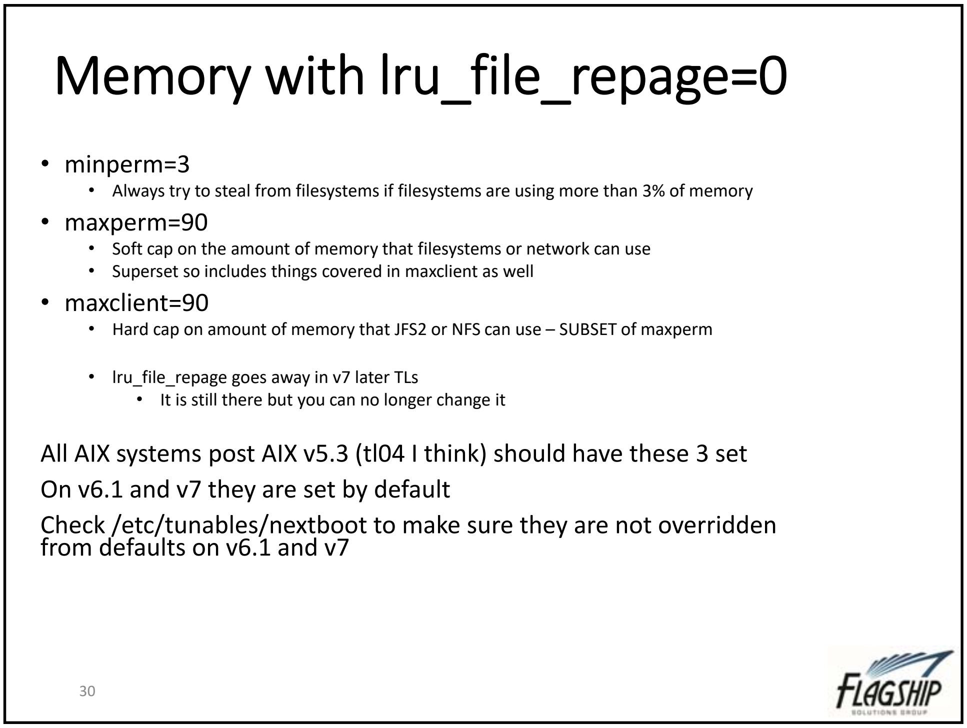
MEMORY

28



Memory Types

- Persistent
  - Backed by filesystems
- Working storage
  - Dynamic
  - Includes executables and their work areas
  - Backed by page space
  - Shows as avm in a vmstat –I (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



Memory with lru_file_repage=0

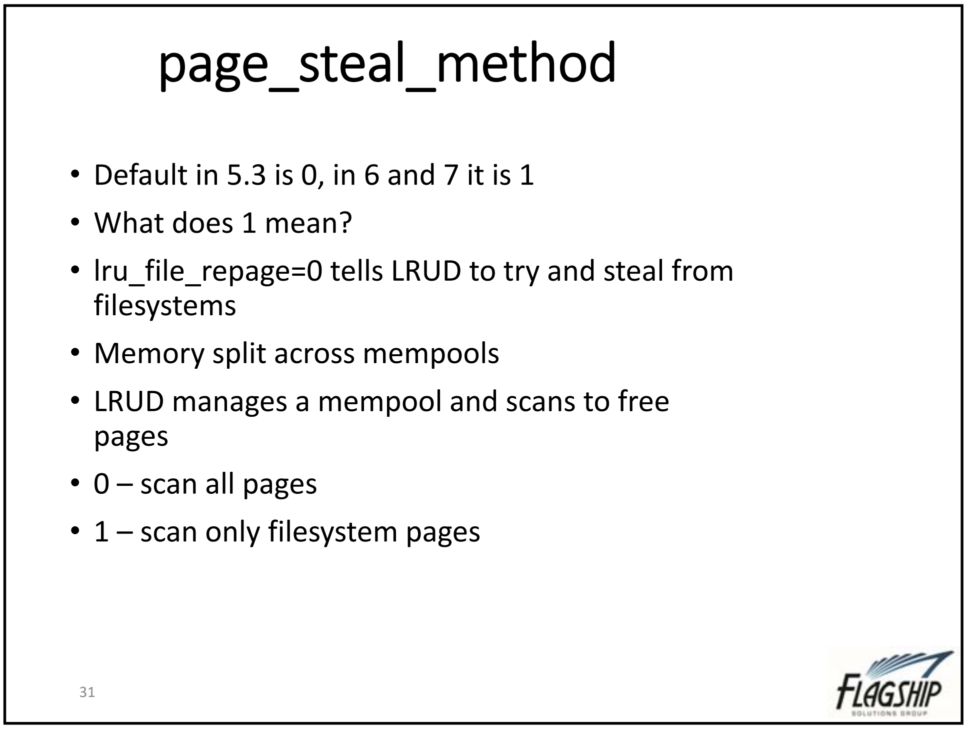
- minperm=3
  - Always try to steal from filesystems if filesystems are using more than 3% of memory
- maxperm=90
  - Soft cap on the amount of memory that filesystems or network can use
  - Superset so includes things covered in maxclient as well
- maxclient=90
  - Hard cap on amount of memory that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

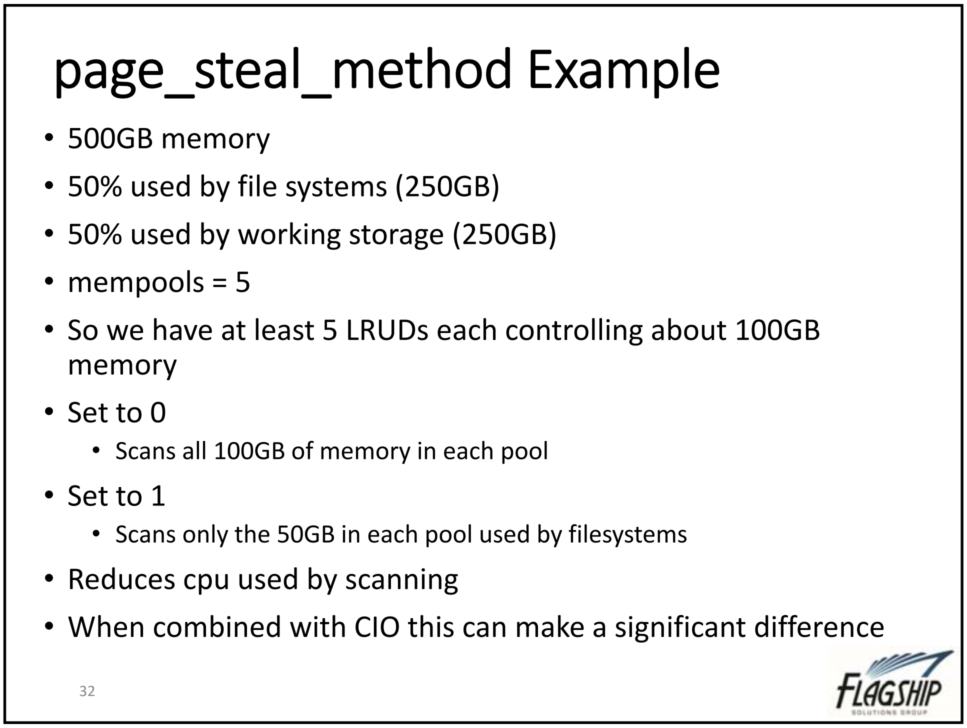
On v6.1 and v7 they are set by default

Check /etc/tunables/nextboot to make sure they are not overridden from defaults on v6.1 and v7



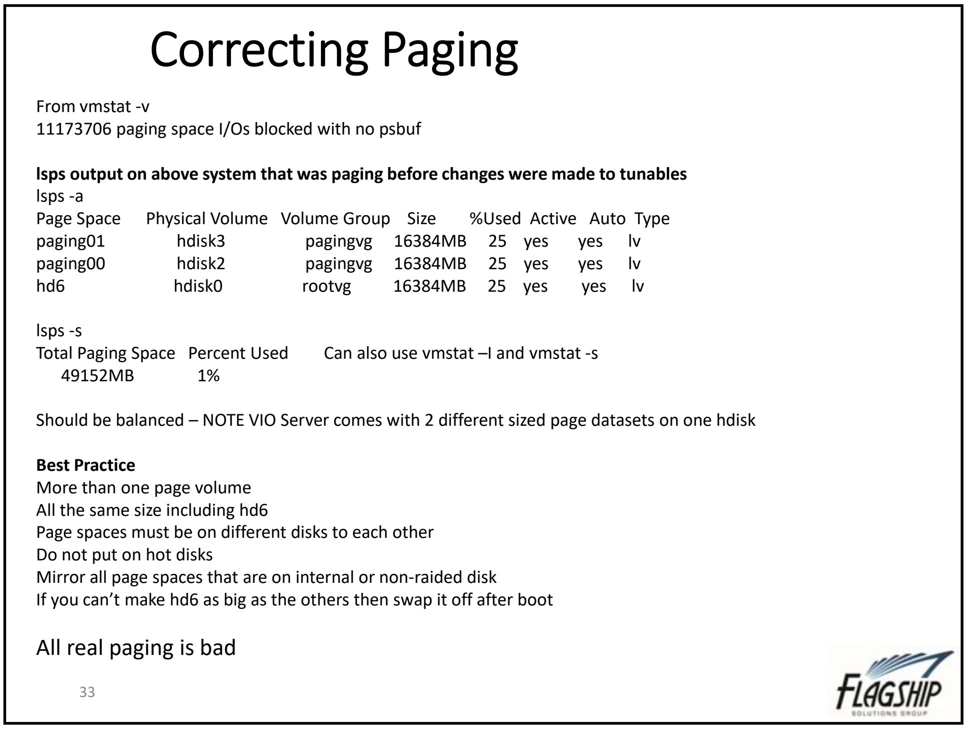
page_steal_method

- 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

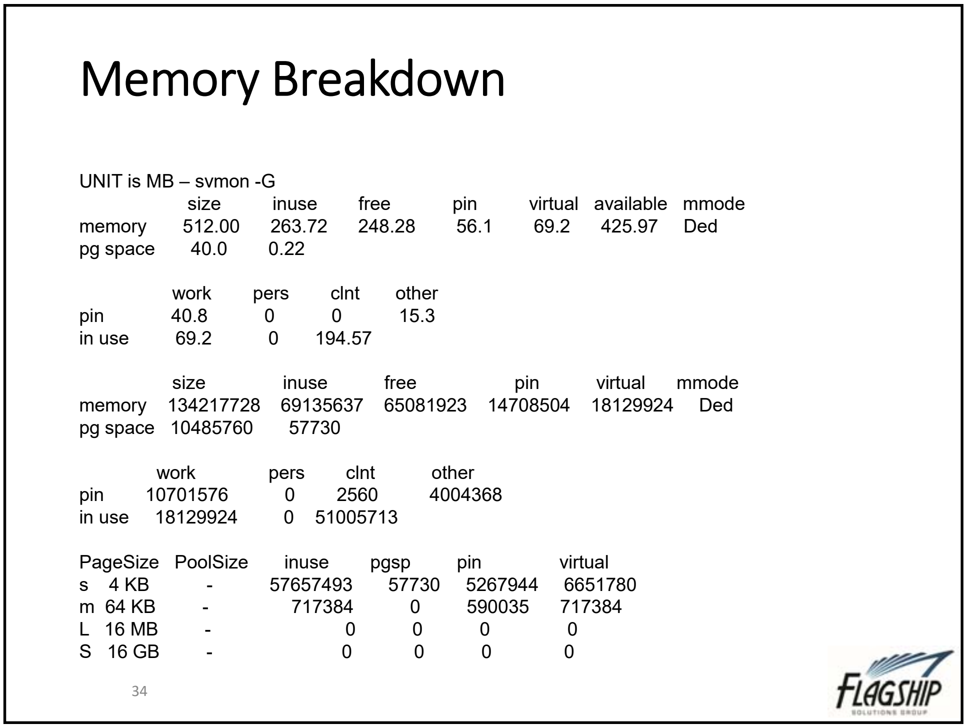
lsps output on above system that was paging before changes were made to tunables

```
lsps -a
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 –I and vmstat -s
     49152MB            1%
```

Should be balanced – NOTE VIO Server comes with 2 different sized page datasets on one hdisk

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



Memory Breakdown

```
UNIT is MB – svmon -G
                   size        inuse        free          pin         virtual   available   mmode
memory       512.00      263.72      248.28        56.1         69.2       425.97      Ded
pg space       40.0        0.22

                work       pers       clnt      other
pin           40.8         0           0          15.3
in use         69.2         0       194.57

                size              inuse          free                pin          virtual      mmode
memory    134217728   69135637   65081923    14708504    18129924     Ded
pg space   10485760       57730

             work               pers       clnt          other
pin        10701576           0        2560         4004368
in use    18129924           0    51005713

PageSize   PoolSize      inuse       pgsp        pin             virtual
s    4 KB          -          57657493      57730     5267944     6651780
m  64 KB          -               717384          0       590035      717384
L   16 MB         -                      0           0           0             0
S   16 GB         -                      0           0           0             0
```

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

35

Memory Problems

- Look at computational memory use
  - Shows as avm in a vmstat –I (multiply by 4096 to get bytes instead of pages)

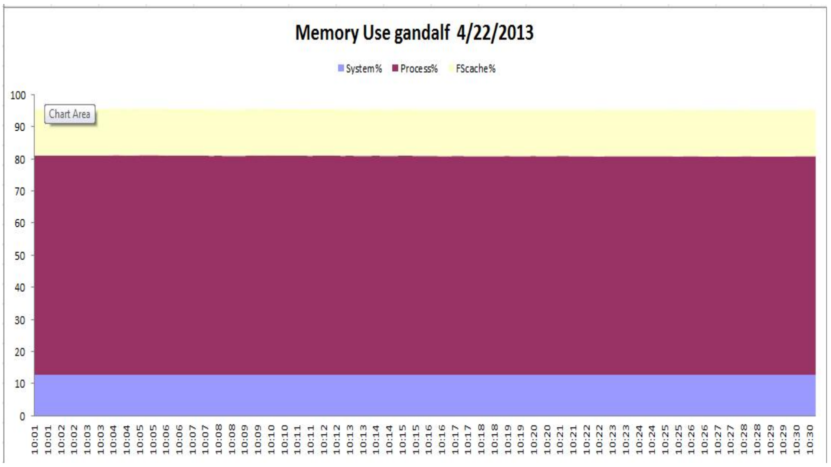
```
• System configuration: lcpu=48 mem=32768MB ent=0.50
•  r b p w  avm       fre          fi fo pi po fr  sr in  sy  cs   us  sy   id  wa   pc   ec
•  0 0 0 0 807668    7546118       0  0  0  0  0   0  1  159 161  0   0   99   0    0.01  1.3
```

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 –Osortseg=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

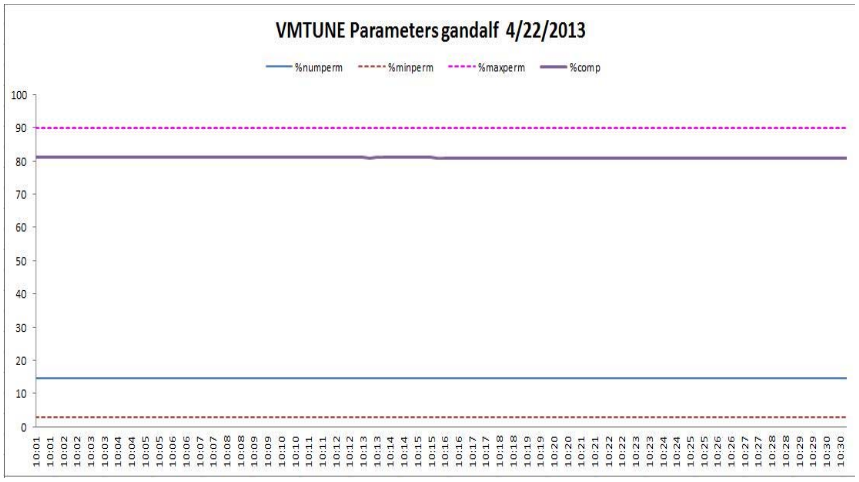
36

nmon memnew tab



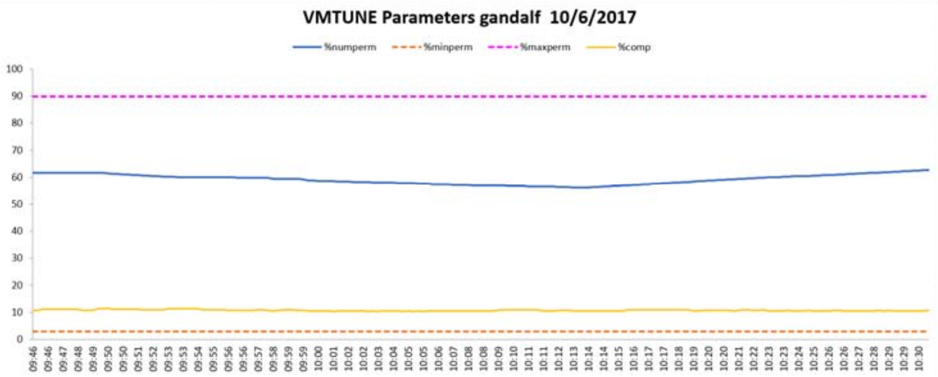
37

nmon memuse tab



38

nmon memuse tab – lots of caching



39

Affinity

- LOCAL SRAD, within the same chip, shows as s3
- NEAR SRAD, 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)

40

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.

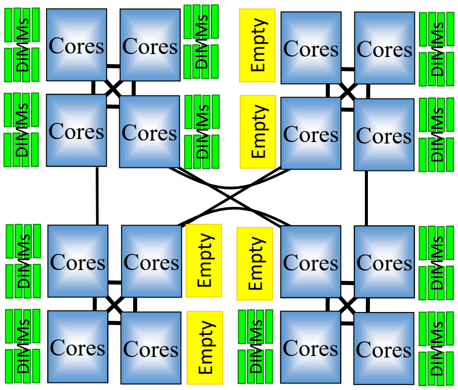


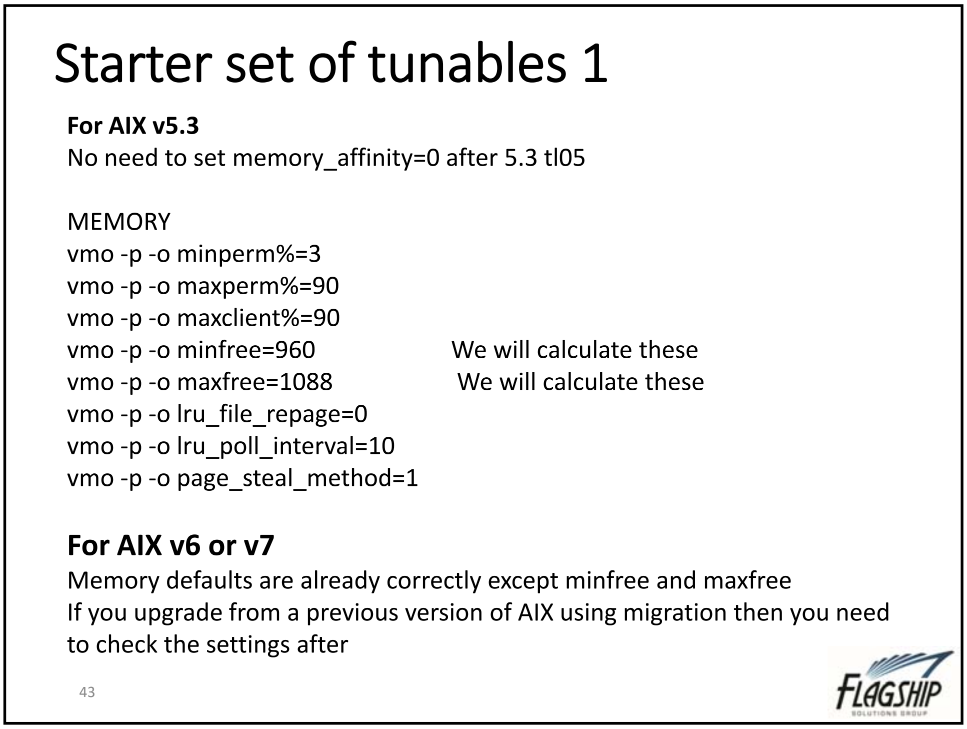
Diagram courtesy of IBM

mpstat –d Example from POWER8

b814aix1: mpstat -d

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

                                                                                        local   near  far
cpu    cs    ics        bound    rq  push S3pull S3grd S0rd    S1rd  S2rd  S3rd  S4rd  S5rd  ilcs    vlcs     S3hrd S4hrd S5hrd
 0  82340  11449         1        2    0    0      0   98.9     0.0   0.0   1.1   0.0   0.0 23694 120742    100.0   0.0   0.0
 1     81     81         0        0    0    0      0    0.0   100.0   0.0   0.0   0.0   0.0  9488   9541    100.0   0.0   0.0
 2     81     81         0        0    0    0      0    0.0   100.0   0.0   0.0   0.0   0.0  9501   9533    100.0   0.0   0.0
 3     82     82         0        0    0    0      0    1.2    98.8   0.0   0.0   0.0   0.0  9515   9876    100.0   0.0   0.0
 4     81     81         0        0    0    0      0    0.0   100.0   0.0   0.0   0.0   0.0  9515   9525    100.0   0.0   0.0
 5     81     81         0        0    0    0      0    0.0   100.0   0.0   0.0   0.0   0.0  9522   9527    100.0   0.0   0.0
 6     81     81         0        0    0    0      0    0.0   100.0   0.0   0.0   0.0   0.0  9522   9518    100.0   0.0   0.0
 7     82     81         0        0    0    0      0    0.0   100.0   0.0   0.0   0.0   0.0  9526   9511    100.0   0.0   0.0
```

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



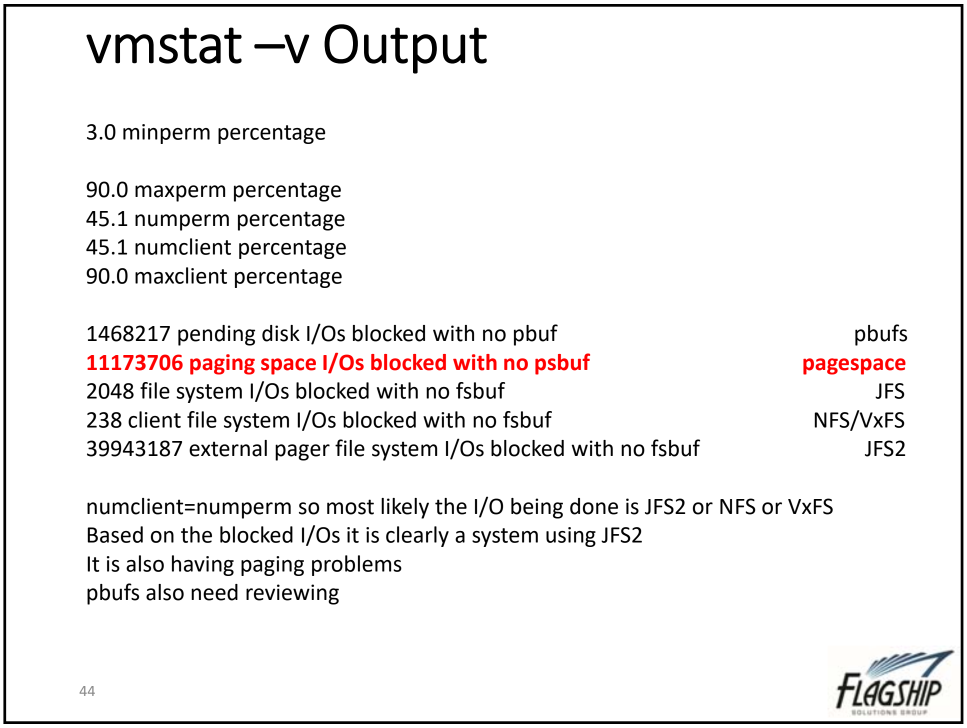
Starter set of tunables 1

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



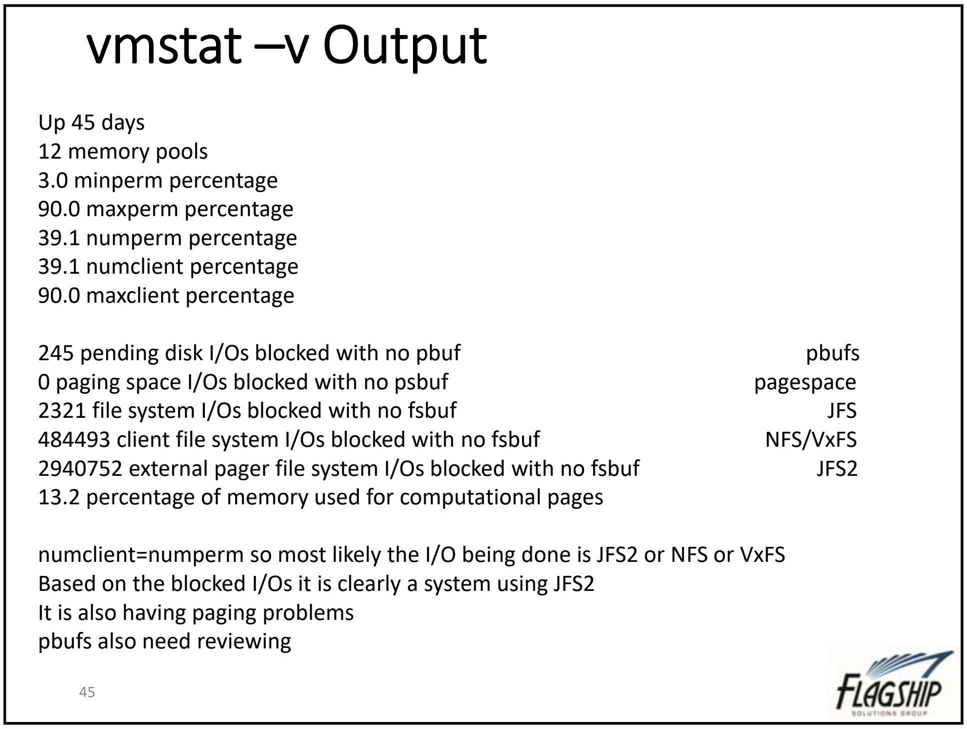
vmstat –v Output

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

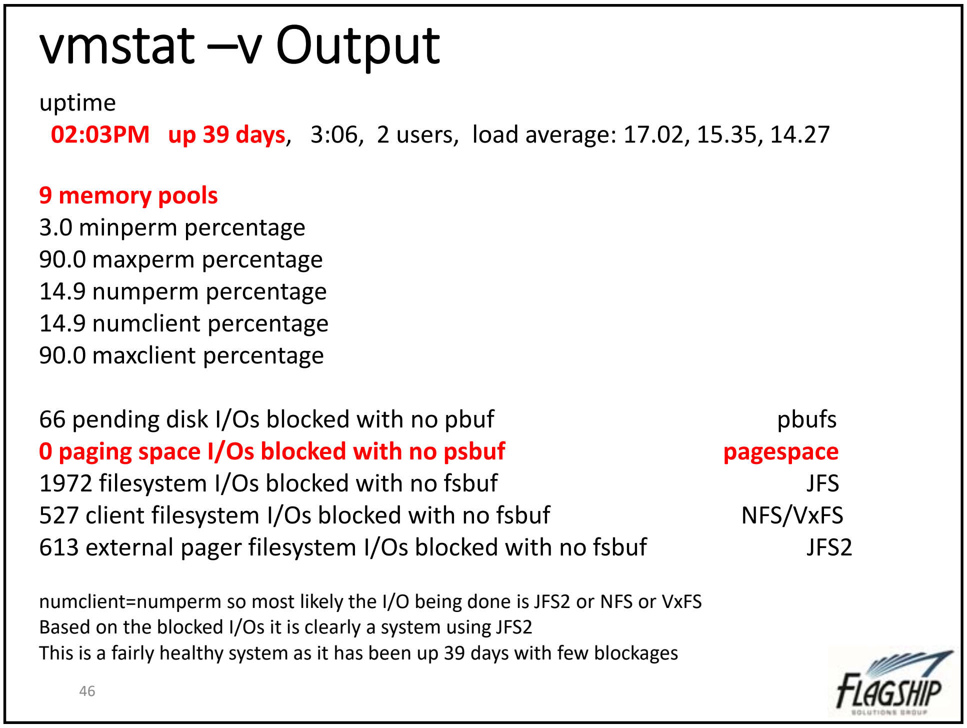


vmstat –v Output

Up 45 days
12 memory pools
3.0 minperm percentage
90.0 maxperm percentage
39.1 numperm percentage
39.1 numclient percentage
90.0 maxclient percentage

245 pending disk I/Os blocked with no pbuf	pbufs
0 paging space I/Os blocked with no psbuf	pagespace
2321 file system I/Os blocked with no fsbuf	JFS
484493 client file system I/Os blocked with no fsbuf	NFS/VxFS
2940752 external pager file system I/Os blocked with no fsbuf	JFS2
13.2 percentage of memory used for computational pages	

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



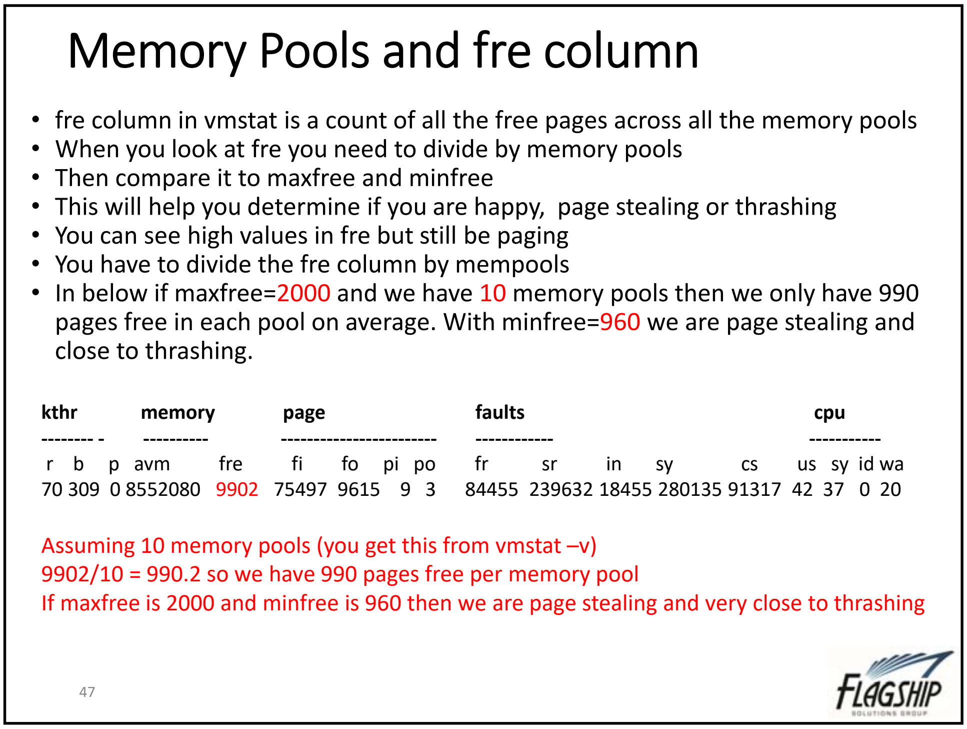
vmstat –v Output

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	pbufs
0 paging space I/Os blocked with no psbuf	**pagespace**
1972 filesystem I/Os blocked with no fsbuf	JFS
527 client filesystem I/Os blocked with no fsbuf	NFS/VxFS
613 external pager filesystem 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
This is a fairly healthy system as it has been up 39 days with few blockages

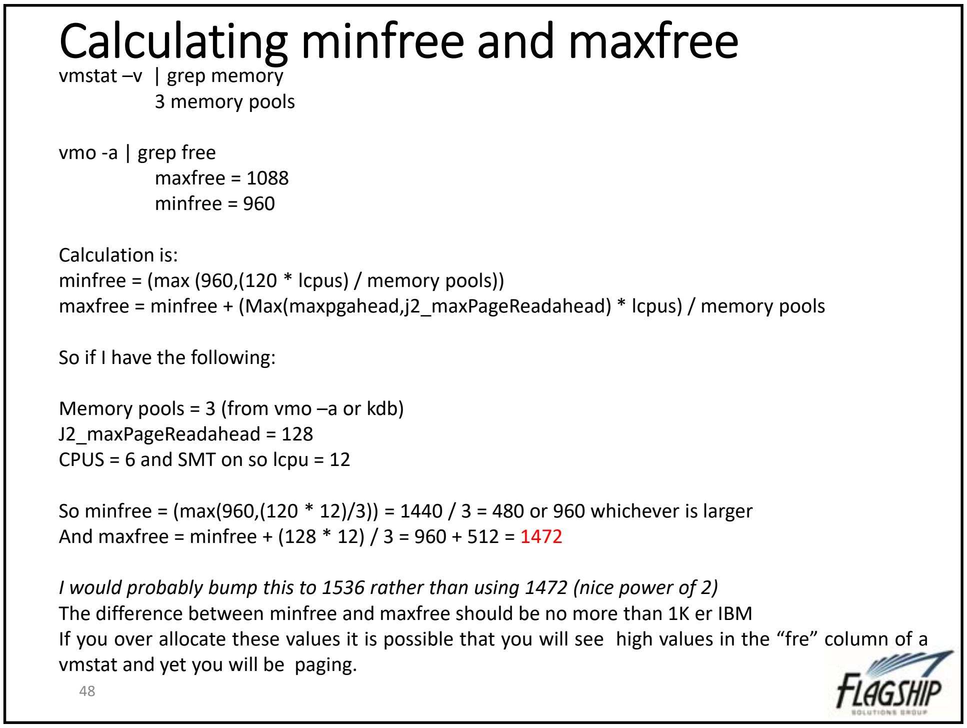


Memory Pools and fre column

- 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.

```
kthr        memory          page                     faults                                   cpu
-------- -   ----------     ------------------------  ------------                             -----------
 r   b   p  avm        fre      fi      fo    pi   po   fr        sr       in      sy          cs      us  sy  id wa
70 309  0 8552080  9902  75497  9615   9   3   84455  239632 18455 280135 91317  42  37   0  20
```

Assuming 10 memory pools (you get this from vmstat –v)
9902/10 = 990.2 so we have 990 pages free per memory pool
If maxfree is 2000 and minfree is 960 then we are page stealing and very 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)
The difference between minfree and maxfree should be no more than 1K er IBM
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
```

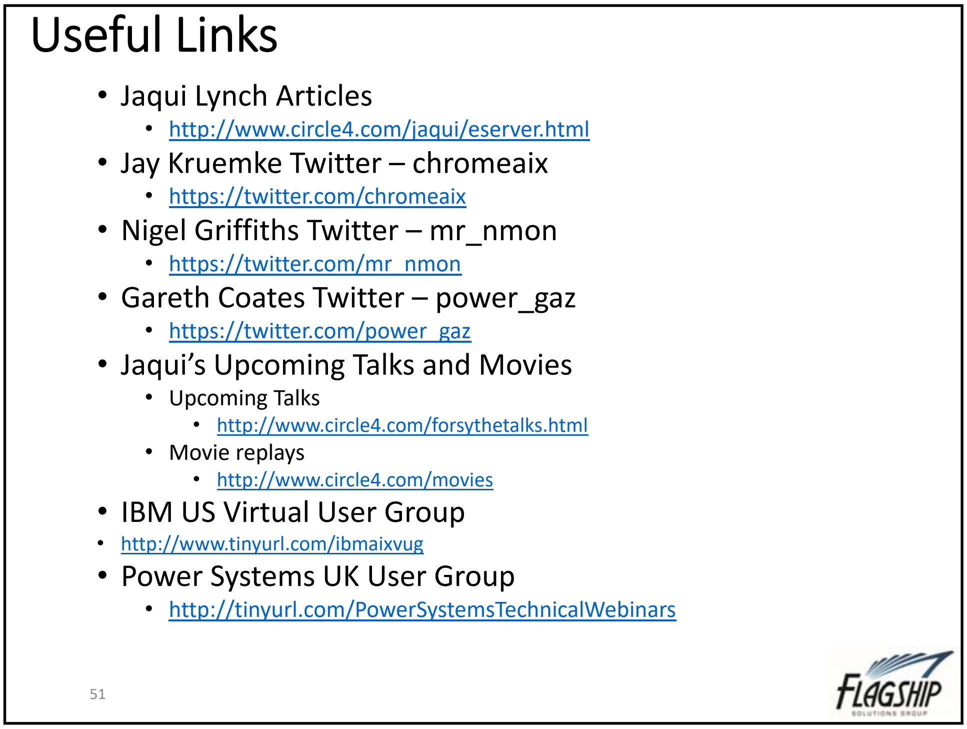
VIOS performance advisor also reports on the SEAs

Thank you for your time

If you have questions please email me at:
jaqui@circle4.com or jlynch@flagshipsg.com

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

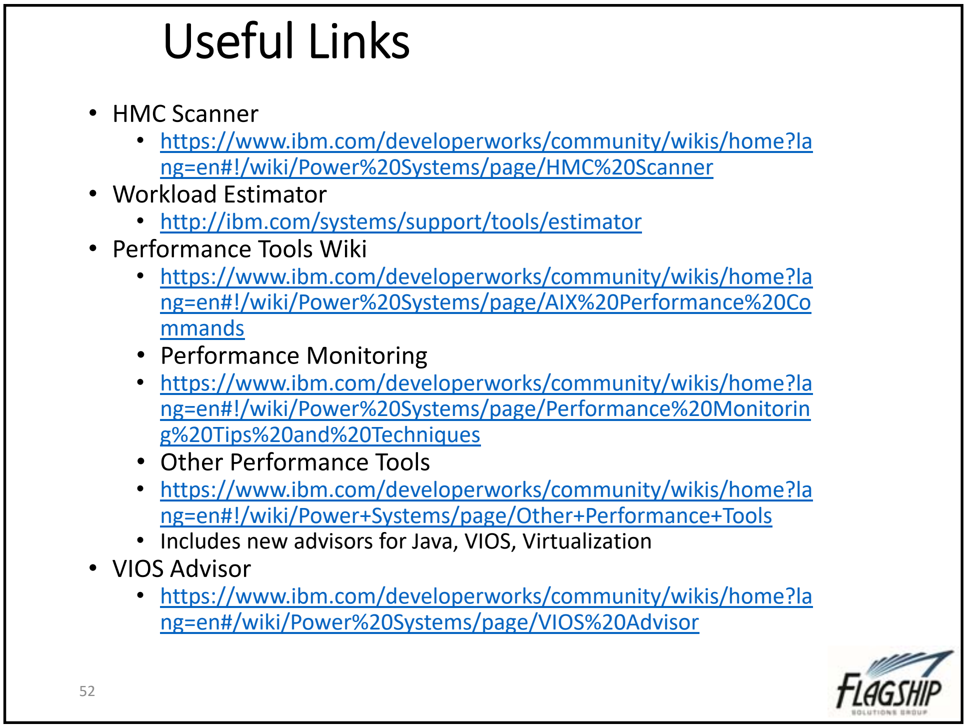
Don't forget to complete your evaluations!



Useful Links

- Jaqui Lynch Articles
  - http://www.circle4.com/jaqui/eserver.html
- Jay Kruemke Twitter – chromeaix
  - https://twitter.com/chromeaix
- Nigel Griffiths Twitter – mr_nmon
  - https://twitter.com/mr_nmon
- Gareth Coates Twitter – power_gaz
  - https://twitter.com/power_gaz
- Jaqui's Upcoming Talks and Movies
  - Upcoming Talks
    - http://www.circle4.com/forsythetalks.html
  - Movie replays
    - http://www.circle4.com/movies
- IBM US Virtual User Group
- http://www.tinyurl.com/ibmaixvug
- Power Systems UK User Group
  - http://tinyurl.com/PowerSystemsTechnicalWebinars

51



Useful Links

- HMC Scanner
  - https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power%20Systems/page/HMC%20Scanner
- Workload Estimator
  - http://ibm.com/systems/support/tools/estimator
- Performance Tools Wiki
  - https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power%20Systems/page/AIX%20Performance%20Commands
  - Performance Monitoring
  - https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power%20Systems/page/Performance%20Monitoring%20Tips%20and%20Techniques
  - Other Performance Tools
  - https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power+Systems/page/Other+Performance+Tools
  - Includes new advisors for Java, VIOS, Virtualization
- VIOS Advisor
  - https://www.ibm.com/developerworks/community/wikis/home?lang=en#/wiki/Power%20Systems/page/VIOS%20Advisor

52

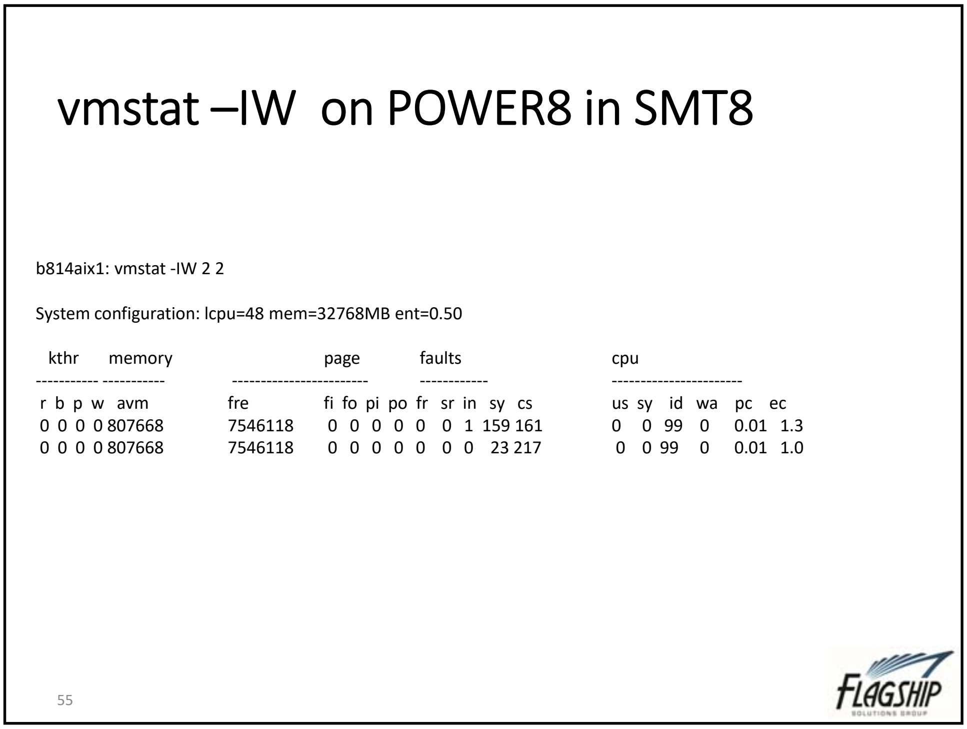
References

- Processor Utilization in AIX by Saravanan Devendran
  - https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Power%20Systems/page/Understanding%20CPU%20utilization%20on%20AIX
- Rosa Davidson Back to Basics Part 1 and 2 –Jan 24 and 31, 2013
  - https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Power%20Systems/page/AIX%20Virtual%20User%20Group%20-%20USA
- SG24-7940 - PowerVM Virtualization - Introduction and Configuration
  - http://www.redbooks.ibm.com/redbooks/pdfs/sg247940.pdf
- SG24-7590 – PowerVM Virtualization – Managing and Monitoring
  - http://www.redbooks.ibm.com/redbooks/pdfs/sg247590.pdf
- SG24-8171 – Power Systems Performance Optimization
  - http://www.redbooks.ibm.com/redbooks/pdfs/sg248171.pdf
- Redbook Tip on Maximizing the Value of P7 and P7+ through Tuning and Optimization
  - http://www.redbooks.ibm.com/technotes/tips0956.pdf

53

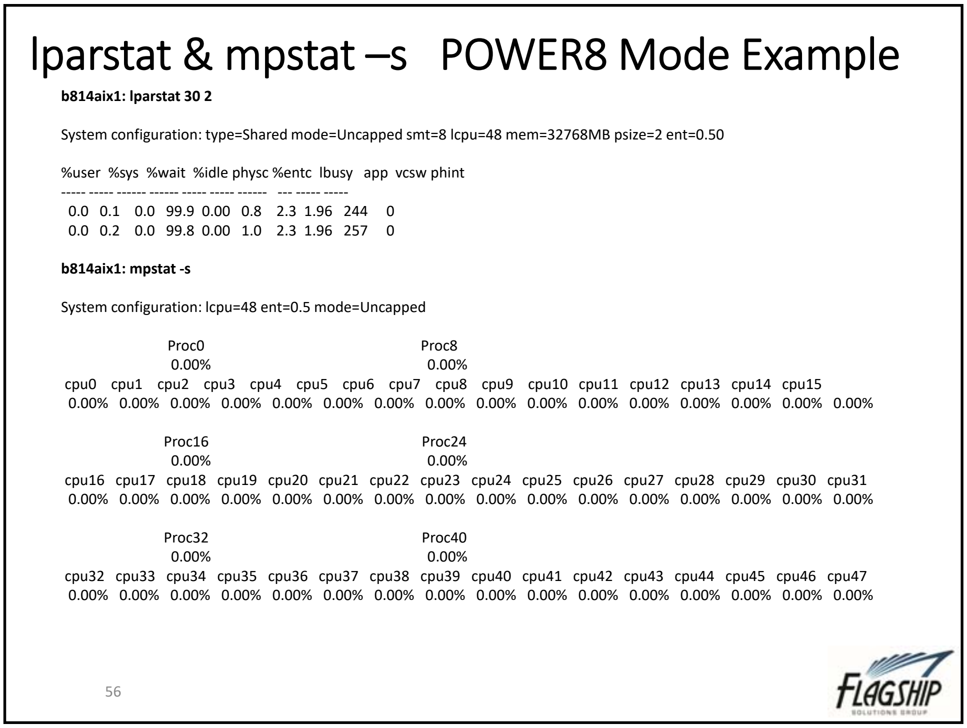
Backup Slides

54



vmstat –IW on POWER8 in SMT8

```
b814aix1: vmstat -IW 2 2

System configuration: lcpu=48 mem=32768MB ent=0.50

  kthr      memory                         page            faults                         cpu
----------- -----------            ------------------------          ------------                   -----------------------
 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    0   1  159 161             0    0   99    0     0.01   1.3
 0  0  0  0 807668              7546118      0   0   0   0   0    0   0    23 217              0    0  99    0     0.01   1.0
```



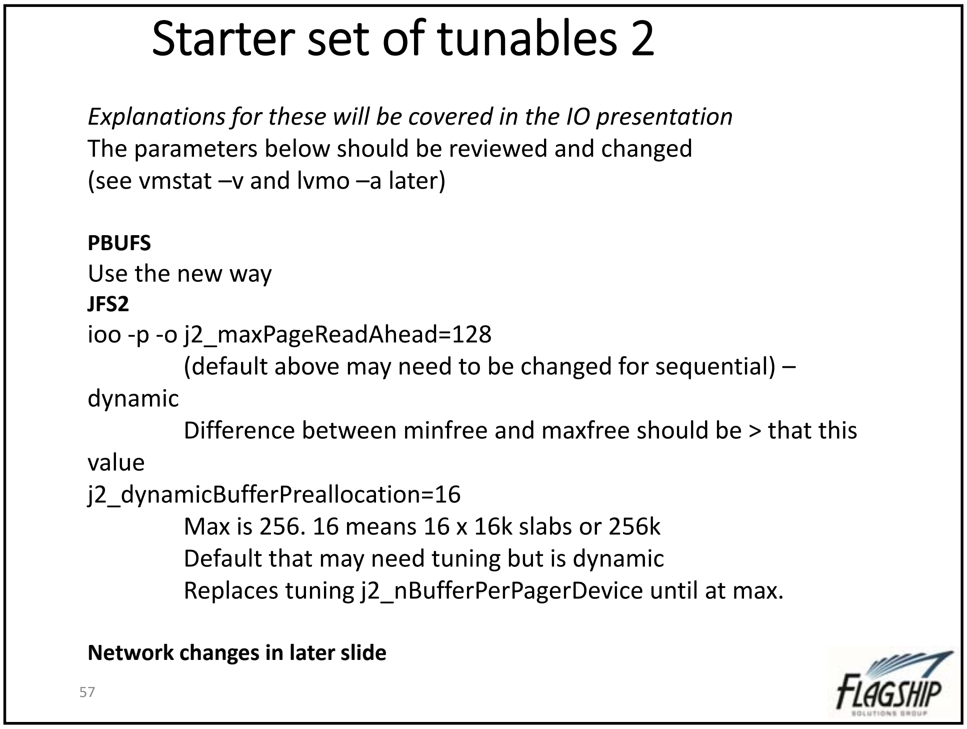
lparstat & mpstat –s   POWER8 Mode Example

b814aix1: lparstat 30 2

```
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                                        Proc8
                     0.00%                                        0.00%
 cpu0    cpu1    cpu2    cpu3    cpu4    cpu5    cpu6    cpu7    cpu8    cpu9    cpu10   cpu11   cpu12   cpu13   cpu14   cpu15
  0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%

                    Proc16                                       Proc24
                     0.00%                                        0.00%
 cpu16   cpu17   cpu18   cpu19   cpu20   cpu21   cpu22   cpu23   cpu24   cpu25   cpu26   cpu27   cpu28   cpu29   cpu30   cpu31
  0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%

                    Proc32                                       Proc40
                     0.00%                                        0.00%
 cpu32   cpu33   cpu34   cpu35   cpu36   cpu37   cpu38   cpu39   cpu40   cpu41   cpu42   cpu43   cpu44   cpu45   cpu46   cpu47
  0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%   0.00%
```

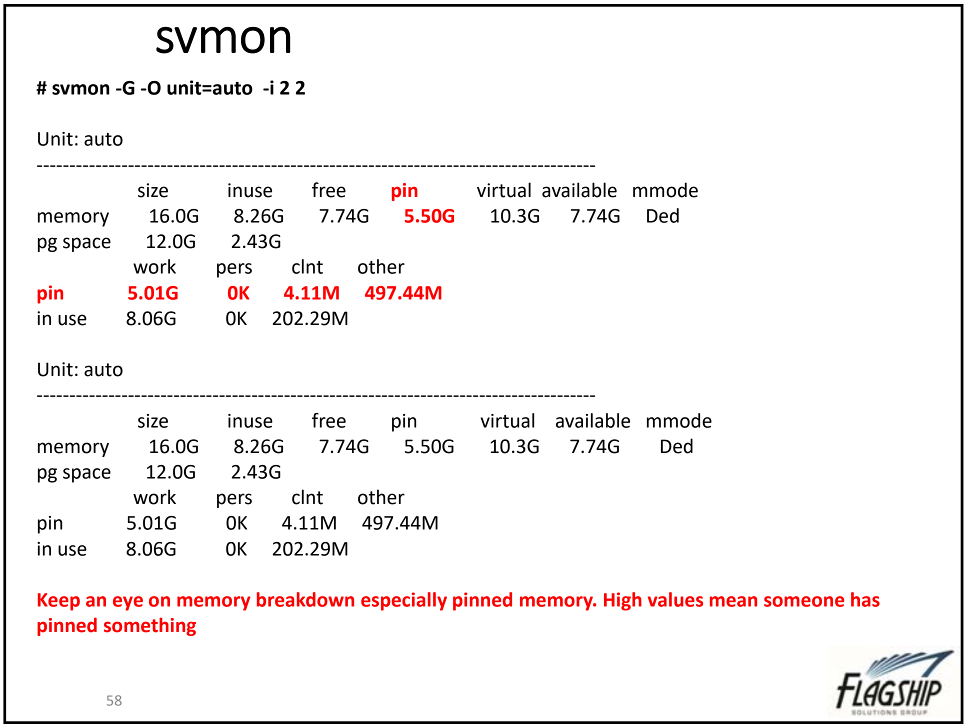


Starter set of tunables 2

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

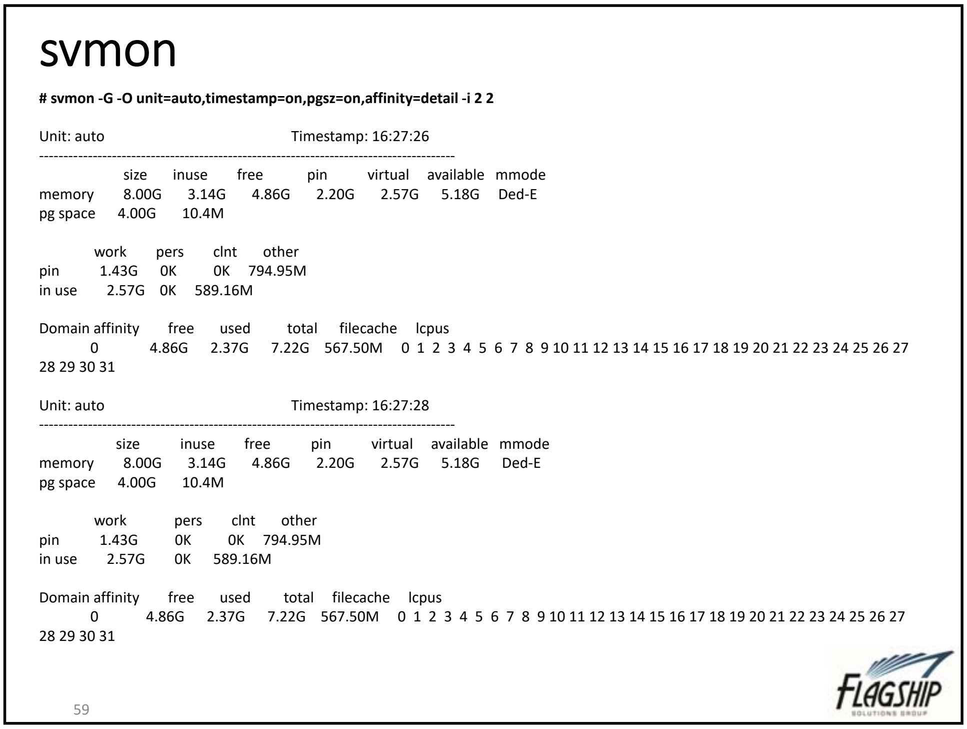


## svmon

# 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       clnt       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       clnt       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

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

```
Unit: auto                               Timestamp: 16:27:26
--------------------------------------------------------------------------------------------
              size      inuse      free         pin        virtual    available   mmode
memory      8.00G      3.14G      4.86G      2.20G      2.57G      5.18G     Ded-E
pg space    4.00G      10.4M

         work       pers       clnt      other
pin         1.43G      0K         0K    794.95M
in use       2.57G    0K     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

Unit: auto                               Timestamp: 16:27:28
--------------------------------------------------------------------------------------------
             size          inuse       free          pin         virtual    available   mmode
memory      8.00G      3.14G      4.86G      2.20G      2.57G      5.18G      Ded-E
pg space    4.00G      10.4M

         work           pers       clnt      other
pin         1.43G        0K         0K    794.95M
in use       2.57G       0K      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
```