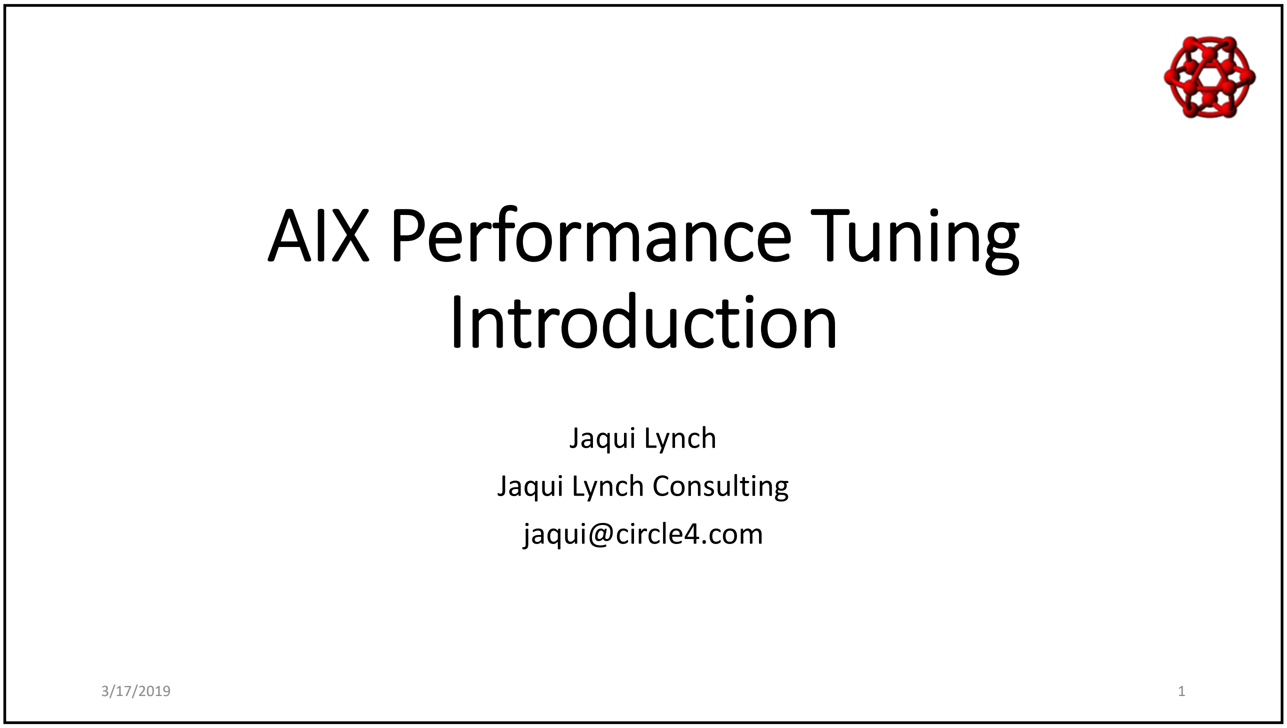


AIX Performance Tuning Introduction

Jaqui Lynch
Jaqui Lynch Consulting
jaqui@circle4.com

Agenda

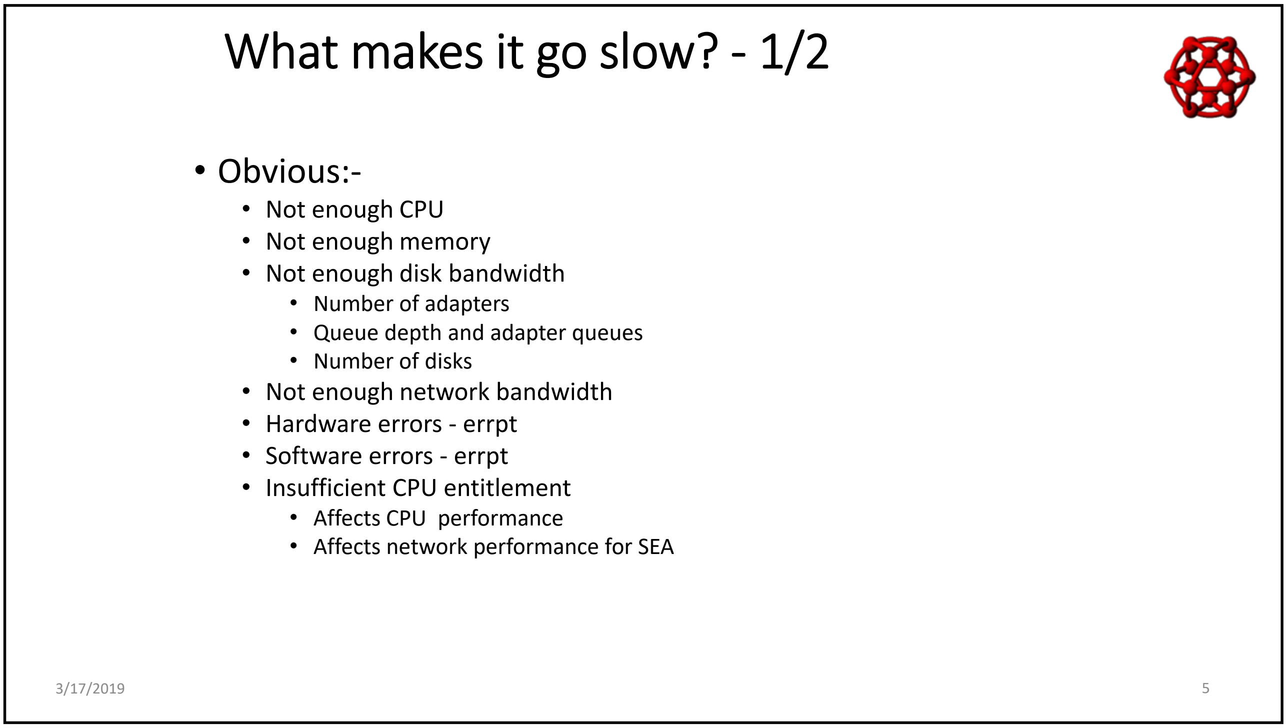
- Introduction
- CPU
- Memory
- I/O
- Network
- Notes

Introduction

What are you measuring?

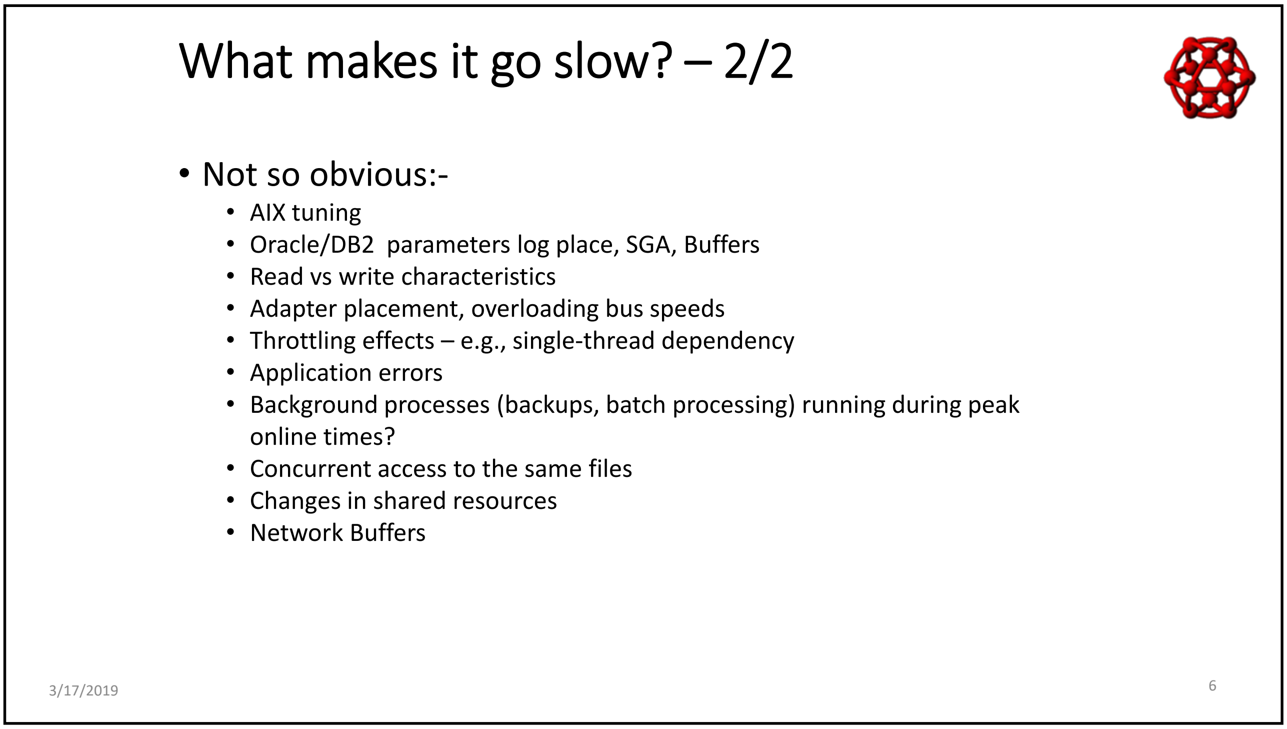
- *Response time* is the elapsed time between when a request is submitted and when the response from that request is returned.
  - Amount of time for a database query
  - Amount of time it takes to echo characters to the terminal
  - Amount of time it takes to access a Web page
  - How much time does my user wait?

- *Throughput* is a measure of the amount of work that can be accomplished over some unit of time.
  - Database transactions per minute
  - File transfer speed in KBs per second
  - File Read or Write KBs per second
  - Web server hits per minute



What makes it go slow? - 1/2

- Obvious:-
  - Not enough CPU
  - Not enough memory
  - Not enough disk bandwidth
    - Number of adapters
    - Queue depth and adapter queues
    - Number of disks
  - Not enough network bandwidth
  - Hardware errors - errpt
  - Software errors - errpt
  - Insufficient CPU entitlement
    - Affects CPU performance
    - Affects network performance for SEA



What makes it go slow? – 2/2

- Not so obvious:-
  - AIX tuning
  - Oracle/DB2 parameters log place, SGA, Buffers
  - Read vs write characteristics
  - Adapter placement, overloading bus speeds
  - Throttling effects – e.g., single-thread dependency
  - Application errors
  - Background processes (backups, batch processing) running during peak online times?
  - Concurrent access to the same files
  - Changes in shared resources
  - Network Buffers

Before you start - take a deep breath!

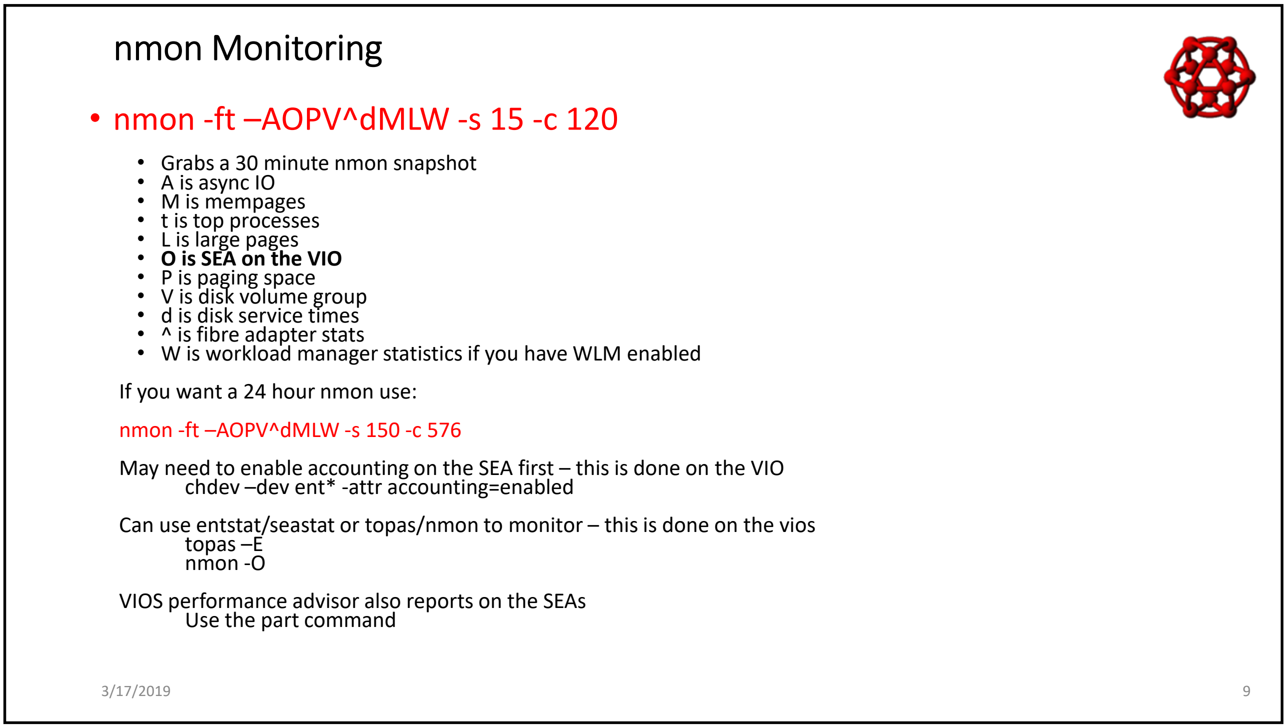
Image courtesy of teefury.com

Take a baseline before you start

- You can use perfpmr, nmon or whatever works for you
  - Must be consistent and repeatable
- On AIX v5.3 you must download nmon12 – don't use the nmon_topas version
- I collect nmon data 7/24 as follows:
- Crontab entry:
  - 59 23 * * * /usr/local/bin/runnmon.sh >/dev/null 2>&1
- runnmon.sh:

```
#!/bin/ksh
#
cd /usr/local/perf
/usr/bin/nmon -ft –AOPV^dMLW -s 150 -c 576
```

Without a baseline you have nothing to compare to after changes are made



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

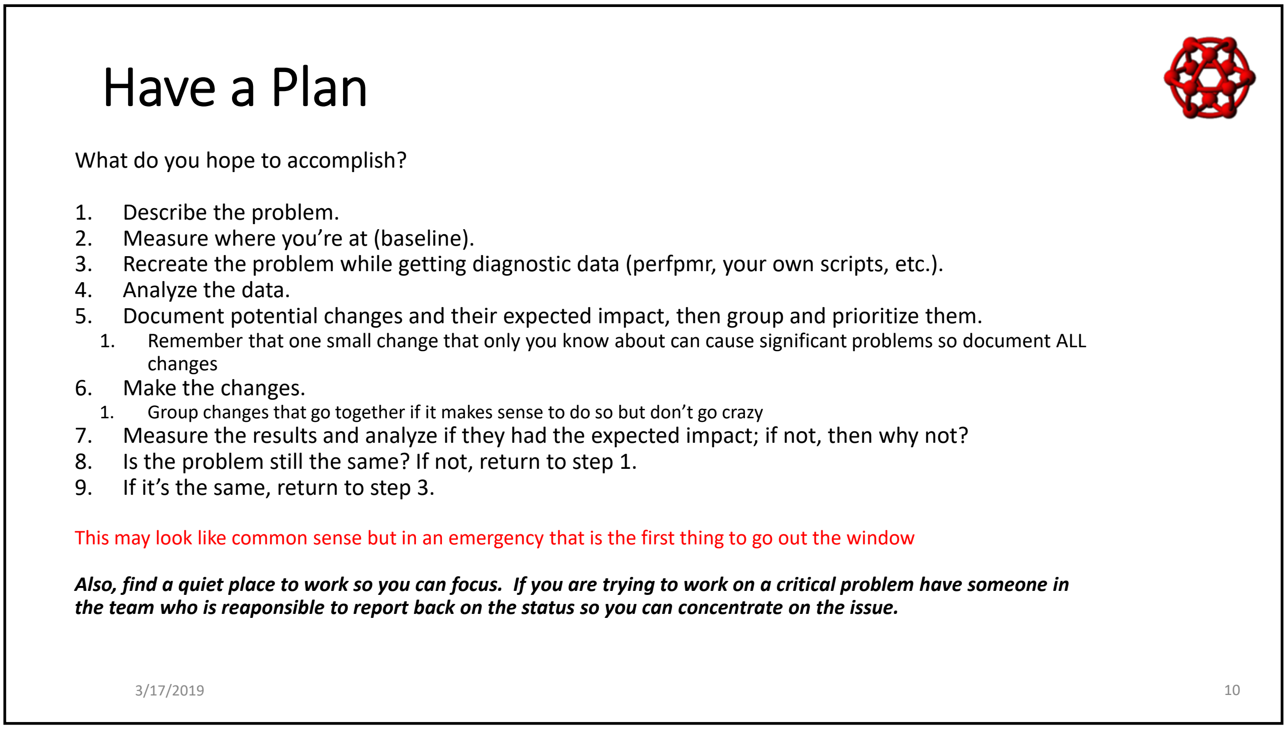
```
chdev –dev ent* -attr accounting=enabled
```

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

Use the part command



Have a Plan

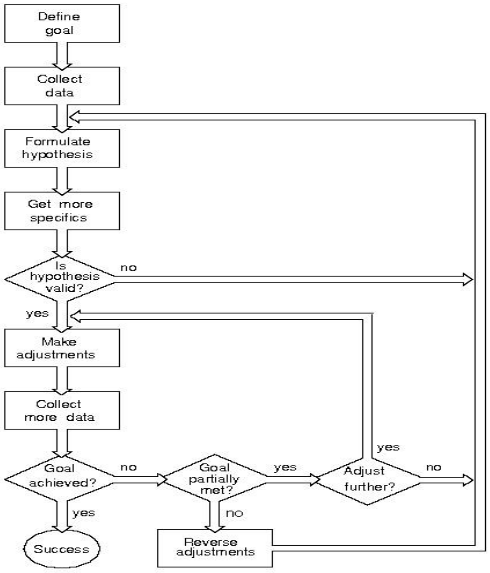
What do you hope to accomplish?

1. Describe the problem.
2. Measure where you’re at (baseline).
3. Recreate the problem while getting diagnostic data (perfpmr, your own scripts, etc.).
4. Analyze the data.
5. Document potential changes and their expected impact, then group and prioritize them.
   1. Remember that one small change that only you know about can cause significant problems so document ALL changes
6. Make the changes.
   1. Group changes that go together if it makes sense to do so but don’t go crazy
7. Measure the results and analyze if they had the expected impact; if not, then why not?
8. Is the problem still the same? If not, return to step 1.
9. If it’s the same, return to step 3.

This may look like common sense but in an emergency that is the first thing to go out the window

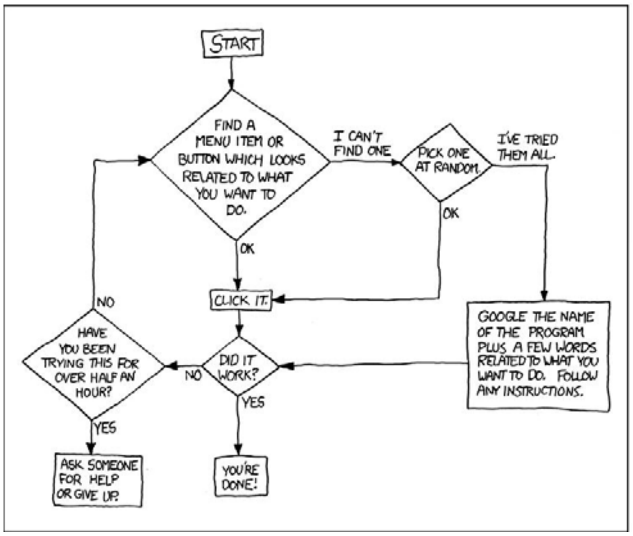
Also, find a quiet place to work so you can focus. If you are trying to work on a critical problem have someone in the team who is reaponsible to report back on the status so you can concentrate on the issue.

Pick a Methodology



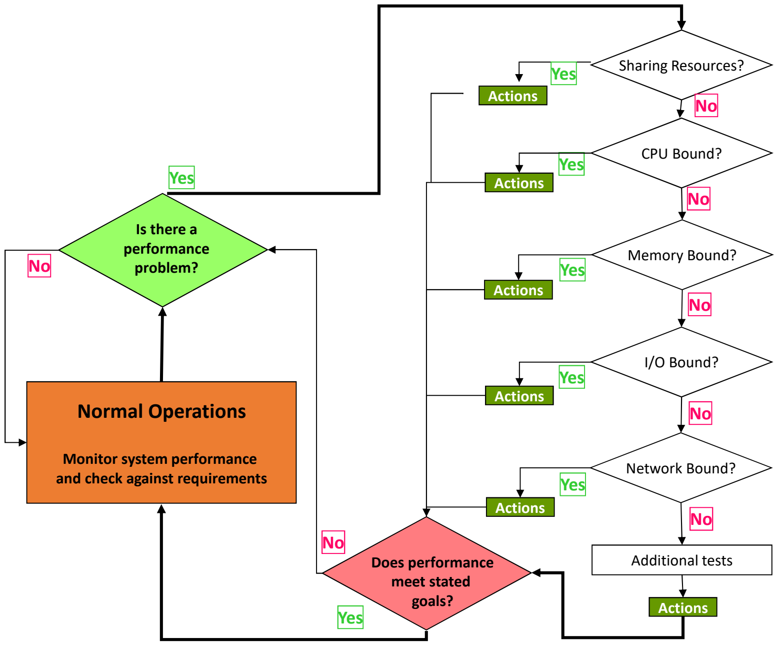
Flowchart illustrating the methodology for system performance tuning

Performance Support Flowchart


START
FIND A MENU ITEM OR BUTTON WHICH LOOKS RELATED TO WHAT YOU WANT TO DO.
I CAN'T FIND ONE
PICK ONE AT RANDOM.
I'VE TRIED THEM ALL.
OK
OK
CLICK IT.
NO
HAVE YOU BEEN TRYING THIS FOR OVER HALF AN HOUR?
NO
DID IT WORK?
GOOGLE THE NAME OF THE PROGRAM PLUS A FEW WORDS RELATED TO WHAT YOU WANT TO DO. FOLLOW ANY INSTRUCTIONS.
YES
YES
ASK SOMEONE FOR HELP OR GIVE UP.
YOU'RE DONE!

Courtesy of XKCD

Performance Analysis Flowchart



Normal Operations

Monitor system performance and check against requirements

Is there a performance problem?

Yes

No

Sharing Resources?

CPU Bound?

Memory Bound?

I/O Bound?

Network Bound?

Additional tests

Actions

Does performance meet stated goals?

3/17/2019

13

13

CPU

3/17/2019

14

14

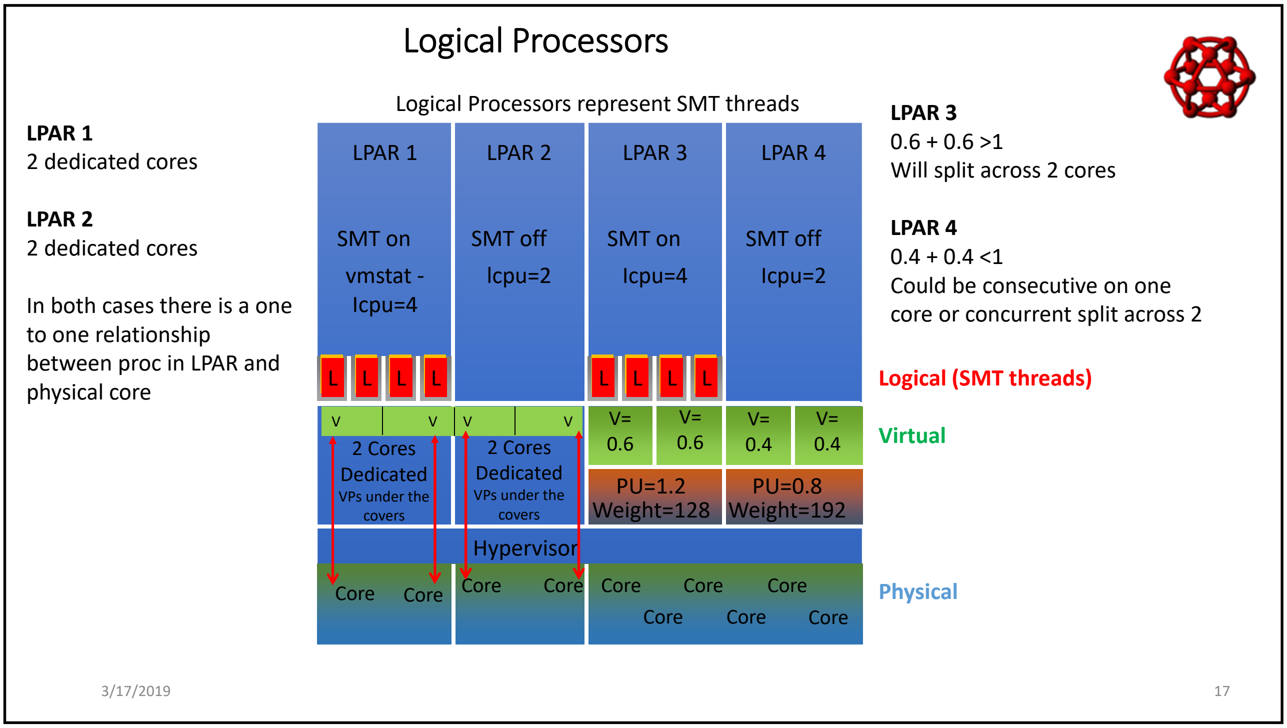
Terms to understand

- Process
  - A process is an activity within the system that is started with a command, a shell script, or another process.
- Run Queue
  - Each CPU has a dedicated run queue. A run queue is a list of runnable threads, sorted by thread priority value. There are 256 thread priorities (zero to 255). There is also an additional global run queue where new threads are placed.
- Time Slice
  - The CPUs on the system are shared among all of the threads by giving each thread a certain slice of time to run. The default time slice of one clock tick is 10 ms
- Virtual Processor (VP)
  - LPARs view of the real core – shows as proc???
- Logical processor (LP)
  - LPARs view of the SMT threads – shows as cpu??? In mpstat
- Core (real processor)
  - Actual physical cores that get matched to VPs

Monitoring CPU

- User, system, wait and idle are fine for dedicated LPARs
- They are not fine for SPLPAR or dedicated donating LPARs
- You need to measure and charge back based on used CPU cycles
- Moral of the story – use Physc (Physical consumed)

- lparstat
  - Use with no flags to view partition configuration and processor usage
- mpstat
- schedo
- vmstat
- topas, nmon
- sar
- emstat
- tprof, gprof, curt, splat, curl, trace, probevue
- time
- Many many more



Logical Processors

LPAR 1
2 dedicated cores

LPAR 2
2 dedicated cores

In both cases there is a one to one relationship between proc in LPAR and physical core

Logical Processors represent SMT threads

LPAR 3
0.6 + 0.6 >1
Will split across 2 cores

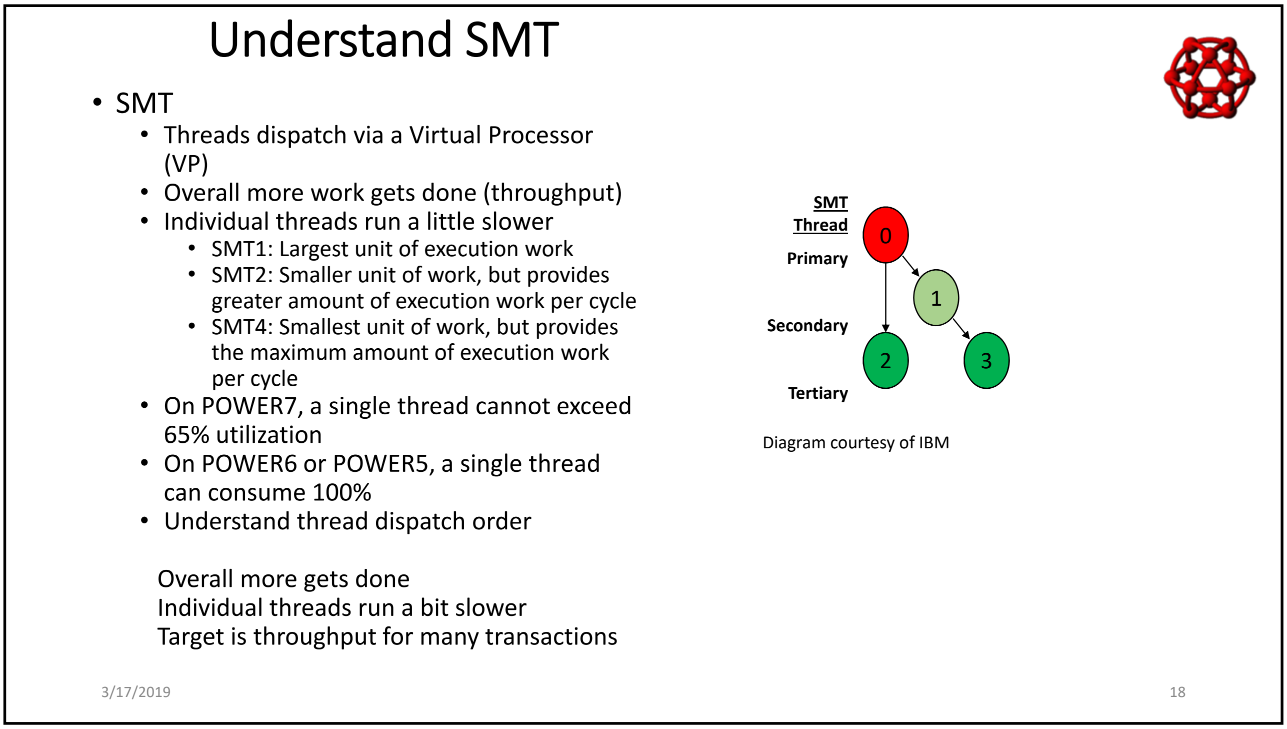
LPAR 4
0.4 + 0.4 <1
Could be consecutive on one core or concurrent split across 2

Logical (SMT threads)

Virtual

Physical

3/17/2019 17



Understand SMT

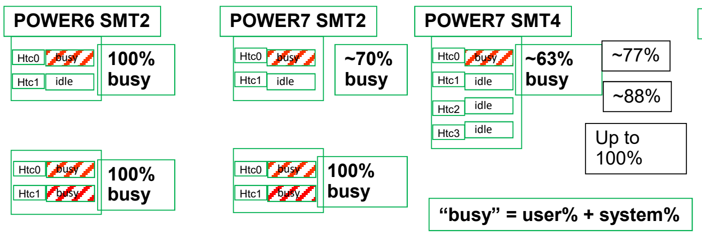
- SMT
  - Threads dispatch via a Virtual Processor (VP)
  - Overall more work gets done (throughput)
  - Individual threads run a little slower
    - SMT1: Largest unit of execution work
    - SMT2: Smaller unit of work, but provides greater amount of execution work per cycle
    - SMT4: Smallest unit 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

Overall more gets done
Individual threads run a bit slower
Target is throughput for many transactions

Diagram courtesy of IBM

3/17/2019 18

POWER5/6 vs POWER7/8 - SMT Utilization



"busy" = user% + system%

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

SMT Notes	POWER8	POWER9
Uplift from SMT1 to SMT2	45%	70%
Uplift from SMT2 to SMT4 is about	30%	38%
Uplift from SMT4 to SMT8 is about	7%	26%

Check published rPerf Numbers
Dependent on workload and how SMT friendly it is

SMT Mode	Core utilization% 1 busy* thread (1 thread / vcpu)		
OS Architecture	Linux P7/P8/P9	AIX POWER8	AIX POWER9
Mode	P8/P9	P8	P8/P9
Single Thread	90-99%	99%	99%
2	50%	77%	50%
4	25%	60%	44%
8	12.5%	56%	32%

*Single core, single VP

Diagrams courtesy of Nigel Griffiths Power7 Affinity and Steve Nasypany What's new in Performance

Dispatching in the shared pool

- Dedicated LPAR has a 1-1 relationship between core and VP
- Shared pool VP gets assigned core at dispatch time

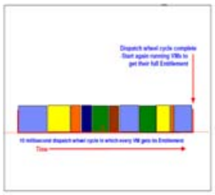
- VP gets dispatched to a core
  - First time this becomes the home node
  - All SMT threads for the VP go with the VP
- VP runs to the end of its entitlement
  - If it has more work to do and noone else wants the core it gets more
  - If it has more work to do but other VPs want the core then it gets 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

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

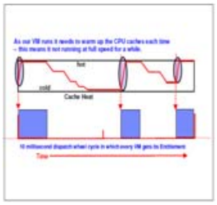

As our VM runs it needs to warm up the CPU caches each time
– this means it not running at full speed for a while.
hot
cold
Cache Heat
10 millisecond dispatch wheel cycle in which every VM gets its Entitlement
Time

Diagram courtesy of 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

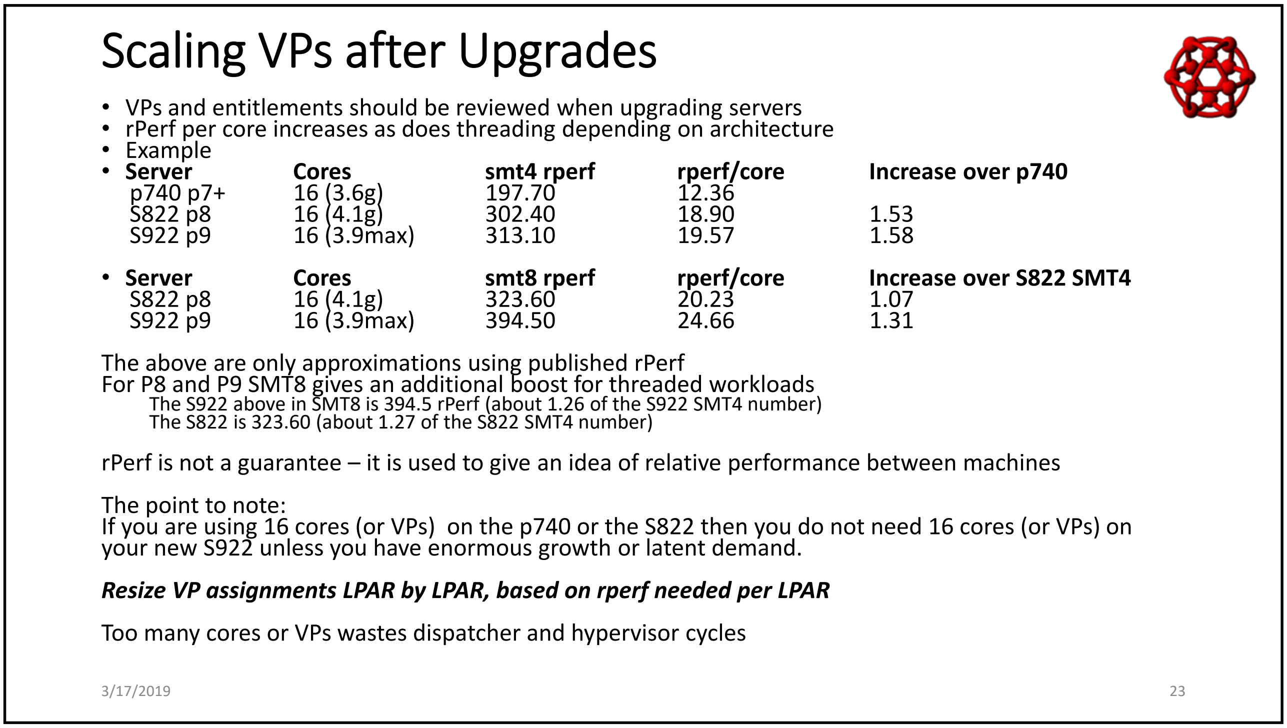
- This means that in POWER7 you need to pay more attention to VPs
  - You may see more cores activated a lower utilization levels
  - But you will see higher idle
  - If only primary SMT threads in use then you have excess VPs

- 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**
    - **This applies to all LPARs using the SEA as well**

  - 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**



Scaling VPs after Upgrades

- VPs and entitlements should be reviewed when upgrading servers
- rPerf per core increases as does threading depending on architecture
- Example

Server	Cores	smt4 rperf	rperf/core	Increase over p740
p740 p7+	16 (3.6g)	197.70	12.36	
S822 p8	16 (4.1g)	302.40	18.90	1.53
S922 p9	16 (3.9max)	313.10	19.57	1.58

Server	Cores	smt8 rperf	rperf/core	Increase over S822 SMT4
S822 p8	16 (4.1g)	323.60	20.23	1.07
S922 p9	16 (3.9max)	394.50	24.66	1.31

The above are only approximations using published rPerf
For P8 and P9 SMT8 gives an additional boost for threaded workloads

The S922 above in SMT8 is 394.5 rPerf (about 1.26 of the S922 SMT4 number)
The S822 is 323.60 (about 1.27 of the S822 SMT4 number)

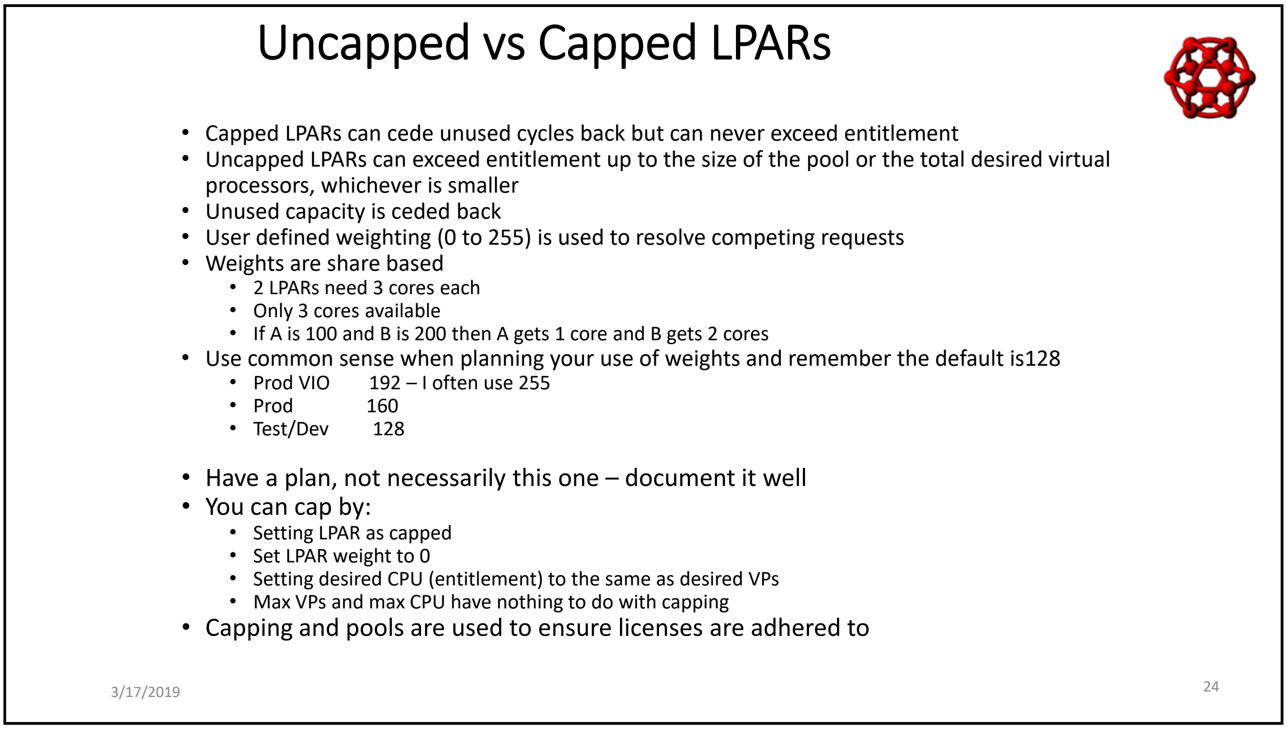
rPerf is not a guarantee – it is used to give an idea of relative performance between machines

The point to note:
If you are using 16 cores (or VPs) on the p740 or the S822 then you do not need 16 cores (or VPs) on your new S922 unless you have enormous growth or latent demand.

Resize VP assignments LPAR by LPAR, based on rperf needed per LPAR

Too many cores or VPs wastes dispatcher and hypervisor cycles

23



Uncapped vs Capped LPARs

- Capped LPARs can cede unused cycles back but can never exceed entitlement
- Uncapped LPARs can exceed entitlement up to the size of the pool or the total desired virtual processors, whichever is smaller
- Unused capacity is ceded back
- User defined weighting (0 to 255) is used to resolve competing requests
- Weights are share based
  - 2 LPARs need 3 cores each
  - Only 3 cores available
  - If A is 100 and B is 200 then A gets 1 core and B gets 2 cores
- Use common sense when planning your use of weights and remember the default is128
  - Prod VIO 192 – I often use 255
  - Prod 160
  - Test/Dev 128
- Have a plan, not necessarily this one – document it well
- You can cap by:
  - Setting LPAR as capped
  - Set LPAR weight to 0
  - Setting desired CPU (entitlement) to the same as desired VPs
  - Max VPs and max CPU have nothing to do with capping
- Capping and pools are used to ensure licenses are adhered to

24

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 & above load the first thread on the VP to 49% 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

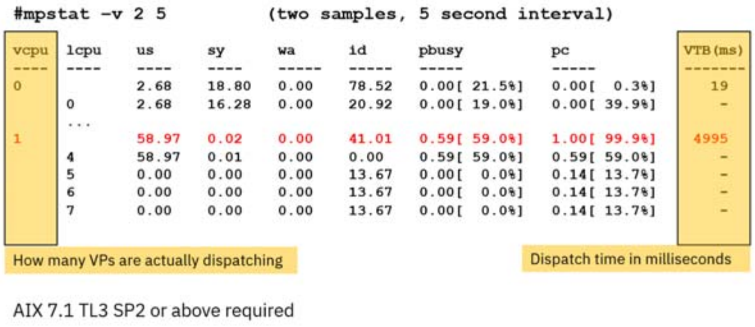
Why?

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”

Dispatch changes in POWER9

- See Steve Nasypany What’s new in Power Performance – page 38
  - https://www.ibm.com/developerworks/community/files/basic/anonymous/api/library/d7e42915-b8b6-4e55-a820-a6d9eea3cde0/document/28b7e2c3-db04-4c70-942c-9f18d0306aae/media/Whats_New_Performance_RTP_UG.pdf
- Dispatch Behavior in AIX / POWER9 POWER9/AIX in POWER9 Mode behaves a little differently. Dispatcher tries to collapse smaller workloads (single thread less than 32%)
- SMT8 will be the new default on POWER9 at AIX v7.2 TL3
- Use mpstat –v to monitor VP and SMT activity
- This shows the actual Virtual Time Base (VTB) in ms, the dispatch time for each VP at the physical layer, physical consumption(pc) and the actual dispatch time of the SMT threads.



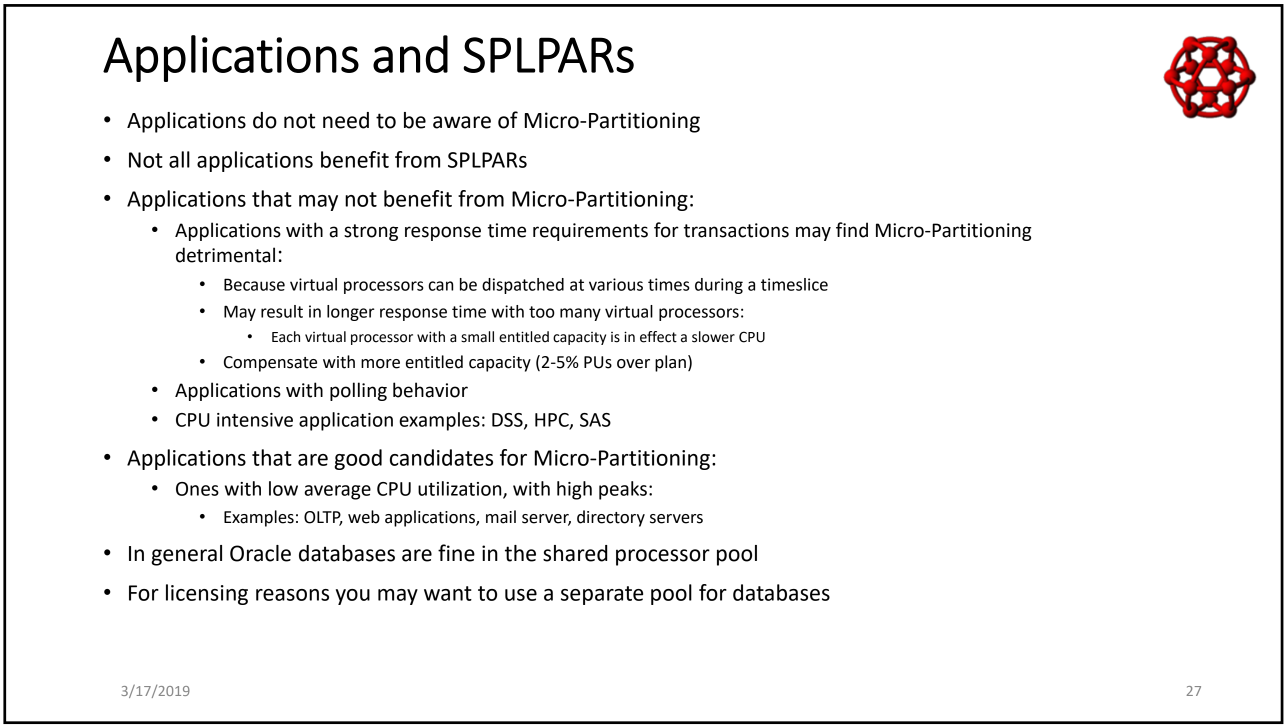
```
#mpstat -v 2 5          (two samples, 5 second interval)

vcpu  lcpu    us      sy      wa      id      pbusy          pc             VTB(ms)
----  ----    ----    ----    -----   -----   -----          -----          -------
0             2.68    18.80   0.00    78.52   0.00[ 21.5%]   0.00[  0.3%]        19
      0       2.68    16.28   0.00    20.92   0.00[ 19.0%]   0.00[ 39.9%]         -
      ...
1             58.97   0.02    0.00    41.01   0.59[ 59.0%]   1.00[ 99.9%]      4995
      4       58.97   0.01    0.00    0.00    0.59[ 59.0%]   0.59[ 59.0%]         -
      5       0.00    0.00    0.00    13.67   0.00[  0.0%]   0.14[ 13.7%]         -
      6       0.00    0.00    0.00    13.67   0.00[  0.0%]   0.14[ 13.7%]         -
      7       0.00    0.00    0.00    13.67   0.00[  0.0%]   0.14[ 13.7%]         -
```

How many VPs are actually dispatching

Dispatch time in milliseconds

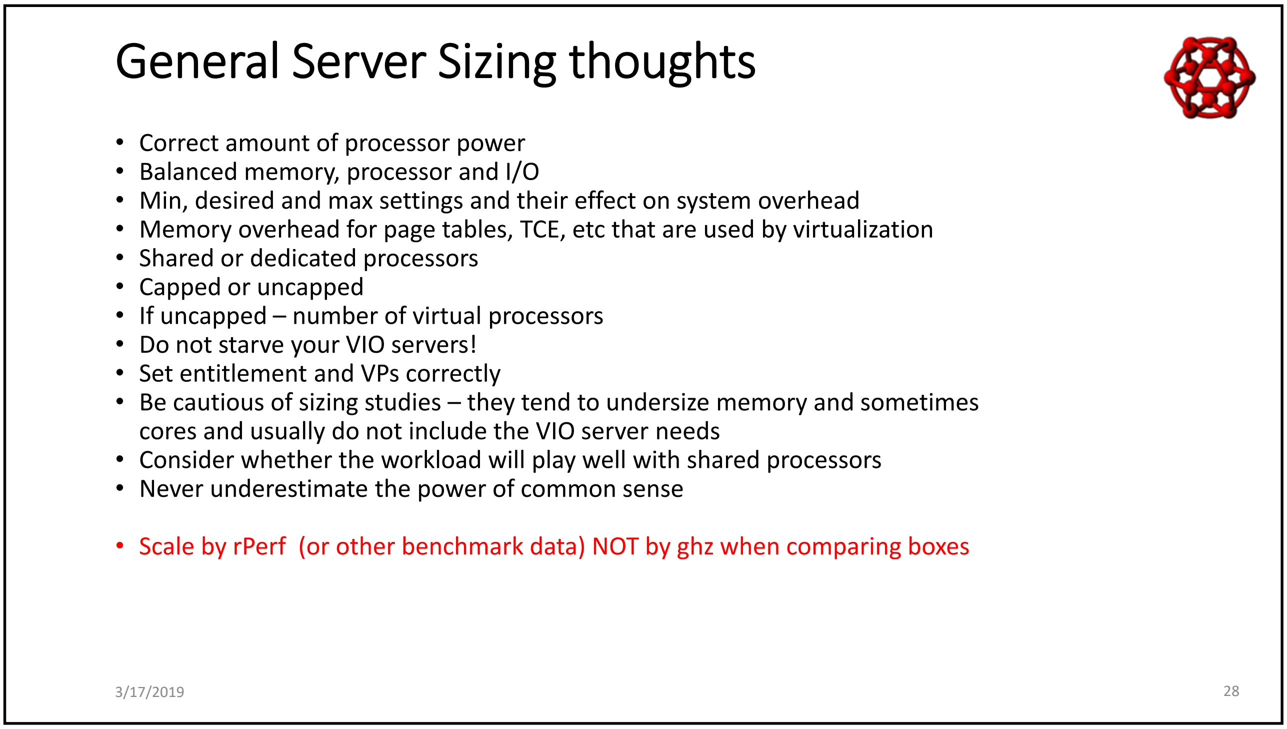
AIX 7.1 TL3 SP2 or above required

Diagram from P102718



Applications and SPLPARs

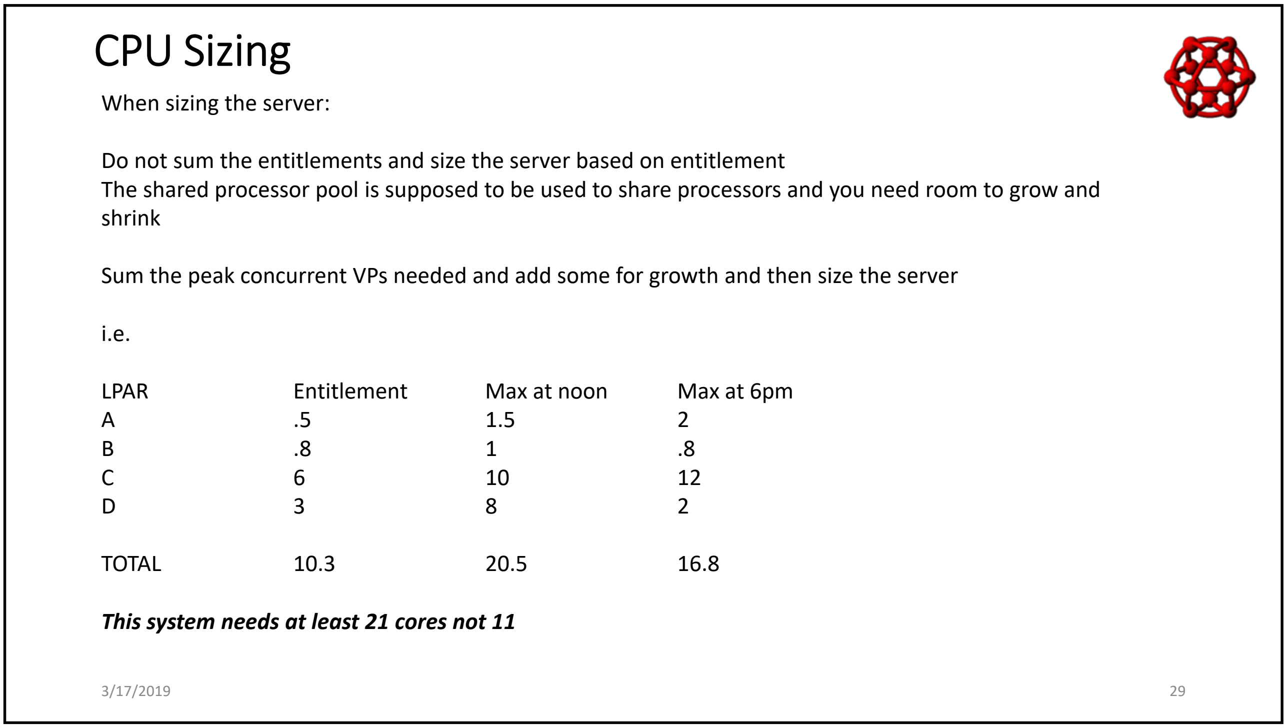
- Applications do not need to be aware of Micro-Partitioning
- Not all applications benefit from SPLPARs
- Applications that may not benefit from Micro-Partitioning:
  - Applications with a strong response time requirements for transactions may find Micro-Partitioning detrimental:
    - Because virtual processors can be dispatched at various times during a timeslice
    - May result in longer response time with too many virtual processors:
      - Each virtual processor with a small entitled capacity is in effect a slower CPU
    - Compensate with more entitled capacity (2-5% PUs over plan)
  - Applications with polling behavior
  - CPU intensive application examples: DSS, HPC, SAS
- Applications that are good candidates for Micro-Partitioning:
  - Ones with low average CPU utilization, with high peaks:
    - Examples: OLTP, web applications, mail server, directory servers
- In general Oracle databases are fine in the shared processor pool
- For licensing reasons you may want to use a separate pool for databases



General Server Sizing thoughts

- Correct amount of processor power
- Balanced memory, processor and I/O
- Min, desired and max settings and their effect on system overhead
- Memory overhead for page tables, TCE, etc that are used by virtualization
- Shared or dedicated processors
- Capped or uncapped
- If uncapped – number of virtual processors
- Do not starve your VIO servers!
- Set entitlement and VPs correctly
- Be cautious of sizing studies – they tend to undersize memory and sometimes cores and usually do not include the VIO server needs
- Consider whether the workload will play well with shared processors
- Never underestimate the power of common sense

- Scale by rPerf (or other benchmark data) NOT by ghz when comparing boxes



CPU Sizing

When sizing the server:

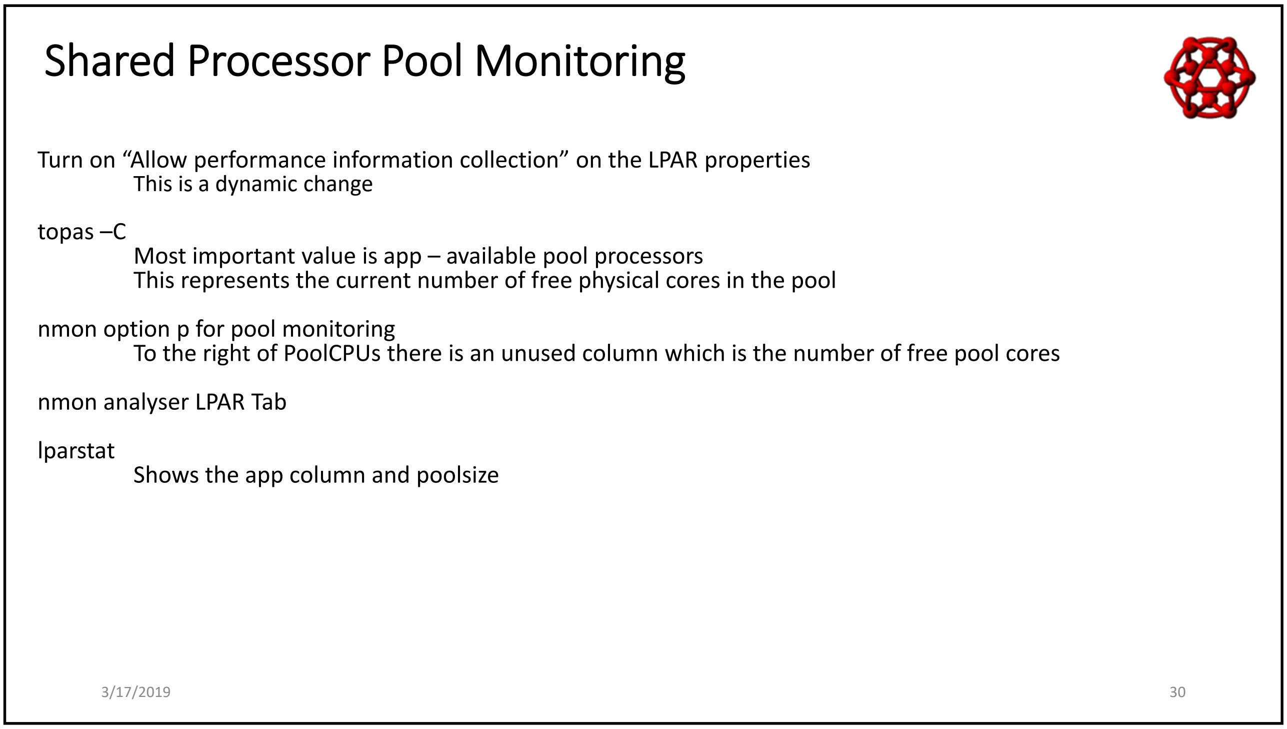
Do not sum the entitlements and size the server based on entitlement
The shared processor pool is supposed to be used to share processors and you need room to grow and shrink

Sum the peak concurrent VPs needed and add some for growth and then size the server

i.e.

LPAR	Entitlement	Max at noon	Max at 6pm
A	.5	1.5	2
B	.8	1	.8
C	6	10	12
D	3	8	2
TOTAL	10.3	20.5	16.8

This system needs at least 21 cores not 11



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

emstat

```
emstat -a 30 2
```

```
 Alignment  Alignment  Emulation  Emulation
 SinceBoot      Delta  SinceBoot      Delta
         0          0          0          0
         0          0          0          0
```

Check for emulations and alignments
Caused when old code run on new architectures or operating systems
Code should be recompiled whenever possible when upgrades occur

Binary compatibility statement means it will work but the best performance is obtained when you run a version compiled for the architecture

MEMORY

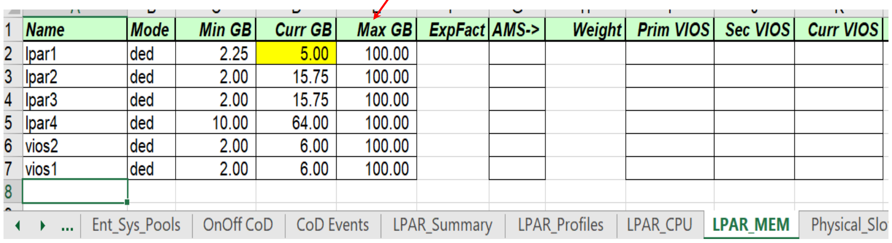
Memory Types

- Kernel
  - This is AIX itself and the minimum is around 2GB.
- Persistent
  - Backed by filesystems (called file cache)
  - Raw devices, GPFS, Oracle ASM does not make use of file cache.
- 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

Server Memory or Hypervisor Overhead

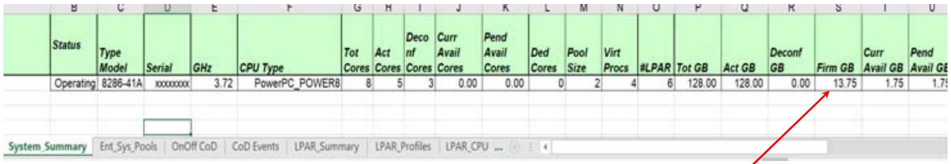
- Reserved Memory is based on max memory for an LPAR, not on desired
- This is because memory gets reserved for HPTs (hypervisor page tables) and must be able to accomodate the maximum memory that can be in an LPAR after a DLPAR operation

Look at Max GB



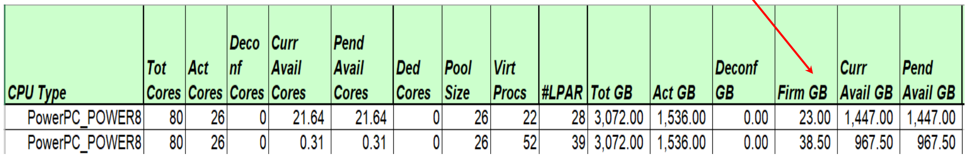
1	Name	Mode	Min GB	Curr GB	Max GB	ExpFact	AMS->	Weight	Prim VIOS	Sec VIOS	Curr VIOS
2	lpar1	ded	2.25	5.00	100.00						
3	lpar2	ded	2.00	15.75	100.00						
4	lpar3	ded	2.00	15.75	100.00						
5	lpar4	ded	10.00	64.00	100.00						
6	vios2	ded	2.00	6.00	100.00						
7	vios1	ded	2.00	6.00	100.00						
8											

Server Memory in HMCScanner report



	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	Status	Type Model	Serial	GHz	CPU Type	Tot Cores	Act Cores	Deconf Cores	Curr Avail Cores	Pend Avail Cores	Ded Cores	Pool Size	Virt Procs	#LPAR	Tot GB	Act GB	Deconf GB	Firm GB	Curr Avail GB	Pend Avail GE
	Operating	8286-41A	xxxxxxxx	3.72	PowerPC_POWER8	8	5	3	0.00	0.00	0	2	4	6	128.00	128.00	0.00	13.75	1.75	1.75

Look at Firm GB

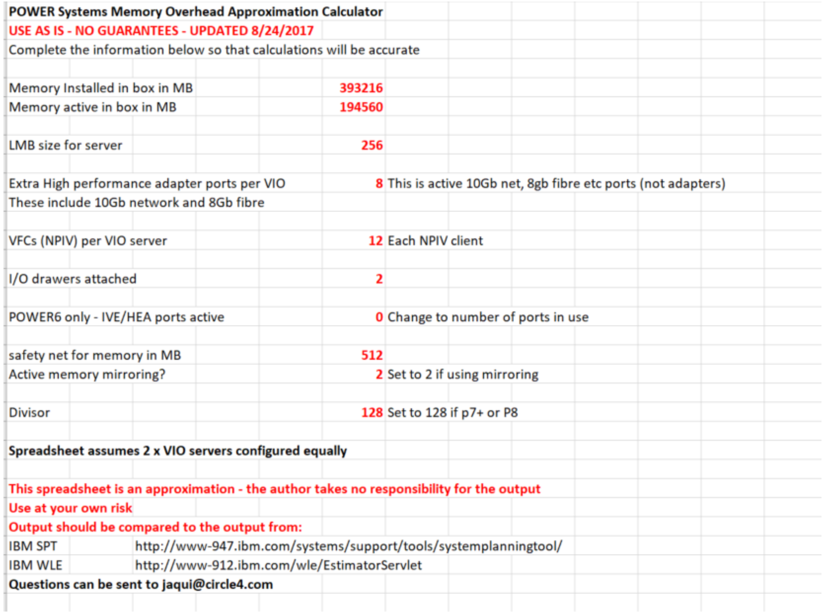


CPU Type	Tot Cores	Act Cores	Deconf Cores	Curr Avail Cores	Pend Avail Cores	Ded Cores	Pool Size	Virt Procs	#LPAR	Tot GB	Act GB	Deconf GB	Firm GB	Curr Avail GB	Pend Avail GB
PowerPC_POWER8	80	26	0	21.64	21.64	0	26	22	28	3,072.00	1,536.00	0.00	23.00	1,447.00	1,447.00
PowerPC_POWER8	80	26	0	0.31	0.31	0	26	52	39	3,072.00	1,536.00	0.00	38.50	967.50	967.50

Look at Firm GB in HMCScanner under System Summary Tab
Latest is 0.11.35
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power%20Systems/page/HMC%20Scanner

Memory Planning

http://www.circle4.com/ptechu/memoryplan.xlsx
Note div - use 64 for all pre p7+ and IBM I, – 128 for p7+ and p8



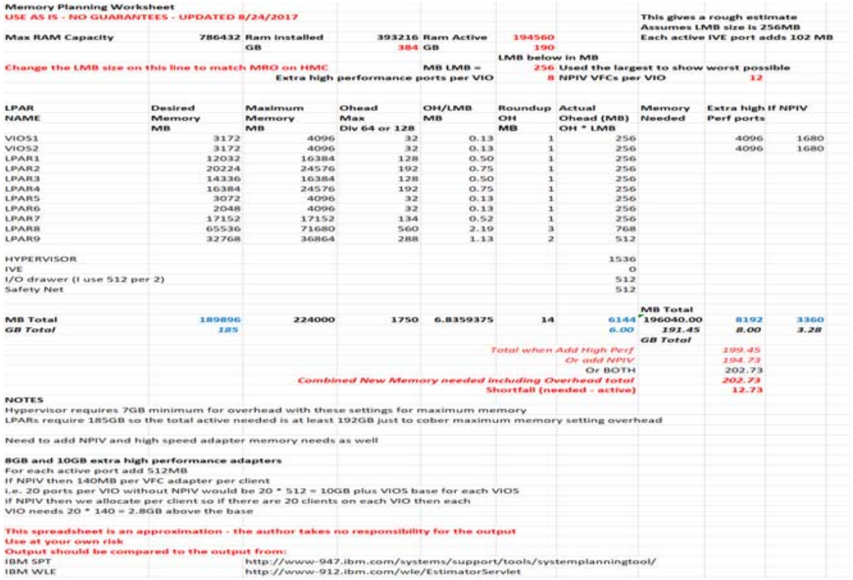
POWER Systems Memory Overhead Approximation Calculator
USE AS IS - NO GUARANTEES - UPDATED 8/24/2017
Complete the information below so that calculations will be accurate

Memory Installed in box in MB	393216	
Memory active in box in MB	194560	
LMB size for server	256	
Extra High performance adapter ports per VIO These include 10Gb network and 8Gb fibre	8	This is active 10Gb net, 8gb fibre etc ports (not adapters)
VFCs (NPIV) per VIO server	12	Each NPIV client
I/O drawers attached	2	
POWER6 only - IVE/HEA ports active	0	Change to number of ports in use
safety net for memory in MB	512	
Active memory mirroring?	2	Set to 2 if using mirroring
Divisor	128	Set to 128 if p7+ or P8

Spreadsheet assumes 2 x VIO servers configured equally

This spreadsheet is an approximation - the author takes no responsibility for the output
Use at your own risk
Output should be compared to the output from:
IBM SPT http://www-947.ibm.com/systems/support/tools/systemplanningtool/
IBM WLE http://www-912.ibm.com/wle/EstimatorServlet
Questions can be sent to jaqui@circle4.com

Cover Sheet



Memory Planning Worksheet
USE AS IS - NO GUARANTEES - UPDATED 8/24/2017

This gives a rough estimate
Assumes LMB size is 256MB
Each active IVE port adds 102 MB

Max RAM Capacity 786432 Ram installed 393216 Ram Active 194560
GB 384 GB 190
LMB below in MB
Change the LMB size on this line to match MRO on HMC MB LMB = 256 Used the largest to show worst possible
Extra high performance ports per VIO 8 NPIV VFCs per VIO 12

LPAR NAME	Desired Memory MB	Maximum Memory MB	Ohead Max Div 64 or 128	OH/LMB MB	Roundup OH MB	Actual Ohead (MB) OH * LMB	Memory Needed	Extra high Perf ports	If NPIV
VIOS1	3172	4096	32	0.13	1	256		4096	1680
VIOS2	3172	4096	32	0.13	1	256		4096	1680
LPAR1	12032	16384	128	0.50	1	256			
LPAR2	20224	24576	192	0.75	1	256			
LPAR3	14336	16384	128	0.50	1	256			
LPAR4	16384	24576	192	0.75	1	256			
LPAR5	3072	4096	32	0.13	1	256			
LPAR6	2048	4096	32	0.13	1	256			
LPAR7	17152	17152	134	0.52	1	256			
LPAR8	65536	71680	560	2.19	3	768			
LPAR9	32768	36864	288	1.13	2	512			
HYPERVISOR						1536			
IVE						0			
I/O drawer (I use 512 per 2)						512			
Safety Net						512			
							MB Total		
MB Total	189896	224000	1750	6.8359375	14	6144	196040.00	8192	3360
GB Total	*185*					*6.00*	*191.45*	*8.00*	*3.28*
							GB Total		
						Total when Add High Perf		*199.45*	
						Or add NPIV		*194.73*	
						Or BOTH		202.73	
						Combined New Memory needed including Overhead total		*202.73*	
						Shortfall (needed - active)		12.73	

NOTES
Hypervisor requires 7GB minimum for overhead with these settings for maximum memory
LPARs require 185GB so the total active needed is at least 192GB just to cober maximum memory setting overhead

Need to add NPIV and high speed adapter memory needs as well

8GB and 10GB extra high performance adapters
For each active port add 512MB
If NPIV then 140MB per VFC adapter per client
i.e. 20 ports per VIO without NPIV would be 20 * 512 = 10GB plus VIOS base for each VIOS
if NPIV then we allocate per client so if there are 20 clients on each VIO then each
VIO needs 20 * 140 = 2.8GB above the base

This spreadsheet is an approximation - the author takes no responsibility for the output
Use at your own risk
Output should be compared to the output from:
IBM SPT http://www-947.ibm.com/systems/support/tools/systemplanningtool/
IBM WLE http://www-912.ibm.com/wle/EstimatorServlet

Actual Data

Memory with lru_file_repage=0

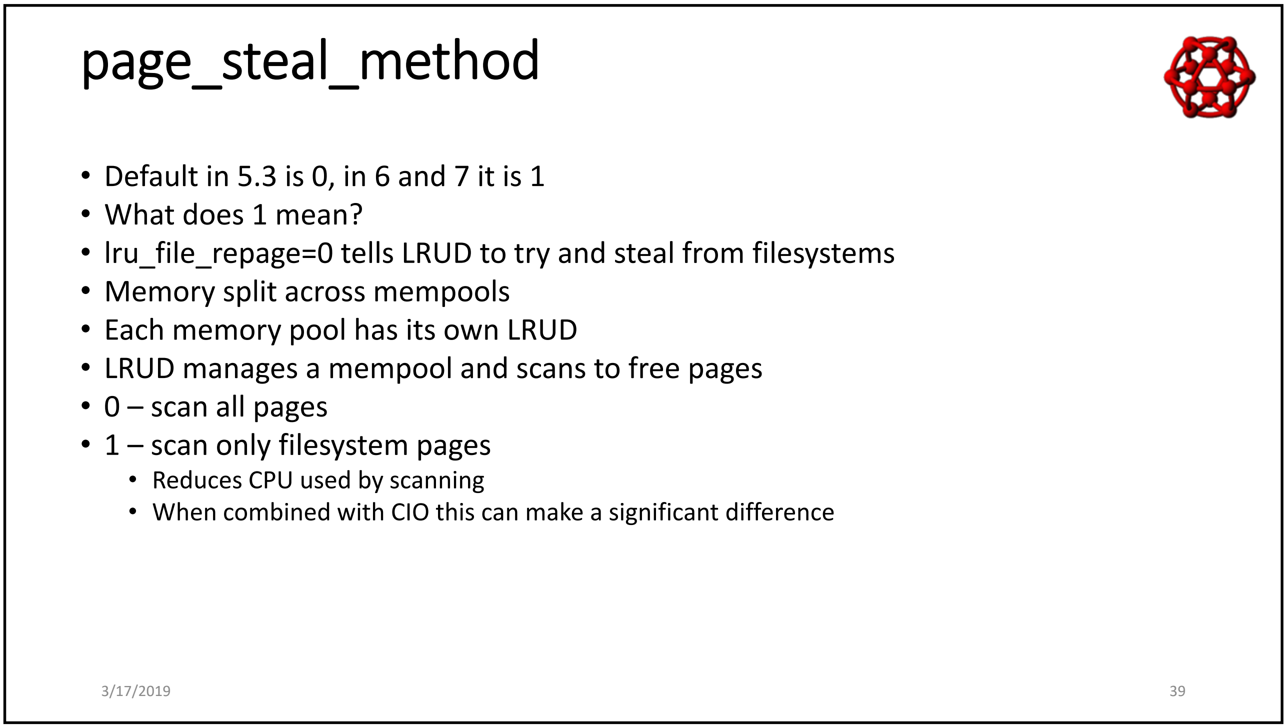
- minperm=3
  - Always try to steal from filesystems if filesystems are using more than 3% of memory
- maxperm=90
  - Soft cap on the amount of memory that filesystems or network can use
  - Superset so includes things covered in maxclient as well
- maxclient=90
  - Hard cap on amount of memory that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

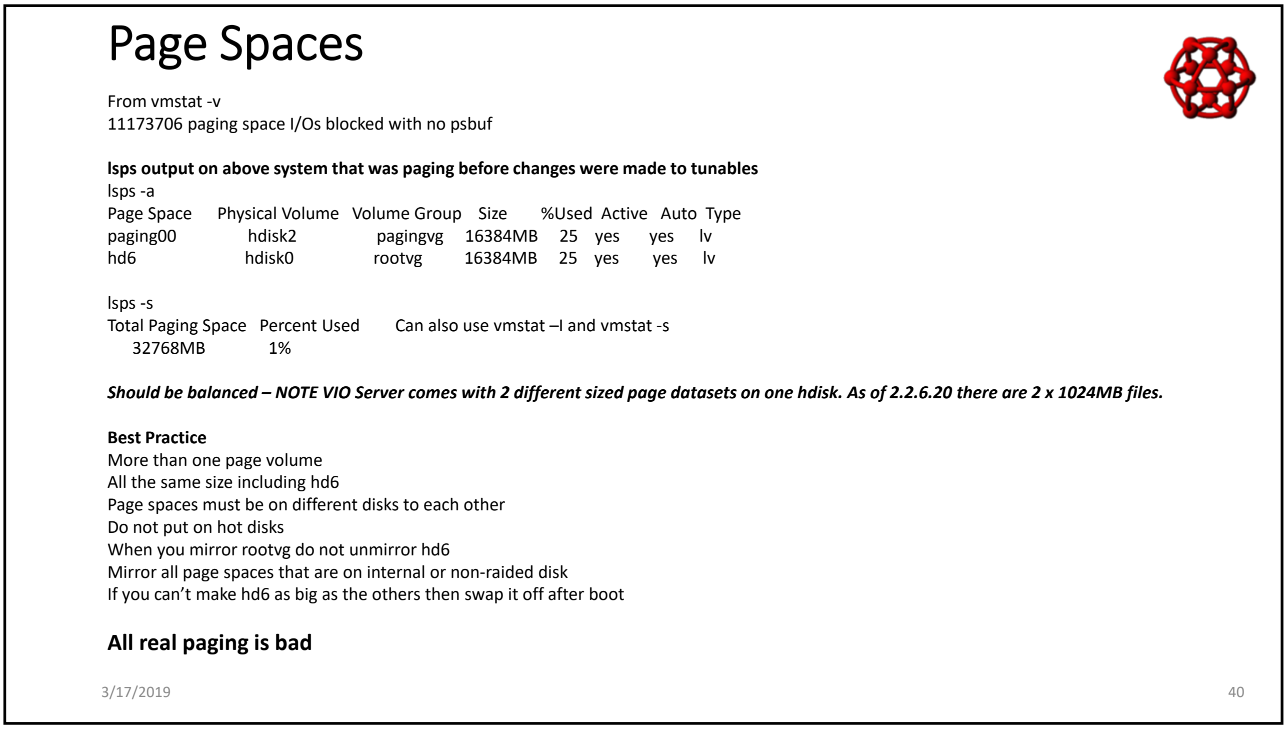
On v6.1 and v7 they are set by default

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
- Each memory pool has its own LRUD
- LRUD manages a mempool and scans to free pages
- 0 – scan all pages
- 1 – scan only filesystem pages
  - Reduces CPU used by scanning
  - When combined with CIO this can make a significant difference



Page Spaces

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
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
     32768MB              1%
```

Should be balanced – NOTE VIO Server comes with 2 different sized page datasets on one hdisk. As of 2.2.6.20 there are 2 x 1024MB files.

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
When you mirror rootvg do not unmirror hd6
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 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
  - Shows as %comp in nmon analyser
  - Shows 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

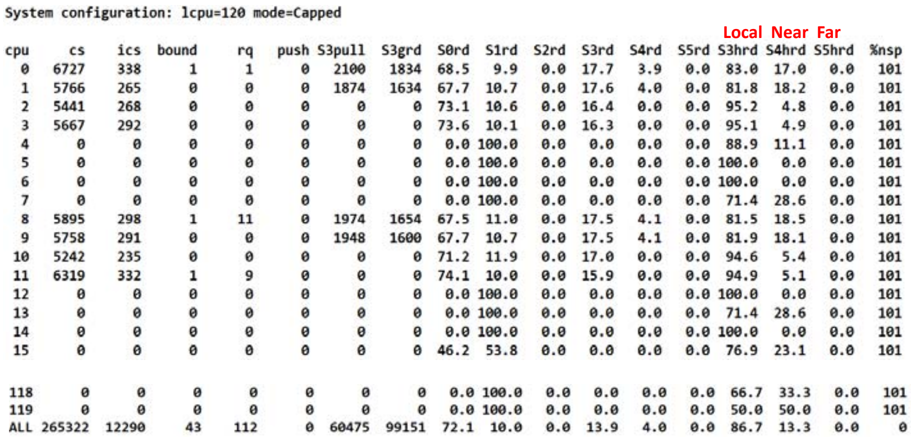
Looking for Problems

- lssrad –av
- mpstat –d
- topas –M
- vmstat –v
  - Look at blocked I/Os and computational memory used
- svmon
  - Try –G –O unit=auto,timestamp=on,pgsz=on,affinity=detail options
  - Look at Domain affinity section of the report
- Etc etc

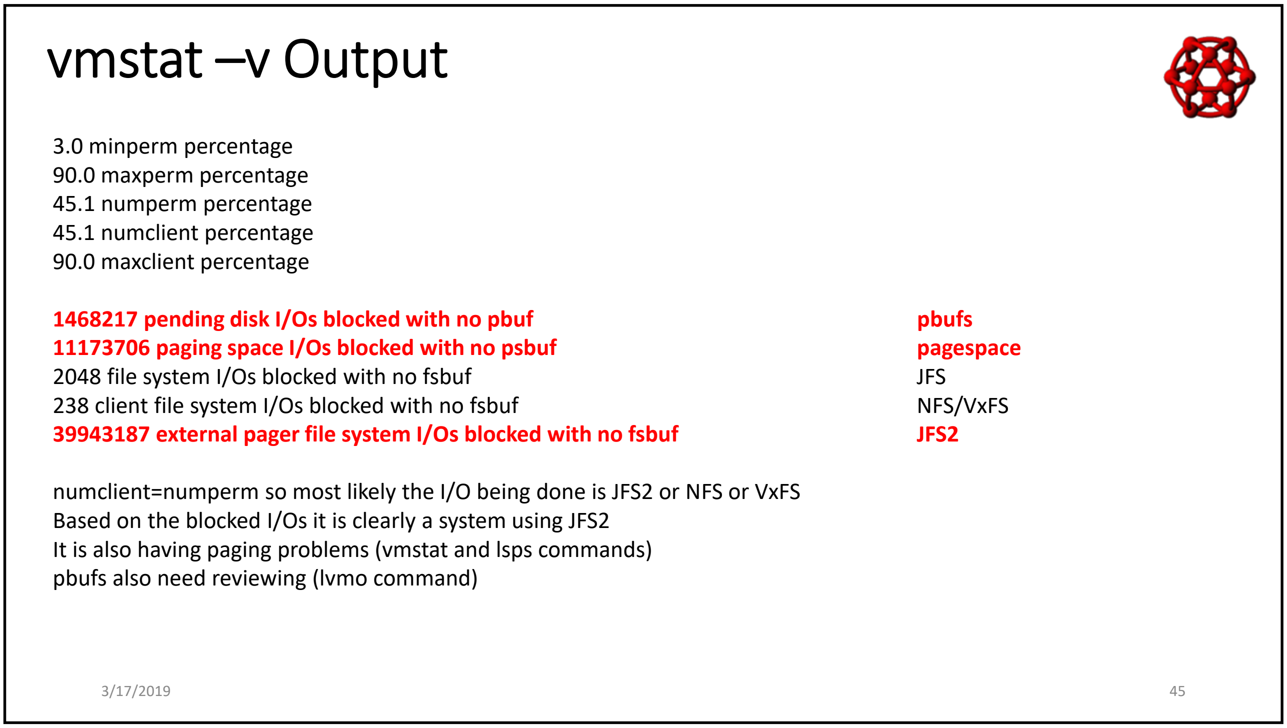
Affinity

- LOCAL SRAD
  - within the same chip, shows as s3
- NEAR SRAD
  - within the same node (if single node) or intra-node if multi-node (think E980), shows as s4
- FAR SRAD
  - on another node – inter-node, shows as s5
- Command is lssrad –av or you 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, LPAR memory sizing and LPAR bring-up order

E880 mpstat –d SMT8



```
System configuration: lcpu=120 mode=Capped
                                                                                          Local  Near  Far
cpu     cs    ics  bound     rq   push S3pull  S3grd  S0rd   S1rd  S2rd  S3rd  S4rd  S5rd S3hrd S4hrd S5hrd  %nsp
  0   6727    338      1      1      0   2100   1834  68.5    9.9   0.0  17.7   3.9   0.0  83.0  17.0   0.0   101
  1   5766    265      0      0      0   1874   1634  67.7   10.7   0.0  17.6   4.0   0.0  81.8  18.2   0.0   101
  2   5441    268      0      0      0      0      0  73.1   10.6   0.0  16.4   0.0   0.0  95.2   4.8   0.0   101
  3   5667    292      0      0      0      0      0  73.6   10.1   0.0  16.3   0.0   0.0  95.1   4.9   0.0   101
  4      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0  88.9  11.1   0.0   101
  5      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0 100.0   0.0   0.0   101
  6      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0 100.0   0.0   0.0   101
  7      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0  71.4  28.6   0.0   101
  8   5895    298      1     11      0   1974   1654  67.5   11.0   0.0  17.5   4.1   0.0  81.5  18.5   0.0   101
  9   5758    291      0      0      0   1948   1600  67.7   10.7   0.0  17.5   4.1   0.0  81.9  18.1   0.0   101
 10   5242    235      0      0      0      0      0  71.2   11.9   0.0  17.0   0.0   0.0  94.6   5.4   0.0   101
 11   6319    332      1      9      0      0      0  74.1   10.0   0.0  15.9   0.0   0.0  94.9   5.1   0.0   101
 12      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0 100.0   0.0   0.0   101
 13      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0  71.4  28.6   0.0   101
 14      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0 100.0   0.0   0.0   101
 15      0      0      0      0      0      0      0  46.2   53.8   0.0   0.0   0.0   0.0  76.9  23.1   0.0   101

118      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0  66.7  33.3   0.0   101
119      0      0      0      0      0      0      0   0.0  100.0   0.0   0.0   0.0   0.0  50.0  50.0   0.0   101
ALL 265322  12290     43    112      0  60475  99151  72.1   10.0   0.0  13.9   4.0   0.0  86.7  13.3   0.0     0
```

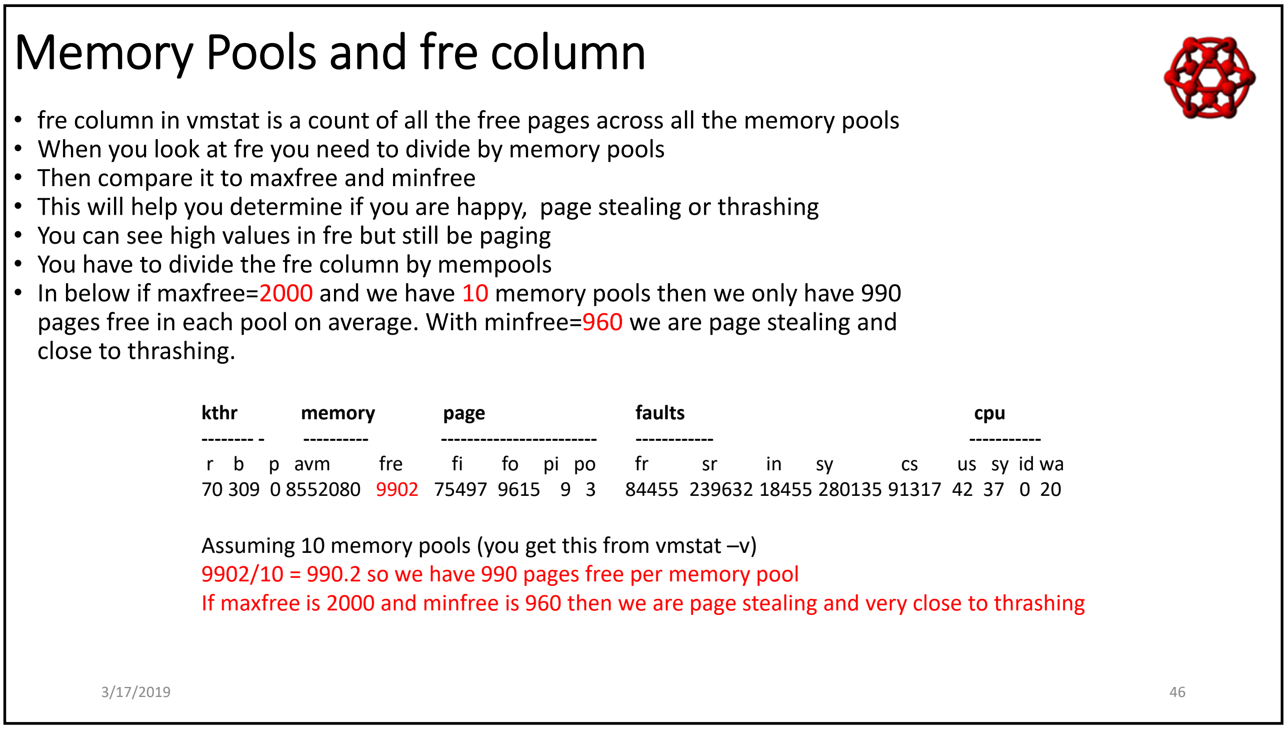


vmstat –v Output

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 (vmstat and lsps commands)
pbufs also need reviewing (lvmo command)



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 per 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.

I/O

Kernel I/O Layers

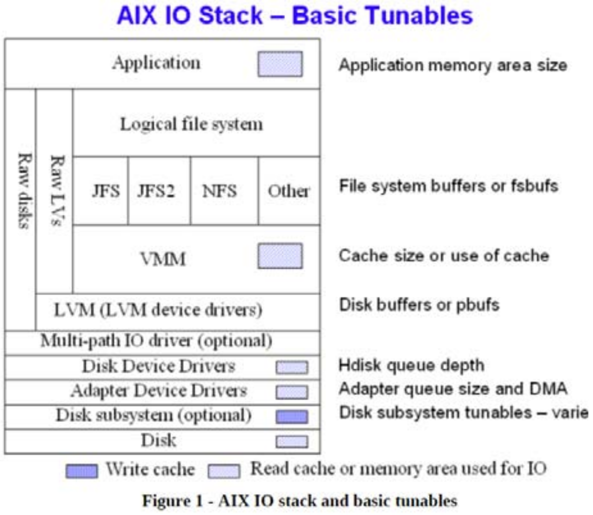
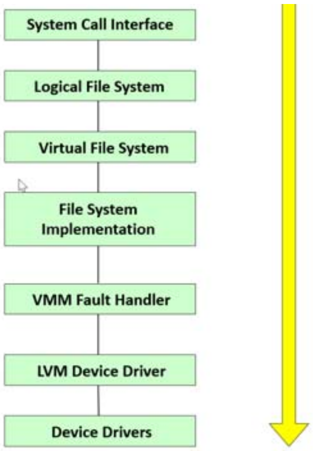

AIX IO Stack – Basic Tunables
Application
Logical file system
Raw disks
Raw LV's
JFS
JFS2
NFS
Other
VMM
LVM (LVM device drivers)
Multi-path IO driver (optional)
Disk Device Drivers
Adapter Device Drivers
Disk subsystem (optional)
Disk
Application memory area size
File system buffers or fsbufs
Cache size or use of cache
Disk buffers or pbufs
Hdisk queue depth
Adapter queue size and DMA
Disk subsystem tunables – varie
Write cache
Read cache or memory area used for IO

Figure 1 - AIX IO stack and basic tunables



I/O request

Logical and Virtual Layers form a single mount homogenous view

JFS, JFS2

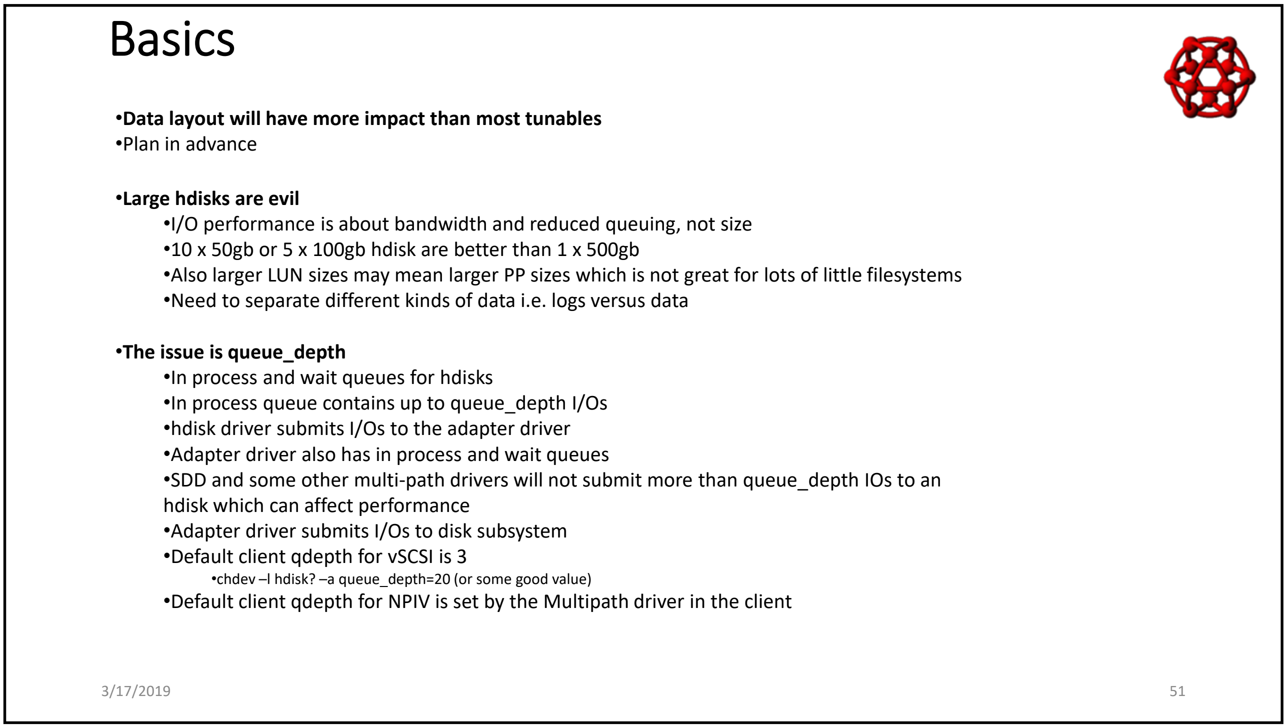
VMM

LVM

Physical I/O operation

Rough Anatomy of an I/O

- LVM requests a PBUF
  - Pinned memory buffer to hold I/O request in LVM layer
- Then placed into an FSBUF
  - 3 types
  - These are also pinned
  - Filesystem JFS
  - Client NFS and VxFS
  - External Pager JFS2
- If paging then need PSBUFs (also pinned)
  - Used for I/O requests to and from page space
- Then queue I/O to an hdisk (queue_depth)
- Then queue it to an adapter (num_cmd_elems)
- Adapter queues it to the disk subsystem
- Additionally, every 60 seconds the sync daemon (syncd) runs to flush dirty I/O out to filesystems or page space

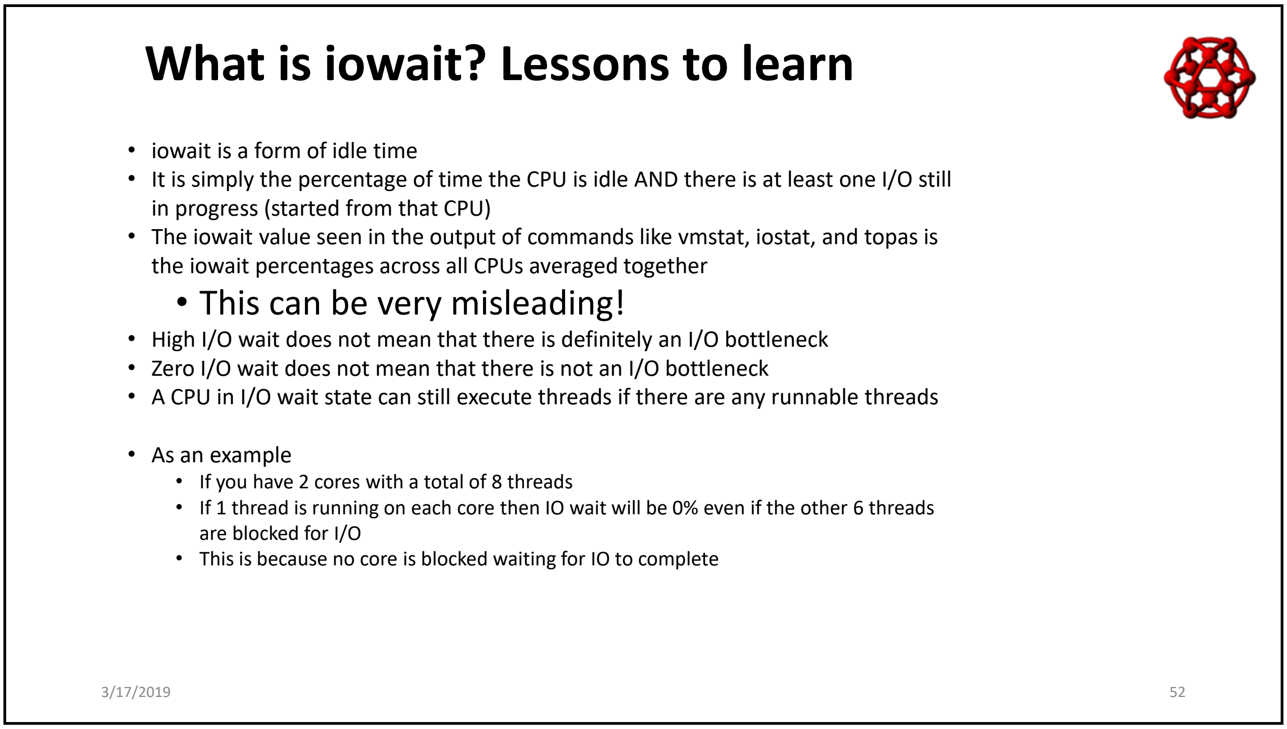


Basics

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

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

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



What is iowait? Lessons to learn

- iowait is a form of idle time
- It is simply the percentage of time the CPU is idle AND there is at least one I/O still in progress (started from that CPU)
- The iowait value seen in the output of commands like vmstat, iostat, and topas is the iowait percentages across all CPUs averaged together
  - This can be very misleading!
- High I/O wait does not mean that there is definitely an I/O bottleneck
- Zero I/O wait does not mean that there is not an I/O bottleneck
- A CPU in I/O wait state can still execute threads if there are any runnable threads

- As an example
  - If you have 2 cores with a total of 8 threads
  - If 1 thread is running on each core then IO wait will be 0% even if the other 6 threads are blocked for I/O
  - This is because no core is blocked waiting for IO to complete

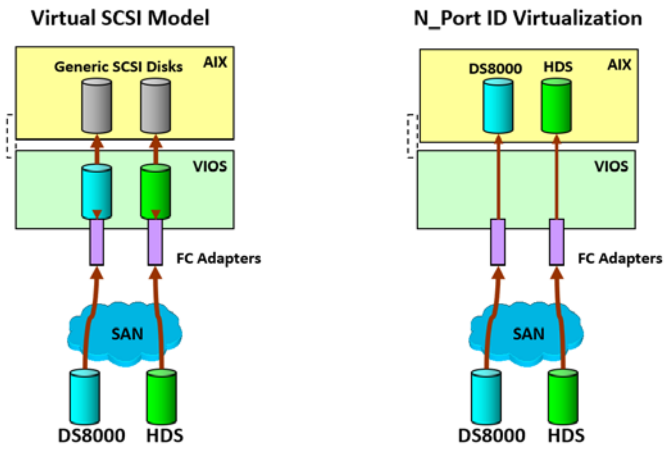
queue depth

- •Disk and adapter drivers each have a queue to handle I/O
- •Note that disks also have a max_xfer_size that you may want to change
- •Queues are split into in-service (aka in-flight) and wait queues
- •IO requests in the in-service queue are sent to storage and the queue slot is freed when the IO is complete
- •IO requests in the wait queue stay there till an in-service slot is free

- •queue depth is the size of the in-service queue for the hdisk
  - •Default for vSCSI hdisk is 3
  - •Default for NPIV or direct attach depends on the HAK (host attach kit) or MPIO drivers used
- •num_cmd_elems is the size of the in-service queue for the HBA

- •Maximum in-flight IOs submitted to the SAN is the smallest of:
  - •Sum of hdisk queue depths
  - •Sum of the HBA num_cmd_elems
  - •Maximum in-flight IOs submitted by the application
- •For HBAs
  - •num_cmd_elems defaults to 200 typically
  - •Max range is 2048 to 4096 depending on storage vendor
  - •As of AIX v7.1 tl2 (or 6.1 tl8) num_cmd_elems is limited to 256 for VFCs
    - •See http://www-01.ibm.com/support/docview.wss?uid=isg1IV63282

Disk Adapter Sharing Options



N_Port ID Virtualization

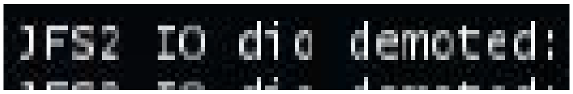
Multiple Virtual World Wide Port Names per FC port – PCIe 8 Gb adapter
LPARs have direct visibility on SAN (Zoning/Masking)
I/O Virtualization configuration effort is reduced
Drivers must be in client LPAR

Demoted I/O

- CIO write fails because IO is not aligned to FS blocksize
  - i.e app writing 512 byte blocks but FS has 4096
- Ends up getting redone
  - Demoted I/O consumes more kernel CPU
  - And more physical I/O
- To find demoted I/O (if JFS2)

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

Look in the report for:

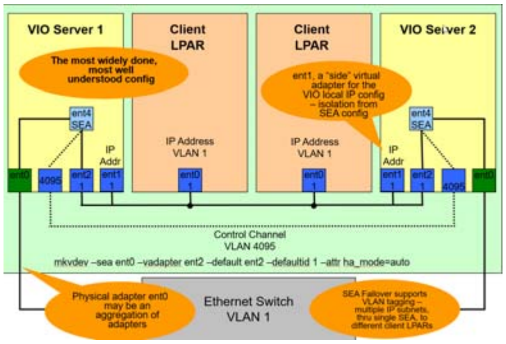


```
JFS2 IO dio demoted:
```

Network

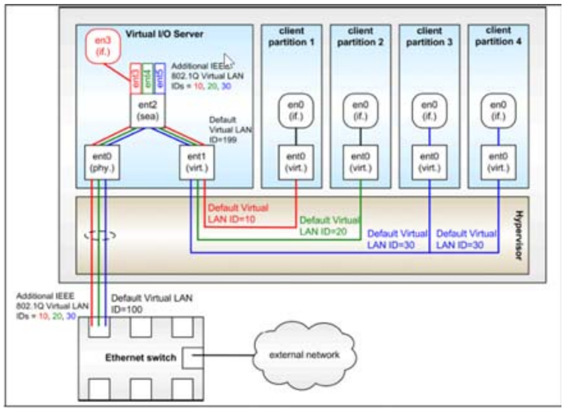
Network – Virtual Ethernet and SEA

SEA Failover – no VLAN Tagging


VIO Server 1
Client LPAR
Client LPAR
VIO Server 2
The most widely done, most well understood config
ent1, a "side" virtual adapter for the VIO local IP config – isolation from SEA config
ent4 SEA
ent4 SEA
IP Address VLAN 1
IP Address VLAN 1
IP Addr
IP Addr
ent0
4095
ent2 1
ent1 1
ent0 1
ent0 1
ent1 1
ent2 1
4095
ent0
Control Channel VLAN 4095
Physical adapter ent0 may be an aggregation of adapters
Ethernet Switch VLAN 1
SEA Failover supports VLAN tagging – multiple IP subnets, thru single SEA, to different client LPARs

```
mkvdev –sea ent0 –vadapter ent2 –default ent2 –defaultid 1 –attr ha_mode=auto
```

SEA with VLAN Tagging



As of VIO 2.2.3 we no longer define the control channel
It now defaults to VLAN 4095

Adapter Settings

See backup slides for definitions
ifconfig -a output

```
en0: flags=1e080863,480<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
        inet 10.2.0.37 netmask 0xfffffe00 broadcast 10.2.1.255
         tcp_sendspace 65536 tcp_recvspace 65536 rfc1323 0
lo0: flags=e08084b<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64BIT>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1/0
         tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

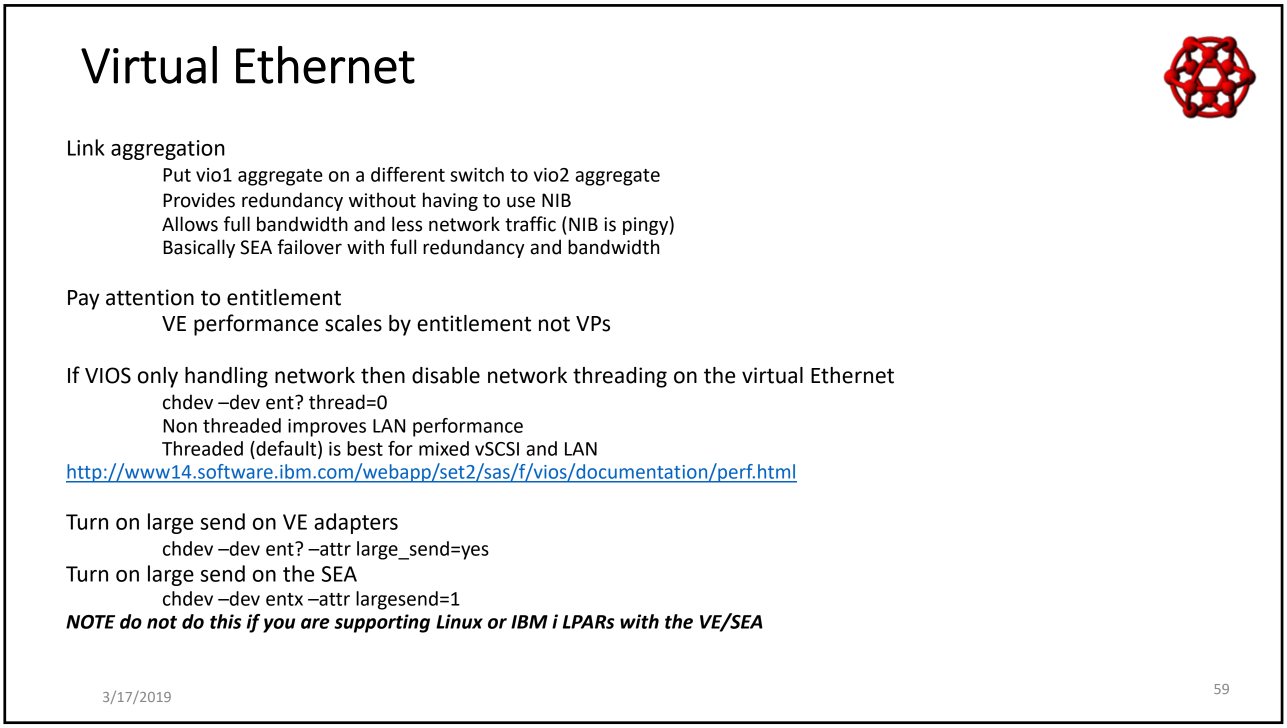
These override no, so they will need to be set at the adapter.
Additionally you will want to ensure you set the adapter to the correct setting if it runs at less than GB, rather than allowing auto-negotiate
Can also check with: lsattr –El ent0 and lsattr –El en0

Stop inetd and use chdev to reset adapter (i.e. en0)
Or use chdev with the –P and the changes will come in at the next reboot

```
chdev -l en0 -a tcp_recvspace=262144 –a tcp_sendspace=262144 –a rfc1323=1 –P
```

On a VIO server I normally bump the transmit queues on the real (underlying adapters) for the aggregate/SEA
Example for a 1Gbe adapter:

```
chdev -l ent? -a txdesc_que_sz=1024 -a tx_que_sz=16384 -P
```



Virtual Ethernet

Link aggregation

- Put vio1 aggregate on a different switch to vio2 aggregate
- Provides redundancy without having to use NIB
- Allows full bandwidth and less network traffic (NIB is pingy)
- Basically SEA failover with full redundancy and bandwidth

Pay attention to entitlement

- VE performance scales by entitlement not VPs

If VIOS only handling network then disable network threading on the virtual Ethernet

```
chdev –dev ent? thread=0
```

- Non threaded improves LAN performance
- Threaded (default) is best for mixed vSCSI and LAN

http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/perf.html

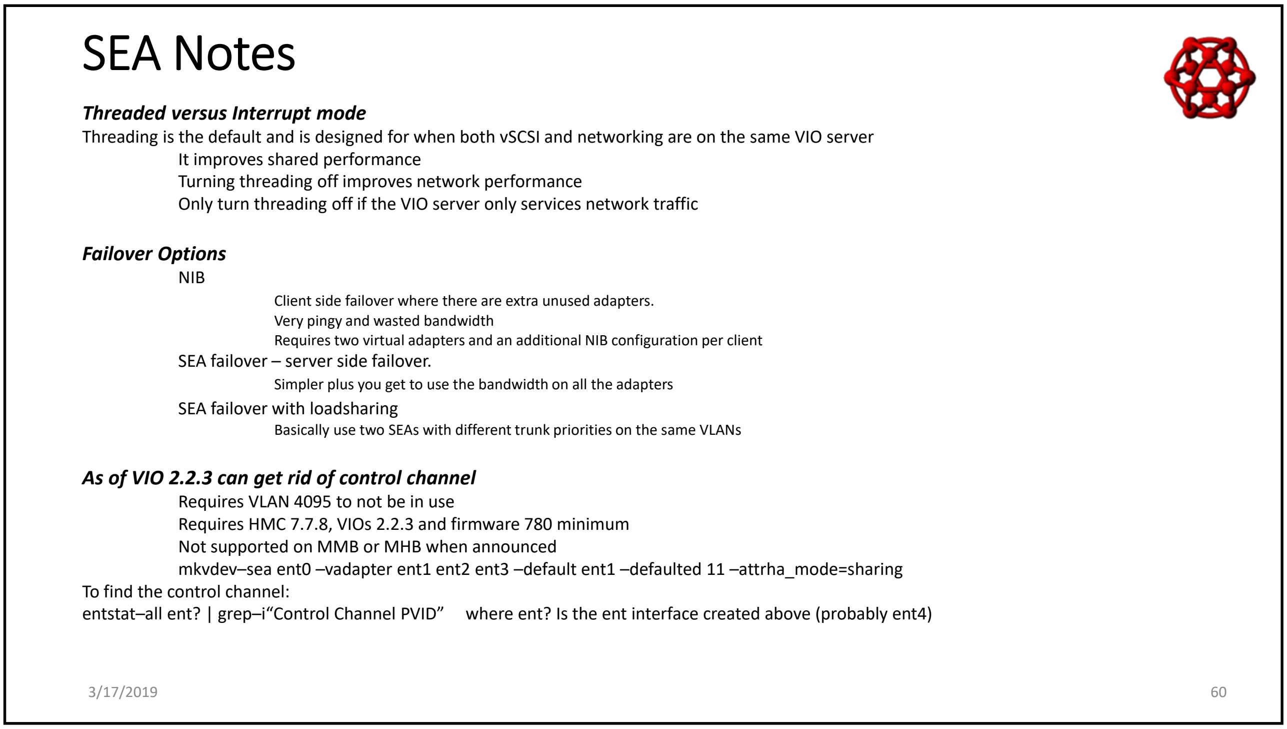
Turn on large send on VE adapters

```
chdev –dev ent? –attr large_send=yes
```

Turn on large send on the SEA

```
chdev –dev entx –attr largesend=1
```

NOTE do not do this if you are supporting Linux or IBM i LPARs with the VE/SEA



SEA Notes

Threaded versus Interrupt mode

Threading is the default and is designed for when both vSCSI and networking are on the same VIO server

- It improves shared performance
- Turning threading off improves network performance
- Only turn threading off if the VIO server only services network traffic

Failover Options

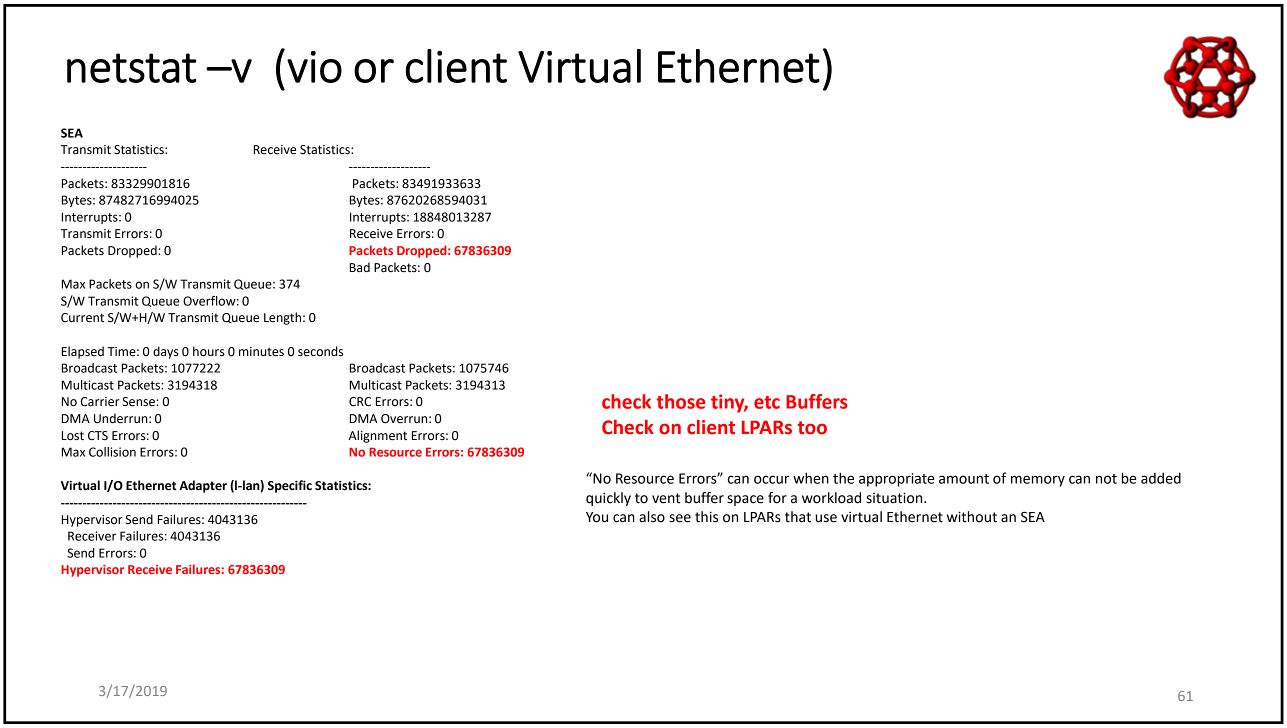
- NIB
  - Client side failover where there are extra unused adapters.
  - Very pingy and wasted bandwidth
  - Requires two virtual adapters and an additional NIB configuration per client
- SEA failover – server side failover.
  - Simpler plus you get to use the bandwidth on all the adapters
- SEA failover with loadsharing
  - Basically use two SEAs with different trunk priorities on the same VLANs

As of VIO 2.2.3 can get rid of control channel

- Requires VLAN 4095 to not be in use
- Requires HMC 7.7.8, VIOs 2.2.3 and firmware 780 minimum
- Not supported on MMB or MHB when announced
- mkvdev–sea ent0 –vadapter ent1 ent2 ent3 –default ent1 –defaulted 11 –attrha_mode=sharing

To find the control channel:

entstat–all ent? | grep–i“Control Channel PVID”   where ent? Is the ent interface created above (probably ent4)



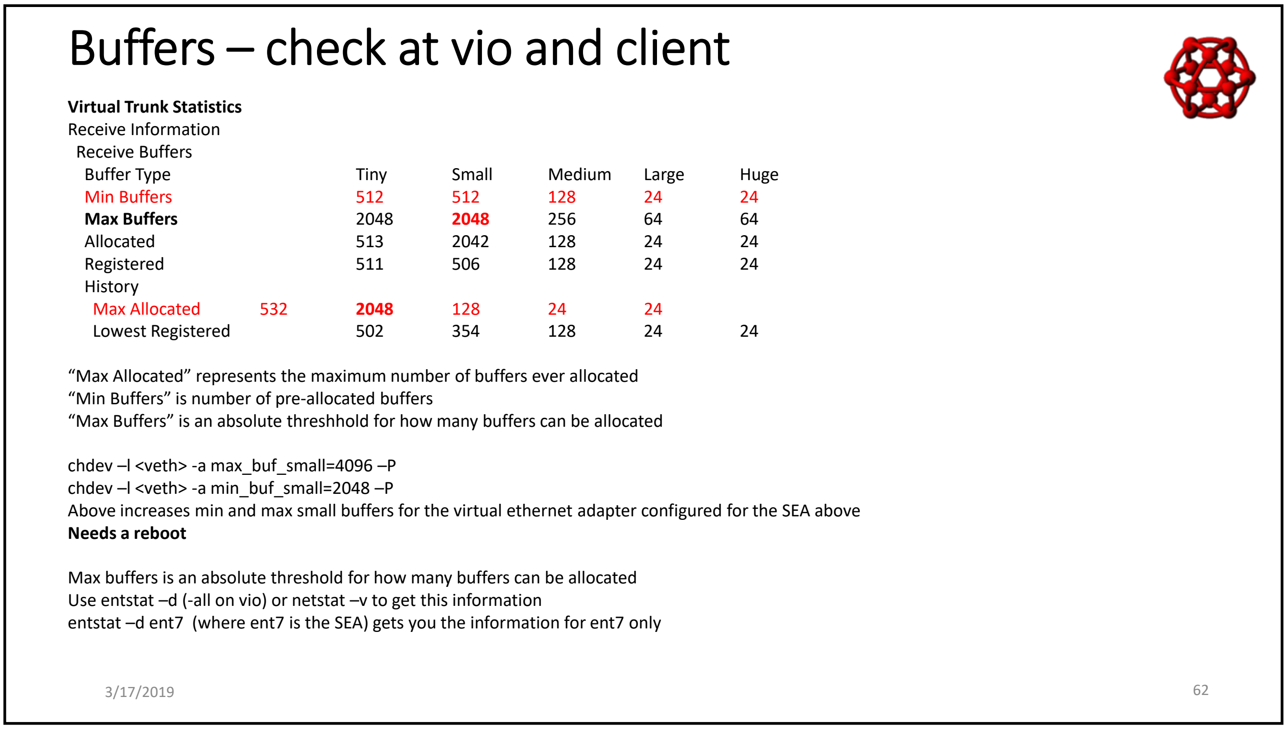
netstat –v (vio or client Virtual Ethernet)

```
SEA
Transmit Statistics:                  Receive Statistics:
--------------------                  -------------------
Packets: 83329901816                  Packets: 83491933633
Bytes: 87482716994025                 Bytes: 87620268594031
Interrupts: 0                         Interrupts: 18848013287
Transmit Errors: 0                    Receive Errors: 0
Packets Dropped: 0                    Packets Dropped: 67836309
                                      Bad Packets: 0
Max Packets on S/W Transmit Queue: 374
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0

Elapsed Time: 0 days 0 hours 0 minutes 0 seconds
Broadcast Packets: 1077222            Broadcast Packets: 1075746
Multicast Packets: 3194318            Multicast Packets: 3194313
No Carrier Sense: 0                   CRC Errors: 0
DMA Underrun: 0                       DMA Overrun: 0
Lost CTS Errors: 0                    Alignment Errors: 0
Max Collision Errors: 0               No Resource Errors: 67836309

Virtual I/O Ethernet Adapter (l-lan) Specific Statistics:
---------------------------------------------------------
Hypervisor Send Failures: 4043136
  Receiver Failures: 4043136
  Send Errors: 0
Hypervisor Receive Failures: 67836309
```

check those tiny, etc Buffers
Check on client LPARs too

“No Resource Errors” can occur when the appropriate amount of memory can not be added quickly to vent buffer space for a workload situation.
You can also see this on LPARs that use virtual Ethernet without an SEA

3/17/2019 61



Buffers – check at vio and client

```
Virtual Trunk Statistics
Receive Information
 Receive Buffers
  Buffer Type                     Tiny        Small       Medium      Large       Huge
  Min Buffers                     512         512         128         24          24
  Max Buffers                     2048        2048        256         64          64
  Allocated                       513         2042        128         24          24
  Registered                      511         506         128         24          24
  History
   Max Allocated         532      2048        128         24          24
   Lowest Registered              502         354         128         24          24
```

“Max Allocated” represents the maximum number of buffers ever allocated
“Min Buffers” is number of pre-allocated buffers
“Max Buffers” is an absolute threshhold for how many buffers can be allocated

chdev –l <veth> -a max_buf_small=4096 –P
chdev –l <veth> -a min_buf_small=2048 –P
Above increases min and max small buffers for the virtual ethernet adapter configured for the SEA above
Needs a reboot

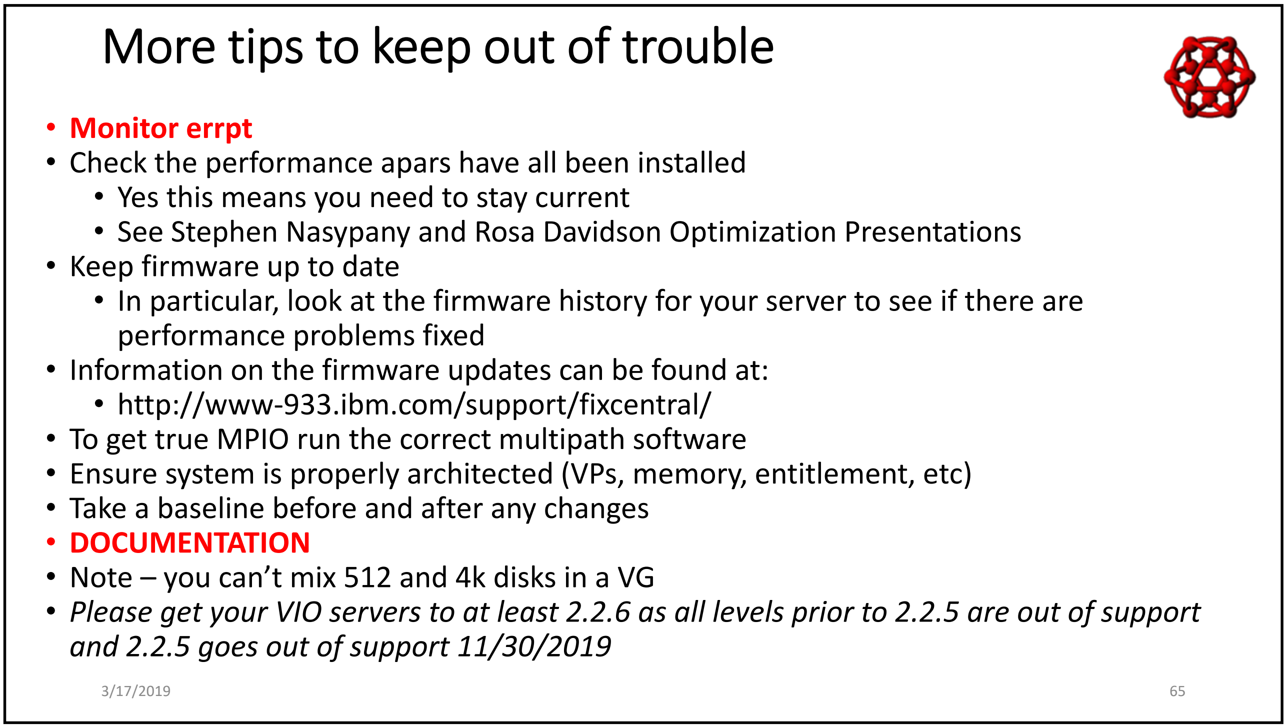
Max buffers is an absolute threshold for how many buffers can be allocated
Use entstat –d (-all on vio) or netstat –v to get this information
entstat –d ent7 (where ent7 is the SEA) gets you the information for ent7 only

3/17/2019 62

Notes

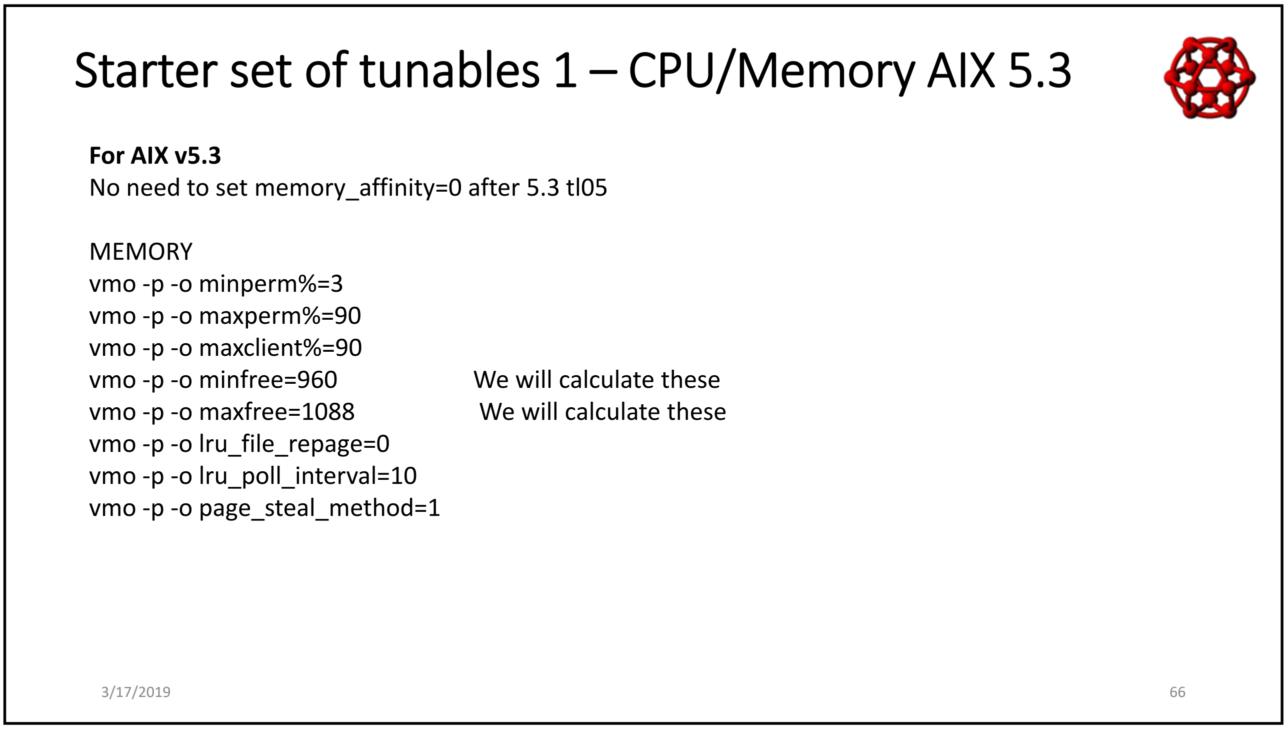
Avoiding Problems

- Stay current
- Known memory issues with 6.1 tl9 sp1 and 7.1 tl3 sp1
- Java JDK8 SR5 is pre-req for best optimization on POWER9
- Java 7.1 SR1 is the preferred minimum level for POWER7 and POWER8
- Java 6 SR7 is minimal on POWER7 but you should go to at least Java 7
- WAS 8.5.2.2
- Ensure the rest of the software stack is current
- Ensure compilers are current and that compiled code turns on optimization
- HMC v8 required for POWER8 – does not support servers prior to POWER6
  - All v8 HMC code is now out of support
  - Upgrade to v9, replacing HMC if necessary
  - HMC V9 R1 M910 supports the following HMC models:
    - X86 : 7042-CR7, 7042-CR8, 7042-CR9, 7042-OE1 and 7042-OE2
    - Open Power: 7063-CR1
  - HMC V9 R1 M910 does not support 7042 models CR2, CR3, CR4, CR5, CR6 , C03, C04, C05, C06, C07 and C08.
- 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)



More tips to keep out of trouble

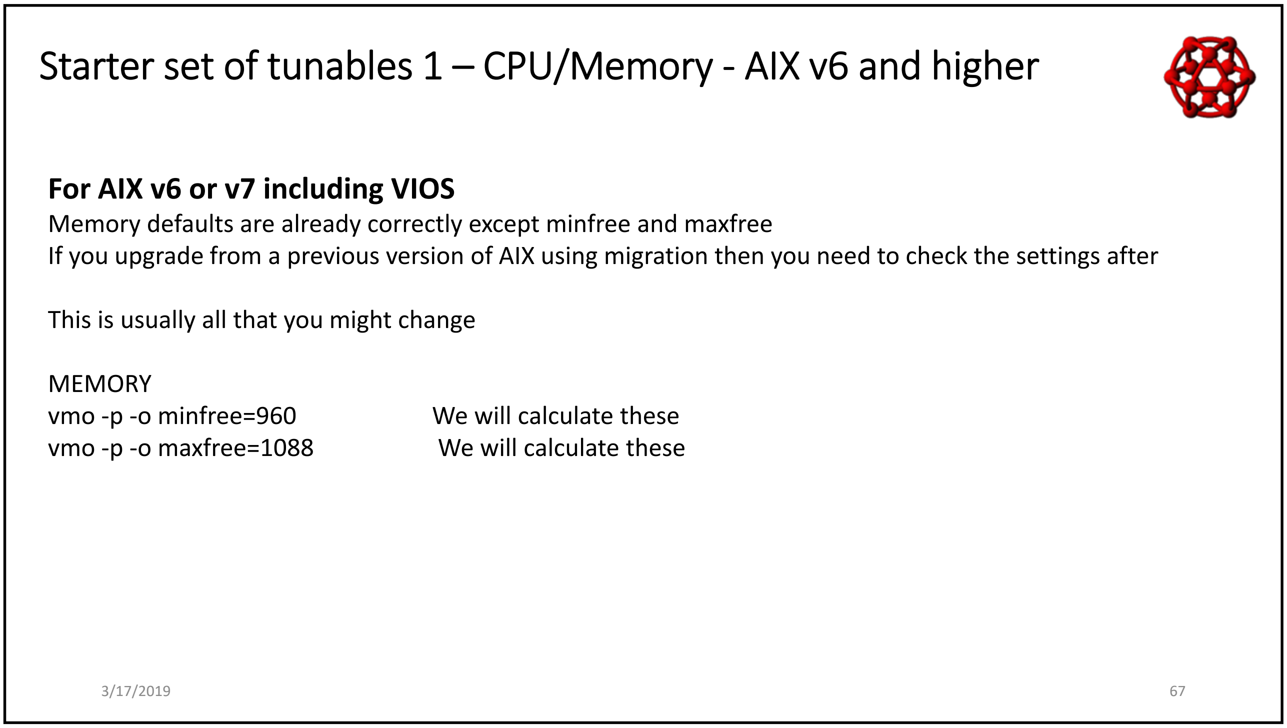
- **Monitor errpt**
- Check the performance apars have all been installed
  - Yes this means you need to stay current
  - See Stephen Nasypany and Rosa Davidson Optimization Presentations
- Keep firmware up to date
  - In particular, look at the firmware history for your server to see if there are performance problems fixed
- Information on the firmware updates can be found at:
  - http://www-933.ibm.com/support/fixcentral/
- To get true MPIO run the correct multipath software
- Ensure system is properly architected (VPs, memory, entitlement, etc)
- Take a baseline before and after any changes
- **DOCUMENTATION**
- Note – you can't mix 512 and 4k disks in a VG
- *Please get your VIO servers to at least 2.2.6 as all levels prior to 2.2.5 are out of support and 2.2.5 goes out of support 11/30/2019*



Starter set of tunables 1 – CPU/Memory AIX 5.3

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
```



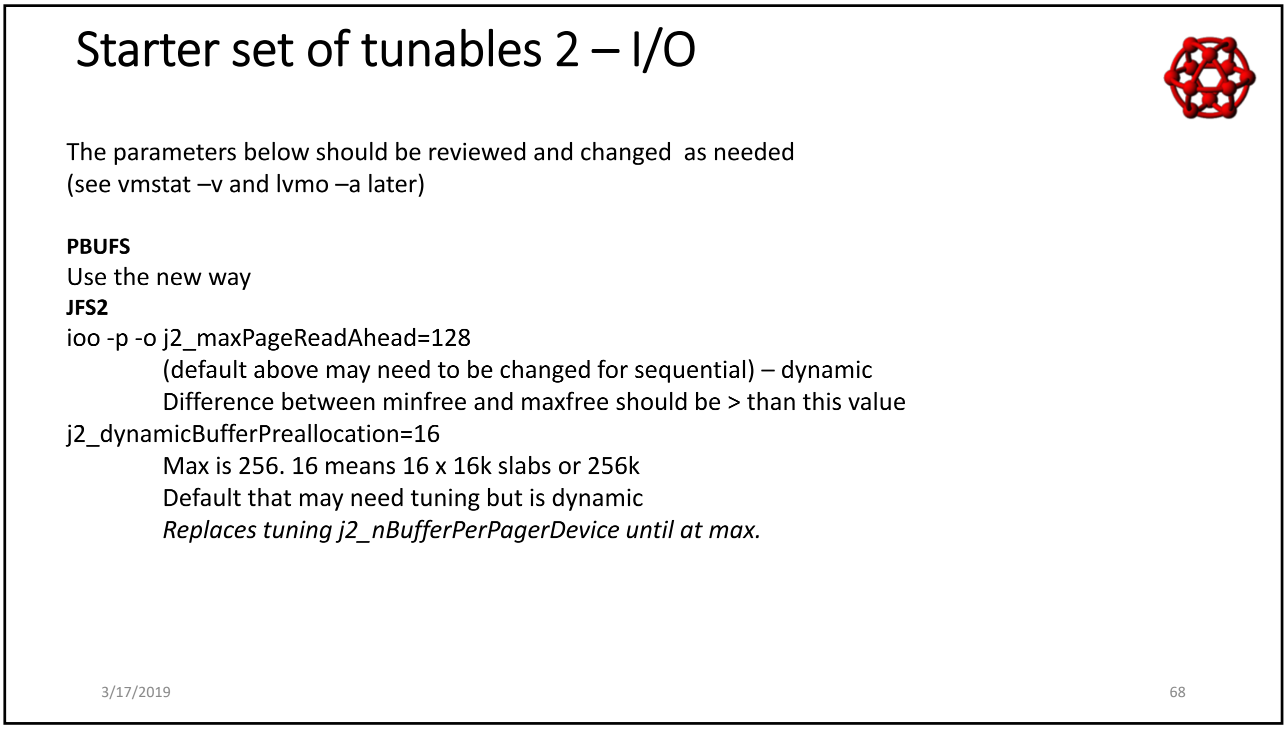
Starter set of tunables 1 – CPU/Memory - AIX v6 and higher

For AIX v6 or v7 including VIOS
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

This is usually all that you might change

MEMORY

```
vmo -p -o minfree=960          We will calculate these
vmo -p -o maxfree=1088          We will calculate these
```



Starter set of tunables 2 – I/O

The parameters below should be reviewed and changed as needed
(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 > than 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.

Starter set of tunables 3 - Network

Typically we set the following for both versions:

NETWORK

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

Also check the actual NIC interfaces and make sure they are set to at least these values
You can’t set udp_sendspace > 65536 as IP has an upper limit of 65536 bytes per packet

Check sb_max is at least 1040000 – increase as needed

Thank you for your time

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

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

Useful Links

- Jaqui Lynch Articles
  - http://www.circle4.com/jaqui/eserver.html
- Nigel Griffiths AIXpert Blog
  - https://www.ibm.com/developerworks/community/blogs/aixpert?lang=en
- Nigel Griffiths Twitter – mr_nmon
  - https://twitter.com/mr_nmon
- Gareth Coates Twitter – power_gaz
  - https://twitter.com/power_gaz
- Jaqui’s 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

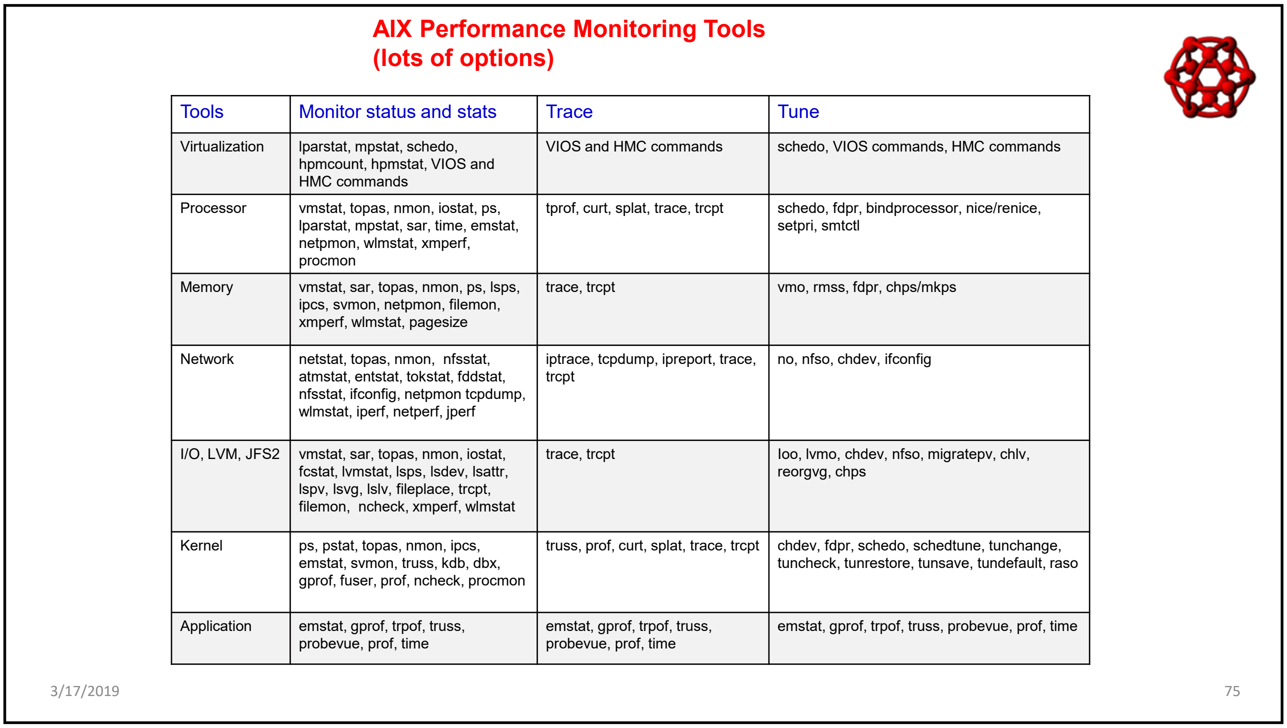
Useful Links

- HMC Scanner
  - https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power%20Systems/page/HMC%20Scanner
- Performance Tools Wiki
  - AIX Performance Tools and Commands
    - https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power%20Systems/page/AIX%20Performance%20Commands
  - Performance Monitoring Tips abd Techniques
    - 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
  - https://www.ibm.com/support/knowledgecenter/TI0002C/p8hcg/p8hcg_part.htm

References

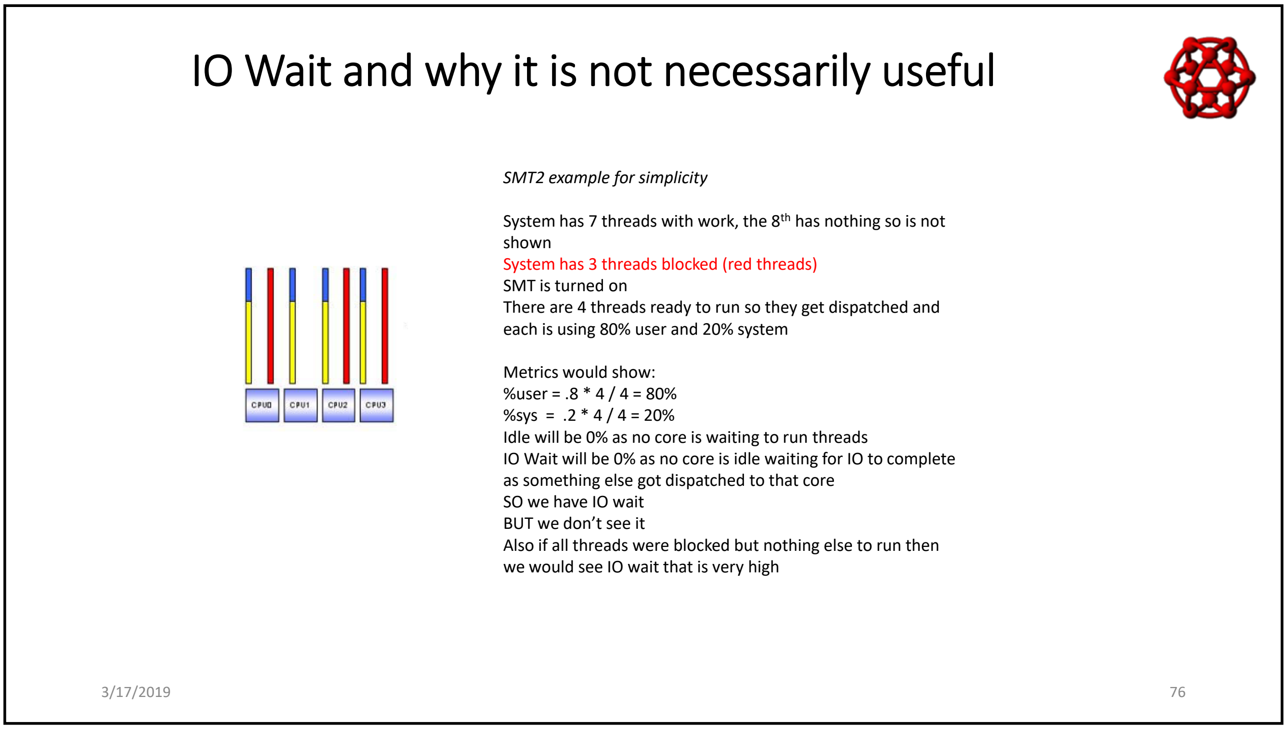
- Technical Introduction and Overview Redbooks
  - Got to http://www.redbooks.com and search for the above redbook for your server
  - As an example the E980 Redbook is:
    - http://www.redbooks.ibm.com/redpapers/pdfs/redp5510.pdf
- Processor Utilization in AIX by Saravanan Devendran
  - https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/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 including POWER8
  - http://www.redbooks.ibm.com/redbooks/pdfs/sg248171.pdf

Backup Slides



AIX Performance Monitoring Tools (lots of options)

Tools	Monitor status and stats	Trace	Tune
Virtualization	lparstat, mpstat, schedo, hpmcount, hpmstat, VIOS and HMC commands	VIOS and HMC commands	schedo, VIOS commands, HMC commands
Processor	vmstat, topas, nmon, iostat, ps, lparstat, mpstat, sar, time, emstat, netpmon, wlmstat, xmperf, procmon	tprof, curt, splat, trace, trcpt	schedo, fdpr, bindprocessor, nice/renice, setpri, smtctl
Memory	vmstat, sar, topas, nmon, ps, lsps, ipcs, svmon, netpmon, filemon, xmperf, wlmstat, pagesize	trace, trcpt	vmo, rmss, fdpr, chps/mkps
Network	netstat, topas, nmon, nfsstat, atmstat, entstat, tokstat, fddstat, nfsstat, ifconfig, netpmon tcpdump, wlmstat, iperf, netperf, jperf	iptrace, tcpdump, ipreport, trace, trcpt	no, nfso, chdev, ifconfig
I/O, LVM, JFS2	vmstat, sar, topas, nmon, iostat, fcstat, lvmstat, lsps, lsdev, lsattr, lspv, lsvg, lslv, fileplace, trcpt, filemon, ncheck, xmperf, wlmstat	trace, trcpt	Ioo, lvmo, chdev, nfso, migratepv, chlv, reorgvg, chps
Kernel	ps, pstat, topas, nmon, ipcs, emstat, svmon, truss, kdb, dbx, gprof, fuser, prof, ncheck, procmon	truss, prof, curt, splat, trace, trcpt	chdev, fdpr, schedo, schedtune, tunchange, tuncheck, tunrestore, tunsave, tundefault, raso
Application	emstat, gprof, trpof, truss, probevue, prof, time	emstat, gprof, trpof, truss, probevue, prof, time	emstat, gprof, trpof, truss, probevue, prof, time


CPU0
CPU1
CPU2
CPU3

IO Wait and why it is not necessarily useful

SMT2 example for simplicity

System has 7 threads with work, the 8th has nothing so is not shown
System has 3 threads blocked (red threads)
SMT is turned on
There are 4 threads ready to run so they get dispatched and each is using 80% user and 20% system

Metrics would show:
%user = .8 * 4 / 4 = 80%
%sys = .2 * 4 / 4 = 20%
Idle will be 0% as no core is waiting to run threads
IO Wait will be 0% as no core is idle waiting for IO to complete as something else got dispatched to that core
SO we have IO wait
BUT we don't see it
Also if all threads were blocked but nothing else to run then we would see IO wait that is very high

Queue Depth

- Try sar –d, nmon –D, iostat -D
- sar –d 2 6 shows:

```
device   %busy   avque   r+w/s   Kbs/s    avwait  avserv

hdisk7     0       0.0       2      160     0.0    1.9
hdisk8    19       0.3     568    14337    23.5    2.3
hdisk9     2       0.0      31      149     0.0    0.9
```

- avque
  Average IOs in the wait queue
  Waiting to get sent to the disk (the disk's queue is full)
  Values > 0 indicate increasing queue_depth may help performance
  Used to mean number of IOs in the disk queue
- avwait
  Average time waiting in the wait queue (ms)
- avserv
  Average I/O service time when sent to disk (ms)
- See articles by Dan Braden:
  - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD105745
  - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106122

Adapter Queue Problems

- Look at BBBF Tab in NMON Analyzer or run fcstat command

- Adapter device drivers use DMA for IO
- From fcstat on each fcs
- NOTE these are since boot

FC SCSI Adapter Driver Information
No DMA Resource Count: 0
No Adapter Elements Count: 2567
No Command Resource Count: 34114051

- No DMA resource – adjust max_xfer_size
- No adapter elements – adjust num_cmd_elems
- No command resource - adjust num_cmd_elems

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

Tunables

- **The tcp_recvspace tunable**
  - The *tcp_recvspace* tunable specifies how many bytes of data the receiving system can buffer in the kernel on the receiving sockets queue.
- **The tcp_sendspace tunable**
  - The *tcp_sendspace* tunable specifies how much data the sending application can buffer in the kernel before the application is blocked on a send call.
- **The rfc1323 tunable**
  - The *rfc1323* tunable enables the TCP window scaling option.
  - By default TCP has a 16 bit limit to use for window size which limits it to 65536 bytes. Setting this to 1 allows for much larger sizes (max is 4GB)
- **The sb_max tunable**
  - The *sb_max* tunable sets an upper limit on the number of socket buffers queued to an individual socket, which controls how much buffer space is consumed by buffers that are queued to a sender's socket or to a receiver's socket. *The tcp_sendspace attribute must specify a socket buffer size less than or equal to the setting of the sb_max attribute*

UDP Send and Receive

udp_sendspace

Set this parameter to 65536, which is large enough to handle the largest possible UDP packet. There is no advantage to setting this value larger

udp_recvspace

Controls the amount of space for incoming data that is queued on each UDP socket. Once the *udp_recvspace* limit is reached for a socket, incoming packets are discarded.

Set this value high as multiple UDP datagrams could arrive and have to wait on a socket for the application to read them. If too low packets are discarded and sender has to retransmit.

Suggested starting value for *udp_recvspace* is 10 times the value of *udp_sendspace*, because UDP may not be able to pass a packet to the application before another one arrives.

Some definitions

- TCP large send offload
  - Allows AIX TCP to build a TCP message up to 64KB long and send It in one call down the stack. The adapter resegments into multiple packets that are sent as either 1500 byte or 9000 byte (jumbo) frames.
  - Without this it takes 44 calls (if MTU 1500) to send 64KB data. With this set it takes 1 call. Reduces CPU. Can reduce network CPU up to 60-75%.
  - It is enabled by default on 10Gb adapters but not on VE or SEA.
- TCP large receive offload
  - Works by aggregating incoming packets from a single stream into a larger buffer before passing up the network stack. Can improve network performance and reduce CPU overhead.
- TCP Checksum Offload
  - Enables the adapter to compute the checksum for transmit and receive. Offloads CPU by between 5 and 15% depending on MTU size and adapter.

Large Receive

- Important note
  - Do not enable on the sea if used by Linux or IBM I client partitions (disabled by default)
  - Do not enable if used by AIX partitions set up for IP forwarding
  - Also called Receive TCP Segment Aggregation
  - If choose to enable this then make sure underlying adapter also has it enabled
  - *See* http://tinyurl.com/gpe5zgd for update on changes for Linux and Large receive
    - *Now supported if VIOS 2.2.4.10 with specific Linux levels*
      - *RHEL 7 rel 2 BE and LE*
      - *SLES 12 SP1*
      - *SLES 11 SP4*
      - *RHEL 6.8*
      - *Ubuntu 16.04 LTS*

Some more definitions

- MTU Size
  - The use of large MTU sizes allows the operating system to send fewer packets of a larger size to reach the same network throughput. The larger packets greatly reduce the processing required in the operating system, assuming the workload allows large messages to be sent. If the workload is only sending small messages, then the larger MTU size will not help. Choice is 1500 or 9000 (jumbo frames). Do not change this without talking to your network team.
- MSS – Maximum Segment Size
  - The largest amount of data, specified in bytes, that a computer or communications device can handle in a single, unfragmented piece.
  - The number of bytes in the data segment and the header must add up to less than the number of bytes in the maximum transmission unit (MTU).
- Computers negotiate MTU size
  - Typical MTU size in TCP for a home computer Internet connection is either 576 or 1500 bytes. Headers are 40 bytes long; the MSS is equal to the difference, either 536 or 1460 bytes.

Network Performance and Throughput

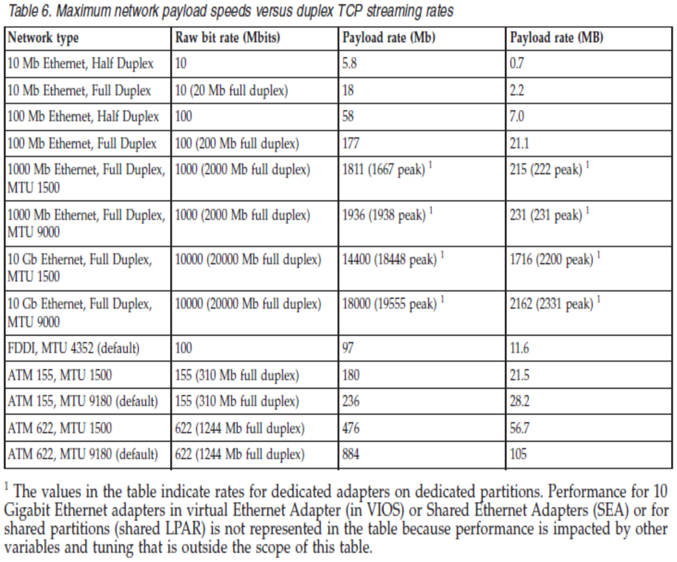


Table 6. Maximum network payload speeds versus duplex TCP streaming rates

Network type	Raw bit rate (Mbits)	Payload rate (Mb)	Payload rate (MB)
10 Mb Ethernet, Half Duplex	10	5.8	0.7
10 Mb Ethernet, Full Duplex	10 (20 Mb full duplex)	18	2.2
100 Mb Ethernet, Half Duplex	100	58	7.0
100 Mb Ethernet, Full Duplex	100 (200 Mb full duplex)	177	21.1
1000 Mb Ethernet, Full Duplex, MTU 1500	1000 (2000 Mb full duplex)	1811 (1667 peak) [1]	215 (222 peak) [1]
1000 Mb Ethernet, Full Duplex, MTU 9000	1000 (2000 Mb full duplex)	1936 (1938 peak) [1]	231 (231 peak) [1]
10 Gb Ethernet, Full Duplex, MTU 1500	10000 (20000 Mb full duplex)	14400 (18448 peak) [1]	1716 (2200 peak) [1]
10 Gb Ethernet, Full Duplex, MTU 9000	10000 (20000 Mb full duplex)	18000 (19555 peak) [1]	2162 (2331 peak) [1]
FDDI, MTU 4352 (default)	100	97	11.6
ATM 155, MTU 1500	155 (310 Mb full duplex)	180	21.5
ATM 155, MTU 9180 (default)	155 (310 Mb full duplex)	236	28.2
ATM 622, MTU 1500	622 (1244 Mb full duplex)	476	56.7
ATM 622, MTU 9180 (default)	622 (1244 Mb full duplex)	884	105

[1] The values in the table indicate rates for dedicated adapters on dedicated partitions. Performance for 10 Gigabit Ethernet adapters in virtual Ethernet Adapter (in VIOS) or Shared Ethernet Adapters (SEA) or for shared partitions (shared LPAR) is not represented in the table because performance is impacted by other variables and tuning that is outside the scope of this table.

AIX v7.1 - http://www-01.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.performance/prftungd_pdf.pdf