AIX General Performance Tuning

Common
Session 560532
Oct 4, 2010 – 2.15pm
Director's Room 1

Jaqui Lynch
Solutions Architect
Forsythe Technology Inc.
lynchj@forsythe.com

FORSYTHE

Agenda

- Page space
- Some Pointers
- Starter set of tunables
- Determining what to set tunables to
- Memory tuning
- Network
- Volume groups and filesystems
- Performance Tools

FORSYTHE

PAGE SPACE

FORSYTHE

Paging Algorithms

- Late Allocation Algorithm (LPSA)
  - The paging space disk blocks are not allocated until corresponding pages in RAM are touched.
- Early Allocation Algorithm (EPSA)
  - This algorithm causes the appropriate number of paging space slots to be allocated at the time the virtual-memory address range is allocated, for example, with the malloc() subroutine. If there are not enough paging space slots to support the malloc() subroutine, an error code is set.
  - To enable EPSA, set the environment variable PSALLOC=early.
- **Deferred Allocation Algorithm (default)**
  - Delays allocation of paging space until it is necessary to page out the
  - page, which results in no wasted paging space allocation. This method can save huge amounts of paging space, which means disk space.

FORSYTHE

lsps –a - (similar to pstat)

- Ensure all page datasets the same size although hd6 can be bigger - ensure more page space than memory
  - Especially if not all page datasets are in rootvg
  - Rootvg page datasets must be big enough to hold the kernel
- Only includes pages allocated (default)
- Use lsps -s to get all pages (includes reserved via early allocation (PSALLOC=early)
- Use multiple page datasets on multiple disks
  - Parallelism

FORSYTHE

Correcting Paging

11173706 paging space I/Os blocked with no psbuf (from vmstat –v)

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

What you want to see
lsps -a

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
paging01	hdisk3	pagingvg	16384MB	1	yes	yes	lv
paging00	hdisk2	pagingvg	16384MB	1	yes	yes	lv
hd6	hdisk0	rootvg	16384MB	1	yes	yes	lv

lsps -s

Total Paging Space	Percent Used
16384MB	1%

Can also use vmstat –I and vmstat -s

Should be balanced – NOTE VIO Server comes with 2 different sized page datasets on hdisk0

FORSYTHE

Default Page Space Calculation

- AIX Client default
  - hd6 must be >= 64MB, others must be >= 16MB
  - Page space can use no more than 20% disk
  - If real <256MB then page space = 2 x real
  - If real >= 256MB then page space = 256MB
- VIO Server
  - 1 x 512MB and 1 x 1024MB page space both on the same disk

7

FORSYTHE

PAGE SPACE BEST PRACTICE

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

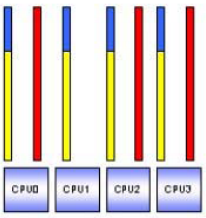
8

FORSYTHE

BASICS

FORSYTHE

IO Wait and why it is not necessarily useful



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

Metrics would show:
%user = .8 * 4 / 4 = 80%
%sys = .2 * 4 / 4 = 20%
Idle will be 0% as no core is waiting to run threads
IO Wait will be 0% as no core is idle waiting for IO to complete as something else got dispatched to that core
SO we have IO wait
BUT we don't see it

FORSYTHE

So what does iowait mean?

- Basically NOT MUCH
- High %iowait does not necessarily indicate a problem
  - Application could be IO intensive, e.g. a backup or some other streaming I/O
  - You could be running a bunch of CPU intensive jobs
    - iowait can go to 0 in this case as the CPUs are always busy even when I/O is outstanding
- Low %iowait does not necessarily mean you don't have a problem
  - The CPUs can be busy while IOs are taking unreasonably long times
- Net net is that iowait is misleading and you need to look at other things

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

```
System configuration: lcpu=32 mem=122880MB ent=8.00
kthr     memory              page              faults              cpu
-------- ----------- ----------------------- ------------ ----------------------
 r  b  p   avm        fre         fi   fo   pi  po  fr   sr  in    sy    cs   us  sy  id  wa   pc     ec
 1  0  0 16760611 13937801  36    0    0   0   0    0  405 4413 1165   5   1  94  0   0.79   9.9
```

Terms to understand – 1/2

- 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

FORSYTHE

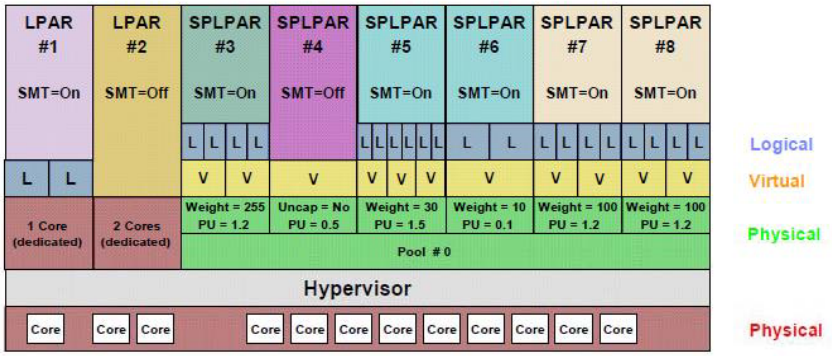
Terms to understand – 2/2

- Cache Coherency
  - All processors work with the same virtual and real address space and share the same real memory. However, each processor may have its own cache, holding a small subset of system memory. To guarantee cache coherency the processors use a snooping logic. Each time a word in the cache of a processor is changed, this processor sends a broadcast message over the bus. The processors are “snooping” on the bus, and if they receive a broadcast message about a modified word in the cache of another processor, they need to verify if they hold this changed address in their cache. If they do, they invalidate this entry in their cache.
- Processor Affinity
  - If a thread is running on a CPU and gets interrupted and redispatched, the thread is placed back on the same CPU (if possible) because the processor’s cache may still have lines that belong to the thread. If it is dispatched to a different CPU, the thread may have to get its information from main memory. Alternatively, it can wait until the CPU where it was previously running is available, which may result in a long delay.
  - AIX automatically tries to encourage processor affinity by having one run queue per CPU. Processor affinity can also be forced by binding a thread to a processor with the bindprocessor command. CPUs in the system.

FORSYTHE

Logical Processors

Simultaneous Multi-Threading (SMT) threads are represented by logical processors



Courtesy - IBM

FORSYTHE

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

FORSYTHE

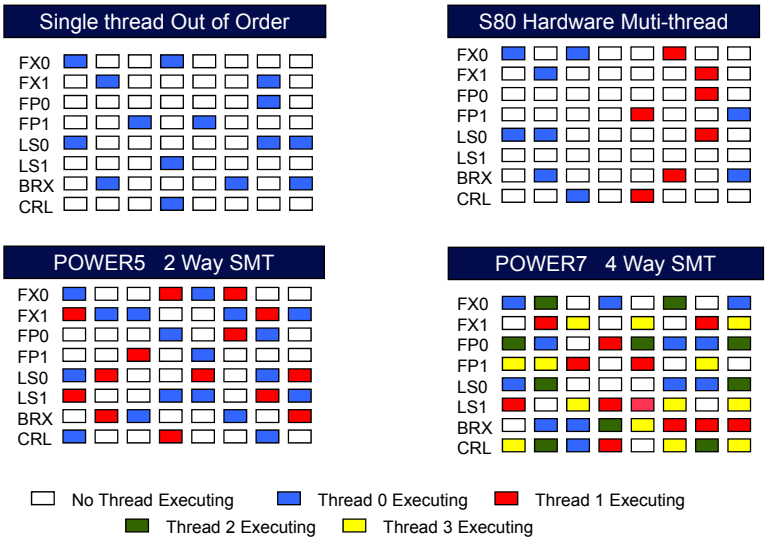
Useful processor Commands

- lsdev –Cc processor
- lsattr –EL proc0
- bindprocessor –q
- sar –P ALL
- topas, nmon
- lparstat
- vmstat (use –I or -v)
- iostat
- mpstat -s

17

FORSYTHE

Multi-threading Evolution



Courtesy - IBM

18

FORSYTHE

Using sar –P ALL

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

12:18:01	cpu	%usr	%sys	%wio	%idle	%physc	%entc
12:18:01	0	0	7	0	93	0.03	3.3
	1	100	0	0	0	0.37	46.8
	2	100	0	0	0	0.38	46.9
	3	0	1	0	99	0.02	3.1
	-	**94**	**0**	**0**	**6**	**0.80**	**100**

System is clearly busy – now map this to the mpstat command
mpstat –s 1 1
System configuration: lcpu=4 ent=0.80

Proc0		Proc1	
39.99%		39.76%	
cpu0	cpu1	cpu2	cpu3
2.55%	37.45%	37.57%	2.19%

FORSYTHE

MEMORY

FORSYTHE

Memory Segments

- Persistent
  - Have a permanent storage location on disk.
  - Files containing data or executable programs are mapped to persistent segments.
  - When a JFS file is opened and accessed, the file data is copied into RAM. VMM parameters control when physical memory frames allocated to persistent pages should be overwritten and used to store other data.
  - For JFS2, the file pages will be cached as local client pages. File data will be copied into RAM, unless the file is accessed through Direct I/O (DIO) or Concurrent I/O (CIO).
- Working
  - Are transitory and exist only during their use by a process.
  - Working segments have no permanent disk storage location. Process stack and data regions are mapped to working segments and shared library text segments.
  - Pages of working segments must also occupy disk storage locations when they cannot be kept in real memory. The disk paging space is used for this purpose.
  - When a program exits, all of its working pages are placed back on the free list immediately.
- Client
  - Are saved and restored over the network to their permanent locations on a remote file system rather than being paged out to the local system.
  - CD-ROM page-ins and compressed pages are classified as client segments.
  - JFS2 pages are also mapped into client segments.

Memory Types

- Persistent
  - Backed by filesystems
- Working storage
  - Dynamic
  - Includes executables and their work areas
  - Backed by page space
- Prefer to steal from persistent as it is cheap
- minperm, maxperm, maxclient, lru_file_repage and page_steal_method all impact these decisions

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
- maxclient=90
  - Hard cap on amount of memory that JFS2 or NFS can use – SUBSET of maxperm

FORSYTHE

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

FORSYTHE

page_steal_method Example

- 500GB memory
- Split determined by numperm numclient, and looking at vmstat –v output
- 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**

FORSYTHE

Starter set of tunables 1

For AIX v5.3
No need to set memory_affinity=0 after 5.3 tl05
MEMORY
vmo -p -o minperm%=3
vmo -p -o maxperm%=90
vmo -p -o maxclient%=90
vmo -p -o minfree=960
vmo -p -o maxfree=1088
vmo -p -o lru_file_repage=0
vmo -p -o lru_poll_interval=10
vmo -p -o page_steal_method=1

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

PBUFS
Use the new way (coming up)

JFS2
ioo -p -o j2_maxPageReadAhead=128
j2_dynamicBufferPreallocation=16
Default that may need tuning
Replaces tuning j2_nBufferPerPagerDevice

JFS (only if you are using JFS otherwise do not change)
ioo -p -o numfsbufs=1024
ioo -p -o maxpgahead=16

FORSYTHE

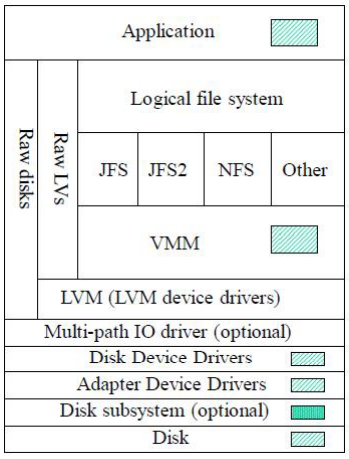
Rough Anatomy of an I/O

- LVM requests a PBUF
  - Pinned memory buffer to hold I/O request in LVM layer
- Then placed into an FSBUF
  - 3 types
  - These are also pinned
  - Filesystem — JFS
  - Client — NFS and VxFS
  - External Pager — JFS2
- If paging also need PSBUFs (also pinned)
  - Used for I/O requests to and from page space
- Then queue I/O to hdisk (queue_depth)
- Then queue it to adapter (num_cmd_elems)
- Adapter queues it to the disk subsystem

27

FORSYTHE

Anatomy of an I/O



File system buffers

Disk buffers, pbufs

queue_depth

num_cmd_elems and max_xfer_size

Diagram courtesy of IBM – Dan Braden AIX I/O Tuning presentation 2009

28

FORSYTHE

lvmo –a Output

2725270 pending disk I/Os blocked with no pbuf

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

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

	blocked	total_vg_bufs
nimvg	29	512
sasvg	2719199	1024
backupvg	6042	4608

lvmo –v sasvg –o pv_pbuf_count=2048

29

FORSYTHE

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

30

FORSYTHE

Starter set of tunables 2

For AIX v6 or v7
Make the network changes
Memory defaults are already correctly set and should not be changed
If you upgrade from a previous version of AIX using migration then you need to check the settings though

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

PBUFS
Tune these using lvmo for the individual volume group
pv_min_pbuf is now a restricted tunable

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

JFS (only if you are using JFS otherwise do not change)
ioo -p -o numfsbufs=1024 (now restricted)
ioo -p -o maxpgahead=16 (now restricted)

FORSYTHE

vmstat –I Output

vmstat -I 2 10

System Configuration: lcpu=22 mem=90112MB

```
 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
79 285  0 8537038   9371   83963 7568  44    2    84266 230503 19400 406846 77938  58  37  0  5
56 301  0 8540516   8895   91385 8912  12    3   101110 253980 17943 388340 86999  52  38  0 10
48 306  0 8544771   9565  101529 9966  14    3   112865 277552 16930 358515 82444  50  41  0  9
73 285  0 8544667   8763   94305 5915  25    3    95071 277963 19299 438769 83214  49  35  0 16
23 317  0 8547888   9846   91608 5481  12    1    97364 235613 19148 393468 74293  55  34  0 11
16 352  0 8541280   8845   92946 5246  14    0    93028 244146 18471 448516 87874  44  37  0 19
```

fre is meaningless if you do not know the minfree, maxfree and mempools values (next slide)
SR:FR should be <= 4:1
244146: 93028 is around 2.61 : 1

System configuration: lcpu=32 mem=122880MB ent=8.00

```
 kthr    memory             page              faults           cpu
-------- ----------- ------------------------ ------------ -----------------------
 r  b  p   avm        fre          fi   fo   pi  po  fr   sr  in     sy     cs   us sy id  wa   pc     ec
 1  0  0 16760611 13937801  36    0    0   0   0    0  405 4413 1165   5    1  94  0    0.79   9.9
 1  0  0 16760407 13938004   0    0    0   0   0    0  357 4445  979   5    1  93  0    0.81  10.1
```


FORSYTHE

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 page stealing or thrashing

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

minfree and maxfree

vmo –a | grep mempools
mempools = 3
You may need to look using kdb as mempools seems to have disappeared in some levels of 6.1
kdb
memp *
Quit
vmo -a | grep free
maxfree = 1088
minfree = 960

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

So if I have the following:
Memory pools = 3 (from vmo –a or kdb)
J2_maxPageReadahead = 128
CPUS = 6 and SMT on so lcpu = 12
So minfree = (max(960,(120 * 12)/3)) = 1440 / 3 = 480 or 960 whichever is larger
And maxfree = minfree + (128 * 12) / 3 = 960 + 512 = 1472

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

NETWORK

Starter set of tunables 3

Typically we set the following for both versions:

NETWORK
no -p -o rfc1323=1
no -p -o sb_max=1310720
no -p -o tcp_sendspace=262144
no -p -o tcp_recvspace=262144
no -p -o udp_sendspace=65536
no -p -o udp_recvspace=655360

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

ifconfig

ifconfig -a output

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

These override no, so they will need to be set at the adapter. Additionally you will want to ensure you set the adapter to the correct setting if it runs at less than GB, rather than allowing auto-negotiate
Stop inetd and use chdev to reset adapter (i.e. en0)

FORSYTHE

Network

Interface	Speed	MTU	tcp_sendspace	tcp_recvspace	rfc1323
lo0	N/A	16896	131072	131072	1
Ethernet	10/100 mb				
Ethernet	1000 (Gb)	1500	131072	165536	1
Ethernet	1000 (Gb)	9000	262144	131072	1
Ethernet	**1000 (Gb)**	**1500**	**262144**	**262144**	**1**
Ethernet	1000 (Gb)	9000	262144	262144	1
Virtual Ethernet	N/A	any	262144	262144	1
InfiniBand	N/A	2044	131072	131072	1

Above taken from Page 247 SC23-4905-04 November 2007 edition
Check up to date information at:
http://publib.boulder.ibm.com/infocenter/pseries/v5r3/topic/com.ibm.aix.prftungd/doc/prftungd/prftungd.pdf
AIX v6.1
http://publib.boulder.ibm.com/infocenter/aix/v6r1/topic/com.ibm.aix.prftungd/doc/prftungd/prftungd_pdf.pdf

FORSYTHE

Other Network

- netstat –v
  - Look for overflows and memory allocation failures
    Max Packets on S/W Transmit Queue: 884
    S/W Transmit Queue Overflow: 9522
  - “Software Xmit Q overflows” or “packets dropped due to memory allocation failure”
    - Increase adapter xmit queue
    - Use lsattr –EL ent? To see setting
  - Look for receive errors or transmit errors
  - dma underruns or overruns
  - mbuf errors
- tcp_nodelay
  - Disabled by default
  - 200ms delay by default as it waits to piggy back acks on packets
- Also check errpt – people often forget this

FORSYTHE

Volume groups and file systems

FORSYTHE

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
•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

FORSYTHE

Finding queue_depth and adapter queue problems

- iostat –D
- SDDPCM
  - pcmpath query devstats
  - pcmpath query adaptstats
- SDD
  - datapath query devstats
  - datapath query adaptstats
- sar –D
- Interactive nmon
  - -D option
- lsattr –El hdisk?
- lsattr –EL fcs?
- For queue_depth look at avsqsz, avgwqsz and sqfull
- fcstat command

FORSYTHE

iostat -D

Extended Drive Report **Also check out the –aD option**

```
hdisk3       xfer:  %tm_act     bps      tps      bread     bwrtn
                      0.5     29.7K     6.8      15.0K     14.8K
             read:     rps  avgserv  minserv  maxserv   timeouts     fails
                      29.3     0.1      0.1      784.5        0           0
            write:     wps  avgserv  minserv  maxserv   timeouts     fails
                     133.6     0.0      0.3      2.1S         0           0
             wait:       avgtime  mintime  maxtime   avgqsz     sqfull
                         0.0      0.0      0.2       0.0        0
```

tps	Transactions per second – transfers per second to the adapter
avgserv	Average service time
Avgtime	Average time in the wait queue
avgwqsz	Average wait queue size
	If regularly >0 increase queue-depth
avgsqsz	Average service queue size (waiting to be sent to disk)
	Can't be larger than queue-depth for the disk
sqfull	Number times the service queue was full

Look at iostat –aD for adapter queues

If avgwqsz > 0 or sqfull high then increase queue_depth. Also look at avgsqsz.

Per IBM Average IO sizes:

read = bread/rps

write = bwrtn/wps

43

FORSYTHE

sar -d

- sar –d 2 6 shows:
- avque

  Average IOs in the wait queue

  Waiting to get sent to the disk (the disk's queue is full)

  Values > 0 indicate increasing queue_depth may help performance

  Used to mean number of IOs in the disk queue
- avwait

  Time waiting in the wait queue (ms)
- avserv

  I/O service time when sent to disk (ms)

44

FORSYTHE

Adapter Tuning 1/2

- From iostat –aD

fcs0 xfer: Kbps tps bkread bkwrtn partition-id
1.6 0.2 0.0 0.2 0
read: rps avgserv minserv maxserv
0.0 20.9S 0.1 31.1
write: wps avgserv minserv maxserv
1622.2 0.0 0.2 335.1
queue: avgtime mintime maxtime avgwqsz avgsqsz sqfull
0.0 0.0 0.2 0.0 0.0 0.0

45

FORSYTHE

Adapter Tuning – 2/2

fcs0

nit_link al INIT Link flags True

lg_term_dma 0x800000 Long term DMA True

max_xfer_size 0x100000 Maximum Transfer Size True (16MB DMA)

num_cmd_elems 200 Maximum number of COMMANDS to queue to the adapter True

Changes I often make (test first)

max_xfer_size 0x200000 Maximum Transfer Size True **(128MB DMA)**

DMA area for data I/O

num_cmd_elems 2048 Maximum number of COMMANDS to queue to the adapter True

lg_term_dma is the DMA area for control I/O

Check these are ok with your disk vendor!!!

46

FORSYTHE

Adapter Queue Problems

- Adapter device drivers use DMA for IO
- From fcstat on each fcs

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

47

FORSYTHE

Parameter Settings - Summary

	DEFAULTS			NEW	
PARAMETER	**AIXv5.3**	**AIXv6**	**AIXv7**	**SET ALL TO**	
NETWORK (no)					
Rfc1323	0	0	0	1	
tcp_sendspace	16384	16384	16384	262144 (1Gb)	
tcp_recvspace	16384	16384	16384	262144 (1Gb)	
udp_sendspace	9216	9216	9216	65536	
udp_recvspace	42080	42080	42080	655360	
MEMORY (vmo)					
minperm%	20	3	3	3	
maxperm%	80	90	90	90	JFS, NFS, VxFS, JFS2
maxclient%	80	90	90	90	JFS2, NFS
lru_file_repage	1	0	0	0	
lru_poll_interval	?	10	10	10	
Minfree	960	960	960	calculation	
Maxfree	1088	1088	1088	calculation	
page_steal_method	0	0 /1 (TL)	1	1	
JFS2 (ioo)					
j2_maxPageReadAhead	128	128	128	as needed	
j2_dynamicBufferPreallocation	16	16	16	as needed	
JFS (ioo) – if at all possible do not use JFS					
Numfsbufs	196	196	196	use JFS2 – if JFS adjust as needed	
Maxpgahead	8	8	8	use JFS2 – if JFS adjust as needed	

48

FORSYTHE

Performance Tools

FORSYTHE

Tools

- topas
  - New –L flag for LPAR view
- nmon
- nmon analyzer
  - Windows tool so need to copy the .nmon file over in ascii mode
  - Opens as an excel spreadsheet and then analyses the data
  - Also look at nmon consolidator
- sar
  - sar -A -o filename 2 30 >/dev/null
  - Creates a snapshot to a file – in this case 30 snaps 2 seconds apart
  - Must be post processed on same level of system
- lparstat, mpstat
- ioo, vmo, schedo
- vmstat –v
- lvmo
- iostat
- Check out Alphaworks for the Graphical LPAR tool
- Ganglia - http://ganglia.info
- Nmonrrd and nmon2web and pGraph
- Commercial IBM
  - PM for AIX
  - Performance Toolbox
  - Tivoli ITM
  - Lots of other commercial products

FORSYTHE

Other tools

- filemon
  - filemon -v -o filename -O all
  - sleep 30
  - trcstop
- pstat to check async I/O
  - pstat -a | grep aio | wc –l
- perfpmr to build performance info for IBM if reporting a PMR
  - /usr/bin/perfpmr.sh 300

FORSYTHE

nmon

- nmon -ft -A -s 15 -c 120
  - Grabs a 30 minute nmon snapshot with async I/O
- nmon -ft -A -M -L -^ -s 150 -c 576
  - Same as above but includes large pages and runs for 24 hours
- Must be running nmon12e or higher
- Nmon comes with AIX at 5.3 tl09 or 6.1 tl01 and higher
- Creates a file in the working directory that ends .nmon
- This file can be transferred to your PC and interpreted using nmon analyser or other tools
- nmon –f –O – now gets seastats for VIO server
- nmon –f –K - dump libperfstat structures
- http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmon
- http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmonanalyser
- http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmonconsolidator

FORSYTHE

nmon

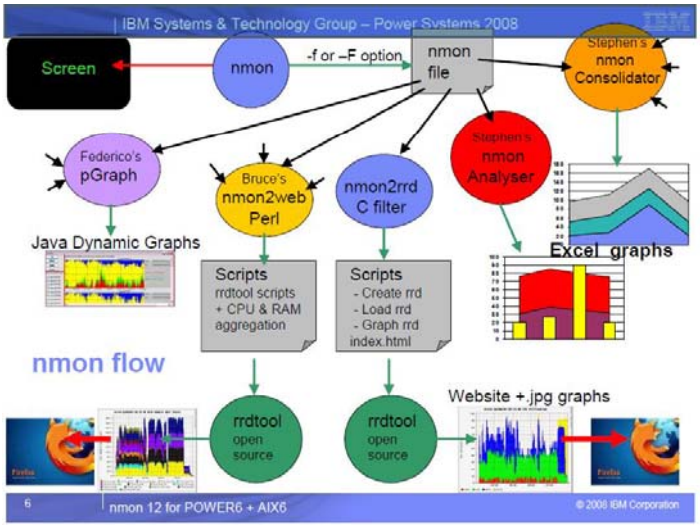


Diagram courtesy Nigel Griffiths nmon presentation 2008

FORSYTHE

nmon on POWER6 & AIX6 + New Features for V12

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

FORSYTHE

Questions???