

AIX Performance Tuning for Databases

Jaqui Lynch

Mainline Information Systems

Jaqui.lynch@mainline.com

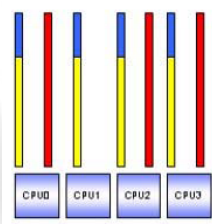
Mainline: solutions you need
from people you trust

Agenda

- Starter Pointers
- Starter set of tunables
- Determining what to set tunables to
- Page space
- Memory tuning
- Oracle and disk
- Volume groups and filesystems
- Asynchronous and Concurrent I/O

Mainline: solutions you need
from people you trust 2

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

Mainline: solutions you need
from people you trust 3

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

Mainline: solutions you need
from people you trust 4

Terms to understand

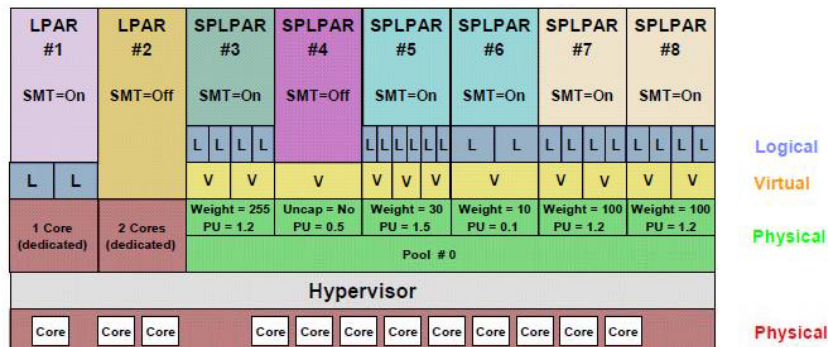
- Process
 - A process is an activity within the system that is started with a command, a shell script, or another process.
- Run Queue
 - Each CPU has a dedicated run queue. A run queue is a list of runnable threads, sorted by thread priority value. There are 256 thread priorities (zero to 255). There 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
- 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 redispached, the thread is placed back on the same CPU (if possible) because the processor's cache may still have lines that belong to the thread. If it is dispatched to a different CPU, the thread may have to get its information from main memory. Alternatively, it can wait until the CPU where it was previously running is available, which may result in a long delay.
 - 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.

Mainline: solutions you need from people you trust 5

Logical Processors

Logical Processors

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



Courtesy - IBM

Mainline: solutions you need from people you trust 6

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

Mainline: solutions you need
from people you trust

7

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

Mainline: solutions you need
from people you trust

8

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 2.55%	cpu1 37.45%
	cpu2 37.57%
	cpu3 2.19%

Mainline: solutions you need
from people you trust 9

NETWORK

Mainline: solutions you need
from people you trust 10

Starter set of tunables 1/3

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

Mainline: solutions you need
from people you trust 11

ifconfig

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 0xffffe00 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)

Mainline: solutions you need
from people you trust 12

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.prfungd/doc/prfungd/prfungd.pdf>

Mainline: solutions you need
from people you trust 13

MEMORY

Mainline: solutions you need
from people you trust 14

Starter set of tunables 2/3

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
vmo -p -o maxfree=1088
vmo -p -o lru_file_repage=0
vmo -p -o lru_poll_interval=10
```

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

PBUFS

ioo -p -o pv_min_pbuf=1024 – **old way** – 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

```
ioo -p -o numfsbufs=1024
ioo -p -o maxpgahead=16
```

Mainline: solutions you need
from people you trust 15

j2_dynamicBufferPreallocation

The number of 16k chunks to preallocate when the filesystem is running low of bufstructs.

Old method – tune j2_nBufferPerPagerDevice

Minimum number of file system bufstructs for Enhanced JFS.

New method

Leave j2_nBufferPerPagerDevice at the default
Increase j2_dynamicBufferPreallocation as needs be.

16k slabs, per filesystem and requires a filesystem remount.

vmstat -v

Increase if "external pager filesystem I/Os blocked with no fsbuf" increases
I/O load on the filesystem may be exceeding the speed of preallocation.

Mainline: solutions you need
from people you trust 16

pv_min_pbuf

pv_min_pbuf

Purpose:

Specifies the minimum number of pbufs per PV that the LVM uses. This is a global value that applies to all VGs on the system.

Values:

Default: 256 on 32-bit kernel; 512 on 64-bit kernel.

Range: 512 to 2G-1

Type: Dynamic

vmstat -v

"pending disk I/Os blocked with no pbuf"

Indicates that the LVM had to block I/O requests waiting for pbufs to become available.

We now tune this at the **individual volume group using lvmo** and no longer tune this variable across the board

In AIX v6 this becomes a restricted variable

Mainline: solutions you need
from people you trust 17

lvmo -a Output – AIX v6.1

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 xxxvvg -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

Mainline: solutions you need
from people you trust 18

lvmo -a Output

1468217 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 = 84953           this is rootvg
pv_min_pbuf = 512
global_blocked_io_count = 1468217       this is the others
```

```
vgname = datavg
pv_pbuf_count = 1024
total_vg_pbufs = 3072
max_vg_pbuf_count = 32768
pervg_blocked_io_count = 1675892
pv_min_pbuf = 1024
global_blocked_io_count = 1675892
```

`lvmo -v datavg -o pv_pbuf_count=2048`

Mainline: solutions you need
from people you trust 19

vmstat -v Output - v6.1

```
3.0 minperm percentage
90.0 maxperm percentage
21.7 numperm percentage
21.7numclient percentage
90.0 maxclient percentage
0 pending disk I/Os blocked with no pbuf           pbufs
0 paging space I/Os blocked with no psbuf          page space
2484 filesystem I/Os blocked with no fsbuf         JFS
0 client filesystem I/Os blocked with no fsbuf     NFS
3998149 external pager filesystem I/Os blocked with no fsbuf JFS2
```

System up 15 days

ioo -a shows:

```
j2_dynamicBufferPreallocation = 16
Try increasing to 32
```

Mainline: solutions you need
from people you trust 20

vmstat -v Output

```
20.0 minperm percentage
80.0 maxperm percentage
73.1 numperm percentage
0.0 numclient percentage
80.0 maxclient percentage
1468217 pending disk I/Os blocked with no pbuf          pbufs
11173706 paging space I/Os blocked with no psbuf        page space
39943187 filesystem I/Os blocked with no fsbuf          JFS
0 client filesystem I/Os blocked with no fsbuf          NFS
31386 external pager filesystem I/Os blocked with no fsbuf JFS2
```

This is clearly a system using JFS, not JFS2
And it is probably having paging problems too

Mainline: solutions you need
from people you trust 21

Starter set of tunables 3/3

For AIX v6

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

ioo -p -o numfsbufs=1024

(now restricted)

ioo -p -o maxpgahead=16

(now restricted)

Mainline: solutions you need
from people you trust 22

vmstat -l Output

```
vmstat -l 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		
27	337	0	8549988	10014	75648	8579	30	2	81664	184745	18899	264787	88177	48	35	0	17		
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		
21	326	0	8542672	8870	100228	6572	5	4	103091	272120	17680	453253	90718	43	39	0	18		
24	325	0	8548576	10259	90164	6838	10	0	98884	236616	18452	416076	79798	52	36	0	12		
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

Mainline: solutions you need
from people you trust 23

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:

$\text{minfree} = (\max(960, (120 * \text{lcpu}) / \text{memory pools}))$

$\text{maxfree} = \text{minfree} + (\text{Max}(\text{maxpgahead}, \text{j2_maxPageReadahead}) * \text{lcpu}) / \text{memory pools}$

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 $\text{minfree} = (\max(960, (120 * 12) / 3)) = 1440 / 3 = 480$ or 960 whichever is larger

And $\text{maxfree} = \text{minfree} + (128 * 12) / 3 = 960 + 512 = 1472$

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.

Mainline: solutions you need
from people you trust 24

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.

Mainline: solutions you need
from people you trust 25

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.

Mainline: solutions you need
from people you trust 26

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

Mainline: solutions you need
from people you trust

Correcting Paging

11173706 paging space I/Os blocked with no psbuf

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

```
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  Can also use vmstat -l and vmstat -s
16384MB            1%
```

Should be balanced – NOTE VIO Server comes with 2 different sized page datasets on hdisk0
Make hd6 the same size as the others in a mixed environment like this

Best practice

More than one page volume
All the same size including hd6

Mainline: solutions you need
from people you trust 28

Default Page Space Calculation

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

Mainline: solutions you need
from people you trust 29

SVMON Terminology

- *persistent*
 - Segments used to manipulate files and directories
- *working*
 - Segments used to implement the data areas of processes and shared memory segments
- *client*
 - Segments used to implement some virtual file systems like Network File System (NFS) and the CD-ROM file system
- <http://publib.boulder.ibm.com/infocenter/pseries/topic/com.ibm.aix.doc/cmds/aixcmds5/svmon.htm>

Mainline: solutions you need
from people you trust

svmon -G

	size	inuse	free	pin	virtual
memory	26279936	18778708	7501792	3830899	18669057
pg space	7995392	53026			

	work	pers	clnt	lpage
pin	3830890	0	0	0
in use	18669611	80204	28893	0

In GB Equates to: calculate GB by multiplyng by page size (4096) and then dividing by (1024*1024)

	size	inuse	free	pin	virtual
memory	100.25	71.64	28.62	14.61	71.22
pg space	30.50	0.20			

	work	pers	clnt	lpage
pin	14.61	0	0	0
in use	71.22	0.31	0.15	0

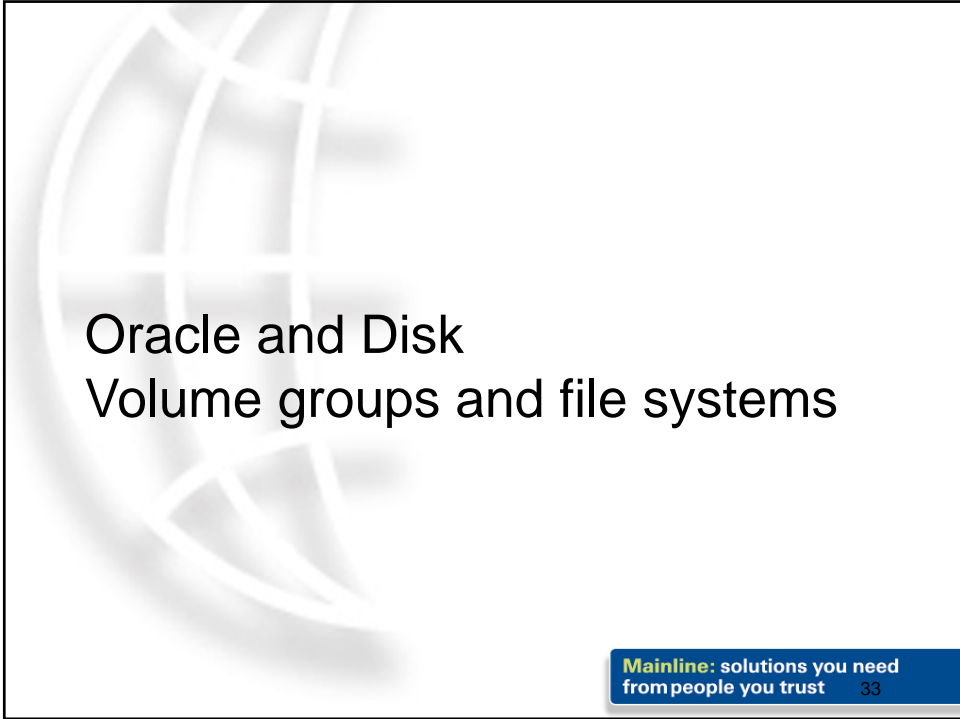
Mainline: solutions you need
from people you trust

svmon -P nnnnn -l

MEMORY ESTIMATES in MB
svmon shows memory in 4KB pages so we multiply by 4k
we then divide by 1k and again by 1k to get to mb

COMMAND		jaqui1	jaqui2	
TOTALS	INUSE	96.98	95.78	MB in RAM
	PIN	23.29	23.29	MB pinned in AM
	PGSP	0.00	0.00	MB in page space
	VIRTUAL	94.37	93.83	MB allocated to procs virtual space
SYSTEM SHARED	INUSE	89.57	89.57	179.14 work shared library text
	PIN	23.23	23.23	& work kernel segment
	PGSP	0.00	0.00	
	VIRTUAL	89.57	89.57	
PRIVATE	INUSE	5.23	4.35	private areas
	PIN	0.06	0.05	
	PGSP	0.00	0.00	
	VIRTUAL	4.80	4.26	
OTHER SHARED	INUSE	2.18	1.86	shared areas with
	PIN	0.00	0.00	other processes
	PGSP	0.00	0.00	
	VIRTUAL	0.00	0.00	
Number of Procs		12	10	
Proc * Priv Inuse		62.76	43.5	106.26

Mainline: solutions you need
from people you trust 32



Oracle and Disk

Volume groups and file systems

Mainline: solutions you need
from people you trust 33



Basics

- **Data layout will have more impact than most tunables**
- Plan in advance
- Look into whether you can use Oracle ASM
- Focus here is on JFS2
- **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 queues for hdisks
 - hdisk driver submits I/Os to the adapter driver
 - SDD and some other multi-path drivers will not submit more than queue_depth I/Os to an hdisk which can affect performance

Mainline: solutions you need
from people you trust 34

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  mintage 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
 avgqsz 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 `avgqsz`.
 Per IBM Average IO sizes:
 read = bread/rps
 write = bwrtn/wps

Mainline: solutions you need
from people you trust

Adapter Tuning 1/2

- From `iostat -aD`

```

fcs0     xfer:  Kbps  tps  bkbread  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  mintage maxtime  avgwqsz  avgqsz  sqfull
          0.0    0.0    0.2    0.0    0.0    0.0
    
```

Mainline: solutions you need
from people you trust 36

Adapter Tuning – 2/2

fcs0

bus_intr_lvl	115	Bus interrupt level	False
bus_io_addr	0xdfc00	Bus I/O address	False
bus_mem_addr	0xe8040000	Bus memory address	False
init_link	al	INIT Link flags	True
intr_priority	3	Interrupt priority	False
lg_term_dma	0x800000	Long term DMA	True
max_xfer_size	0x100000	Maximum Transfer Size	True
num_cmd_elems	200	Maximum number of COMMANDS to queue to the adapter	True
pref_alpa	0x1	Preferred AL_PA	True
sw_fc_class	2	FC Class for Fabric	True

Changes I often make (test first)

max_xfer_size	0x200000	Maximum Transfer Size	True
num_cmd_elems	2048	Maximum number of COMMANDS to queue to the adapter	True

Check these are ok with your disk vendor!!!

Mainline: solutions you need
from people you trust 37

General

- Do not put only 1 filesystem per volume group
 - You lose flexibility in solving performance problems
- If using external JFS2 logs
 - Make them 2 to 4 PPs in size so they never run out
 - Put them on a different disk that is not busy
- Per Oracle
 - Stripe LVs across disks to parallelize
 - Or set to maximum so the filesystem is spread across the disks (PP striping)
 - Offset the stripes if striping multiple LVs across the same hdisks
 - Choose a reasonable stripe size
 - Break instance out into multiple sensibly named file systems
 - Defaults of /u01, /u02 do not make it obvious
 - How about /instance1-redos and /instance1-dbf
- Mirror read intensive data
- Ensure LVs are contiguous
 - Use lslv and look at in-band % and distrib (not as useful with SAN)
 - reorgvg if needed to reorg LVs
- Increase maxfree if you adjust read ahead maximums

Mainline: solutions you need
from people you trust

Filesystem Layout

```
lsfs -q
/dev/lvlocal -- /usr/local jfs2 2621440 rw yes no
(lv size: 2621440, fs size: 2621440, block size: 4096, sparse files: yes, inline log: no, inline log
size: 0, EAformat: v1, Quota: no, DMAP1: no, VIX: no)
```

- Use `lsfs -q` to determine the current block size
- Break instance out into multiple sensibly named filesystems so people can tell what they are
- Redo logs and control files should be in their own filesystem or filesystems with an `agblksize` of 512 (not the default 4096)
 - I/O size is always a multiple of 512 anyway
- DBF database filesystems should be calculated as follows:
 - `db_block_size * db_file_multiblock_read_count`
 - If the block size ends up being 4096 or more than 4096 then use 4096 otherwise Oracle recommends 1024 or 2048
- Other filesystems can be left at the default of 4096
- Use CIO where useful (coming up)

Mainline: solutions you need
from people you trust

Asynchronous I/O and Concurrent I/O

Mainline: solutions you need
from people you trust 40

Async I/O - v5.3

Total number of AIOs in use

```
pstat -a | grep aios | wc -l  
Maximum AIOservers started since boot
```

AIO maxservers

```
lsattr -El aio0 -a maxservers  
maxservers 320 MAXIMUM number of servers per cpu True  
NB - maxservers is a per processor setting in AIX 5.3
```

Or new way for Posix AIOs is:

```
ps -k | grep aio | wc -l  
4205
```

Look at using fastpath

Fastpath can now be enabled with DIO/CIO
At tl05 this is controlled by aioo command

Also iostat -A

THIS ALL CHANGES IN AIX V6 - SETTINGS WILL BE UNDER IOO THERE

```
lsattr -El aio0  
autoconfig defined STATE to be configured at system restart True  
fastpath enable State of fast path True  
kprocprio 39 Server PRIORITY True  
maxreqs 4096 Maximum number of REQUESTS True  
maxservers 10 MAXIMUM number of servers per cpu True  
minservers 1 MINIMUM number of servers True  
#
```

Mainline: solutions you need
from people you trust

iostat -A

iostat -A async IO

System configuration: lcpu=16 drives=15

```
aio: avgc avfc maxg maif maxr avg-cpu: % user % sys % idle % iowait  
150 0 5652 0 12288 21.4 3.3 64.7 10.6
```

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk6	23.4	1846.1	195.2	381485298	61892856
hdisk5	15.2	1387.4	143.8	304880506	28324064
hdisk9	13.9	1695.9	163.3	373163558	34144512

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

Old calculation - no longer recommended

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

***Reboot anytime the AIO Server parameters are changed

Oracle now recommending the following

	5.3	6.1 (non CIO)
Minservers	100	default (3)
Maxservers	200	200
Maxreqs	16384	default (65536)

These are per CPU

Mainline: solutions you need
from people you trust

Async I/O – AIX v6

ioo -a -F | more

```
aio_active = 0
aio_maxreqs = 65536
aio_maxservers = 30
aio_minservers = 3
aio_server_inactivity = 300
posix_aio_active = 0
posix_aio_maxreqs = 65536
posix_aio_maxservers = 30
posix_aio_minservers = 3
posix_aio_server_inactivity = 300
```

##Restricted tunables

```
aio_fastpath = 1
aio_fsfastpath = 1
aio_kprocprio = 39
aio_multitidsusp = 1
aio_sample_rate = 5
aio_samples_per_cycle = 6
posix_aio_fastpath = 1
posix_aio_fsfastpath = 1
posix_aio_kprocprio = 39
posix_aio_sample_rate = 5
posix_aio_samples_per_cycle = 6
```

pstat -a | grep aio

```
22 a 1608e 1 1608e 0 0 1
aioPpool
24 a 1804a 1 1804a 0 0 1
aioLpool
```

Mainline: solutions you need
from people you trust

DIO and CIO

- DIO
 - Direct I/O
 - Around since AIX v5.1, also in Linux
 - Used with JFS
 - CIO is built on it
 - Effectively bypasses filesystem caching to bring data directly into application buffers
 - Does not like compressed JFS or BF (lfe) filesystems
 - Performance will suffer due to requirement for 128kb I/O
 - Reduces CPU and eliminates overhead copying data twice
 - Reads are asynchronous
 - Bypasses filesystem readahead
 - Inode locks still used
 - Benefits heavily random access workloads

Mainline: solutions you need
from people you trust

DIO and CIO

- CIO
 - Concurrent I/O – AIX only, not in Linux
 - Only available in JFS2
 - Allows performance close to raw devices
 - No system buffer caching
 - **Designed for apps (such as RDBs) that enforce write serialization at the app**
 - Allows non-use of inode locks
 - Implies DIO as well
 - Benefits heavy update workloads
 - Speeds up writes significantly
 - Saves memory and CPU for double copies
 - **Not all apps benefit from CIO and DIO – some are better with filesystem caching and some are safer that way**
- When to use it
 - Database DBF files, redo logs and control files and flashback log files.
 - Not for Oracle binaries or archive log files

Mainline: solutions you need
from people you trust

DIO/CIO Oracle Specifics

- Use CIO where it will benefit you
 - Do not use for Oracle binaries
 - Ensure redo logs are in their own filesystem with the correct (512) blocksize
 - I give each instance its own filesystem and their redo logs are also separate
- Leave `DISK_ASYNC_IO=TRUE` in Oracle
- Tweak the maxservers AIO settings
- Remember CIO uses DIO under the covers
- If using JFS
 - Do not allocate JFS with BF (LFE)
 - It increases DIO transfer size from 4k to 128k
 - 2gb is largest file size
 - Do not use compressed JFS – defeats DIO

Mainline: solutions you need
from people you trust

Telling Oracle to use CIO and AIO

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

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

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

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

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

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

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

Also read Metalink Notes #257338.1, #360287.1

Mainline: solutions you need
from people you trust

PERFORMANCE TOOLS

Mainline: solutions you need
from people you trust 48

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
- ioo, vmo, schedo, vmstat -v
- lvmo
- lparstat, mpstat
- 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

Mainline: solutions you need
from people you trust

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

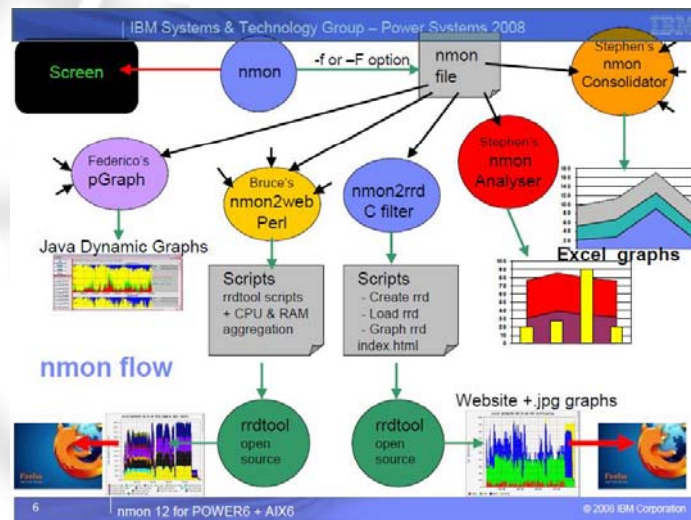
Mainline: solutions you need
from people you trust

nmon

- `nmon -ft -A -s 15 -c 120`
- Grabs a 30 minute nmon snapshot with async I/O
- Must be running `nmon12e`
- 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>

Mainline: solutions you need
from people you trust 51

nmon



Mainline: solutions you need
from people you trust 52

nmon on POWER6 & AIX6 + New Features for V12

1. Disk Service Times
2. Selecting Particular Disks
3. Time Drift
4. Multiple Page Sizes
5. Timestamps in UTC & no. of digits
6. More Kernel & Hypervisor Stats *
7. High Priority nmon

Advanced, POWER6 and AIX6 items

8. Virtual I/O Server SEA
9. Partition Mobility (POWER6)
10. WPAR & Application Mobility (AIX6)
11. Dedicated Donating (POWER6)
12. Folded CPU count (SPLPAR)
13. Multiple Shared Pools (POWER6)
14. Fibre Channel stats via entstat

Mainline: solutions you need
from people you trust 53



Questions???

Mainline: solutions you need
from people you trust 54