

AIX Performance Tuning for Databases

Jaqui Lynch

Mainline Information Systems

Jaqui.lynch@mainline.com

<http://www.circle4.com/papers/aixoracle-perf-apr09.pdf>

<http://www.mainline.com/powerpresentations>

<http://mainline.com/KnowledgeCenter>

Mainline: solutions you need
from people you trust

Agenda

- I AM NOT A DBA but I know one 😊
- 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
- Oracle AWR

Mainline: solutions you need
from people you trust 2

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 3

ifconfig

ifconfig -a output

```
en0:
flags=1e080863,480<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,
GROUPPR,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 4

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/prftungd/prftungd.pdf>

Mainline: solutions you need
from people you trust 5

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 lvm0 -a later)

PBUFS

```
ioo -p -o pv_min_pbuf=1024 – old way – use the new way (next slide)
```

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 6

numfsbufs

numfsbufs (JFS)

Purpose:
Specifies the number of file system bufstructs.

Values:
Default: 196 (value is dependent on the size of the bufstruct)

Type: Mount

Increase based on `vmstat -v` output
39943187 filesystem I/Os blocked with no fsbuf

Numbers here mean that VMM queued the I/O as if could not get a free bufstruct for it.

In AIX v6 this becomes a restricted variable

Mainline: solutions you need
from people you trust 7

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 8

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 9

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 10

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 11

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

iio -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

iio -p -o numfsbufs=1024

(now restricted)

iio -p -o maxpgahead=16

(now restricted)

Mainline: solutions you need
from people you trust 12

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 13

minfree and maxfree

```
vm0 -a | grep mempools  
mempools = 3
```

You may need to look using dbx as mempools seems to have disappeared in some levels of 6.1

```
dbx  
memp *  
quit
```

```
vm0 -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 vm0 -a or dbx)

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

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 14

Correcting Paging

11173706 paging space I/Os blocked with no psbuf

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

```
Isps -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

```
Isps -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
```

```
Isps -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** 15

Oracle and Disk Volume groups and file systems

**Mainline: solutions you need
from people you trust** 16

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 IOs to an hdisk which can affect performance

Mainline: solutions you need
from people you trust 17

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
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  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
```

Mainline: solutions you need
from people you trust 19

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 20

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
 - 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 23

Async I/O - v5.3

Total number of AIOs in use

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

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  
#
```

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
```

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 synchronous
 - 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

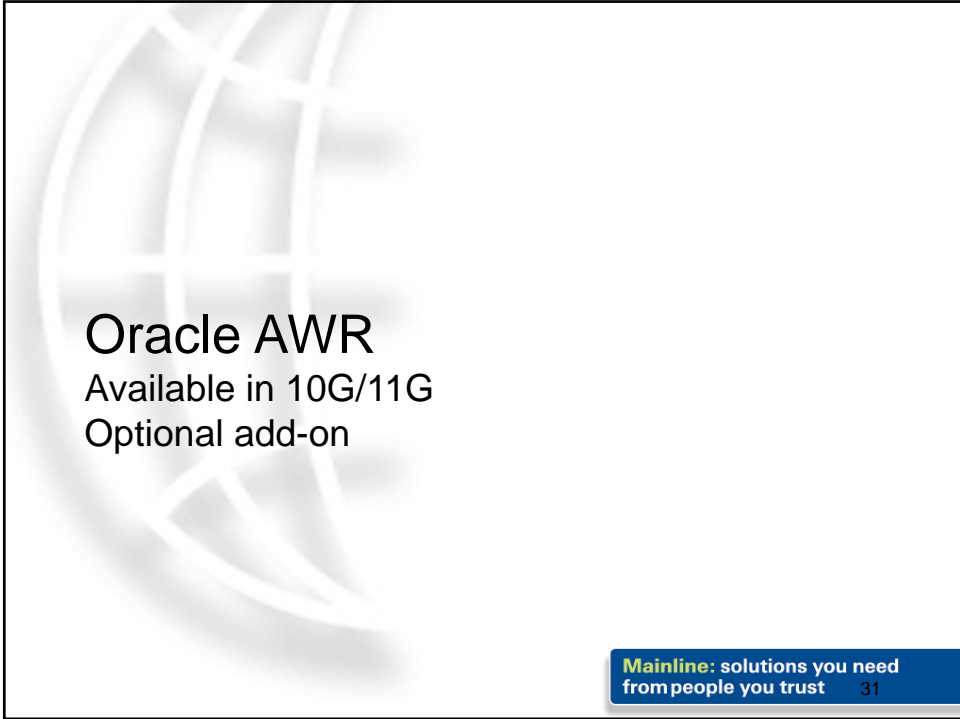
```
options           = cio      (/etc/filesystems)
disk_async_io     = true     (init.ora)
```

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



Oracle AWR

Available in 10G/11G
Optional add-on

Mainline: solutions you need
from people you trust 31



Using an AWR

- If problem is reproducible
 - Have the DBA get a snap
 - Then reproduce the problem
 - Take another snap
 - Pull the AWR which will compare those two snaps
 - Analyze the results
- AWR is an optional product but should be in any production environment

Mainline: solutions you need
from people you trust 32

Indicators of I/O Issues

- Top waits are reads and writes
- Buffer busy waits
- Write complete waits
- DB file parallel waits
- Enqueue waits
- File I/O statistics section shows high waits
- AVG Buffer wait time high

Mainline: solutions you need
from people you trust 33

Reading the AWR

- Top 5 Timed Events Report
 - Examples of issues you may see listed
 - Log file
 - Latch:redo
 - Buffer busy waits
 - Write complete waits
 - DB file parallel waits
 - Enqueue waits

Mainline: solutions you need
from people you trust 34

Top 5 Timed Events

Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
enq: TX - row lock contention	52,222	21,646	415	64.9	Application
CPU time		8,508		25.5	
log file sync	369,659	821	2	2.5	Commit
log file parallel write	380,085	650	2	1.9	System I/O
db file parallel write	172,627	239	1	.7	System I/O

Mainline: solutions you need
from people you trust 35

AWR Load Profile

	Per Second	Per Transaction
Redo size:	1,194,091.79	11,090.60
Logical reads:	119,564.90	1,110.51
Block changes:	6,262.07	58.16
Physical reads:	7.26	0.07
Physical writes:	220.09	2.04
User calls:	3,448.87	32.03
Parses:	928.27	8.62
Hard parses:	0.01	0.00
Sorts:	540.82	5.02
Logons:	0.01	0.00
Executes:	2,438.63	22.65
Transactions:	107.67	

% Blocks changed per Read:	5.24	Recursive Call %:	6.52
Rollback per transaction %:	0.02	Rows per Sort:	6.28

Mainline: solutions you need
from people you trust 36

AWR Instance Efficiency

Instance Efficiency Percentages (Target 100%)

Buffer Nowait %:	99.99	Redo NoWait %:	99.99
Buffer Hit %:	99.99	In-memory Sort %:	100.00
Library Hit %:	100.00	Soft Parse %:	100.00
Execute to Parse %:	61.93	Latch Hit %:	99.75
Parse CPU to Parse Elapsed %:	1.36	% Non-Parse CPU:	99.99

Mainline: solutions you need
from people you trust 37

Wait Class

Wait Class

- s - second
- cs - centisecond - 100th of a second
- ms - millisecond - 1000th of a second
- us - microsecond - 1000000th of a second
- ordered by wait time desc, waits desc

Wait Class	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
Application	52,228	0.72	21,647	414	0.13
System I/O	558,049	0.00	892	2	1.44
Commit	369,659	0.00	821	2	0.95
Network	15,142,262	0.00	214	0	39.05
User I/O	26,730	0.00	82	3	0.07
Configuration	756	0.00	6	8	0.00
Concurrency	25,719	0.00	6	0	0.07
Other	4,565	1.25	0	0	0.01

Mainline: solutions you need
from people you trust 38

Background Wait Events

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait (ms)	Waits /txn
log file parallel write	380,085	0.00	650	2	0.98
db file parallel write	172,627	0.00	239	1	0.45
control file parallel write	2,016	0.00	3	1	0.01
control file sequential read	1,237	0.00	0	0	0.00
events in waitclass Other	4,393	0.00	0	0	0.01
os thread startup	4	0.00	0	21	0.00
direct path write	484	0.00	0	0	0.00
latch: redo writing	172	0.00	0	0	0.00
log file single write	46	0.00	0	0	0.00
log file sequential read	46	0.00	0	0	0.00
direct path read	483	0.00	0	0	0.00
latch: cache buffers chains	1	0.00	0	0	0.00
rdbms ipc message	2,743,031	0.70	60,414	22	7.07
Streams AQ: qmn slave idle wait	129	0.00	3,529	27357	0.00
Streams AQ: qmn coordinator idle wait	262	50.76	3,529	13470	0.00
pmon timer	1,206	100.00	3,514	2914	0.00
smon timer	13	84.62	3,162	243194	0.00
Streams AQ: waiting for time management or cleanup tasks	1	100.00	838	838309	0

Mainline: solutions you need
from people you trust 39

SQL Statistics

SQL Statistics

[SQL ordered by Elapsed Time](#)

[SQL ordered by CPU Time](#)

[SQL ordered by Gets](#)

[SQL ordered by Reads](#)

[SQL ordered by Executions](#)

[SQL ordered by Parse Calls](#)

[SQL ordered by Sharable](#)

[Memory](#)

[SQL ordered by Version Count](#)

[Complete List of SQL Text](#)

Mainline: solutions you need
from people you trust 40

Advisory Statistics

- [Instance Recovery Stats](#)
- [Buffer Pool Advisory](#)
- [PGA Aggr Summary](#)
- [PGA Aggr Target Stats](#)
- [PGA Aggr Target Histogram](#)
- [PGA Memory Advisory](#)
- [Shared Pool Advisory](#)
- [SGA Target Advisory](#)
- [Streams Pool Advisory](#)
- [Java Pool Advisory](#)

Mainline: solutions you need
from people you trust 41

PGA Memory Advisory

PGA Target Est (MB)	Size Factr	Estd Extra W/A MB Processed	Estd PGA W/A MB Read/ Written to Disk	Estd PGA Cache Hit %	PGA Overalloc Count
512	0.1	196,318.4	78,050.6	72.0	1,997
1,024	0.3	196,318.4	17,581.9	92.0	237
2,048	0.5	196,318.4	13,674.6	93.0	34
3,072	0.8	196,318.4	13,582.9	94.0	0
4,096	1.0	196,318.4	13,582.9	94.0	0
4,915	1.2	196,318.4	13,577.8	94.0	0
5,734	1.4	196,318.4	13,577.8	94.0	0
6,554	1.6	196,318.4	13,577.8	94.0	0
7,373	1.8	196,318.4	13,577.8	94.0	0
8,192	2.0	196,318.4	13,577.8	94.0	0
12,288	3.0	196,318.4	13,577.8	94.0	0
16,384	4.0	196,318.4	13,577.8	94.0	0
24,576	6.0	196,318.4	13,577.8	94.0	0
32,768	8.0	196,318.4	13,577.8	94.0	0

Mainline: solutions you need
from people you trust 42

SGA Target Advisory

SGA Target Size (M)	SGA Sz Factor	Est DB Time (s)	Est Physical Reads
5,000	0.3	125,949	51,450,490
10,000	0.5	121,371	48,652,655
15,000	0.8	113,268	42,764,484
20,000	1.0	112,745	42,391,439
25,000	1.3	112,332	42,081,981
30,000	1.5	112,064	41,886,981
35,000	1.8	112,020	41,857,307
40,000	2.0	111,561	41,513,936

Mainline: solutions you need
from people you trust 43

Other Reports

- Instance activity stats
- Tablespace I/O Stats
- File I/O Stats
- Buffer Pool Statistics
- Other Statistics
 - Buffer waits
 - Enqueue Activity
 - Undo Statistics
 - Latch Statistics
- Last section lists the init.ora parameters that are changed from the defaults

Mainline: solutions you need
from people you trust 44



Questions???

Mainline: solutions you need
from people you trust 45