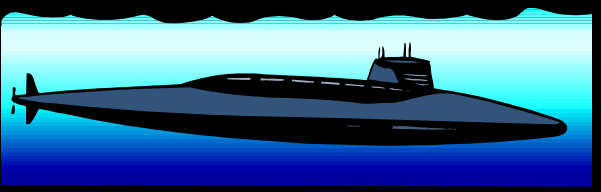
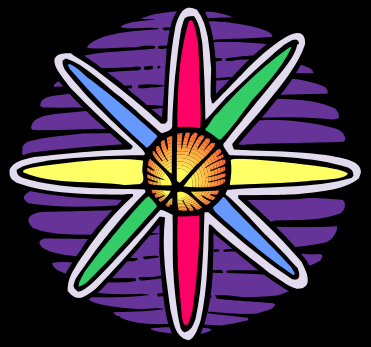
Using AWR For Wait Analysis

Mike Ault
Oracle Guru
Texas Memory Systems

A Texas Memory Systems Presentation

Michael R. Ault
Oracle Guru





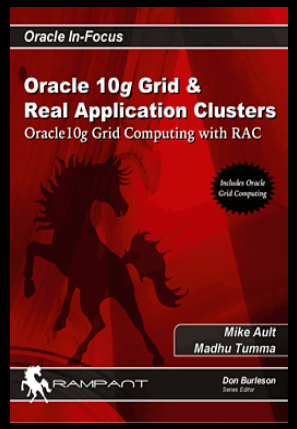
- Nuclear Navy 6 years
- Nuclear Chemist/Programmer 10 years
- Kennedy Western University Graduate
- Bachelors Degree Computer Science
- Certified in all Oracle Versions Since 6
- Oracle DBA, author, since 1990

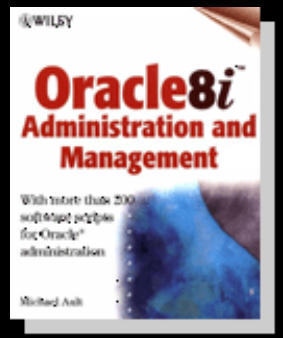
ORACLE | CERTIFIED PROFESSIONAL

A Texas Memory Systems Presentation

TMS
TEXAS MEMORY SYSTEMS

Books by Michael R. Ault





A Texas Memory Systems Presentation

TMS
TEXAS MEMORY SYSTEMS

Statspackanalyzer.com

Free Statspack/AWR Analysis

Sponsored by Texas Memory Systems

-Looks for IO bottlenecks and other configuration issues.

-Straightforward tuning advice

A Texas Memory Systems Presentation

Introduction

- Statspack was introduced in 8.1.7
- AWR came out in Oracle10g
- Both are very similar
- Both provide a top-down look at performance statistics

What Is AWR

- A background process
- A set of tables
- A set of reports
- Takes snapshots of statistics every hour
- Takes snapshot of high-cost SQL every hour

Preparation for Analysis

- Know your systems normal performance fingerprint
- Be familiar with Concepts and Tuning Guides
- Have “normal” AWR/Statspacks for comparison

Top-Down Approach

- Report starts with settings overview
- Next provides Top-5 waits
- Use the Waits to guide further investigation

AWR Report Header

```
WORKLOAD REPOSITORY report for
DB Name         DB Id    Instance     Inst Num Startup Time    Release     RAC
------------ ---------- ------------ -------- --------------- ----------- ---
AULTDB        4030696936 aultdb1             1 04-Aug-08 10:16 11.1.0.6.0  YES
Host Name        Platform                         CPUs Cores Sockets Memory(GB)
---------------- -------------------------------- ---- ----- ------- ----------
aultlinux3       Linux IA (32-bit)                   2     1       1       2.97
              Snap Id      Snap Time      Sessions Curs/Sess
            --------- ------------------- -------- ---------
Begin Snap:        91 04-Aug-08 12:00:15       41       1.2
  End Snap:        92 04-Aug-08 13:00:28       47       1.1
   Elapsed:               60.22 (mins)
   DB Time:              139.52 (mins)
Cache Sizes                       Begin        End
~~~~~~~~~~~                  ---------- ----------
               Buffer Cache:     1,312M     1,312M  Std Block Size:          8K
           Shared Pool Size:       224M       224M      Log Buffer:     10,604K
```

Know Your Load Type!

- Online Transaction Processing
  - Few reads
  - Many writes
  - Many small transactions
  - Look for redo/undo and sequential read issues
- Decision Support/Data Warehouse
  - Many reads
  - Few writes (other then possible temp)
  - Few transactions
  - Look for sort/workarea and scattered read issues
- Mixed or Hybrid

Load Profile Section

Load Profile	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	2.3	7.1	0.63	1.05
DB CPU(s):	0.3	0.9	0.07	0.13
Redo size:	800.5	2,461.8		
Logical reads:	6,307.6	19,396.7		
Block changes:	3.6	10.9		
Physical reads:	2,704.9	8,317.8		
Physical writes:	86.9	267.3		
User calls:	2.2	6.8		
Parses:	2.0	6.1		
Hard parses:	0.0	0.1		
W/A MB processed:	932,965.4	2,868,990.9		
Logons:	0.1	0.2		
Executes:	3.7	11.3		
Rollbacks:	0.1	0.3		
Transactions:	0.3			

What Are Your Efficiencies

- Should be close to 100%
- Parse issues usually are a result of:
  - Bad bind variable usage
  - Insufficient memory
  - Will also be co-indicated by low percentage of memory for multiple SQL execution

Load Profile Section

```
Instance Efficiency Percentages (Target 100%)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

            Buffer Nowait %:  100.00       Redo NoWait %:   99.97
            Buffer  Hit   %:   96.09    In-memory Sort %:  100.00
            Library Hit   %:   98.17        Soft Parse %:   97.88
         Execute to Parse %:   45.80         Latch Hit %:   99.95
Parse CPU to Parse Elapsd %:    0.00     % Non-Parse CPU:   99.77
 Shared Pool Statistics        Begin    End
                              ------  ------
             Memory Usage %:   81.53   85.39
    % SQL with executions>1:   79.29   79.48
  % Memory for SQL w/exec>1:   76.73   78.19
```

Top 5 Waits Section

- Critical to look closely at this section
- Use highest wait times to guide investigation
  - DB FILE type waits – physical IO
  - BUFFER type waits – Logical IO
  - LOG type waits – Redo related
  - PX – Parallel Query
  - GC – Global Cache (RAC related)
  - Undo – Undo or rollback segment related

Top 5 Waits Section

Top 5 Timed Foreground Events
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
db file sequential read	465,020	3,969	9	47.4	User I/O
DB CPU		995		11.9	
db file parallel read	2,251	322	143	3.8	User I/O
db file scattered read	15,268	153	10	1.8	User I/O
gc current block 2-way	108,739	116	1	1.4	Cluster
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

DB File Type Waits

- DB File Sequential Reads – memory starvation, non-selective indexes
- DB File Scattered Reads – full table scans, insufficient indexing
- Direct Path Writes – Appends, data loads
- Direct Path Reads – Parallel slaves used to retrieve data
- DB File Parallel Writes – Backup and partition use
- DB File Parallel Reads – Partition use
- DB File Single Write – File header writes, excessive data files
- Direct path read temp – Temp file activity (sorts, hashes, temp tables, bitmaps)
- Direct path write temp – Temp file activity (sorts, hashes, temp tables, bitmaps)

Buffer Type Waits

- latch: cache buffers chains – Hot blocks, check for hot objects
- free buffer waits – Insufficient buffers, processes holding buffers too long, IO subsystem over loaded
- buffer busy waits – See what is causing them further along in report
- gc buffer busy – Overloaded interconnect, find problem objects and tune
- log buffer space – High load, too small a log buffer, increase log buffer size
- latch: cache buffers lru chain – Freelist issues, hot blocks
- latch: cache buffer handles – Freelist issues, hot blocks
- buffer busy - See what is causing them further along in report
- no free buffers – Insufficient buffers, dbwr contention

Log Type Waits

- log file parallel write – Look for log file contention
- log buffer space – Look at increasing log buffer size
- log file switch (checkpoint incomplete) – May indicate excessive db files or slow IO subsystem
- log file switch (archiving needed) – Indicates archive files are written too slowly
- log file switch completion – May need more log files per thread
- log file sync – Could indicate excessive commits

PX Type Waits

- PX Deq: Msg Fragment – PEM maybe too small
- PX qref latch - Data is produced faster than it is consumed, look at PEM
- PX Deq Credit: send blkd – Look at PEM size and parallel query into non-parallel DML

GC Events

- gc cr multi block request – Full table or index scans
- gc current multi block request – Full table or index scans
- gc cr block 2-way – Blocks are busy in another instance, check for block level contention or hot blocks
- gc cr block 3-way – Blocks are busy in another instance, check for block level contention or hot blocks
- gc cr block busy – Blocks are busy in another instance, check for block level contention or hot blocks
- gc cr block congested – cr block congestion, check for hot blocks or busy interconnect
- gc cr block lost – Indicates interconnect issues and contention
- gc current block 2-way – Blocks are busy in another instance, check for block level contention or hot blocks
- gc current block 3-way – Blocks are busy in another instance, check for block level contention or hot blocks
- gc current block busy – Block is already involved in GC operation, shows hot blocks or congestion
- gc current block congested – current block congestion, check for hot blocks or busy interconnect
- gc current block lost - Indicates interconnect issues and contention

Undo Events

- undo segment extension – If excessive, tune undo
- latch: In memory undo latch – If excessive could be bug, check for your version, may have to turn off in memory undo
- wait for a undo record – Usually only during recovery of large transactions, look at turning off parallel undo recovery.

What Next?

- Determine wait events of concern
- Drill down to specific sections of report for deeper analysis
- Use custom scripts, ADDM and Ash to investigate issues

Questions/Comments?

Thank You!

Mike Ault

Mike.ault@texmemsys.com

http://www.statspackanalyzer.com/