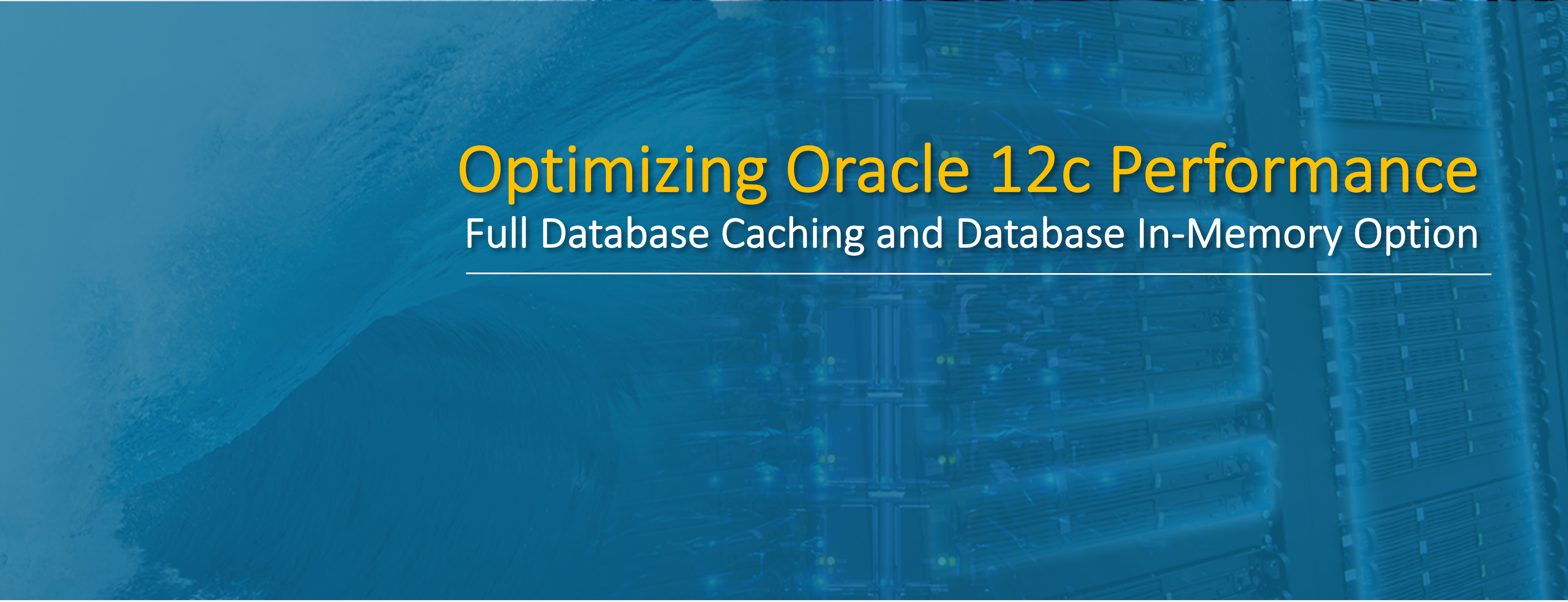


# Optimizing Oracle 12c Performance

## Full Database Caching and Database In-Memory Option

TidalScale™

Sharding Motherboards Since 2013

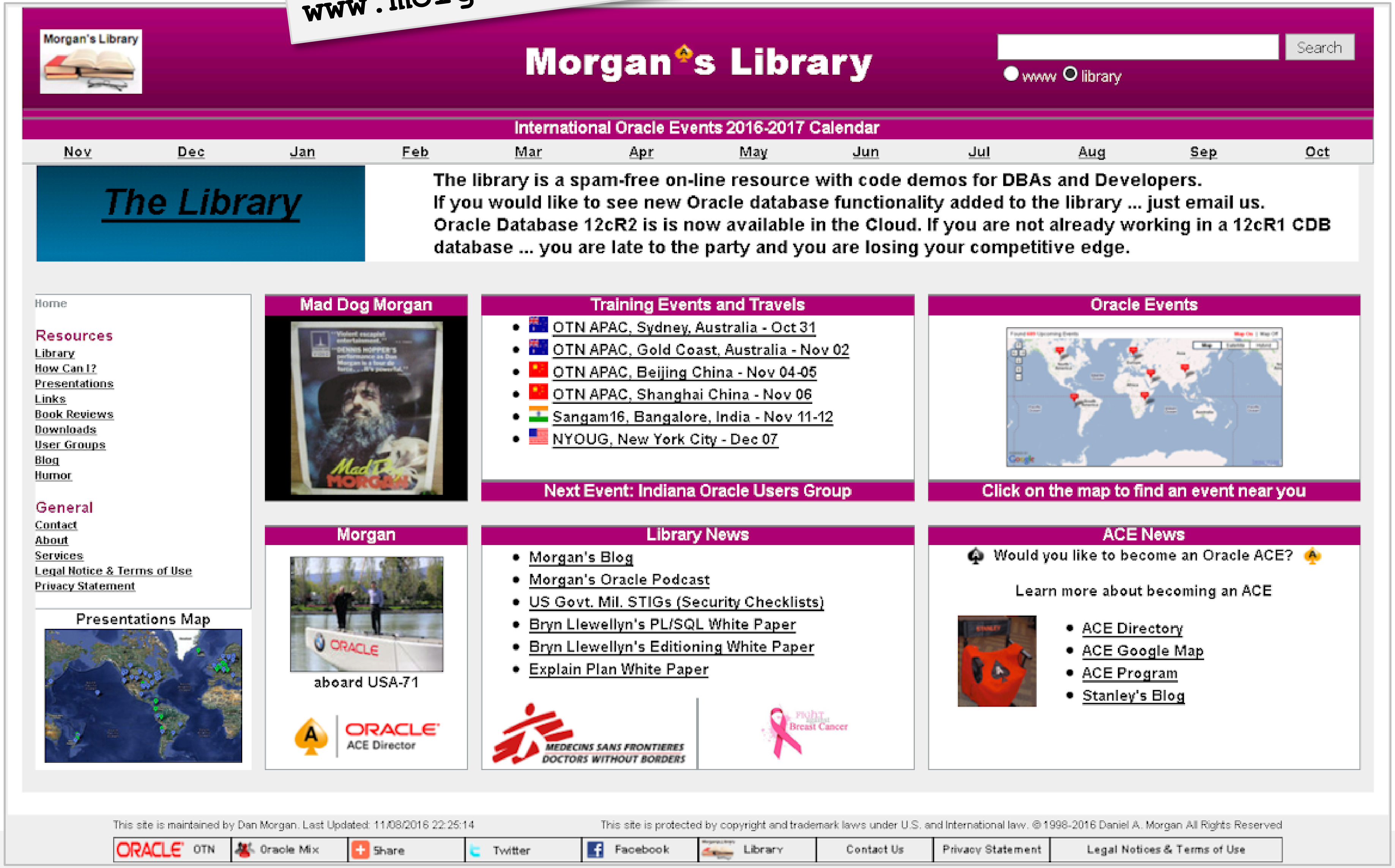
# daniel.morgan@tidalscale.com

- Director of Applications @ TidalScale
- Oracle ACE Director Alum
- Oracle Educator
  - Curriculum author and primary instructor, Oracle Program, University of Washington 1998-2009
  - Consultant: Harvard University
  - Guest lecturer at universities in Canada, Chile, Costa Rica, New Zealand, Norway, Panama
  - Frequent lecturer at Oracle conferences … 43 countries since 2008
- IT Professional
  - 2019 will be my 50th year in IT
  - First computer: IBM 360/40 in 1969: Fortran IV
  - Oracle Database since 1988-9 and Oracle Beta tester
  - The Morgan behind www.morganslibrary.org
  - Member Oracle Data Integration Solutions Partner Advisory Council
  - Founding member International TidalScale User Community (ITUC)

TidalScale™

# My Personal Website

www.morganslibrary.org



Morgan's Library

# Morgan's Library

Search

○ www ○ library

International Oracle Events 2016-2017 Calendar

Nov | Dec | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct

**The Library**

The library is a spam-free on-line resource with code demos for DBAs and Developers.
If you would like to see new Oracle database functionality added to the library ... just email us.
Oracle Database 12cR2 is is now available in the Cloud. If you are not already working in a 12cR1 CDB database ... you are late to the party and you are losing your competitive edge.

Home

**Resources**

- Library
- How Can I?
- Presentations
- Links
- Book Reviews
- Downloads
- User Groups
- Blog
- Humor

**General**

- Contact
- About
- Services
- Legal Notice & Terms of Use
- Privacy Statement

Presentations Map

**Mad Dog Morgan**

**Training Events and Travels**

- OTN APAC, Sydney, Australia - Oct 31
- OTN APAC, Gold Coast, Australia - Nov 02
- OTN APAC, Beijing China - Nov 04-05
- OTN APAC, Shanghai China - Nov 06
- Sangam16, Bangalore, India - Nov 11-12
- NYOUG, New York City - Dec 07

Next Event: Indiana Oracle Users Group

**Oracle Events**

Click on the map to find an event near you

**Morgan**

aboard USA-71

A ORACLE ACE Director

**Library News**

- Morgan's Blog
- Morgan's Podcast
- US Govt. Mil. STIGs (Security Checklists)
- Bryn Llewellyn's PL/SQL White Paper
- Bryn Llewellyn's Editioning White Paper
- Explain Plan White Paper

MEDECINS SANS FRONTIERES DOCTORS WITHOUT BORDERS

Fight Against Breast Cancer

**ACE News**

Would you like to become an Oracle ACE?

Learn more about becoming an ACE

- ACE Directory
- ACE Google Map
- ACE Program
- Stanley's Blog

This site is maintained by Dan Morgan. Last Updated: 11/08/2016 22:25:14

This site is protected by copyright and trademark laws under U.S. and International law. © 1998-2016 Daniel A. Morgan All Rights Reserved

ORACLE OTN | Oracle Mix | Share | Twitter | Facebook | Library | Contact Us | Privacy Statement | Legal Notices & Terms of Use

TidalScale™

## Performance Problems Have Serious Consequences . . .

- Internal and External customers have expectations
- There is a long history of disappoint
- Thus, we have Service Level Agreements

## When We Fail To Deliver . . .

- Internal customers develop their own solutions
- External customers go elsewhere
- SLA violations result in financial penalties
- Management wonders whether we are providing value

# Only 2 Things Matter In Business Computing . . .

## QoS

- Stability
- Security
- Scalability
- Usability
- Performance

## TCO

- Affordability

TidalScale™

# The History of Oracle Performance Tuning . . .

# How Many Books Read?

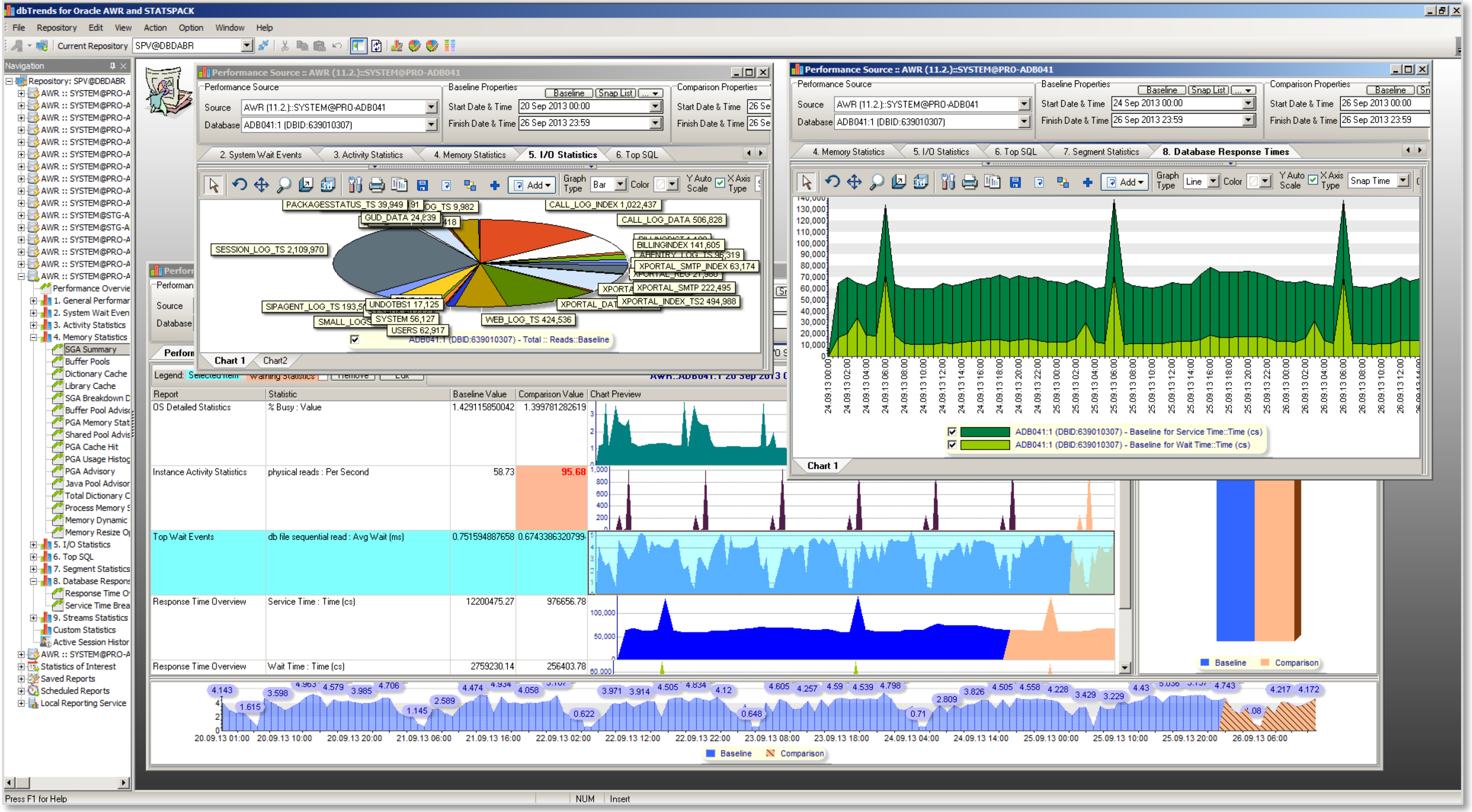
# How Many Oracle Tools Deployed?

- DBMS_SUPPORT (version 7.2)
- DBMS_TRACE (version 8.1.5)
- DBMS_MONITOR (version 10gR1)
- Oracle Enterprise Manager (OEM)
- StatsPack, ADDM, ASH, AWR, TKPROF, ....

Troubleshooting Oracle Performance
Christian Antognini
Apress
Oracle Database 11g Release 2 Performance Tuning Tips & Techniques
Oracle Performance Firefighting
OraPub
Optimizing Oracle Performance
O'REILLY
Oracle Database 12c Release 2 Performance Tuning Tips and Techniques
Oracle Performance Tuning
O'REILLY
Effective Oracle by Design
Thomas Kyte
Oracle Performance Survival Guide
A Systematic Approach to Database Optimization
Cost-Based Oracle Fundamentals
Jonathan Lewis
Apress
IOUG
Troubleshooting Oracle Performance
Apress
IOUG
Oracle Database 12c Performance Tuning Recipes
A Problem-Solution Approach
Apress
Christopher Lawson
The Art and Science of Oracle Performance Tuning
Oracle RAC Performance Tuning
Brian Peasland
101
Oracle Performance Tuning
Oracle Wait Interface: A Practical Guide to Performance Diagnostics & Tuning
Oracle Insights
Tales of the Oak Table
Apress
WILEY
Relational Database Index Design and the Optimizers
Oracle Performance Tuning Expert
100 Powerful Tips
Christopher Lawson

```
ALTER SESSION SET tracefile_identifier = 'test_plan1';

ALTER SESSION SET EVENTS '10053 trace name context forever, level 1';

ALTER SESSION SET EVENTS '10046 trace name context forever, level 12';

-- execute SQL

ALTER SESSION SET EVENTS '10053 trace name context OFF';
ALTER SESSION SET EVENTS '10046 trace name context OFF';
or
ALTER SESSION SET SQL_TRACE=FALSE;

review the trace file in $ORACLE_BASE/diag/orabase/orabase/trace
```

# How Many Tools Have You Purchased?



Performance Source :: AWR (11.2.)::SYSTEM@PRO-ADB041

Performance Source — Source: AWR (11.2.)::SYSTEM@PRO-ADB041; Database: ADB041:1 (DBID:639010307)

Baseline Properties — Start Date & Time: 20 Sep 2013 00:00; Finish Date & Time: 26 Sep 2013 23:59

Comparison Properties — Start Date & Time: 26 Se; Finish Date & Time: 26 Se

5. I/O Statistics — Graph Type: Bar

Performance Source :: AWR (11.2.)::SYSTEM@PRO-ADB041

Performance Source — Source: AWR (11.2.)::SYSTEM@PRO-ADB041; Database: ADB041:1 (DBID:639010307)

Baseline Properties — Start Date & Time: 24 Sep 2013 00:00; Finish Date & Time: 26 Sep 2013 23:59

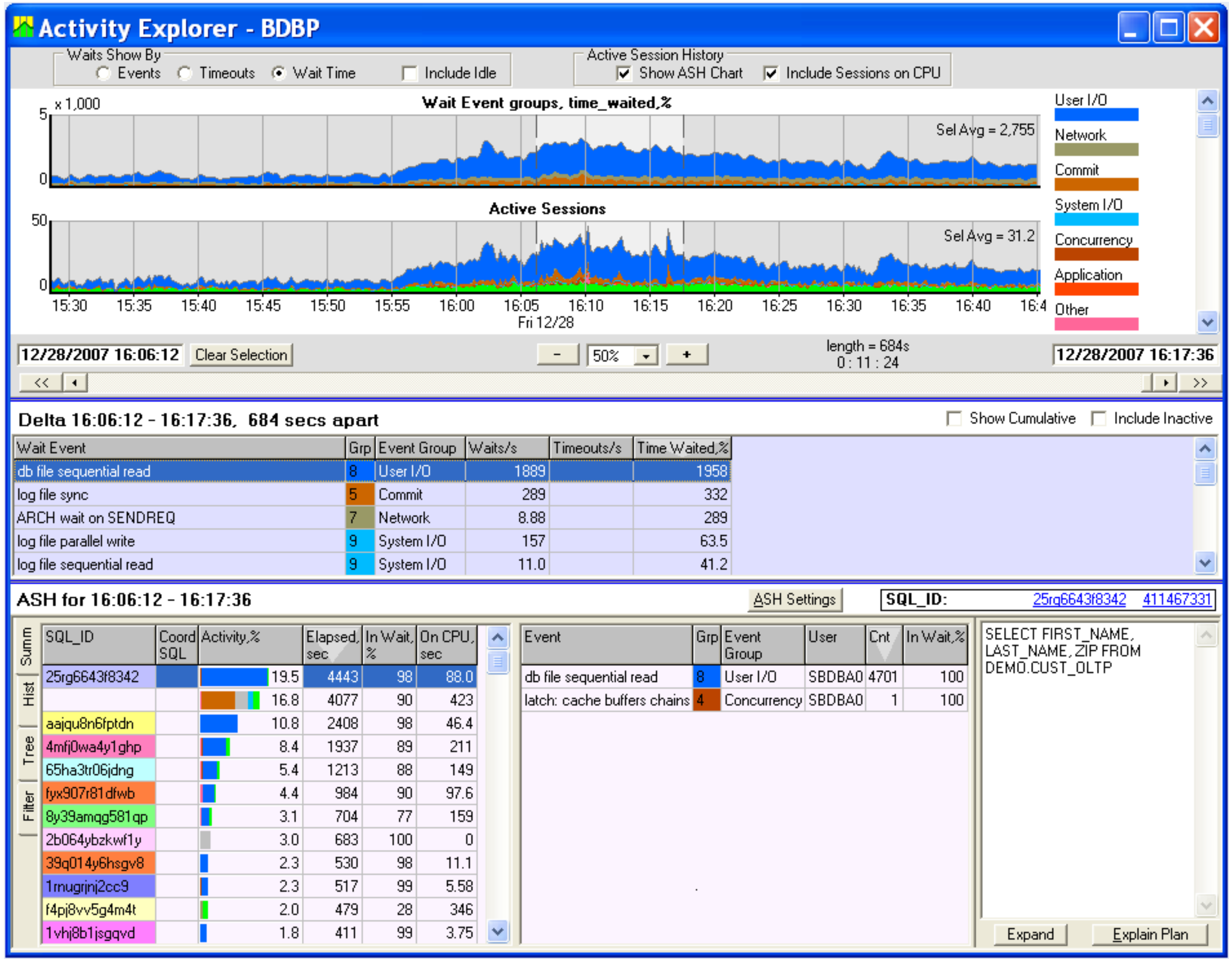
Comparison Properties — Start Date & Time: 26 Sep 2013 00:00; Finish Date & Time: 26 Sep 2013 23:59

8. Database Response Times — Graph Type: Line; Y Auto Scale ☑; X Axis Type: Snap Time

| Report | Statistic | Baseline Value | Comparison Value |
|---|---|---|---|
| OS Detailed Statistics | % Busy : Value | 1.429115850042 | 1.399781282619 |
| Instance Activity Statistics | physical reads : Per Second | 58.73 | 95.68 |
| Top Wait Events | db file sequential read : Avg Wait (ms) | 0.751594887658 | 0.6743386320799 |
| Response Time Overview | Service Time : Time (cs) | 12200475.27 | 976656.78 |
| Response Time Overview | Wait Time : Time (cs) | 2759230.14 | 256403.78 |

TidalScale™

# How Many Tools Have You Purchased?



Waits Show By: ○ Events ○ Timeouts ● Wait Time □ Include Idle

Active Session History: ☑ Show ASH Chart ☑ Include Sessions on CPU

12/28/2007 16:06:12 | 50% | length = 684s 0 : 11 : 24 | 12/28/2007 16:17:36

**Delta 16:06:12 - 16:17:36, 684 secs apart** □ Show Cumulative □ Include Inactive

| Wait Event | Grp | Event Group | Waits/s | Timeouts/s | Time Waited,% |
|---|---|---|---|---|---|
| db file sequential read | 8 | User I/O | 1889 | | 1958 |
| log file sync | 5 | Commit | 289 | | 332 |
| ARCH wait on SENDREQ | 7 | Network | 8.88 | | 289 |
| log file parallel write | 9 | System I/O | 157 | | 63.5 |
| log file sequential read | 9 | System I/O | 11.0 | | 41.2 |

**ASH for 16:06:12 - 16:17:36**

**SQL_ID:** 25rg6643f8342 411467331

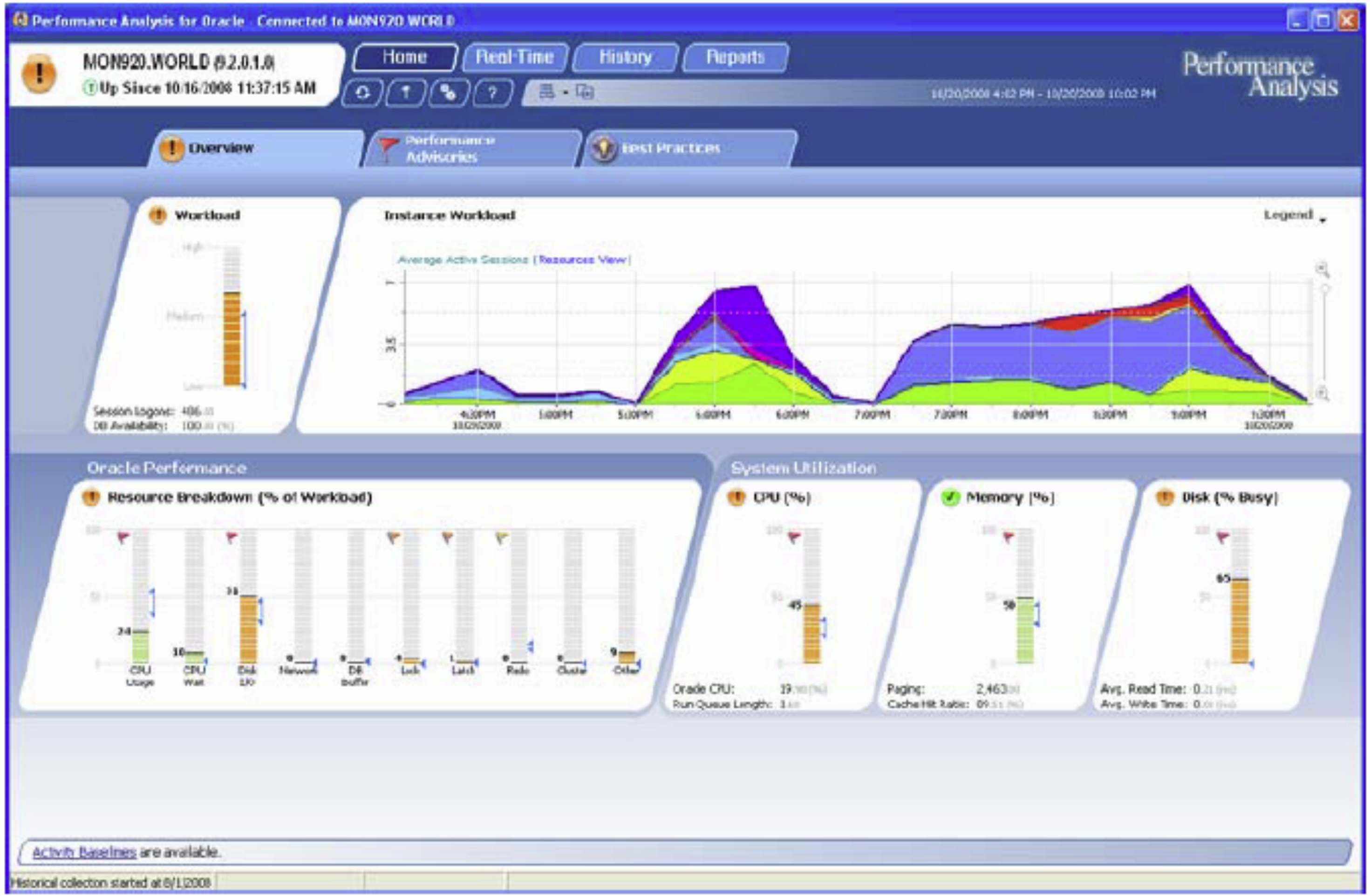
| SQL_ID | Coord SQL | Activity,% | Elapsed, sec | In Wait, % | On CPU, sec |
|---|---|---|---|---|---|
| 25rg6643f8342 | | 19.5 | 4443 | 98 | 88.0 |
| | | 16.8 | 4077 | 90 | 423 |
| aajqu8n6fptdn | | 10.8 | 2408 | 98 | 46.4 |
| 4mfj0wa4y1ghp | | 8.4 | 1937 | 89 | 211 |
| 65ha3tr06jdng | | 5.4 | 1213 | 88 | 149 |
| fyx907r81dfwb | | 4.4 | 984 | 90 | 97.6 |
| 8y39amqg581qp | | 3.1 | 704 | 77 | 159 |
| 2b064ybzkwf1y | | 3.0 | 683 | 100 | 0 |
| 39q014y6hsgv8 | | 2.3 | 530 | 98 | 11.1 |
| 1rnugrjnj2cc9 | | 2.3 | 517 | 99 | 5.58 |
| f4pj8vv5g4m4t | | 2.0 | 479 | 28 | 346 |
| 1vhj8b1jsgqvd | | 1.8 | 411 | 99 | 3.75 |

| Event | Grp | Event Group | User | Cnt | In Wait,% |
|---|---|---|---|---|---|
| db file sequential read | 8 | User I/O | SBDBA0 | 4701 | 100 |
| latch: cache buffers chains | 4 | Concurrency | SBDBA0 | 1 | 100 |

```
SELECT FIRST_NAME,
LAST_NAME, ZIP FROM
DEMO.CUST_OLTP
```

TidalScale™

# How Many Tools Have You Purchased?



MON920.WORLD (9.2.0.1.0)
Up Since 10/16/2008 11:37:15 AM

Session logons: 406
DB Availability: 100 (%)

Oracle CPU: 19 (%)
Run Queue Length: 1

Paging: 2,463
Cache Hit Ratio: 09 (%)

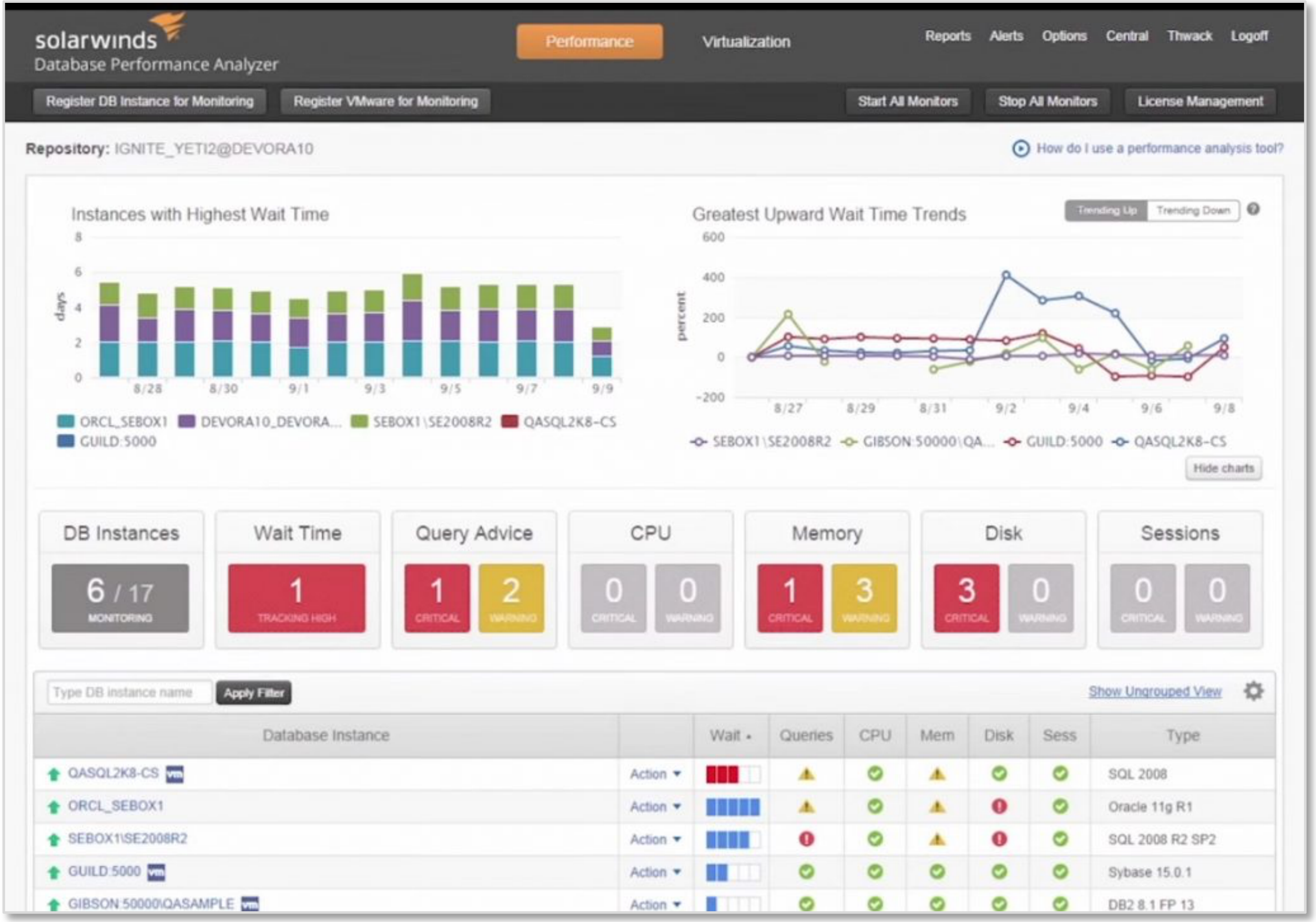
Avg. Read Time: 0.21 (ms)
Avg. Write Time: 0.0 (ms)

Activity Baselines are available.

Historical collection started at 8/1/2008

TidalScale™

# How Many Tools Have You Purchased?



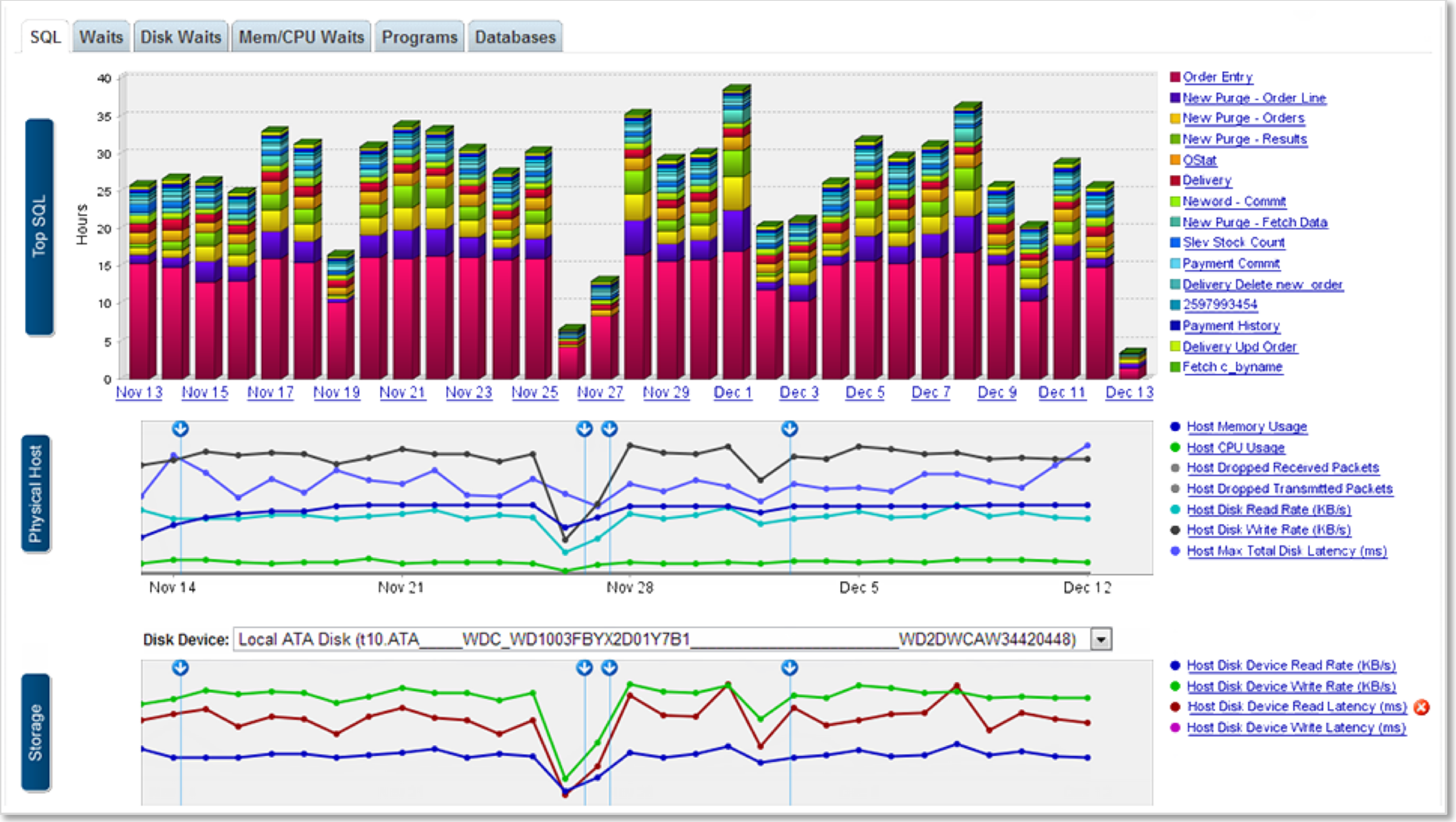
Repository: IGNITE_YETI2@DEVORA10

| DB Instances | Wait Time | Query Advice | | CPU | | Memory | | Disk | | Sessions | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 / 17<br>MONITORING | 1<br>TRACKING HIGH | 1<br>CRITICAL | 2<br>WARNING | 0<br>CRITICAL | 0<br>WARNING | 1<br>CRITICAL | 3<br>WARNING | 3<br>CRITICAL | 0<br>WARNING | 0<br>CRITICAL | 0<br>WARNING |

| Database Instance | | Wait | Queries | CPU | Mem | Disk | Sess | Type |
|---|---|---|---|---|---|---|---|---|
| QASQL2K8-CS vm | Action ▾ | ▮▮▮▯▯ | ⚠ | ✓ | ⚠ | ✓ | ✓ | SQL 2008 |
| ORCL_SEBOX1 | Action ▾ | ▮▮▮▮▮ | ⚠ | ✓ | ⚠ | ! | ✓ | Oracle 11g R1 |
| SEBOX1\SE2008R2 | Action ▾ | ▮▮▮▮▯ | ! | ✓ | ⚠ | ! | ✓ | SQL 2008 R2 SP2 |
| GUILD:5000 vm | Action ▾ | ▮▮▯▯▯ | ✓ | ✓ | ✓ | ✓ | ✓ | Sybase 15.0.1 |
| GIBSON:50000\QASAMPLE vm | Action ▾ | ▮▯▯▯▯ | ✓ | ✓ | ✓ | ✓ | ✓ | DB2 8.1 FP 13 |

TidalScale™

# How Many Tools Have You Purchased?



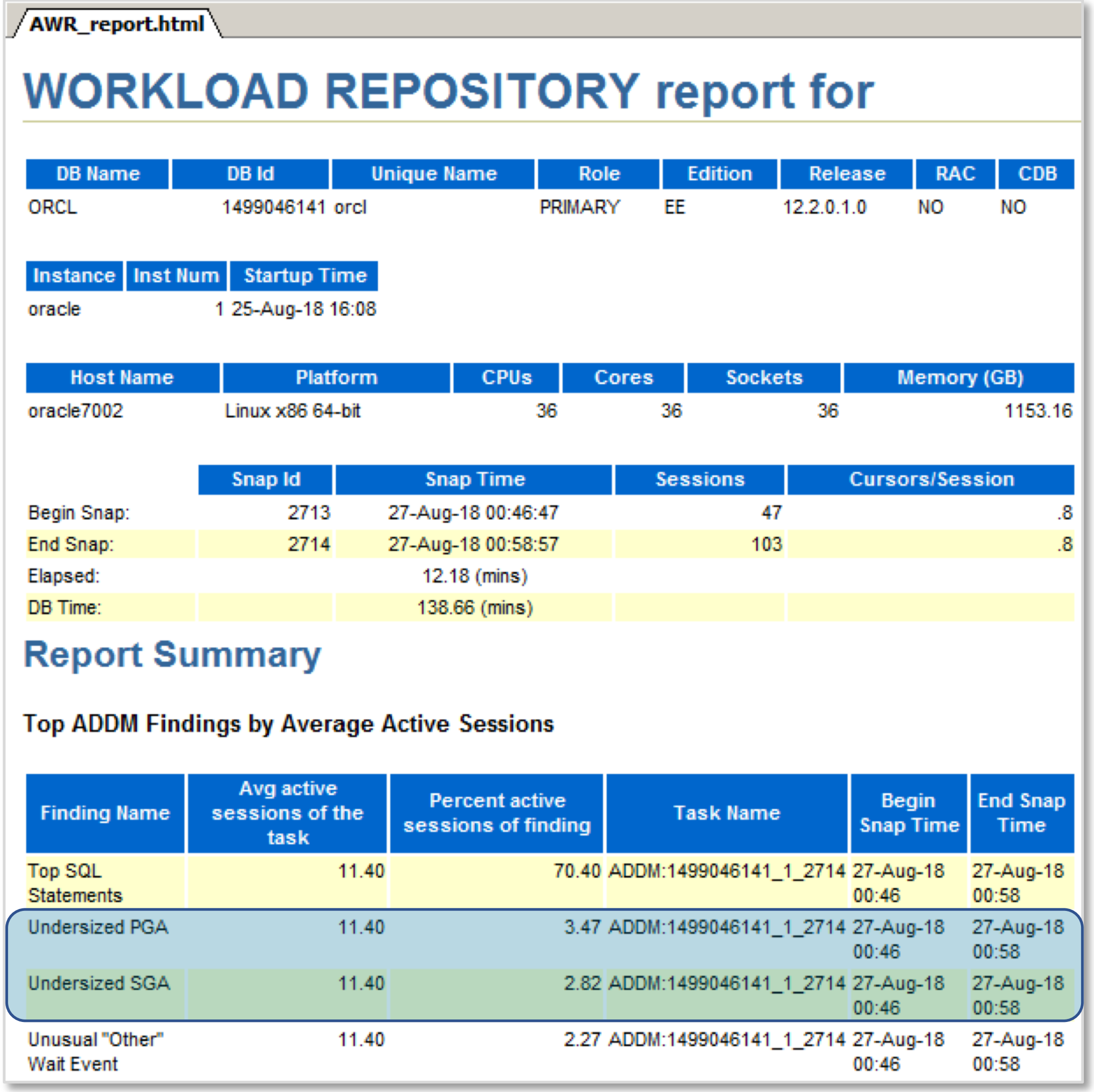
Disk Device: Local ATA Disk (t10.ATA_____WDC_WD1003FBYX2D01Y7B1_____________________WD2DWCAW34420448)

# How Many Startup Parameters Configured?

| Initialization Parameter | Description |
|---|---|
| BITMAP_MERGE_AREA_SIZE | Specifies the amount of memory used to merge bitmaps retrieved from an index range scan |
| DB_BIG_TABLE_CACHE_PERCENT_TARGET | Specifies the cache section target size for automatic big table caching, as a percentage of the buffer cache |
| DB_nK_CACHE_SIZE | Holds 8K table and index blocks |
| CREATE_BITMAP_AREA_SIZE | Memory allocated for bitmap creation a larger value may speed up index creation |
| DB_BLOCK_BUFFERS | Specifies the number of database buffers in the buffer cache |
| DB_CACHE_SIZE | Specifies the size of the DEFAULT buffer pool for buffers with the primary block size |
| DB_FLASH_CACHE_SIZE | Specifies the size of the Database Smart Flash Cache |
| DB_KEEP_CACHE_SIZE | Specifies the size of the KEEP buffer pool |
| DB_RECYCLE_CACHE_SIZE | Specifies the size of the RECYCLE buffer pool |
| HASH_AREA_SIZE | Specifies the maximum amount of memory, in bytes, to be used for hash joins |
| JAVA_MAX_SESSIONSPACE_SIZE | Memory that holds Java state from one database call to another |
| JAVA_POOL_SIZE | Pool, from which the Java memory manager allocates most Java state during runtime execution |
| LARGE_POOL_SIZE | Specifies (in bytes) the size of the large pool allocation heap |
| LOG_BUFFER | Memory used when buffering redo entries to a redo log file |
| MEMOPTIMIZE_POOL_SIZE | Specifies the size of the memoptimize pool, a memory area in the SGA used by the Memoptimized Rowstore |

| Initialization Parameter | Description |
|---|---|
| MEMORY_MAX_TARGET | Specifies the maximum value to which a DBA can set the MEMORY_TARGET initialization parameter |
| MEMORY_TARGET | Specifies the Oracle system-wide usable memory |
| OBJECT_CACHE_MAX_SIZE_PERCENT | specifies the percentage of the optimal cache size that the session object cache can grow past the optimal size |
| OBJECT_CACHE_OPTIMAL_SIZE | Specifies the size by which the session object cache is reduced when the cache size exceeds the maximum size |
| OLAP_PAGE_POOL_SIZE | Specifies the size of the OLAP page pool |
| PGA_AGGREGATE_LIMIT | Specifies a limit on the aggregate PGA memory consumed by the instance |
| PGA_AGGREGATE_TARGET | Specifies the target aggregate PGA memory available to all server processes attached to the instance |
| PRE_PAGE_SGA | Specifies whether Oracle reads the entire SGA into memory at startup so that O/S page table entries are pre-built for the SGA |
| SGA_MAX_SIZE | Specifies the maximum size of the SGA for the lifetime of the instance |
| SGA_MIN_SIZE | Specifies the minimum size of the SGA for the lifetime of the instance |
| SGA_TARGET | Specifies the total size of all SGA components |
| SHARED_POOL_RESERVED_SIZE | Specifies the shared pool space reserved for large contiguous requests for shared pool memory |
| SHARED_POOL_SIZE | Specifies the size of the shared pool which  contains shared cursors, stored procedures, control and other structures |
| SORT_AREA_RETAINED_SIZE | Specifies the maximum amount of the user global area (UGA) memory retained after a sort run completes |
| SORT_AREA_SIZE | Specifies the maximum amount of memory Oracle will use for a sort |
| STREAMS_POOL_SIZE | Specifies the memory allocated for Streams, GoldenGate Integrated Capture and other related processes |
| USE_LARGE_PAGES | Specify the management of the database's use of large pages for SGA memory |

TidalScale™



AWR_report.html

# WORKLOAD REPOSITORY report for

| DB Name | DB Id | Unique Name | Role | Edition | Release | RAC | CDB |
|---|---|---|---|---|---|---|---|
| ORCL | 1499046141 | orcl | PRIMARY | EE | 12.2.0.1.0 | NO | NO |

| Instance | Inst Num | Startup Time |
|---|---|---|
| oracle | 1 | 25-Aug-18 16:08 |

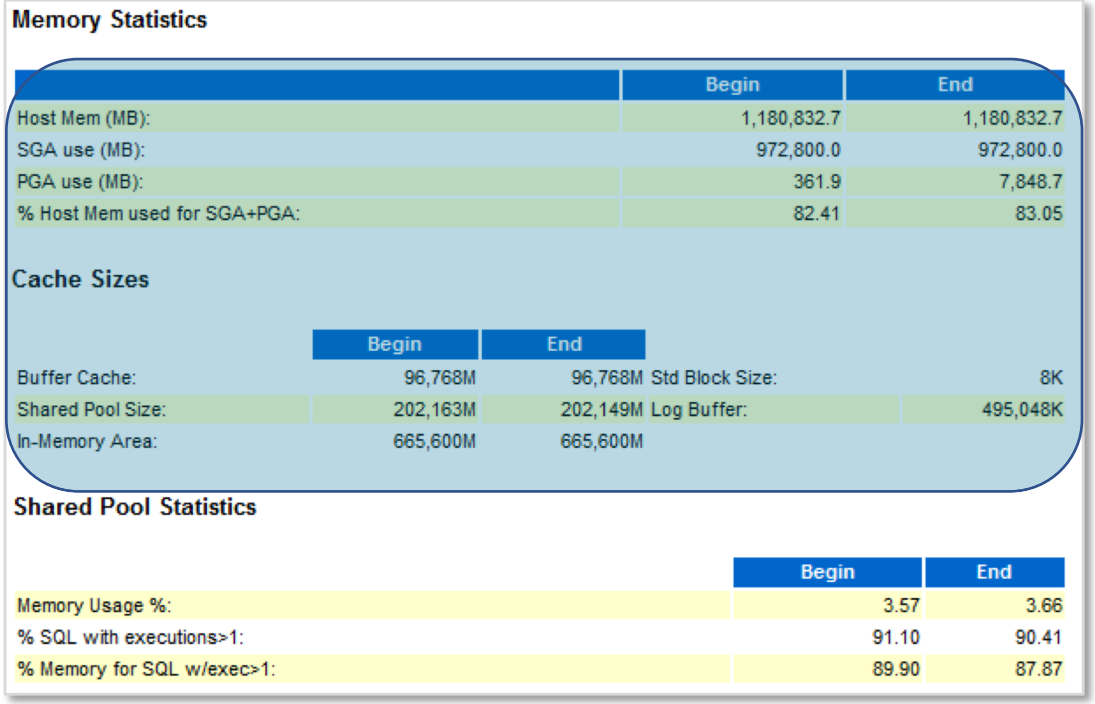
| Host Name | Platform | CPUs | Cores | Sockets | Memory (GB) |
|---|---|---|---|---|---|
| oracle7002 | Linux x86 64-bit | 36 | 36 | 36 | 1153.16 |

| | Snap Id | Snap Time | Sessions | Cursors/Session |
|---|---|---|---|---|
| Begin Snap: | 2713 | 27-Aug-18 00:46:47 | 47 | .8 |
| End Snap: | 2714 | 27-Aug-18 00:58:57 | 103 | .8 |
| Elapsed: | | 12.18 (mins) | | |
| DB Time: | | 138.66 (mins) | | |

## Report Summary

### Top ADDM Findings by Average Active Sessions

| Finding Name | Avg active sessions of the task | Percent active sessions of finding | Task Name | Begin Snap Time | End Snap Time |
|---|---|---|---|---|---|
| Top SQL Statements | 11.40 | 70.40 | ADDM:1499046141_1_2714 | 27-Aug-18 00:46 | 27-Aug-18 00:58 |
| Undersized PGA | 11.40 | 3.47 | ADDM:1499046141_1_2714 | 27-Aug-18 00:46 | 27-Aug-18 00:58 |
| Undersized SGA | 11.40 | 2.82 | ADDM:1499046141_1_2714 | 27-Aug-18 00:46 | 27-Aug-18 00:58 |
| Unusual "Other" Wait Event | 11.40 | 2.27 | ADDM:1499046141_1_2714 | 27-Aug-18 00:46 | 27-Aug-18 00:58 |



### Memory Statistics

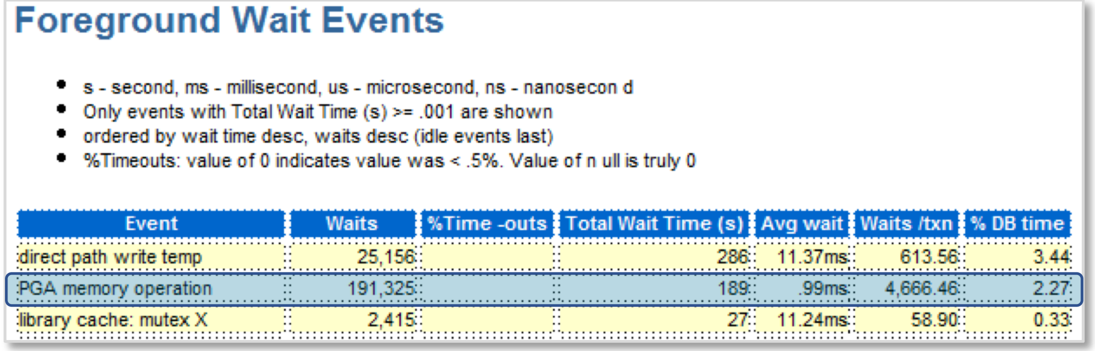
| | Begin | End |
|---|---|---|
| Host Mem (MB): | 1,180,832.7 | 1,180,832.7 |
| SGA use (MB): | 972,800.0 | 972,800.0 |
| PGA use (MB): | 361.9 | 7,848.7 |
| % Host Mem used for SGA+PGA: | 82.41 | 83.05 |

### Cache Sizes

| | Begin | End | | |
|---|---|---|---|---|
| Buffer Cache: | 96,768M | 96,768M | Std Block Size: | 8K |
| Shared Pool Size: | 202,163M | 202,149M | Log Buffer: | 495,048K |
| In-Memory Area: | 665,600M | 665,600M | | |

### Shared Pool Statistics

| | Begin | End |
|---|---|---|
| Memory Usage %: | 3.57 | 3.66 |
| % SQL with executions>1: | 91.10 | 90.41 |
| % Memory for SQL w/exec>1: | 89.90 | 87.87 |



## Foreground Wait Events

- s - second, ms - millisecond, us - microsecond, ns - nanosecon d
- Only events with Total Wait Time (s) >= .001 are shown
- ordered by wait time desc, waits desc (idle events last)
- %Timeouts: value of 0 indicates value was < .5%. Value of n ull is truly 0

| Event | Waits | %Time -outs | Total Wait Time (s) | Avg wait | Waits /txn | % DB time |
|---|---|---|---|---|---|---|
| direct path write temp | 25,156 | | 286 | 11.37ms | 613.56 | 3.44 |
| PGA memory operation | 191,325 | | 189 | .99ms | 4,666.46 | 2.27 |
| library cache: mutex X | 2,415 | | 27 | 11.24ms | 58.90 | 0.33 |

## Full Database Caching . . .

- Primary use case: OLTP
  - Table scans and LOBs
- Disables the LRU algorithm so blocks in the Buffer Cache do not age out
- COMPATIBLE must be 12.0.0 or higher
- If using AMM (MEMORY_TARGET) or ASMM (SGA_TARGET) it is possible the buffer cache will resize, making the cache too small to hold the entire database
- Either size the memory parameters appropriately, or better still set the minimum size of the buffer cache by setting DB_CACHE_SIZE to an large value
- Objects are cached as accessed ... no pre-emptive loading
- LOBs defined as NOCACHE are cached when force full database cache is enabled
- Applies to the CDB and all PDBs when using multitenant
- To recover a control file force full database cache mode must be enabled

# Enabling Full Database Caching . . .

```
conn / as sysdba

SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE force full database caching;
ALTER DATABASE OPEN;

SELECT force_full_db_caching FROM v$database;

FOR
---
YES
```

# In-Memory Column Store . . .

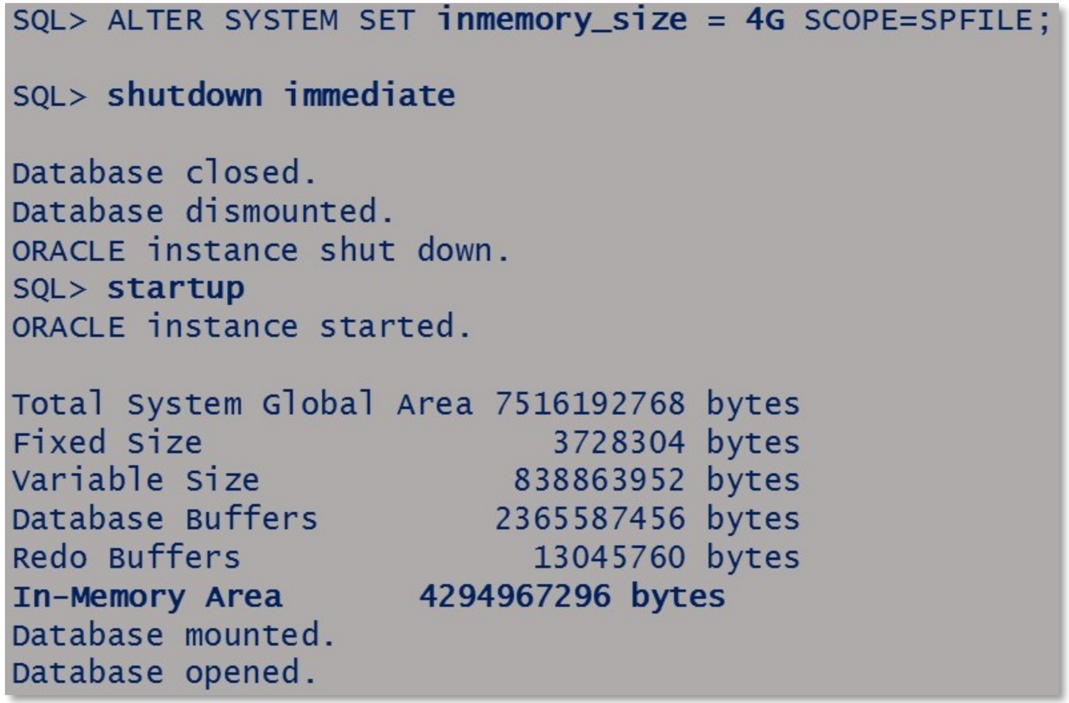
- Primary Use Case: DSS and DW where data will be accessed regularly for aggregation or analysis
- Data stored in row format in the block cache is also stored in a highly optimized columnar format that makes possible extreme compression
- In-Memory columnary storage is substantially more efficient than the HCC available in Exadata and ZFS
- Requires purchase of the Database In-Memory Option
- Data can be populated on-the-fly but is usually populated by identifying specific tables and columns

> In-memory is probably the biggest performance booster out there, because its *not* just about putting things into memory - its about compression, restructuring the rows, using special CPU optimizations etc.
> ~ Connor McDonald ( Oracle ACE Director Alum, Oak Table Network)

- How are you going to put data in memory ... if you don't have enough DRAM?

# Typically . . .

- Examples of Database In-Memory look like this ... a 4G store
- With TidalScale we build them big because we can



```
SQL> ALTER SYSTEM SET inmemory_size = 4G SCOPE=SPFILE;

SQL> shutdown immediate

Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 7516192768 bytes
Fixed Size                  3728304 bytes
Variable Size             838863952 bytes
Database Buffers         2365587456 bytes
Redo Buffers               13045760 bytes
In-Memory Area        4294967296 bytes
Database mounted.
Database opened.
```

www.dbta.com/Editorial/Trends-and-Applications/Tips-for-Accelerating-Analytics-with-Oracle-Database-In-Memory-110880.aspx

- With TidalScale Software Defined Servers we can aggregate memory to 64TB
- We can create SGAs up to ~50TB
- An In-Memory Column Store can easily exceed the amount of available memory due to the high quality of columnar compression

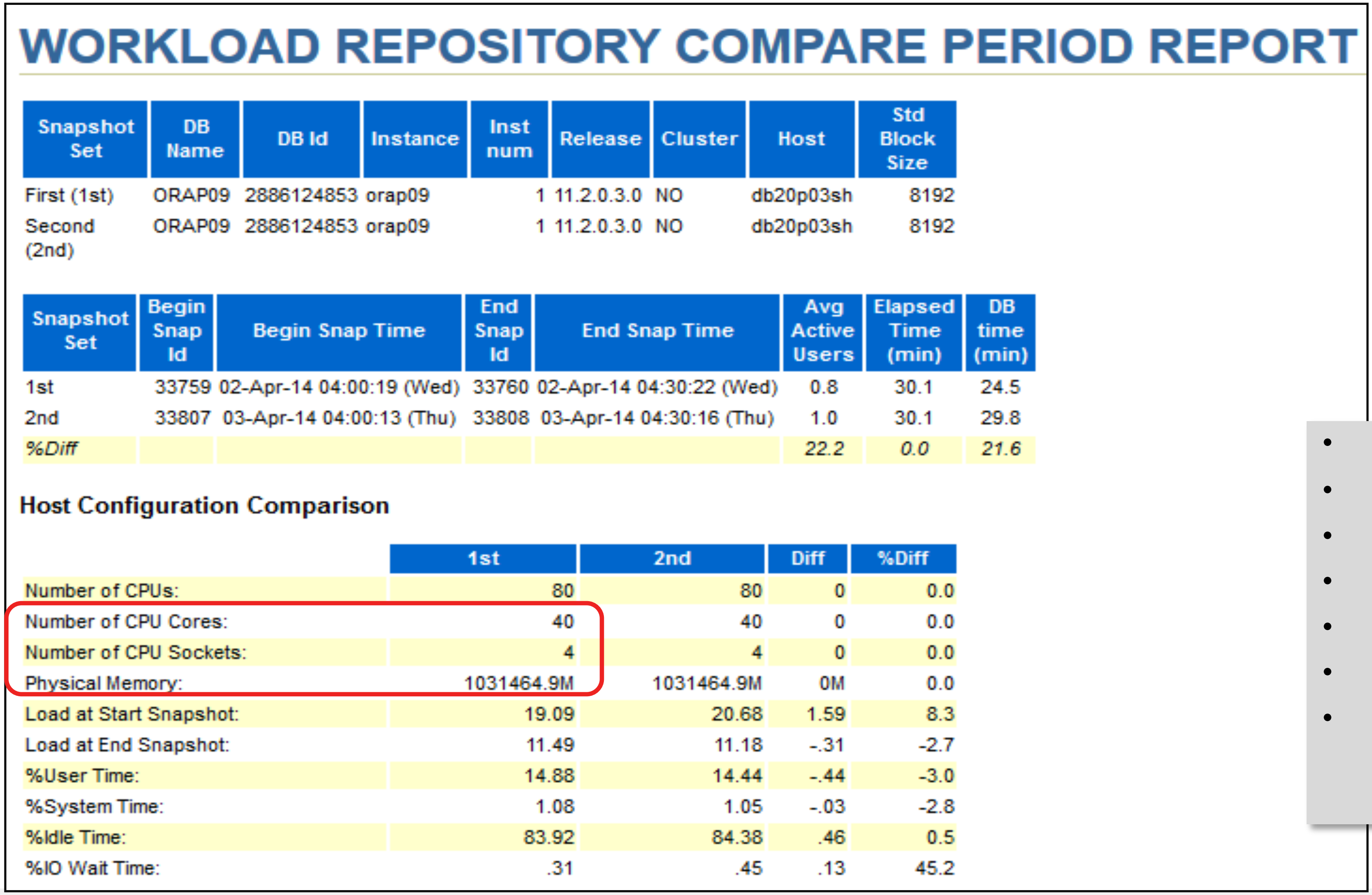
## In-Memory Aggregation . . .

- Primary Use Case: DSS and DW
- **In-Memory Aggregation.** Designed especially for star schema query performance improvements
- Leverages a completely new set of optimizer methods collectively known as *vector transformation* to quickly identify the matching dimension keys for the query and then apply those limiting filters to the fact table
- Vector transformation leverages *single-instruction processing multiple data values* (SIMD) vector processing so that it's possible to read all of the possible values in one or more dimensional row sets *at one time*
- This reduces the need for complex bitmap join index structures to weed out non-matching dimension data
- SIMD vector processing can also benefit analytic functions that require aggregation – SUM, COUNT, AVERAGE, and many others

# In-Memory Joins . . .

- Primary Use Case: DSS and DW
- **In-Memory Joins and Filtering**
- Joining data sources together – especially using hash joins when data sources are enormous – can consume equally huge amounts of memory and cpu
- IMCS leverages *Bloom Filters,* present "under the covers" in the Oracle optimizer since Oracle 10*g*R2, to convert the join to a filter that can be applied efficiently to a massive row sources reducing processing cycles
- IMCS has adapted and leveraged the Oracle Exadata concept of *storage indexes* to identify exactly where data isn't present and often completely avoids scanning vast sections of database objects when applying filters

# This Is What 1TB Of Memory Looks Like To Most DBAs and CFOs



## WORKLOAD REPOSITORY COMPARE PERIOD REPORT

| Snapshot Set | DB Name | DB Id | Instance | Inst num | Release | Cluster | Host | Std Block Size |
|---|---|---|---|---|---|---|---|---|
| First (1st) | ORAP09 | 2886124853 | orap09 | 1 | 11.2.0.3.0 | NO | db20p03sh | 8192 |
| Second (2nd) | ORAP09 | 2886124853 | orap09 | 1 | 11.2.0.3.0 | NO | db20p03sh | 8192 |

| Snapshot Set | Begin Snap Id | Begin Snap Time | End Snap Id | End Snap Time | Avg Active Users | Elapsed Time (min) | DB time (min) |
|---|---|---|---|---|---|---|---|
| 1st | 33759 | 02-Apr-14 04:00:19 (Wed) | 33760 | 02-Apr-14 04:30:22 (Wed) | 0.8 | 30.1 | 24.5 |
| 2nd | 33807 | 03-Apr-14 04:00:13 (Thu) | 33808 | 03-Apr-14 04:30:16 (Thu) | 1.0 | 30.1 | 29.8 |
| *%Diff* | | | | | *22.2* | *0.0* | *21.6* |

### Host Configuration Comparison

| | 1st | 2nd | Diff | %Diff |
|---|---|---|---|---|
| Number of CPUs: | 80 | 80 | 0 | 0.0 |
| Number of CPU Cores: | 40 | 40 | 0 | 0.0 |
| Number of CPU Sockets: | 4 | 4 | 0 | 0.0 |
| Physical Memory: | 1031464.9M | 1031464.9M | 0M | 0.0 |
| Load at Start Snapshot: | 19.09 | 20.68 | 1.59 | 8.3 |
| Load at End Snapshot: | 11.49 | 11.18 | -.31 | -2.7 |
| %User Time: | 14.88 | 14.44 | -.44 | -3.0 |
| %System Time: | 1.08 | 1.05 | -.03 | -2.8 |
| %Idle Time: | 83.92 | 84.38 | .46 | 0.5 |
| %IO Wait Time: | .31 | .45 | .13 | 45.2 |

- 4 Sockets
- 40 cpu cores
- 80 threads
- 20 Oracle EE licenses ($950,000)
- 20 Diag & Tuning Licenses ($250,000)
- Total Licenses: $1,200,000
- Annual Support: $264,000

And all you get is 1TB of memory

TidalScale™

# TidalScale Metrics

- We used HammerDB to build three identical 500GB Oracle 12.2.0.1 databases on Oracle Enterprise Linux 7.4
  - Env 1: 512GB RAM … Oracle installation performed by OUI and DBCA <next><next><next>
  - Env 2: 1024GB RAM ... A 2 node TidalScale pod with Database In-Memory enabled
- Nothing was customized
- The databases are 500GB and identical ... created with the same script
- Adaptive Queries were not disabled … but we've tested both ways
- Evolving Baselines were not disabled … but we've tested both ways
- No Explain Plans were run
- No SQL Tuning was performed
- Every DML statement was generic TPC-H benchmark
- Are you ready to view the results?

TidalScale™

# Test Results: Timing

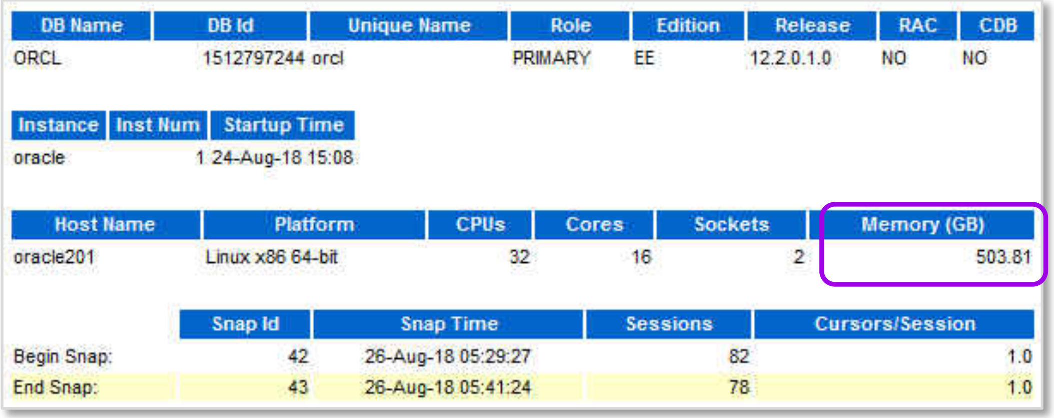
Env 1: Run time – 13,249 seconds

Env 2: Run time – 558 seconds

A 25X performance improvement

# Test Results: AWR Report (1:4)

## Env 1: Bare Metal



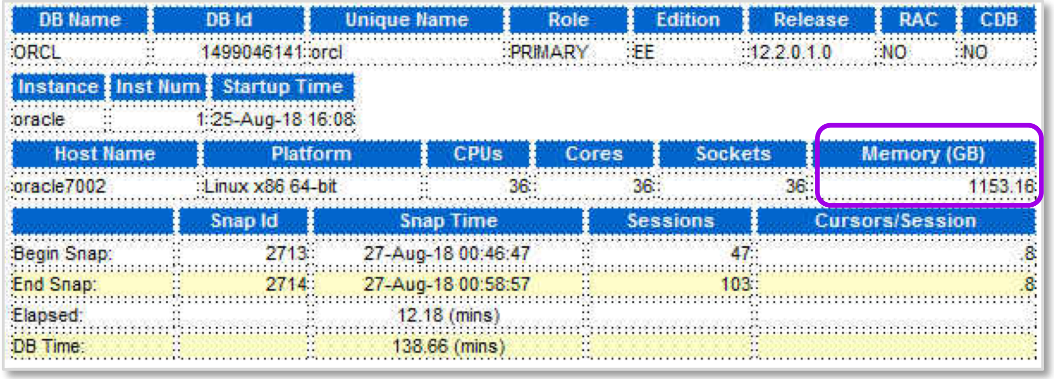
| DB Name | DB Id | Unique Name | Role | Edition | Release | RAC | CDB |
|---|---|---|---|---|---|---|---|
| ORCL | 1512797244 | orcl | PRIMARY | EE | 12.2.0.1.0 | NO | NO |

| Instance | Inst Num | Startup Time |
|---|---|---|
| oracle | 1 | 24-Aug-18 15:08 |

| Host Name | Platform | CPUs | Cores | Sockets | Memory (GB) |
|---|---|---|---|---|---|
| oracle201 | Linux x86 64-bit | 32 | 16 | 2 | 503.81 |

| | Snap Id | Snap Time | Sessions | Cursors/Session |
|---|---|---|---|---|
| Begin Snap: | 42 | 26-Aug-18 05:29:27 | 82 | 1.0 |
| End Snap: | 43 | 26-Aug-18 05:41:24 | 78 | 1.0 |

## Env 2: TidalScale 2 Node Pod



| DB Name | DB Id | Unique Name | Role | Edition | Release | RAC | CDB |
|---|---|---|---|---|---|---|---|
| ORCL | 1499046141 | orcl | PRIMARY | EE | 12.2.0.1.0 | NO | NO |

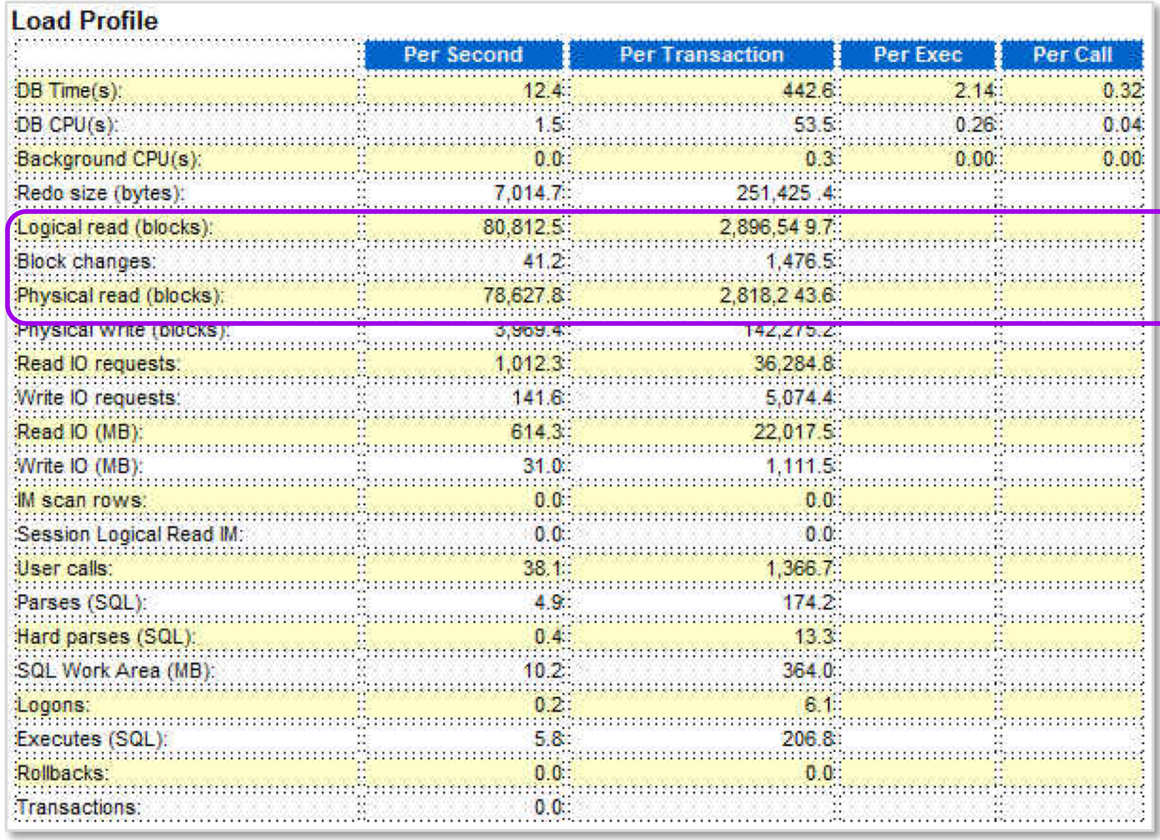
| Instance | Inst Num | Startup Time |
|---|---|---|
| oracle | 1 | 25-Aug-18 16:08 |

| Host Name | Platform | CPUs | Cores | Sockets | Memory (GB) |
|---|---|---|---|---|---|
| oracle7002 | Linux x86 64-bit | 36 | 36 | 36 | 1153.16 |

| | Snap Id | Snap Time | Sessions | Cursors/Session |
|---|---|---|---|---|
| Begin Snap: | 2713 | 27-Aug-18 00:46:47 | 47 | .8 |
| End Snap: | 2714 | 27-Aug-18 00:58:57 | 103 | .8 |
| Elapsed: | | 12.18 (mins) | | |
| DB Time: | | 138.66 (mins) | | |

# Test Results: AWR Report (2:4)

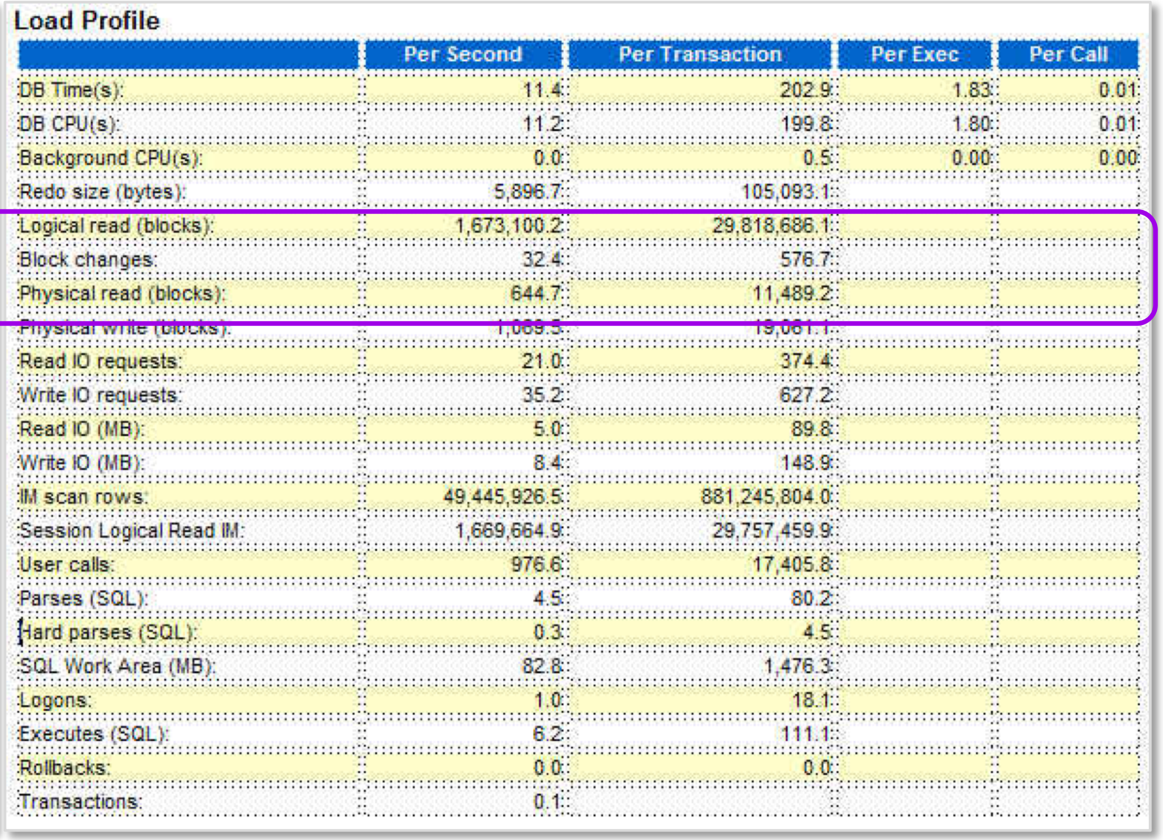
## Env 1: Bare Metal



**Load Profile**

| | Per Second | Per Transaction | Per Exec | Per Call |
|---|---|---|---|---|
| DB Time(s): | 12.4 | 442.6 | 2.14 | 0.32 |
| DB CPU(s): | 1.5 | 53.5 | 0.26 | 0.04 |
| Background CPU(s): | 0.0 | 0.3 | 0.00 | 0.00 |
| Redo size (bytes): | 7,014.7 | 251,425 .4 | | |
| Logical read (blocks): | 80,812.5 | 2,896,54 9.7 | | |
| Block changes: | 41.2 | 1,476.5 | | |
| Physical read (blocks): | 78,627.8 | 2,818,2 43.6 | | |
| Physical write (blocks): | 3,969.4 | 142,275.2 | | |
| Read IO requests: | 1,012.3 | 36,284.8 | | |
| Write IO requests: | 141.6 | 5,074.4 | | |
| Read IO (MB): | 614.3 | 22,017.5 | | |
| Write IO (MB): | 31.0 | 1,111.5 | | |
| IM scan rows: | 0.0 | 0.0 | | |
| Session Logical Read IM: | 0.0 | 0.0 | | |
| User calls: | 38.1 | 1,366.7 | | |
| Parses (SQL): | 4.9 | 174.2 | | |
| Hard parses (SQL): | 0.4 | 13.3 | | |
| SQL Work Area (MB): | 10.2 | 364.0 | | |
| Logons: | 0.2 | 6.1 | | |
| Executes (SQL): | 5.8 | 206.8 | | |
| Rollbacks: | 0.0 | 0.0 | | |
| Transactions: | 0.0 | | | |

## Env 2: TidalScale 2 Node Pod



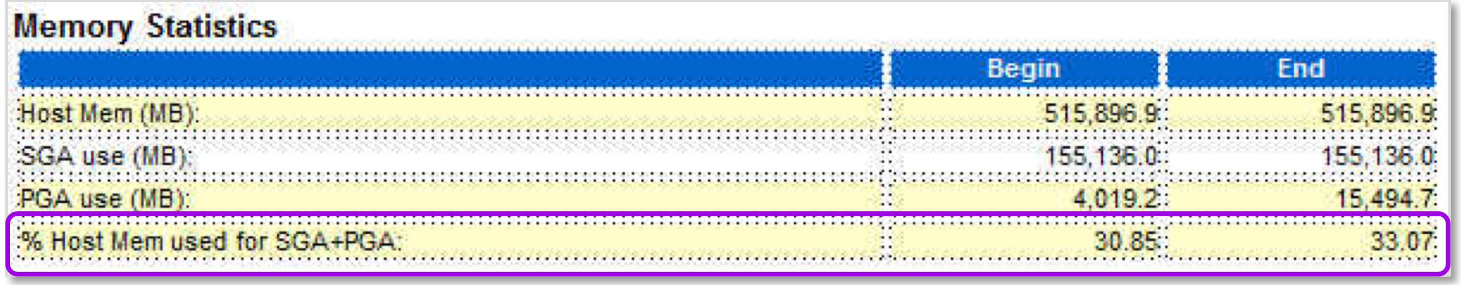
**Load Profile**

| | Per Second | Per Transaction | Per Exec | Per Call |
|---|---|---|---|---|
| DB Time(s): | 11.4 | 202.9 | 1.83 | 0.01 |
| DB CPU(s): | 11.2 | 199.8 | 1.80 | 0.01 |
| Background CPU(s): | 0.0 | 0.5 | 0.00 | 0.00 |
| Redo size (bytes): | 5,896.7 | 105,093.1 | | |
| Logical read (blocks): | 1,673,100.2 | 29,818,686.1 | | |
| Block changes: | 32.4 | 576.7 | | |
| Physical read (blocks): | 644.7 | 11,489.2 | | |
| Physical write (blocks): | 1,069.5 | 19,061.1 | | |
| Read IO requests: | 21.0 | 374.4 | | |
| Write IO requests: | 35.2 | 627.2 | | |
| Read IO (MB): | 5.0 | 89.8 | | |
| Write IO (MB): | 8.4 | 148.9 | | |
| IM scan rows: | 49,445,926.5 | 881,245,804.0 | | |
| Session Logical Read IM: | 1,669,664.9 | 29,757,459.9 | | |
| User calls: | 976.6 | 17,405.8 | | |
| Parses (SQL): | 4.5 | 80.2 | | |
| Hard parses (SQL): | 0.3 | 4.5 | | |
| SQL Work Area (MB): | 82.8 | 1,476.3 | | |
| Logons: | 1.0 | 18.1 | | |
| Executes (SQL): | 6.2 | 111.1 | | |
| Rollbacks: | 0.0 | 0.0 | | |
| Transactions: | 0.1 | | | |

- When you have sufficient memory logical reads replace far slower physical reads

# Test Results: AWR Report (3:4)

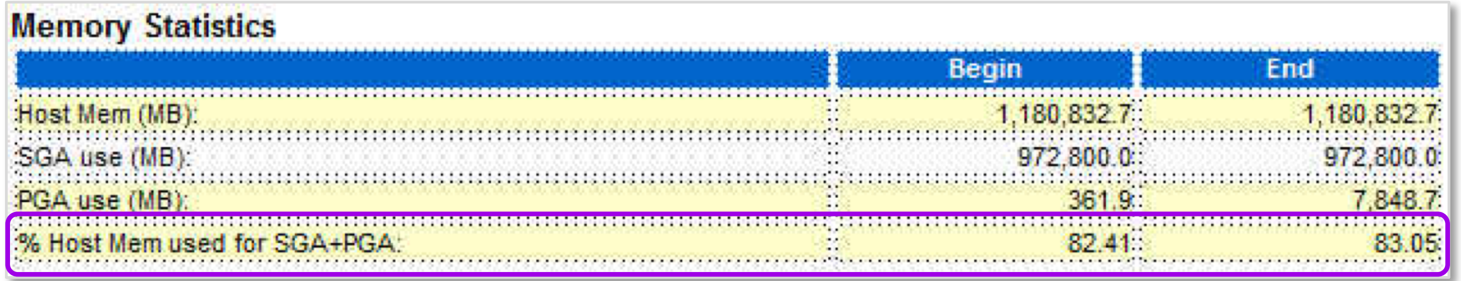
## Env 1: Bare Metal



Memory Statistics

| | Begin | End |
|---|---|---|
| Host Mem (MB): | 515,896.9 | 515,896.9 |
| SGA use (MB): | 155,136.0 | 155,136.0 |
| PGA use (MB): | 4,019.2 | 15,494.7 |
| % Host Mem used for SGA+PGA: | 30.85 | 33.07 |

## Env 2: TidalScale 2 Node Pod

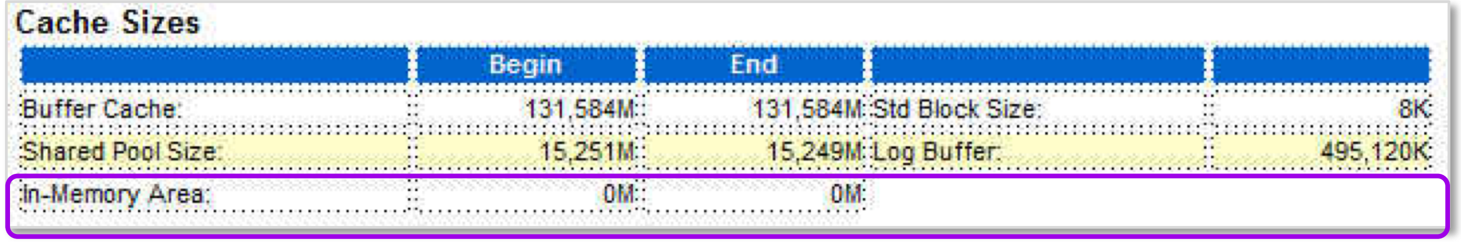


Memory Statistics

| | Begin | End |
|---|---|---|
| Host Mem (MB): | 1,180,832.7 | 1,180,832.7 |
| SGA use (MB): | 972,800.0 | 972,800.0 |
| PGA use (MB): | 361.9 | 7,848.7 |
| % Host Mem used for SGA+PGA: | 82.41 | 83.05 |

# Test Results: AWR Report (4:5)

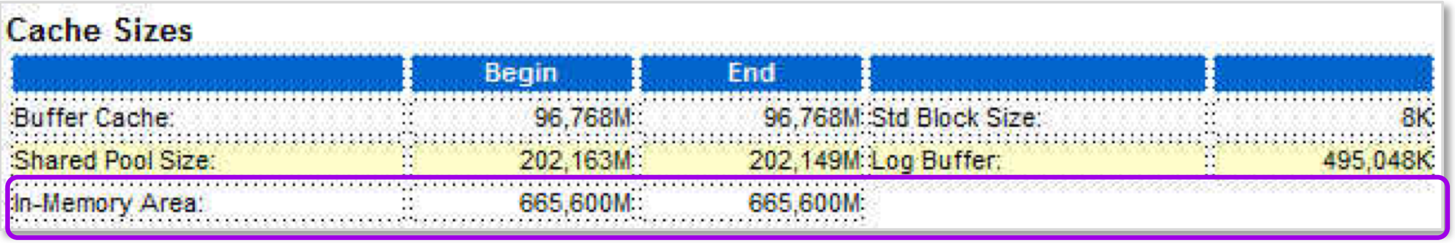
**Env 1: Bare Metal**



Cache Sizes

| | Begin | End | | |
|---|---|---|---|---|
| Buffer Cache: | 131,584M | 131,584M | Std Block Size: | 8K |
| Shared Pool Size: | 15,251M | 15,249M | Log Buffer: | 495,120K |
| In-Memory Area: | 0M | 0M | | |

**Env 2: TidalScale 2 Node Pod**

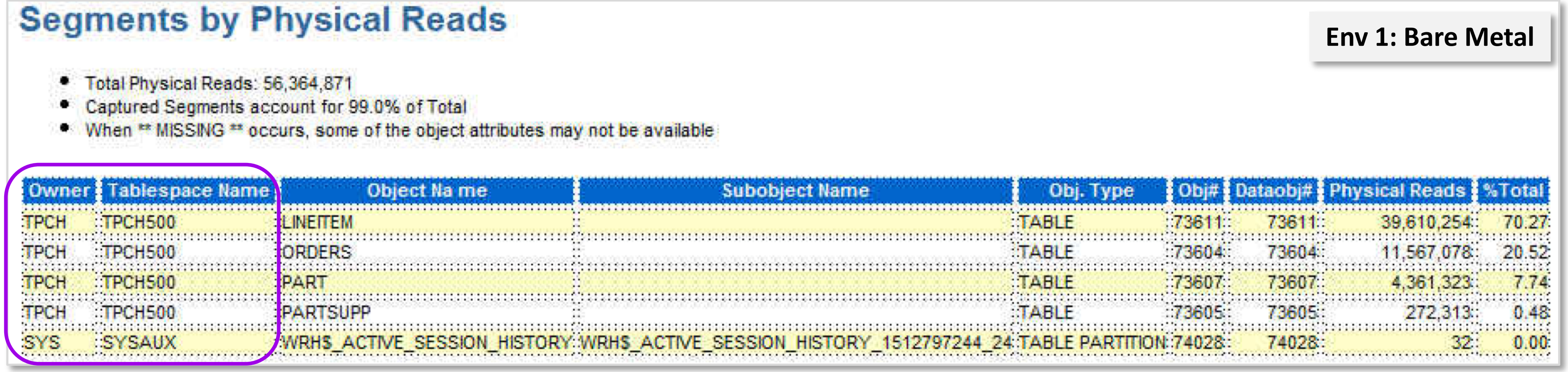


Cache Sizes

| | Begin | End | | |
|---|---|---|---|---|
| Buffer Cache: | 96,768M | 96,768M | Std Block Size: | 8K |
| Shared Pool Size: | 202,163M | 202,149M | Log Buffer: | 495,048K |
| In-Memory Area: | 665,600M | 665,600M | | |

- With TidalScale there is sufficient memory to put 500GB of segments into memory

# Test Results: AWR Report (5:5)

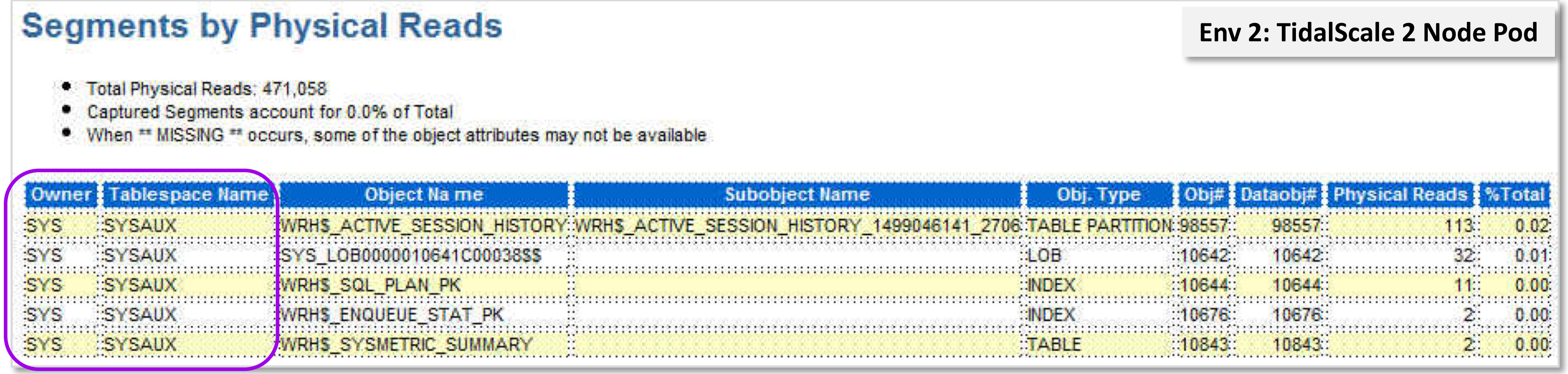


Env 1: Bare Metal

### Segments by Physical Reads

- Total Physical Reads: 56,364,871
- Captured Segments account for 99.0% of Total
- When ** MISSING ** occurs, some of the object attributes may not be available

| Owner | Tablespace Name | Object Na me | Subobject Name | Obj. Type | Obj# | Dataobj# | Physical Reads | %Total |
|---|---|---|---|---|---|---|---|---|
| TPCH | TPCH500 | LINEITEM | | TABLE | 73611 | 73611 | 39,610,254 | 70.27 |
| TPCH | TPCH500 | ORDERS | | TABLE | 73604 | 73604 | 11,567,078 | 20.52 |
| TPCH | TPCH500 | PART | | TABLE | 73607 | 73607 | 4,361,323 | 7.74 |
| TPCH | TPCH500 | PARTSUPP | | TABLE | 73605 | 73605 | 272,313 | 0.48 |
| SYS | SYSAUX | WRH$_ACTIVE_SESSION_HISTORY | WRH$_ACTIVE_SESSION_HISTORY_1512797244_24 | TABLE PARTITION | 74028 | 74028 | 32 | 0.00 |



Env 2: TidalScale 2 Node Pod

### Segments by Physical Reads

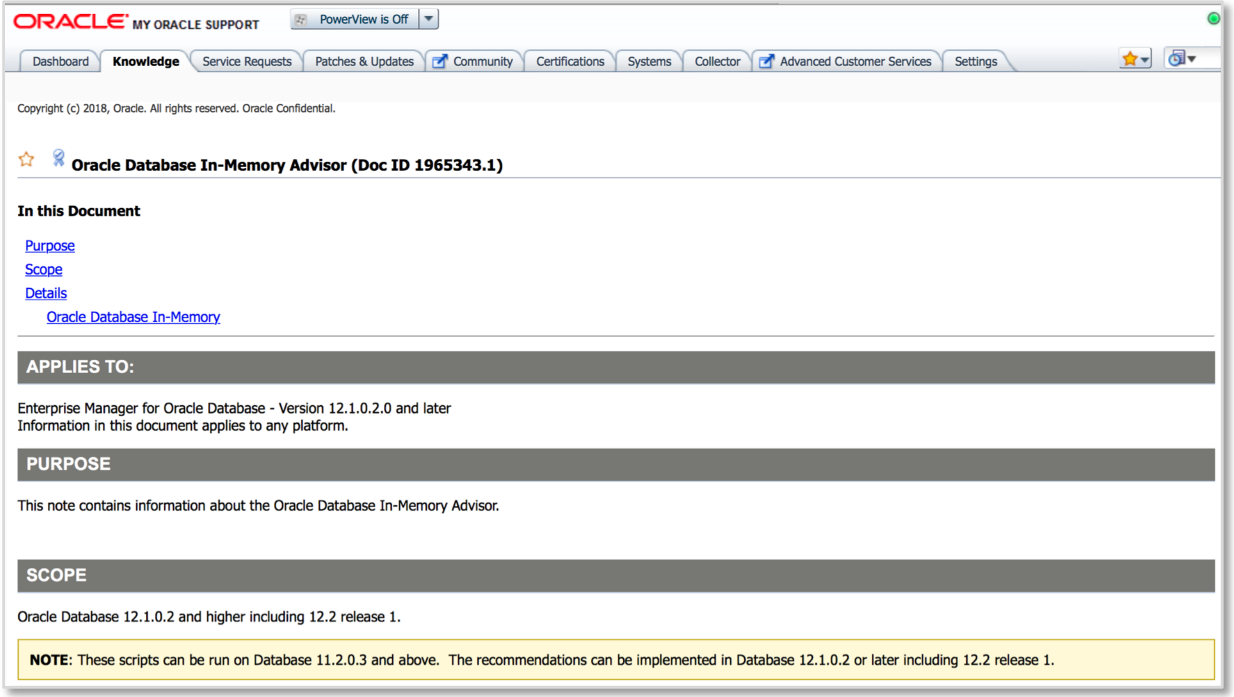
- Total Physical Reads: 471,058
- Captured Segments account for 0.0% of Total
- When ** MISSING ** occurs, some of the object attributes may not be available

| Owner | Tablespace Name | Object Na me | Subobject Name | Obj. Type | Obj# | Dataobj# | Physical Reads | %Total |
|---|---|---|---|---|---|---|---|---|
| SYS | SYSAUX | WRH$_ACTIVE_SESSION_HISTORY | WRH$_ACTIVE_SESSION_HISTORY_1499046141_2706 | TABLE PARTITION | 98557 | 98557 | 113 | 0.02 |
| SYS | SYSAUX | SYS_LOB0000010641C00038$$ | | LOB | 10642 | 10642 | 32 | 0.01 |
| SYS | SYSAUX | WRH$_SQL_PLAN_PK | | INDEX | 10644 | 10644 | 11 | 0.00 |
| SYS | SYSAUX | WRH$_ENQUEUE_STAT_PK | | INDEX | 10676 | 10676 | 2 | 0.00 |
| SYS | SYSAUX | WRH$_SYSMETRIC_SUMMARY | | TABLE | 10843 | 10843 | 2 | 0.00 |

- TPCH tables are in memory so physical reads from TPCH500 are negligible

# DBMS_INMEMORY_ADVISOR

- Released by Oracle Support and announced by Maria Colgan February, 2015
  - Compatible 11.2.0.3 and above
- Use to identify analytic workloads that will benefit from Database In-Memory
- Works by analyzing ASH and AWR data (diagnostic + tuning pack required)
- Download MOS note 1965343.1
- Oracle's tool will advise you on the value of DBIM
- TidalScale will advise you on the performance and security benefits of moving to Software Defined Servers



Copyright (c) 2018, Oracle. All rights reserved. Oracle Confidential.

**Oracle Database In-Memory Advisor (Doc ID 1965343.1)**

**In this Document**

Purpose
Scope
Details
Oracle Database In-Memory

**APPLIES TO:**

Enterprise Manager for Oracle Database - Version 12.1.0.2.0 and later
Information in this document applies to any platform.

**PURPOSE**

This note contains information about the Oracle Database In-Memory Advisor.

**SCOPE**

Oracle Database 12.1.0.2 and higher including 12.2 release 1.

**NOTE**: These scripts can be run on Database 11.2.0.3 and above. The recommendations can be implemented in Database 12.1.0.2 or later including 12.2 release 1.

## Wrap Up . . .

- Memory is 300X faster than flash
- The more of your data you can cache in memory the more your performance will improve due to reducing PIO and increasing LIO
- If you are running OLTP loads use FORCE FULL to disable the LRU algorithm
- If you are running DW or DSS loads they will dramatically benefit from Database In-Memory
  - To  not be limited to a few tables and a few columns
  - To avoid having to invest the time required to figure out which ones
  - Use TidalScale Software Defined Servers and build the right environment
- TidalScale Software Defined Servers let you create an ideal environment with the cpu and memory customized to
  - Improve performance
  - Control your licensing cost

TidalScale™

# Next Steps

**Contact me directly to**

- Answer questions about TidalScale Software Defined Servers
- Present TidalScale Software Defined Servers to your team
- Identify opportunities in your organization for Software Defined Servers

**daniel.morgan@tidalscale.com**