Optimizing the Oracle Database

on Software Defined Motherboards

18c ORACLE Database

TidalScale™

Sharding Motherboards Since 2013

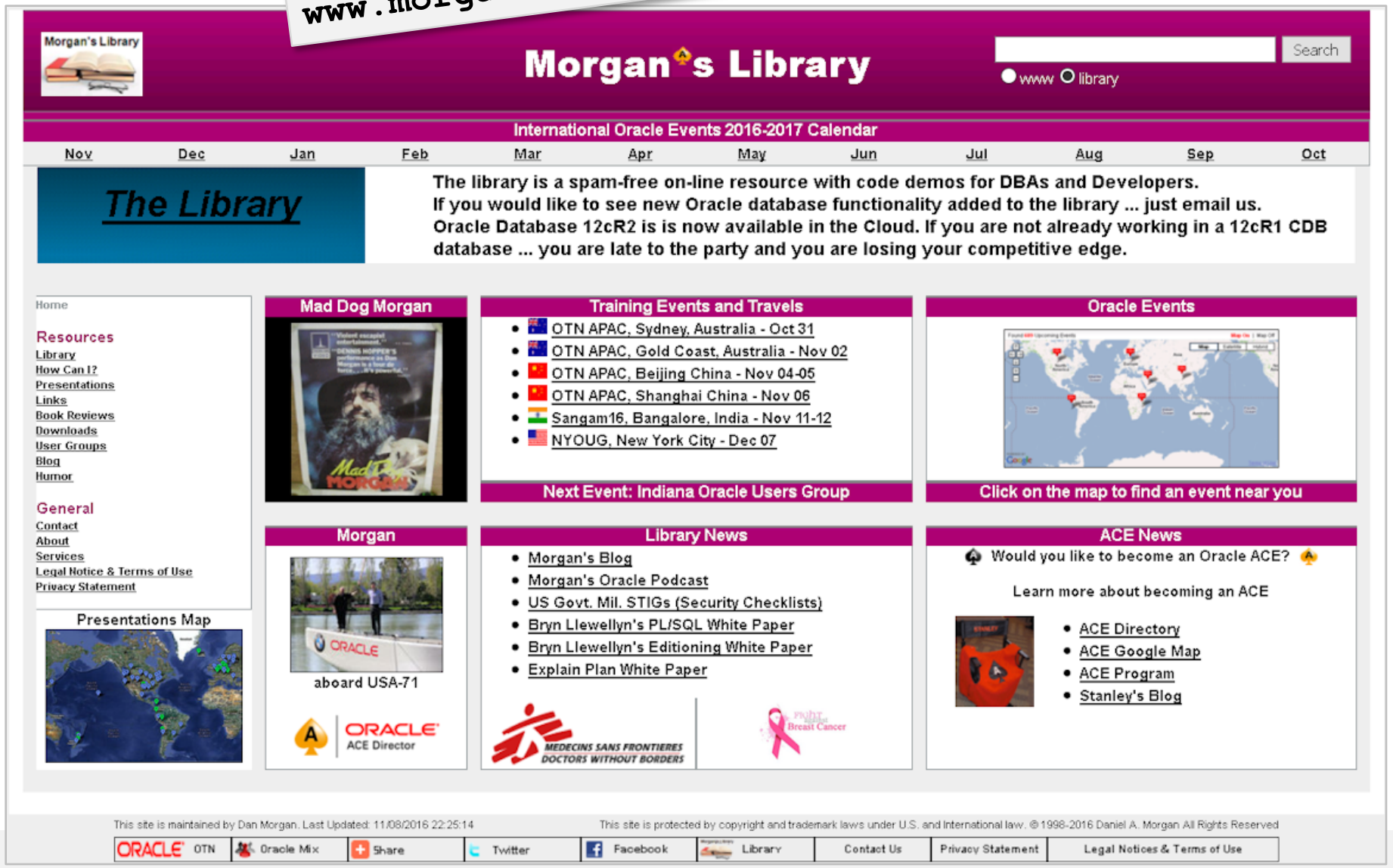
daniel.morgan@tidalscale.com

- Director of Applications @ TidalScale
- Oracle ACE Director Alumni
- 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



TidalScale™

Performance Problems Have Serious Consequences . . .

- Internal and External customers have expectations
- They have 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

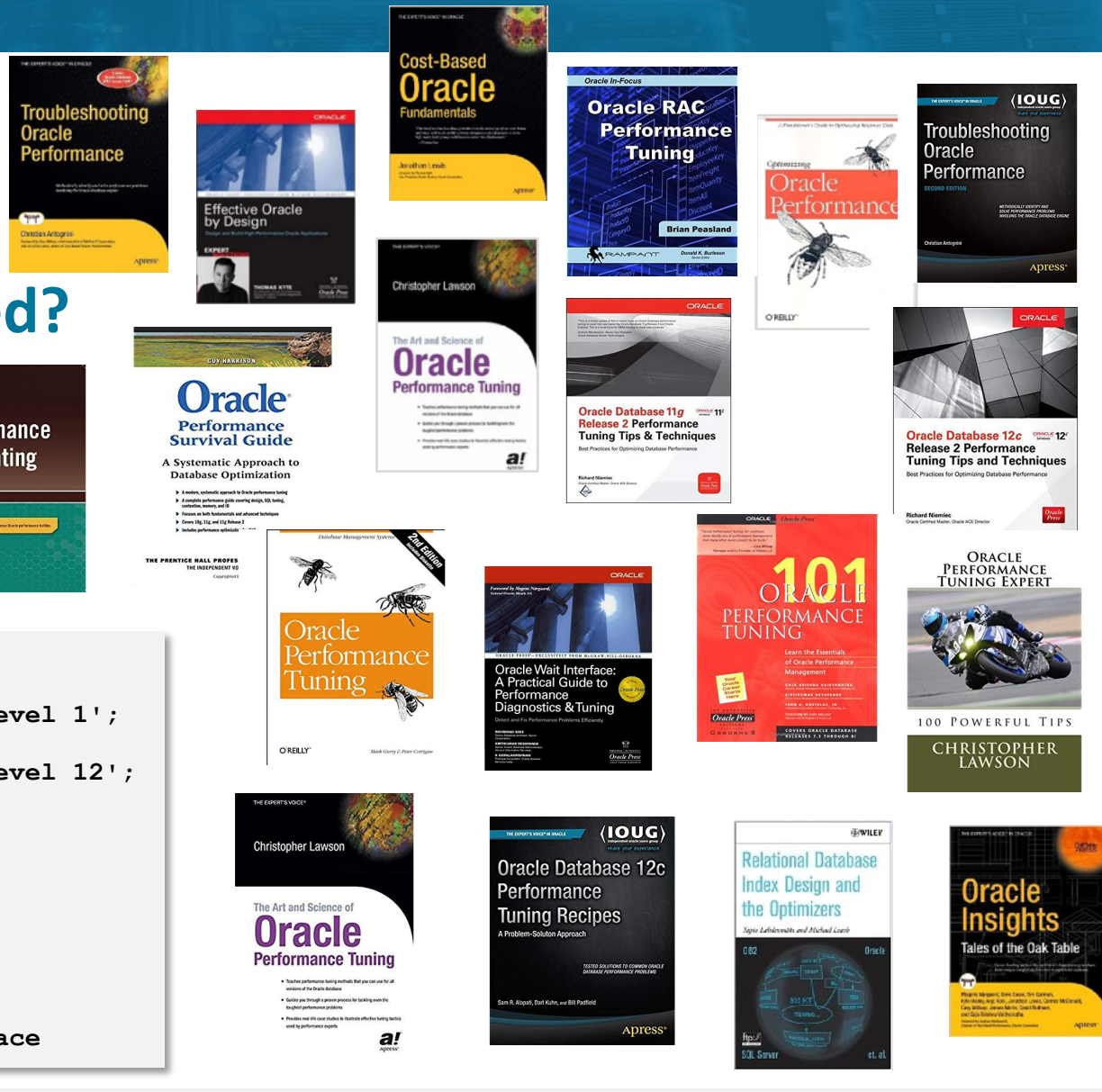
- Stability
- Security
- Scalability
- Usability
- Performance

TCO

- Affordability

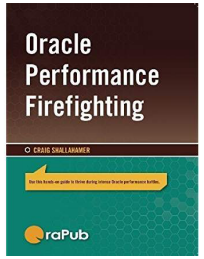
The History of Oracle Performance Tuning . . .

How Many Books Have You Read?



How Many Tools Have You Deployed?

- DBMS_SUPPORT (version 7.2)
- DBMS_TRACE (version 8.1.5)
- DBMS_MONITOR (version 10gR1)
- Oracle Enterprise Manager (OEM)



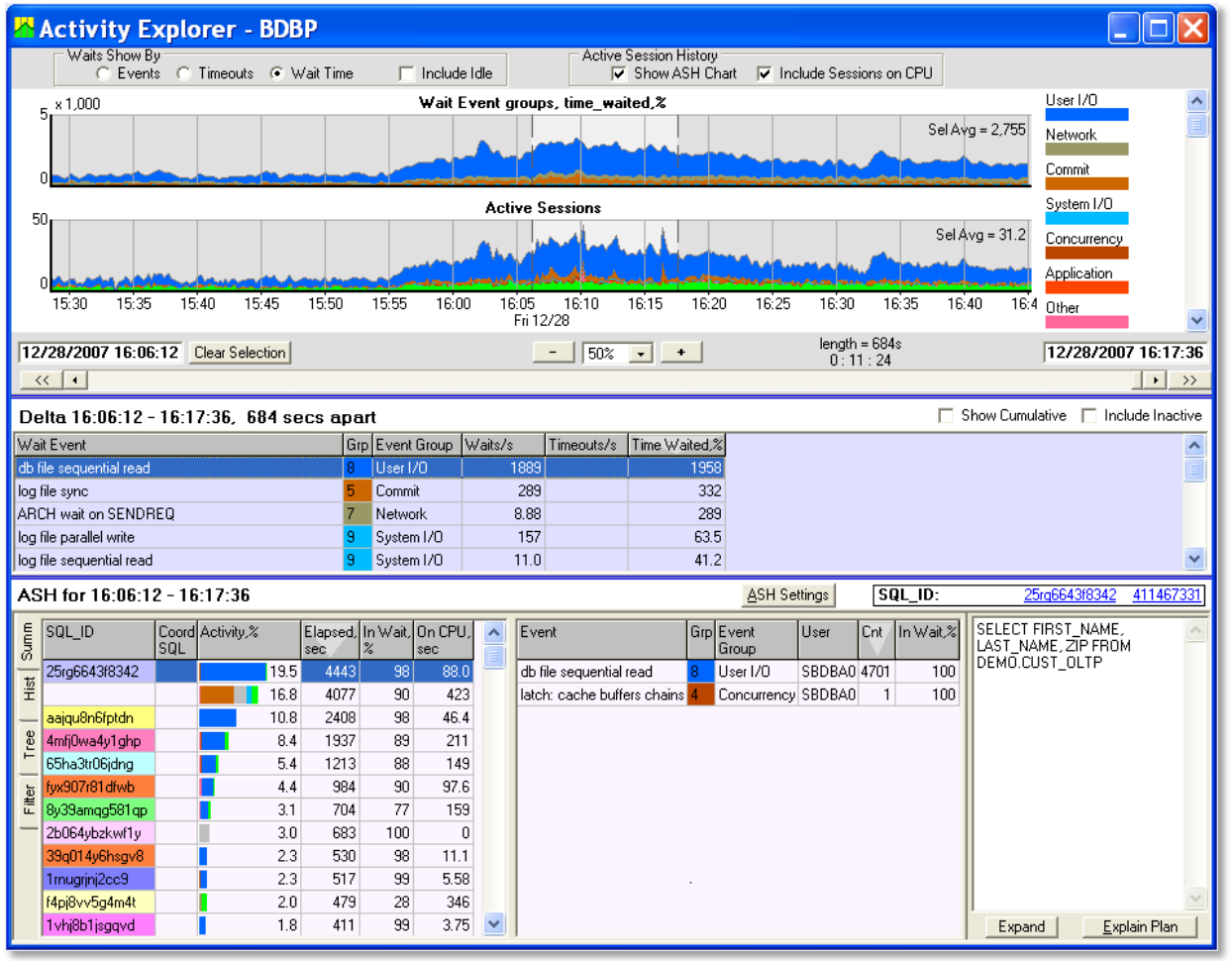

```
ALTER SESSION SET tracefile_identifier = 'test_plan1';

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
```

TidalScale™

How Many Tools Have You Purchased?

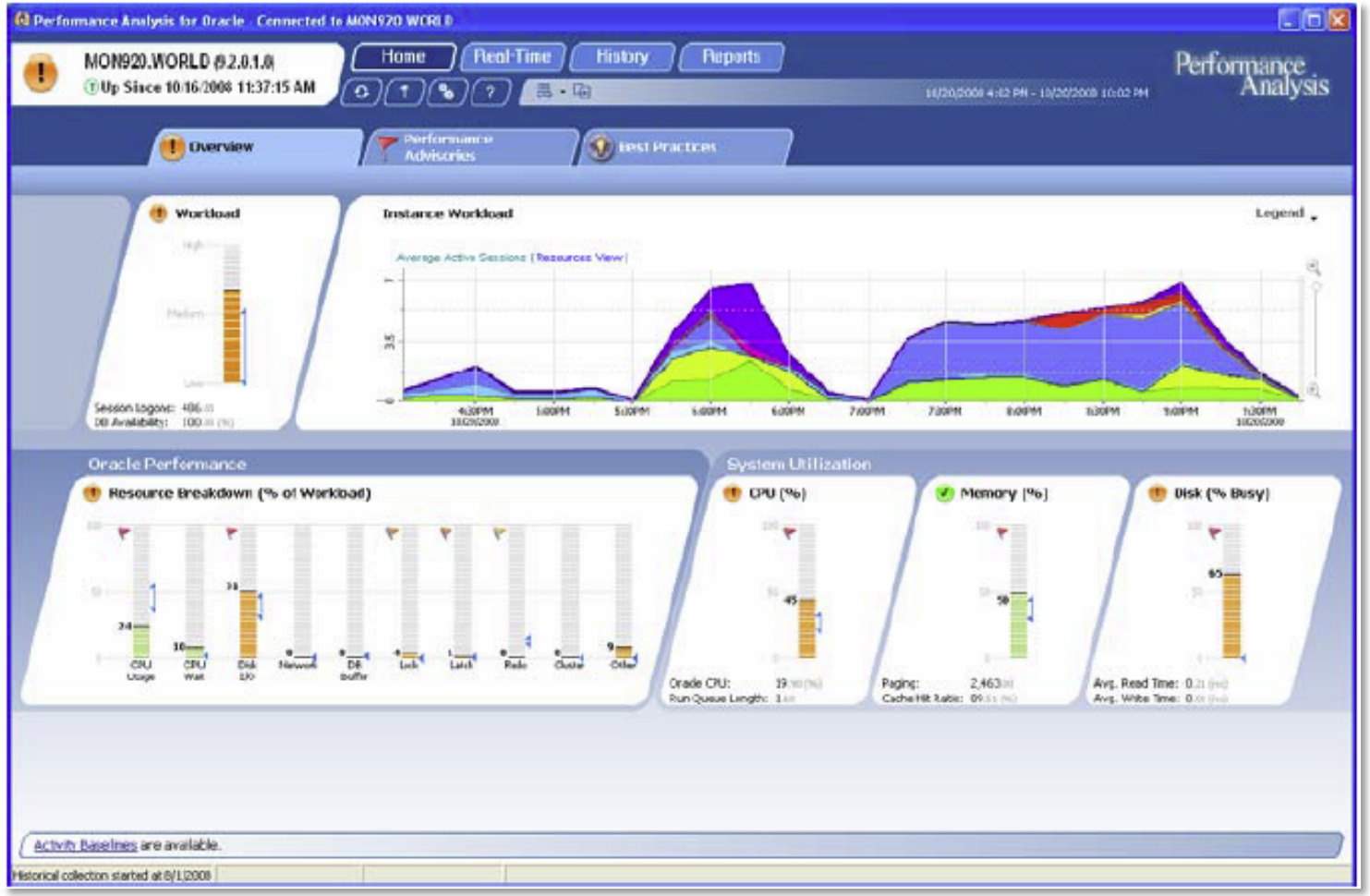
TidalScale™

How Many Tools Have You Purchased?



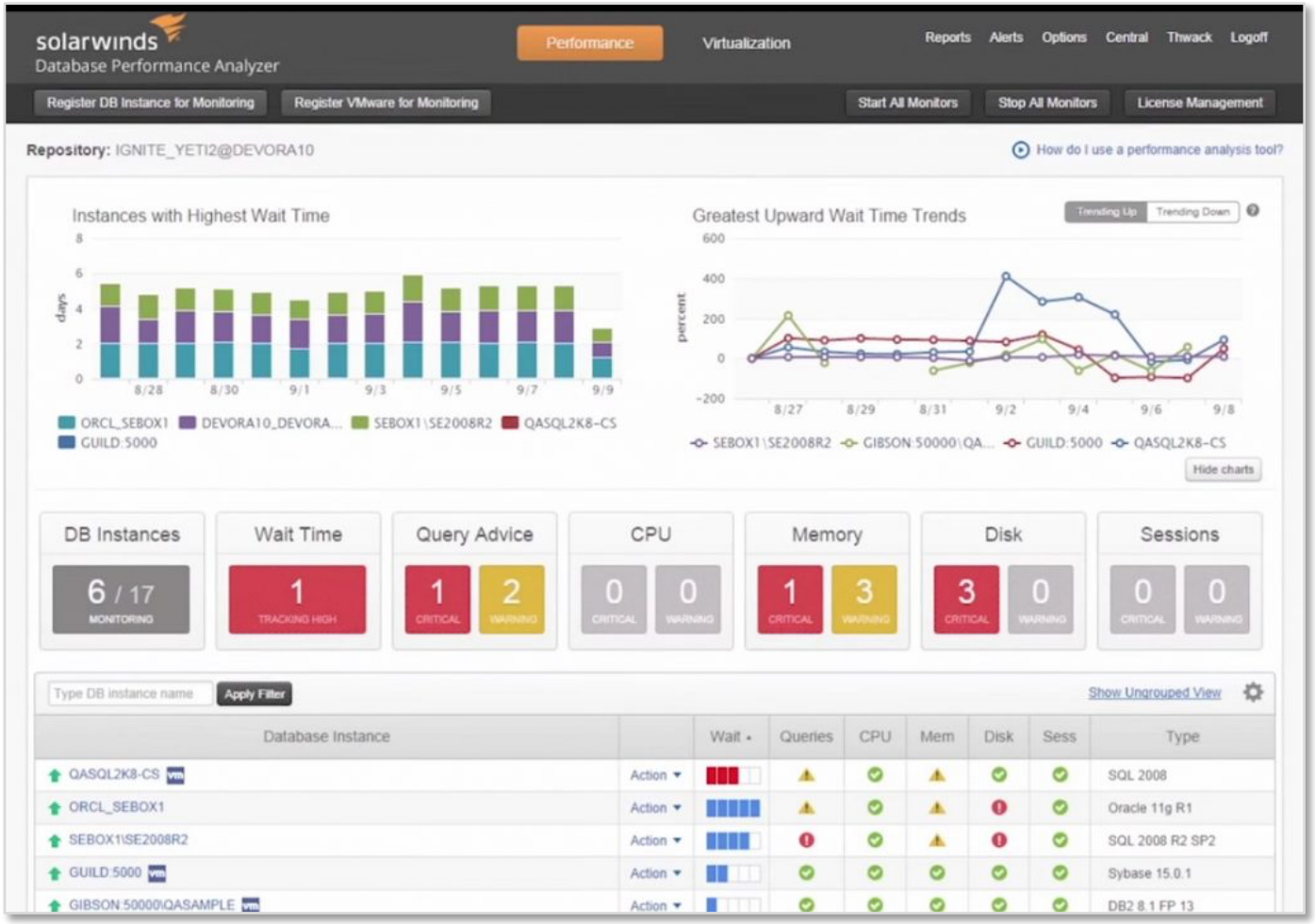
TidalScale™

How Many Tools Have You Purchased?



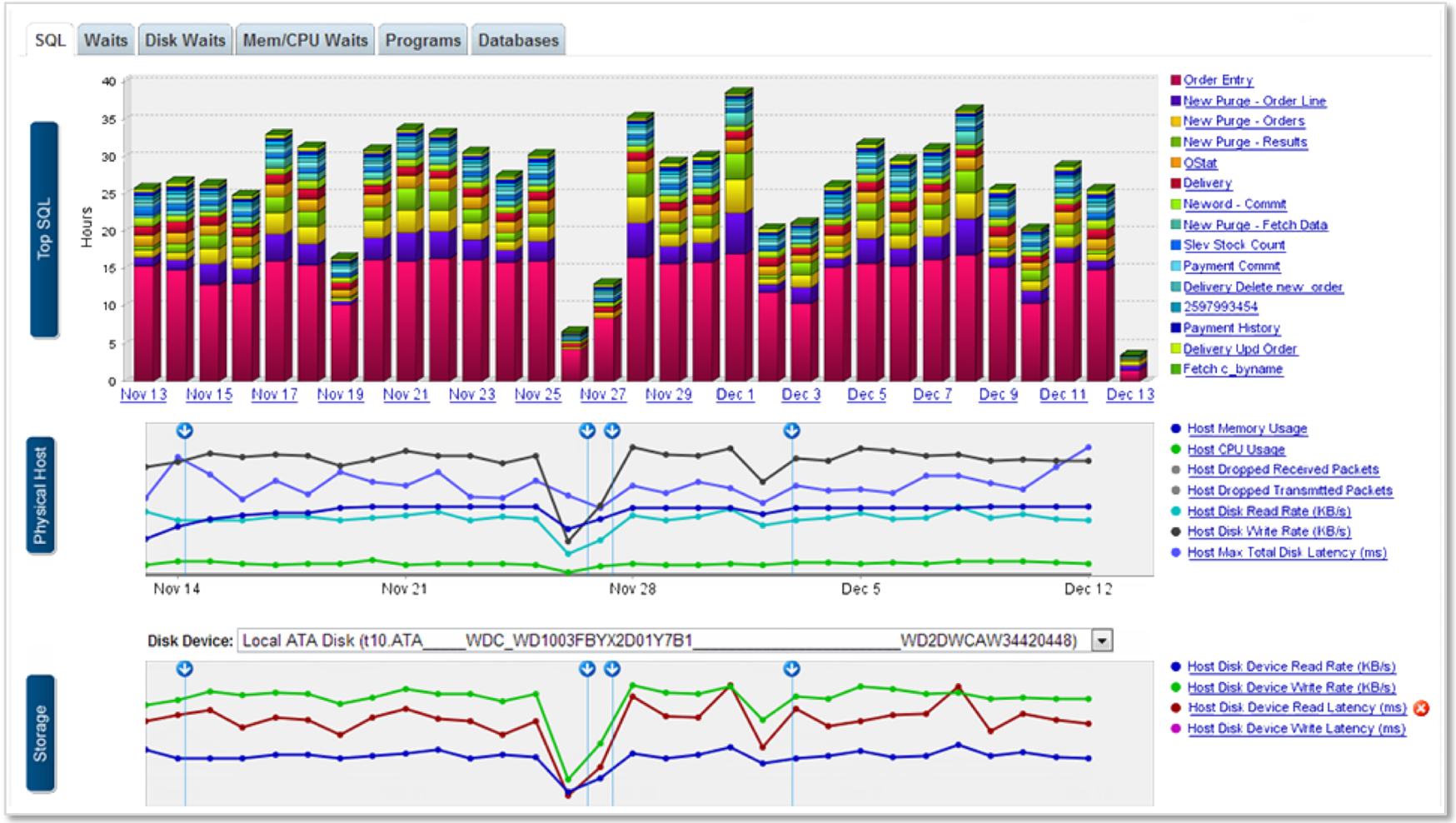
TidalScale™

How Many Tools Have You Purchased?



TidalScale™

How Many Tools Have You Purchased?



TidalScale™

We Know In-Memory Computing Is Faster . . .

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

TidalScale™

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

TidalScale™

ASMM and AMM Help But Are Far From Perfect

```
SQL> SELECT component, parameter, start_time, end_time, initial_size, target_size, final_size
  2  FROM v$memory_resize_ops
  3* WHERE initial_size + target_size + final_size <> 0;

COMPONENT             PARAMETER            START_TIME          END_TIME            INITIAL_SIZE TARGET_SIZE FINAL_SIZE
--------------------- -------------------- ------------------- ------------------- ------------ ----------- ----------
shared pool           shared_pool_size     05-OCT-2018 08:14:29 05-OCT-2018 08:14:29            0   503316480  503316480
large pool            large_pool_size      05-OCT-2018 08:14:29 05-OCT-2018 08:14:29            0   150994944  150994944
SGA Target            sga_target           05-OCT-2018 08:14:29 05-OCT-2018 08:14:29            0  2550136832 2550136832
DEFAULT buffer cache  db_cache_size        05-OCT-2018 08:14:29 05-OCT-2018 08:14:29   1862270976  1862270976 1862270976
java pool             java_pool_size       05-OCT-2018 08:14:29 05-OCT-2018 08:14:29            0    16777216   16777216
PGA Target            pga_aggregate_target 05-OCT-2018 08:14:29 05-OCT-2018 08:14:29            0   855638016  855638016
DEFAULT buffer cache  db_cache_size        05-OCT-2018 08:14:29 05-OCT-2018 08:14:29            0  1862270976 1862270976
DEFAULT buffer cache  db_cache_size        05-OCT-2018 08:15:28 05-OCT-2018 08:15:29   1862270976  1979711488 1979711488
large pool            large_pool_size      05-OCT-2018 08:15:28 05-OCT-2018 08:15:29    150994944    33554432   33554432
```

TidalScale™

Oracle Database Deployment Strategies . . .

- In the 1980s and 90s we deployed databases in what is referred to as a Client-Server architecture
- Databases were, for the most part, on Unix servers in the data center and applications were installed locally on customer's Windows desktops
- Every resource was slow: networks, storage, cpu, and we did what we could to leverage every possible advantage such as putting data files on RAW disk
  - Disks were small ... but so were our most of our databases
- Beginning in about 2000 and continuing through today Client-Server has been replaced by n-Tier architecture with databases still on Unix in the data center
- Applications have come back to the data center and are being hosted using a combination of Web Servers and Application Servers supplemented by in-memory caches such as Coherence

TidalScale™

Oracle Database Deployment Strategies . . .

- Along with n-Tier architecture we have 10gEth, 25gEth, Jumbo Frames, InfiniBand
- Storage includes solid state drives and new technologies such as ASM
- Servers architecture has greatly improved with many advancements including newer more capable processors
- But most importantly we now have
  - Software defined networks
  - Thin provisioned software defined storage
  - And now the ability to use software defined servers for optimal provisioning

Software Defined Servers: The Performance Business Case . . .

- Software Defined Servers allow us to create a single Linux O/S environment of any size
- We self-limit ourselves to 64TB only because that is the largest size that Oracle and Redhat will support
- While solid state technology has made persistent storage far faster than spinning disk it is still 1000X slower than DRAM
- With TidalScale Software Defined Servers leveraging Oracle's In-Memory Database Option and other technologies we can put the of Oracle Databases completely into memory
- For the first time technologists can improve performance by more than an order of magnitude putting them ahead of the curve of growing data sets

Software Defined Servers: The Security Business Case . . .

- If you have a 4 node RAC cluster you have
  - 4 physical servers that must be secured
  - 4 copies of Linux that must be patched, upgraded, and secured
  - 4 copies of Oracle Grid Infrastructure that must be patched, upgraded, and secured
  - 4 Oracle Home directories that must be patched, upgraded, and secured
  - 4 Oracle Listeners with Scan IPs and VIPs
- We call that a large attack surface
- If you have a 4 node TidalPod you have
  - 4 physical servers that must be secured
  - 1 copy of Linux to patch, upgrade, and secure
  - 1 copy of Oracle Grid Infrastructure that must be patched, upgraded, and secured
  - 1 Oracle Home that must be patched, upgraded, and secured
  - 1 Oracle listener with no Scan IPs or VIPs
- That is a lot fewer things to maintain and secure ... a much smaller attach surface

TidalScale™

Software Defined Servers: The Licensing Business Case . . .

- Software Defined Servers let you separate OEM fixed relationships between sockets, cores, threads, and memory
- Use any Intel Processor on any motherboard
- Get the threads you need with the minimum number of cores reducing licensing
- Which gets you the cpu with the lowest core count?
  - 4 two socket servers?
  - 2 four socket servers?
  - 1 eight socket server?
- Unlike with VMware that takes, and segments, a single physical server TidalScale Software Defined Servers allow you to create Linux resources from combining multiple physical servers up to 64TB of DRAM

TidalScale™

Wrap Up . . .

- Oracle Database SGA, PGA, Buffer Cache, etc.
  - Areas, Caches, Pools
  - Full Database Caching
  - Database In-Memory
  - Memoptimize Pool
  - Private Temporary Tables
- GoldenGate: Uses the SGA to cache for long running DML transactions
  - Integrated Capture
  - Parallel Integrated Capture
- Coherence: Caches data for WebLogic
- OBIEE
- Hyperion + Essbase
- TimesTen Database

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

TidalScale™