

<IOUG>

# A Day of Real-World Performance

Andrew Holdsworth, Tom Kyte, Graham Wood

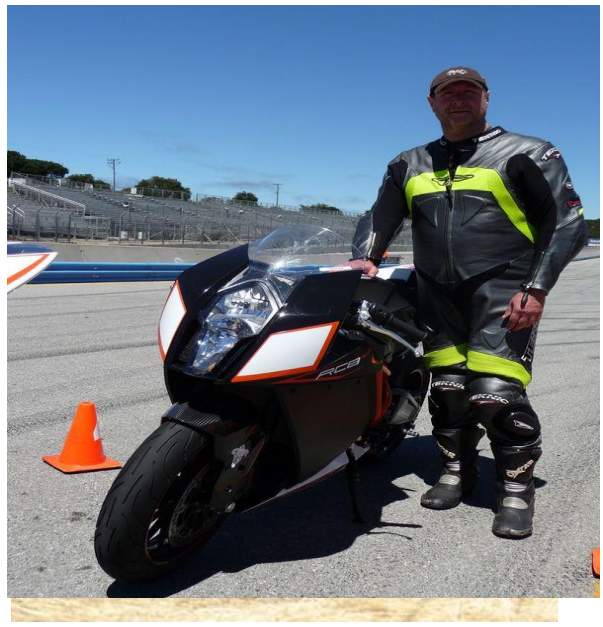
# Introductions

Andrew Holdsworth

- 26 Years at Oracle
- Vice President Real World Performance
  - Good performance is rarely an accident
  - Most people get the systems they deserve
  - Good enough rarely is, aspire for excellence not good enough.



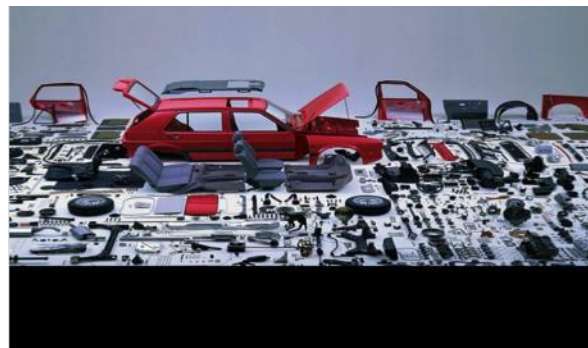
# Graham Wood Architect Server Technologies



# Recent Results 1000X Project



Baseline: 4.3 Hours  
 Code Changes: 4.3 Hours  
 Correct Usage: 29 Secs  
 Bug Fixes: 12 Secs  
 Final: 12 Secs  
 Speed up: **1355.57**



Baseline: 2.4 Days  
 Code Changes: 27 Mins  
 Correct Usage: 7.5 Mins  
 Bug Fixes: 4.5 Mins  
 Final: 4.5 Mins  
 Speed up: **768**



Baseline: 2.5 Hours  
 Code Changes: 2.5 Hours  
 Correct Usage: 4 Secs  
 Bug Fixes: 0.90 Secs  
 Final: 0.90 Secs  
 Speed up: **9000**

# How Does Your Organization Do Performance ?

- Conventional

- Focus is on “Good Enough” or “What the Business Needs”
- Process Orientated/Part of QA
- Spends most the time on Platform Tuning Issues
- Only changes things within limited scope
- Bottom up tuning approach
- Looking for incremental gains

- Real-World

- Focus on excellence and what the HW and SW can do
- Innovate excellent performance and add intellectual property to your code
- Everything is within scope
- Holistic top down approach
- Focus on orders of magnitude gains

<IOUG>

# Real-World Performance OLTP

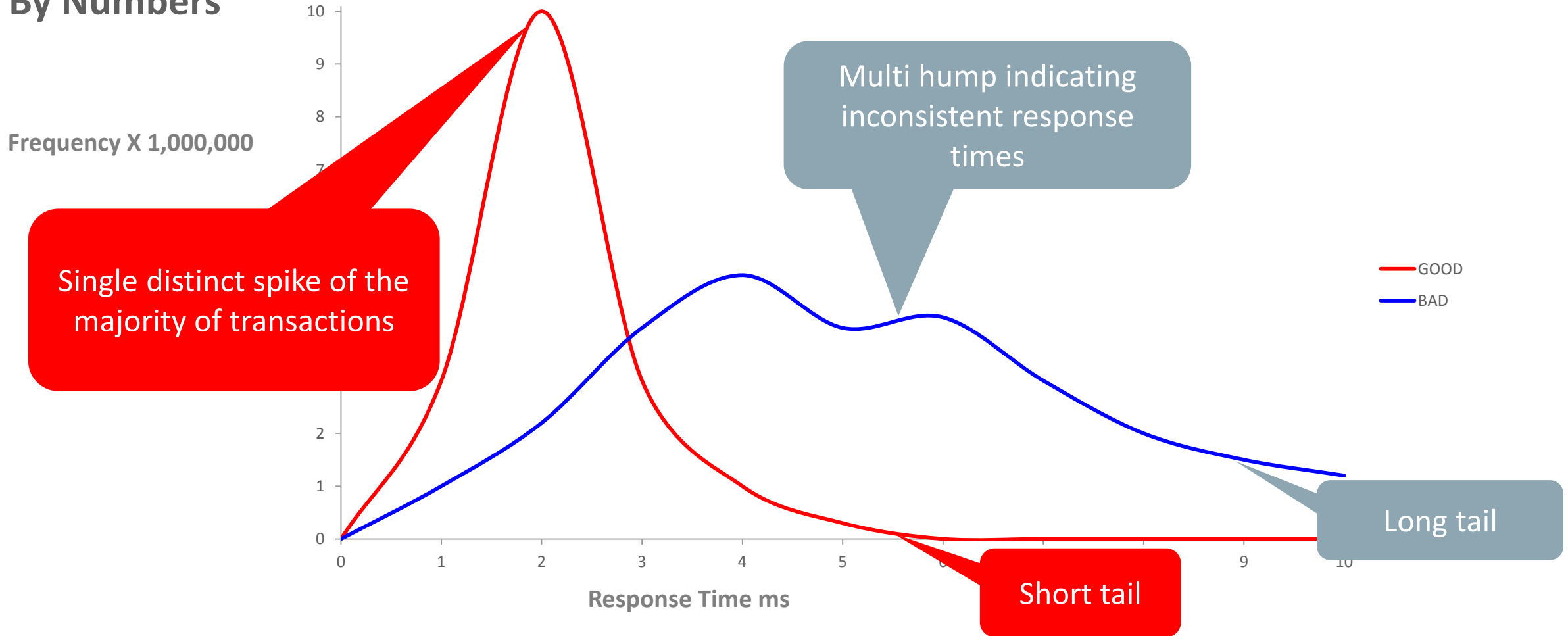
# Response Time

## What it means

- Response time defines your users (customers) experience
- Response time is a measure of performance quality
- Consistency of response time is an equally important measure of performance quality
- If response time is not consistent, bad things happen !

# Response Time

## By Numbers





# Response Time - Demo

## Observations

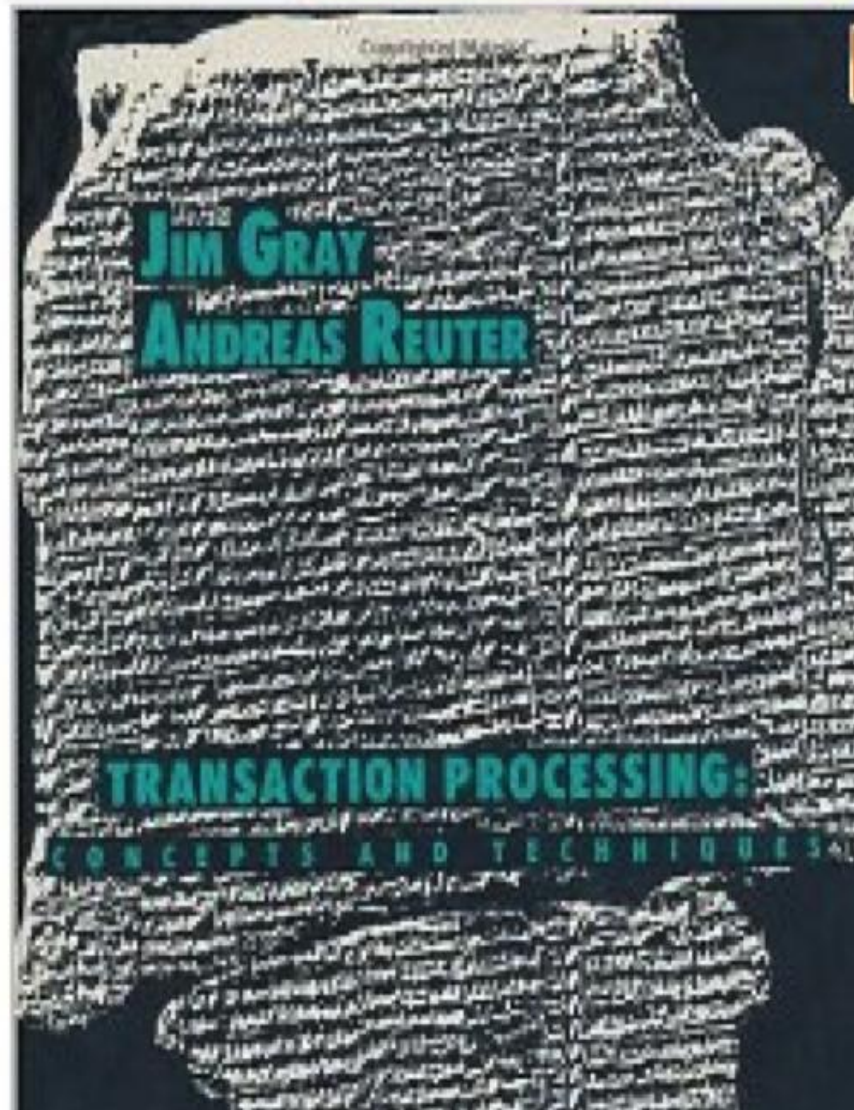
- Users experiencing poor response time
- Low overall system throughput
- Wait events observed in the database
- Culture of blame:
  - Blame the database for all performance issues
  - Development blames the DBA
  - The DBA blames the SW/HW or system administrators

# Response Time

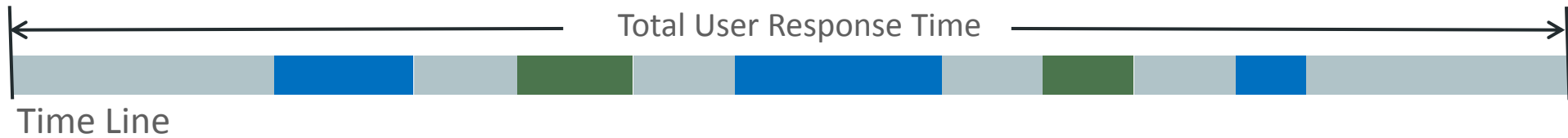
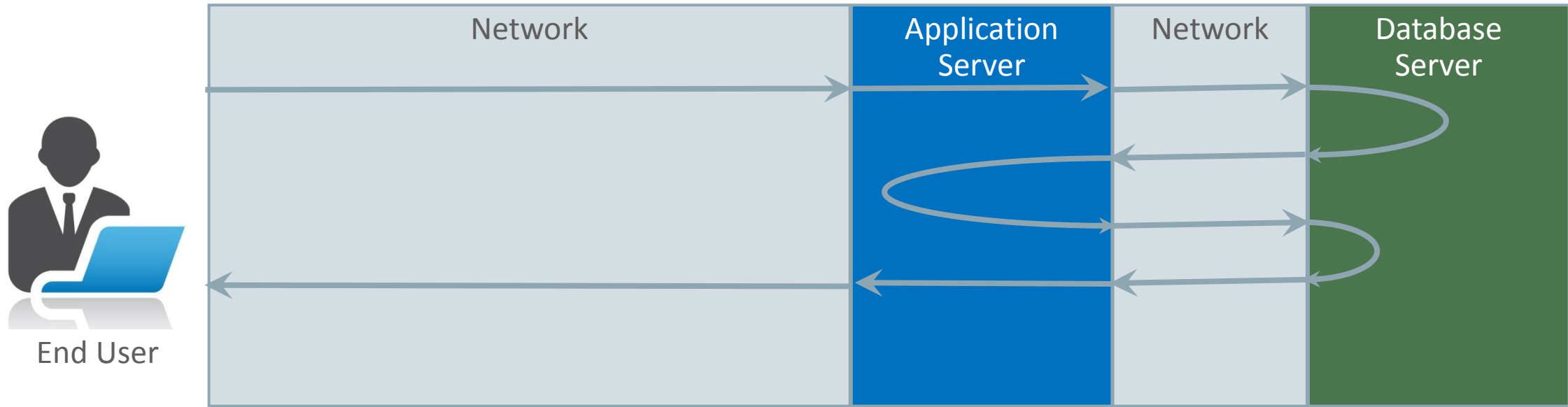
## Application Server Bottleneck

- Data analysis shows:
  - Small proportion of the actual response time is in the database
  - Majority of response time spent in application logic
  - CPU is overloaded on the application servers
  - Potential root cause:
    - Capacity planning mistake ?
    - Application code change last week ?

# Some Computer Science Basics

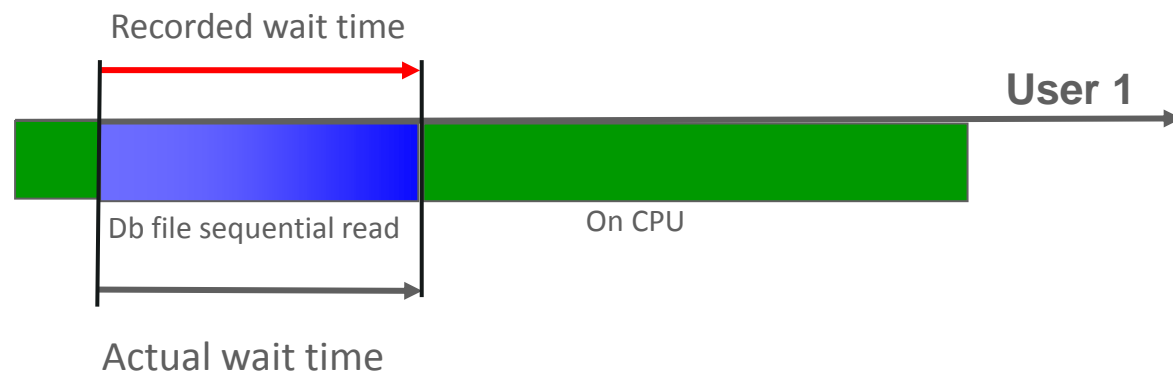


# Response Time v DB Time v Latency

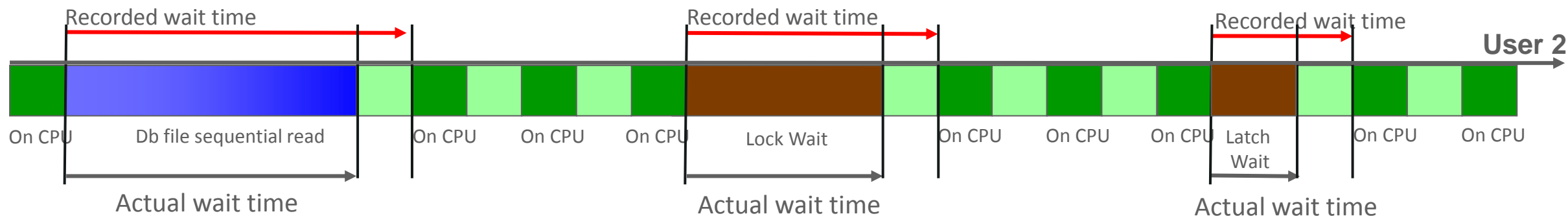


# Database Time – Total time spent in database

## ON IDLE SYSTEM



## ON DEGRADED SYSTEM



# Latency - Some Important Numbers

## Best Block Access Speeds

Block Location	Access Time
L2 CPU cache	~ 1 nano sec ( $10^{-9}$ )
Virtual Memory	~ 1 micro sec ( $10^{-6}$ )
NUMA Far Memory	~ 10 micro sec ( $10^{-6}$ )
Flash Memory (PCI)	~ 0.01 milli sec ( $10^{-3}$ )
Flash Memory (Networked)	~ 0.1 milli sec ( $10^{-3}$ )
Disk I/O	~ 1-10 milli sec ( $10^{-3}$ )

# Database Performance Core Principles

- The Oracle database is a process based architecture and to perform efficiently each process requires:
  - To be efficiently scheduled by the O/S until the process completes the SQL statement, or blocks on an operation required to complete the SQL statement e.g. Disk I/O
  - If the process has to fight to get scheduled, or needs to be scheduled for an over extended period of time due to SQL inefficiencies, or any blocking operation takes a long time, then database performance will be poor
- Database performance engineers spend most of their time looking for CPU-consuming processes and eliminating blocking events

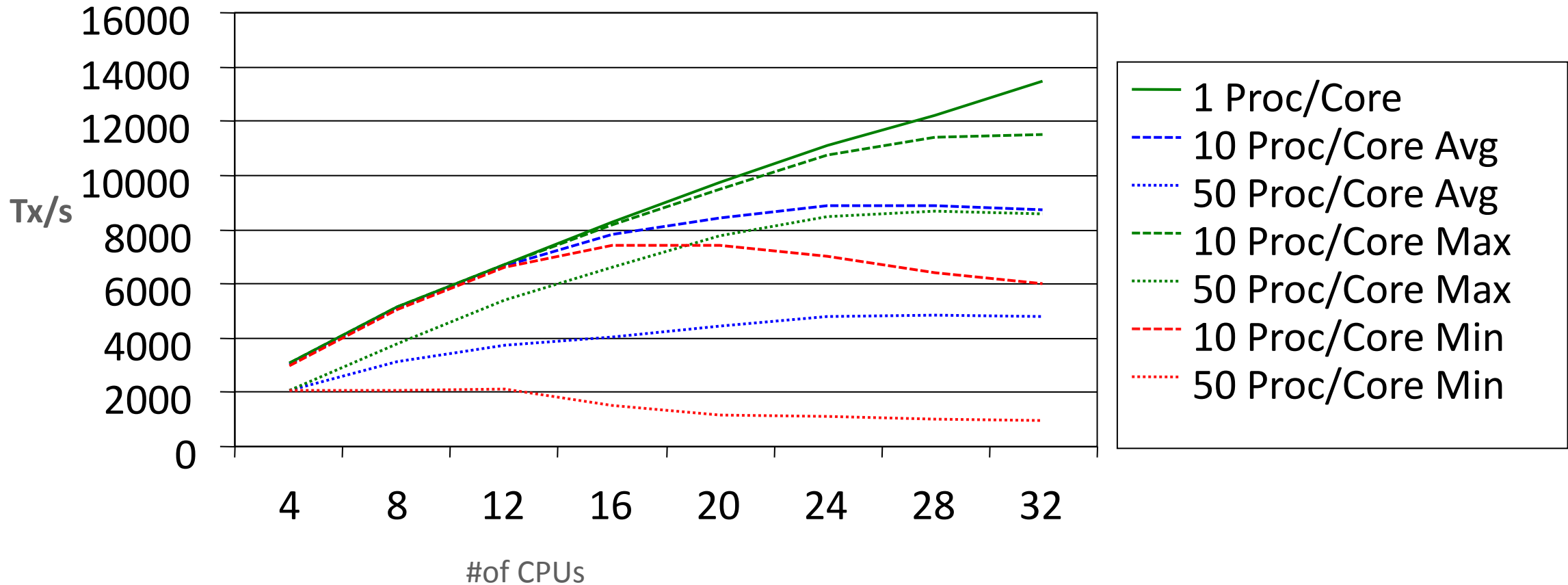
# Database Performance Core Principles

- To determine acceptable CPU utilization take a probabilistic approach to the subject.
  - If a CPU is 50% busy the chance of getting scheduled is 1 in 2
  - If a CPU is 66% busy the chance of getting scheduled is 1 in 3
  - If a CPU is 80% busy the chance of getting scheduled is 1 in 5
  - If a CPU is 90% busy the chance of getting scheduled is 1 in 10
- If the probabilities are used as indicator of the predictability of user response time, then the variance in user response time becomes noticeable at about 60-65%
- This has been observed in production and laboratory conditions for many years.



# Database Core Principles

## Impact of Too Many Processes



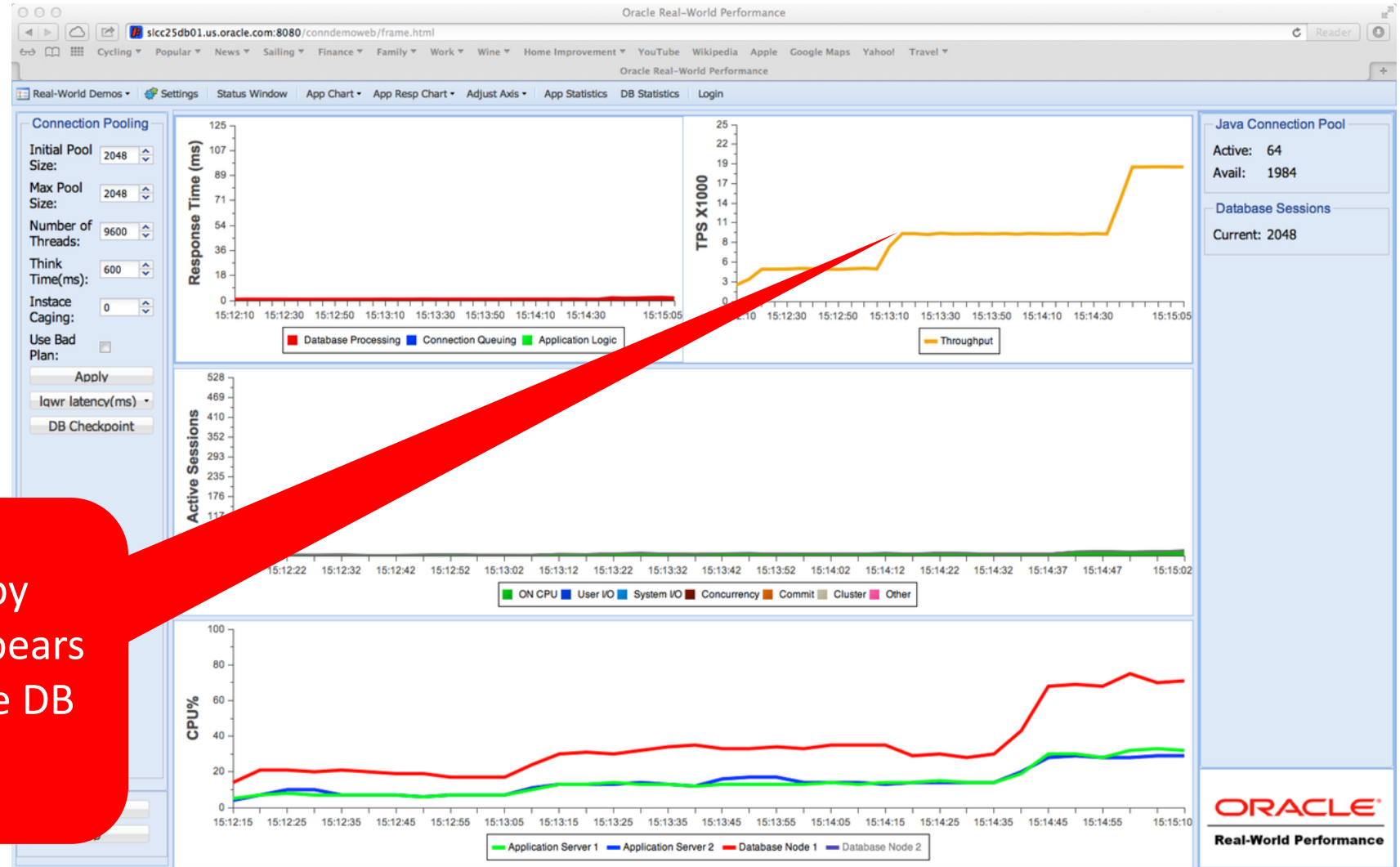
# Connection Pooling

ORACLE®  
RWP Video



# Connection Pools

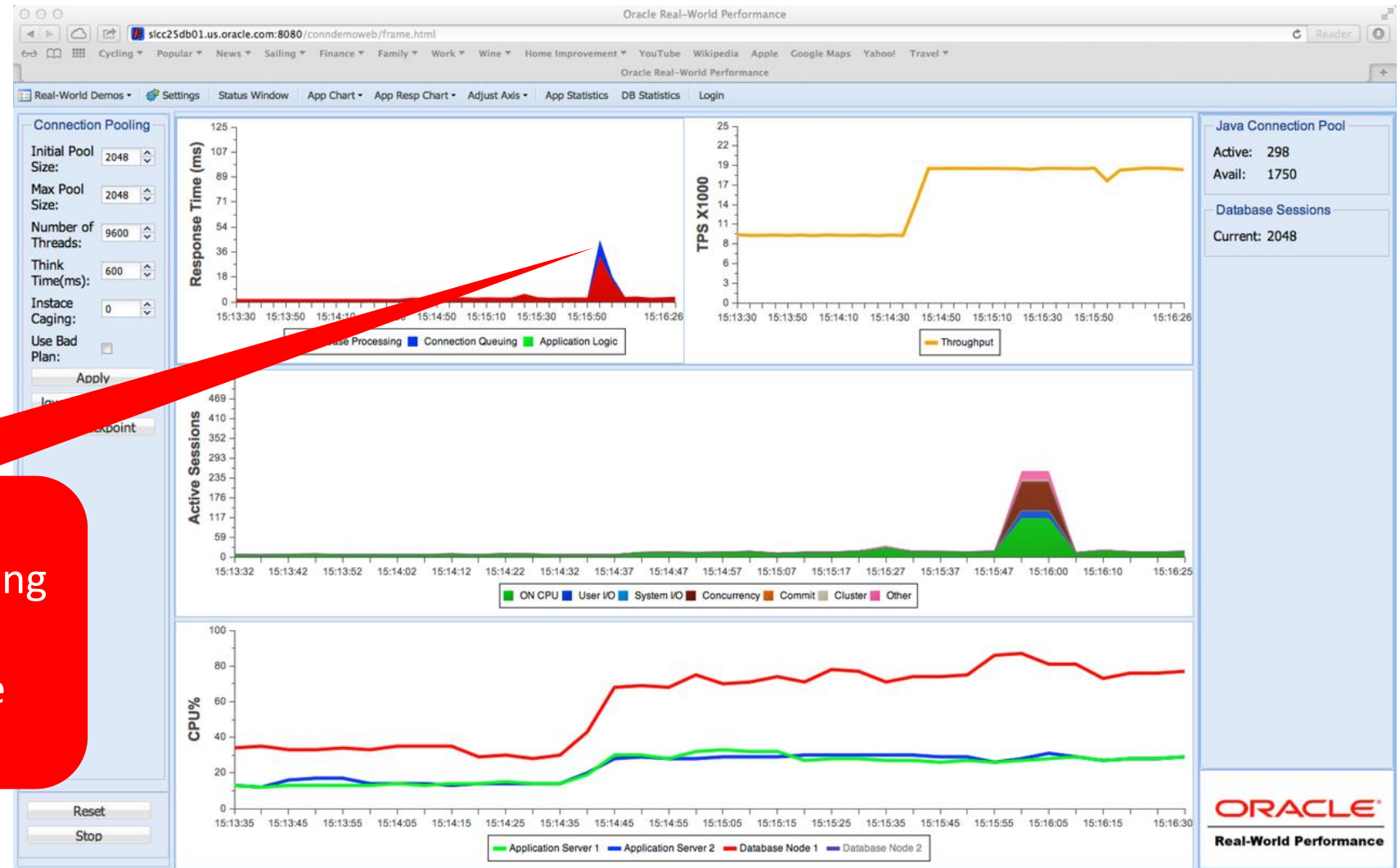
## Performance Data



The workload is increased by doubling the load. System appears scalable up to 60% CPU on the DB server.

# Connection Pools

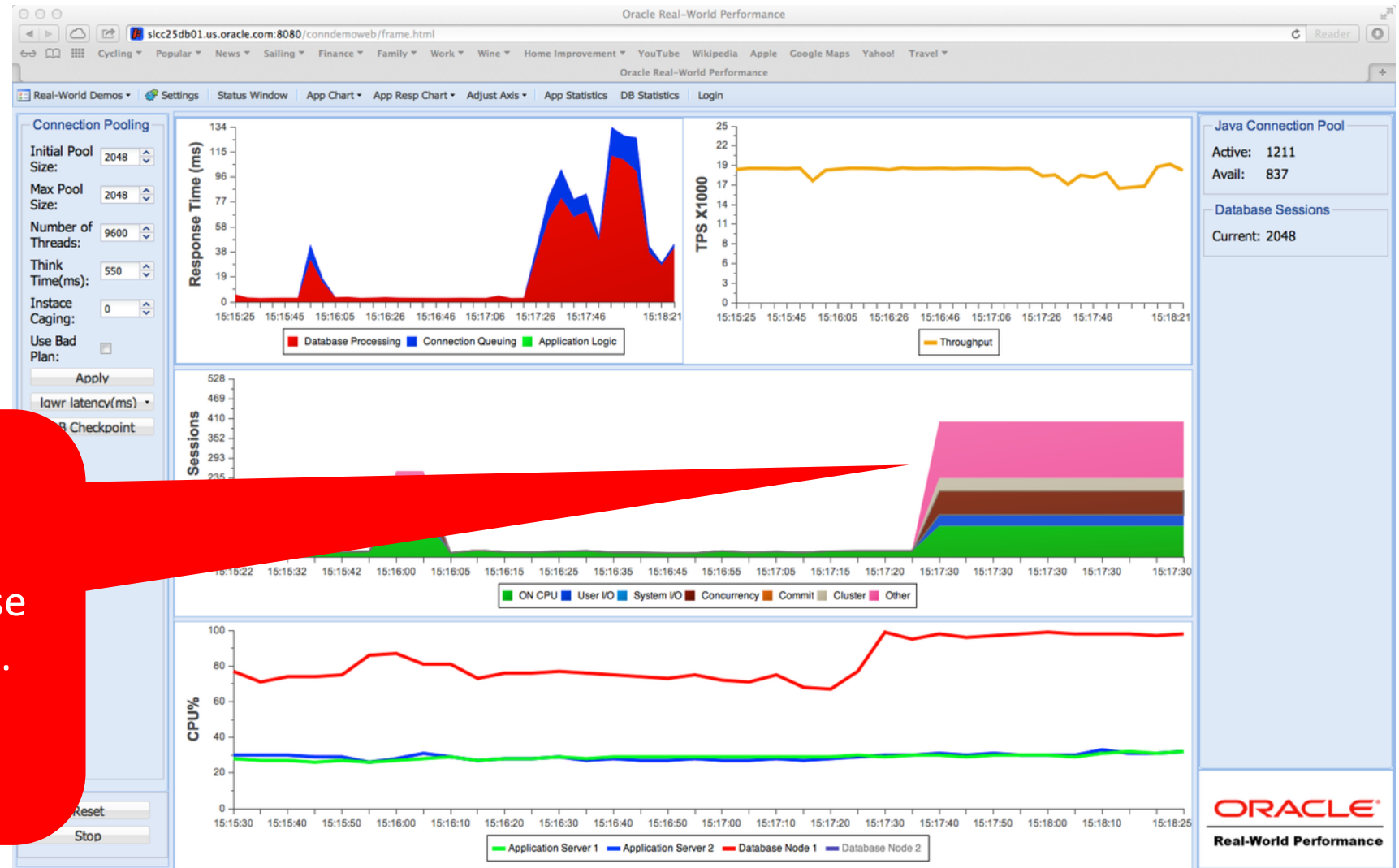
## Performance Data



A checkpoint is initiated, creating a CPU spike that results in unpredictable response time

# Connection Pools

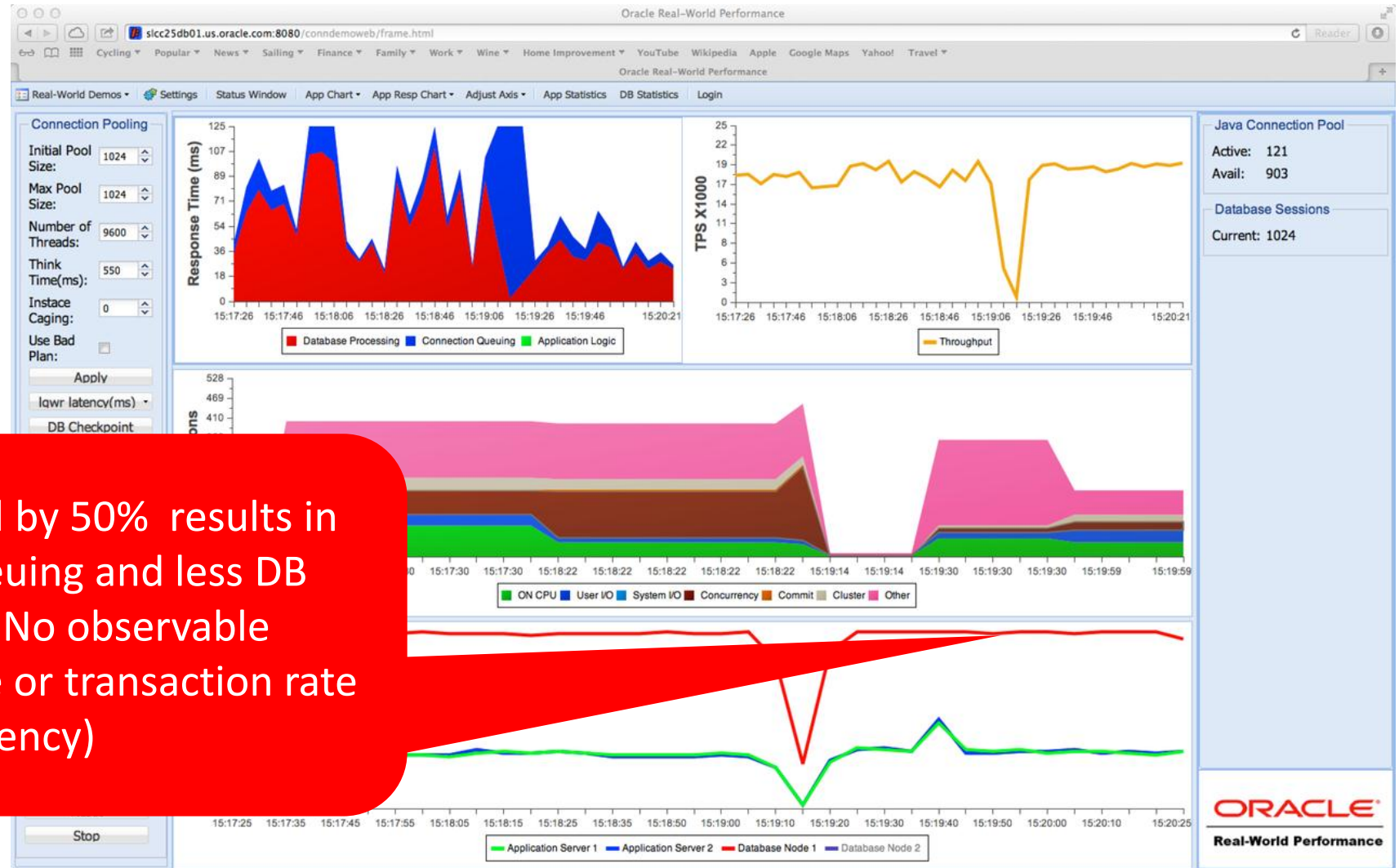
## Performance Data



A slight increase to the workload results in a disproportionate CPU increase and response time degrades. System monitoring tools become unreliable

# Connection Pools

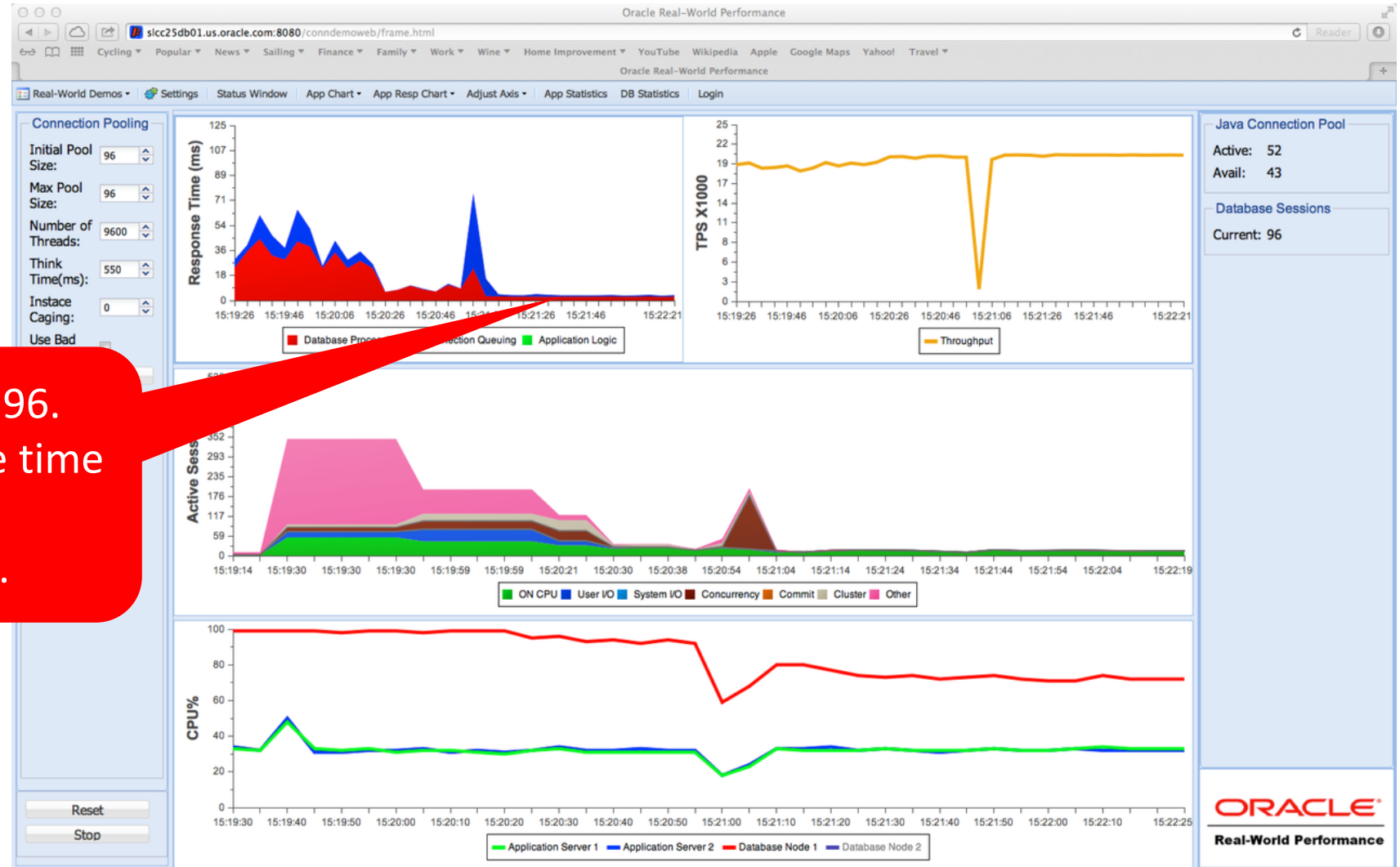
## Performance Data



Reducing the connection pool by 50% results in more application server queuing and less DB processes in a wait state. No observable improvement in response time or transaction rate (value or consistency)

# Connection Pools

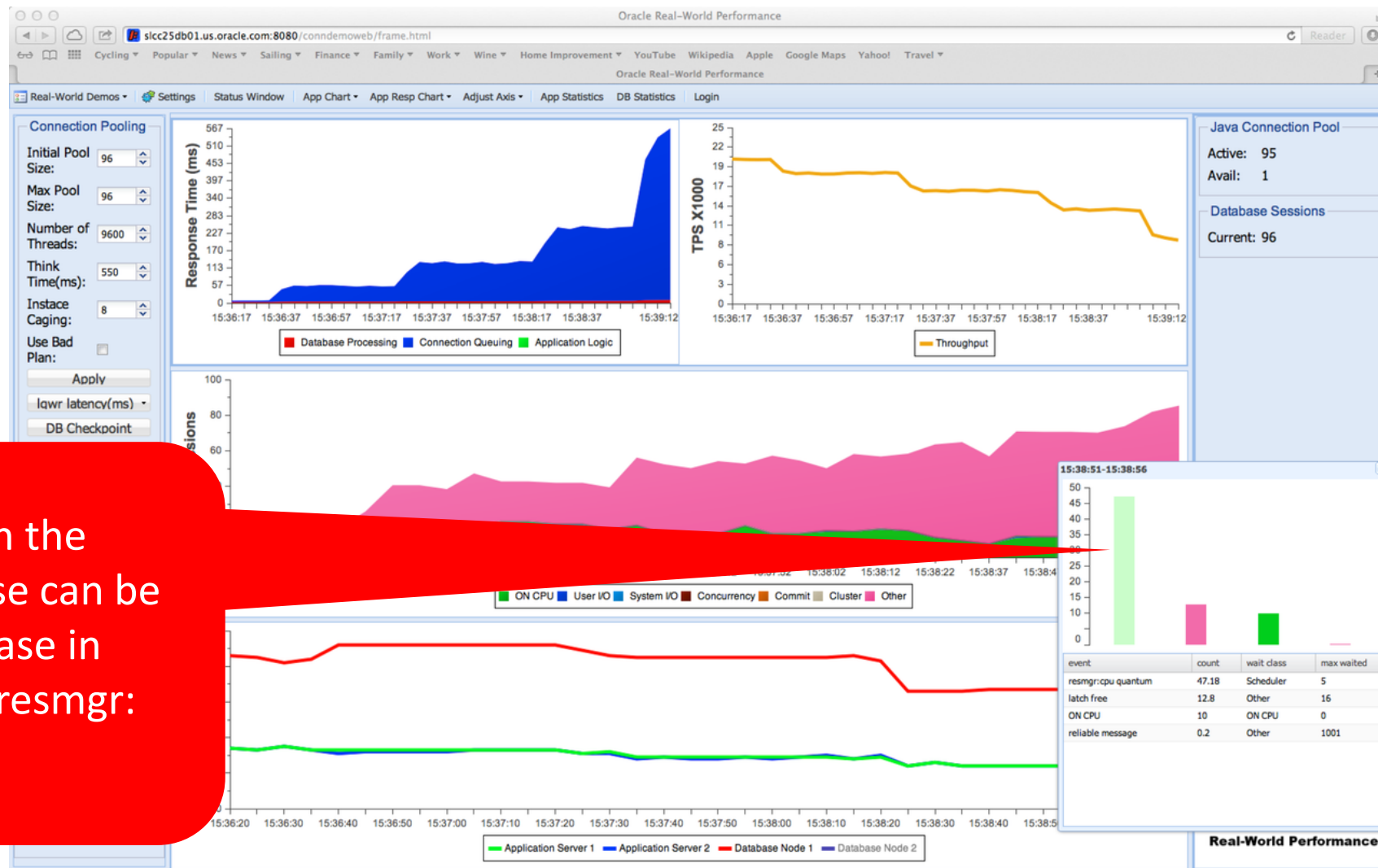
## Performance Data



Connection pool reduced to 96.  
Note improvement in response time  
and transaction rate.  
CPU utilization is reduced.

# Resource Management

## Performance Data



By reducing the CPU\_COUNT in the resource manager, the database can be throttled back. Note the increase in response time and wait event resmgr:cpu quantum



# Bad Performance

## Observations

- A problematic application has never met performance objectives
- Response time and throughput are poor
- Everybody blames the database
  - DBA sees no real issues
  - DBA suggests adding more connections to drive up workload
- The system must be able to execute a minimum of 35,000 transactions per second to survive Thanksgiving and Black Friday

# Parsing Demo



# Parsing Demo

## Bad Performance with Logons



Poor response time

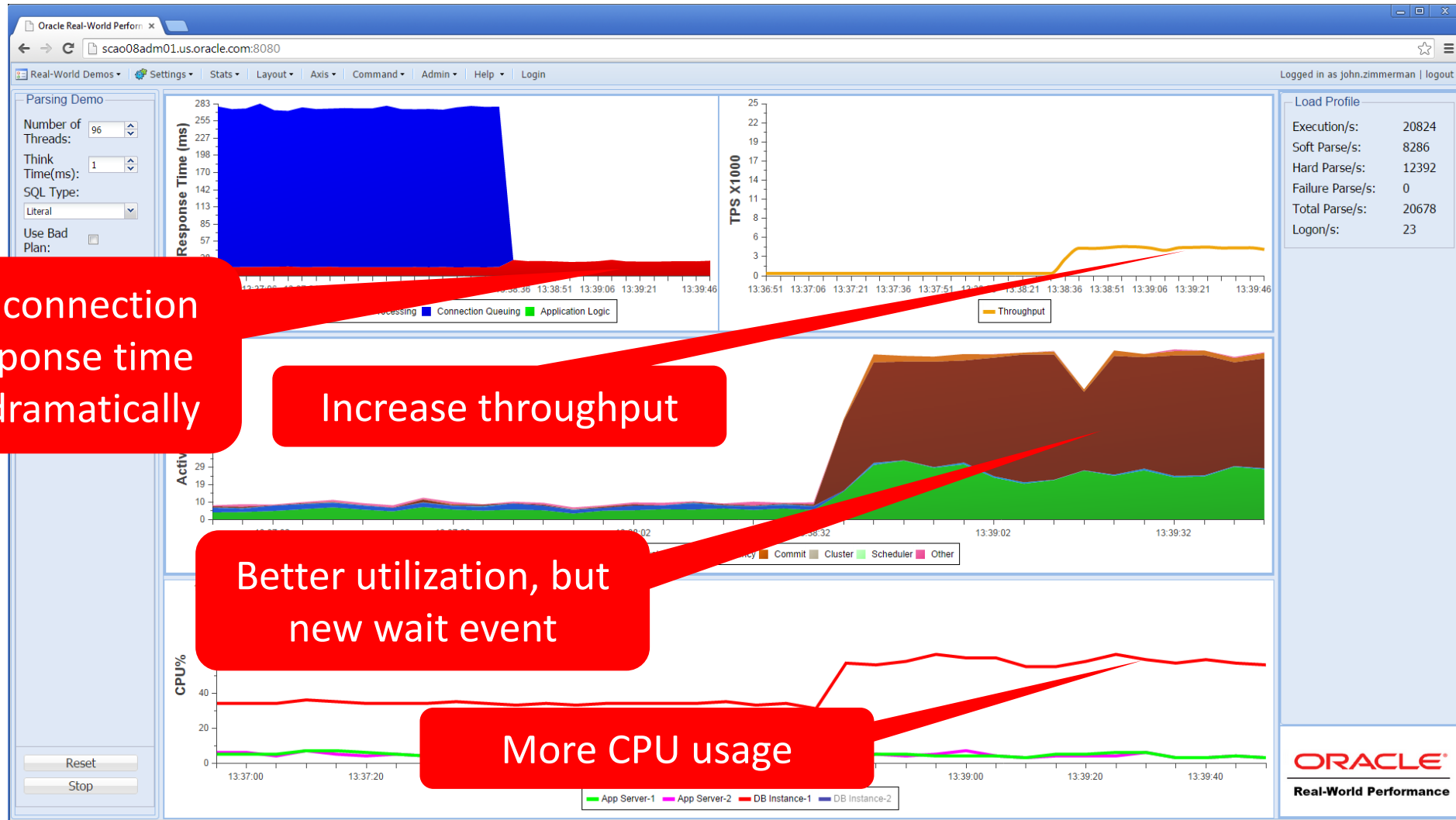
Low transaction rate

Cannot make the system busy

# Parsing Demo

## Connection Pools and Hard Parse

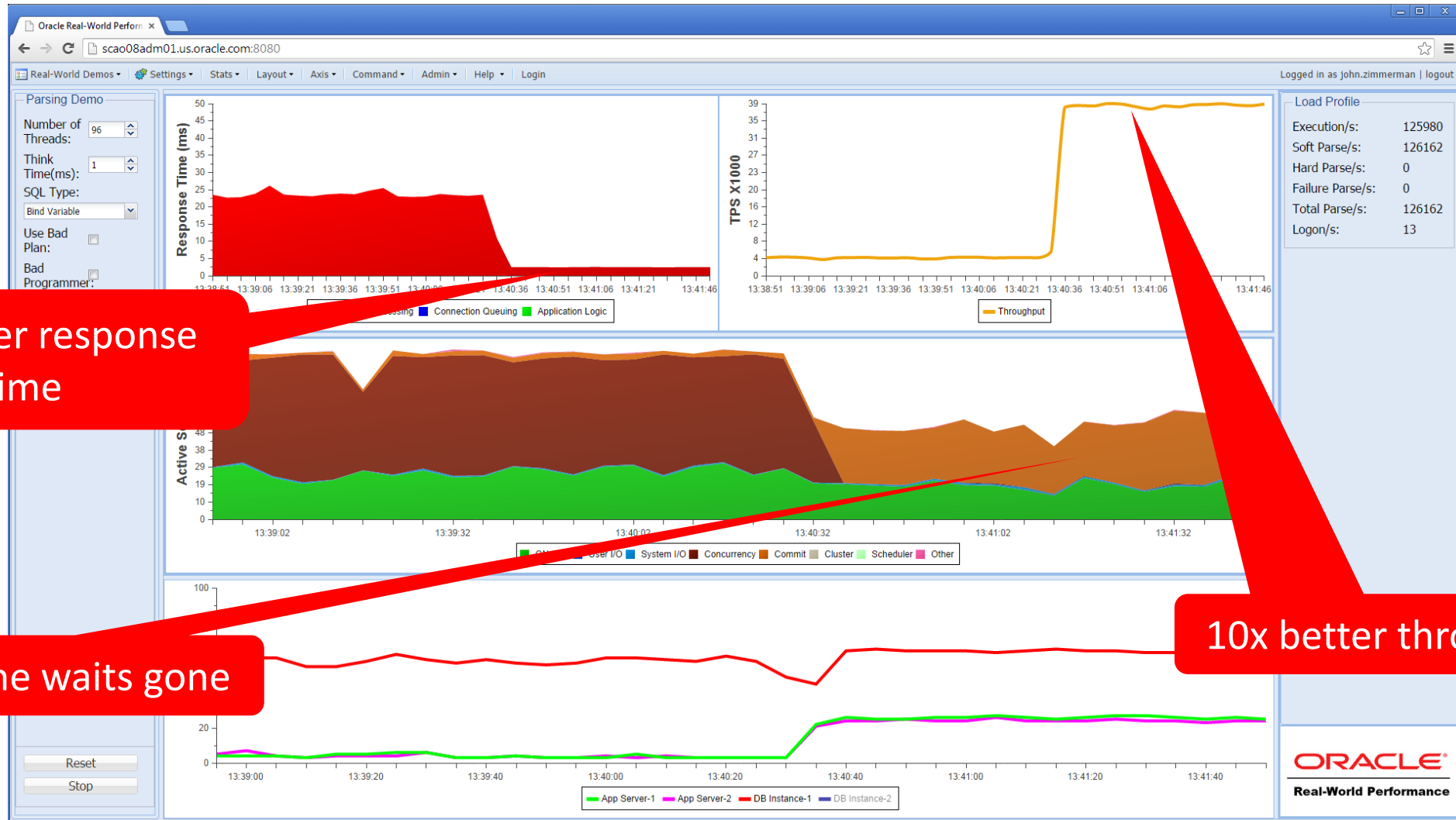
ORACLE®  
RWP Video



# Parsing Demo

## Bind Variables and Soft Parse

ORACLE®  
RWP Video



10x better response time

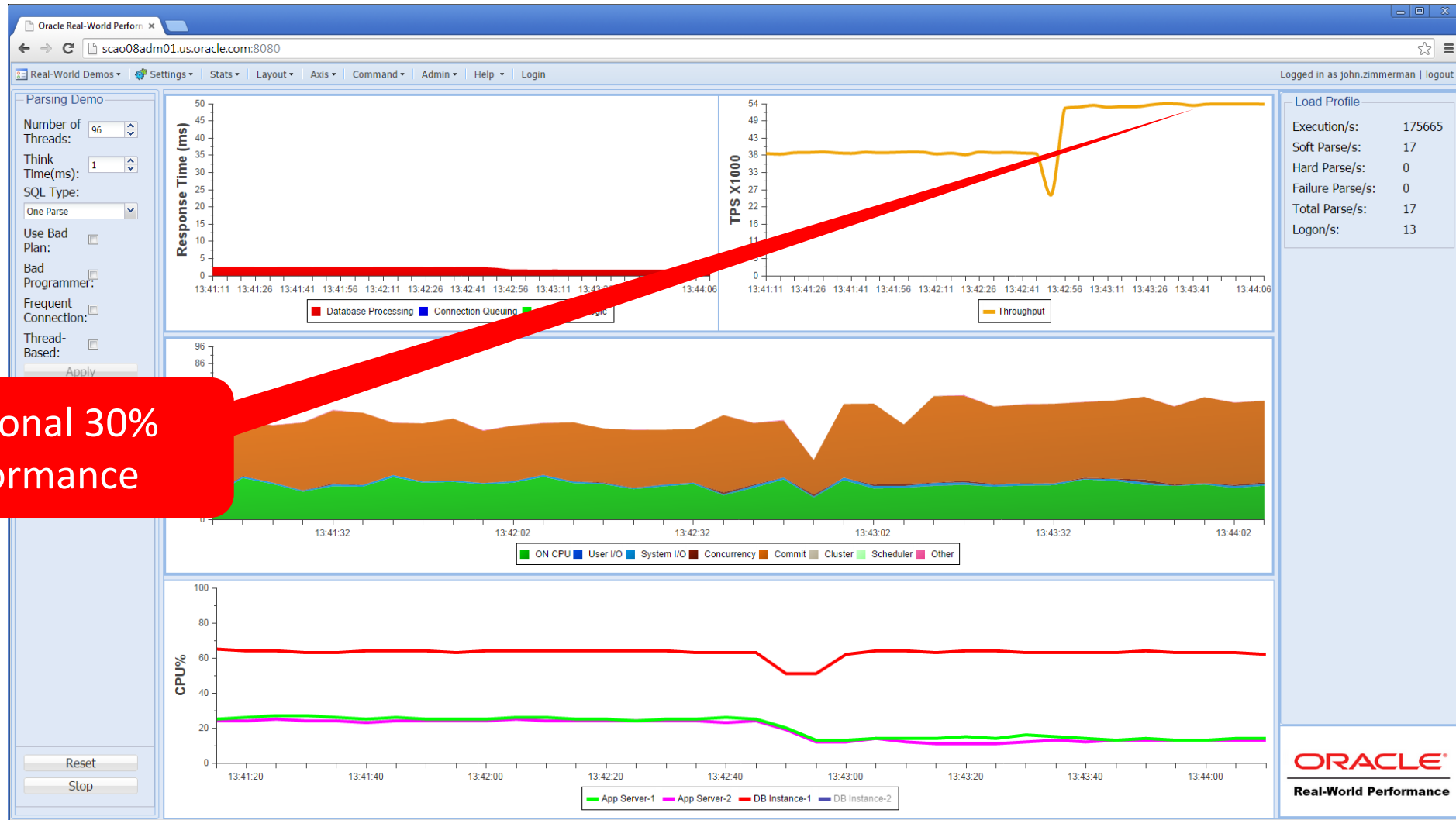
Row cache waits gone

10x better throughput

# Parsing Demo

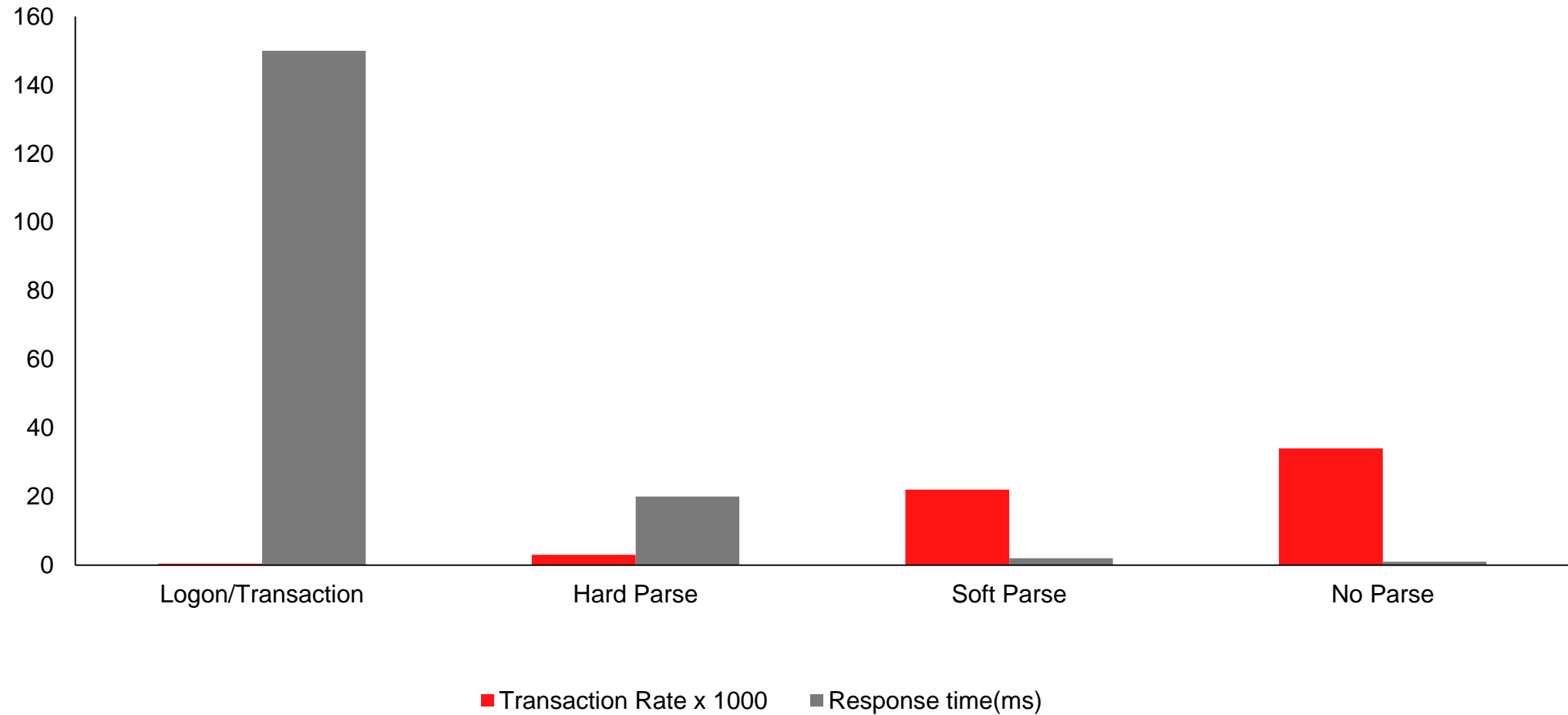
## Shared Cursors and One Parse

ORACLE®  
RWP Video



# Parsing Demo

## Incorrect Use of Sessions and Cursors



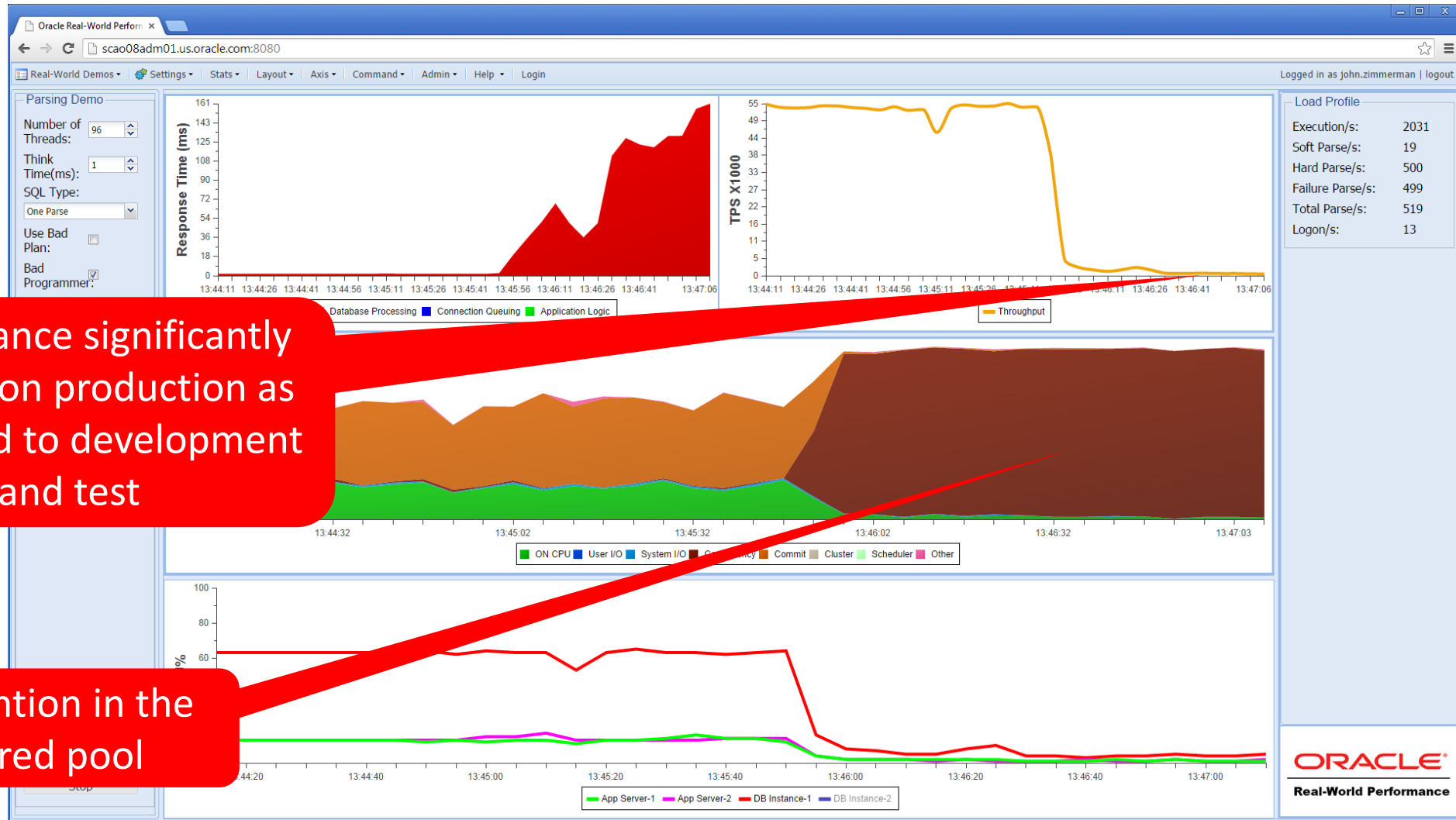
# Parsing Demo

## Observations

- Both the development and DBA teams are confused.
  - Performance in the development and test systems is as anticipated
  - Performance in production is nowhere near level of test system
  - DBAs see shared pool contention but developers have coded diligently to ensure no parsing
  - Development has confirmed the same code is running in both test and production



# Invalid SQL Performance Data



Performance significantly reduced on production as compared to development and test

Contention in the shared pool

# Invalid SQL

- A page refresh trigger attempts to set a session-level initialization parameter to enable a diagnostic patch that is not installed in production
  - This results in a failed parse
- All users are frequently attempting to parse the same SQL, sessions serialize within the shared pool
- How to find invalid SQL:
  - Look for parse count failures from v\$sqlstat
  - Check session traces for error messages
  - Look for SQL\*Net Break/Reset

Leaking

**ORACLE**  
RWP Video

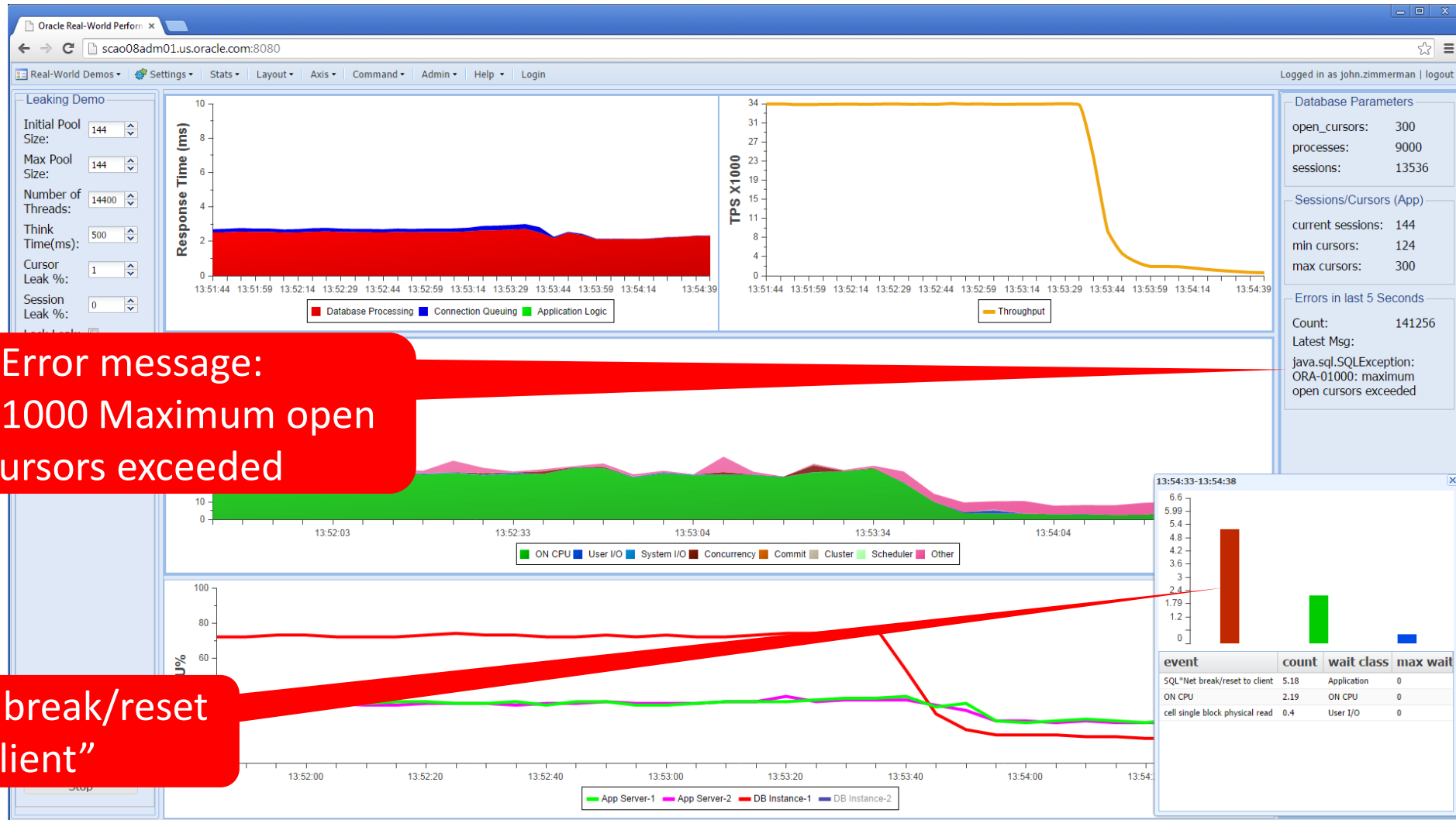


# Leaking

## Observations

- Intermittent error: “ORA-01000: Maximum number of cursors exceeded”. Application server fails and must be restarted
- The DBA has suggested that the init.ora parameter open\_cursors be reset to 30,000 to make the problem “go away for a while”.
- Symptoms of cursor leaking

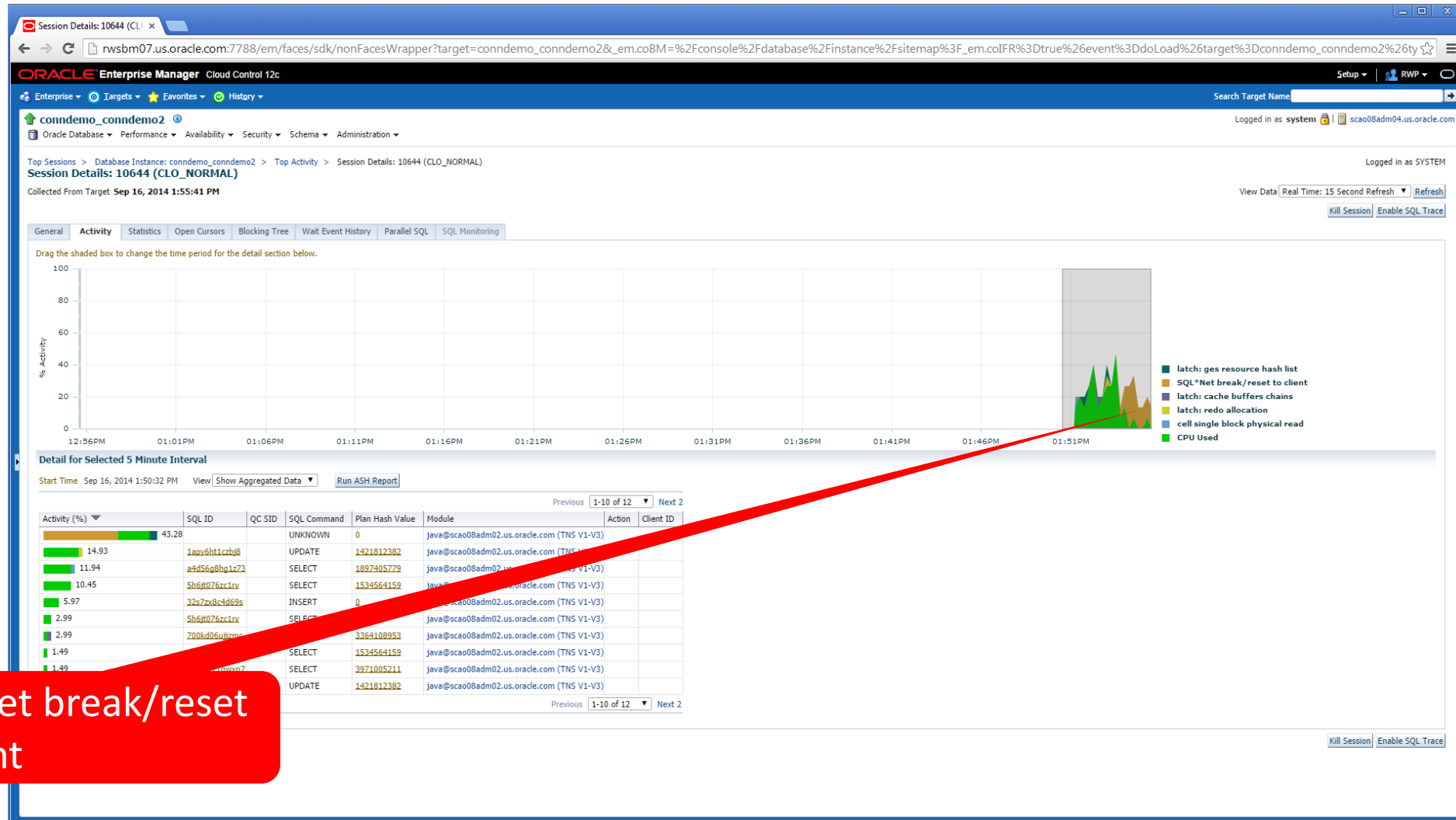
# Leaking Performance Data



Error message:  
ORA-01000 Maximum open cursors exceeded

“SQL\*Net break/reset to client”

# Leaking Session Details



SQL\*Net break/reset to client

# Leaking Cursor Data

Session Details: 10644 (CLO\_NORMAL)

Collected From Target Sep 16, 2014 1:56:36 PM

View Data Real Time: 15 Second Refresh Refresh

Kill Session Enable SQL Trace

SQL ID	SQL Text	Count
5h6j076zc1rv	SELECT DBNODEID,CARDID,BALANCE,PRIZETAX,PRIZENOTAX,ISCLIENTR	58
0u580ctbbsg4f	SELECT DBNODEID,CARDID,BALANCE,PRIZETAX,PRIZENOTAX,ISCLIENTR	9
2fm8g72fjwk09	INSERT INTO t_game_XYDC VALUES(:1 ,:2 , 12, 1234, 123456781	10
cb2rs70z99yt	SELECT DBNODEID,CARDID,BALANCE,PRIZETAX,PRIZENOTAX,ISCLIENTR	10
5wvwp7spyq2fh	select privilege#, bitand(nvl(options, 0), 8) from sysauth\$	1
7f3hgy7qb86g	UPDATE t_game_XYDC_detail set BONUSPRIZE = 12345678 WHERE s	12
bkq9pjcfm9vn	select /*+ connect_by_filtering */ privilege#, bitand(nvl(op	1
865qwpdcyggkk	select spare6 from user\$ where user#=:1	1
0zs820bmk4rrv	UPDATE T_SESSION SET PRIZENOTAX =654321, ISCLIENTRECEIVED	15
32s7zv8c4d69s	INSERT INTO t_game_XYDC_detail VALUES(:1 ,:2 ,:3 ,'HallID',	72
f0h5pzmhju11	select SYS_CONTEXT('USERENV', 'SERVER_HOST'), SYS_CONTEXT('U	1
700kd06ujzmc	UPDATE t_game_XYDC SET FinishBonus = 2 WHERE serial=:1 an	17
9tg4g8y4rvy8	select type#,blocks,extents,minexts,maxexts,extsize,extpct,u	1
4xujvn21qwxn7	SELECT BonusCount,LineCost FROM T_game_XYDC WHERE serial=:	13
1apy6ht1czbj8	UPDATE T_SESSION SET WFDETAILID = 1,PRIZENOTAX = 100,LAST	64
bvkckyaa5hvqx	select decode(upper(fallover_method), NULL, 0, 'BASIC', 1,	1
a4d5g8hg1z73	SELECT seatName from clo_normal.seat where id = :1 FOR UPD	1
4kdu2v3s2uy	INSERT INTO t_game_XYDCbonus_detail VALUES(:1 ,:2 ,:3 ,:4 ,:5	11
9zq9qd9bm4spu	update user\$ set spare6=DECODE(to_char(sysdate,'DDMMYY'),	1
095ydz7c3nu7q	UPDATE T_SESSION SET PRIZENOTAX = 100, ISCLIENTRECEIVED	19

Kill Session Enable SQL Trace

Cursor list with Count > 1  
implies "leaked" cursors

# Leaking

## Observations

- After a period of time, the system performance begins to decline and then degrades rapidly
- After rapid degradation, the application servers time out and the system is unavailable
- The DBA claims the database is not the problem and simply needs more connections
  - The init.ora parameter processes is increased to 20,000



# Leaking Performance Data



Load diminishes to zero

# Leaking

## Session Leaking

- Due to coding errors on exception handling, the application leaks connections in the connection pool making them programmatically impossible to use
- This reduces the effective size of the connection pool
- The remaining connections are unable to keep up with the incoming workload
- The rate of connection leakage is accelerated until there are no useable connections left in the pool

# Leaking

## Session Leaking

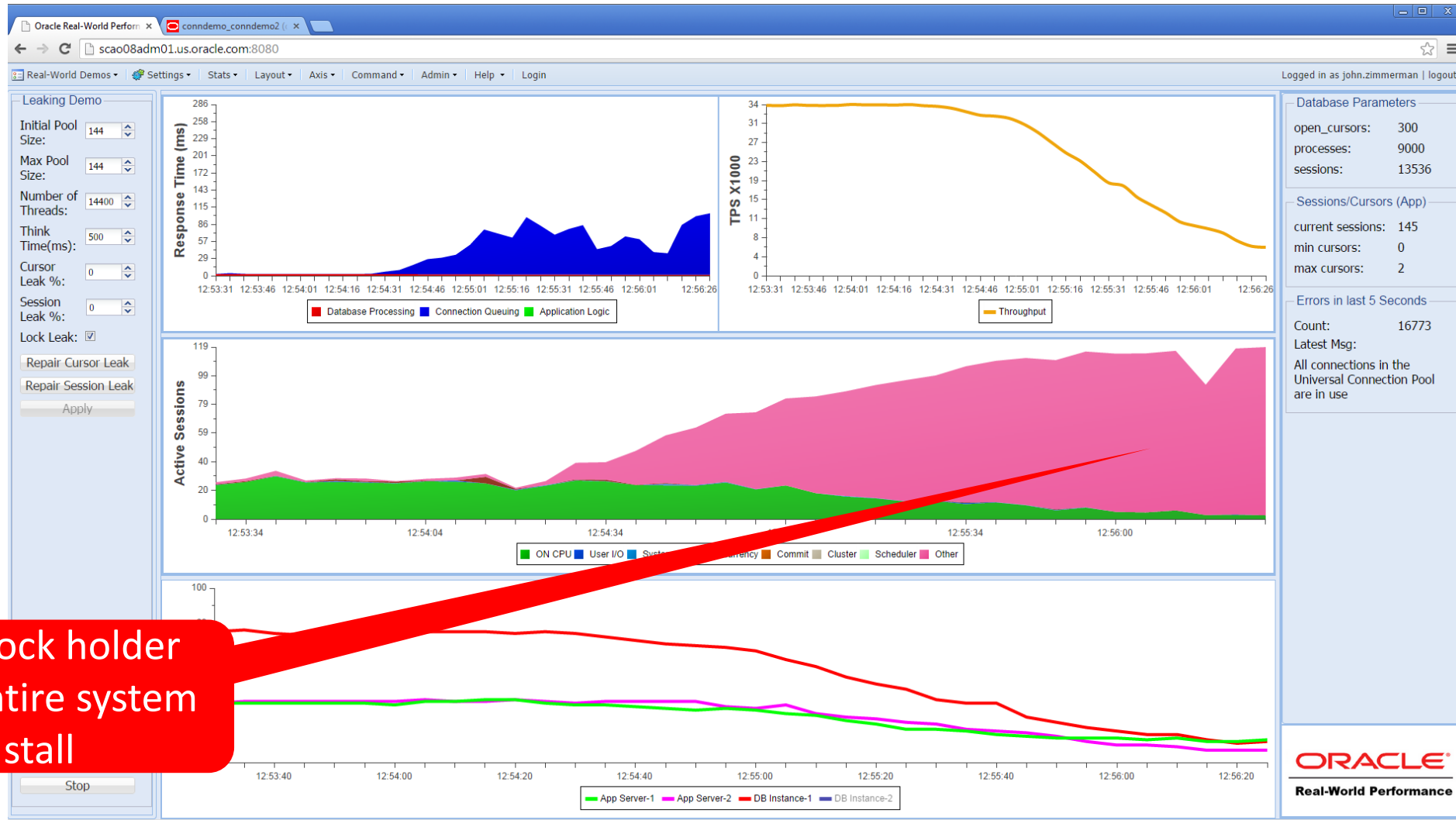
- Potential indicators of session leaking:
  - Frequent application server resets
  - init.ora parameters process and sessions set very high
  - Configuration of large and dynamic connection pools
  - Large number of idle connections connected to the database
  - Free memory on database server continually reduced
  - Presence of idle connection kill scripts or middleware configured to kill idle sessions

# Leaking

## Observations

- Without warning, the database appears to hang and the application servers time out simultaneously
- The DBA sees that all connections are waiting on a single lock held by a process that has not been active for a while.
- Each time the problem occurs, the DBA responds by running a script to kill sessions held by long time lock holders and allowing the system to restart.

# Leaking Performance Data



Leaked lock holder causes entire system to stall

# Leaking Blocking Tree

Oracle Real-World Performance | Instance Locks: conndemo2

Oracle Enterprise Manager Cloud Control 12c

conndemo\_conndemo2

Cluster Database Locks > Database Instance: conndemo\_conndemo2 > Instance Locks: conndemo2

View: Blocking Locks

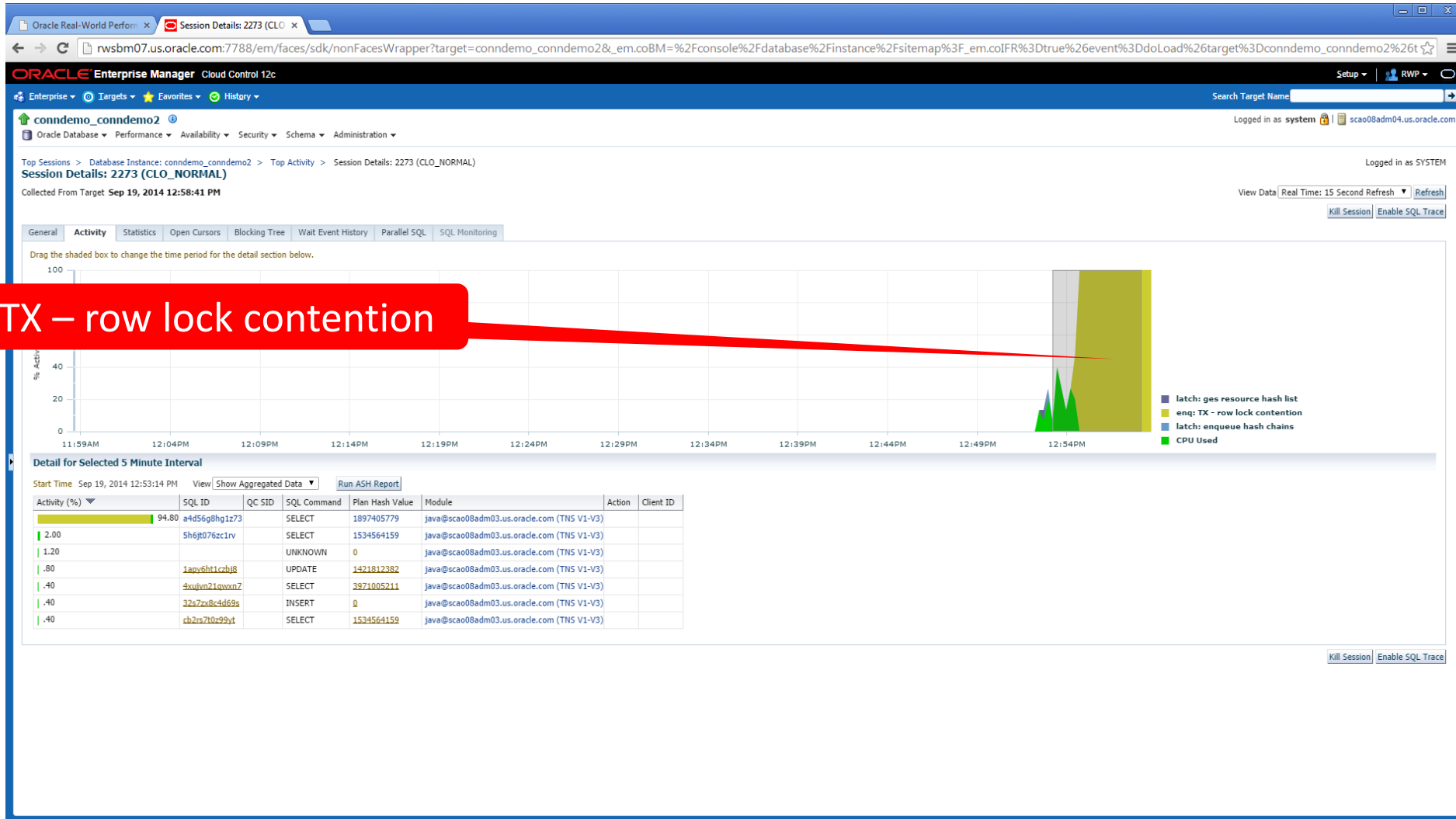
Kill Session | Session Details | View Object

Expand All | Collapse All

Select	Username	Sessions Blocked	Session ID	Serial Number	Process ID	SQL ID	Lock Type	Mode Held	Mode Requested	Object Type	Object Owner	Object Name	ROWID	Time in current mode (seconds)
⊖	Blocking Locks	123	12438	17500	126737	bf8837wngfz1b	TX	EXCLUSIVE	NONE	TABLE	CLO_NORMAL	SEAT		171
⊖	CLO_NORMAL1	0	5	25997	120163	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.00DA.0400	146
⊖	CLO_NORMAL	0	8	52226	120120	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.00D9.0400	151
⊖	CLO_NORMAL	0	11	20145	120191	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0145.0400	115
⊖	CLO_NORMAL	0	12	26227	120307	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.004A.0400	127
⊖	CLO_NORMAL	0	16	35556	120044	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0167.0400	147
⊖	CLO_NORMAL	0	17	20603	120335	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083063.0050.0400	101
⊖	CLO_NORMAL	0	18	5638	120092	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083063.0042.0400	105
⊖	CLO_NORMAL	0	19	22617	120010	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083063.000C.0400	149
⊖	CLO_NORMAL	0	20	63548	120068	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0177.0400	140
⊖	CLO_NORMAL	0	22	57800	120249	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.006B.0400	100
⊖	CLO_NORMAL	0	1134	41301	120046	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083063.0053.0400	153
⊖	CLO_NORMAL	0	1135	7335	120193	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.00B6.0400	97
⊖	CLO_NORMAL	0	1137	17863	120283	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0052.0400	127
⊖	CLO_NORMAL	0	1140	19749	120012	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0172.0400	112
⊖	CLO_NORMAL	0	1141	16568	120251	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.00DC.0400	143
⊖	CLO_NORMAL	0	1142	37518	120229	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.00F1.0400	76
⊖	CLO_NORMAL	0	1143	39631	120070	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.016C.0400	82
⊖	CLO_NORMAL	0	1144	60299	120311	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0005.0400	156
⊖	CLO_NORMAL	0	1145	3124	120094	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0037.0400	100
⊖	CLO_NORMAL	0	1146	17253	120337	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083063.002A.0400	140
⊖	CLO_NORMAL	0	1147	44291	120165	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0005.0400	141
⊖	CLO_NORMAL	0	2265	32530	120048	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.00F6.0400	155
⊖	CLO_NORMAL	0	2267	54863	120096	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0104.0400	87
⊖	CLO_NORMAL	0	2268	17878	120124	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.008F.0400	148
⊖	CLO_NORMAL	0	2269	57688	120339	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.019B.0400	137
⊖	CLO_NORMAL	0	2271	12212	120167	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.00C1.0400	137
⊖	CLO_NORMAL	0	2273	30681	120227	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.005C.0400	170
⊖	CLO_NORMAL	0	2274	58213	120014	a4d56g8hg1z73	TX	NONE	EXCLUSIVE	TABLE	CLO_NORMAL	SEAT	00083067.0152.0400	82

Block tree reveals  
lock holder which  
can be killed

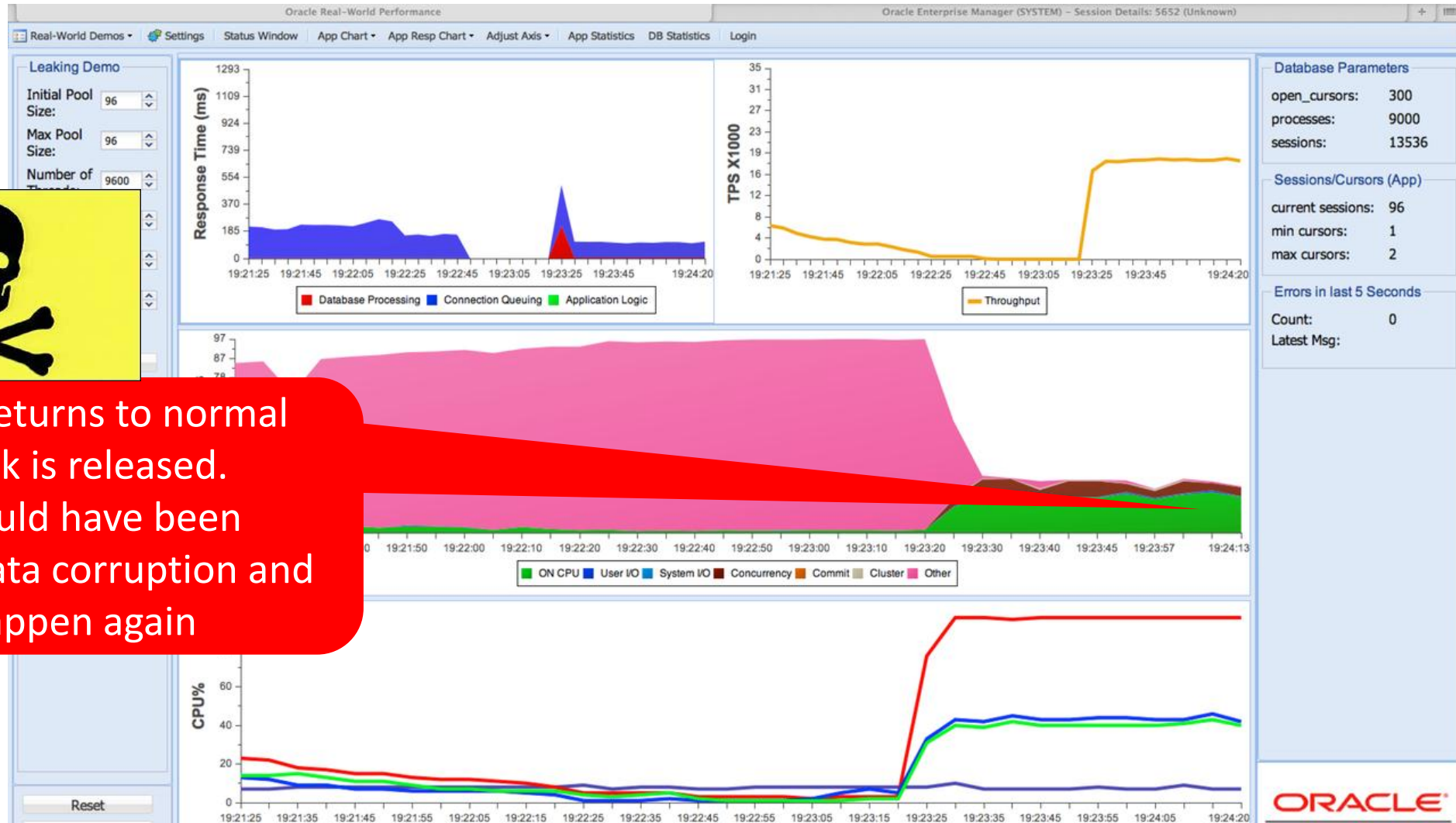
# Leaking Performance Data



# Leaking Performance Data



System returns to normal when lock is released. There could have been logical data corruption and it may happen again





# Leaking

## Lock Leaking

- Lock leaking is usually a side effect of session leaking and the exception handling code failing to execute a commit or rollback in the exception handling process.
- A leaked session may be programmatically lost to the connection while holding locks and uncommitted changes to the database.

# Leaking

## Lock Leaking

- Programming error impact:
  - Potential system hangs: all connections queue up for the held lock
  - Potential database logical corruptions: end users may have thought transactions were committed when in fact they have not been
  - If sessions return to the connection pool but still have uncommitted changes, it is not deterministic, if and/or when the changes are committed or rolled back. This is a serious data integrity issue.

# Leaking

## How to Develop High Performance Applications

- Developer Bugs
  - Incorrect/untested exception handling
    - Cursor, session and lock leaking
      - High values for init.ora ( open\_cursors, processes, sessions )
      - Idle process and lock holder kill scripts
      - Oversized connection pools of largely idle processes

# AWR Architecture Analysis

## More than just wait events and top SQL

- Large amount of data in the AWR report
- Tells us about the way that the system has been architected and designed as well as about how it is performing
- Often see common mistakes

# AWR from online system

Ready for Black Friday?

# AWR from Online system

- Testing system for Black Friday readiness
- Cannot generate load expected on test system
- Do you see any problems with this system scaling up from this test?
- Will we survive Black Friday ?

# AWR Header

## WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
SHOPPING	1722515684	SHOPPING02	2	07-Oct-13 16:10	11.2.0.2.0	YES

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
hlq11db.na.bigretail.com	Linux x86 64-bit	64	32	4	504.03

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	3037	15-Oct-13 02:45:11	3351	8.2
End Snap:	3044	15-Oct-13 04:30:00	4635	26.2
Elapsed:		104.82 (mins)		
DB Time:		27,641.09 (mins)		

## Report Summary

### Cache Sizes

	Begin	End		
Buffer Cache:	30,720M	30,720M	Std Block Size:	16K
Shared Pool Size:	16,384M	16,384M	Log Buffer:	14,772K

### Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	263.7	0.2	0.00	0.00
DB CPU(s):	39.0	0.0	0.00	0.00
Redo size:	9,829,605.5	7,222.2		
Logical reads:	802,177.1	589.4		

- 32 Cores available

- Over processed **ORACLE®  
RWP Video**

– Sessions is 100x cores

– Session count growing

- Session leak
- Dynamic connection pools

- Cursors per session growing **ORACLE®  
RWP Video**

– Cursor leakage

# Load Profile

## Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	263.7	0.2	0.00	0.00
DB CPU(s):	39.0	0.0	0.00	0.00
Redo size:	9,829,605.5	7,222.2		
Logical reads:	802,177.1	589.4		
Block changes:	44,854.5	33.0		
Physical reads:	10,749.2	7.9		
Physical writes:	2,085.7	1.5		
User calls:	104,774.4	77.0		
Parses:	39,775.4	29.2		
Hard parses:	0.1	0.0		
W/A MB processed:	96.0	0.1		
Logons:	10.5	0.0		
Executes:	59,560.7	43.8		
Rollbacks:	878.3	0.7		
Transactions:	1,361.0			

- ~260 sessions active on average
- ~40 on CPU
  - Only have 32 cores
  - System CPU limited
- 10 logons per second
  - In a stable system?
    - Session leaks
    - Dynamic connection pools
- 40% of user txns are rollbacks
  - Coding for failure



# Init.ora

## init.ora Parameters

Parameter Name	Begin value	End value (if different)
_bloom_filter_enabled	FALSE	
_disable_image_check	TRUE	
_fix_control	6239971:off	
_kghdsidx_count	1	
_kgl_debug	hash='69cd9f156d9ccd9b245ebaccc65c558' debug=33554432	
_kgl_hot_object_copies	16	
_memory_imm_mode_without_autosga	FALSE	
_shared_pool_reserved_min_alloc	6000	
aq_tm_processes	4	
audit_file_dest	/u01/app/oracle/admin/SHOPPING02/adump	
audit_sys_operations	TRUE	
audit_trail	DB	
cluster_database	TRUE	
cluster_database_instances	2	
compatible	11.2.0.2.0	
control_file_record_keep_time	35	
control_files	+SHOPPING_DATA01/SHOPPING/controlfile/current.762.823865529,+SHOPPING_RECO01/SHOPPING/controlfile/current.375.823865531	
core_dump_dest	/u01/app/oracle/admin/SHOPPING02/cdump	
cursor_sharing	FORCE	
db_block_size	16384	
db_cache_size	32212254720	
db_create_file_dest	+SHOPPING_DATA01	
db_domain	world	
db_file_multiblock_read_count	32	
db_files	4096	
db_name	SHOPPING	
db_recovery_file_dest	+SHOPPING_RECO01	

- Underscore parameters
- Db\_block\_size=16384
- Cursor\_sharing=FORCE
- Db\_file\_multiblock\_read\_count=32

# Init.ora

db_file_multiblock_read_count	32
db_files	4096
db_name	SHOPPING
db_recovery_file_dest	+SHOPPING_RECO01
db_recovery_file_dest_size	64424509440
db_recycle_cache_size	4429185024
db_writer_processes	12
diagnostic_dest	/u01/app/oracle
disk_asynch_io	TRUE
fail_client	SHOPPING
fail_server	op99eodb01_linux
fast_start_mttr_target	3600
instance_name	SHOPPING02
instance_number	2
java_pool_size	134217728
job_queue_processes	20
large_pool_size	2147483648
local_listener	SHOPPING02_local
log_archive_config	dg_config=(SHOPPING, op99eodb)
log_archive_dest_1	location=+SHOPPING_RECO01
log_archive_dest_2	
log_archive_dest_state_2	DEFER
log_archive_max_processes	10
log_archive_min_succeed_dest	1
log_buffer	3407872
nls_date_format	DD-MON-RR
open_cursors	2000
open_links	255
optimizer_capture_sql_plan_baselines	TRUE
optimizer_dynamic_sampling	1
optimizer_index_cost_adj	50
optimizer_mode	ALL_ROWS
optimizer_secure_view_merging	FALSE

- Db\_writer\_processes=12
  - On a system that supports asynchIO?
- Open\_cursors=2000
  - Per session limit
  - Implies cursor leaking

ORACLE®  
RWP Video

# Init.ora

open_cursors	2000
open_links	255
optimizer_capture_sql_plan_baselines	TRUE
optimizer_dynamic_sampling	1
optimizer_index_cost_adj	50
optimizer_mode	ALL_ROWS
optimizer_secure_view_merging	FALSE
os_authent_prefix	
parallel_adaptive_multi_user	FALSE
parallel_max_servers	180
parallel_min_servers	8
pga_aggregate_target	16106127360
processes	5500
recovery_parallelism	90
remote_listener	qlocdb17:50000
remote_login_passwordfile	EXCLUSIVE
resource_manager_plan	
sec_case_sensitive_logon	FALSE
service_names	SHOPPINGsvc, SHOPPING02svc
session_cached_cursors	200
session_max_open_files	20
sessions	8320
sga_max_size	56505663488
shared_pool_reserved_size	262144000
shared_pool_size	17179869184
spfile	+SHOPPING_DATA01/SHOPPING/spfile
sql_trace	FALSE
star_transformation_enabled	false
streams_pool_size	134217728
thread	2
timed_statistics	TRUE
trace_enabled	TRUE
undo_management	AUTO
undo_retention	18000
undo_tablespace	undo02
workarea_size_policy	AUTO

- Optimizer\_index\_cost\_adj=50  
– Classic hack parameter
- Processes=5500
- Sessions=8320

# Top events

Where is the time going?

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
library cache: mutex X	51,206,131	377,179	7	22.74	Concurrency
enq: TX - row lock contention	259,719	298,891	1151	18.02	Application
db file sequential read	40,457,009	282,531	7	17.04	User I/O
latch: row cache objects	141,091	247,016	1751	14.89	Concurrency
DB CPU		245,536		14.81	

- Concurrency waits > 35% of time
  - Library cache: mutex X
  - Latch:row cache objects
  - Typical of high CPU load
  - A symptom, not the problem
- Row lock contention 18% of time
- IO with 7ms avg read time
- CPU only 15% of DB Time
- Log file sync?

# Top SQL

## Where is the time going?

### SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User IO Time as a percentage of Elapsed Time
- Captured SQL account for 43.8% of Total DB Time (s): 1,658,465
- Captured PL/SQL account for 13.9% of Total DB Time (s): 1,658,465

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
204,008.97	1,967,640	0.10	12.30	0.59	2.83	ccfrfn4vuh0	ScheduleReturn	SELECT /*SHOP*/ YFS_ORDER_HE...
116,892.49	26,600,256	0.00	7.05	6.12	0.00	4xrbd7rw15at7	INT&CAP_AgentServer	BEGIN DBMS_APPLICATION_INFO.SE...
110,919.39	28,684,565	0.00	6.69	6.30	0.00	bmx990q2tsqib	JDBC Thin Client	BEGIN DBMS_APPLICATION_INFO.SE...
79,569.63	13,172,207	0.01	4.80	8.40	0.00	8pvr9fks6m1r9	ADJUSTEOMSINV	SELECT /*SHOP*/ TO_CHAR(sysd...
57,474.41	108,429	0.53	3.47	0.17	0.05	ff72kz03ydv4x	Sourcing	SELECT /*SHOP*/ YFS_ORDER_HE...
20,853.07	53,990	0.39	1.26	0.17	0.00	ff10q31ahqbb5	CONFIRMFPSHIPMENT	SELECT /*SHOP*/ YFS_ORDER_HE...
19,865.13	1,580,843	0.01	1.20	6.48	0.00	2tq3dknsjum6d	JDBC Thin Client	SELECT /*SYS_B_0*/ FROM DUAL...
19,820.14	93,037	0.21	1.20	0.16	0.01	4htst3fakwa0m	CONSOLIDATE_ADDNL_INV	SELECT /*SHOP*/ YFS_INVENTOR...
19,064.37	19	1,003.39	1.15	4.03	97.93	d7ufw6t6n4grn	JDBC Thin Client	SELECT /*SHOP*/ count(/*SYS_...
16,954.77	2,418,713	0.01	1.02	8.52	91.24	fqq27u174x4tt	RETEK_ORDER_CREATE	SELECT /*SHOP*/ YFS_ORDER_LI...

### Top statement

```
SELECT /*SHOP*/ YFS_ORDER_HEADER.*
FROM YFS_ORDER_HEADER
WHERE (ORDER_HEADER_KEY = :1 )
FOR UPDATE
```

- 12% of load
- 2 million executions
- Average execution 0.1 sec

# Top SQL

## Segments by Row Lock Waits

- % of Capture shows % of row lock waits for each top segment compared
- with total row lock waits for all segments captured by the Snapshot

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Row Lock Waits	% of Capture
SHOP	SHOPDATA	YFS_ORDER_HEADER		TABLE	250,888	38.42
SHOP	SHOPDATA	YFS_INVENTORY_ITEM		TABLE	83,732	12.82
SHOP	SHOPINDEX2	YFS_ORDER_HEADER_I9		INDEX	63,183	9.68
SHOP	SHOPINDEX1	YFS_STATISTICS_DETAIL_PK		INDEX	30,675	4.70
SHOP	SHOPINDEX1	YFS_ORDER_RELEASE_STATUS_PK		INDEX	25,905	3.97

## • Top statement

```
SELECT /*SHOP*/ YFS_ORDER_HEADER.*
FROM YFS_ORDER_HEADER
WHERE (ORDER_HEADER_KEY = :1 )
FOR UPDATE
```

– 40% of RLWs are on that object

# Top SQL

## SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 43.8% of Total DB Time (s): 1,658,465
- Captured PL/SQL account for 13.9% of Total DB Time (s): 1,658,465

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
204,008.97	1,967,640	0.10	12.30	0.59	2.83	ccfrfn4vuh0	ScheduleReturn	SELECT #SHOP# YFS_ORDER_HE...
116,892.49	26,600,256	0.00	7.05	6.12	0.00	4xrbd7rw15at7	INT&CAP_AgentServer	BEGIN DBMS_APPLICATION_INFO.SE...
110,919.39	28,684,565	0.00	6.69	6.30	0.00	bmx990q2tsqib	JDBC Thin Client	BEGIN DBMS_APPLICATION_INFO.SE...
79,569.63	13,172,207	0.01	4.80	8.40	0.00	8pvr9fks6m1r9	ADJUSTEOMSINV	SELECT #SHOP# TO_CHAR(sysd...
57,474.41	108,429	0.53	3.47	0.17	0.05	ff72kz03ydv4x	Sourcing	SELECT #SHOP# YFS_ORDER_HE...
20,853.07	53,990	0.39	1.26	0.17	0.00	ff10q31ahqbb5	CONFIRMFPSHIPMENT	SELECT #SHOP# YFS_ORDER_HE...
19,865.13	1,580,843	0.01	1.20	6.48	0.00	2tq3dknsjum6d	JDBC Thin Client	SELECT :SYS_B_0 FROM DUAL...
19,820.14	93,037	0.21	1.20	0.16	0.01	4htst3fakwa0m	CONSOLIDATE_ADDNL_INV	SELECT #SHOP# YFS_INVENTOR...
19,064.37	19	1,003.39	1.15	4.03	97.93	d7ufw6t6n4grn	JDBC Thin Client	SELECT #SHOP# count('#SYS_...
16,954.77	2,418,713	0.01	1.02	8.52	91.24	fqq27u174x4tt	RETEK_ORDER_CREATE	SELECT #SHOP# YFS_ORDER_LI...

- Next two statements
  - Call of DBMS\_APPLICATION\_INFO
    - Application instrumentation
  - 14% of load
  - 26M executions each
  - Instrumentation is a good thing BUT
  - Not needed since Oracle 10g
  - Use parameters to OCI or Java instead

# Other SQL

```

98sp181sbt4vc select /* SHOP */:"SYS_B_0" as ENTITY, sum(quantity) QUANTITY, DEMAND_TYPE, B.shipnode_key from yfs_inventory_demand B where B.inventory_item_key = (select inventory_item_key from yfs_inventory_item A where A.item_id = 1
and A.product_class = 2 and A.UOM = 3 and A.organization_code = 4) and B.demand_ship_date >= 5 and B.demand_ship_date <= 6 and B.shipnode_key in (.7 .8 .9 .10 .11 .12 .13 .14 .15 .16 .17 .18 .19 .20 .21 .22 .
23 .24 .25 .26 .27 .28 .29 .30 .31 .32 .33 .34 .35 .36 .37 .38 .39 .40 .41 .42 .43 .44 .45 .46 .47 .48 .49 .50 .51 .52 .53 .54 .55 .56 .57 .58 .59 .60 .61 .62 .63 .64 .65 .66 .67 .68 .69 .70 .
71 .72 .73 .74 .75 .76 .77 .78 .79 .80 .81 .82 .83 .84 .85 .86 .87 .88 .89 .90 .91 .92 .93 .94 .95 .96 .97 .98 .99 .100 .101 .102 .103 .104 .105 .106 .107 .108 .109 .110 .111 .112 .113 .114 .
115 .116 .117 .118 .119 .120 .121 .122 .123 .124 .125 .126 .127 .128 .129 .130 .131 .132 .133 .134 .135 .136 .137 .138 .139 .140 .141 .142 .143 .144 .145 .146 .147 .148 .149 .150 .151 .152 .
153 .154 .155 .156 .157 .158 .159 .160 .161 .162 .163 .164 .165 .166 .167 .168 .169 .170 .171 .172 .173 .174 .175 .176 .177 .178 .179 .180 .181 .182 .183 .184 .185 .186 .187 .188 .189 .190 .
191 .192 .193 .194 .195 .196 .197 .198 .199 .200 .201 .202 .203 .204 .205 .206 .207 .208 .209 .210 .211 .212 .213 .214 .215 .216 .217 .218 .219 .220 .221 .222 .223 .224 .225 .226 .227 .228 .
229 .230 .231 .232 .233 .234 .235 .236 .237 .238 .239 .240 .241 .242 .243 .244 .245 .246 .247 .248 .249 .250 .251 .252 .253 .254 .255 .256 .257 .258 .259 .260 .261 .262 .263 .264 .265 .266 .
267 .268 .269 .270 .271 .272 .273 .274 .275 .276 .277 .278 .279 .280 .281 .282 .283 .284 .285 .286 .287 .288 .289 .290 .291 .292 .293 .294 .295 .296 .297 .298 .299 .300 .301 .302 .303 .304 .
305 .306 .307 .308 .309 .310 .311 .312 .313 .314 .315 .316 .317 .318 .319 .320 .321 .322 .323 .324 .325 .326 .327 .328 .329 .330 .331 .332 .333 .334 .335 .336 .337 .338 .339 .340 .341 .342 .
343 .344 .345 .346 .347 .348 .349 .350 .351 .352 .353 .354 .355 .356 .357 .358 .359 .360 .361 .362 .363 .364 .365 .366 .367 .368 .369 .370 .371 .372 .373 .374 .375 .376 .377 .378 .379 .380 .
381 .382 .383 .384 .385 .386 .387 .388 .389 .390 .391 .392 .393 .394 .395 .396 .397 .398 .399 .400 .401 .402 .403 .404 .405 .406 .407 .408 .409 .410 .411 .412 .413 .414 .415 .416 .417 .418 .
419 .420 .421 .422 .423 .424 .425 .426 .427 .428 .429 .430 .431 .432 .433 .434 .435 .436 .437 .438 .439 .440 .441 .442 .443 .444 .445 .446 .447 .448 .449 .450 .451 .452 .453 .454 .455 .456 .
457 .458 .459 .460 .461 .462 .463 .464 .465 .466 .467 .468 .469 .470 .471 .472 .473 .474 .475 .476 .477 .478 .479 .480 .481 .482 .483 .484 .485 .486 .487 .488 .489 .490 .491 .492 .493 .494 .
495 .496 .497 .498 .499 .500 .501 .502 .503 .504 .505 .506 .507 .508 .509 .510 .511 .512 .513 .514 .515 .516 .517 .518 .519 .520 .521 .522 .523 .524 .525 .526 .527 .528 .529 .530 .531 .532 .
533 .534 .535 .536 .537 .538 .539 .540 .541 .542 .543 .544 .545 .546 .547 .548 .549 .550 .551 .552 .553 .554 .555 .556 .557 .558 .559 .560 .561 .562 .563 .564 .565 .566 .567 .568 .569 .570 .
571 .572 .573 .574 .575 .576 .577 .578 .579 .580 .581 .582 .583 .584 .585 .586 .587 .588 .589 .590 .591 .592 .593 .594 .595 .596 .597 .598 .599 .600 .601 .602 .603 .604 .605 .606 .607 .608 .
609 .610 .611 .612 .613 .614 .615 .616 .617 .618 .619 .620 .621 .622 .623 .624 .625 .626 .627 .628 .629 .630 .631 .632 .633 .634 .635 .636 .637 .638 .639 .640 .641 .642 .643 .644 .645 .646 .
647 .648 .649 .650 .651 .652 .653 .654 .655 .656 .657 .658 .659 .660 .661 .662 .663 .664 .665 .666 .667 .668 .669 .670 .671 .672 .673 .674 .675 .676 .677 .678 .679 .680 .681 .682 .683 .684 .
685 .686 .687 .688 .689 .690 .691 .692 .693 .694 .695 .696 .697 .698 .699 .700 .701 .702 .703 .704 .705 .706 .707 .708 .709 .710 .711 .712 .713 .714 .715 .716 .717 .718 .719 .720 .721 .722 .
723 .724 .725 .726 .727 .728 .729 .730 .731 .732 .733 .734 .735 .736 .737 .738 .739 .740 .741 .742 .743 .744 .745 .746 .747 .748 .749 .750 .751 .752 .753 .754 .755 .756 .757 .758 .759 .760 .
761 .762 .763 .764 .765 .766 .767 .768 .769 .770 .771 .772 .773 .774 .775 .776 .777 .778 .779 .780 .781 .782 .783 .784 .785 .786 .787 .788 .789 .790 .791 .792 .793 .794 .795 .796 .797 .798 .
799 .800 .801 .802 .803 .804 .805 .806 .807 .808 .809 .810 .811 .812 .813 .814 .815 .816 .817 .818 .819 .820 .821 .822 .823 .824 .825 .826 .827 .828 .829 .830 .831 .832 .833 .834 .835 .836 .
837 .838 .839 .840 .841 .842 .843 .844 .845 .846 .847 .848 .849 .850 .851 .852 .853 .854 .855 .856 .857 .858 .859 .860 .861 .862 .863 .864 .865 .866 .867 .868 .869 .870 .871 .872 .873 .874 .
875 .876 .877 .878 .879 .880 .881 .882 .883 .884 .885 .886 .887 .888 .889 .890 .891 .892 .893 .894 .895 .896 .897 .898 .899 .900 .901 .902 .903 .904 .905 .906 .907 .908 .909 .910 .911 .912 .
913 .914 .915 .916 .917 .918 .919 .920 .921 .922 .923 .924 .925 .926 .927 .928 .929 .930 .931 .932 .933 .934 .935 .936 .937 .938 .939 .940 .941 .942 .943 .944 .945 .946 .947 .948 .949 .950 .
951 .952 .953 .954 .955 .956 .957 .958 .959 .960 .961 .962 .963 .964 .965 .966 .967 .968 .969 .970 .971 .972 .973 .974 .975 .976 .977 .978 .979 .980 .981 .982 .983 .984 .985 .986 .987 .988 .
989 .990 .991 .992 .993 .994 .995 .996 .997 .998 .999 .1000 .1001 .1002 .1003 .1004 .1005 .1006)group by B.demand_type, B.shipnode_key union select /* SHOP */:"SYS_B_1" as ENTITY, sum(quantity) QUANTITY,
DEMAND_TYPE, B.shipnode_key from yfs_inventory_demand_addnl B where B.inventory_item_key = (select inventory_item_key from yfs_inventory_item A where A.item_id = 1007 and A.product_class = 1008 and A.UOM = 1009 and
A.organization_code = 1010) and B.demand_ship_date >= 1011 and B.demand_ship_date <= 1012 and B.shipnode_key in (.1013 .1014 .1015 .1016 .1017 .1018 .1019 .1020 .1021 .1022 .1023 .1024 .1025 .1026 .1027 .
1028 .1029 .1030 .1031 .1032 .1033 .1034 .1035 .1036 .1037 .1038 .1039 .1040 .1041 .1042 .1043 .1044 .1045 .1046 .1047 .1048 .1049 .1050 .1051 .1052 .1053 .1054 .1055 .1056 .1057 .1058 .1059 .
1060 .1061 .1062 .1063 .1064 .1065 .1066 .1067 .1068 .1069 .1070 .1071 .1072 .1073 .1074 .1075 .1076 .1077 .1078 .1079 .1080 .1081 .1082 .1083 .1084 .1085 .1086 .1087 .1088 .1089 .1090 .1091 .
1092 .1093 .1094 .1095 .1096 .1097 .1098 .1099 .1100 .1101 .1102 .1103 .1104 .1105 .1106 .1107 .1108 .1109 .1110 .1111 .1112 .1113 .1114 .1115 .1116 .1117 .1118 .1119 .1120 .1121 .1122 .1123 .1124 .
1125 .1126 .1127 .1128 .1129 .1130 .1131 .1132 .1133 .1134 .1135 .1136 .1137 .1138 .1139 .1140 .1141 .1142 .1143 .1144 .1145 .1146 .1147 .1148 .1149 .1150 .1151 .1152 .1153 .1154 .1155 .1156 .
1157 .1158 .1159 .1160 .1161 .1162 .1163 .1164 .1165 .1166 .1167 .1168 .1169 .1170 .1171 .1172 .1173 .1174 .1175 .1176 .1177 .1178 .1179 .1180 .1181 .1182 .1183 .1184 .1185 .1186 .1187 .1188 .1189 .
1190 .1191 .1192 .1193 .1194 .1195 .1196 .1197 .1198 .1199 .1200 .1201 .1202 .1203 .1204 .1205 .1206 .1207 .1208 .1209 .1210 .1211 .1212 .1213 .1214 .1215 .1216 .1217 .1218 .1219 .1220 .1221 .
1222 .1223 .1224 .1225 .1226 .1227 .1228 .1229 .1230 .1231 .1232 .1233 .1234 .1235 .1236 .1237 .1238 .1239 .1240 .1241 .1242 .1243 .1244 .1245 .1246 .1247 .1248 .1249 .1250 .1251 .1252 .1253 .
1254 .1255 .1256 .1257 .1258 .1259 .1260 .1261 .1262 .1263 .1264 .1265 .1266 .1267 .1268 .1269 .1270 .1271 .1272 .1273 .1274 .1275 .1276 .1277 .1278 .1279 .1280 .1281 .1282 .1283 .1284 .1285 .
1286 .1287 .1288 .1289 .1290 .1291 .1292 .1293 .1294 .1295 .1296 .1297 .1298 .1299 .1300 .1301 .1302 .1303 .1304 .1305 .1306 .1307 .1308 .1309 .1310 .1311 .1312 .1313 .1314 .1315 .1316 .1317 .
1318 .1319 .1320 .1321 .1322 .1323 .1324 .1325 .1326 .1327 .1328 .1329 .1330 .1331 .1332 .1333 .1334 .1335 .1336 .1337 .1338 .1339 .1340 .1341 .1342 .1343 .1344 .1345 .1346 .1347 .1348 .1349 .
1350 .1351 .1352 .1353 .1354 .1355 .1356 .1357 .1358 .1359 .1360 .1361 .1362 .1363 .1364 .1365 .1366 .1367 .1368 .1369 .1370 .1371 .1372 .1373 .1374 .1375 .1376 .1377 .1378 .1379 .1380 .1381 .
1382 .1383 .1384 .1385 .1386 .1387 .1388 .1389 .1390 .1391 .1392 .1393 .1394 .1395 .1396 .1397 .1398 .1399 .1400 .1401 .1402 .1403 .1404 .1405 .1406 .1407 .1408 .1409 .1410 .1411 .1412 .1413 .
1414 .1415 .1416 .1417 .1418 .1419 .1420 .1421 .1422 .1423 .1424 .1425 .1426 .1427 .1428 .1429 .1430 .1431 .1432 .1433 .1434 .1435 .1436 .1437 .1438 .1439 .1440 .1441 .1442 .1443 .1444 .1445 .
1446 .1447 .1448 .1449 .1450 .1451 .1452 .1453 .1454 .1455 .1456 .1457 .1458 .1459 .1460 .1461 .1462 .1463 .1464 .1465 .1466 .1467 .1468 .1469 .1470 .1471 .1472 .1473 .1474 .1475 .1476 .1477 .
1478 .1479 .1480 .1481 .1482 .1483 .1484 .1485 .1486 .1487 .1488 .1489 .1490 .1491 .1492 .1493 .1494 .1495 .1496 .1497 .1498 .1499 .1500 .1501 .1502 .1503 .1504 .1505 .1506 .1507 .1508 .1509 .
1510 .1511 .1512 .1513 .1514 .1515 .1516 .1517 .1518 .1519 .1520 .1521 .1522 .1523 .1524 .1525 .1526 .1527 .1528 .1529 .1530 .1531 .1532 .1533 .1534 .1535 .1536 .1537 .1538 .1539 .1540 .1541 .
1542 .1543 .1544 .1545 .1546 .1547 .1548 .1549 .1550 .1551 .1552 .1553 .1554 .1555 .1556 .1557 .1558 .1559 .1560 .1561 .1562 .1563 .1564 .1565 .1566 .1567 .1568 .1569 .1570 .1571 .1572 .1573 .
1574 .1575 .1576 .1577 .1578 .1579 .1580 .1581 .1582 .1583 .1584 .1585 .1586 .1587 .1588 .1589 .1590 .1591 .1592 .1593 .1594 .1595 .1596 .1597 .1598 .1599 .1600 .1601 .1602 .1603 .1604 .1605 .
1606 .1607 .1608 .1609 .1610 .1611 .1612 .1613 .1614 .1615 .1616 .1617 .1618 .1619 .1620 .1621 .1622 .1623 .1624 .1625 .1626 .1627 .1628 .1629 .1630 .1631 .1632 .1633 .1634 .1635 .1636 .1637 .
1638 .1639 .1640 .1641 .1642 .1643 .1644 .1645 .1646 .1647 .1648 .1649 .1650 .1651 .1652 .1653 .1654 .1655 .1656 .1657 .1658 .1659 .1660 .1661 .1662 .1663 .1664 .1665 .1666 .1667 .1668 .1669 .
1670 .1671 .1672 .1673 .1674 .1675 .1676 .1677 .1678 .1679 .1680 .1681 .1682 .1683 .1684 .1685 .1686 .1687 .1688 .1689 .1690 .1691 .1692 .1693 .1694 .1695 .1696 .1697 .1698 .1699 .1700 .1701 .
1702 .1703 .1704 .1705 .1706 .1707 .1708 .1709 .1710 .1711 .1712 .1713 .1714 .1715 .1716 .1717 .1718 .1719 .1720 .1721 .1722 .1723 .1724 .1725 .1726 .1727 .1728 .1729 .1730 .1731 .1732 .1733 .
1734 .1735 .1736 .1737 .1738 .1739 .1740 .1741 .1742 .1743 .1744 .1745 .1746 .1747 .1748 .1749 .1750 .1751 .1752 .1753 .1754 .1755 .1756 .1757 .1758 .1759 .1760 .1761 .1762 .1763 .1764 .1765 .
1766 .1767 .1768 .1769 .1770 .1771 .1772 .1773 .1774 .1775 .1776 .1777 .1778 .1779 .1780 .1781 .1782 .1783 .1784 .1785 .1786 .1787 .1788 .1789 .1790 .1791 .1792 .1793 .1794 .1795 .1796 .1797 .
1798 .1799 .1800 .1801 .1802 .1803 .1804 .1805 .1806 .1807 .1808 .1809 .1810 .1811 .1812 .1813 .1814 .1815 .1816 .1817 .1818 .1819 .1820 .1821 .1822 .1823 .1824 .1825 .1826 .1827 .1828 .1829 .
1830 .1831 .1832 .1833 .1834 .1835 .1836 .1837 .1838 .1839 .1840 .1841 .1842 .1843 .1844 .1845 .1846 .1847 .1848 .1849 .1850 .1851 .1852 .1853 .1854 .1855 .1856 .1857 .1858 .1859 .1860 .1861 .
1862 .1863 .1864 .1865 .1866 .1867 .1868 .1869 .1870 .1871 .1872 .1873 .1874 .1875 .1876 .1877 .1878 .1879 .1880 .1881 .1882 .1883 .1884 .1885 .1886 .1887 .1888 .1889 .1890 .1891 .1892 .1893 .
1894 .1895 .1896 .1897 .1898 .1899 .1900 .1901 .1902 .1903 .1904 .1905 .1906 .1907 .1908 .1909 .1910 .1911 .1912 .1913 .1914 .1915 .1916 .1917 .1918 .1919 .1920 .1921 .1922 .1923 .1924 .1925 .
1926 .1927 .1928 .1929 .1930 .1931 .1932 .1933 .1934 .1935 .1936 .1937 .1938 .1939 .1940 .1941 .1942 .1943 .1944 .1945 .1946 .1947 .1948 .1949 .1950 .1951 .1952 .1953 .1954 .1955 .1956 .1957 .
1958 .1959 .1960 .1961 .1962 .1963 .1964 .1965 .1966 .1967 .1968 .1969 .1970 .1971 .1972 .1973 .1974 .1975 .1976 .1977 .1978 .1979 .1980 .1981 .1982 .1983 .1984 .1985 .1986 .1987 .1988 .1989 .
1990 .1991 .1992 .1993 .1994 .1995 .1996 .1997 .1998 .1999 .2000 .2001 .2002 .2003 .2004 .2005 .2006 .2007 .2008 .2009 .2010 .2011 .2012)group by B.demand_type, B.shipnode_key

```



# Online Summary

## Not looking good for Black Friday

- System is CPU bound at test load levels
- System seems to be leaking both cursors and sessions (and maybe locks)
- System is running far too many processes
- High overhead application instrumentation