

**ORACLE®**



ORACLE®

# Maximizing Database Performance Using Database Replay

Kurt Engeleiter, Oracle

Arup Nanda, Starwood Hotels and Resorts



ORACLE  
**OPEN**  
WORLD

**HARDWARE  
AND SOFTWARE  
ENGINEERED  
TO WORK  
TOGETHER**

# Safe Harbor Statements

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Program Agenda

- Database Performance Fundamentals
- Database Replay
- Maximizing Performance - Case Studies
- Capacity Planning

# Database Performance Fundamentals



# Database Tuning Methodology

## Ensure host resources are not the bottleneck

- Check that memory is not exhausted (not swapping)
- Tune for CPU first when CPU constrained

## Tune to reduce DB Time

- Performance improvement means accomplishing the same amount of work with less DB Time

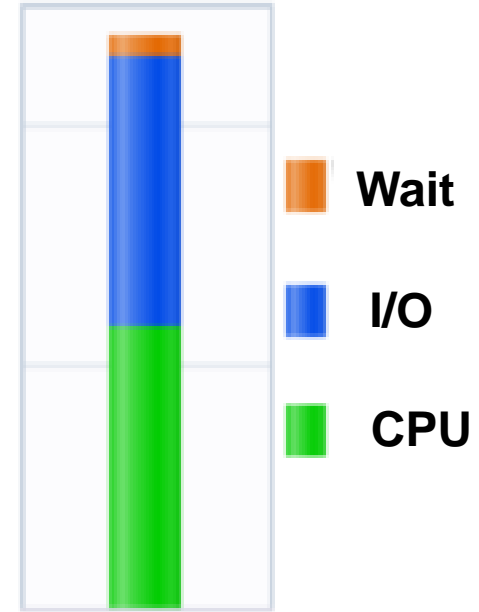
## Validate tuning

- Corroborate tuning results with Real Application Testing

# DB Time Definition

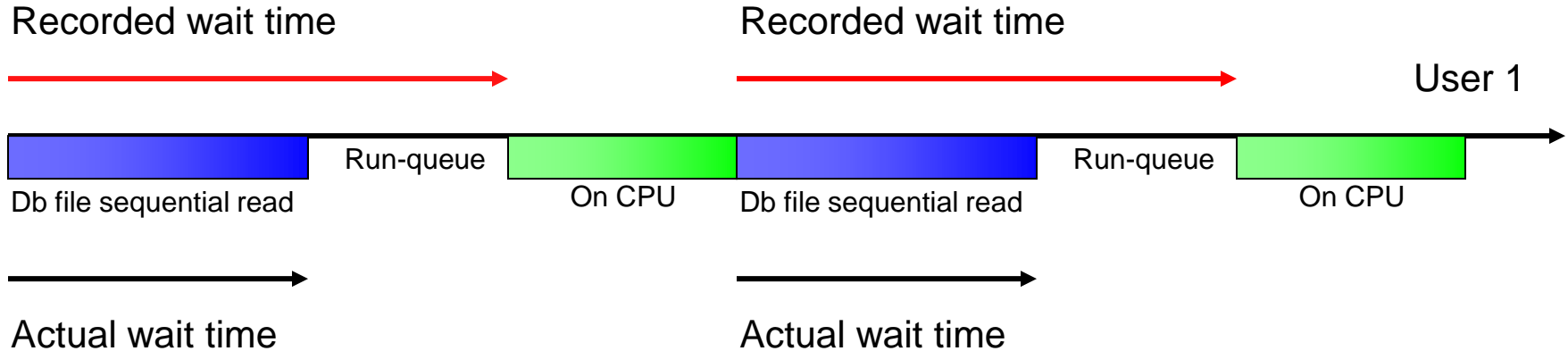
- Total time in database calls by **foreground sessions**
- Includes
  - CPU time
  - IO time
  - Active wait (non-idle wait) time
- Basic unit for Oracle performance analysis

DB Time



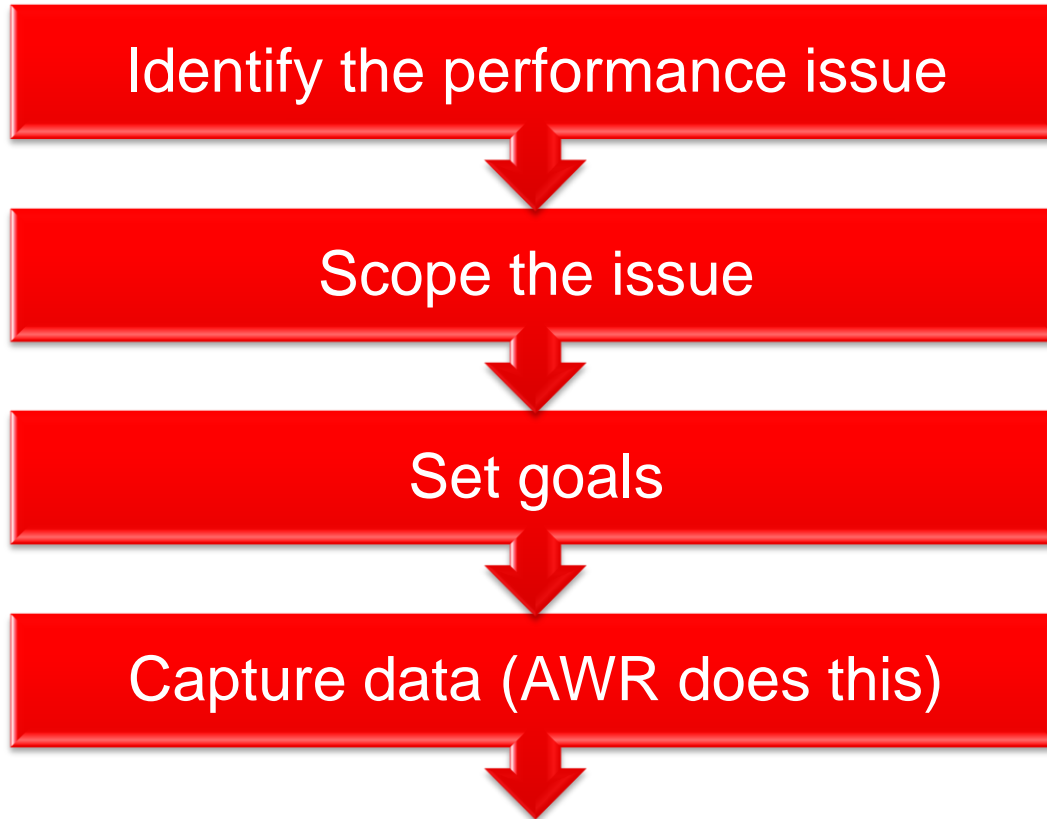
# CPU Run-Queue and DB Time

- DB Time is inflated when CPU Bound

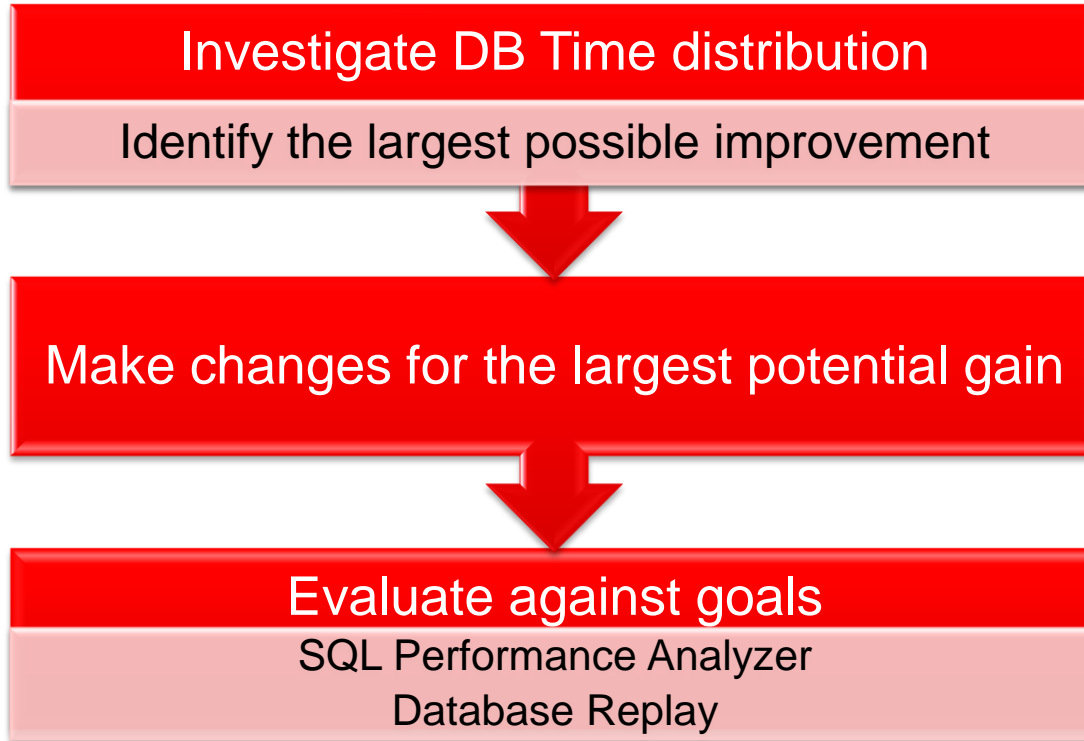




# Database System Tuning Process



# Database System Tuning Process

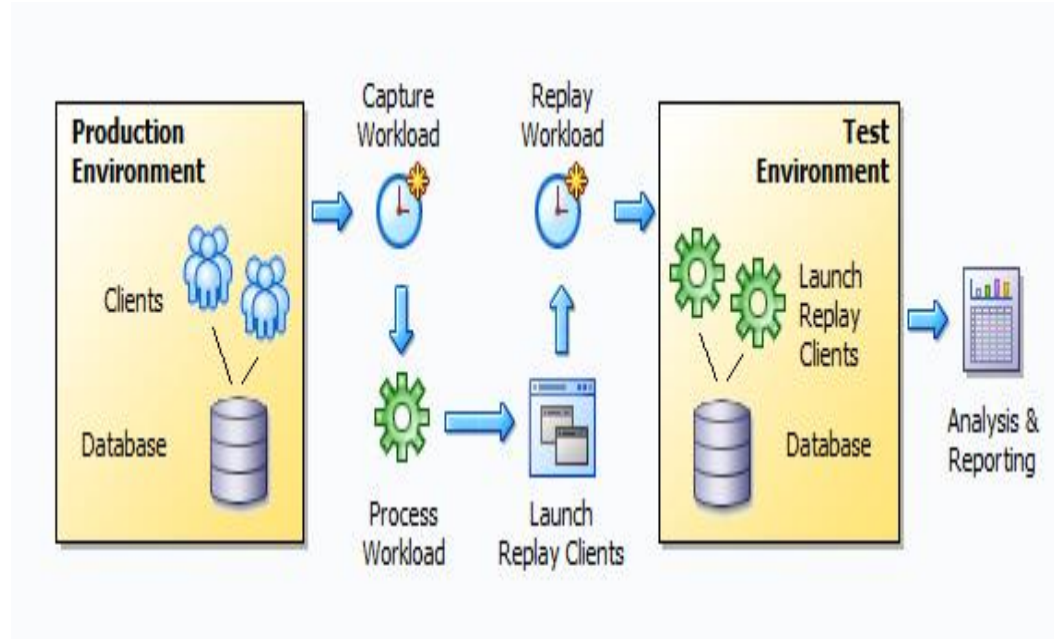


# Database Replay



# Database Replay

- Database load and performance testing with real production workloads
  - Production workload characteristics such as timing, transaction dependency, think time, etc., fully maintained
- Identify and remediate application scalability and concurrency problems in multitenant and non-CDB databases
- Allows scheduling, scaleup, subsetting, of multiple workloads
- Concurrent database replay available for 11.2.0.2 and above, MOS Note: 1453789.1



# Database Replay Workflow

Database Replay > Capture: P\_SALES\_01

Capture: P\_SALES\_01 Auto Refresh Off

Summary | Replay Tasks | Workload Subsets | Review

### Capture Summary

Name: P\_SALES\_01  
Status: In Progress Stop Capture  
Owner: SYSMAN  
Description: Batch two hour workload  
Concurrent Capture: No  
Database Replay Capture Job: DBREPLAY\_P\_SALES\_01\_1376154165504\_C... (Running)  
Database Target: o12c.us.oracle.com  
Database Name: O12C  
Database Version: 12.1.0.1.0  
Cluster Database: No  
DBID: 3445445623  
Capture Error Code:  
Capture Error Message:  
Captured Data Size (MB): 1.642  
Start SCN: 4852638  
End SCN:  
SQL Tuning Set Name: P\_SALES\_01\_c\_26150135  
Storage Host: slc00trp.us.oracle.com  
Storage Location: /scratch/kengelei/ooow2013/captures/DBRepla...  
Capture Duration (hh:mm:ss):  
Scheduled Capture Start Time: 10:02:45 AM GMT-07:00  
Scheduled Capture End Time: 12:02:45 PM GMT-07:00  
AWR Data Export Schedule: Start immediately after capture completes.

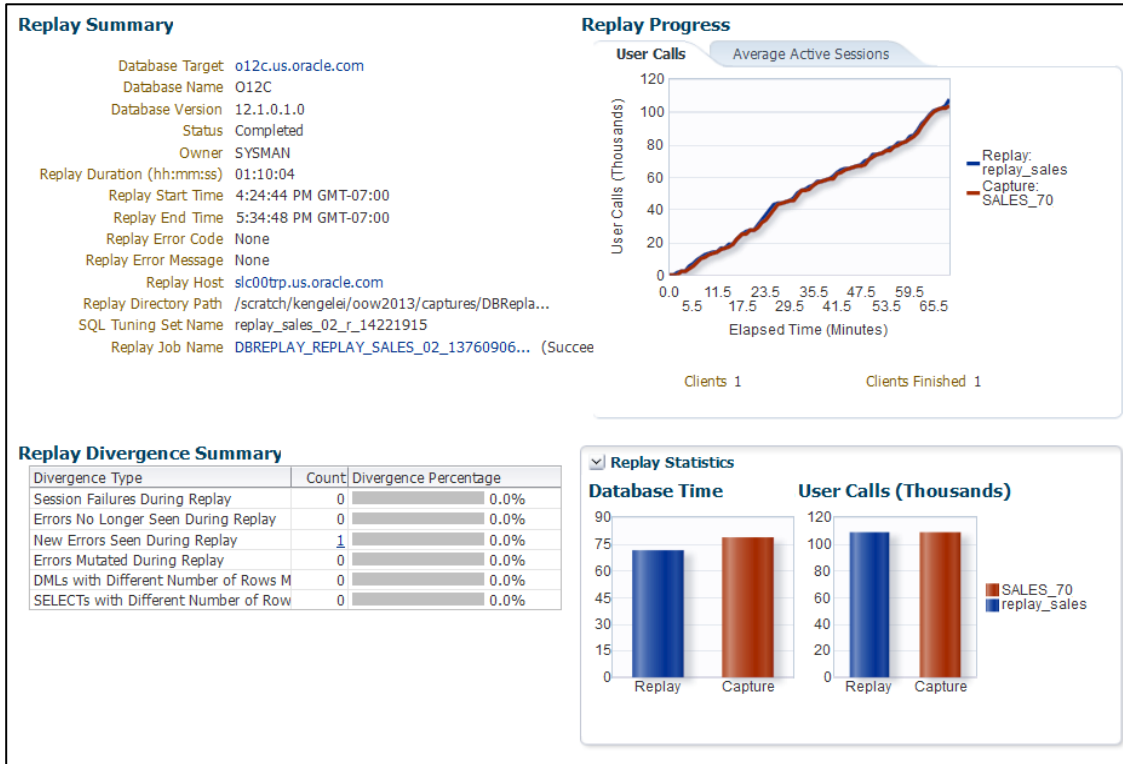
### Average Active Sessions

Comparison

	Capture	Total	Percentage of Total
Database Time (hh:mm:ss)	00:06:19	00:06:28	97.47%
Average Active Sessions	0.604	0.619	97.47%
User Calls	18,122	18,388	98.55%
Transactions	0	37	0%
Session Logins	147	189	77.78%

- Comprehensive interface for Database Replay in Enterprise Manager
- Database Capture page highlights performance data along with workload information
- DB Time displayed in Average Active Session graph

# Database Replay Workflow



- Database Replay workflow monitoring:
  - User call progress
  - DB Time summarized in bar chart
  - Replay Divergence Summary enables easy analysis with click through links



# Database Replay Workflow

Database Replay Page Refreshed 3:15:26 PM PDT

Database Replay > Replay Task: REPLAY\_SALES > Replay: replay\_sales\_01

## Replay: replay\_sales\_01

Home Reports Review

### Replay Reports

- Database Replay Report [View](#)
- Compare Period ADDM Report [View](#)
- SQL Performance Analyzer Report [View](#)
- Replay Compare Period Report [View](#)
- Replay ASH Analytics Report [Replayed Workload SQL and Wait Events by Wait Class](#) [View](#)

[Regenerate Reports](#)

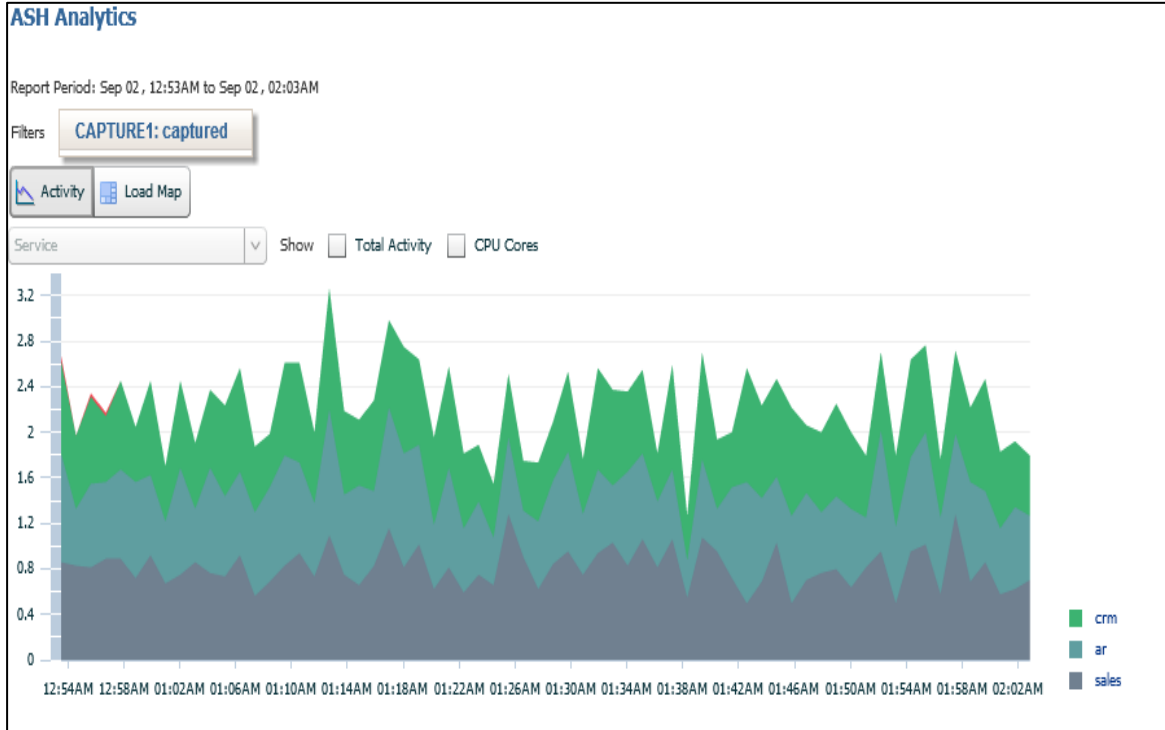
**Information**  
If the replay reports were not generated due to blocking issues, you can regenerate the replay reports after any blocking issues are resolved.

### Replay Issues

Step Name	Step Status	Start Time	End Time	Target Name	Target Type	Job Name
No replay issues found						

- Rich reporting infrastructure leverages ASH and AWR data
- ASH Analytics predefined reports - new in Enterprise Manager 12c

# Database Replay Workflow



- Predefined ASH Analytics reports by:
  - Wait Class
  - User
  - Service
  - Module
- Additional ASH Analytics reports can be generated at will



# Maximizing Performance Case Studies

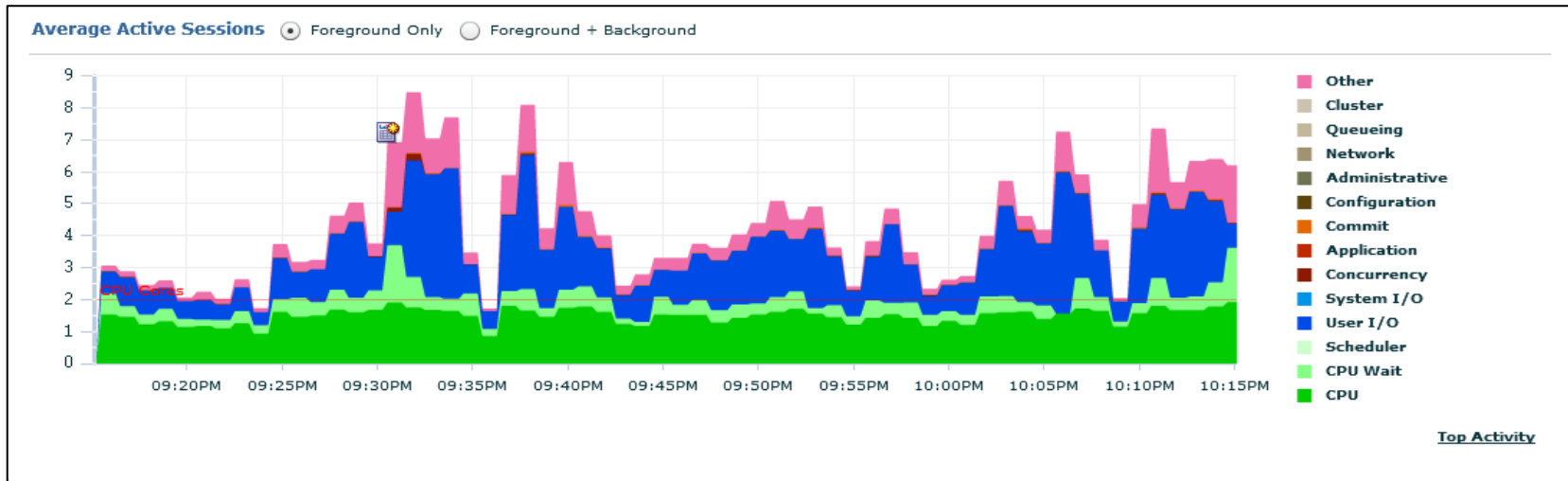


ORACLE

# Case 1



# Tuning with DB Time – Case 1



- Current workload, two CPU system
- Average Active Sessions consistently greater than two for this workload
  - Average Active Sessions is DBTime over Time
- Workload to be tuned is captured with Database Replay

# Tuning With DB Time – Case 1

- Many DB Time tuning options:

Manual Tuning – Read AWR Report

Automatic Database Diagnostics Monitor (ADDM)

SQL Tuning Advisor

SQL Access Advisor

# Tuning With DB Time – Case 1

- In this case:

## ADDM

- Recommends running SQL Tuning Advisor on high DB Time SQL




## SQL Tuning Advisor

- Recommends additional access structures
- Suggests running SQL Access Advisor to confirm

## SQL Access Advisor

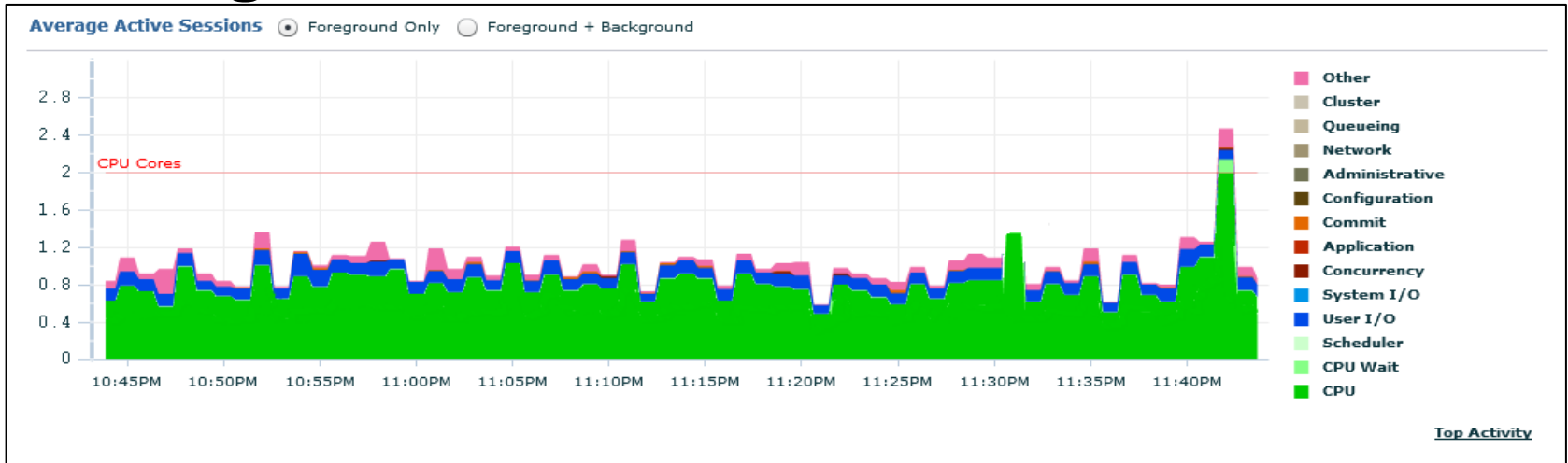
- Verifies the benefit for the whole workload

# Tuning with DB Time – Case 1

Action	Object Name	Object Attributes	Indexed Columns	Base Table	Schema	Tablespace	Partition Key	SQL Partition
PARTITION_TABLE	H_LINEITEM				SALES		("L_SHIPDATE")	PARTITION BY RANGE ("L_SHIPDATE") INTER...
PARTITION_TABLE	H_ORDER				SALES		("O_ORDERDATE")	PARTITION BY RANGE ("O_ORDERDATE") INTER...
PARTITION_TABLE	H_CUSTOMER				SALES		("C_NATIONKEY")	PARTITION BY RANGE ("C_NATIONKEY") INTER...
PARTITION_TABLE	H_SUPPLIER				SALES		("S_NATIONKEY")	PARTITION BY RANGE ("S_NATIONKEY") INTER...

- SQL Access Advisor recommends partitioning tables as best option for this workload

# Tuning with DB Time – Case 1



- Database copied to test system
- Partitioning implemented per SQL Access Advisor recommendation
- Workload replayed with Database Replay
  - Average Active Sessions reduced from two to one during replay

# Tuning with DB Time – Case 1

## Before

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
O12C	3445445623	o12c	1	16:08	12.1.0.1.0	NO
Host Name	Platform	CPU(s)	Cores	Sockets	Memory (GB)	
slc00trp	Linux x86 64-bit	2	2	2	7.45	
Snap Id	Snap Time	Sessions	Cursors/Session			
Begin Snap:	291	21:14:25	50	2.5		
End Snap:	293	22:24:26	46	2.9		
Elapsed:	70.02 (mins)					
DB Time:	298.95 (mins)					

## After

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
O12C	3445445623	o12c	1	16:08	12.1.0.1.0	NO
Host Name	Platform	CPU(s)	Cores	Sockets	Memory (GB)	
slc00trp	Linux x86 64-bit	2	2	2	7.45	
Snap Id	Snap Time	Sessions	Cursors/Session			
Begin Snap:	296	22:41:36	54	2.5		
End Snap:	297	23:51:37	50	2.8		
Elapsed:	70.02 (mins)					
DB Time:	73.99 (mins)					

- DB Time reduced from 298 minutes to 73 minutes
- Tuning was a success ✓



## Case 2



ORACLE

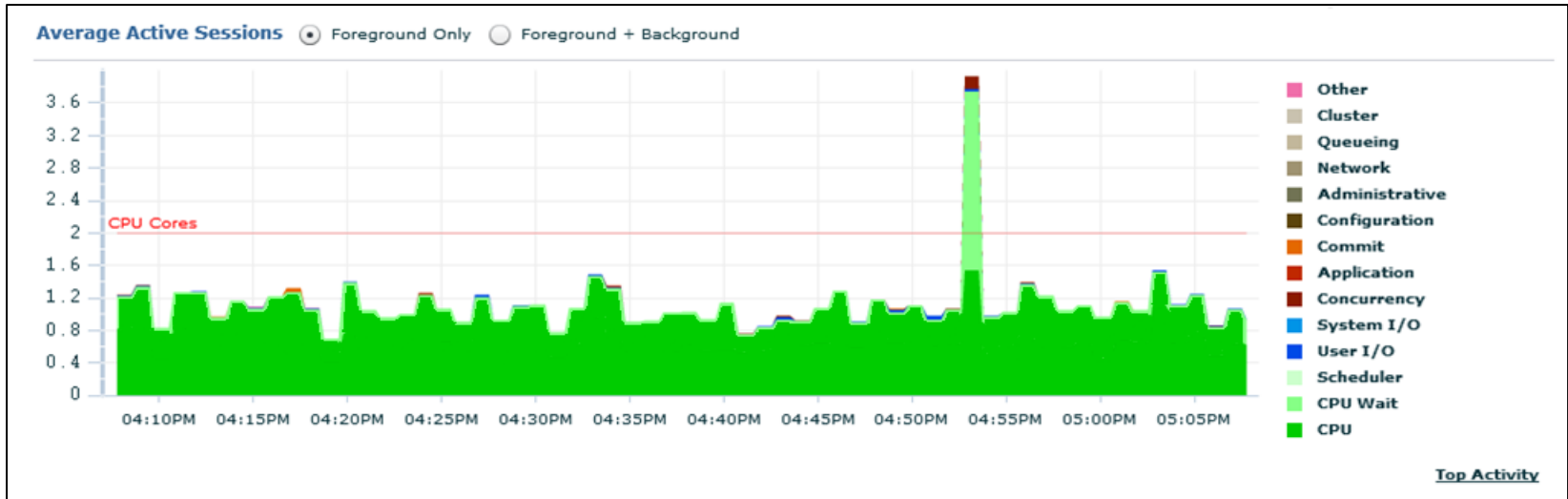
# Tuning – Case 2

Should I use a new database parameter?

- In database 12c, a new parameter, `THREADED_EXECUTIONS`, was introduced
- Definition: “`THREADED_EXECUTION` specifies whether to enable the multithreaded Oracle model.”
  - Most Oracle background processes run as threads
- Question: Will this benefit my workload?

# Tuning – Case 2

Should I use a new database parameter?



- Workload captured with Database Replay

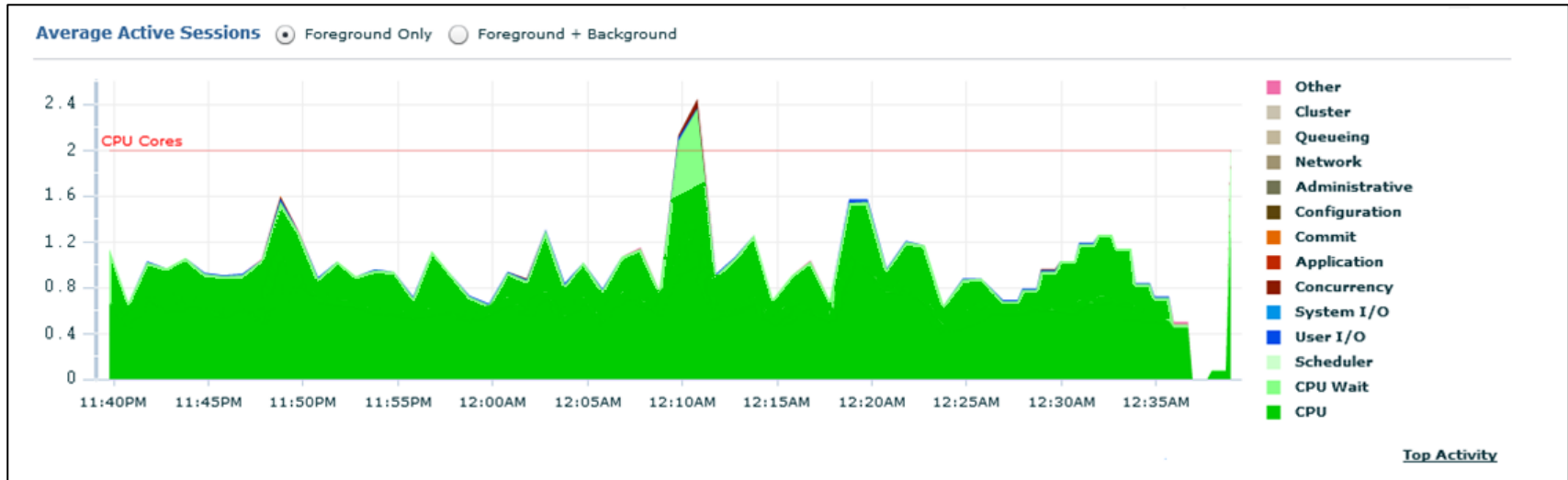
# Tuning – Case 2

## Should I use a new database parameter?

- Database restarted
  - `threaded_executions` set to true
  - only six Oracle processes

```
ora_pmon_o12c  
ora_psp0_o12c  
ora_vktm_o12c  
ora_u004_o12c  
ora_u005_o12c  
ora_dbw0_o12c
```

# Tuning – Case 2



- Workload replayed with Database Replay
- No obvious performance changes

# Tuning with DB Time – Case 1

## Before

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
O12C	3445445623	o12c	1	15:08	12.1.0.1.0	NO

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
slc00trp	Linux x86 64-bit	2	2	2	7.45

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	340	16:07:10	50	2.4
End Snap:	342	17:17:23	46	2.9
Elapsed:		70.22 (mins)		
DB Time:		79.47 (mins)		

## After

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
O12C	3445445623	o12c	1	23:08	12.1.0.1.0	NO

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
slc00trp	Linux x86 64-bit	2	2	2	7.45

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	353	23:24:43	42	2.6
End Snap:	355	00:34:48	47	2.7
Elapsed:		70.08 (mins)		
DB Time:		74.57 (mins)		

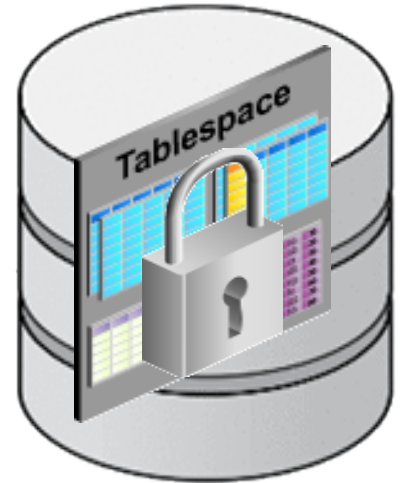
- DB Time improved from 79 to 74 minutes
- Conclusion: **THREADED\_EXECUTIONS** produced a minor improvement for this workload

# Case 3



# Case 3 – Impact of Encryption

- Customer: large non-profit education service provider
- For data security compliance, customer needs to implement Transparent TS Encryption
- Environment:
  - Solaris
  - Oracle 11g
  - Customer built application
- What will be the impact on performance with encryption enabled?



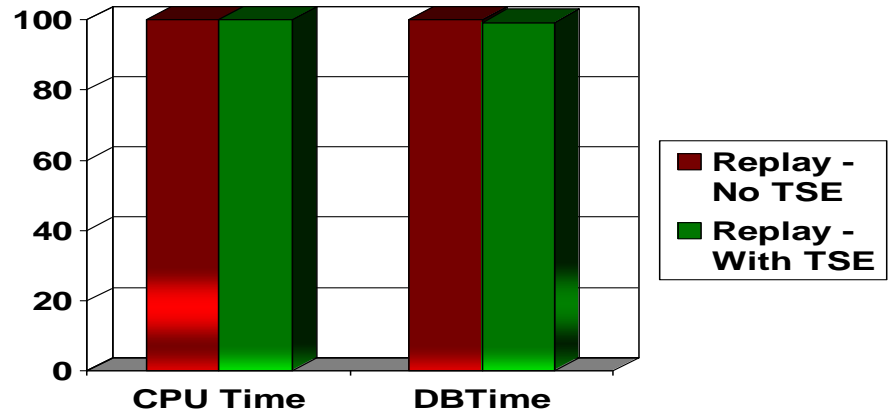


# Case 3 – Impact of Encryption

- Peak activity captured and replayed.
- Very minimal CPU and DB Time impacts observed (less than 1%)
- TSE implementation – success. ✓

## WORKLOAD COMPARE PERIOD REPORT

Snapshot Set	Begin Snap Id	Begin Snap Time	End Snap Id	End Snap Time	Avg Active Users	Elapsed Time (min)	DB time (min)
1st	4167	28-Sep 21:09:41 (Mon)	4169	28-Sep 22:11:54 (Mon)	0.59	62.21	36.51
2nd	4170	28-Sep 18:17:13 (Mon)	4172	28-Sep 19:19:25 (Mon)	0.58	62.20	36.27

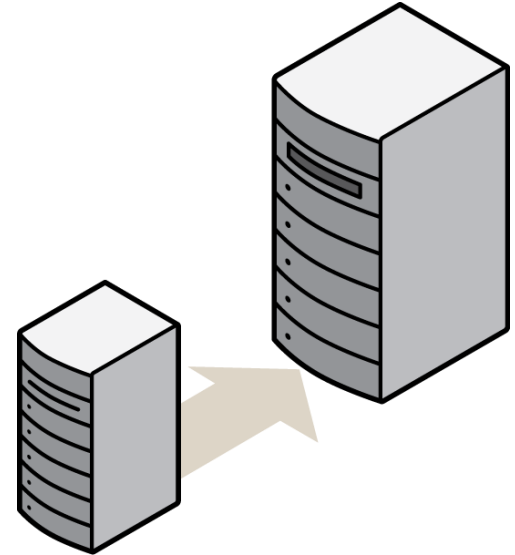


# Capacity Planning: Database Replay



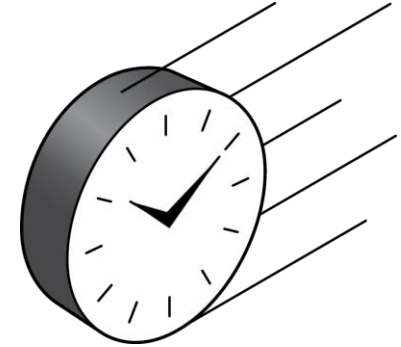
# Capacity Planning Using Database Replay

- Comprehensive scale-up support and what-if scenarios testing
- Scale-up techniques superior to traditional methods
  - Zero-scripting approach extended for scale-up
  - Scales data and user population
  - Realistic data and bindsets
  - Flexible, supports custom workload creation
- Scale-up Strategies
  - Use think time, connect time replay parameters
  - Use scale-up multiplier replay parameter
  - Scale-up by scheduling concurrent replays
  - Scale-up by workload folding
  - Scale-up with multiple PDBs



# Scale-up With Connect Time, Think Time

- Increase effective workload by reducing replay parameters - connect and think time scales
  - Same workload executed in shorter duration
  - `connect_time`: time between session connects
  - `think_time`: time between user transactions
- Pros
  - Good stress test
  - Works best for OLTP applications such as forms order entry
- Cons
  - Batch applications may not benefit as much due to minimal application latency



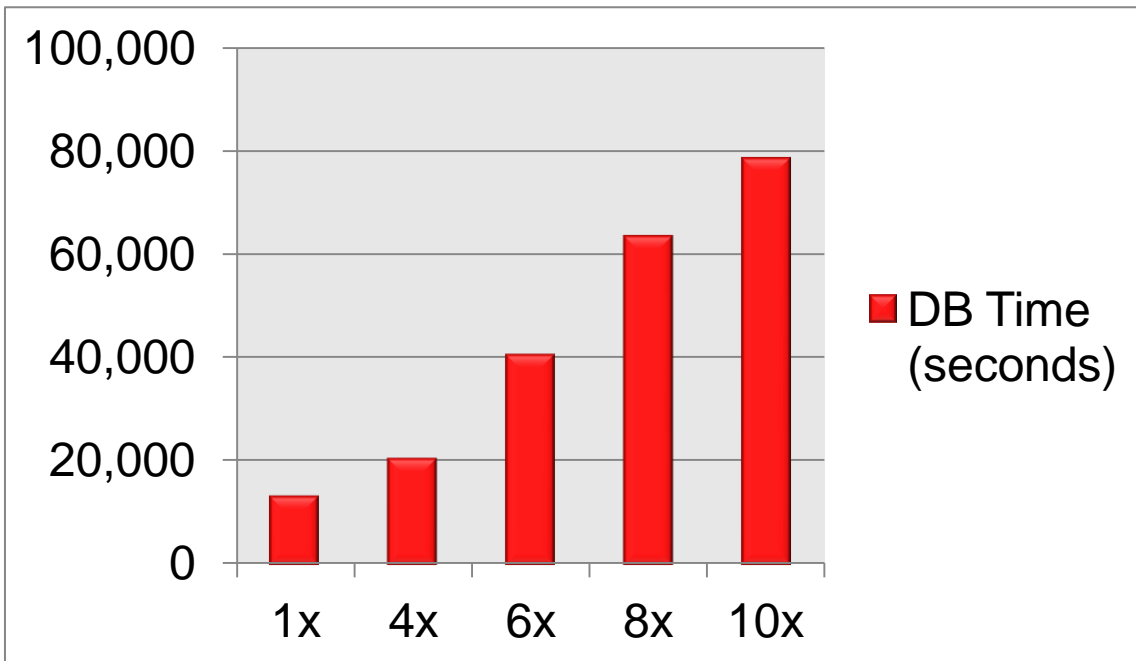
# Scale-up with Multiplier Replay Parameter

- Increase read-only workload using replay parameter **“SCALE\_UP\_MULTIPLIER”**
- Multiplies workload specified number of times
- First replay full DML, all other replays are query only
- Pros:
  - Provides good coverage for databases with significant read portion of workload
- Cons:
  - Full DML is not replicated

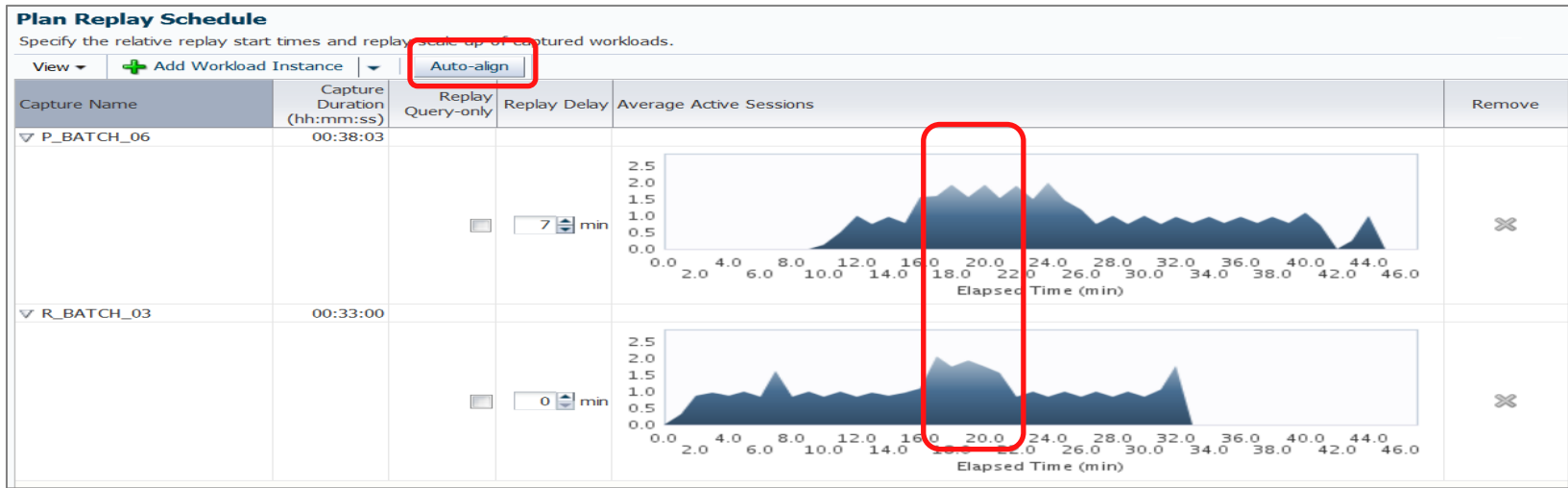


# Scale\_Up\_Multiplier and Exadata

- Large Internet company captured peak production workload and replayed on Exadata
- `SCALE_UP_MULTIPLIER` used to scale workload to 10x
- Spare capacity remained on Exadata after 10x scale up

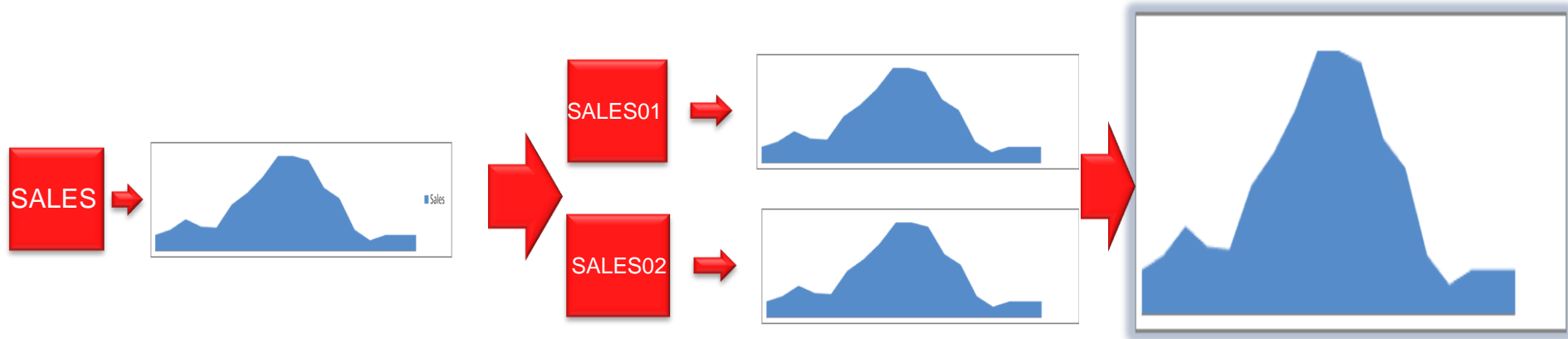


# Scale-up with Workload Scheduling



- Consolidated Database Replay includes workload scheduling
- Workload peaks can be aligned to maximize stress on replay system
- EM Cloud Control 12c also now supports workload scheduling

# Scale-up With Multiple PDBs



- Useful for scale-up testing when significant growth in the workload is expected or for testing Multitenant applications with identical schemas
- Process:
  - Set up test system with a single PDB duplicated from the initial database
  - Replicate this PDB as many times as required
  - Copy workload multiple times into consolidated replay directory and direct separate workloads to specific PDBs (services)



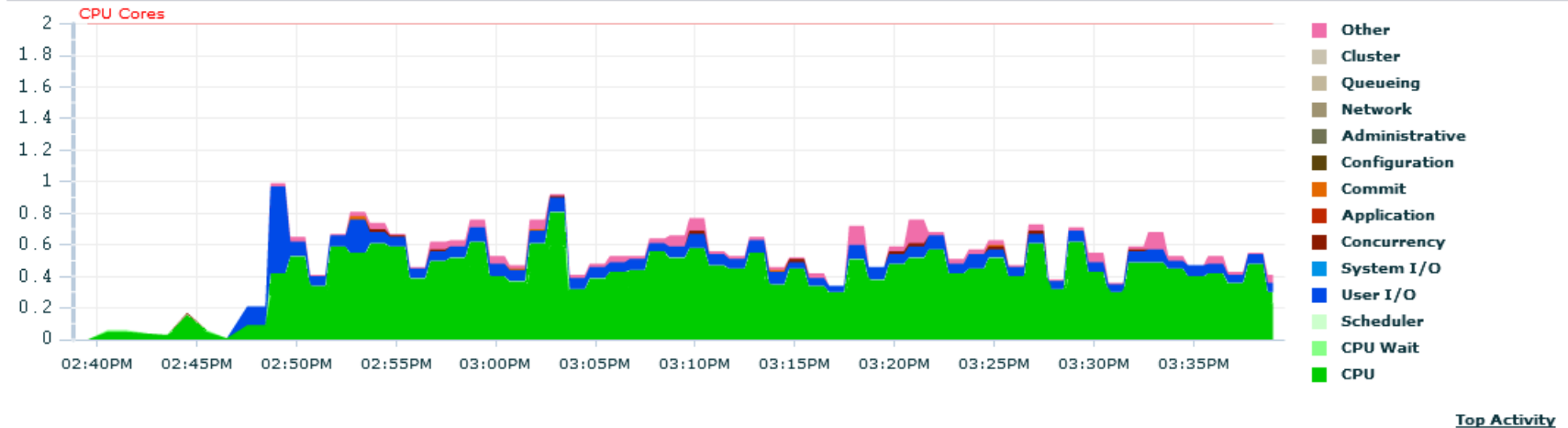
# Scale-up with Multiple PDBs

- SQL to replicate a PDB:

```
CREATE PLUGGABLE DATABASE sales01 FROM
sales
FILE_NAME_CONVERT =
('/u01/oracle/oradata/db12c/sales/',
'/u01/oracle/oradata/db12c/sales01/');
```

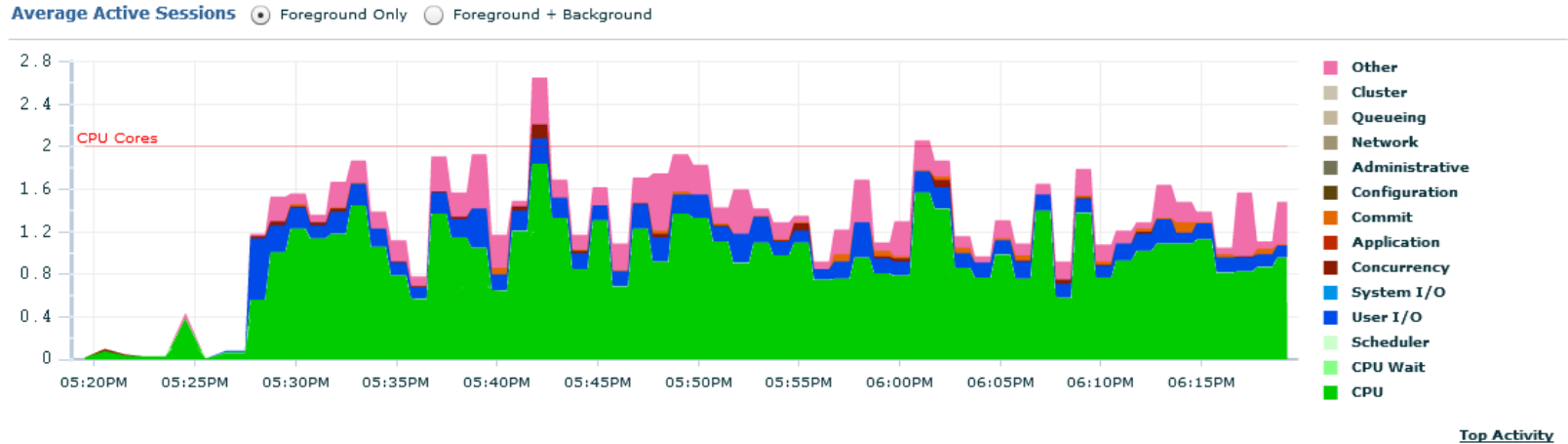
# Scale-up With Multiple PDBs

Average Active Sessions  Foreground Only  Foreground + Background



- Workload to be evaluated for scale up captured.
- System has two CPUs.
- Based on Average Active Sessions, system has room for growth.

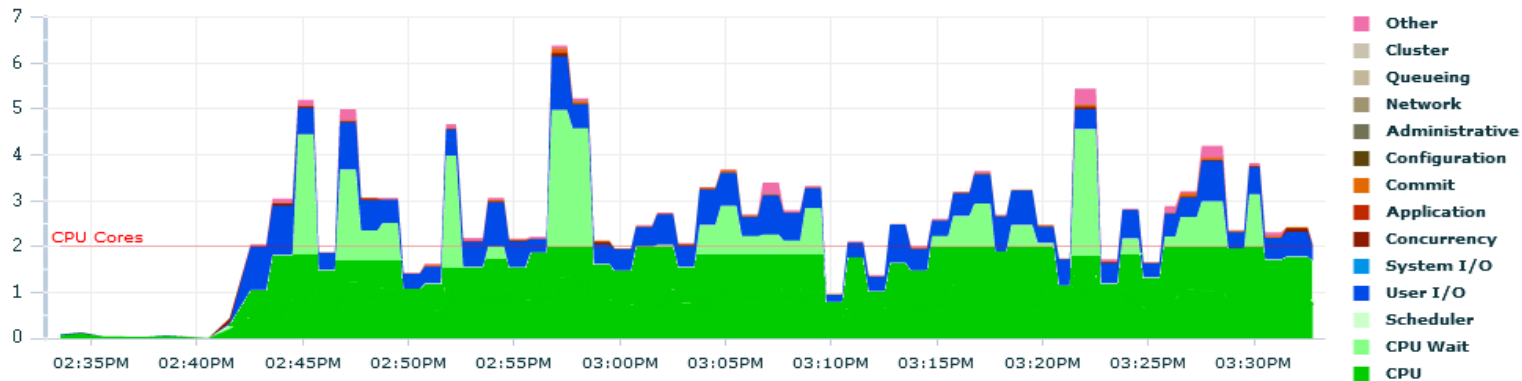
# Scale-up With Multiple PDBs



- Database Replay executed in two identical PDBs.
- Based on CPU usage, system still has room for additional workload.

# Scale-up With Multiple PDBs

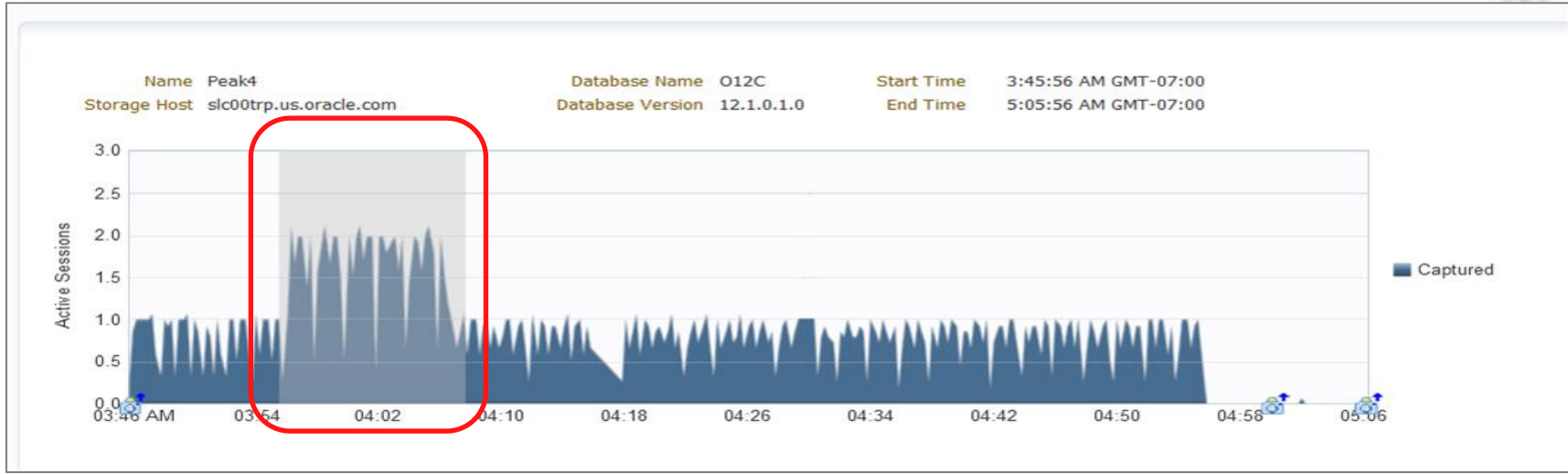
Average Active Sessions  Foreground Only  Foreground + Background



Top Activity

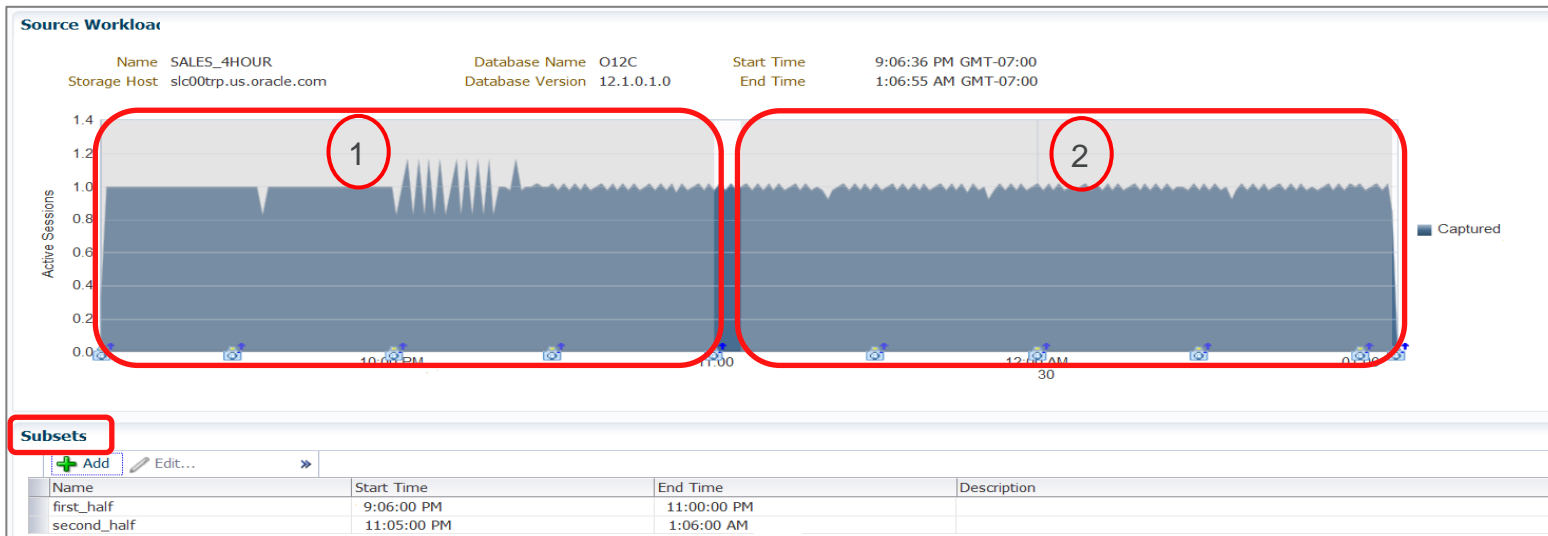
- Database Replay executed in three identical PDBs.
- Result: system is CPU saturated.
- Conclusion: system can handle more than double the current workload, but less than triple.

# Workload Subsetting



- New in Oracle Database 12c – creation of workload subsets (API)
- Select and replay the most interesting workload intervals
- EM Cloud Control 12c also now supports workload subsetting

# Scale-up with Workload Folding



- Scale-up by combining subsetting and scheduling
  - Subset a longer workload into two or more intervals
  - Replay those subsets simultaneously as independent workloads

# Database Replay Scale-up Strategies

## Summary

Scale-up Strategy	Workload Suitability
Think and connect time throttling	<ul style="list-style-type: none"><li>▪ Workloads with think or connect latency</li><li>▪ Suitable for OLTP workloads</li></ul>
Scale-up read-only multiplier	<ul style="list-style-type: none"><li>▪ Workloads that are predominantly query only</li><li>▪ Reader farms, Website read-only activity</li></ul>
Concurrent Database Replay	<ul style="list-style-type: none"><li>▪ Enables consolidation validation</li><li>▪ Can be used for all workloads</li></ul>
Multiple identical PDBs	<ul style="list-style-type: none"><li>▪ Simultaneous workload and data scale-up</li><li>▪ Multitenant what-if scenarios</li></ul>
Workload Folding	<ul style="list-style-type: none"><li>▪ <math>N * \text{Workload scale-up}</math></li><li>▪ Suitable for relatively stateless workloads</li></ul>

# Conclusion

- DB Time is the fundamental metric in database tuning
  - Improve database performance by reducing DB Time
- Validate tuning with Database Replay
- Future proof your environment by using Database Replay for consolidation, capacity planning and scale up testing





**Arup Nanda**  
**Director - Principal Database Architect**  
**Starwood Hotels & Resorts Worldwide, Inc.**

# Identifying Upgrade Issues

# Challenge

- Upgrading from Oracle 10g R2 → 11g R2
- HPUX Itanium → AIX
- Across Data Centers: Houston → Phoenix
- Downtime expected: 2 hours max

***Question: What all can fail?***

# What We Did

- Methodology

- Captured workload for 1 month in 1 hour chunks
- Replayed continuously against the new system

- Result

- Discovered database bug (actually “fixed” an earlier bug 9824198 )
- "ORA-00979: not a GROUP BY expression”
- Where, how many places?

# Identification of SQLs

```
select service, module, action, sql_id, count(*)  
from dba_workload_replay_divergence  
Where observed_error# = 979  
group by service, module, action, sql_id;
```

SERVICE	MODULE	ACTION	SQL_ID	COUNT (*)
xxx	xxx	xxx	4st8fbfa5cpt2	132
xxx	xxx	xxx	4karrnm9kt23k	13
xxx	xxx	xxx	2syvktcdcxxrsh	5
xxx	xxx	xxx	2jku6m3bpk3yc	1
xxx	xxx	xxx	cg7kdz4gsrnu5	5
...				

# Summary of Benefits

- Saved months of work for a team of 30 developers
- System tuning to perfection – in a week
- Reduction of the realm of the unknown

**Hardware and Software**

**ORACLE®**

**Engineered to Work Together**

**ORACLE®**