

**ORACLE®**



**ORACLE®**

**Designing and Tuning Large High Performance systems: Tips and Tricks**

Graham Wood  
Architect, Server Technologies

## Designing and Tuning: Tips and Tricks

# NOT

ORACLE

## Designing for Oracle

- Changes over time
  - CPU
  - IO
  - Networks
- BUT Designing and Tuning, Large High Performance systems is NOT about tips and tricks, or fashion
- It is about good engineering practice

ORACLE



## Database Performance Basics

### Schemas and SQL Statements

- Good schema/data design is the foundation to good database performance.
- Good schema/data design makes downstream activities both simpler and less error prone
  - Selection of Index and Partitioning Columns
  - Gathering and maintenance of schema statistics
  - Writing correct SQL statements
  - Optimization of SQL statements
  - Avoidance of serialization points and transactional bottlenecks

ORACLE

## Scaling

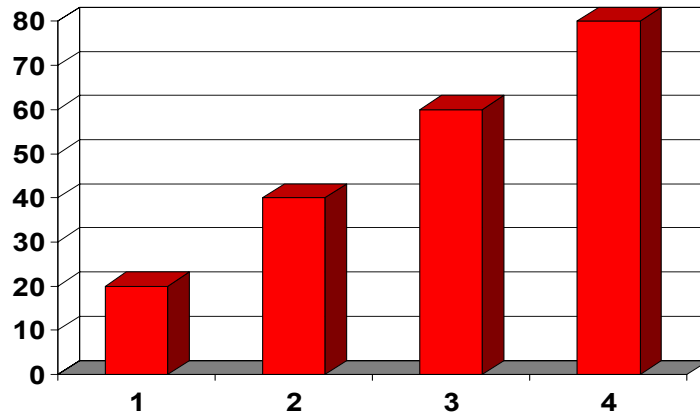
### What is it?

- ① Linear
- ② Unlimited
- ③ Maximum
- ④ Vertical
- ⑤ Horizontal
- ⑥ Unprecedented
- ⑦ Ultimate
- ⑧ Predictable
- ⑨ Diagonal

ORACLE

## Scalability Quiz

Label the axes ?



ORACLE

## Scaling

Some Possible Definitions

- When adding hardware resources to a system:
  - Number of transactions processed increases proportionately
  - Query times reduce proportionately
  - Query times remain constant when data volumes increase proportionately
- Hardware resources include
  - Nodes
  - CPU
  - Memory
  - Network
  - I/O

ORACLE

## What Prevents Scaling

### With respect to CPUs

- Scaling is severely impacted when the following scenarios arise
  - Hardware resources are operating beyond their practical maximum capacity for work
  - Software serialization takes place
    - Application e.g. Row lock contention
    - Database e.g. Latch contention
    - Operating System e.g. Process allocation

ORACLE

## Scaling the Oracle Database

### In theory

- Requirements when scaling the Oracle Database
  - Effective hardware capacity planning
    - A **balanced** design
    - Hardware components sized to absorb peak workload
  - Effective application architecture
    - All software components tested and validated
    - Safe working limits identified

ORACLE

## Scaling the Database

### In reality

- Common Application Scaling Issues
- Single Instance Scaling Issues
- Multiple Instance Scaling Issues

ORACLE

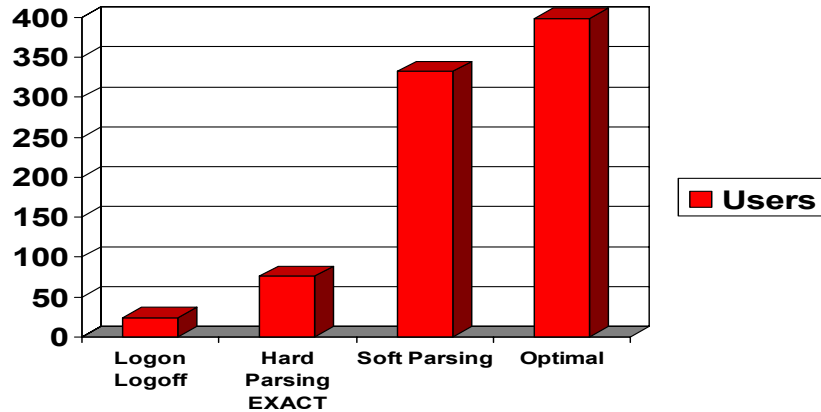
## Common Application Issues

- Incorrect use of Sessions
  - High connect/disconnect rates to the database
    - High load on O/S for process creation/teardown
    - High load mapping/unmapping SGA
    - CPU load on establishing database state
- Incorrect use of Cursors
  - Reparsing is best avoided
    - Hard parses use lots of CPU and will serialize
    - Soft parses scale better but they are not free

ORACLE

## Database Performance Basics

Sessions & Cursors(This slide is over 10 Years old!)



ORACLE

## Common Application Issues

- Poor SQL
  - Poor Access Paths
  - Excessive resource usage (CPU, I/O)
- Poor connection management
  - High numbers of connections can cause problems
  - Use connection pooling/concentration in the middle-tier
  - Use of Shared Servers is often just a band-aid

ORACLE

## Common Configuration Issues

- Non-default initialization parameters
- I/O problems
  - Poor use of available resources resulting in hot spots
    - Bandwidth limitations
    - Poor response times
  - Use ASM to eliminate unpredictable hot spots
- Space management
  - Use Automatic Undo Management
  - Use Auto Segment Storage Management

ORACLE

## Tuning Oracle

- Changes over time
  - Hardware
  - Software
  - Functionality
- BUT Designing and Tuning, Large High Performance systems is NOT about tips and tricks, or fashion
- It is about good engineering practice
- Correct diagnosis is key to successful tuning

ORACLE

## Oracle Tuning Methods: A History

- Prehistoric (v5)
  - Debug code
- Dark Ages (v6)
  - Counters/Ratios
  - BSTAT/ESTAT
  - SQL\*Trace
- Renaissance (v7/v8)
  - Introduction of Wait Event instrumentation
  - Move from counters to timers
  - STATSPACK
- Modernity (v10)
  - DB Time Tuning – Tuning using fundamental notion of time spent in database
  - Multiple scoping levels
  - Always on, non-intrusive
  - Built into infrastructure: instrumentation, ASH, AWR, ADDM, EM

ORACLE

## Why Do We Care About Time?

- Human time is critical to the enterprise
- Systems performance affects business goals
  - Human time + technology resource time
- “Time is money”
- Performance improvement means doing things faster

*Performance is always and only about time*

ORACLE

## Database Time and Average Active Sessions



ORACLE

## Database Time (DB Time)

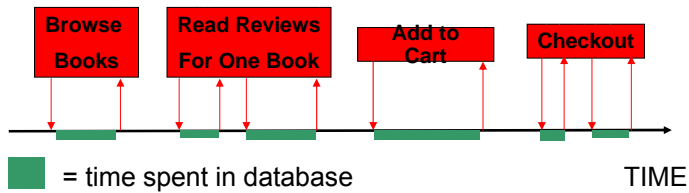
- Total time in database calls by **foreground sessions**
- Includes **CPU** time, **IO** time and **non-idle wait** time
- DB Time <> response time
- **New** lingua franca for Oracle performance analysis

***Database time is total time spent by user processes either actively working or actively waiting in a database call.***

ORACLE

## A Single Session

Single session with Database Black Box server



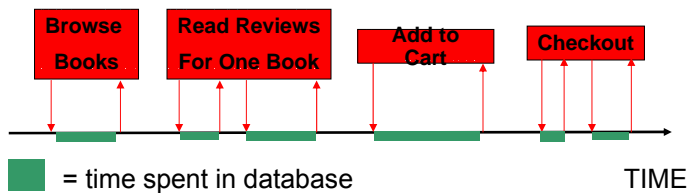
ORACLE

## Fundamental Concepts

**Database Time (DB Time) =**  
Total time session spent in all database calls

**Active Session =**  
Session currently spending time in a database call

**Average Activity of the Session (% Activity) =**  
The ratio of time active to total wall clock time



ORACLE

Oracle Enterprise Manager (GRAHAM.WOOD) - Session Details: 1869 (AFOTHERG) - Microsoft Internet Explorer

Address: http://em1.oracle.com/em/console/database/instance/sessionDetails?SID=1869&serialNum=16111&pageNum=2&leftEdge=1193752570150&rightEdge=1193754370150&direction=0&selectedE

ORACLE Enterprise Manager 10g

Home | Targets | Deployments | Alerts | Setup | Preferences | Help | Logout

Hosts | Databases | Application Servers | Web Applications | Groups | All Targets | Collaboration Suites

Cluster: dbs232\_crs > Cluster Database: BUGAP.US.ORACLE.COM > Top Sessions > Database Instance: BUG1AP\_DBS232 > Top Activity > Logged in As JSARICOS

Session Details: 1869 (AFOTHERG)

Collected From Target Oct 30, 2007 9:49:37 AM CDT View Data | Real Time: 15 Second Refresh | Refresh

Kill Session | Enable SQL Trace

General | Activity | Statistics | Open Cursors | Blocking Tree | Wait Event History

Drag the shaded box to change the time period for the detail section below.

Detail for Selected 5 Minute Interval

Start Time Oct 30, 2007 9:17:05 AM View | Show Aggregated Data | Run ASH Report

Activity (%)	SOL ID	SOL Command	Plan Hash Value	Module	Action	Client ID
100.00	gkmd7xwuz1na0	SELECT	64730335	oraclealan@ap103fam (TNS_V1-V3)	AFOTHERG	

General | Activity | Statistics | Open Cursors | Blocking Tree | Wait Event History

Kill Session | Enable SQL Trace

Home | Targets | Deployments | Alerts | Compliance | Jobs | Reports | Setup | Preferences | Help | Logout

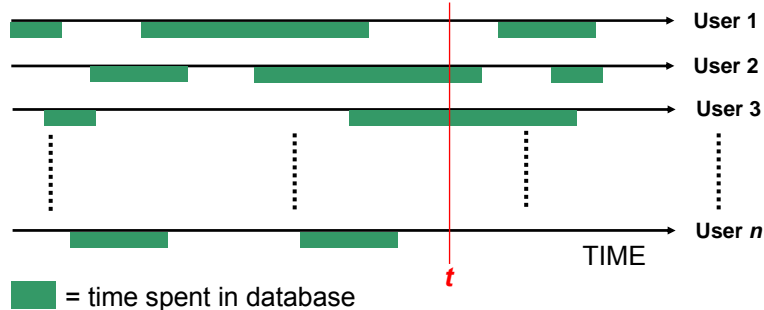
Copyright © 1996, 2007, Oracle. All rights reserved.  
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

## Multiple Sessions

DB Time = Sum of DB Time Over All Sessions

Avg. Active Sessions = Sum of Avg. Activity Over All Sessions

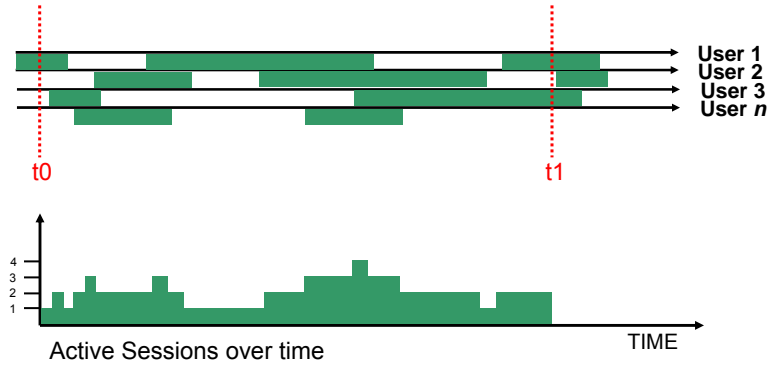
At time  $t$  we have 2 active sessions



ORACLE

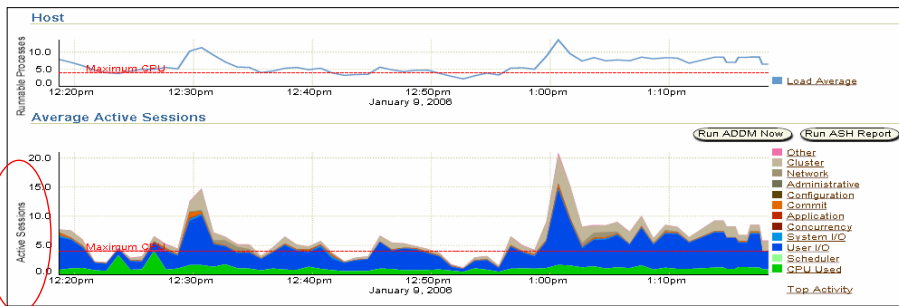
# Visualizing DB Time

$$\text{Avg. Active Sessions} = \frac{\text{Total Database Time}}{\text{Wall Clock (Elapsed) Time}}$$



ORACLE

# EM Performance Page



- Active Sessions by wait class over time
- Colored area = amount of DB time
- “Click on the big stuff”

ORACLE

## Where to find DB Time?

- **V\$SYS\_TIME\_MODEL, V\$SESS\_TIME\_MODEL**
  - STAT\_NAME = 'DB time'
- **V\$SYSMETRIC\_HISTORY**
  - "Database Time Per Second", "CPU Usage Per Sec"
  - 10g units = centi-secs/sec (100xAvg. Active Sessions)
  - 11g new metric "Average Active Sessions"
- **V\$SQL**
  - ELAPSED\_TIME and CPU\_TIME
  - Wait class times:  
APPLICATION, CONCURRENCY, CLUSTER, USER\_IO
- **V\$ACTIVE\_SESSION\_HISTORY**

ORACLE

## Active Session History



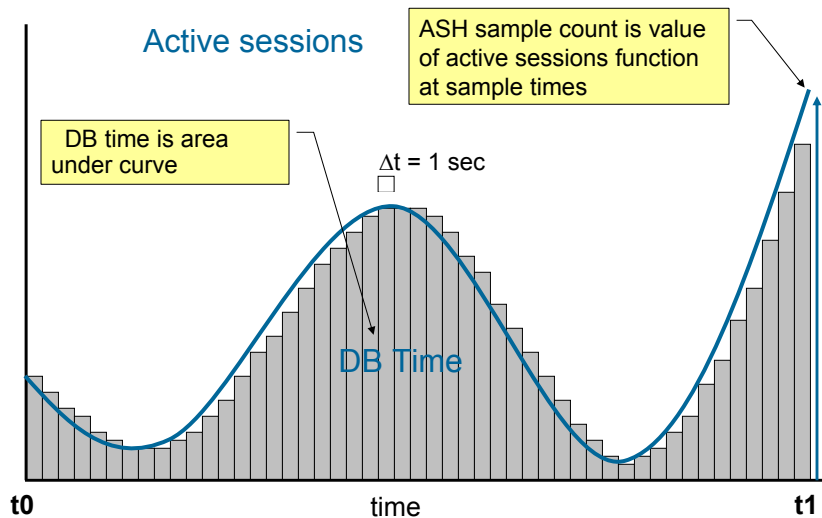
ORACLE

## Active Session History (ASH)

- All 'Active' sessions captured every second
  - Foregrounds and backgrounds are sampled
  - Active foregrounds contribute to DB Time
- In-memory: V\$ACTIVE\_SESSION\_HISTORY
  - Sampling interval = 1 second
- On-disk: DBA\_HIST\_ACTIVE\_SESS\_HISTORY
  - Sampling interval = 10 second
- ASH is a system-wide record of database activity

ORACLE

## Active Sessions and DB Time



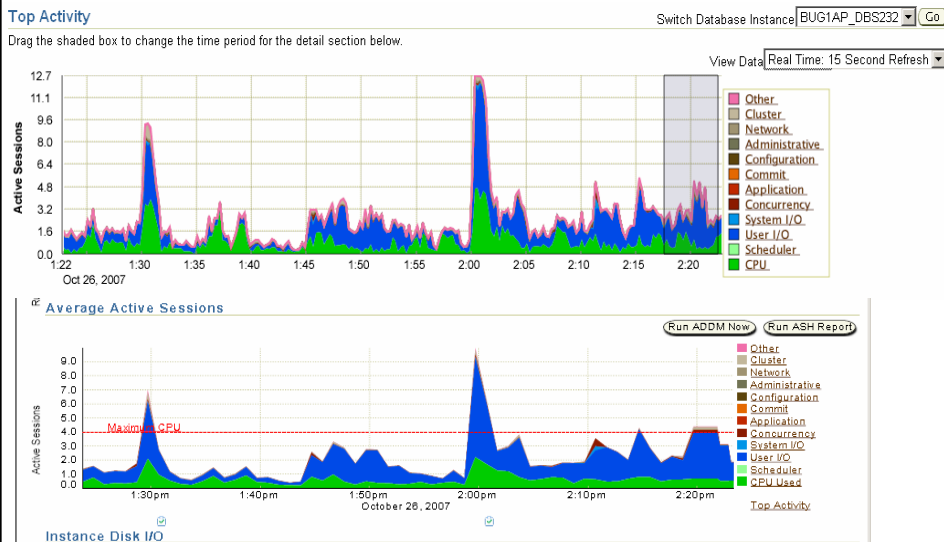
ORACLE

# Estimating DB Time with ASH

- **ASH sample counts = DB Time** in seconds
  - Low sample counts are less reliable
- Enables DB Time analysis over many dimensions
  - Sqlid, session id, instance, service, module, action
  - 10gR2
    - Blocking\_sid (10gR2)
    - XID
  - 11g
    - Row source
    - Execution ID
    - Operation type
      - Connect
      - Java/SQL/PLSQL
      - parse, bind, execute/fetch, close

ORACLE

# DB Time: ASH vs Time Model



## Where is DB Time used?

- ADDM
- EM Performance page and drill downs
- ASH report
- AWR and AWR compare periods reports
- SYSMETRICS and Server-generated Alerts

ORACLE

## Techniques: The DB Time Method



ORACLE

## The DB Time Method: Short Course

*or*  
***just ask ADDM***

ORACLE

## The DB Time Method: Process

1. Identify performance issue
2. Scope the issue
3. Set goals
4. Data capture (NO OP)
5. Investigate DB time distribution
  - Identify the largest potential for improvement
6. Modify system to tune for largest gain
7. Evaluate against goals
  - Repeat from step 4 if goals not met

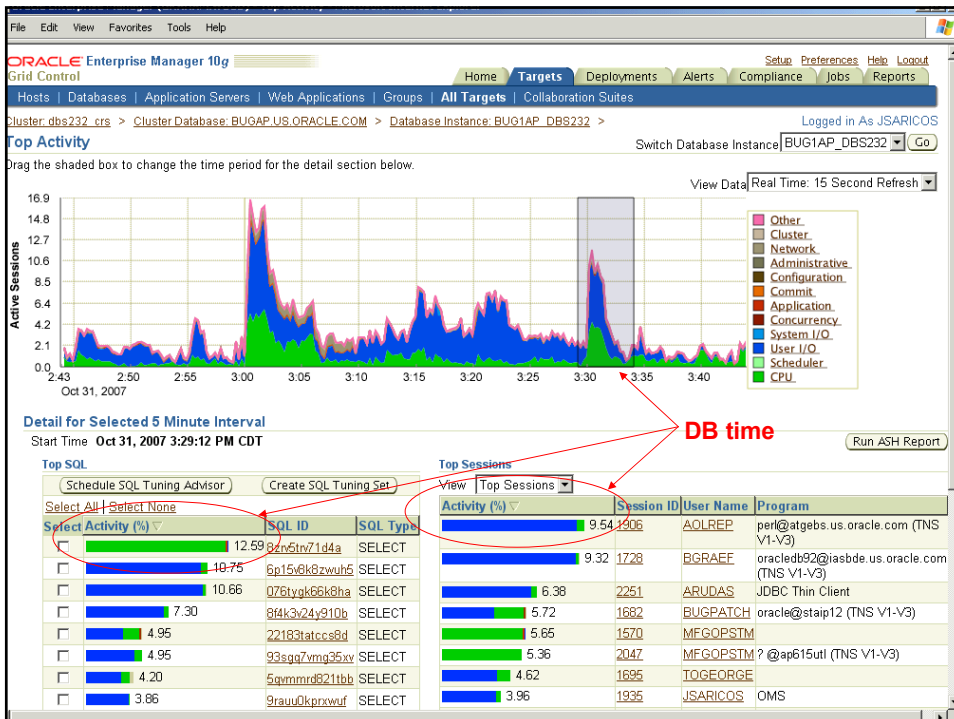
***Performance tuning by removing excess DB time***

ORACLE

# Investigate DB Time Distribution

- Identify uneven distributions of DB time (skew)
  - => Largest potential improvement within scope
- System scope:
  - Resource limits – is problem outside the DB?
- Application scope:
  - Service, module, action
  - Resource contention (e.g. latches)
  - SQLID, rowsource
- Session scope:
  - Long running SQL
  - Resource contention (e.g. enqueues)

ORACLE



## Identify Potential Solutions

- Session contention issues
  - Kill session
  - Fix application
- SQL issues
  - SQL Tuning Advisor => Indexes, SQL profile
  - Re-write SQL
- Design issues
  - Access Advisor => Indexes, physical layout
- System issues
  - Initialization parameters
  - Add resources

ORACLE

## Modify System

- Start with the largest DB time issues first
  - Address root causes, not symptoms
- Match solution scope to problem scope
  - Don't tweak optimizer parameters before tuning SQL
- Proceed iteratively one fix at a time
  - Concurrent fixes should be orthogonal
- Measure and validate results at each successive step
- Stop when goals are met

ORACLE

## The DB Time Method: Advantages

- Tunes the one thing that affects users: Time
- Data capture scoping not necessary
  - 'Always on' data collection
  - No requirement to reproduce problem
- Works for concurrency problems such as locking
- Combines best of current methods
  - Less intrusive, more inclusive

ORACLE

## Method Summary

- DB time is the fundamental performance metric
- The method allows DB time analysis at many scopes
  - Proper scoping of problems and solutions is critical to success
- DB time based diagnosis removes value judgments
  - Scientific method, not sorcerer's magic
- **Performance improvement means doing the same work in less DB Time**

ORACLE

## Tools:

**ADDM**

**Enterprise Manager**

**Reports**



ORACLE

## Tools for Applying DB Time Method

Two use-cases, one method:

### 1. Tuning steady-state performance

- Improve overall workload throughput or response time
- Best practice: use ADDM

### 2. Diagnosing transient performance problems

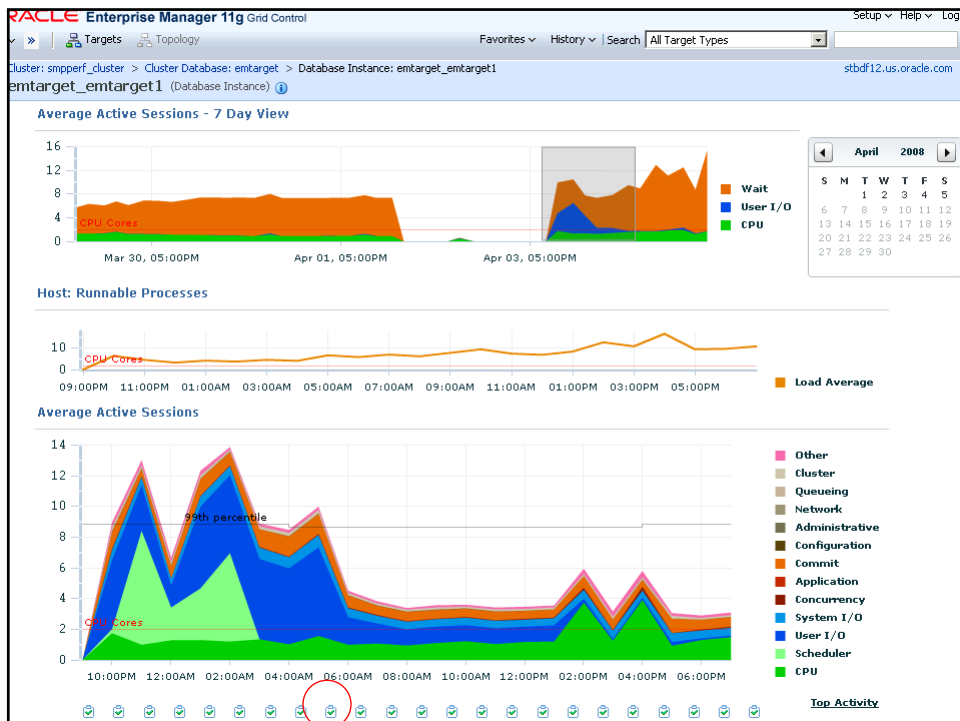
- Confirm and investigate reported performance issues
- Best practice: use EM real-time screens

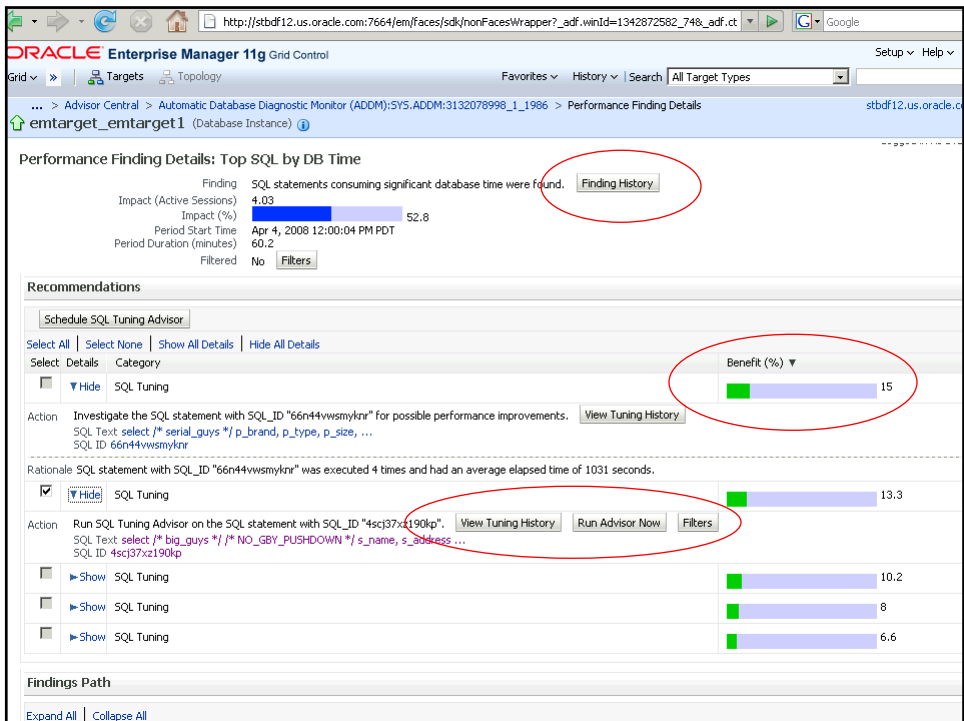
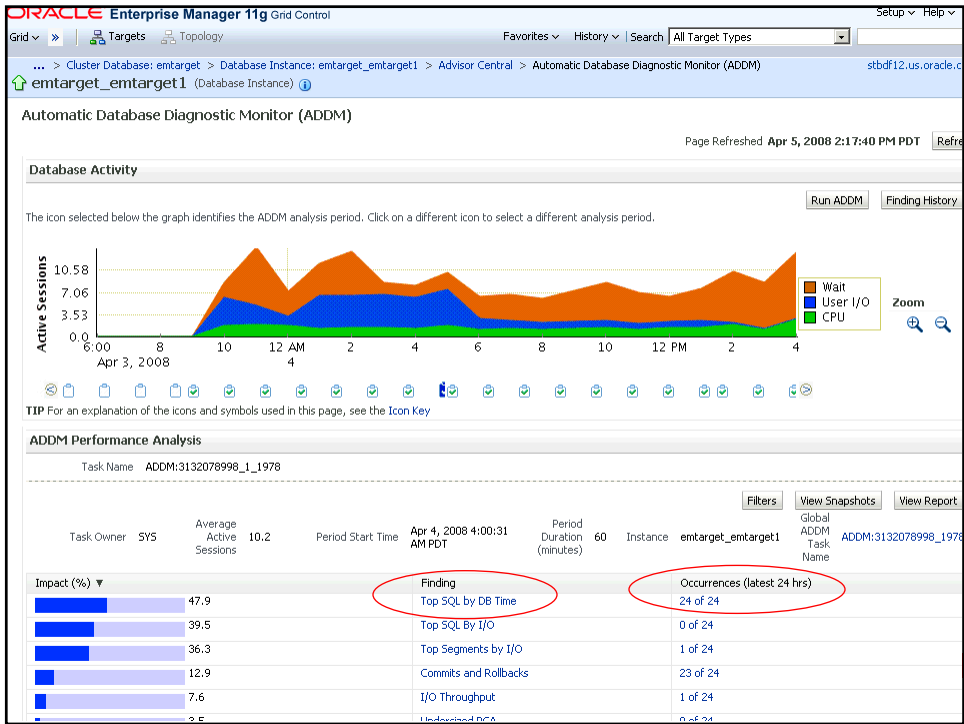
ORACLE

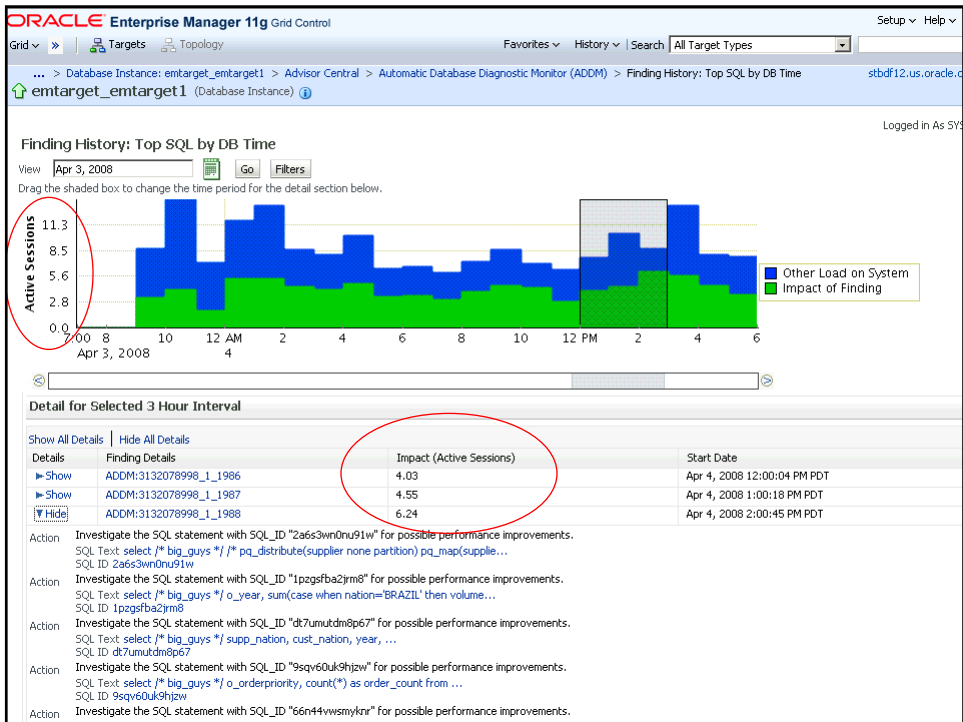
# Best Practice: Use ADDM

- Embedded expert system using the DB time method
  - Identifies root causes behind the symptoms
- Variably scoped:
  - Host to instance to SQL and even database block
  - Scoped to database for RAC (new in 11g)
- Findings prioritized by impact on DB time
  - Finding history allows flexible time scoping
  - Directives can filter findings
- Recommendations by benefit (reduction) to DB time

ORACLE

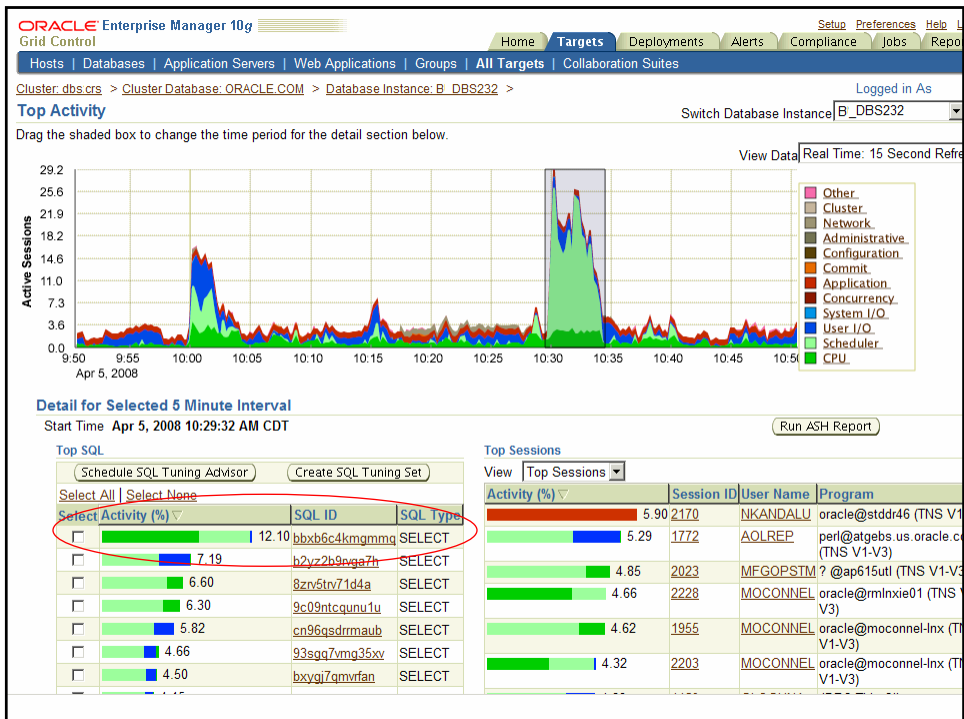
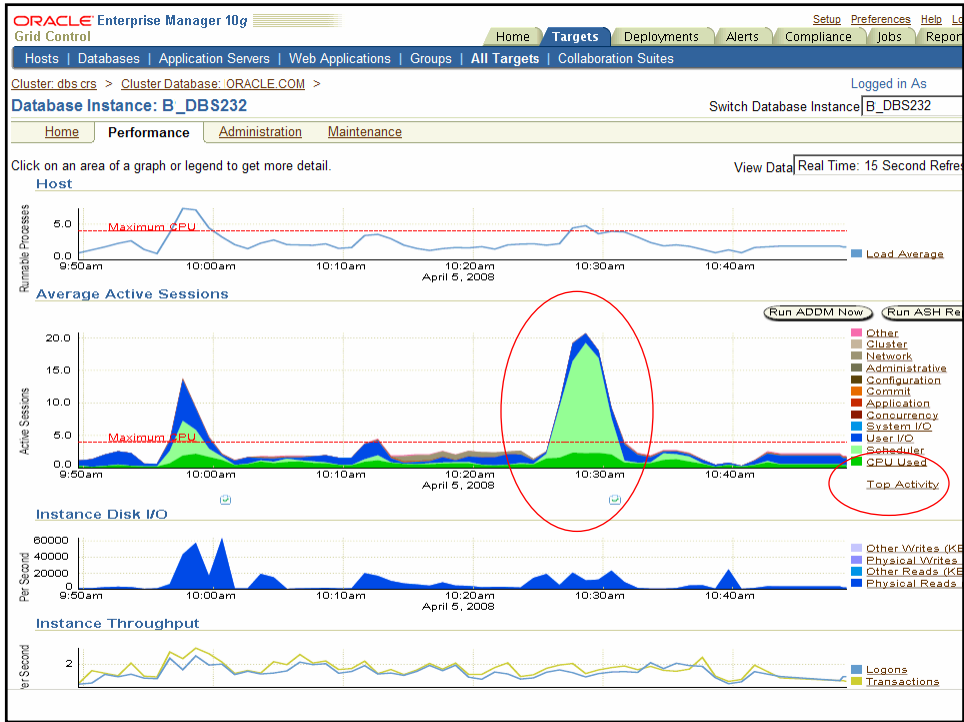






## Best Practice: EM Real-time Interface

- Transient (sub-hour) or immediate time scope
  - Requires interactivity of UI
- 'Click on the big stuff'
  - Data visualizations display skew directly
- Takes some expertise to separate symptoms from root causes



ORACLE Enterprise Manager 10g  
Grid Control

Home Targets Deployments Alerts Compliance Jobs Reports

Hosts | Databases | Application Servers | Web Applications | Groups | All Targets | Collaboration Suites

Cluster: dbcs.crs > Cluster Database: ORACLE.COM > Database Instance: B DBS232 > Top Activity > Logged in As

SQL Details: bxb6c4kmgmmq

Switch to SQL ID:  (Go) View Data: Real Time: Manual Refresh Refresh Schedule SQL Tuning Advisor

Text

```
SELECT /*+ OPAQUE_TRANSFORM */
"RPINO", "RPIDATE", "RPID_BY", "VERSION", "UTILITY_VERSION", "CATEGORY", "STATUS", "SUBJECT", "UPD_BY", "CUSTOMER"
FROM "BG"."RPHEAD" "H" WHERE "RPIDATE">:1 AND "RPID_BY"<>'BATCH' AND "CUSTOMER" LIKE '%WPTG%' AND
```

Details

Select the plan hash value to see the details below. Plan Hash Value: 301316116

Statistics Activity **Plan** Tuning Information

Summary

Drag the shaded box to change the time period for the detail section below.

Active Sessions

2.0  
1.6  
1.2  
0.8  
0.4  
0.0

9:51 9:55 10:00 10:05 10:10 10:15 10:20 10:25 10:30 10:35 10:40 10:45 10:50  
Apr 5, 2008

Legend: resmgr:cpu quantum (yellow), db file sequential read (blue), CPU Used (green)

Detail for Selected 5 Minute Interval

Start Time: Apr 5, 2008 10:29:32 AM Run ASH Report

Activity (%)	SID	User	Program	Service	Plan Hash Value
51.89	2228	MOCONNEL	oracle@mlnxie01 (TNS V1-V3)	boracle.com	301316116
48.11	2203	MOCONNEL	oracle@moconnel-lnx (TNS V1-V3)	boracle.com	301316116

## Conclusions

- Design of Large, High Performance Systems is engineering, not a mystical art
- Good database design leads to
  - Simpler implementation
  - Scalability
- Oracle database instrumentation
  - Simpler performance diagnosis
  - Increases probability of success



## Time for Your Questions



ORACLE