ORACLE®

Using Automatic Workload Repository for Database Tuning: Tips for Expert DBAs

Kurt Engeleiter
Product Manager

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.

Oracle Enterprise Manager
Top-Down, Integrated Application Management

- Complete, Open, Integrated Management for Oracle Technologies
  - Deep, Optimized, Best of Breed
  - Database, Middleware, Packaged Applications, Physical and Virtual Infrastructure

- Business Centric, Top Down Application Management

- Complete Lifecycle Management

- Scalable Grid and Cloud Management
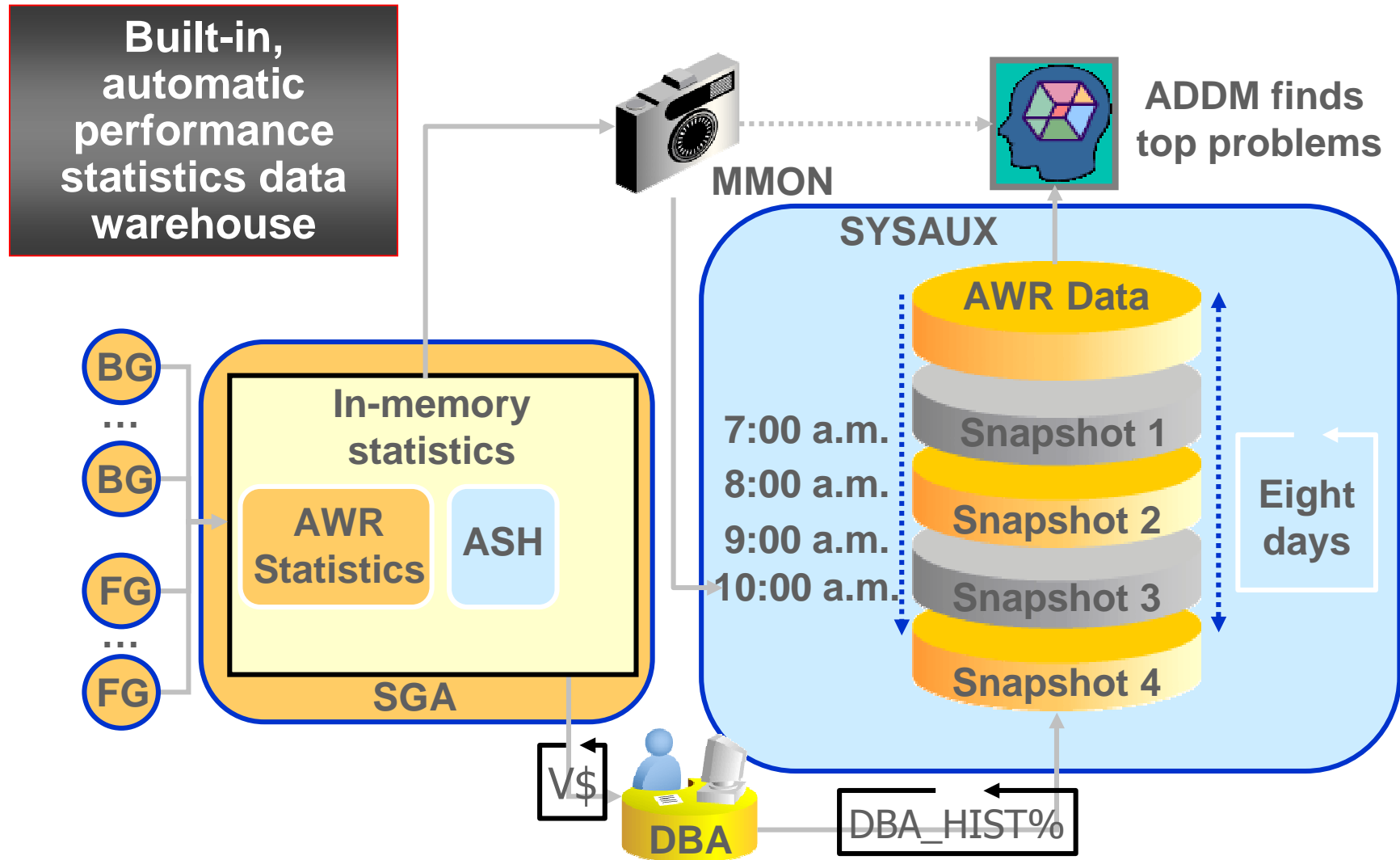 - Manage many as one
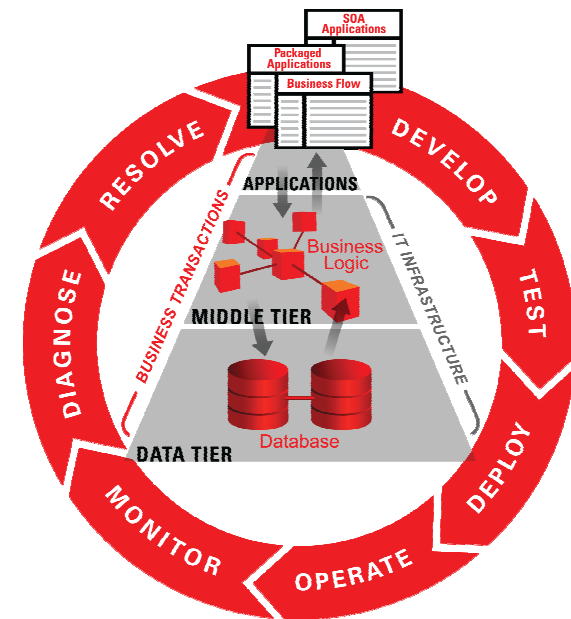
ORACLE®

Agenda

- Automatic Workload Repository Infrastructure

- Out of the box AWR reports

- Active Reports

Automatic Workload Repository (AWR)

Built-in, automatic performance statistics data warehouse

MMON

ADDM finds top problems

SYSAUX

In-memory statistics

AWR Statistics   ASH

SGA

BG
...
BG

FG
...
FG

AWR Data

7:00 a.m.   Snapshot 1
8:00 a.m.   Snapshot 2
9:00 a.m.   Snapshot 3
10:00 a.m.   Snapshot 4

Eight days

V$

DBA

DBA_HIST%

Fundamental Concepts

- **DB Time**
  - Total time in database calls by foreground sessions
  - Includes CPU time, IO time and non-idle wait time
  - DB Time <> response time
  - Total DB time = sum of DB time for all active sessions

➢**Goal: To Reduce Total DB time**

- Active Session
  - Session currently spending time in a database call, i.e., accruing DB time

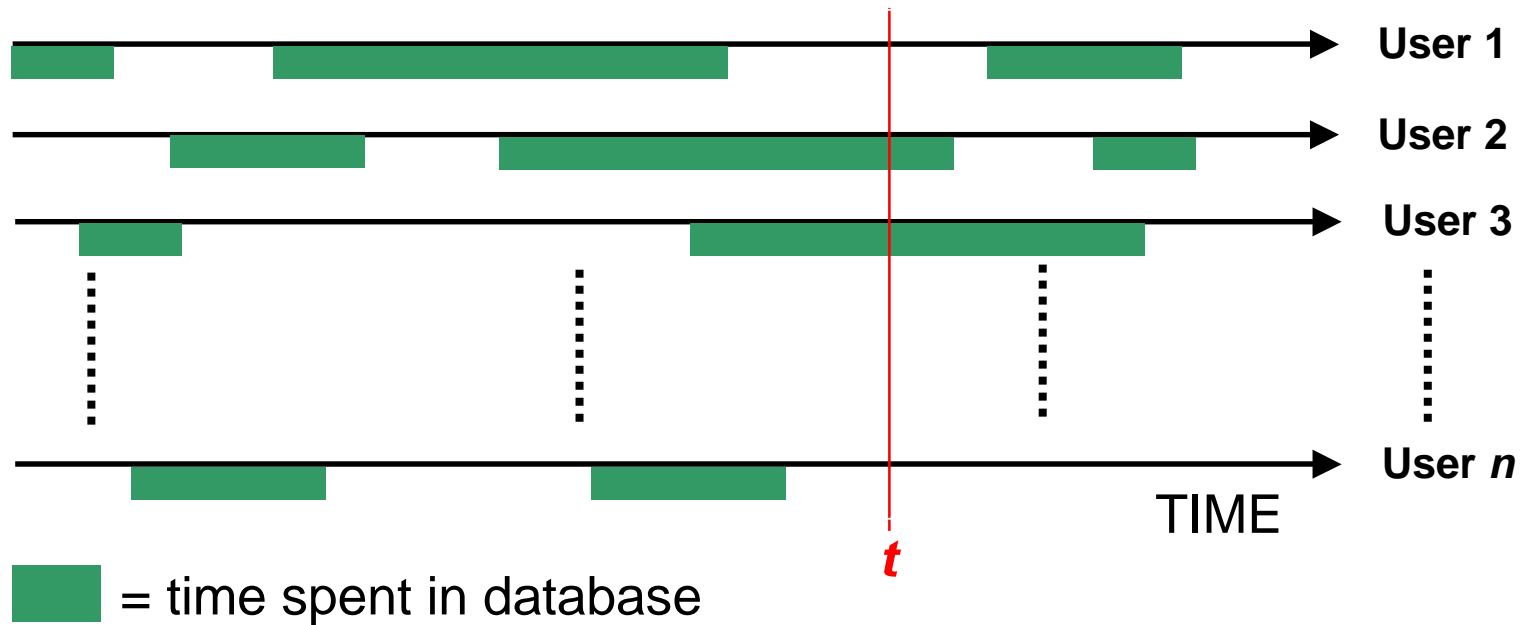- **Average Active Sessions** $= \dfrac{\text{DB Time}}{\text{Wall-Clock (Elapsed) Time}}$

- Average Active Sessions is a key metric for measuring DB load

Multiple Sessions

Total DB Time = Sum of DB Time Over All Sessions

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

$$= \frac{\text{Total Database Time}}{\text{Wall-Clock (Elapsed) Time}}$$

At time t we have 2 active sessions
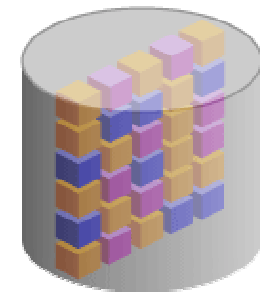

= time spent in database

AWR Data

- Time Model
  - v$sys_time_model => dba_hist_sys_time_model
  - DB Time
  - Automatic Tracking of Operation Times
    - Overall parse time (hard, soft, failed,..)
    - SQL, PLSQL and Java overall execution times
- Wait Model
  - v$system_event => dba_hist_system_event
  - Wait Events Categorized Based On Solution Area
  - More than 900 different wait events. 12 wait classes (Application, Concurrency..)
- SQL statement statistics
  - v$sqlstats => dba_hist_sqlstat
  - Resource Usage: Executions, Physical Reads, Physical Writes
  - Efficient Top SQL identification using deltas in the kernel
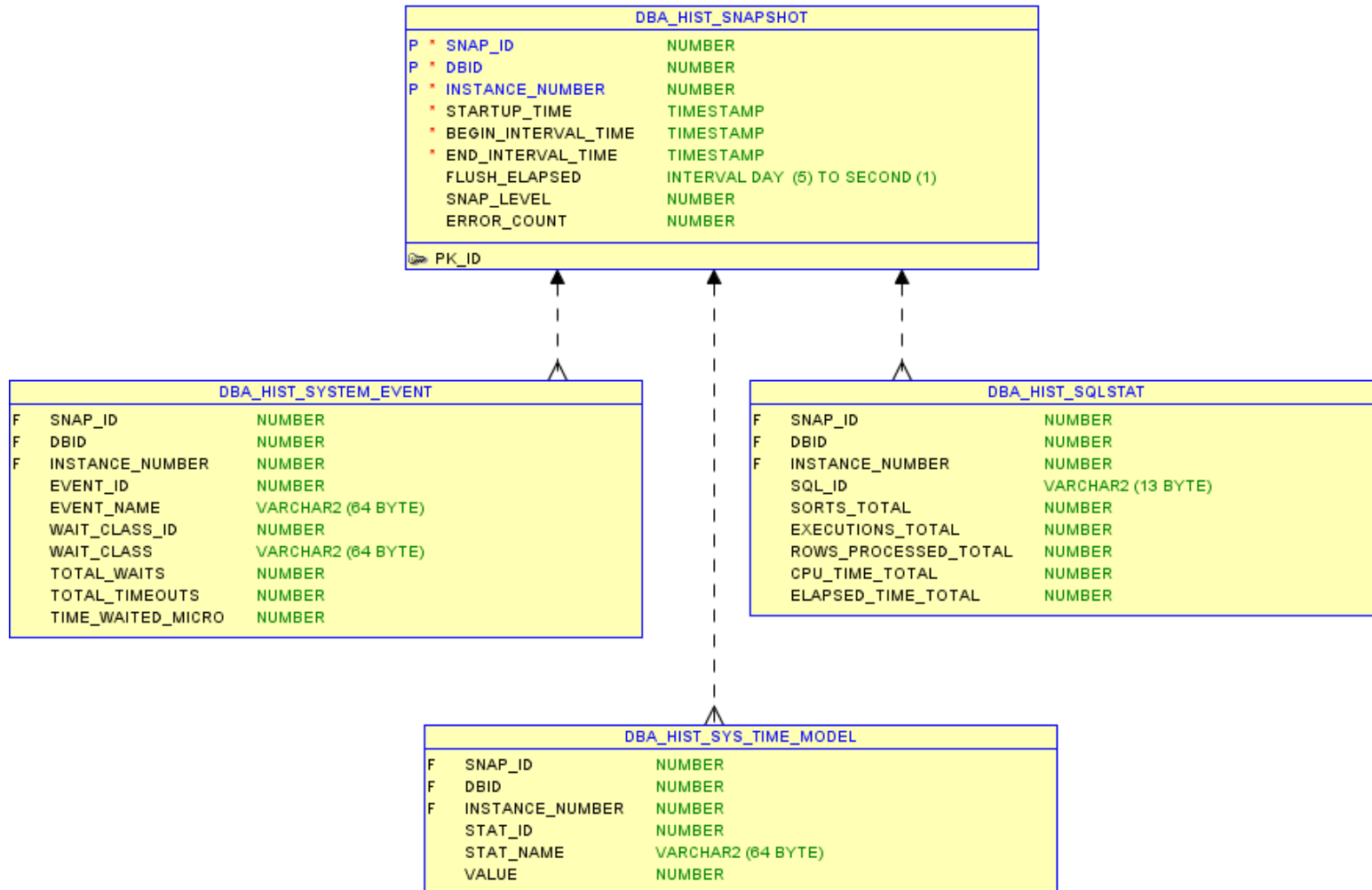- OS Stats
  - v$osstat => dba_hist_osstat
  - CPU + Memory

Commit **CPU**

Application

User I/O
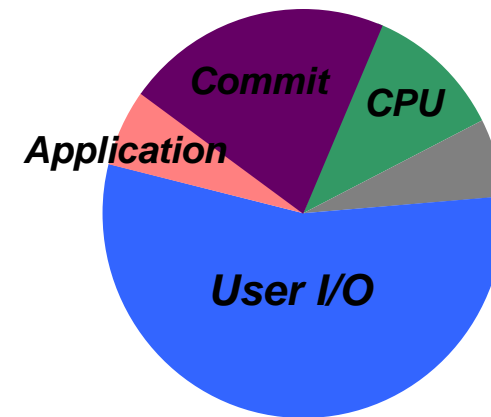
AWR Data

- Snapshots
  - DBA_HIST_SNAPSHOT

    - Tracks Snapshots in the AWR

    - When querying AWR, always join to other tables to constrain the time frame

AWR Data

DBA_HIST_SNAPSHOT

P	*	SNAP_ID	NUMBER
P	*	DBID	NUMBER
P	*	INSTANCE_NUMBER	NUMBER
	*	STARTUP_TIME	TIMESTAMP
	*	BEGIN_INTERVAL_TIME	TIMESTAMP
	*	END_INTERVAL_TIME	TIMESTAMP
		FLUSH_ELAPSED	INTERVAL DAY (5) TO SECOND (1)
		SNAP_LEVEL	NUMBER
		ERROR_COUNT	NUMBER

PK_ID

DBA_HIST_SYSTEM_EVENT

F	SNAP_ID	NUMBER
F	DBID	NUMBER
F	INSTANCE_NUMBER	NUMBER
	EVENT_ID	NUMBER
	EVENT_NAME	VARCHAR2 (64 BYTE)
	WAIT_CLASS_ID	NUMBER
	WAIT_CLASS	VARCHAR2 (64 BYTE)
	TOTAL_WAITS	NUMBER
	TOTAL_TIMEOUTS	NUMBER
	TIME_WAITED_MICRO	NUMBER

DBA_HIST_SQLSTAT

F	SNAP_ID	NUMBER
F	DBID	NUMBER
F	INSTANCE_NUMBER	NUMBER
	SQL_ID	VARCHAR2 (13 BYTE)
	SORTS_TOTAL	NUMBER
	EXECUTIONS_TOTAL	NUMBER
	ROWS_PROCESSED_TOTAL	NUMBER
	CPU_TIME_TOTAL	NUMBER
	ELAPSED_TIME_TOTAL	NUMBER

DBA_HIST_SYS_TIME_MODEL

|		
F	SNAP_ID	NUMBER
F	DBID	NUMBER
F	INSTANCE_NUMBER	NUMBER
	STAT_ID	NUMBER
	STAT_NAME	VARCHAR2 (64 BYTE)
	VALUE	NUMBER

Active Session History (ASH)

- ASH is session level data

- Active sessions are sampled and persisted in-memory
  - Sampling interval = 1 second
  - V$ACTIVE_SESSION_HISTORY
  - Foreground and background sessions are sampled

- On-disk persistence
  - DBA_HIST_ACTIVE_SESS_HISTORY

- ASH is a system-wide trace of what happened

- ASH is a many-dimensional FACT table
  - Dimensions are V$SESSION columns
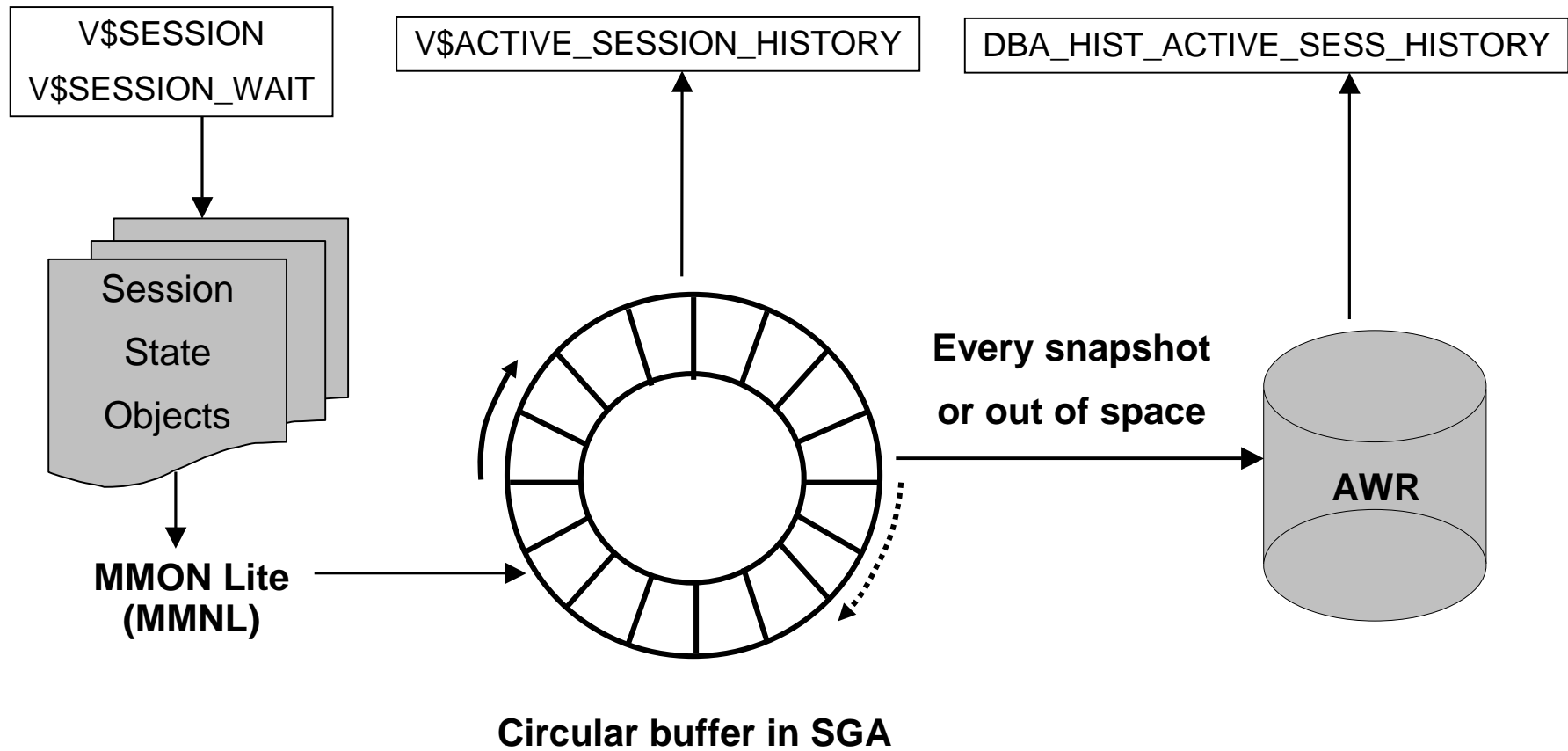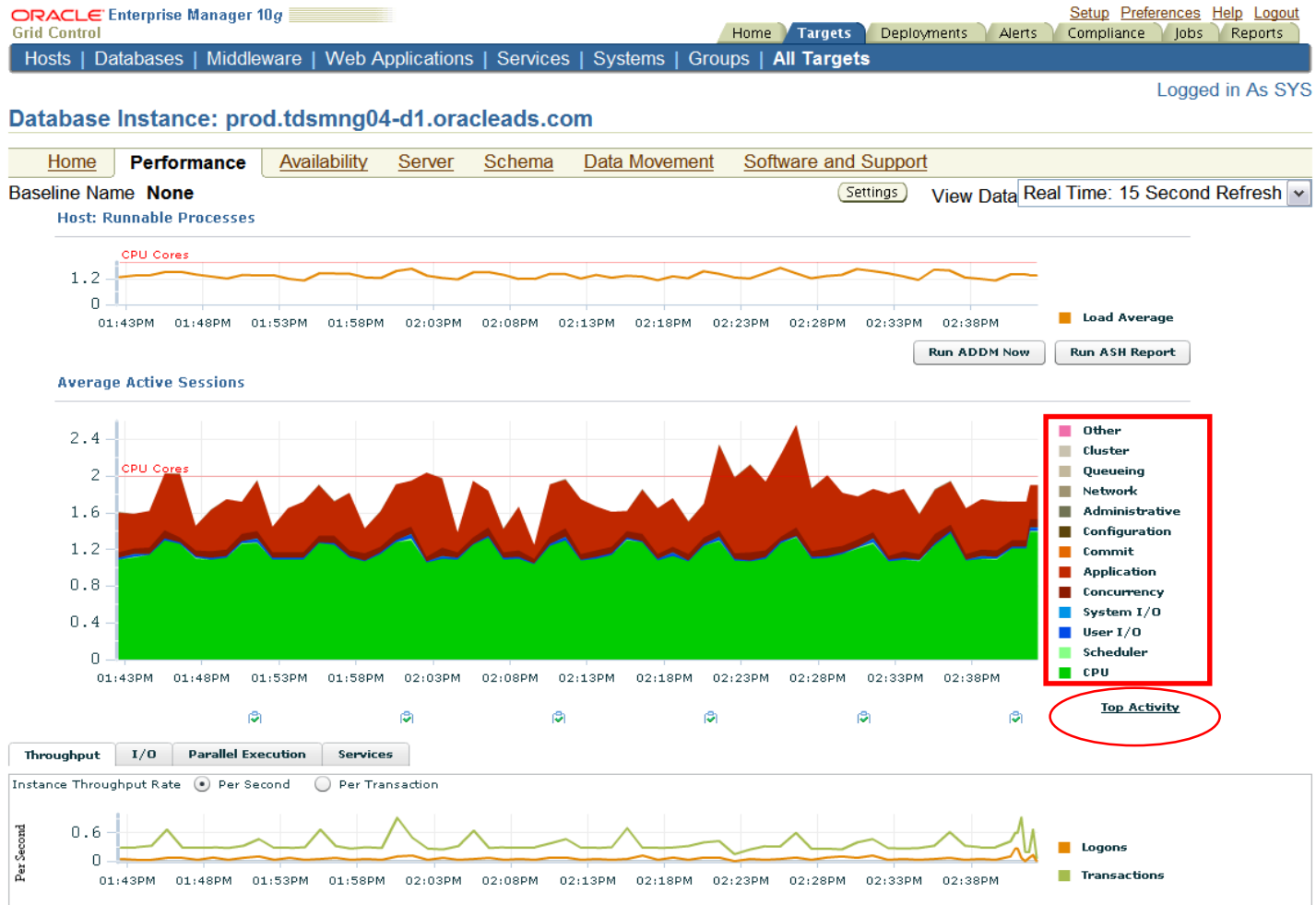 - Fact is that DB time was accumulating over these dimensions

Active Session History (ASH)
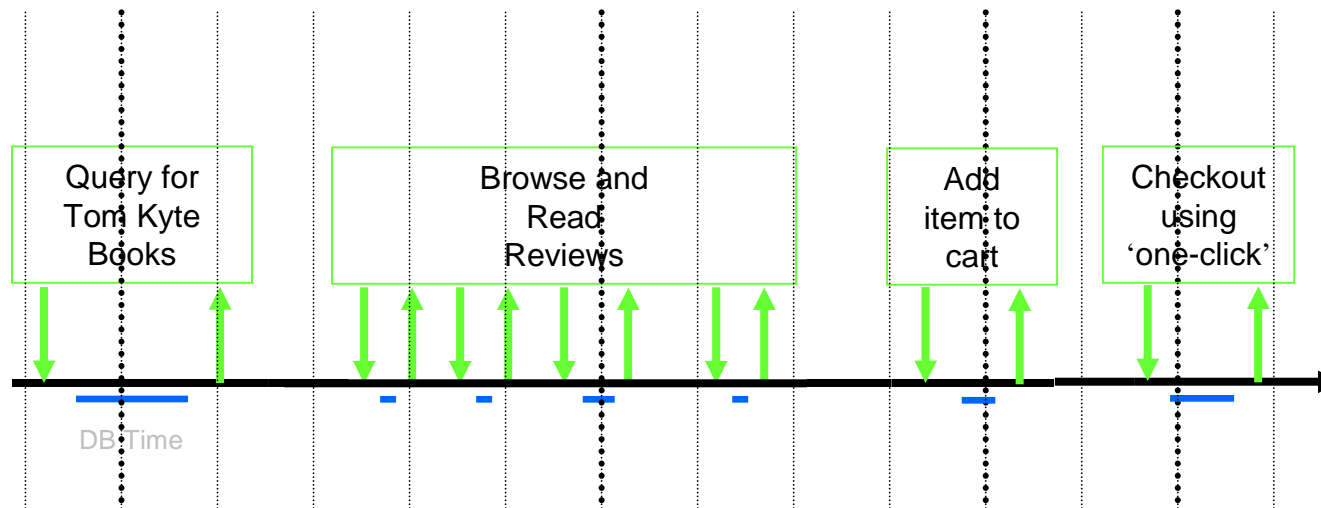
Time	SID	Module	SQL ID	State	Event
7:38:26	213	Book by author	qa324jffritcf	WAITING	db file sequential read
7:42:35	213	Get review id	aferv5desfzs5	CPU	
7:50:59	213	Add to cart	hk32pekfcbdfr	WAITING	buffer busy wait
7:52:33	213	One click	abngldf95f4de	WAITING	log file sync

ORACLE

ASH

V$SESSION
V$SESSION_WAIT

V$ACTIVE_SESSION_HISTORY

DBA_HIST_ACTIVE_SESS_HISTORY

Session

State

Objects

**MMON Lite
(MMNL)**

**Every snapshot
or out of space**

AWR

Circular buffer in SGA

AWR and ASH in Enterprise Manager

Top Activity

SQL Details

Agenda

- Automatic Workload Repository Infrastructure

- Out of the box AWR reports

- Active Reports

AWR Reporting Resources

- Enterprise Manager is the preferred way to view and analyze AWR and ASH data

- In addition, predefined AWR html reports are provided in each Oracle database release

- Each report has a specific function and use case

- The following slides show the major reports and their use cases

ORACLE

AWR Reporting Resources

- Available in
  - Enterprise Manager
  - $ORACLE_HOME/rdbms/admin
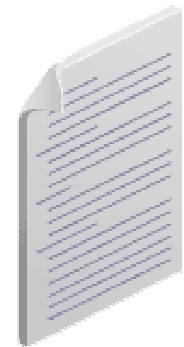
REPORT NAME	SQL Script
Automatic Workload Repository Report	awrrpt.sql
Automatic Database Diagnostics Monitor Report	addmrpt.sql
ASH Report	ashrpt.sql
AWR Diff Periods Report	awrddrpt.sql
AWR Single SQL Statement Report	awrsqrpt.sql
AWR Global Report	awrgrpt.sql
AWR Global Diff Report	awrgdrpt.sql

11gR2

How do I read an AWR report?

AWR Report

- The AWR report is the most well known performance report.

- Oracle tuning professionals frequently start their analysis with this report.

- AWR report contains much data – but contains no concrete recommendations for action.

WORKLOAD REPOSITORY report for

DB Name	DB Id	Instance	Inst num	Startup Time	Release	RAC
PROD	48067057	prod	1	27-Aug-09 18:08	11.1.0.7.0	NO

Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
tdsmng03-d1	Linux IA (32-bit)	4	2	2	5.82

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	26324	29-Aug-09 20:20:41	38	3.2
End Snap:	26325	29-Aug-09 20:30:43	36	2.7
Elapsed:		10.04 (mins)		
DB Time:		29.99 (mins)		

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	3.0	8.7	0.03	0.02
DB CPU(s):	2.4	6.9	0.02	0.01
Redo size:	5,201.5	15,206.5		
Logical reads:	127,103.4	371,583.6		
Block changes:	22.1	64.6		
Physical reads:	1,032.8	3,019.3		
Physical writes:	1.8	5.3		
User calls:	184.6	539.6		
Parses:	72.7	212.5		
Hard parses:	51.5	150.6		
W/A MB processed:	879,631.8	2,571,582.1		
Logons:	0.1	0.2		
Executes:	115.4	337.2		
Rollbacks:	0.2	0.5		
Transactions:	0.3			

Tip: Check ADDM Report When Viewing AWR Report

- When viewing AWR report, always check corresponding ADDM report for actionable recommendations
- ADDM is a self diagnostic engine designed from the experience of Oracle's best tuning experts
- Analyzes AWR data automatically after an AWR snapshot
- Makes specific performance recommendations
- Consistent – never has a 'bad' day
- ADDM also tells you what is NOT a problem

ADDM Report

- ADDM lists the tuning opportunities with the highest benefit.

- ADDM makes specific, actionable recommendations.

- ADDM also lists areas of the system that are performing well – that don't need tuning.

```
            ADDM Report for Task 'TASK_51637'
            ---------------------------------


Summary of Findings
-------------------
   Description                       Active Sessions      Recommendations
                                     Percent of Activity

   -------------------------------   -------------------  ----------------
1  CPU Usage                         2.11 | 75.33         3
2  Top SQL by DB Time                1.47 | 52.57         3
3  Hard Parse Due to Literal Usage   .95 | 34.06          1
4  Row Lock Waits                    .45 | 16             1


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Miscellaneous Information
-------------------------
Wait class "Commit" was not consuming significant database time.
Wait class "Concurrency" was not consuming significant database time.
Wait class "Configuration" was not consuming significant database time.
Wait class "Network" was not consuming significant database time.
Wait class "User I/O" was not consuming significant database time.
Session connect and disconnect calls were not consuming significant database
time.
```

ORACLE

ADDM Impact Breakdown

- In Oracle Database 11g Release 2, ADDM can break down the impact of it's findings by several dimensions including service, and session.
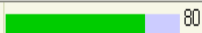
Performance Finding Details: CPU Usage

Finding: Host CPU was a bottleneck and the instance was consuming 96% of the host CPU. All wait times will be inflated by wait for CPU.
[Finding History]

Impact (Active Sessions): 2.97

Percentage of Finding's Impact (%): 35.1

Period Start Time: Oct 7, 2009 4:00:53 PM PDT

Period Duration (minutes): 59.3

Filtered: No [Filters]

Recommendations

Show All Details | Hide All Details

Details	Category	Benefit (%) ▽
▼Hide	Application Analysis	80
Action	Look at the "Top SQL Statements" finding for SQL statements consuming significant time on CPU. For example, the SELECT statement with SQL_ID "5n64ws275g2uq" is responsible for 60% of CPU usage during the analysis period.	
▶Show	Host Configuration	35.1

Finding Impact Breakdown

Category [Top Services ▼]

Top Services

Impact (% of Total Activity) ▽	Service
98.48	orcl.us.oracle.com
1.30	SYS$USERS
.21	SYS$BACKGROUND

Total Sample Count: 2,376

ORACLE®

AWR Global Report - RAC

- RAC AWR Report

- Report rewritten and renamed in Oracle Database 11.2. spawrrac.sql => awrgrpt.sql

- All statistics from AWR placed in comparative format, along with sums, averages and standard deviations, making it easy to compare performance of RAC nodes.

WORKLOAD REPOSITORY REPORT (RAC)

Database Instances Included In Report

- Listed in order of instance number, I#

I#	Instance	Host	Startup	Begin Snap Time	End Snap Time	Release	Elapsed Time(min)	DB time(min)	Up Time(hrs)
1	bug1ap	dbs232	18-Jul-09 01:44	20-Jul-09 07:00	20-Jul-09 10:30	11.2.0.1.0	210.17	1,630.72	56.77
2	bug2ap	dbs233	18-Jul-09 01:45	20-Jul-09 07:00	20-Jul-09 10:30	11.2.0.1.0	210.12	1,403.79	56.75

Time Model Statistics

- Time Model
- Time Model - % of DB time

Back to Top

Time Model

| I#	DB time (s)	DB CPU (s)	SQL Exec Ela (s)	Parse Ela (s)	Hard Parse Ela (s)	PL/SQL Ela (s)	Java Ela (s)	bg time (s)	bg CPU (s)
1	97,843.05	22,933.50	85,066.94	4,744.21	3,785.93	3,596.60	0.00	19,832.38	9,272.84
2	84,227.29	22,788.27	73,345.78	3,393.95	2,549.83	3,382.22	0.00	19,364.87	9,140.41
Sum	182,070.34	45,721.77	158,412.72	8,138.17	6,335.76	6,978.82	0.00	39,197.24	18,413.25
Avg	91,035.17	22,860.89	79,206.36	4,069.08	3,167.88	3,489.41	0.00	19,598.62	9,206.62
Std	9,627.80	102.69	8,288.11	954.78	874.05	151.59	0.00	330.58	93.64

ORACLE

My database was running fine yesterday but it is really slow today? What has changed?

Tip: Use AWR Compare Periods Report to Identify Changes in Database Performance

- AWR Compare Periods Report
  - awrddrpt.sql – single instance
  - awrgdrpt.sql - RAC
- Compares database performance over two time periods
- Good for identifying what changed in performance
- Tip: Save AWR snapshots of time periods with good performance for reference

- Example:
  - Overall system performance resulting from database upgrade
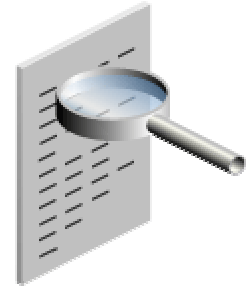
AWR Global Compare Periods Report

- Compares global RAC performance for two time ranges

- This report compares the performance of a two node RAC system, before and after an upgrade from Oracle Database 11.1.0.7 to Oracle Database 11.2.

WORKLOAD REPOSITORY RAC Compare Period Report for

| Set	DB Id	Inst #	Instance	Release	Host	Startup	Begin Snap Time	End Snap Time	Elapsed Time (min)	DB time (min)	Instance Up Time (hrs)	Avg Active Sessions	Platform
1st	1679034986	1	b1rac	11.1.0.7.0	dbs232	03-Jul-09 21:15	13-Jul-09 07:00	13-Jul-09 10:30	210.07	1,929.71	229.24	9.19	Linux IA (32-bit)
		2	b2rac	11.1.0.7.0	dbs233	03-Jul-09 21:16	13-Jul-09 07:00	13-Jul-09 10:30	210.07	1,298.80	229.23	6.18	Linux IA (32-bit)
2nd	1679034986	1	b1rac	11.2.0.1.0	dbs232	18-Jul-09 01:44	20-Jul-09 07:00	20-Jul-09 10:30	210.17	1,630.72	56.77	7.76	Linux IA (32-bit)
		2	b2rac	11.2.0.1.0	dbs233	18-Jul-09 01:45	20-Jul-09 07:00	20-Jul-09 10:30	210.12	1,403.79	56.75	6.68	Linux IA (32-bit)

AWR Global Compare Periods Report (cont.)

- The Load Profile shows a reduction in DB Time per second and per transaction after the upgrade – overall performance has improved. The upgrade was a success.

Load Profile

	1st per sec	2nd per sec	%Diff	1st per txn	2nd per txn	%Diff
DB time:	15.37	14.44	-6.05	1.52	1.43	-5.92
CPU time:	3.06	3.14	2.61	0.30	0.31	3.33
Redo size:	412,176.31	486,745.64	18.09	40,845.08	48,084.27	17.72
Logical reads:	51,390.61	47,690.30	-7.20	5,092.61	4,711.19	-7.49
Block changes:	4,089.18	3,448.61	-15.66	405.22	340.68	-15.93
Physical reads:	4,996.38	5,295.81	5.99	495.12	523.16	5.66
Physical writes:	59.45	52.65	-11.44	5.89	5.20	-11.71
User calls:	894.08	1,051.95	17.66	88.60	103.92	17.29
Parses:	369.98	397.41	7.41	36.66	39.26	7.09
Hard parses:	15.87	19.55	23.19	1.57	1.93	22.93
W/A MB processed:	2,716,342.99	2,558,789.85	-5.80	269,179.10	252,775.84	-5.80
Logons:	9.34	9.19	-1.61	0.93	0.91	-2.15
Executes:	1,269.08	1,521.87	19.92	125.76	150.34	19.55
Transactions:	10.09	10.12	0.30			

Comparative Performance Analysis
with AWR Baselines
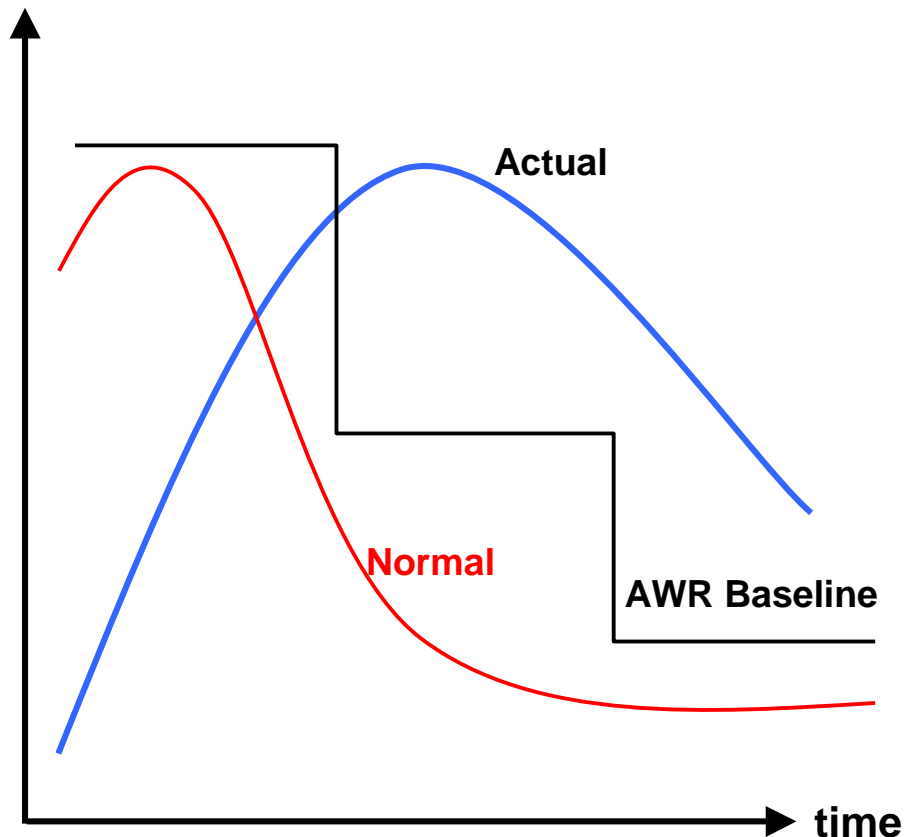
- AWR Baseline contains a set of AWR snapshots for an "interesting or reference" period of time
- Baseline are key for performance tuning to
  - guide set alert thresholds
  - monitor performance
  - compare advisor reports
- User-specifiable, schedulable, e.g.:
  - last Thanksgiving period
  - every Monday 10am-noon for 4 Mondays
- Automatically captures 8-day moving window baseline for week to week comparisons (default)

ORACLE

A user complains that his session seemed to hang for a few minutes. What happened?

Tip: Use ASH for Targeted Performance Diagnostics

- AWR snapshots and reports cover entire system

- Transient events can be averaged over a snapshot and be non-obvious from an AWR report

- ASH can be used for examining:
  - Targeted time range
  - A specific
    - session
    - service
    - wait_class
    - client_id
    - SQL_ID

  - A targeted time range in combination with the above

ASH Report

- Click on 'Run ASH Report' button from performance page
- Select time range and dimension to report on

ASH Report:

- ASH Report of a single session for a 5 minute period

- The session accounted for 52% of database activity for the time period

- The session spent 64% of the time in a concurrency wait event, 36% on CPU. There is clearly some opportunity for tuning.

ASH Report For PROD/prod

(1 Report Target Specified)

DB Name	DB Id	Instance	Inst num	Release	RAC	Host
PROD	48067057	prod	1	11.1.0.7.0	NO	tdsmng04-d1

CPUs	SGA Size	Buffer Cache	Shared Pool	ASH Buffer Size
4	798M (100%)	128M (16.0%)	248M (31.1%)	8.0M (1.0%)

	Sample Time	Data Source
Analysis Begin Time:	13-Sep-09 17:54:31	V$ACTIVE_SESSION_HISTORY
Analysis End Time:	13-Sep-09 17:59:31	V$ACTIVE_SESSION_HISTORY
Elapsed Time:	5.0 (mins)	
Sample Count:	274	
Average Active Sessions:	0.91	
Avg. Active Session per CPU:	0.23	
Report Target:	SESSION_ID = 142	52.2% of total database activity

Top User Events

Event	Event Class	% Event	Avg Active Sessions
cursor: pin S wait on X	Concurrency	64.10	0.08
CPU + Wait for CPU	CPU	35.90	0.05

ORACLE

Has my SQL statement's performance changed over time?

AWR Individual SQL Report

- AWR Report for a particular SQL Statement -awrsqrpt.sql

- Useful for researching individual SQL statement performance over time

- Example: Single SQL statement, before and after tuning

- CPU Time per execution substantially decreased – tuning was successful.

Plan Statistics

Before tuning

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	571,421	2,747.22	41.67
CPU Time (ms)	569,862	2,739.72	41.71
Executions	208		
Buffer Gets	145,778,328	700,857.35	39.82

After tuning

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	33,905	69.48	55.37
CPU Time (ms)	33,920	69.51	56.34
Executions	488		
Buffer Gets	848,144	1,738.00	27.52

ORACLE

# Migrating AWR Data

- Why migrate AWR data?
    - To offload analysis from production database
    - To preserve data longer than the production default retention period
    - To do multi-database comparative analysis
- Scripts are located in $ORACLE_HOME/rdbms/admin

Action	SQL Script
Extract data from AWR into a datapump file	awrextr.sql
Load data from datapump file created by awrextr.sql into a database	awrload.sql
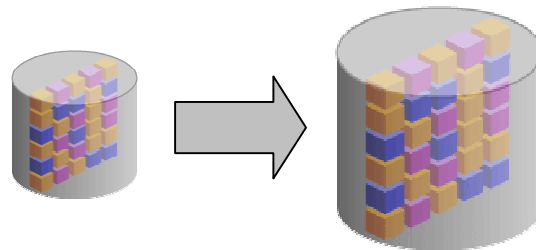
Agenda

- Automatic Workload Repository Infrastructure

- Out of the box AWR reports
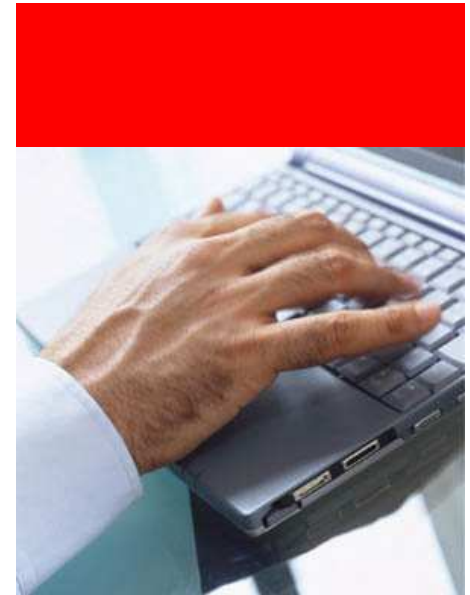
- Active Reports

Tip: Use Real-Time SQL Monitoring And Active Reports

- Automatically monitors long running SQL

- Enabled out-of-the-box with no performance impact

- Monitors each SQL execution

- Exposes monitoring statistics
  - Global execution level
  - Plan operation level
  - Parallel Execution level

- Can be saved or emailed as an 'Active Report'

Demo

Active Reports

Conclusion

- AWR contains vast amounts of database performance data
- Enterprise Manager displays a comprehensive view and analysis of that data
- Standard reports can provide additional insights
  - Use the right report for the right problem
- Active reports are a useful new tool for the Oracle database professional

Oracle Helps You Maximize Customer Value

NetApp — Deploys SOA infrastructure 92% faster	**NOKIA** — Saves 80% time and effort for managing Databases	**TomTom** — Avoids online revenue losses up to 25%
AmTrust Bank — Improves IT productivity by 25%	**Commonwealth Bank** — Drives asset utilization up by 70%	**Cummins** — Cuts configuration management effort by 90%
Telstra — Saves $1.9 million with Oracle Enterprise Manager	**SRM** — Saves $170,000 per year with Oracle Enterprise Manager	**City University London** — Replaces manual tools with automation; saves time by 50%
Starwood — Reduces Database testing time by 90%	**Bayer** — Reduces provisioning effort by 75%	**Meridian Knowledge Solutions, LLC** — Saves weeks on application testing time
CUMIS — Cuts application testing from weeks to hours	**APL — The Johns Hopkins University Applied Physics Laboratory** — Reduces critical patching time by 80%	**City of Evanston** — Delivers 24/7 uptime with Oracle Enterprise Manager

ORACLE®

Additional Oracle Enterprise Manager sessions

Wednesday, Oct. 14	Location
• 11:45 a.m.- Performance-Testing Oracle E-Business Suite: Tips and Tricks from Oracle Experts • 11:45 a.m.- Oracle Enterprise Manager: Monitoring and Security Best Practices • 11:45 a.m.- Upgrade Case Study: Database Replay, Snapshot Standby, and Plan Baselines	• Moscone S. Room 305 • Moscone S. Room 303 • Moscone S. Room 301
• 1:45 p.m.- Quick Tips for Database Performance Tuning • 5:00 p.m.- SQL Gone Wild: Taming Bad SQL the Easy Way (or the Hard Way) • 5:00pm Oracle Enterprise Manager: Beyond the Basics: Getting More from Oracle Enterprise Manager	• Moscone S. Room 104 • Moscone S. Room 102 • Moscone S. Room 305

Additional Oracle Enterprise Manager sessions

Thursday, Oct. 15	Location
• 10:30 a.m.Everyone Is Applying CPUs (Critical Patch Updates), Right? • 10:30 am DBA's New Best Friend: Oracle Database 10g and Oracle Database 11g SQL Performance Analyzer	• Moscone S. Room 303 • Moscone S. Room 102
• 12:00 p.m. Worry-Free Application Upgrade Using Oracle Change Management Pack • 1:30pm Dell IT's Implementation of Oracle Enterprise Manager Patch Provisioning	• Moscone S. Room 102 • Moscone S. Room 306

ORACLE®