THE INFORMATION COMPANY

ORACLE

Gongloor, Prabhaker (GP)
Sr. Product Manager

Oracle Corporation

NoCOUG Winter Conference
Feb 14, 2006

ORACLE

Performance Diagnosis Usage Model in Oracle Database 10g

ORACLE

Agenda

- Oracle DB 10g Performance Monitoring and Diagnostics: Architecture
- Automatic Database Diagnostic Monitor (ADDM)
- Performance Diagnosis: Usage Model
- Best Practices (* time permitting)

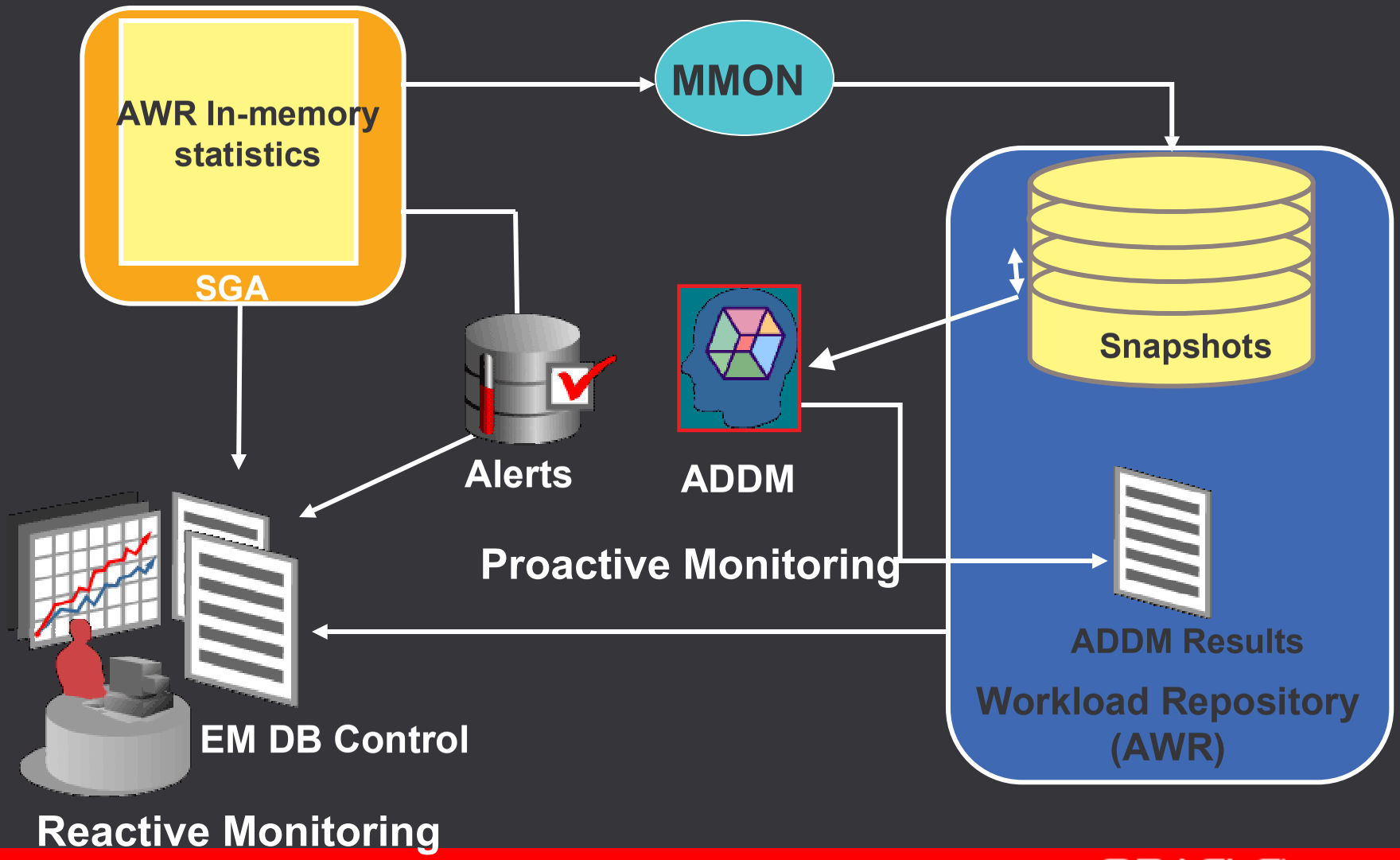
Performance Diagnosis – Life Before Oracle 10g

- Performance Diagnosis & Tuning is Complex
- Diagnosis often requires additional data capture
- Data overload rather than information
- Database wide view of operations is lacking
- Misguided tuning efforts waste time & money

Oracle DB 10g Performance Monitoring and Diagnostics: Architecture

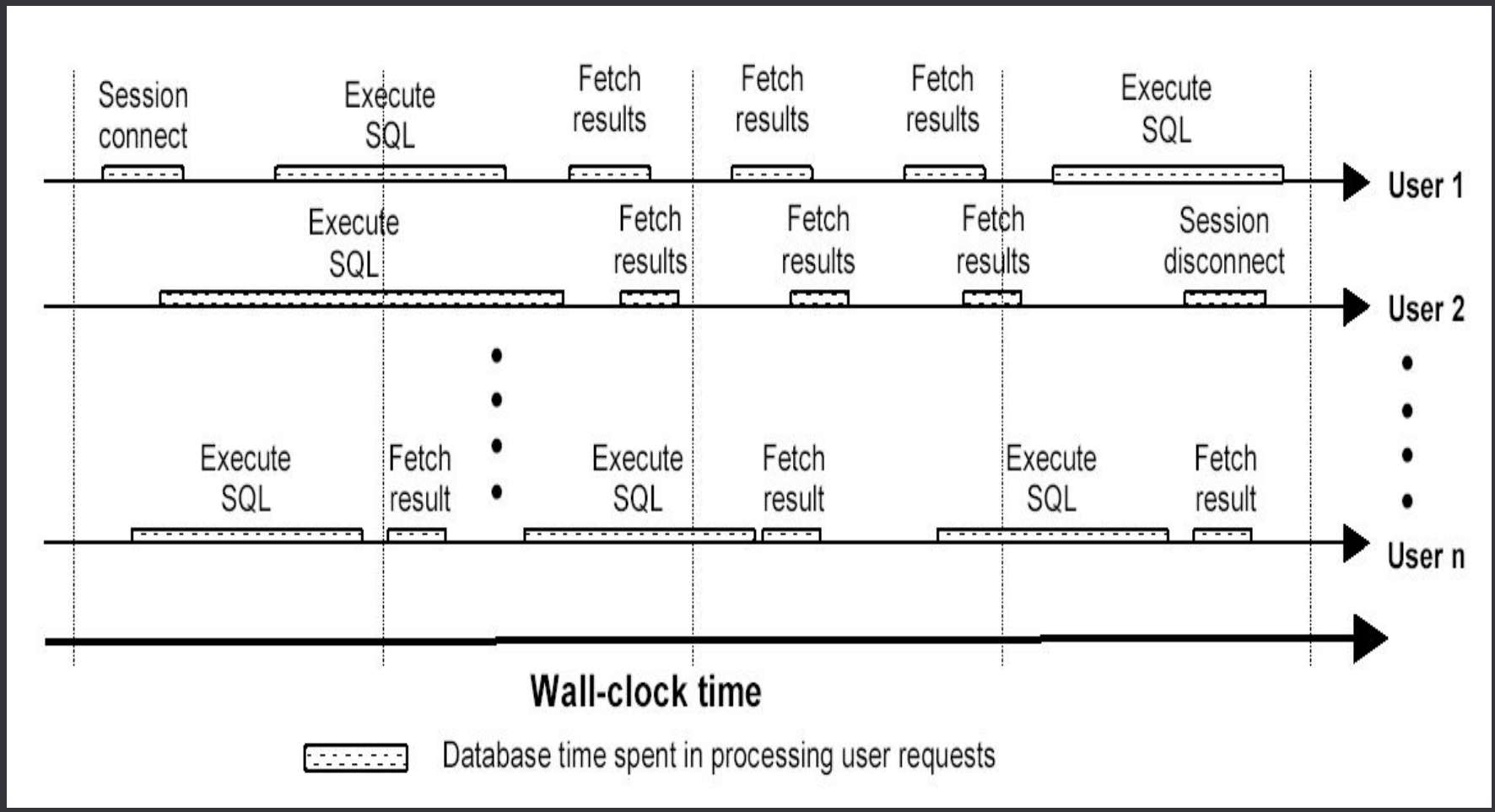
ORACLE

Performance Monitoring and Diagnostics: Architecture



AWR and In Memory Statistics

- OOW 2005 Session: Oracle DB 10g Self-Management Framework Internals: Exploring the AWR: Will be posted on OTN shortly
- AWR
  - a.k.a. STATSPACK++, Efficient, Space automatically managed, Snapshot every 1 hr
  - Foundation for Self-Learning and Managing DB
  - Various Statistics: Base, SQL, Metrics, OS Stats, ASH, etc.
- Enhanced Time and Wait Model
  - Enables Tracking Components through Common Currency "Time" , *DB Time*
  - Events (800+) Classified into 12 Solution Areas
- Active Session History
  - Sampled History of all Active Sessions (analogous to V$SESSION_WAIT sampling)

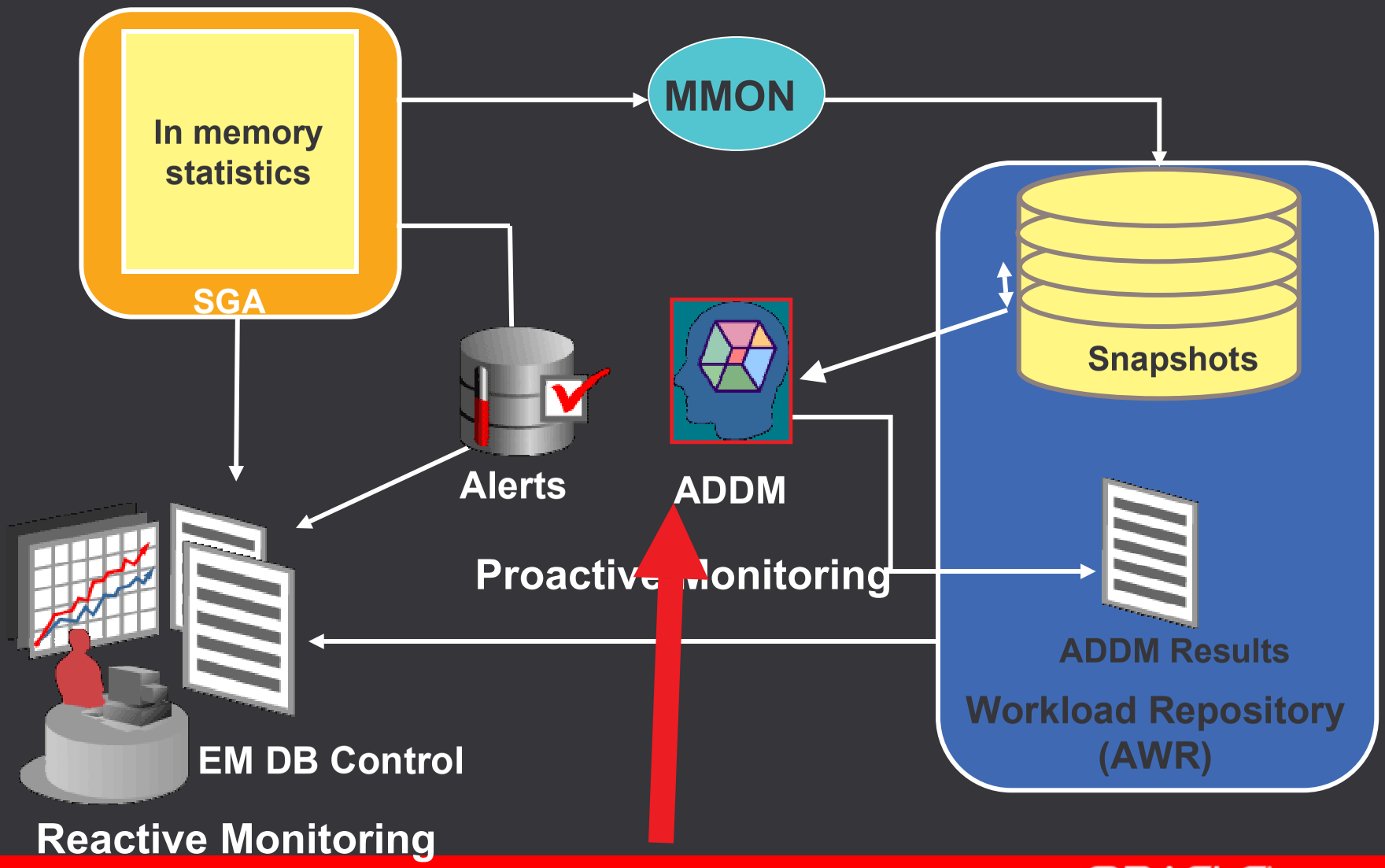


1. Total DB Time = Sum of Time Spent Processing All User Requests

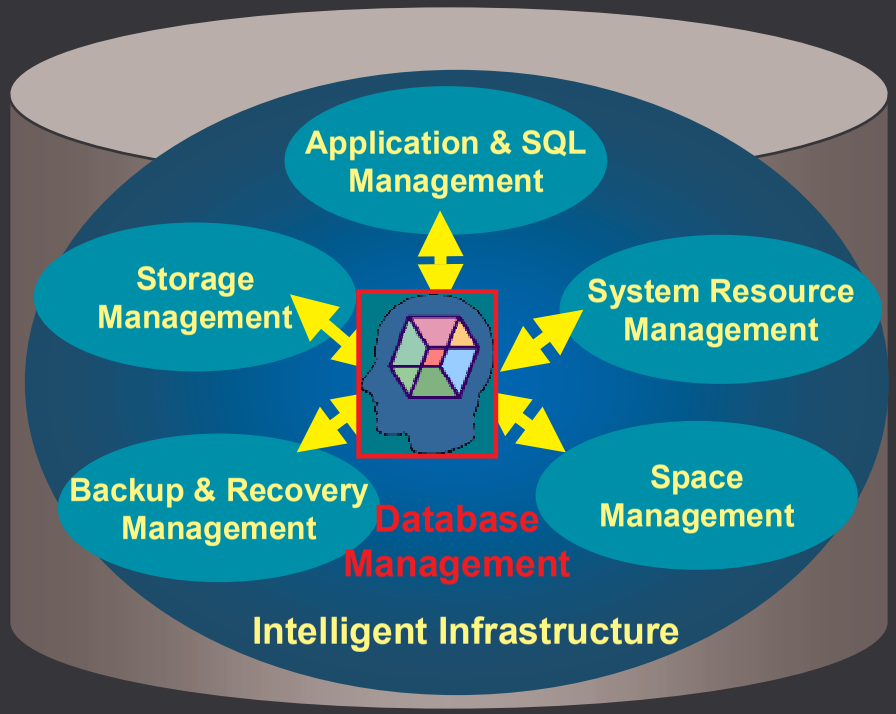
= Sum of Time (Running on CPU + Waiting for Resources)

2. DB Time/Sec (Avg. Active Sessions) = Total DB Time / Wall Clock Time

Performance Monitoring and Diagnostics: Architecture

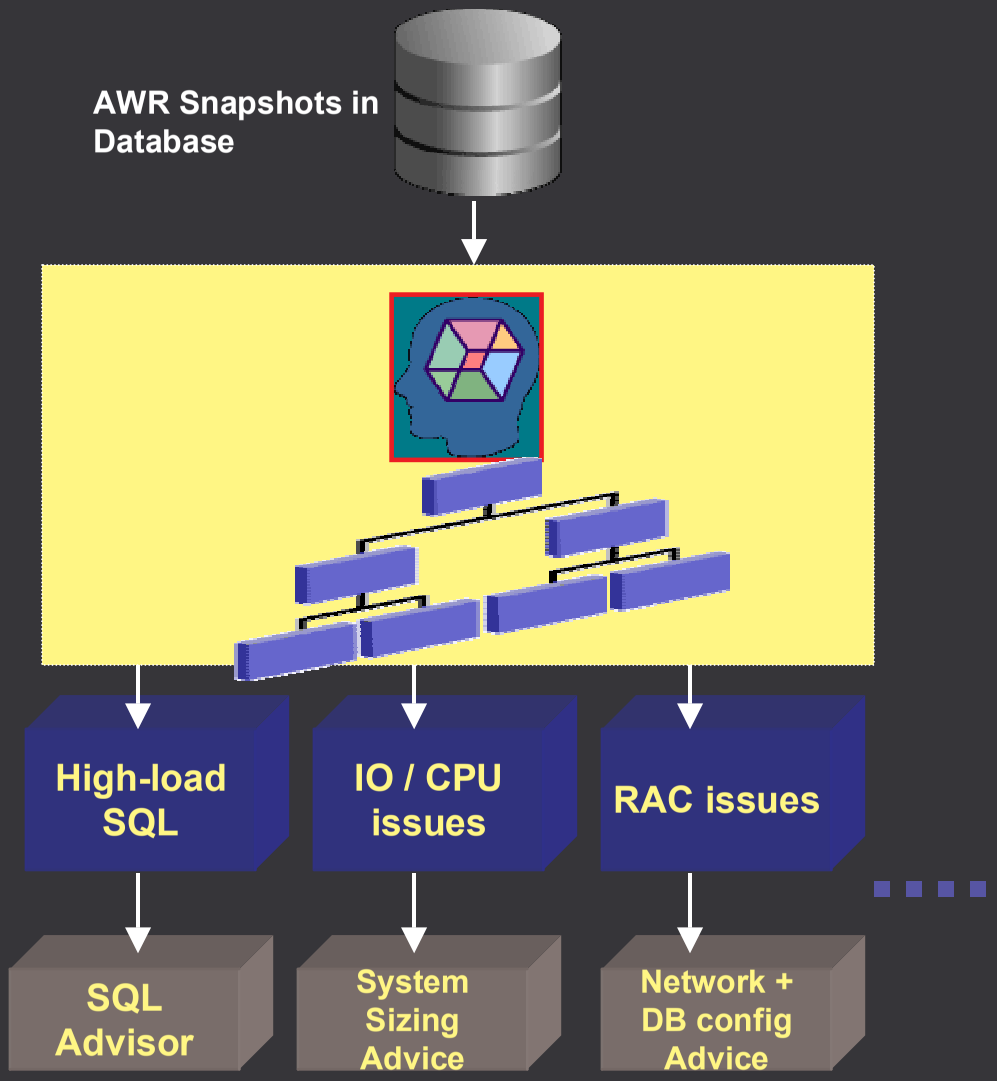


Automatic Database Diagnostic Monitor (ADDM)



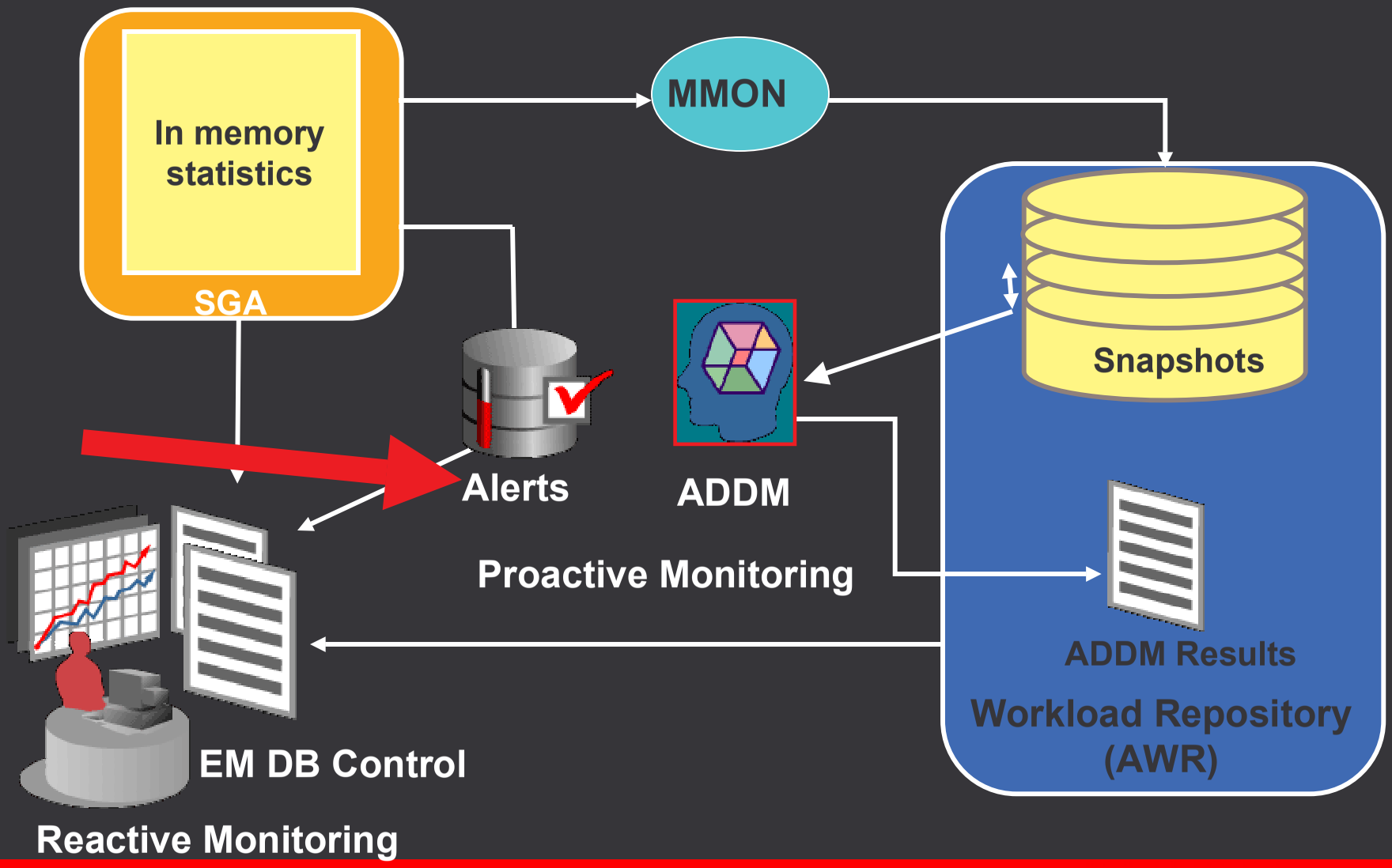
- Integrates all DB components together: DB Time Currency
- Automatic database-wide performance diagnostic, including RAC, Streams..
- Emphasis on Root Cause Analysis vs. Symptoms
- Provides impact and benefit analysis
- ADDM gives recommendations in context of workload running on your system

ADDM's Architecture



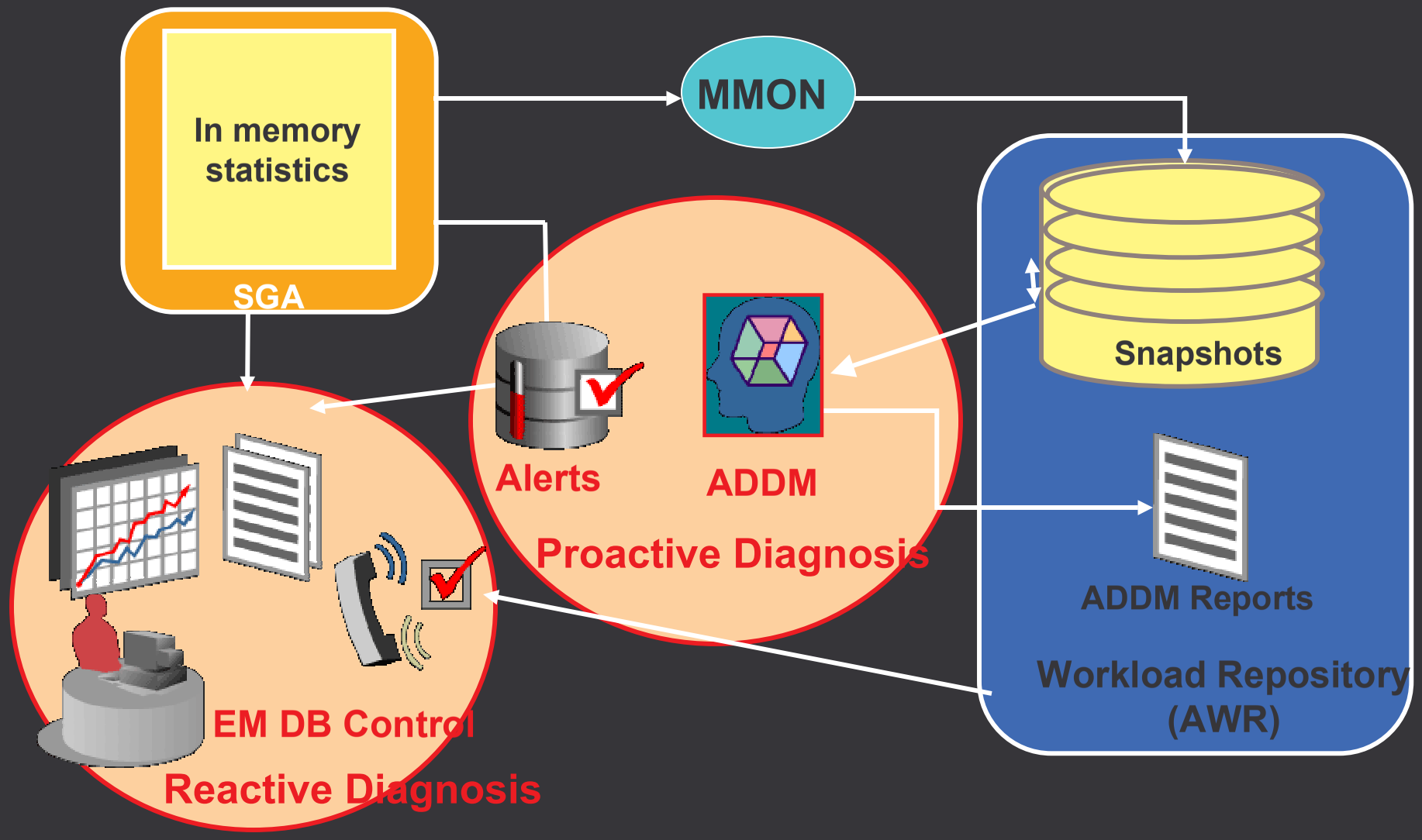
- Top Down Diagnosis Using AWR Snapshots & Time-Wait Model
- Throughput centric - Focus on reducing time 'DB time'
- Classification Tree - based on decades of Oracle performance tuning expertise
- Real-time results
  - Runs proactively, reactive when required
- Pinpoints root cause
  - Distinguishes symptoms from the root cause
- Reports non-problem areas
  - E.g. I/O is not a problem

Performance Monitoring and Diagnostics: Architecture

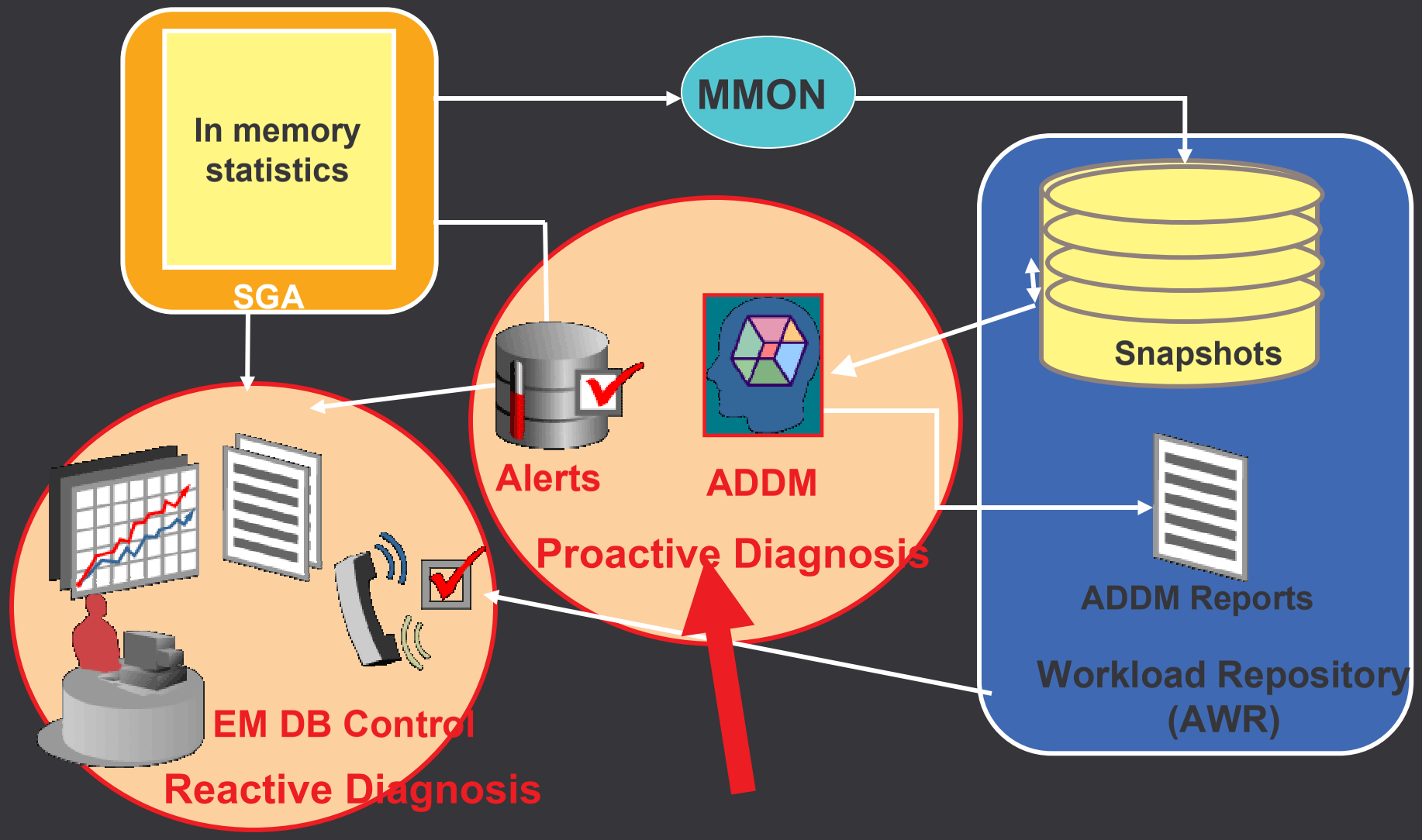


Performance Diagnosis: Usage Model

Performance Diagnosis: Usage Model



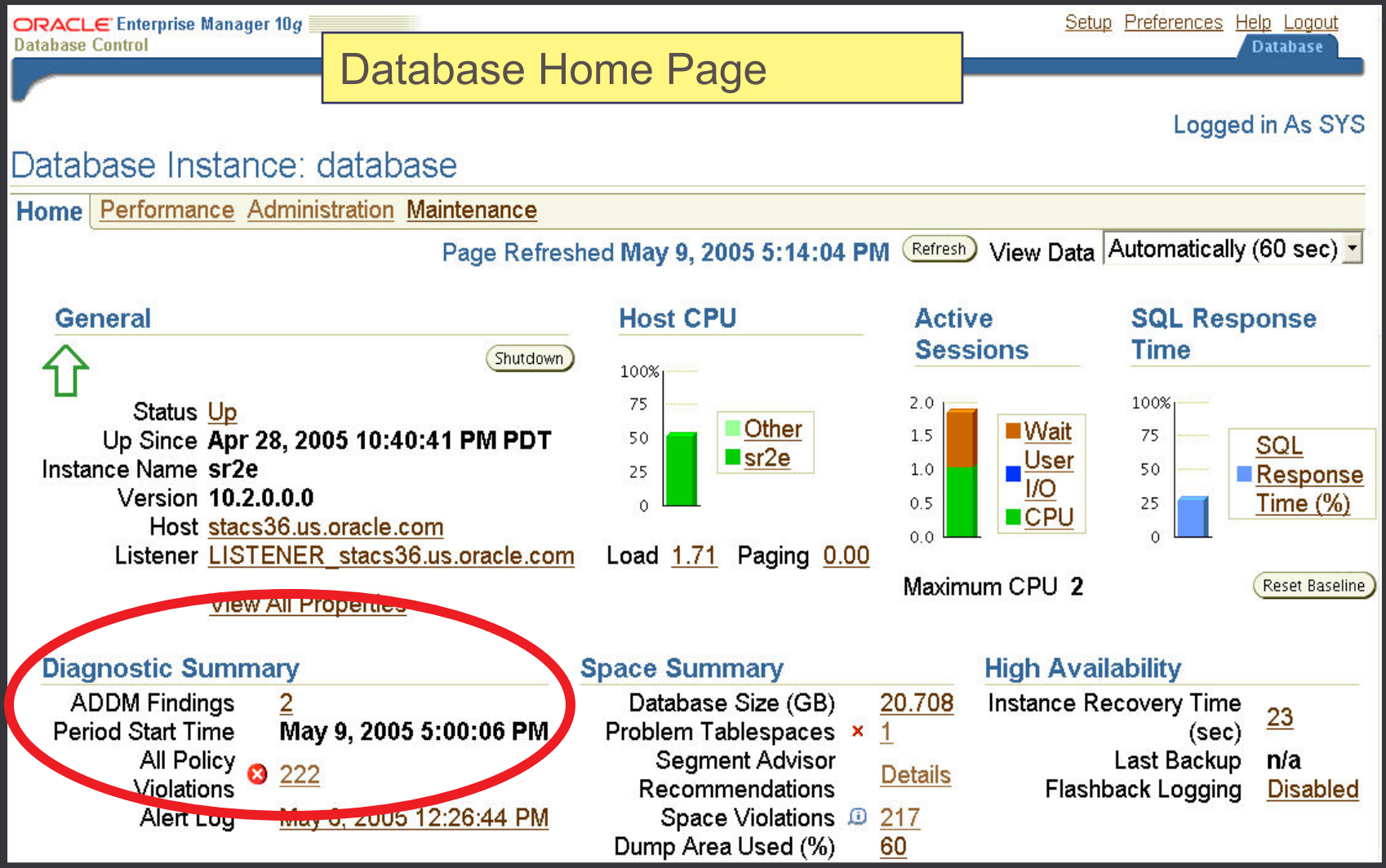
Performance Diagnosis: Usage Model



Proactive Performance Diagnosis

- Database performs proactive diagnosis periodically
  - ADDM findings are stored in Advisory Framework
  - Enables historical performance analysis
  - Can be invoked manually
- EM Interface provides guided problem resolution

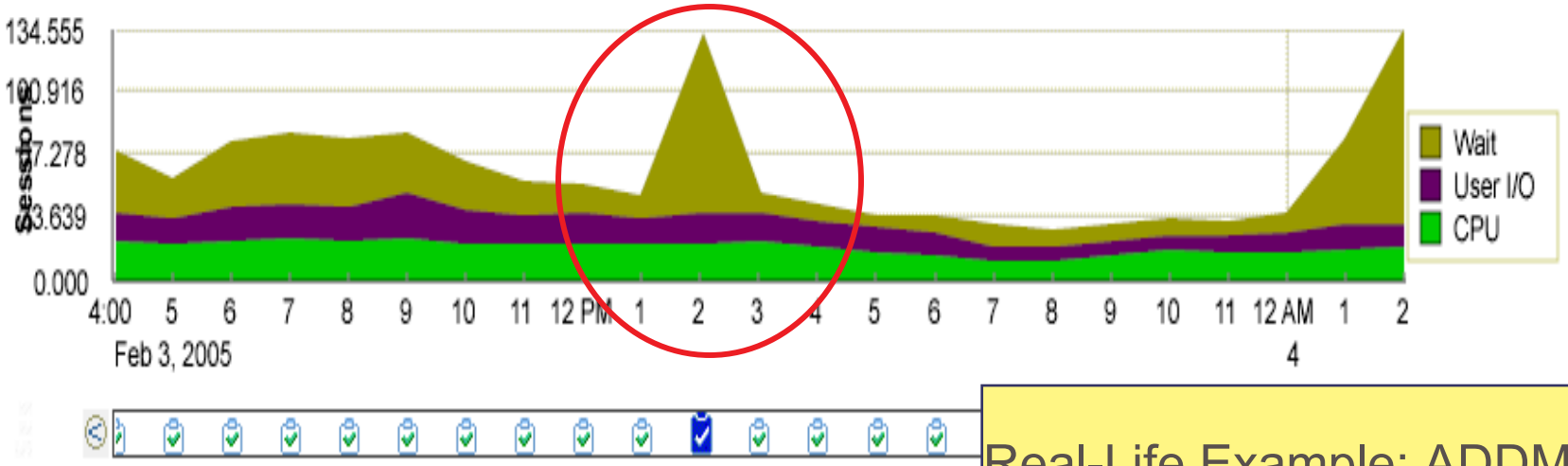
Proactive Performance Diagnosis



Database Activity

Create ADDM Task

The selected icon below the graph identifies the performance analysis period. Click on a different icon to select a different analysis period.



Real-Life Example: ADDM identified the problem around noon

Performance Analysis

Task Name **ADDM:317772662_3_363**

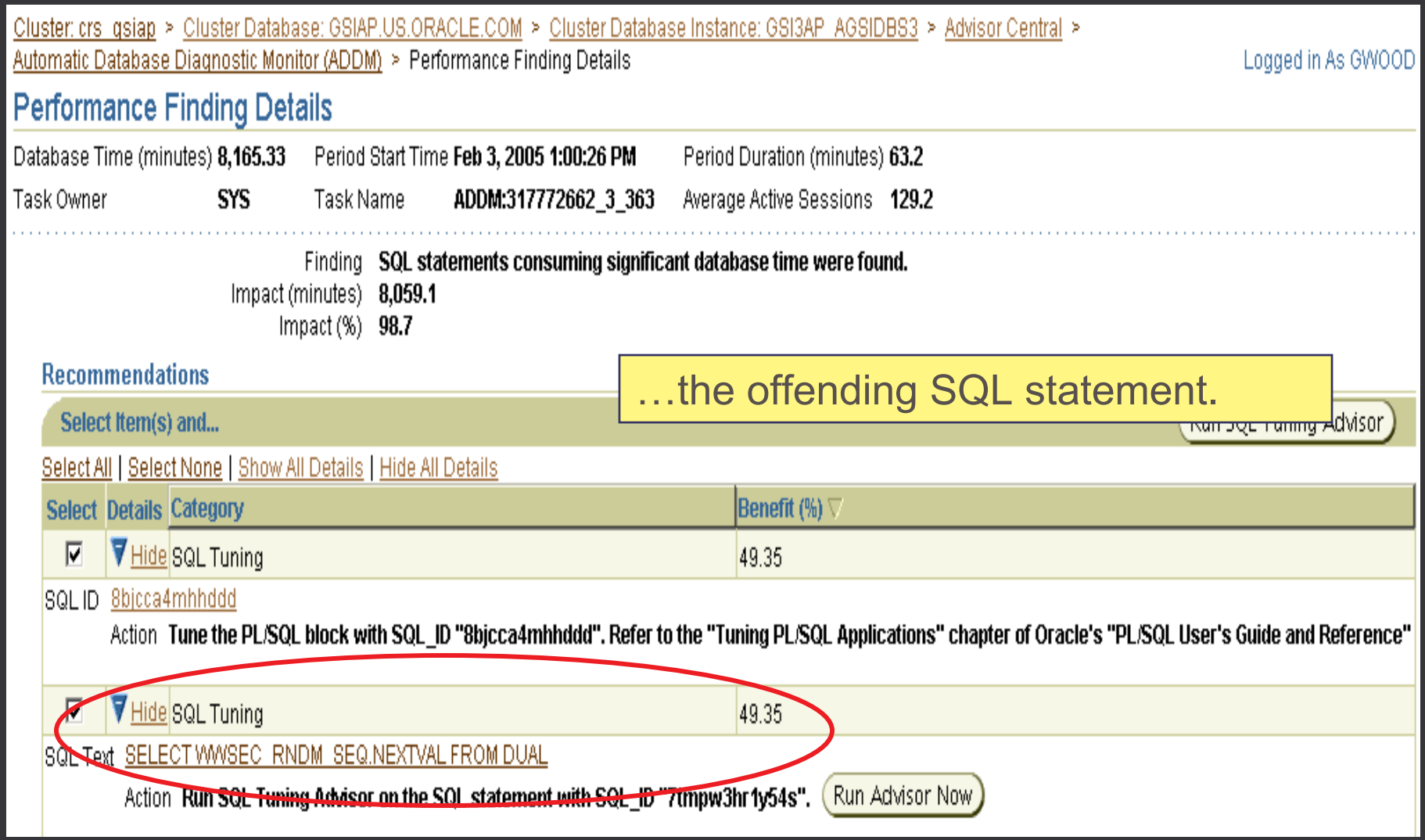
View Snapshots | View Report

Database Time (minutes) **8,165.33** Period Start Time **Feb 3, 2005 1:00:26 PM** Period Duration (minutes) **63.2**

Task Owner **SYS** Average Active Sessions **129.2**

Impact (%)	Finding	Recommendations
98.7	SQL statements consuming significant database time were found.	**2** SQL Tuning
49.92	Wait event "row cache lock" in wait class "Concurrency" was consuming significant database time.	**3** Application Analysis
49.34	Sequence cache misses were consuming significant database time.	**1** Application Analysis

Proactive Performance Diagnosis



Proactive Performance Diagnosis

View Sequence: SSOSDK.WWSEC_RNDM_SEQ

Edit OK

General

Name **WWSEC_RNDM_SEQ**

Schema **SSOSDK**

Type **Ascending**

Values

Maximum **Unlimited**

Minimum **1**

Interval **1**

Last Number **17223385**

Cache

No Cache

Root Cause of the problem…

Reactive Performance Diagnosis

Reactive Performance Diagnosis

- Perform General Manual Analysis
  - React to an alert or validate tuning action
  - Use EM Perf. Page drill-downs

Reactive Performance Diagnosis

- Perform General Manual Analysis
  - React to an alert or validate tuning action
  - Use EM Perf. Page drill-downs
- Diagnosing Transient or Targeted problems
  - Transient performance analysis (few min.)
  - Targeted Analysis: Time, session, module, SQL , etc. or their combination
  - Use ASH report

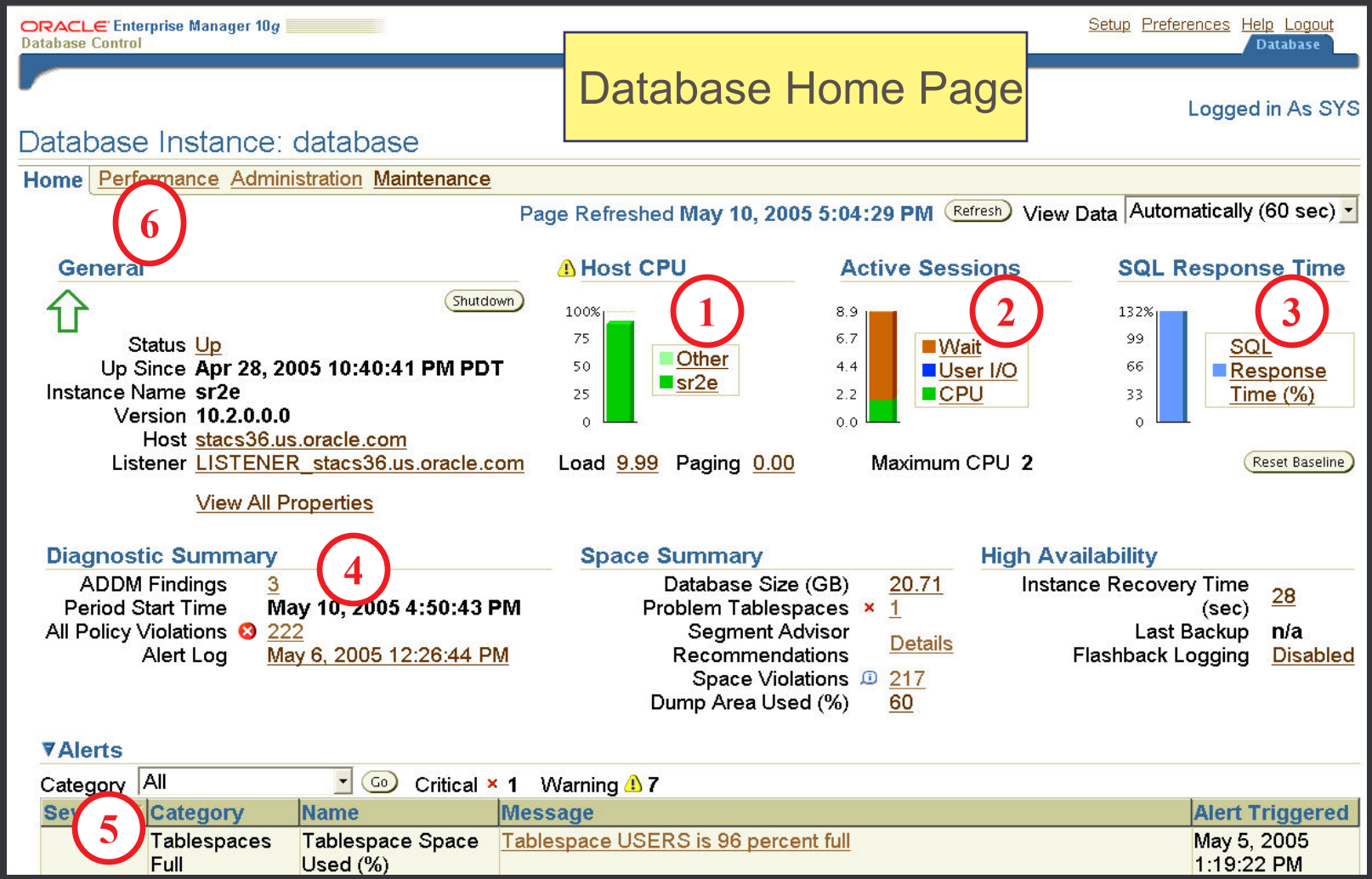
Reactive Performance Diagnosis

- Perform General Manual Analysis
  - React to an alert or validate tuning action
  - Use EM Perf. Page drill-downs
- Diagnosing Transient or Targeted problems
  - Transient performance analysis (few min.)
  - Targeted Analysis: Time, session, module, SQL , etc. or their combination
  - Use ASH report
- Diagnosing problems due to db workload or configuration changes
  - Compare Performance to “Baseline”
  - Use Compare Periods Report

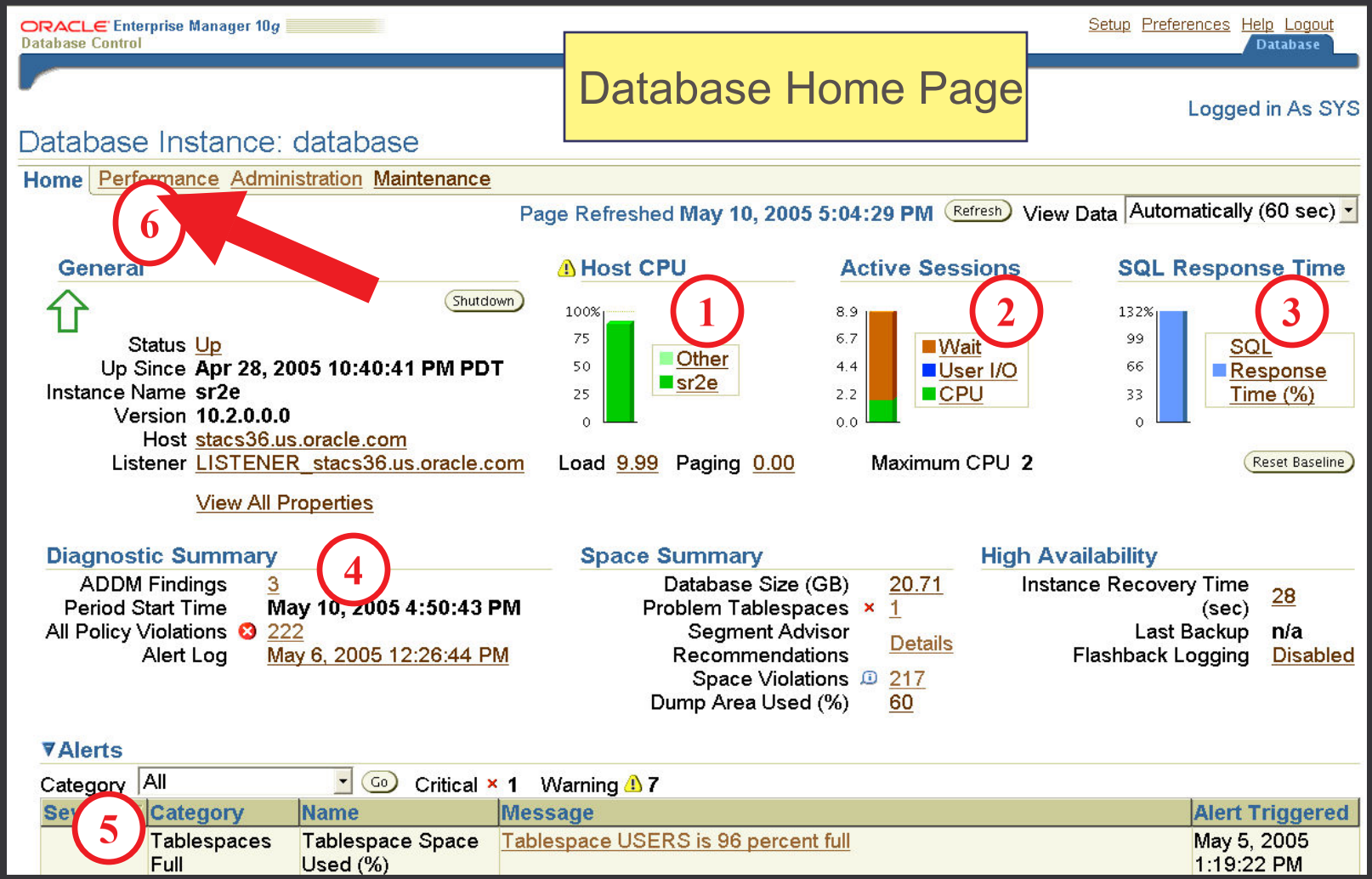
Reactive Performance Diagnosis

- Perform General Manual Analysis
  - React to an alert or validate tuning action
  - Use EM Perf. Page drill-downs
- Diagnosing Transient or Targeted problems
  - Transient performance analysis (few min.)
  - Targeted Analysis: Time, session, module, SQL , etc. or their combination
  - Use ASH report
- Diagnosing problems due to db workload or configuration changes
  - Compare Performance to “Baseline”
  - Use Compare Periods Report
- Diagnosing Slow Moving / Hung databases
  - Use EM Memory Access Mode

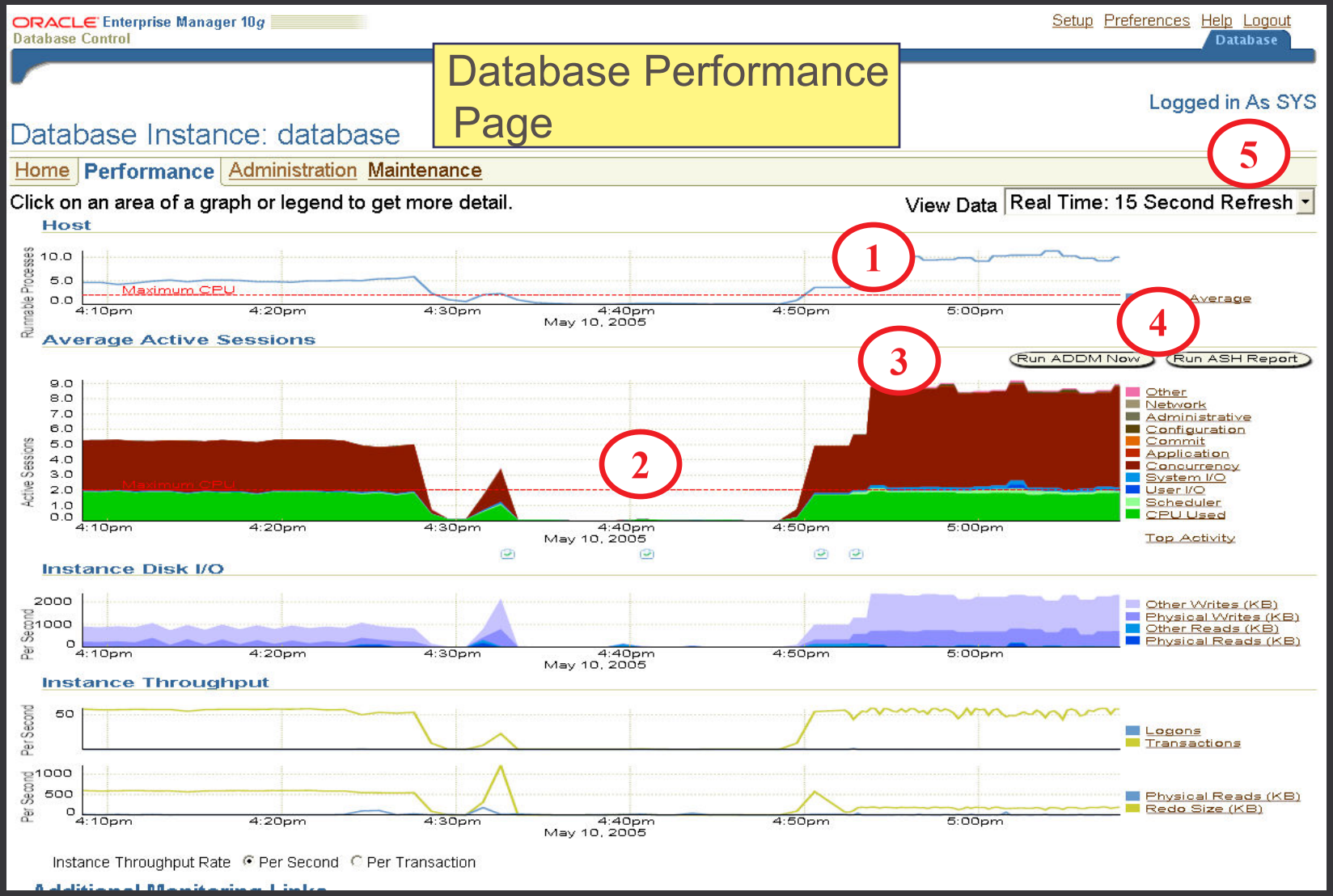
Reactive Perf. Diagnosis: EM Drill-Down



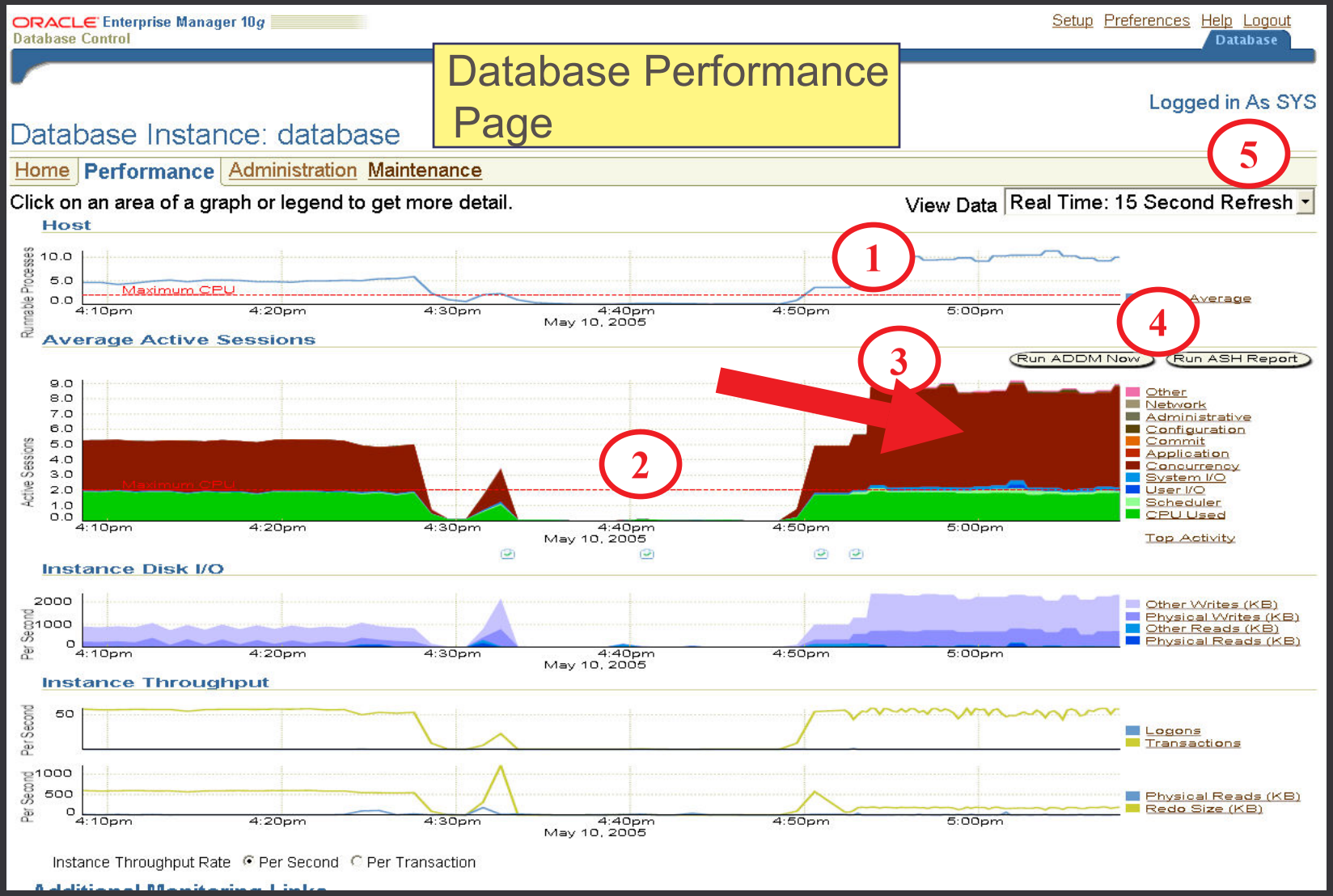
Reactive Perf. Diagnosis: EM Drill-Down



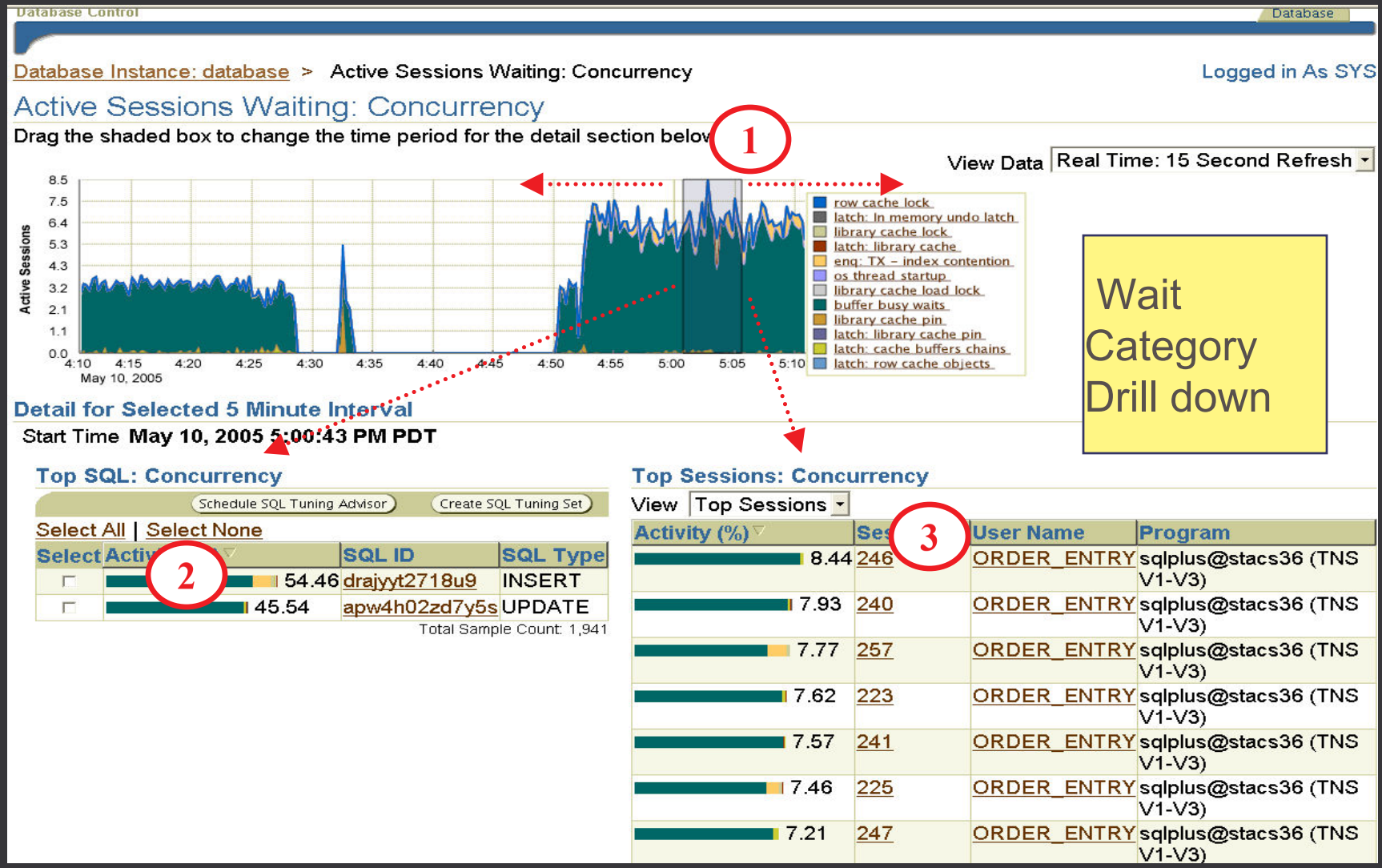
Reactive Perf. Diagnosis: EM Drill-Down



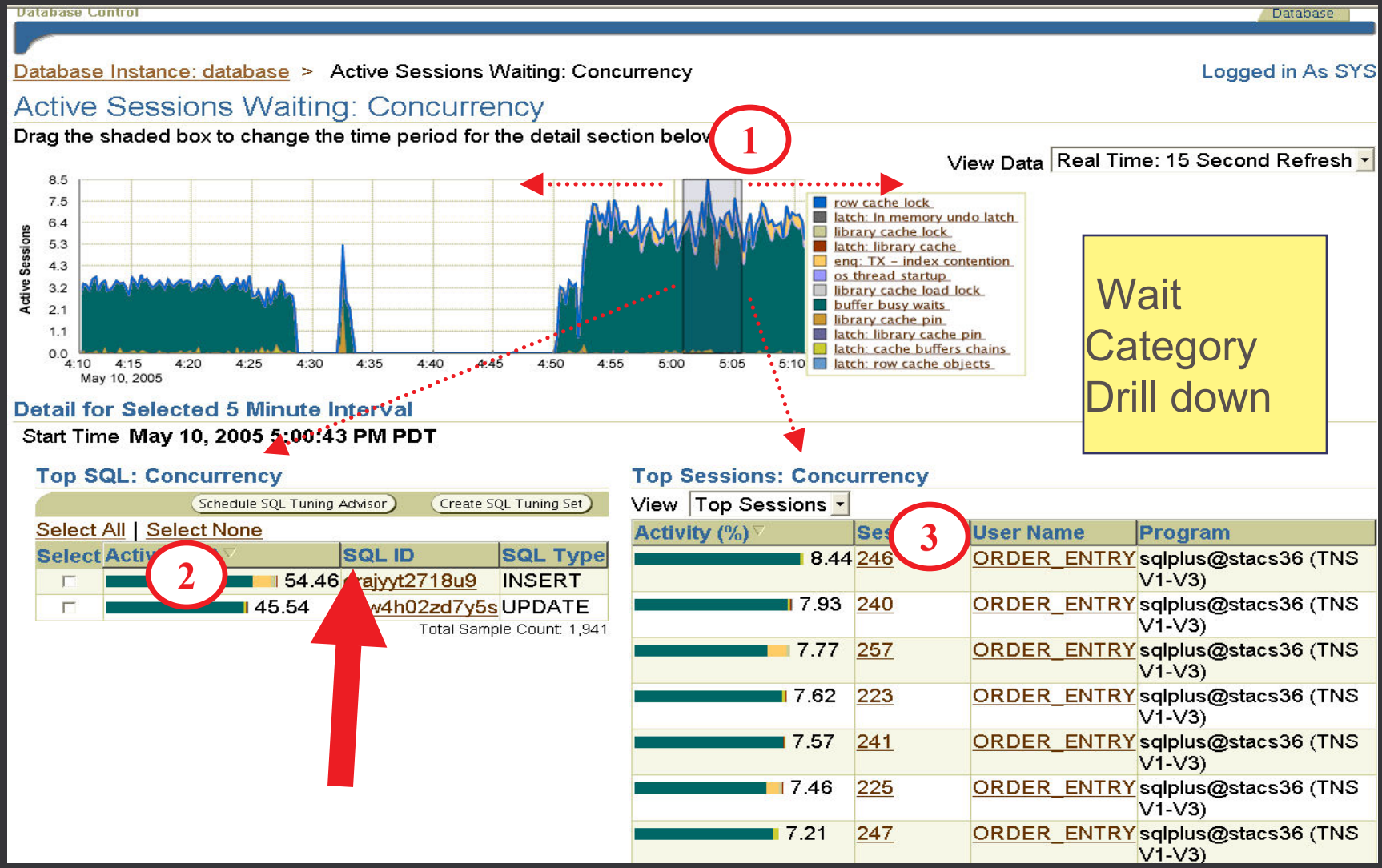
Reactive Perf. Diagnosis: EM Drill-Down



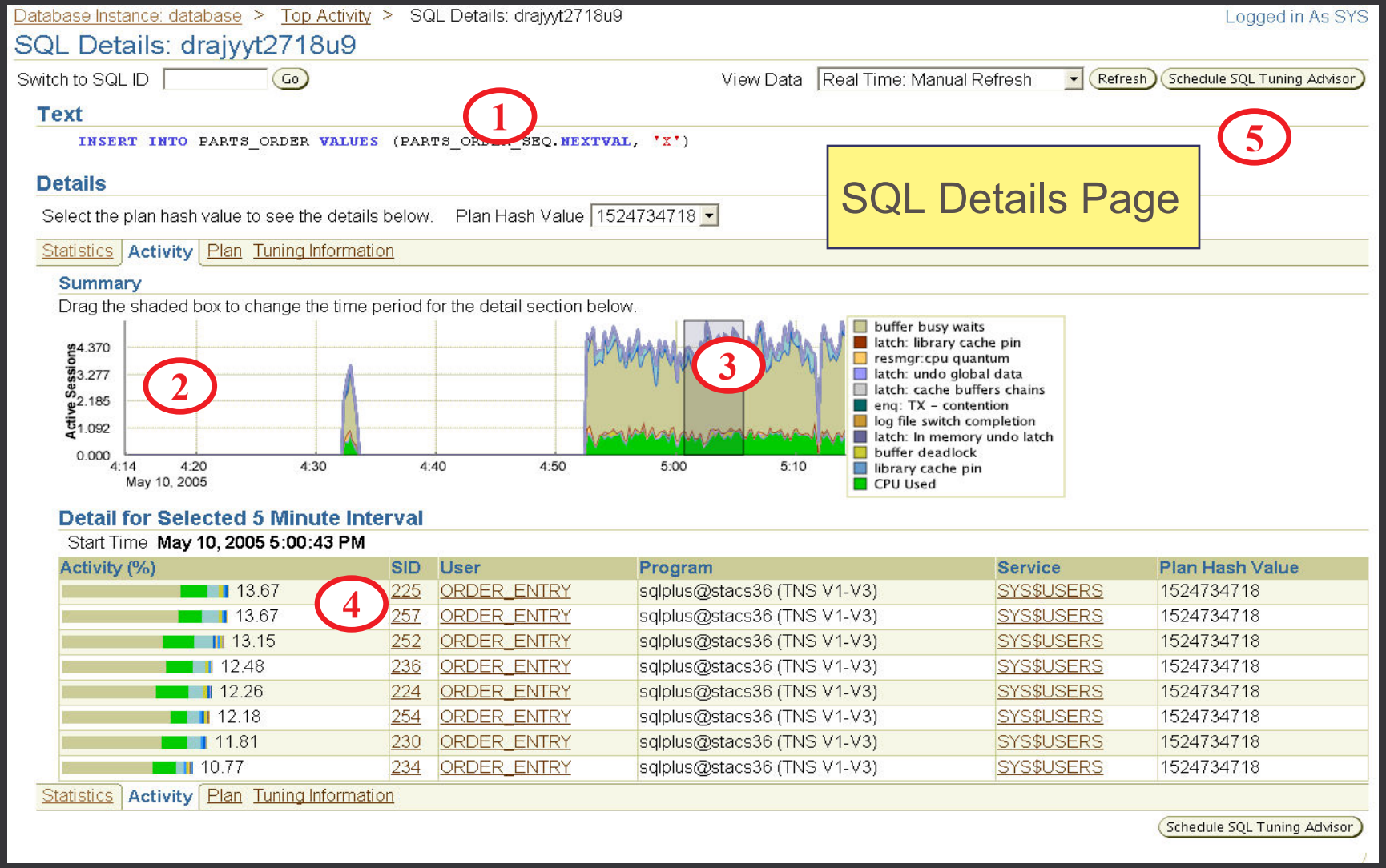
Reactive Perf. Diagnosis: EM Drill-Down



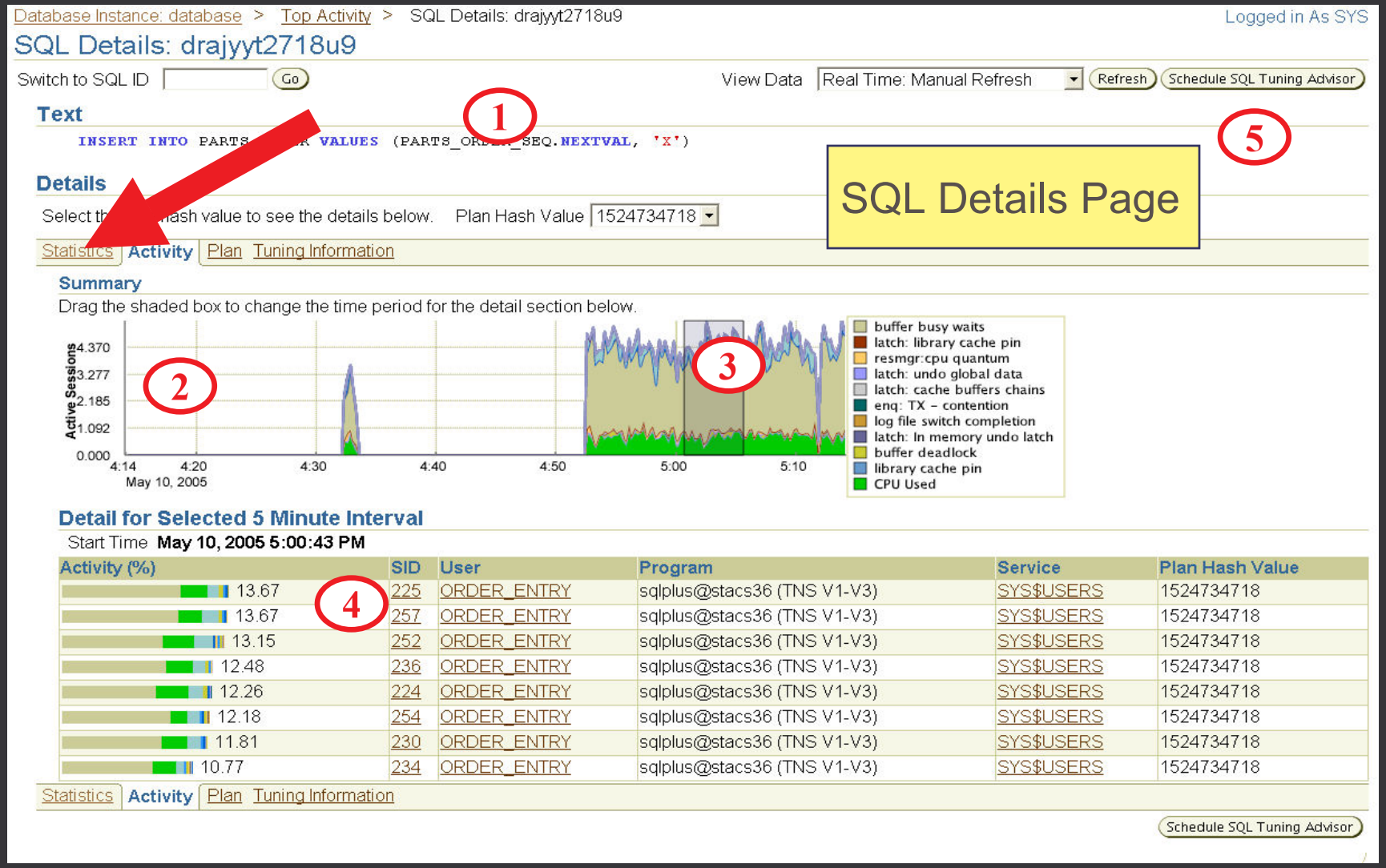
Reactive Perf. Diagnosis: EM Drill-Down



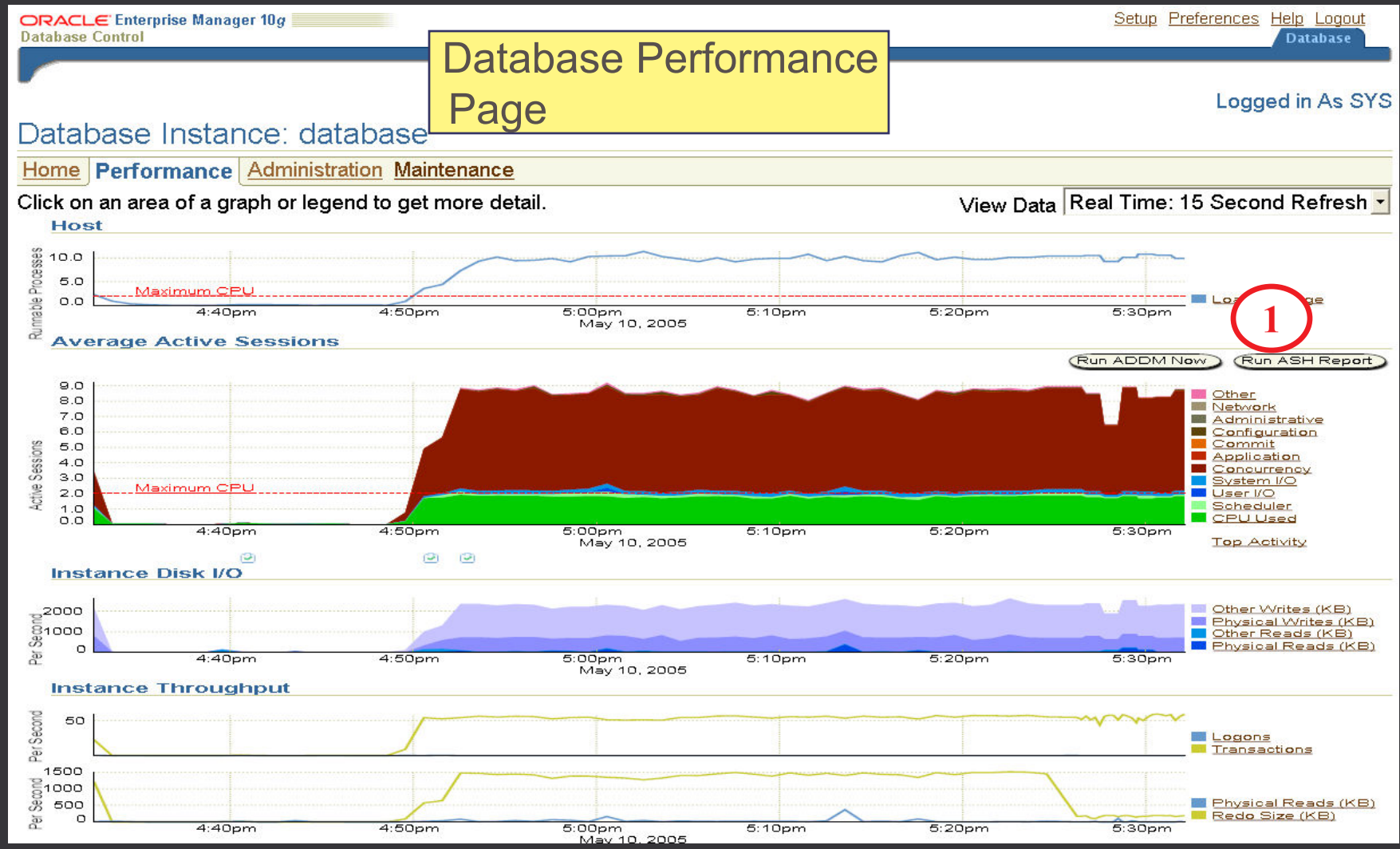
Reactive Perf. Diagnosis: EM Drill-Down



Reactive Perf. Diagnosis: EM Drill-Down



Reactive Perf. Diagnosis: EM Drill-Down



Reactive Diagnosis: Active Session History (ASH) Report

MYTH: With Statspack or SQL_TRACE who needs ASH data?

Reality: Instance Level Stats – Too little detail, Trace Level stats – Too much, intrusive, need to replay workload

- ASH Report
  - Uses “ASH” data within AWR retention period
  - Helps analyze transient performance problems (few minutes)
  - Ability to perform targeted analysis by various dimensions
    - By Time, Session ID, Action, Module, etc. or their combination
    - Provides “foot print” or execution profile
  - Facilitates “blocker” analysis
    - Enqueues, buffer busy waits, latch holders, Transaction-IDs
  - Obviates need for workload replay and SQL trace in most cases
  - Accessible through EM and command line interface

Reactive Diagnosis: ASH Report
Real World Case Study

Top Service/Module

Service	Module	% Activity
SYS$USERS	01@ D:\bug_scr2_month_latest.sql	99.68

ASH Report for a Long Running Session

Top User Events

Event	Event Class	% Activity
db file sequential read	User I/O	89.59
CPU + Wait for CPU	CPU	7.10
latch: shared pool	Configuration	2.27

Activity Over Time

- analysis period is divided into smaller time slots
- top 3 events are reported in each of those slots

Slot Time (Duration)	Event	Sample Cnt	% Activity
03:00:00 (90.0 min)	db file sequential read	258	5.17
	CPU + Wait for CPU	18	0.36
	latch: shared pool	5	0.10
04:30:00 (90.0 min)	db file sequential read	468	9.38
	CPU + Wait for CPU	39	0.78
	latch: shared pool	10	0.20
06:00:00 (90.0 min)	db file sequential read	467	9.36
	CPU + Wait for CPU	37	0.74
	latch: shared pool	13	0.26
07:30:00 (90.0 min)	db file sequential read	474	9.50
	CPU + Wait for CPU	34	0.68

Reactive Diagnosis: ASH Report
Real World Case Study

Top Service/Module

Service	Module	% Activity
SYS$USERS	01@ D:\bug_scr2_month_latest.sql	99.68

ASH Report for a Long Running Session

Skew

Top User Events

Event	Event Class	% Activity
db file sequential read	User I/O	89.59
CPU + Wait for CPU	CPU	7.10
latch: shared pool	Configuration	2.27

Activity Over Time

- analysis period is divided into smaller time slots
- top 3 events are reported in each of those slots

Slot Time (Duration)	Event	Sample Cnt	% Activity
03:00:00 (90.0 min)	db file sequential read	258	5.17
	CPU + Wait for CPU	18	0.36
	latch: shared pool	5	0.10
04:30:00 (90.0 min)	db file sequential read	468	9.38
	CPU + Wait for CPU	39	0.78
	latch: shared pool	10	0.20
06:00:00 (90.0 min)	db file sequential read	467	9.36
	CPU + Wait for CPU	37	0.74
	latch: shared pool	13	0.26
07:30:00 (90.0 min)	db file sequential read	474	9.50
	CPU + Wait for CPU	34	0.68

Reactive Perf. Diagnosis: ASH Report

ASH Report Details…

Top Sessions

- '# Samples Active' sh[illegible] the session was found waiting for [illegible] with respect to wall clock time and not total database activity.

Sid, Serial#	% Activity	Event	% Event	User	Program	# Samples Active
2035,39795	99.68	db file sequential read	89.35	SM	...	4,456/5,400 [83%]
		CPU + Wait for CPU	7.94			351/5,400 [7%]
		latch: shared pool	2.27			113/5,400 [2%]

Top SQL Statements

SQL ID	% Activity	Event	% Event	SQL Text
043s6zg4w225k	91.68	db file sequential read	86.93	SELECT COUNT(*) FROM RPTHEAD_H...
		CPU + Wait for CPU	4.23	
5tjfyxq1k2kaj	3.51	db file sequential read	1.72	SELECT H.RPTNO , H.PROGRAMMER...
		CPU + Wait for CPU	1.04	

Complete List of SQL Text

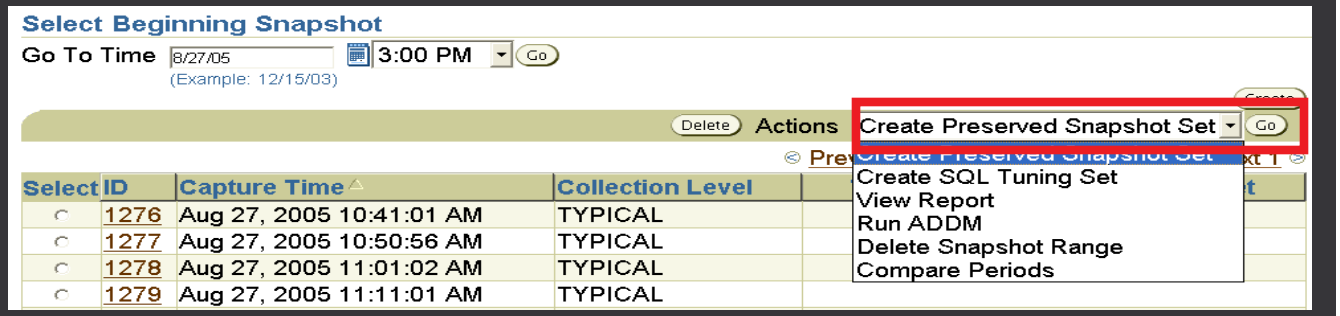
SQL Id	SQL Text
043s6zg4w225k	SELECT COUNT(*) FROM RPTHEAD_HISTORY WHERE RPTNO = :B2 AND OLD_STATUS IN (16, 10) AND NEW_STATUS = 11 AND UPD_BY = :B1
5tjfyxq1k2kaj	SELECT H.RPTNO , H.PROGRAMMER, H.STATUS, H.FIXED_DATE - H.RPTDATE AGE FROM RPTHEAD H WHERE H.FIXED_DATE >= :B2 AND H.FIXED_DATE < :B1 AND H.RPTDATE < :B1 AND CS_PRIORITY = 2 AND H.RPTDATE >= TO_DATE('01-mar-2003', 'DD-mon-yyyy') AND H.PRODUCT_ID IN (1000, 1001, 1027, 1033, 1057, 1119, 1137, 1148, 1168, 1171, 1181, 1192, 1198, 1275, 129, 1351, 166, 169, 174, 187, 2, 201, 208, 21, 214 , 229, 240, 26, 29, 31, 32, 300, 325, 332, 375, 378, 385, 397, 398, 44, 485, 495, 497, 5, 500, 501, 502, 503, 505, 507, 508, 509, 510, 512, 513, 518, 532, 533, 540, 543, 569, 571, 572, 574, 61, 662, 665, 719, 723, 740, 769, 770, 776, 777, 79, 799, 83, 930, 968, 989, 99, 996, 999, 1290, 1119, 1289, 1287, 1286, 1292, 1293)

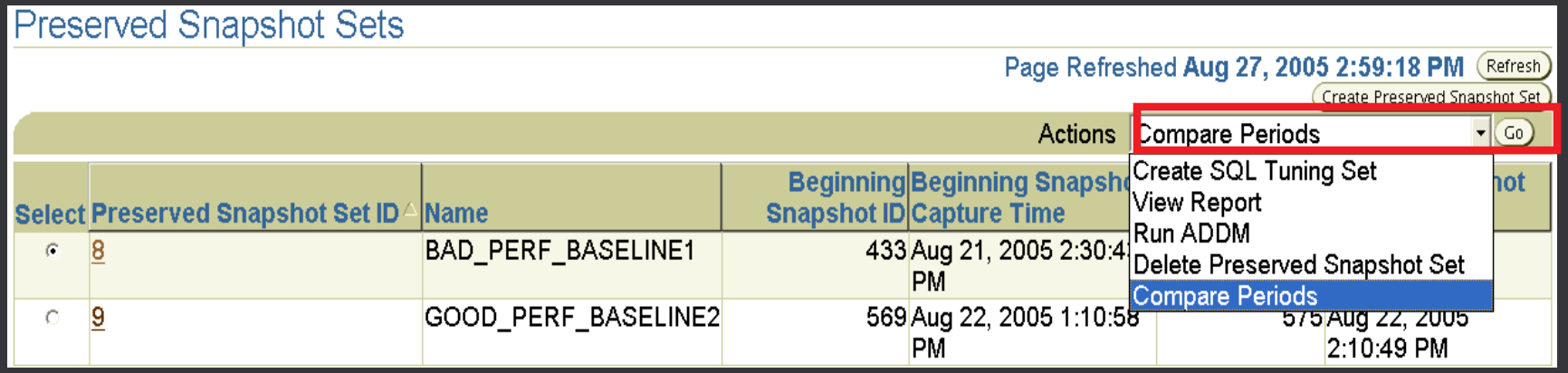
Reactive Perf. Diagnosis: AWR Compare Periods Report

- Easier and accurate diagnosis of problems due to workload or configuration variations
  - Report identifies performance attributes and database settings different
  - Compare Performance to “Baseline”
    - Create “Baseline” or Preserved Snapshot Sets
    - Compare Good Vs Bad Performance “Baselines”
  - Normalizing statistics over “dbtime” enables periods of different durations to be compared
  - Accessible through EM and command line interface

Creating Baseline and Running Compare Periods Report

Compare Periods Report Steps





Creating Baseline and Running Compare Periods Report

Manage Snapshots and Preserved Snapshot Sets

Snapshots 1300
Preserved Snapshot Sets 2
Latest Snapshot Time **Aug 27, 2005 2:40:26 PM**
Earliest Snapshot Time **Aug 18, 2005 3:01:05 PM**

Compare Periods Report Steps

1. Create Baseline (Good Perf.)
2. Create Another Baseline (Bad Perf.)

Select Beginning Snapshot

Go To Time 8/27/05 3:00 PM Go
(Example: 12/15/03)

Delete Actions Create Preserved Snapshot Set Go

Create SQL Tuning Set
View Report
Run ADDM
Delete Snapshot Range
Compare Periods

Select	ID	Capture Time	Collection Level
○	1276	Aug 27, 2005 10:41:01 AM	TYPICAL
○	1277	Aug 27, 2005 10:50:56 AM	TYPICAL
○	1278	Aug 27, 2005 11:01:02 AM	TYPICAL
○	1279	Aug 27, 2005 11:11:01 AM	TYPICAL

Preserved Snapshot Sets

Page Refreshed **Aug 27, 2005 2:59:18 PM** Refresh
Create Preserved Snapshot Set

Actions Compare Periods Go

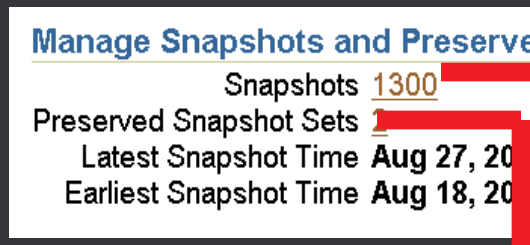
Create SQL Tuning Set
View Report
Run ADDM
Delete Preserved Snapshot Set
Compare Periods

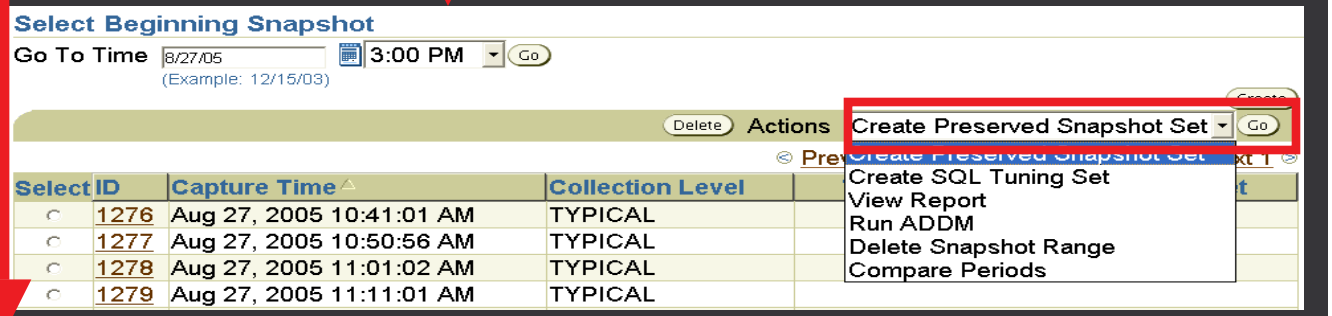
Select	Preserved Snapshot Set ID	Name	Beginning Snapshot ID	Beginning Snapsh Capture Time	...not	
◉	8	BAD_PERF_BASELINE1	433	Aug 21, 2005 2:30:4 PM		
○	9	GOOD_PERF_BASELINE2	569	Aug 22, 2005 1:10:58 PM	575	Aug 22, 2005 2:10:49 PM

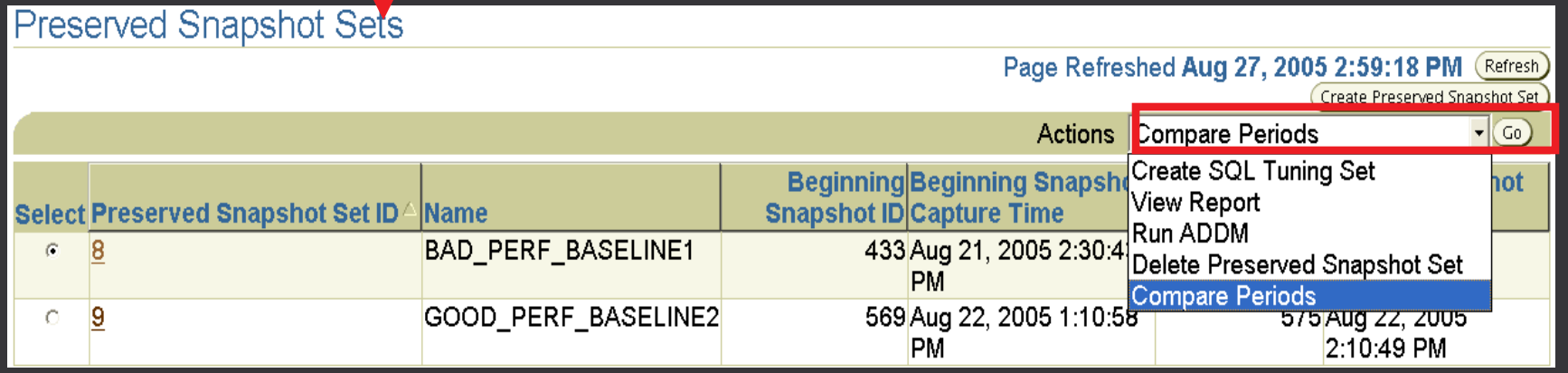
Creating Baseline and Running Compare Periods Report

Compare Periods Report Steps

1. Create Baseline (Good Perf.)
2. Create Another Baseline (Bad Perf.)
3. Run Compare Periods Report







Reactive Perf. Diagnosis: AWR Compare Periods Report (10gR2)

Top 5 Timed Events

Top Wait Events
Compared side-by-side

1st					2nd				
Event	Waits	Time(s)	Percent Total DB Time	Wait Class	Event	Waits	Time(s)	Percent Total DB Time	Wait Class
CPU time		22,191.7	32.97		enq: TX - row lock contention	1,578	3,211.1	65.73	Application
enq: TX - row lock contention	10,634	21,127.4	31.39	Application	CPU time		1,244.4	25.47	
*latch: library cache	559,941	5,302.2	7.88	Concurrency	buffer busy waits	285	145.0	2.97	Concurrency
*latch: shared pool	376,303	3,184.8	4.73	Concurrency	*db file sequential read	25,916	22.0	.45	User I/O
buffer busy waits	1,647	737.3	1.10	Concurrency	*SQL*Net message to client	4,435,906	17.2	.35	Network
-SQL*Net message to client	5,984,422	48.2	.07	Network	-latch: shared pool	33	0.7	.01	Concurrency
-db file sequential read	2,285	0.7	.00	User I/O	-latch: library cache	182	0.4	.01	Concurrency

Reactive Perf. Diagnosis: AWR Compare Periods Report (10gR2)

Top 5 Timed Events

Top Wait Events
Compared side-by-side

1st					2nd				
Event	Waits	Time(s)	Percent Total DB Time	Wait Class	Event	Waits	Time(s)	Percent Total DB Time	Wait Class
CPU time		22,191.7	32.97		enq: TX - row lock contention	1,578	3,211.1	65.73	Application
enq: TX - row lock contention	10,634	21,127.4	31.39	Application	CPU time		1,244.4	25.47	
*latch: library cache	559,941	5,302.2	7.88	Concurrency	buffer busy waits	285	145.0	2.97	Concurrency
*latch: shared pool	376,303	3,184.8	4.73	Concurrency	*db file sequential read	25,916	22.0	.45	User I/O
buffer busy waits	1,647	737.3	1.10	Concurrency	*SQL*Net message to client	4,435,906	17.2	.35	Network
-SQL*Net message to client	5,984,422	48.2	.07	Network	-latch: shared pool	33	0.7	.01	Concurrency
-db file sequential read	2,285	0.7	.00	User I/O	-latch: library cache	182	0.4	.01	Concurrency

Reactive Perf. Diagnosis: AWR Compare Periods Report (10gR2)

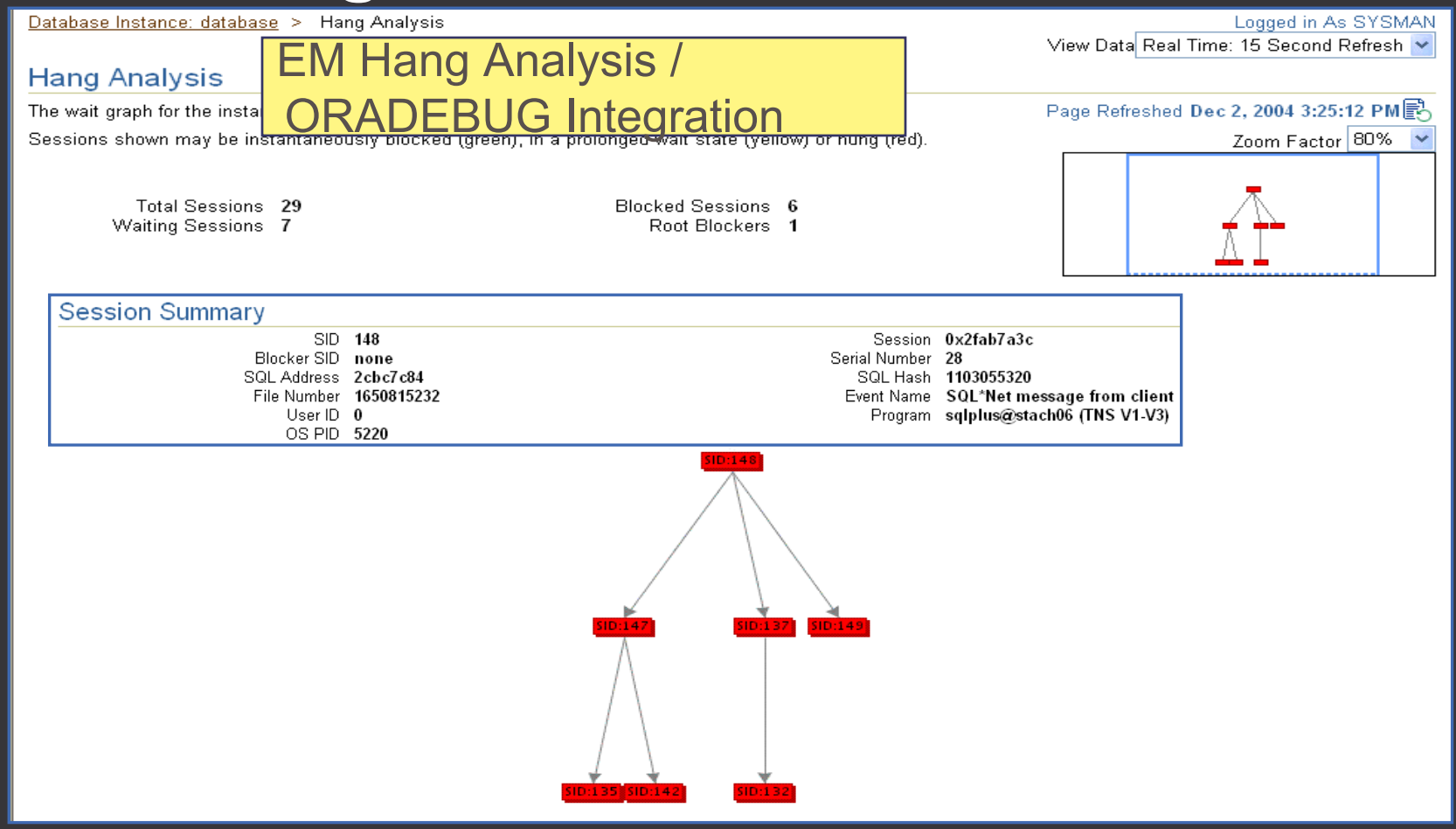
resource_manager_plan	PRI_SIM_PLAN	
resumable_timeout	0	
serial_reuse	disable	
service_names	r2e	
session_cached_cursors	20	
session_max_open_files	10	
sessions	280	
sga_max_size	629145600	
sga_target	146800640	0
shadow_core_dump	partial	
shared_memory_address	0	
shared_pool_reserved_size	4823449	2306867
shared_pool_size	12582912	222298112
shared_servers	0	
skip_unusable_indexes	TRUE	
sort_area_retained_size	0	
sort_area_size	1048576	
spfile	/scratch/pgongloo/10.2/r2e/dbs/s	
sql92_security	FALSE	

DB Configuration Settings Compared…

Reactive Perf. Diagnosis: EM Memory Access Mode

- MYTH: Since Oracle does not use Direct Memory Access, it can't be used for diagnosing hung systems
- Reality: Oracle DB 10g Release 2 does support Direct Memory Access with EM interface!
- Value proposition
  - For diagnosing hung or slow moving databases
  - SQL-based diagnostics are richer, preferable in most cases
  - EM Interface similar for Memory and SQL Access modes
- Dedicated SGA Collector reads metrics directly from the SGA memory – one per instance
- Integration with ORADEBUG Hang Analysis

Reactive Perf. Diagnosis: Diagnosing Slow/Hung Databases



Hang Analysis can be invoked from Performance Page, Related Links

Best Practices from Real-World Deployments

ORACLE

Performance Diagnosis: Best Practices

- Proactive Diagnosis
  - ADDM
  - ADDM: Run ADDM Now
- Reactive Diagnosis
  - General Manual Analysis: EM Performance Page Drill-Downs
  - Transient or Targeted Analysis: ASH Report
  - Workload Change Comparison: AWR Compare Periods Report
  - Database Hung/Slow Moving: EM Memory Access

Best Practices: AWR Vs STATSPACK

- If licensed for database Diagnostic and Tuning Packs, disable Statspack
  - AWR is a super set of Statspack
  - Running both at same time may impact performance
  - If cannot disable Statspack, run it at different times
- Do not change STATISTICS_LEVEL from default setting of TYPICAL
  - TYPICAL: Recommended, Minimal Performance Overhead
  - BASIC: Disables Self-Management, Not Recommended
  - ALL: Advanced Diagnosis, Consult Support

Oracle Database 10g: Self- Management Infrastructure Overhead

- MYTH: With 10g Self-Management Infrastructure and Non-Direct Memory Access method, performance overhead must be high!
- Reality: Oracle uses kernel structures efficiently, incurs minimal overhead: About 2% when system is fully loaded, this includes
  - Oracle 10g DB Self-Management Functionality
  - Other features governed by STATISTICS_LEVEL = TYPICAL
- Most customer deployments have experienced overall performance improvement !!
  - Validated with real world customers for large deployments in production
  - Significant net performance improvement after implementing ADDM recommendations

Best Practices: Implementing Advisor Recommendations

- When to implement ADDM or other Advisor recommendations?
  - Look for persistent problems rather than one-off scenarios
  - Gauge trends by running AWR/ADDM reports for longer time periods, for e.g., daily 8-12 noon for peak OLTP load
- SQL Tuning Sets (STS) for tracking and tuning SQL statements
  - Encompasses bind variables, execution context and important statistics
  - STS can be used for capturing workload
  - Can be transported to another system for testing (10gR2)
  - OOW 2005: Optimizing the Optimizer: Essential SQL Tuning Tips and Techniques: Will be posted on OTN shortly

Best Practices: Implementing Advisor Recommendations (contd.)

- Use rigorous change control mechanism when implementing recommendations
- Preserve interesting Advisory Task data
  - Give meaningful Task Name and Description
  - Change Task Expiration Date appropriately
- Preserve AWR raw data beyond retention using Preserved Snapshot Sets
  - Support Purposes
  - Future Compare Periods Analysis

Best Practices: Use ASH Data

- ***"Sifting through the ASHes**: Performance Analysis with the Oracle 10g Active Session History"* by Graham Wood, on OTN

- Make best use of ASH data
  - Use ASH Report to get footprint of a target (session,SQL_ID)
  - Always on, almost no overhead !!
  - Avoids workload replay in most cases!!
  - ASH simulation is available for even pre-10g databases

Best Practices: Gotchas!

- Possible AWR Library Cache Contention in few cases
  - When already incurring significant library cache contention
  - Metalink Note: 296765.1, Bugs 3941893/4133353, back port / workaround: 10.1.0.3, fixed: 10.1.0.4
- Idle SQL Forms session incorrectly appear in ASH Data
  - Bug 4137362, back port: 10.1.0.3, fixed: 10.1.0.4
- Customize Maintenance Window Time for Your Environment if Necessary
- Use SVG Plug-in for Mozilla / IE Browsers
  - http://www.adobe.com/svg/viewer/install/main.html


QUESTIONS
ANSWERS
ORACLE