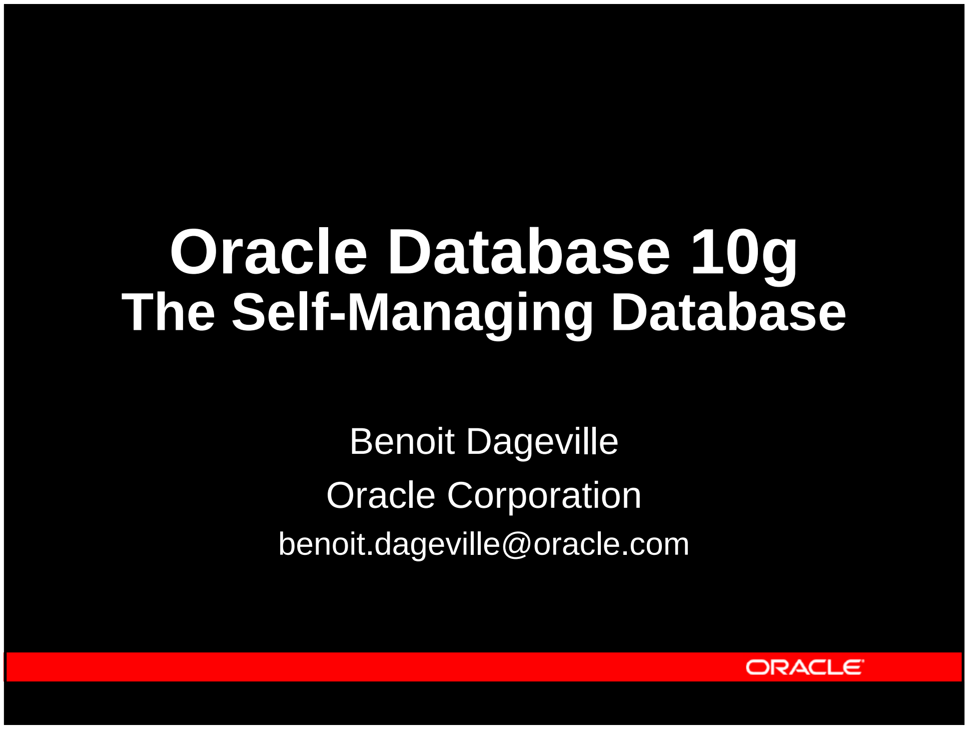
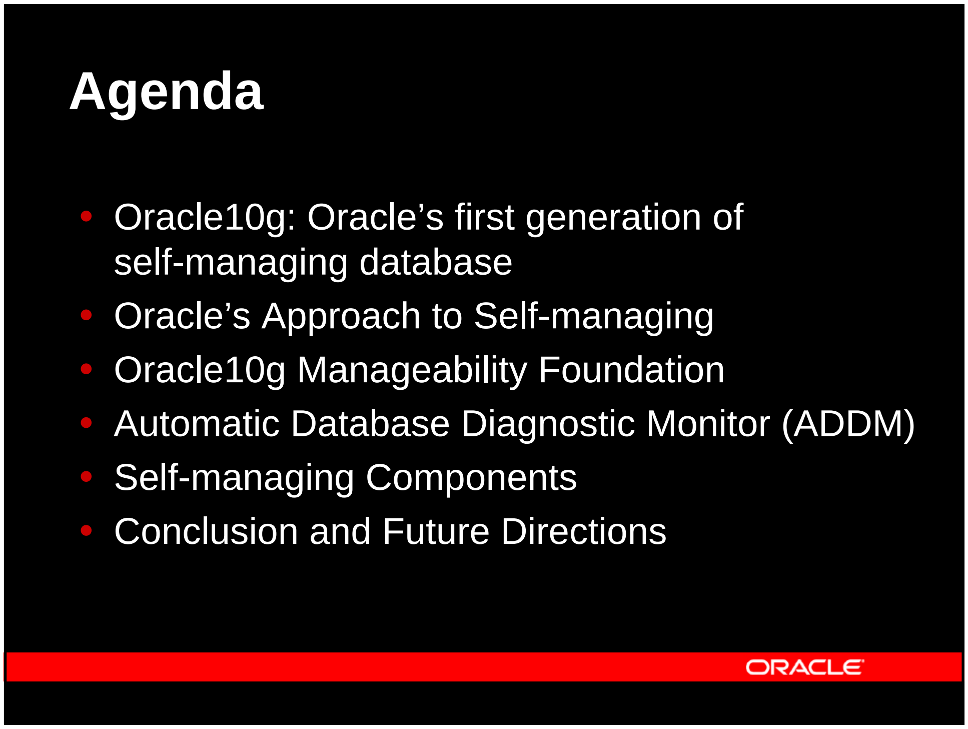
ORACLE®



Oracle Database 10g
The Self-Managing Database

Benoit Dageville
Oracle Corporation
benoit.dageville@oracle.com

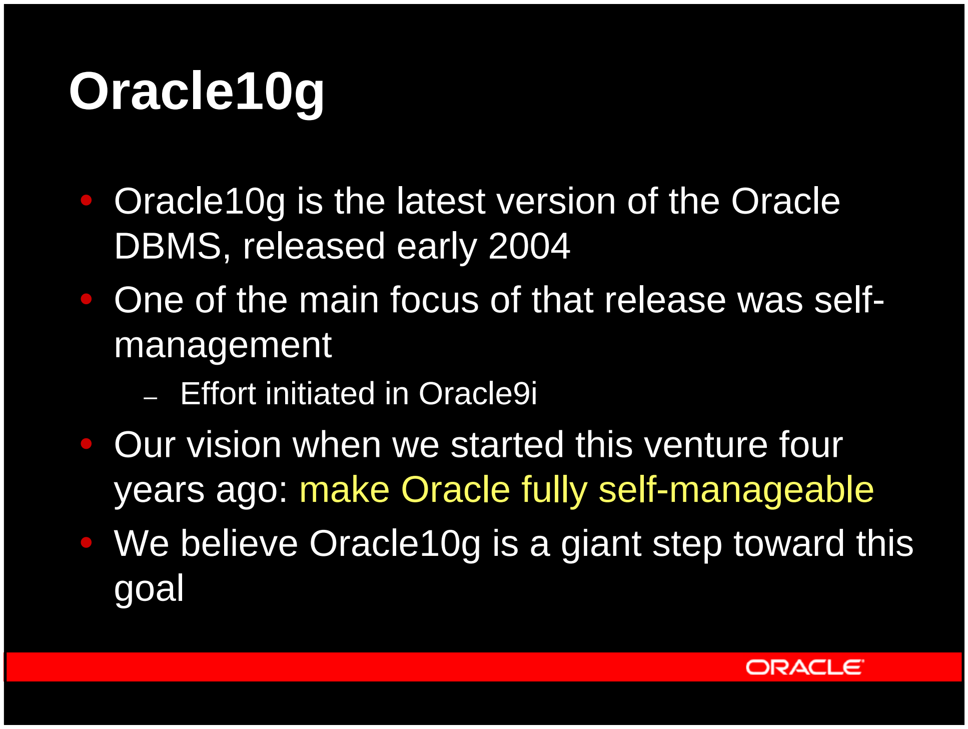
ORACLE



Agenda

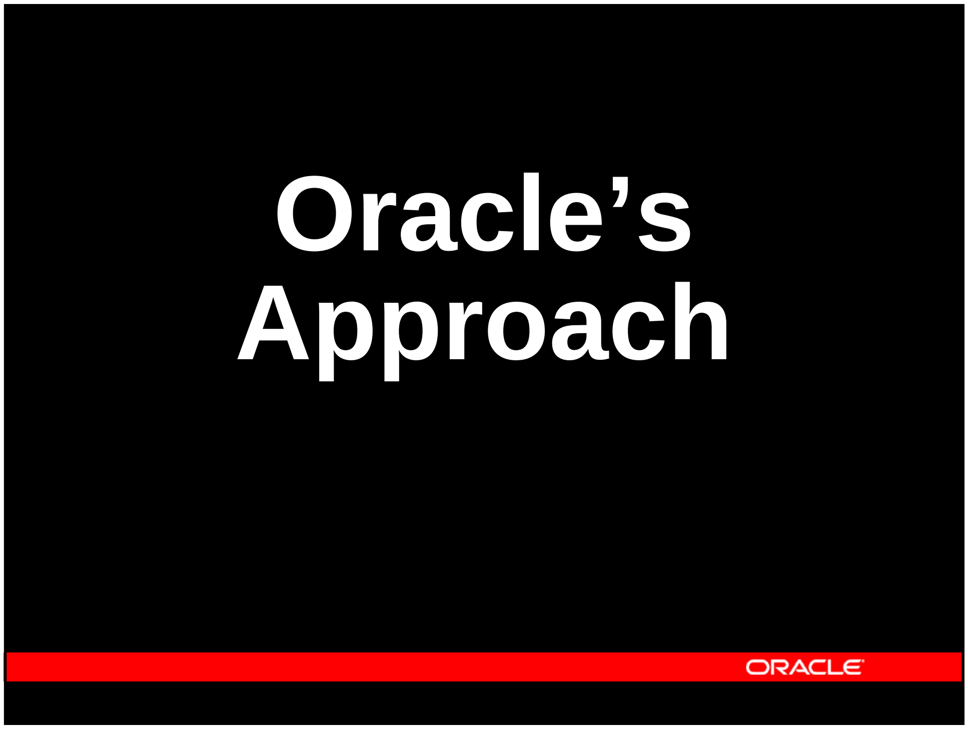
- Oracle10g: Oracle’s first generation of self-managing database
- Oracle’s Approach to Self-managing
- Oracle10g Manageability Foundation
- Automatic Database Diagnostic Monitor (ADDM)
- Self-managing Components
- Conclusion and Future Directions

Oracle10g



Oracle10g

- Oracle10g is the latest version of the Oracle DBMS, released early 2004
- One of the main focus of that release was self-management
  - Effort initiated in Oracle9i
- Our vision when we started this venture four years ago: make Oracle fully self-manageable
- We believe Oracle10g is a giant step toward this goal

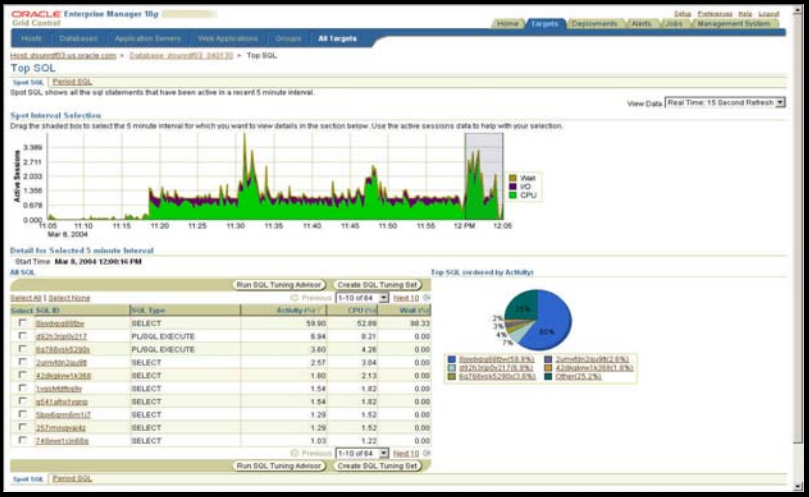


Oracle's Approach

Oracle’s Approach: Server Resident

- Technology built inside the database server
  - Eliminate management problems rather than “hiding” them behind a tool
  - Minimize Performance Impact
  - Act “Just in Time” (e.g. push versus pull)
  - Leverage existing technology
  - Effective solutions require complete integration with various server components
    - server becoming so sophisticated that a tool based solution can no longer be truly effective
  - Mandatory if the end-goal is to build a truly self-managing database server

Oracle’s Approach: Seamless GUI Integration



Host: dsunrdf03.us.oracle.com > Database: dsunrdf03_040130 > Top SQL

Top SQL

Spot SQL shows all the sql statements that have been active in a recent 5 minute interval.

Spot Interval Selection

Drag the shaded box to select the 5 minute interval for which you want to view details in the section below. Use the active sessions data to help with your selection.

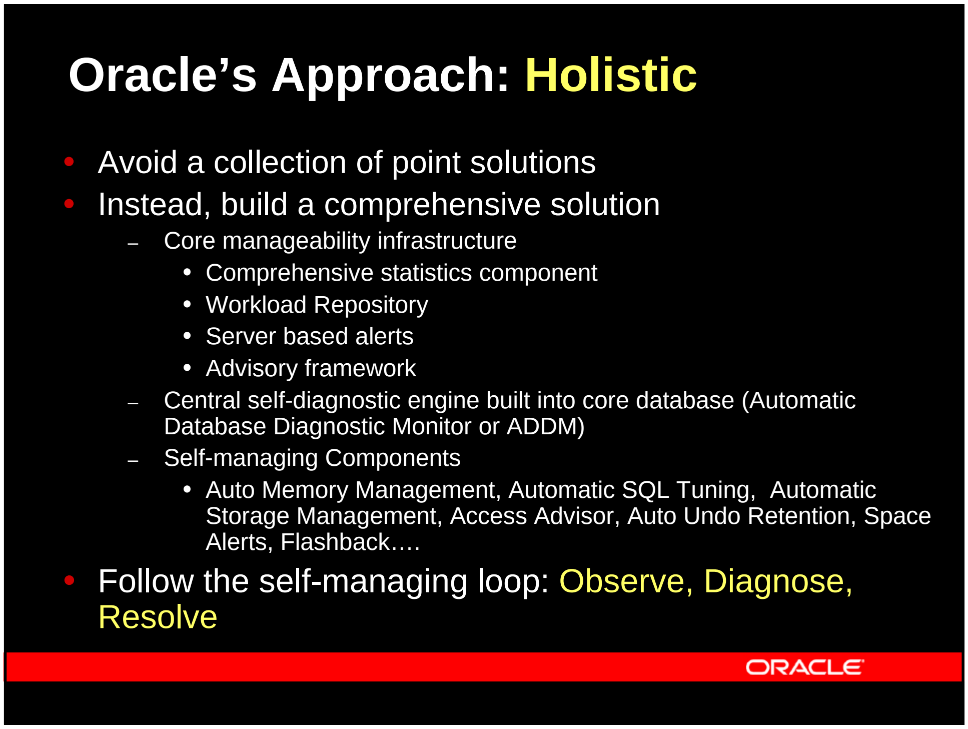
Detail for Selected 5 minute Interval

Start Time Mar 8, 2004 12:00:16 PM

All SQL

Select	SQL ID	SQL Type	Activity (%)	CPU (%)	Wait (%)
□	8bvdvpq88fbw	SELECT	59.90	52.89	98.33
□	d92h3rjp0v217	PL/SQL EXECUTE	6.94	8.21	0.00
□	6a788vsk5290s	PL/SQL EXECUTE	3.60	4.26	0.00
□	2umvfdn2qu98	SELECT	2.57	3.04	0.00
□	42dkqkvw1k368	SELECT	1.80	2.13	0.00
□	1vgsM8ffbq9v	SELECT	1.54	1.82	0.00
□	g541afhz1vgnp	SELECT	1.54	1.82	0.00
□	5bw6qrm6m1j7	SELECT	1.29	1.52	0.00
□	257mmtqvjai4z	SELECT	1.29	1.52	0.00
□	748ewr1cjn68q	SELECT	1.03	1.22	0.00

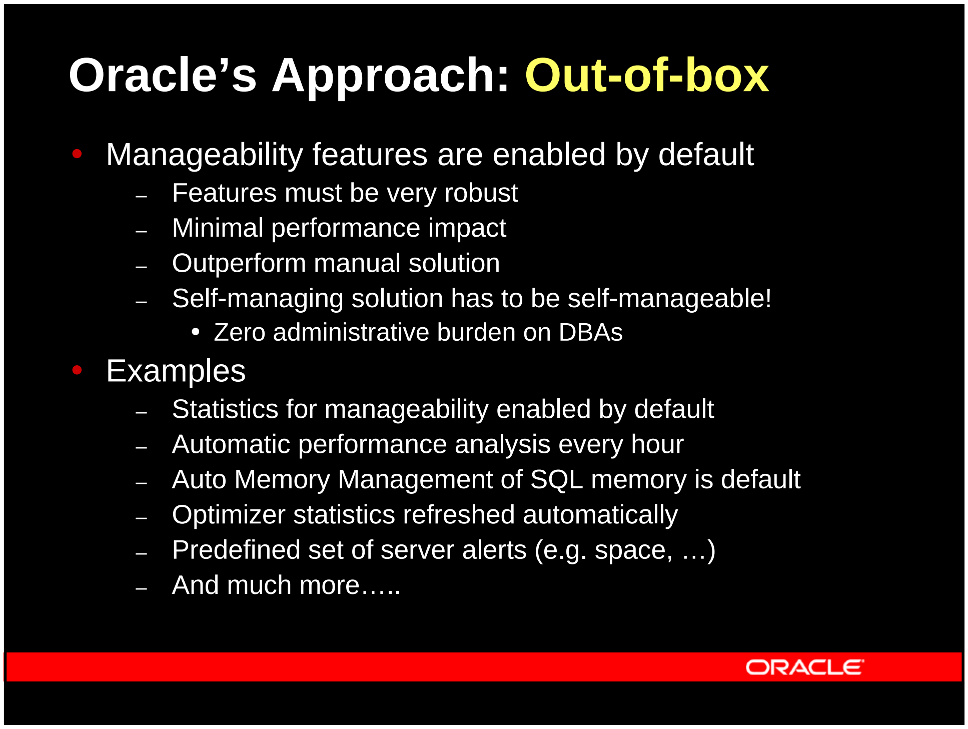
Top SQL (ordered by Activity)



Oracle's Approach: Holistic

- Avoid a collection of point solutions
- Instead, build a comprehensive solution
  - Core manageability infrastructure
    - Comprehensive statistics component
    - Workload Repository
    - Server based alerts
    - Advisory framework
  - Central self-diagnostic engine built into core database (Automatic Database Diagnostic Monitor or ADDM)
  - Self-managing Components
    - Auto Memory Management, Automatic SQL Tuning, Automatic Storage Management, Access Advisor, Auto Undo Retention, Space Alerts, Flashback….
- Follow the self-managing loop: Observe, Diagnose, Resolve

ORACLE



Oracle's Approach: Out-of-box

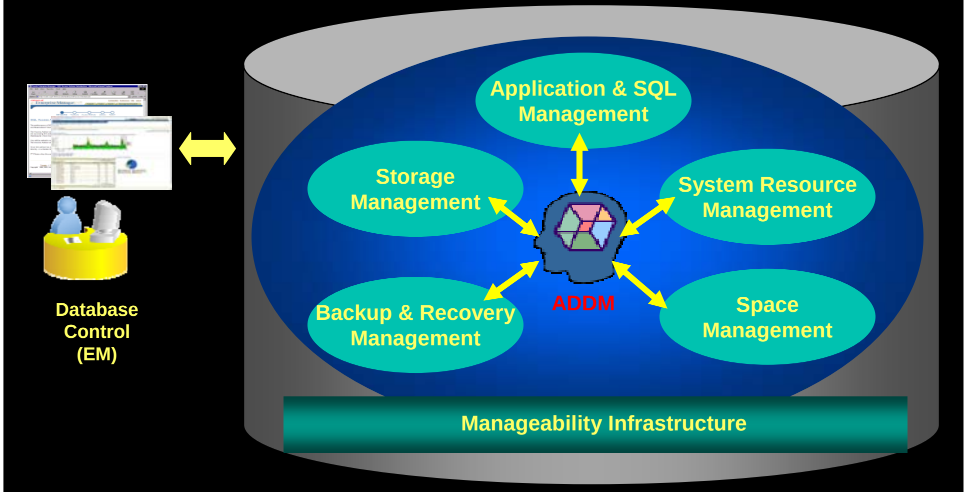
- Manageability features are enabled by default
  - Features must be very robust
  - Minimal performance impact
  - Outperform manual solution
  - Self-managing solution has to be self-manageable!
    - Zero administrative burden on DBAs
- Examples
  - Statistics for manageability enabled by default
  - Automatic performance analysis every hour
  - Auto Memory Management of SQL memory is default
  - Optimizer statistics refreshed automatically
  - Predefined set of server alerts (e.g. space, …)
  - And much more…..

ORACLE

Oracle’s Approach: Manageability for All

- Low End Customers
  - No dedicated administrative staff
  - Automated day to day operations
  - ➔ Optimal performance out of the box, no need to set configuration parameters
- High End Customers
  - Flexibility to adapt product to their needs
  - Self-management features should outperform manual tuning and ensure predictable behavior
  - Need to understand and monitor functioning of self-management operations
  - ➔ Help DBAs in making administrative decisions (no need for DBA to be rocket scientist!)
- Any workload: OLTP, DSS, mixed

Oracle’s Approach: Manageability Architecture



Database Control (EM)

Application & SQL Management

Storage Management

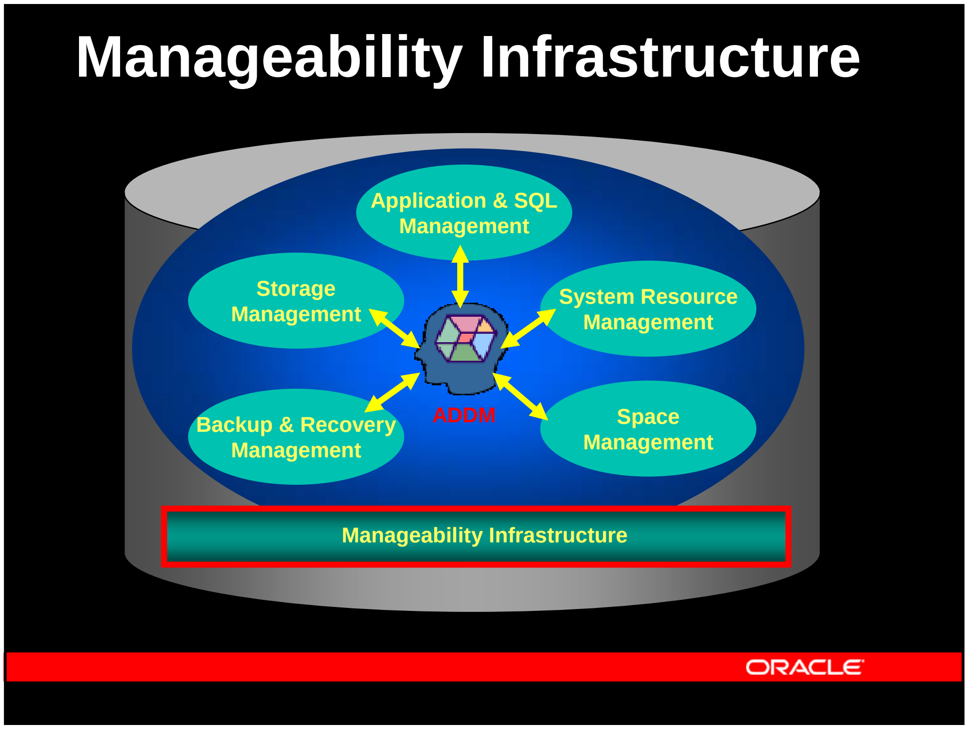
System Resource Management

ADDM

Backup & Recovery Management

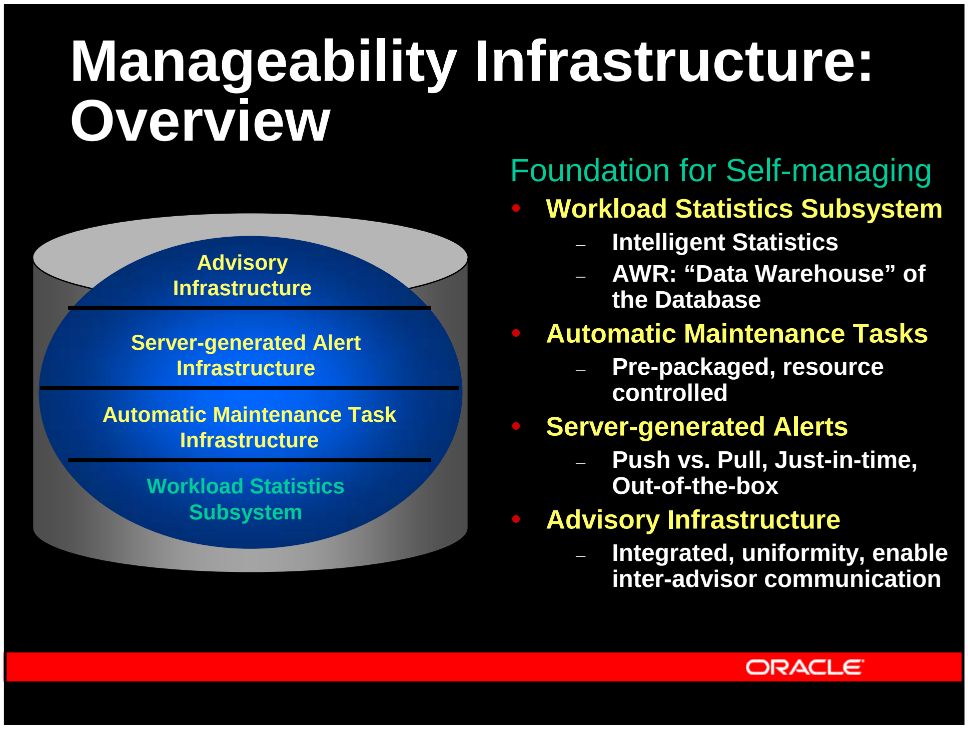
Space Management

Manageability Infrastructure



Manageability Infrastructure

Application & SQL Management
Storage Management
System Resource Management
ADDM
Backup & Recovery Management
Space Management
Manageability Infrastructure
ORACLE

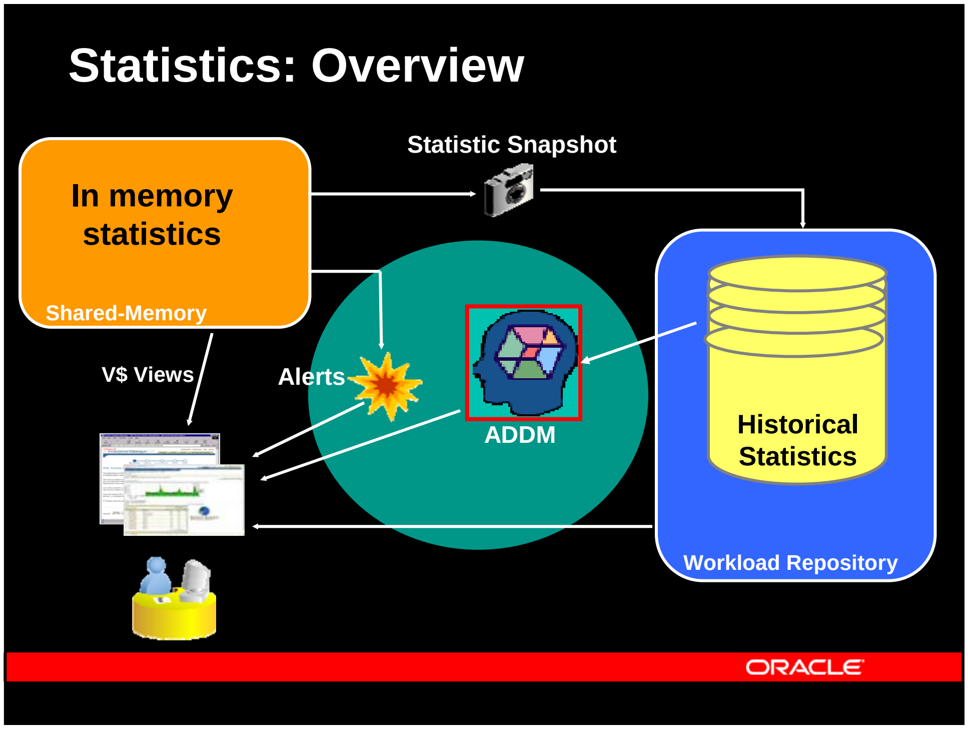


Manageability Infrastructure: Overview

Advisory Infrastructure
Server-generated Alert Infrastructure
Automatic Maintenance Task Infrastructure
Workload Statistics Subsystem

Foundation for Self-managing

- Workload Statistics Subsystem
  - Intelligent Statistics
  - AWR: “Data Warehouse” of the Database
- Automatic Maintenance Tasks
  - Pre-packaged, resource controlled
- Server-generated Alerts
  - Push vs. Pull, Just-in-time, Out-of-the-box
- Advisory Infrastructure
  - Integrated, uniformity, enable inter-advisor communication



Statistics: Overview

Statistic Snapshot

In memory statistics

Shared-Memory

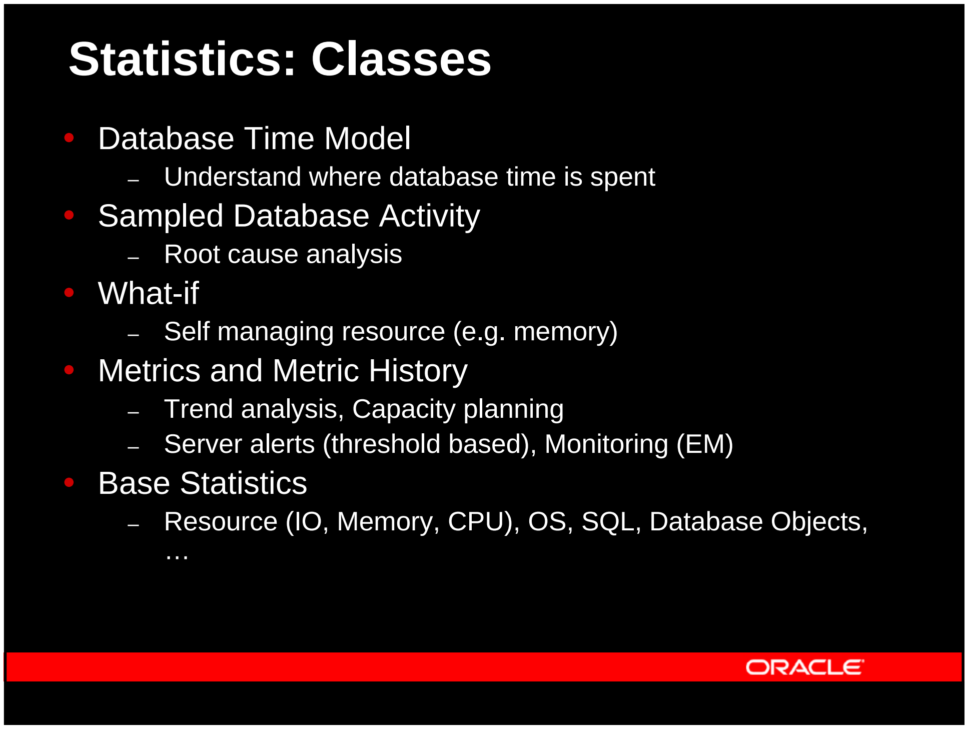
V$ Views

Alerts

ADDM

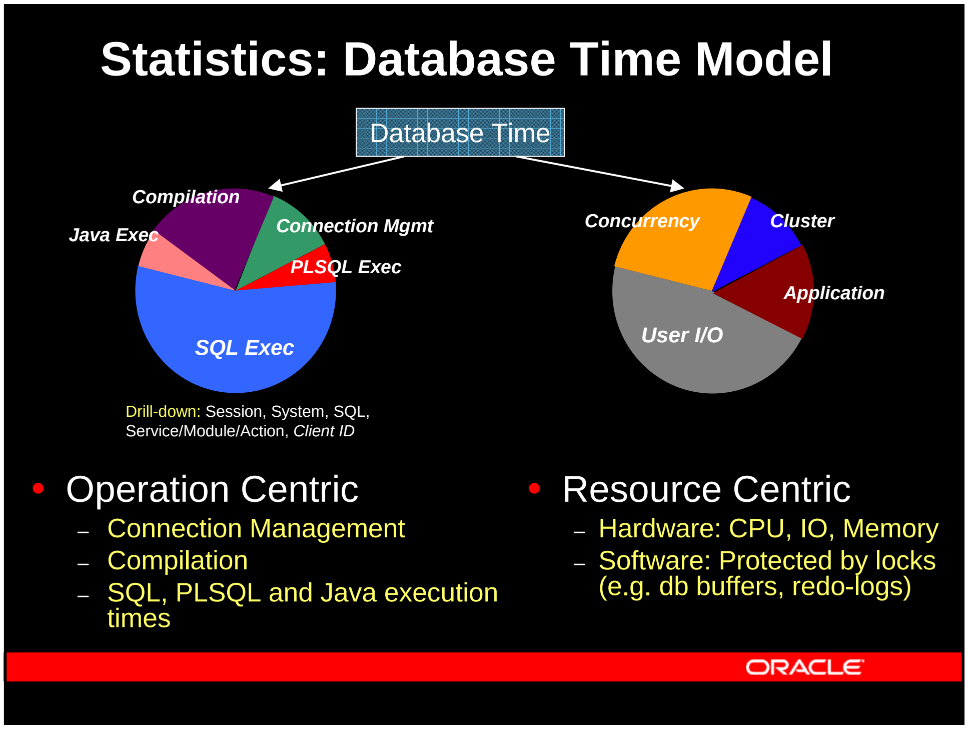
Historical Statistics

Workload Repository



Statistics: Classes

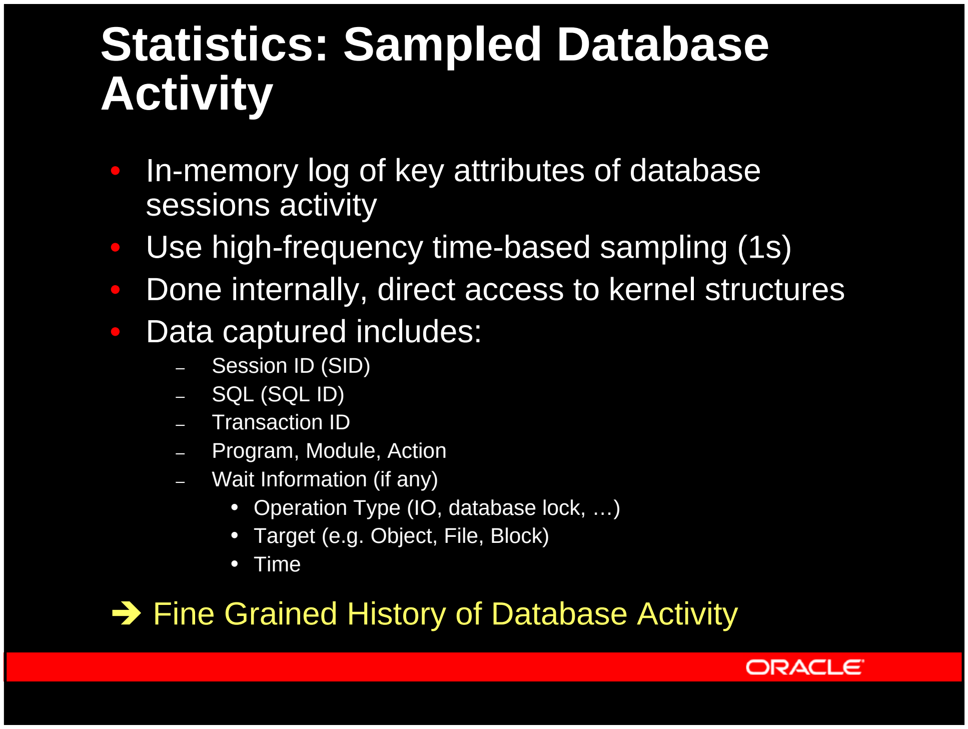
- Database Time Model
  - Understand where database time is spent
- Sampled Database Activity
  - Root cause analysis
- What-if
  - Self managing resource (e.g. memory)
- Metrics and Metric History
  - Trend analysis, Capacity planning
  - Server alerts (threshold based), Monitoring (EM)
- Base Statistics
  - Resource (IO, Memory, CPU), OS, SQL, Database Objects, …



Statistics: Database Time Model

Drill-down: Session, System, SQL, Service/Module/Action, *Client ID*

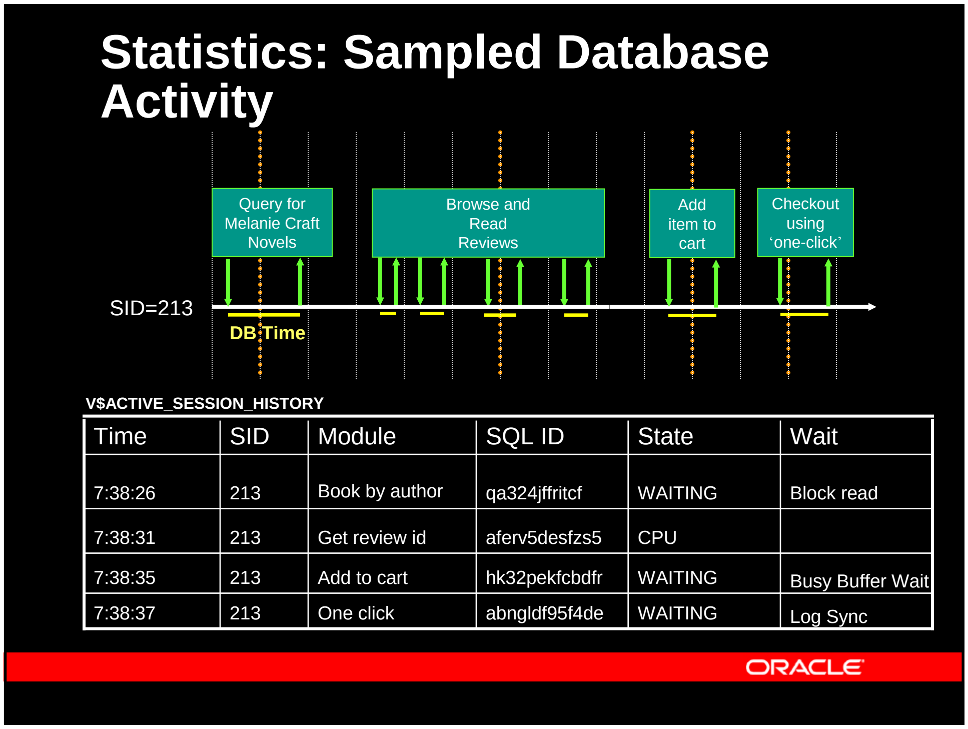
- Operation Centric
  - Connection Management
  - Compilation
  - SQL, PLSQL and Java execution times
- Resource Centric
  - Hardware: CPU, IO, Memory
  - Software: Protected by locks (e.g. db buffers, redo-logs)



Statistics: Sampled Database Activity

- In-memory log of key attributes of database sessions activity
- Use high-frequency time-based sampling (1s)
- Done internally, direct access to kernel structures
- Data captured includes:
  - Session ID (SID)
  - SQL (SQL ID)
  - Transaction ID
  - Program, Module, Action
  - Wait Information (if any)
    - Operation Type (IO, database lock, …)
    - Target (e.g. Object, File, Block)
    - Time

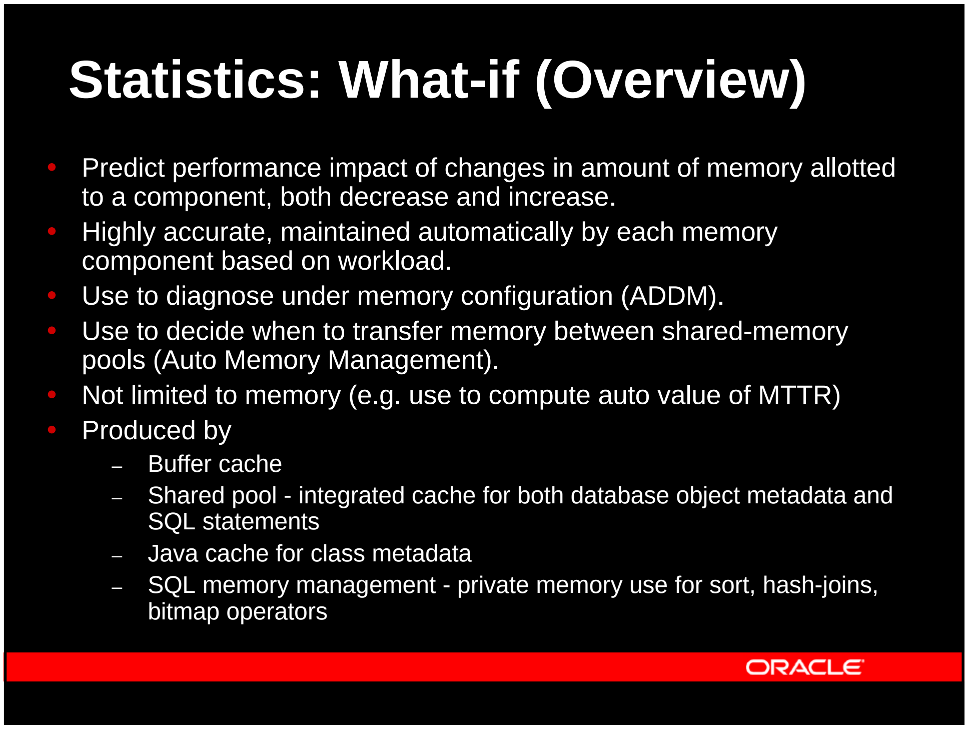
➔ Fine Grained History of Database Activity



Statistics: Sampled Database Activity

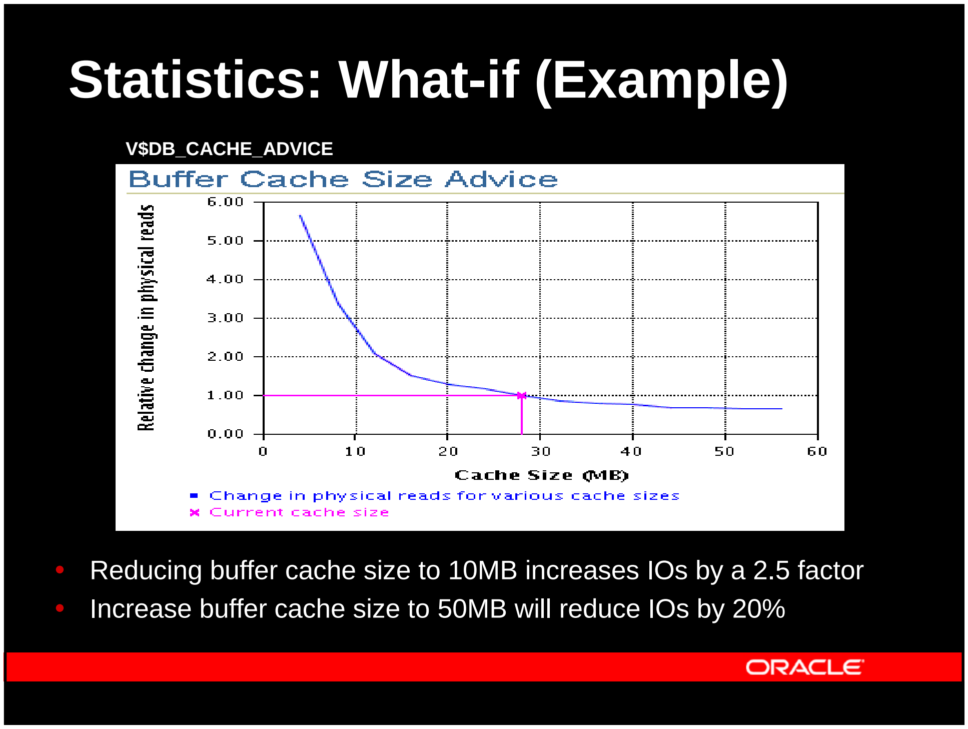
V$ACTIVE_SESSION_HISTORY

Time	SID	Module	SQL ID	State	Wait
7:38:26	213	Book by author	qa324jffritcf	WAITING	Block read
7:38:31	213	Get review id	aferv5desfzs5	CPU	
7:38:35	213	Add to cart	hk32pekfcbdfr	WAITING	Busy Buffer Wait
7:38:37	213	One click	abngldf95f4de	WAITING	Log Sync



Statistics: What-if (Overview)

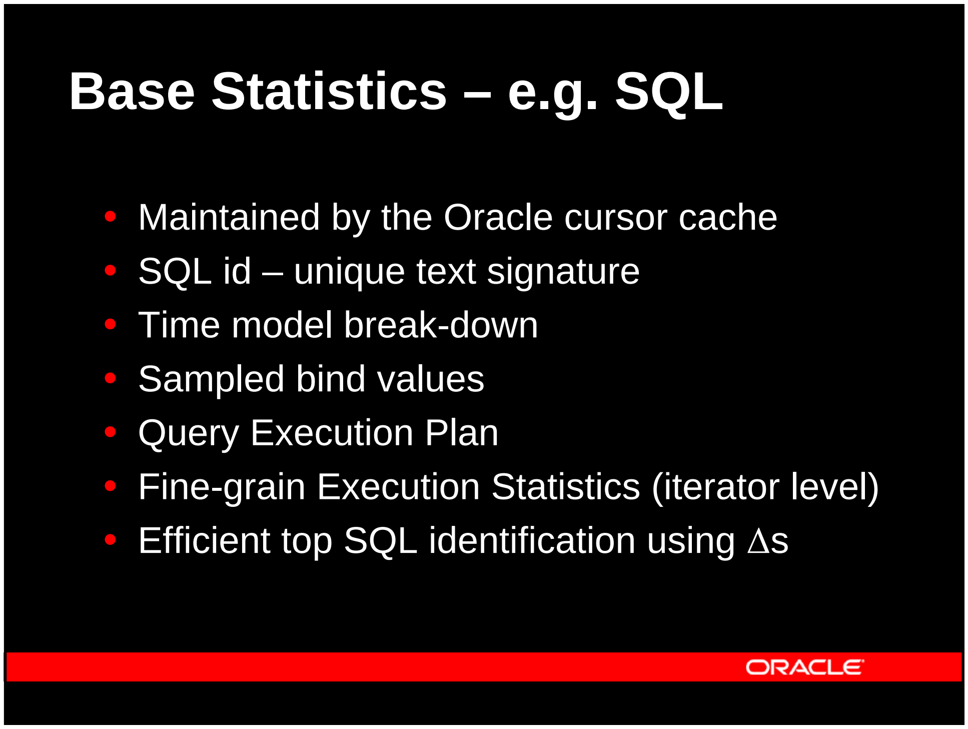
- Predict performance impact of changes in amount of memory allotted to a component, both decrease and increase.
- Highly accurate, maintained automatically by each memory component based on workload.
- Use to diagnose under memory configuration (ADDM).
- Use to decide when to transfer memory between shared-memory pools (Auto Memory Management).
- Not limited to memory (e.g. use to compute auto value of MTTR)
- Produced by
  - Buffer cache
  - Shared pool - integrated cache for both database object metadata and SQL statements
  - Java cache for class metadata
  - SQL memory management - private memory use for sort, hash-joins, bitmap operators



Statistics: What-if (Example)

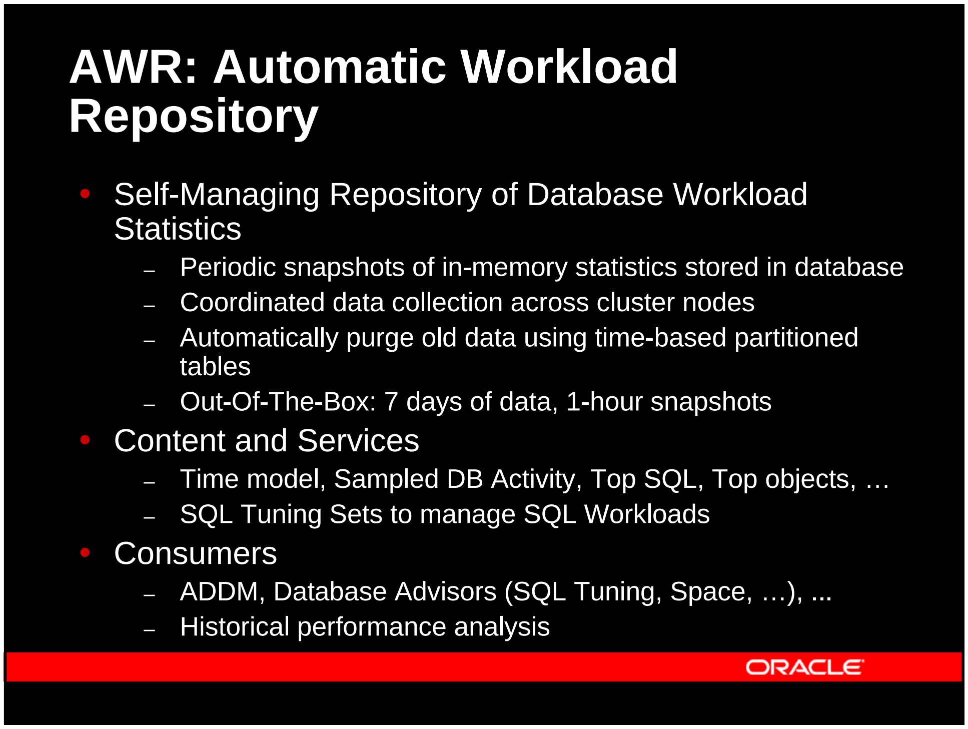
V$DB_CACHE_ADVICE

- Reducing buffer cache size to 10MB increases IOs by a 2.5 factor
- Increase buffer cache size to 50MB will reduce IOs by 20%



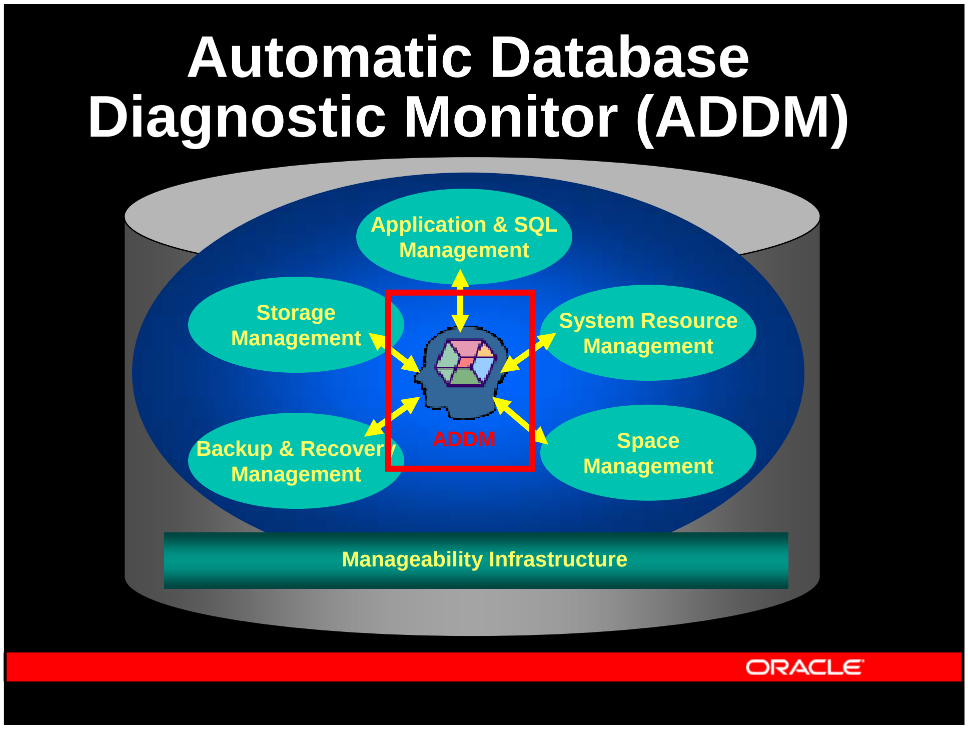
Base Statistics – e.g. SQL

- Maintained by the Oracle cursor cache
- SQL id – unique text signature
- Time model break-down
- Sampled bind values
- Query Execution Plan
- Fine-grain Execution Statistics (iterator level)
- Efficient top SQL identification using Δs



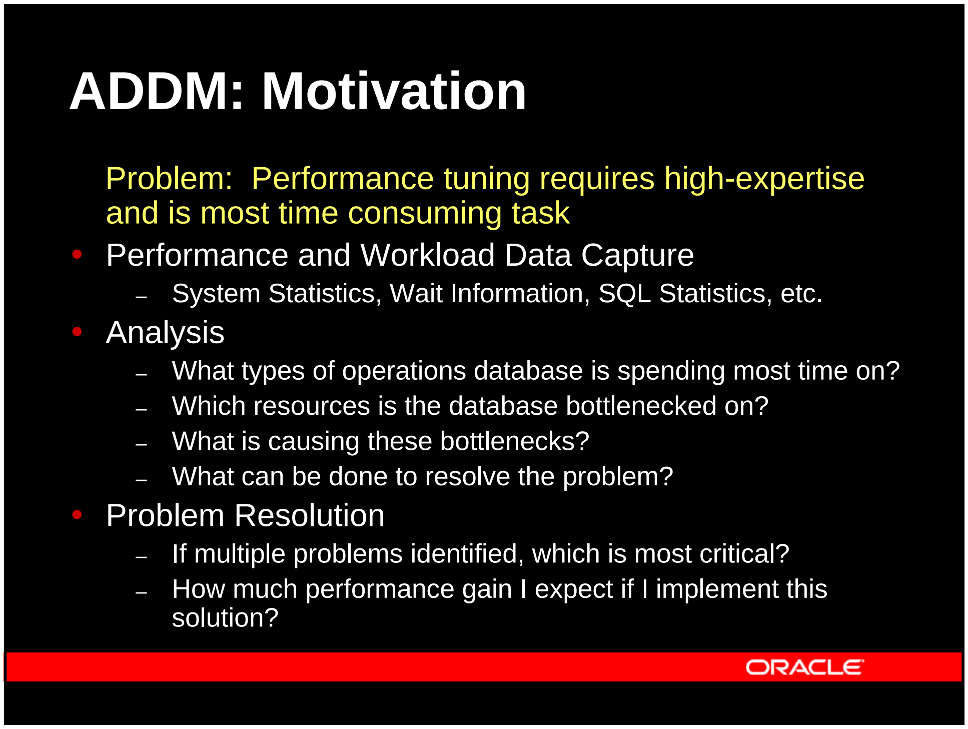
AWR: Automatic Workload Repository

- Self-Managing Repository of Database Workload Statistics
  - Periodic snapshots of in-memory statistics stored in database
  - Coordinated data collection across cluster nodes
  - Automatically purge old data using time-based partitioned tables
  - Out-Of-The-Box: 7 days of data, 1-hour snapshots
- Content and Services
  - Time model, Sampled DB Activity, Top SQL, Top objects, …
  - SQL Tuning Sets to manage SQL Workloads
- Consumers
  - ADDM, Database Advisors (SQL Tuning, Space, …), ...
  - Historical performance analysis



Automatic Database Diagnostic Monitor (ADDM)

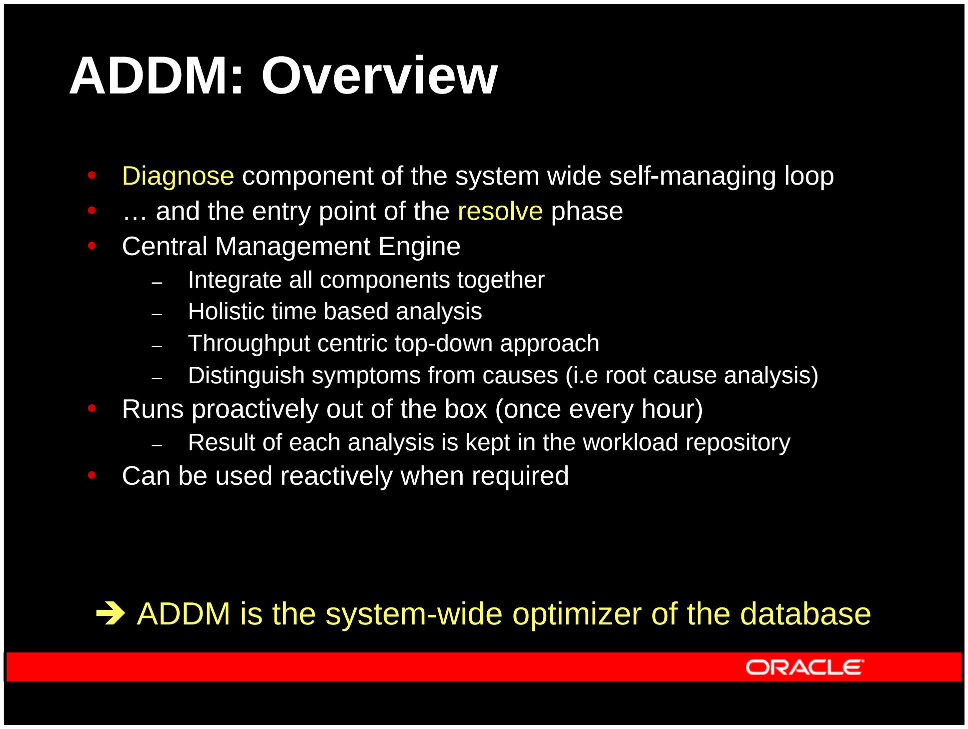
Application & SQL Management
Storage Management
System Resource Management
ADDM
Backup & Recovery Management
Space Management
Manageability Infrastructure



ADDM: Motivation

Problem: Performance tuning requires high-expertise and is most time consuming task

- Performance and Workload Data Capture
  - System Statistics, Wait Information, SQL Statistics, etc.
- Analysis
  - What types of operations database is spending most time on?
  - Which resources is the database bottlenecked on?
  - What is causing these bottlenecks?
  - What can be done to resolve the problem?
- Problem Resolution
  - If multiple problems identified, which is most critical?
  - How much performance gain I expect if I implement this solution?

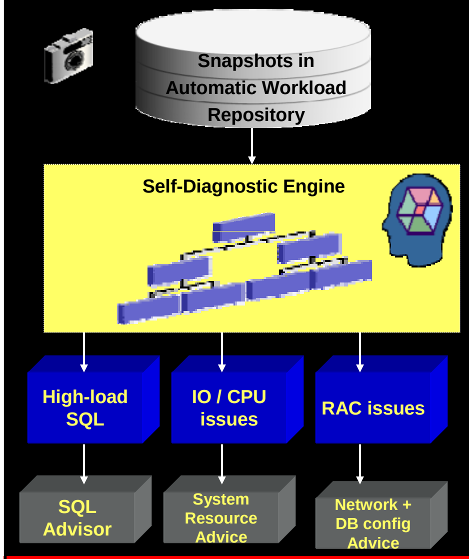


ADDM: Overview

- Diagnose component of the system wide self-managing loop
- … and the entry point of the resolve phase
- Central Management Engine
  - Integrate all components together
  - Holistic time based analysis
  - Throughput centric top-down approach
  - Distinguish symptoms from causes (i.e root cause analysis)
- Runs proactively out of the box (once every hour)
  - Result of each analysis is kept in the workload repository
- Can be used reactively when required

➔ ADDM is the system-wide optimizer of the database

How Does ADDM Work?



Snapshots in Automatic Workload Repository

Self-Diagnostic Engine

High-load SQL

IO / CPU issues

RAC issues

SQL Advisor

System Resource Advice

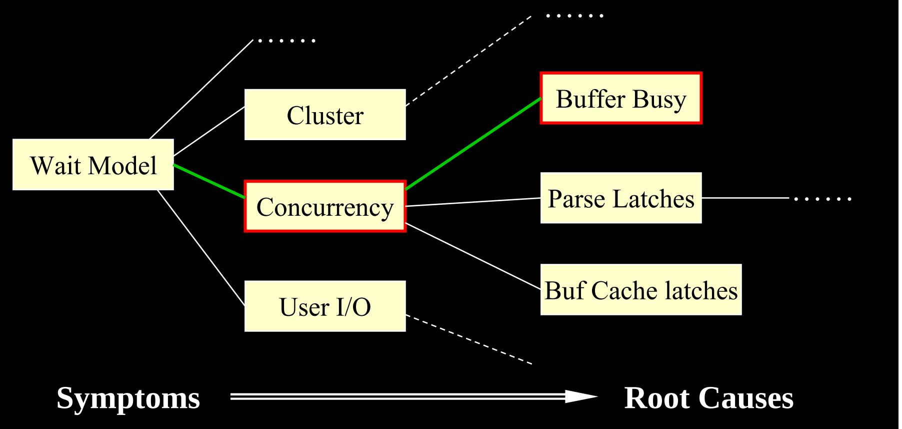
Network + DB config Advice

- Top Down Analysis Using AWR Snapshots
- Classification Tree - based on decades of Oracle tuning expertise
- Identifies main performance bottlenecks using time based analysis
- Pinpoints root cause
- Recommend solutions or next step
- Reports non-problem areas
  - E.g. I/O is not a problem

ADDM: Methodology

Problem classification system

- Decision tree based on the Wait Model and Time Model



Wait Model

Cluster

Concurrency

User I/O

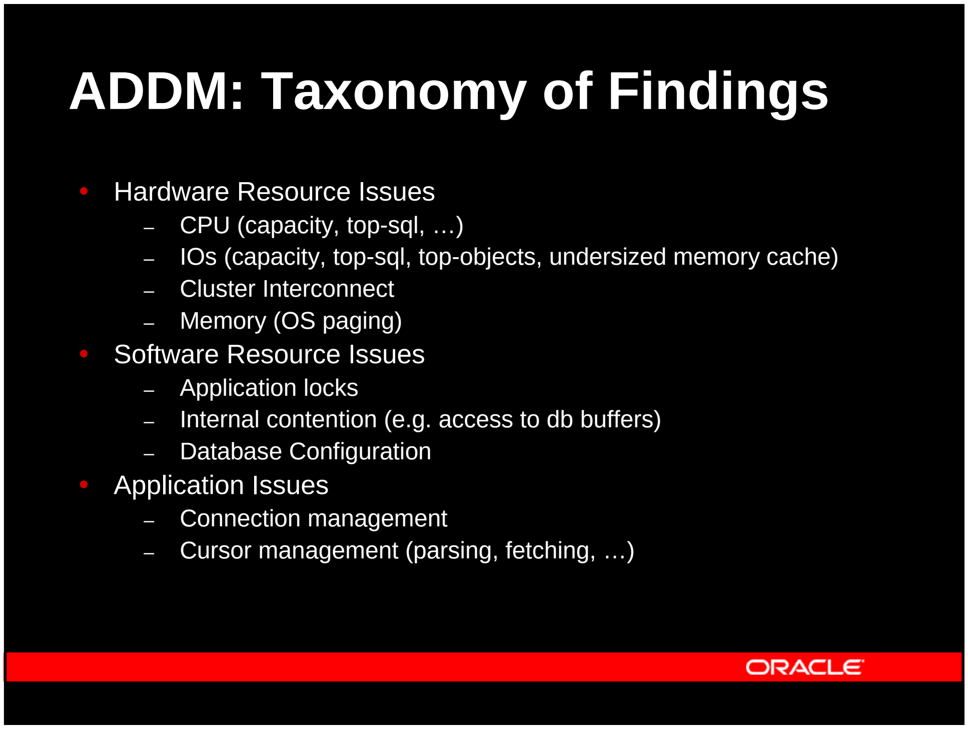
Buffer Busy

Parse Latches

Buf Cache latches

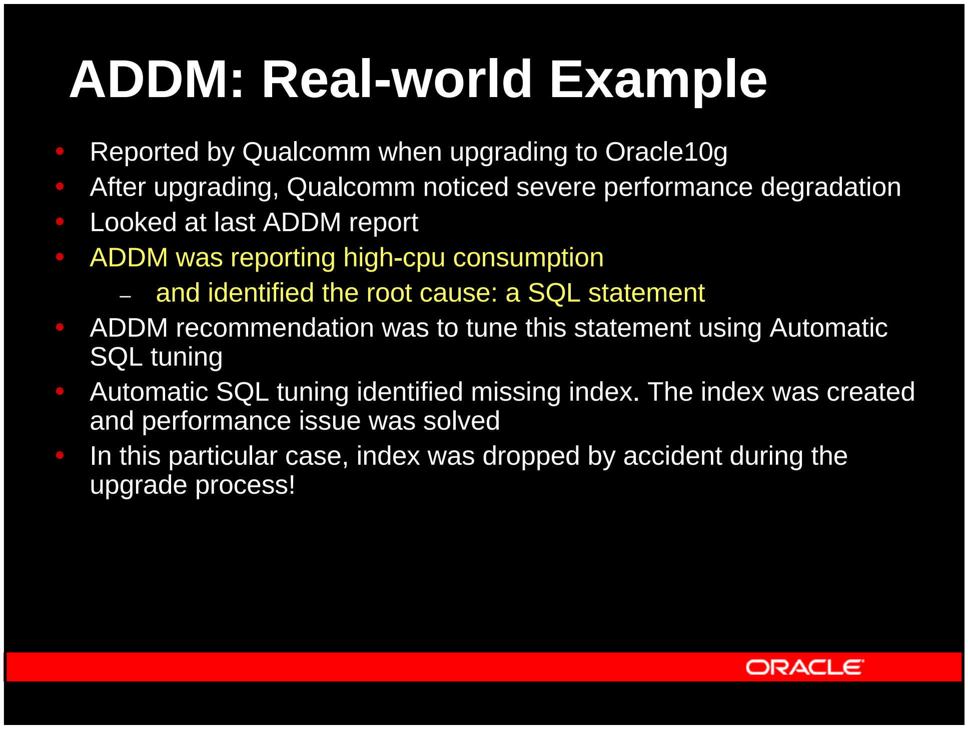
Symptoms

Root Causes



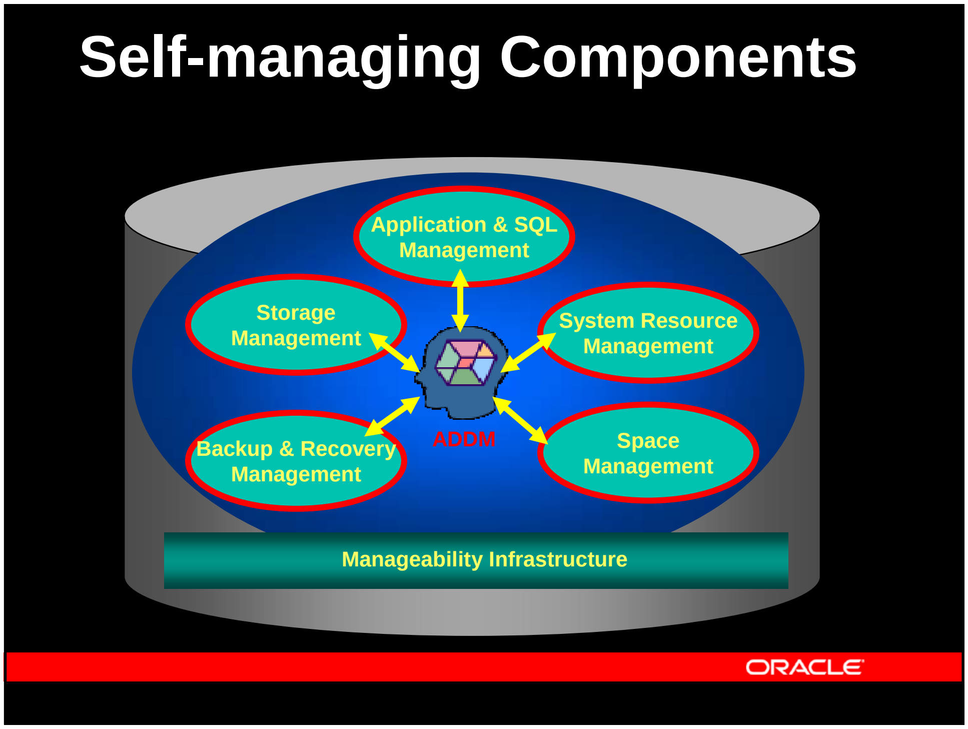
ADDM: Taxonomy of Findings

- Hardware Resource Issues
  - CPU (capacity, top-sql, …)
  - IOs (capacity, top-sql, top-objects, undersized memory cache)
  - Cluster Interconnect
  - Memory (OS paging)
- Software Resource Issues
  - Application locks
  - Internal contention (e.g. access to db buffers)
  - Database Configuration
- Application Issues
  - Connection management
  - Cursor management (parsing, fetching, …)



ADDM: Real-world Example

- Reported by Qualcomm when upgrading to Oracle10g
- After upgrading, Qualcomm noticed severe performance degradation
- Looked at last ADDM report
- ADDM was reporting high-cpu consumption
  - and identified the root cause: a SQL statement
- ADDM recommendation was to tune this statement using Automatic SQL tuning
- Automatic SQL tuning identified missing index. The index was created and performance issue was solved
- In this particular case, index was dropped by accident during the upgrade process!



Self-managing Components

Application & SQL Management

Storage Management

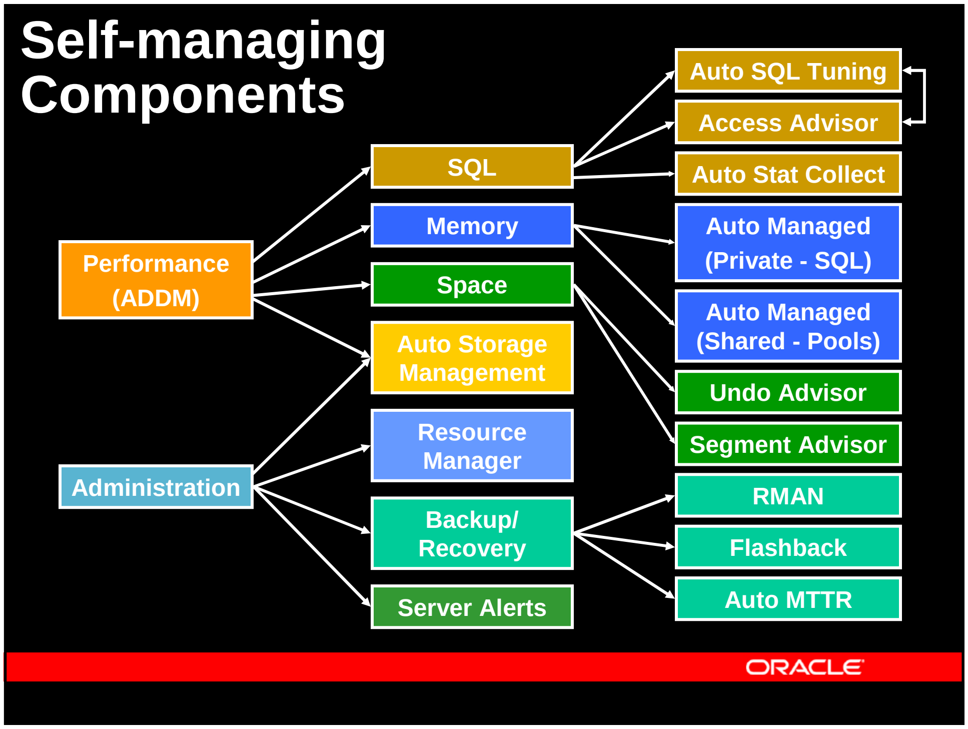
System Resource Management

ADDM

Backup & Recovery Management

Space Management

Manageability Infrastructure



Self-managing Components

Performance (ADDM)

Administration

SQL

Memory

Space

Auto Storage Management

Resource Manager

Backup/ Recovery

Server Alerts

Auto SQL Tuning

Access Advisor

Auto Stat Collect

Auto Managed (Private - SQL)

Auto Managed (Shared - Pools)

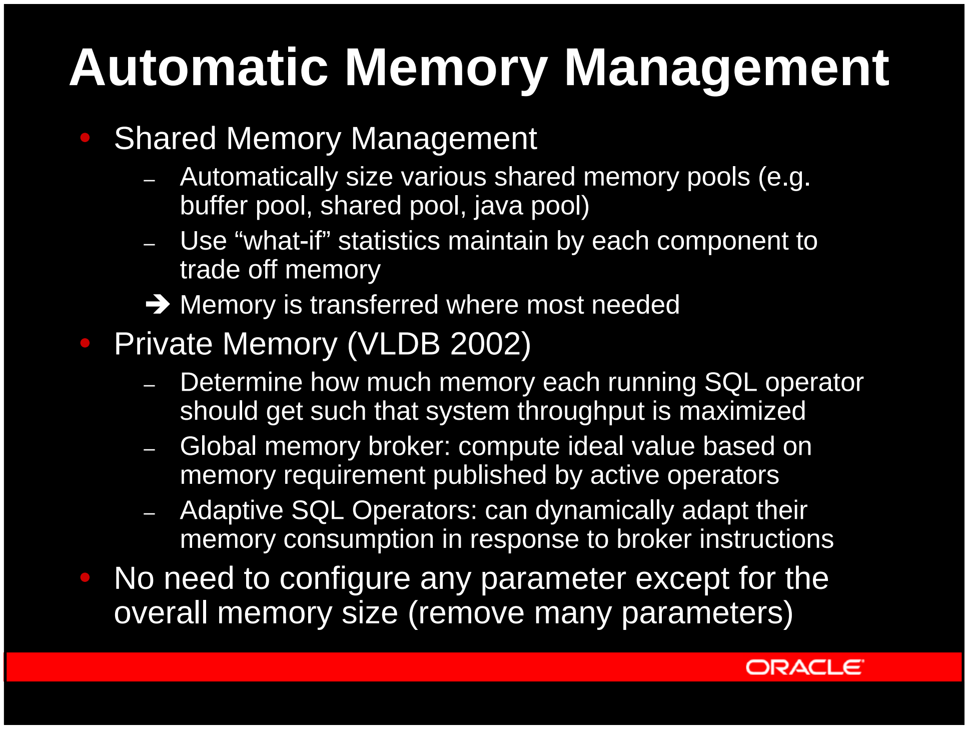
Undo Advisor

Segment Advisor

RMAN

Flashback

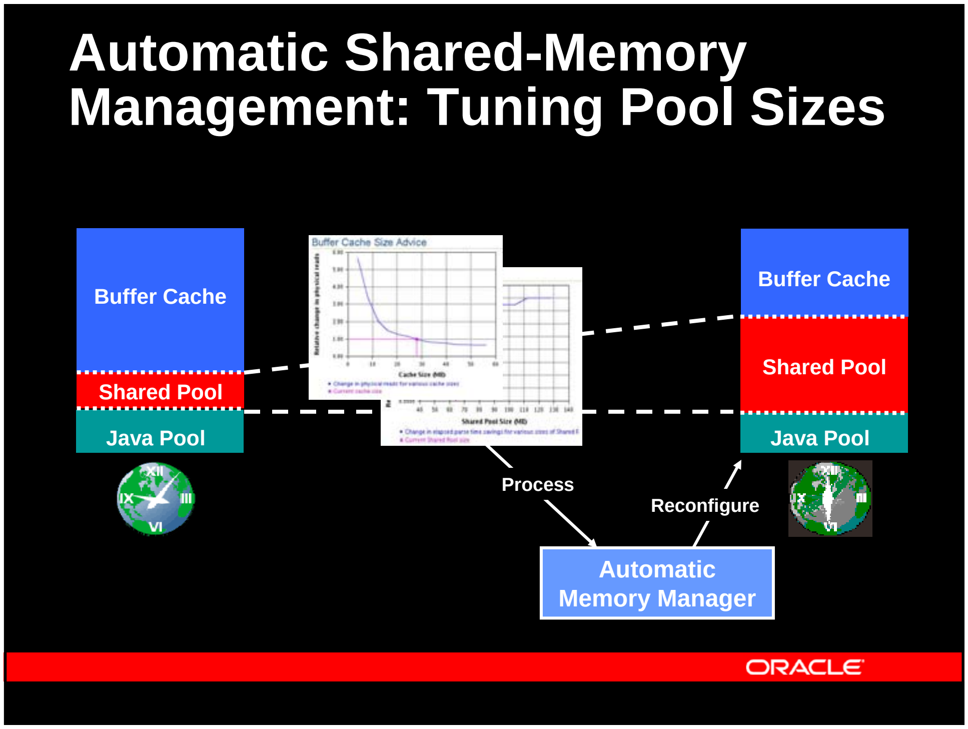
Auto MTTR



Automatic Memory Management

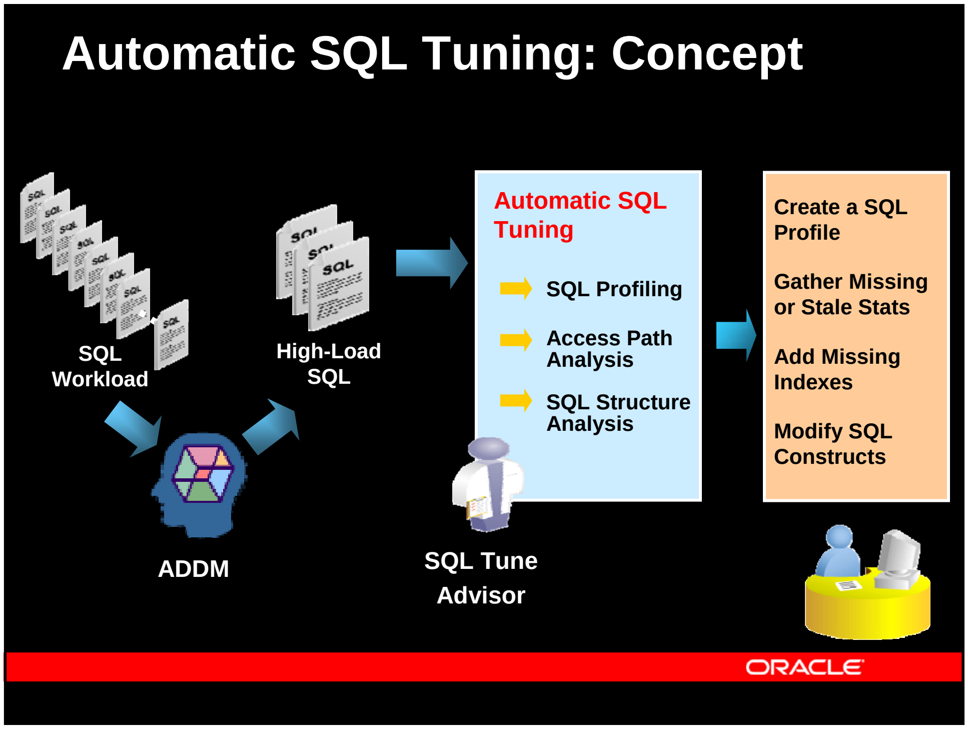
- Shared Memory Management
  - Automatically size various shared memory pools (e.g. buffer pool, shared pool, java pool)
  - Use “what-if” statistics maintain by each component to trade off memory
  - ➔ Memory is transferred where most needed
- Private Memory (VLDB 2002)
  - Determine how much memory each running SQL operator should get such that system throughput is maximized
  - Global memory broker: compute ideal value based on memory requirement published by active operators
  - Adaptive SQL Operators: can dynamically adapt their memory consumption in response to broker instructions
- No need to configure any parameter except for the overall memory size (remove many parameters)

ORACLE



Automatic Shared-Memory Management: Tuning Pool Sizes

ORACLE



Automatic SQL Tuning: Concept

SQL Workload

High-Load SQL

ADDM

Automatic SQL Tuning

- SQL Profiling
- Access Path Analysis
- SQL Structure Analysis

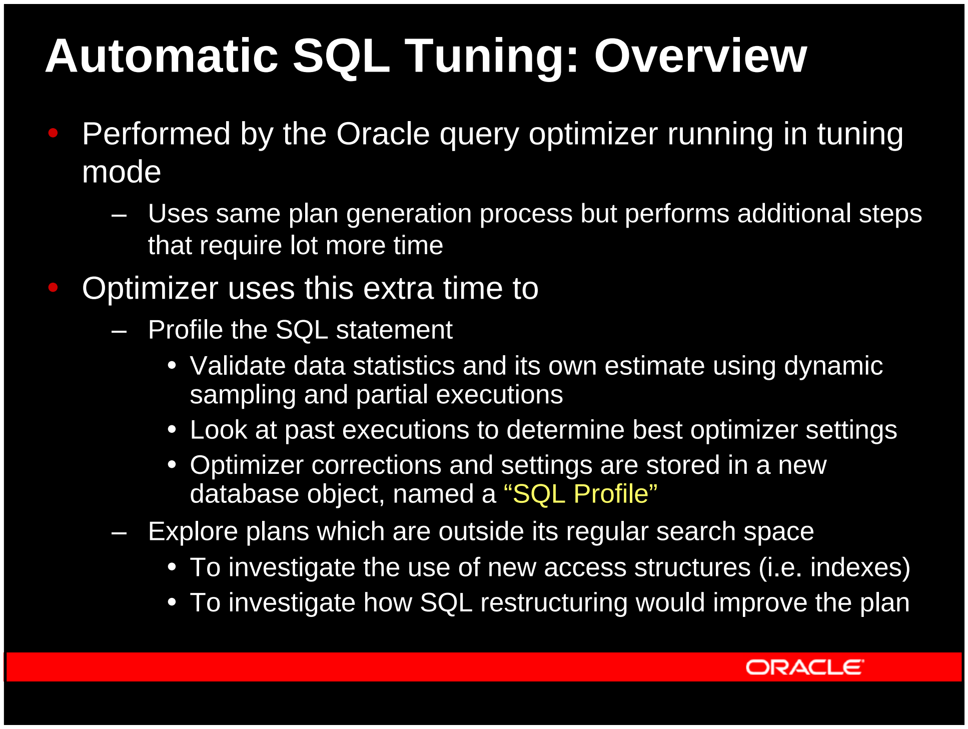
SQL Tune Advisor

Create a SQL Profile

Gather Missing or Stale Stats

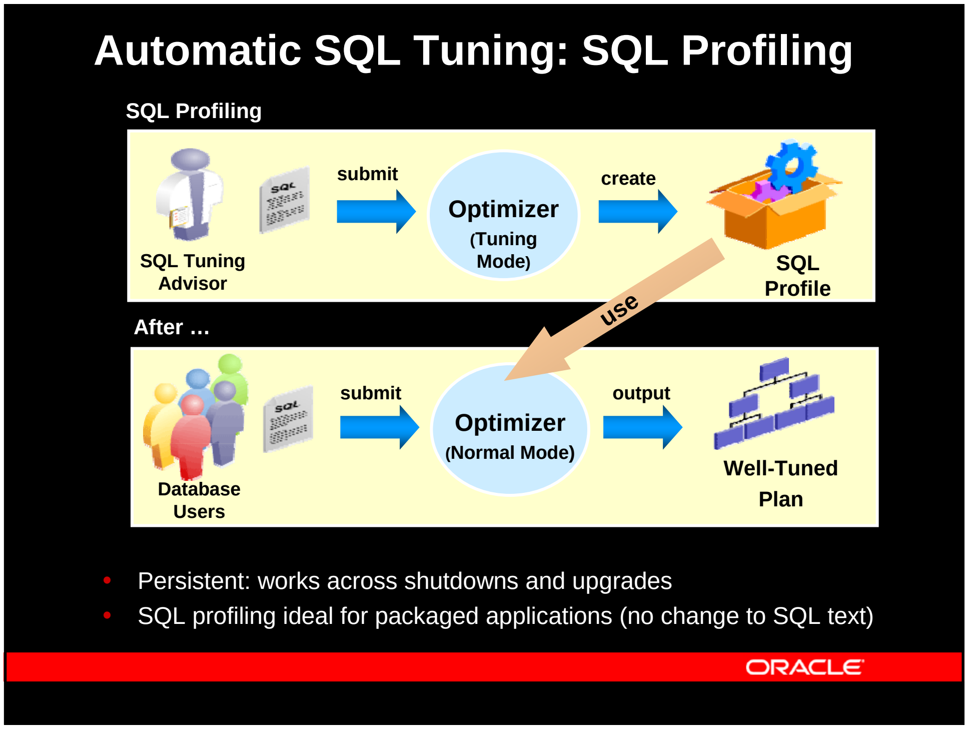
Add Missing Indexes

Modify SQL Constructs



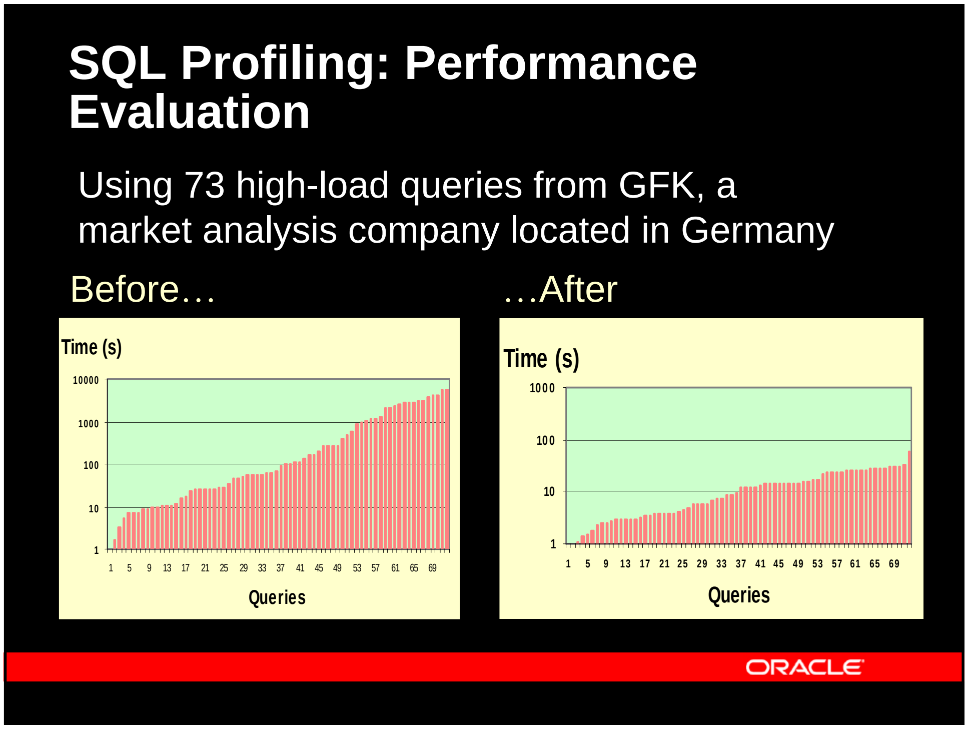
Automatic SQL Tuning: Overview

- Performed by the Oracle query optimizer running in tuning mode
  - Uses same plan generation process but performs additional steps that require lot more time
- Optimizer uses this extra time to
  - Profile the SQL statement
    - Validate data statistics and its own estimate using dynamic sampling and partial executions
    - Look at past executions to determine best optimizer settings
    - Optimizer corrections and settings are stored in a new database object, named a “SQL Profile”
  - Explore plans which are outside its regular search space
    - To investigate the use of new access structures (i.e. indexes)
    - To investigate how SQL restructuring would improve the plan



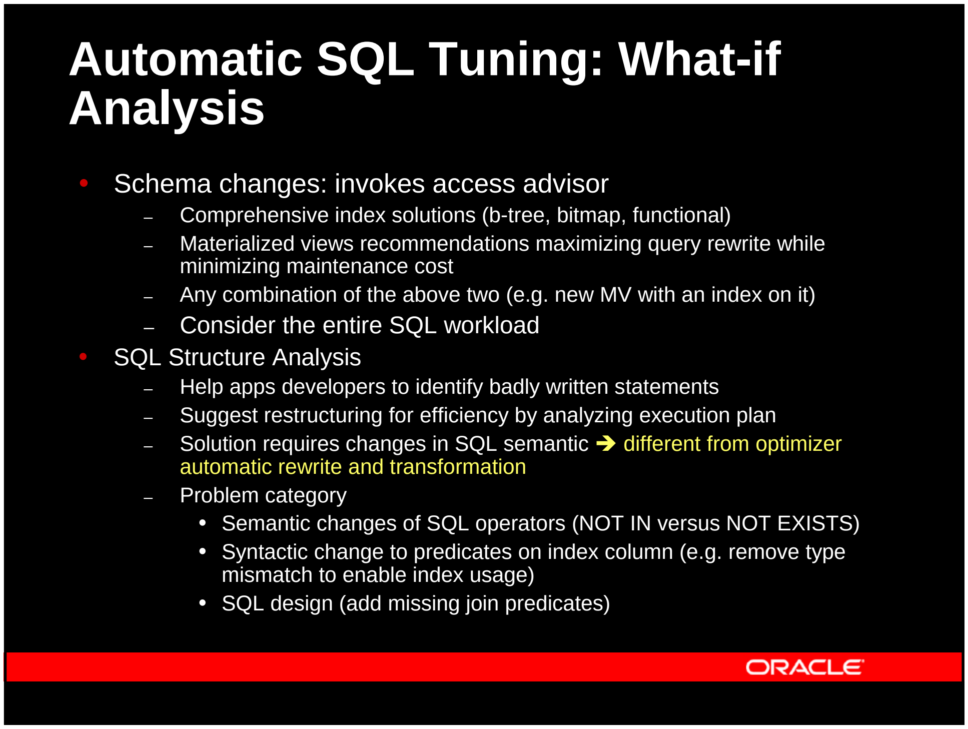
Automatic SQL Tuning: SQL Profiling

- Persistent: works across shutdowns and upgrades
- SQL profiling ideal for packaged applications (no change to SQL text)



SQL Profiling: Performance Evaluation

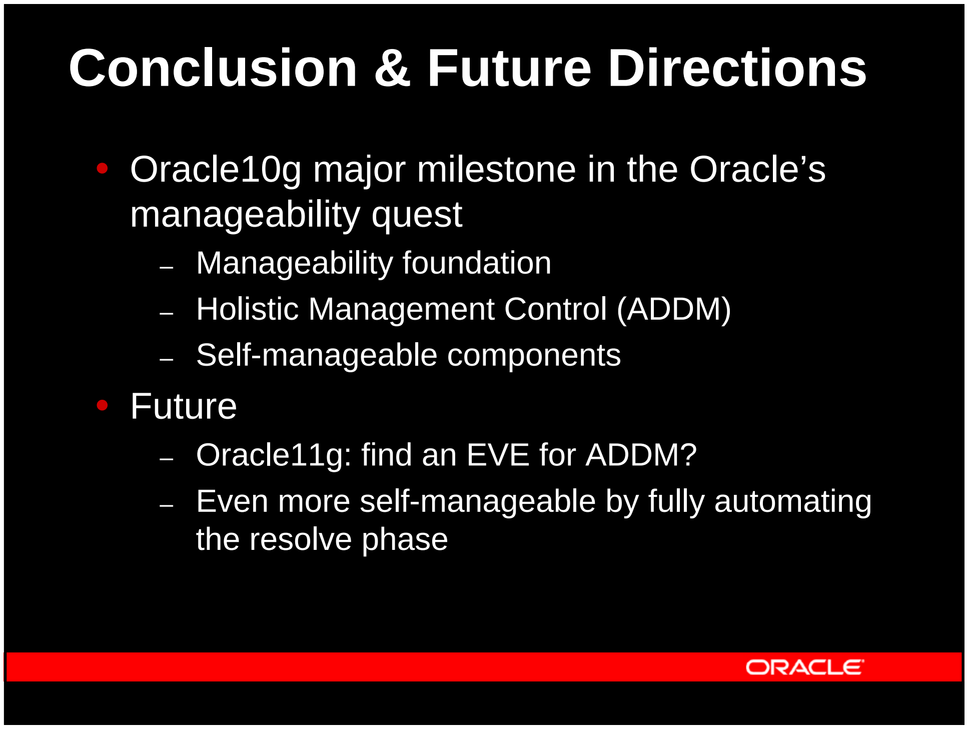
Using 73 high-load queries from GFK, a market analysis company located in Germany



Automatic SQL Tuning: What-if Analysis

- Schema changes: invokes access advisor
  - Comprehensive index solutions (b-tree, bitmap, functional)
  - Materialized views recommendations maximizing query rewrite while minimizing maintenance cost
  - Any combination of the above two (e.g. new MV with an index on it)
  - Consider the entire SQL workload
- SQL Structure Analysis
  - Help apps developers to identify badly written statements
  - Suggest restructuring for efficiency by analyzing execution plan
  - Solution requires changes in SQL semantic ➔ different from optimizer automatic rewrite and transformation
  - Problem category
    - Semantic changes of SQL operators (NOT IN versus NOT EXISTS)
    - Syntactic change to predicates on index column (e.g. remove type mismatch to enable index usage)
    - SQL design (add missing join predicates)

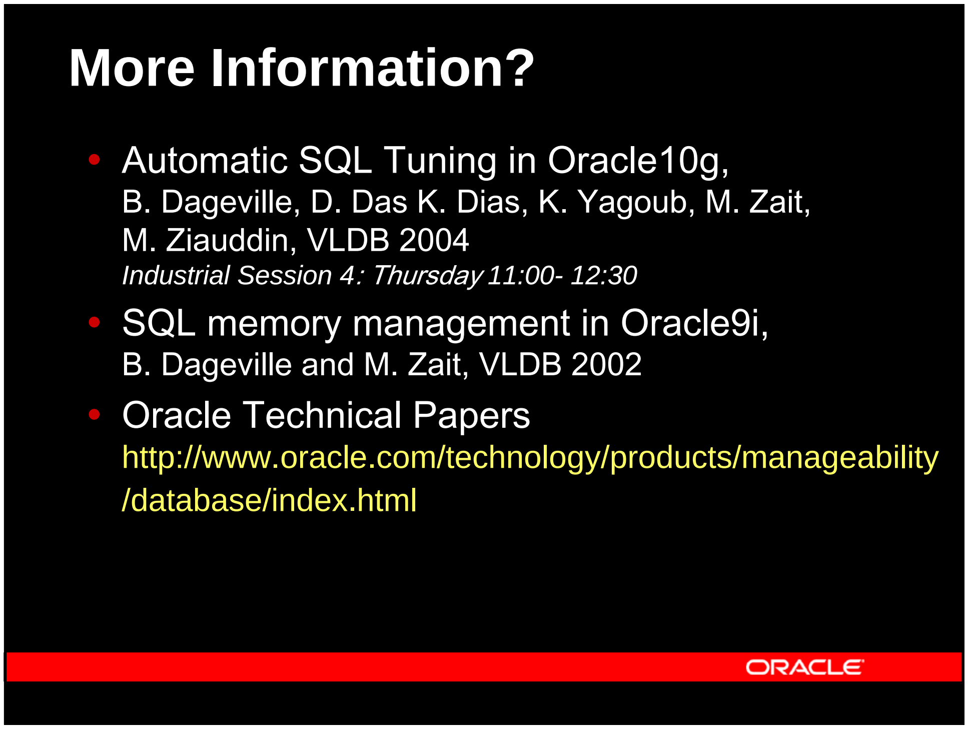
ORACLE



Conclusion & Future Directions

- Oracle10g major milestone in the Oracle's manageability quest
  - Manageability foundation
  - Holistic Management Control (ADDM)
  - Self-manageable components
- Future
  - Oracle11g: find an EVE for ADDM?
  - Even more self-manageable by fully automating the resolve phase

ORACLE



More Information?

- Automatic SQL Tuning in Oracle10g,
  B. Dageville, D. Das K. Dias, K. Yagoub, M. Zait,
  M. Ziauddin, VLDB 2004
  Industrial Session 4: Thursday 11:00- 12:30
- SQL memory management in Oracle9i,
  B. Dageville and M. Zait, VLDB 2002
- Oracle Technical Papers
  http://www.oracle.com/technology/products/manageability/database/index.html

ORACLE

ORACLE®