The Oracle Database: Past, Present, and Future

Lex de Haan
Oracle ST, curriculum development

Aalborg, 27 November 2003
Oracle History

1978: Oracle version 1
- Ran on PDP-11 under RSX, in 128 KB memory
- Written in assembly language
- Separated Oracle code (OPI) and user code (UPI)

1979: Oracle version 2
- Written in PDP-11 assembly language
- Ran on VAX/VMS in compatibility mode
Oracle History

1980: Oracle version 3
- Written in C: *portable source code*
- Retained split architecture
- Introduced the concept of *atomic SQL execution* and *transactions* (commit, rollback)

1984: Oracle version 4
- Introduced *read consistency*
- Ported to many platforms
- Interoperability between PC and server
Oracle History

1986: Oracle version 5
- True client-server (*distributed processing*)
- VAX-cluster support
- Version 5.1: *Distributed queries*

1989: Oracle version 6 (major kernel rewrite)
- OLTP performance enhancements, *savepoints*
- Online backup and recovery
- *Row-level locking*, PL/SQL in the database
- Parallel Server (VAX clusters, nCube)
Oracle History

1993: Oracle7
- Declarative *referential integrity*
- Stored *procedures* and *triggers*
- Shared SQL, parallel execution
- Advanced replication

1997: Oracle8
- Object-relational extensions in the database
- From client/server to *three-tier architecture*
- Partitioning option
Oracle History

1999: Oracle8i
- Java in the database (JVM and SQLJ)
- Partitioning enhancements
- Data warehousing enhancements
- XML support
- Summary management
- Oracle Internet Directory (LDAP)
- Ported to Linux
Oracle History

2001: Oracle9i

- *Real Application Clusters*, with cache fusion
  - Scalability on inexpensive clustered hardware
- Automatic segment-space management
- Internet security enhancements
- Integrated business intelligence functionality
- Data Guard (*standby databases*)
- Oracle managed files
- Globalization support (Unicode, time zones, locales)
Oracle Database 10g

Primary goal: Build a self-managing database that requires minimal human intervention.

- Reduction in administration cost without compromising high availability, scalability, and security.
- Minimal performance impact
- Effective for all configurations and workloads
Manageability: Key Concepts

1. Bring the management of all database components into the database itself
   - SQL tuning, resource management, space and storage management, backup and recovery

2. Integrate the management of these components with a central management engine
   - Enables holistic rather than independent or conflicting decisions

3. Provide an intelligent infrastructure to enable these components to be self-managing
Automatic Workload Repository (AWR)

The most critical infrastructure component; the data warehouse of the database.

- Storage of all the important system statistics and workload information.
- The database kernel is instrumented to capture an essential set of time-based statistics (Time Model)
  - Captured and computed in memory (ASH)
  - Periodically flushed to disk as snapshots & baselines

Using AWR’s information, the database is able to intelligently tune, accurately advise and proactively alert according to the workload and environment that are unique for every system.
Automatic Database Diagnostic Monitor (ADDM)

• The central management engine that integrates all AWR components
• A database performance expert “in the box”
• ADDM uses the data captured in the AWR
• ADDM uses a time-based classification tree to pinpoint root causes of performance problems
• The Time Model enables ADDM to make performance-tuning recommendations between seemingly incomparable trade-offs

A self-diagnostic engine like ADDM built inside the database is essential to good system performance
Automatic Workload Repository

- SGA
  - Efficient in-memory statistics collection
  - V$ DBA_\% MMON
  - AWR snapshots

- Internal clients: ADDM (Self-tuning component)

- External clients: EM SQL*Plus...

Tasks
Alerts
Adv
AWR
**DBMS_SCHEDULER Package**

- **Job class**
  - Job metadata
- **Program**
- **Job**
  - Arguments
- **Schedule**
  - When?
  - How many times?
- **Window**
  - Change prioritization
- **Prioritization**

- **Tasks**
  - Alerts
  - Adv
  - AWR
ADDM: Reactive versus Proactive Performance Monitoring

In-memory statistics

MMON

Snapshots

Reactive monitoring

Proactive monitoring

ADDM

Alerts

DBA

Reactive monitoring

ADDM results

AWR
Other Important Enhancement Areas

• Automatic Storage Management (ASM)
• Automatic shared memory management
• Human error correction
• High availability
  – Backup and recovery optimization
  – Data Guard (logical and physical standby databases)
  – Online redefinition
  – LogMiner
• Performance (wait interface, end-to-end tracing)
• BI and data warehousing
Automatic Storage Management (ASM)

ASM is a *database service* that provides:

- Load balancing in parallel across disk drives
- Prevention of disk space fragmentation
- Online disk space reorganization
- Data redundancy to provide fault tolerance
  - ASM can be built on top of vendor-supplied storage mechanisms
Human Error Correction

- **Flashback Query**
  - Query data at some point-in-time in the past

- **Flashback Versions Query**
  - View changes made over time at the row level

- **Flashback Transaction Query**
  - View changes made at the transaction level

- **Flashback Database**
  - New strategy for doing point-in-time recovery

- **Flashback Table**
  - Recover tables to a specified point in time

- **Flashback Drop**
  - Undrop a table (recycle bin)
The Future?

Databases are vital, and getting more so every day.
• Life sciences, content management, …
• Real-time business intelligence
• Security and availability
• Radio Frequency Identification (RFID)
• Databases versus operating systems
• Taking advantage of new types of hardware

The big challenge for companies like Oracle: Taking academic research results and applying it in practical terms.
ANSI/ISO SQL Language

- SQL-86 and SQL-87
- SQL-89 added referential integrity
- SQL-92
  - One standard with multiple conformance levels
- CLI-95 and PSM-96
- SQL:1999
  - Introduced multiple parts and named features
  - Some parts were backported to SQL-92
- OLB:2000 and JRT:2002
- SQL:2003 bug fixing release
ANSI/ISO SQL Conformance

- **SQL-86/87/89**
  - Levels 1 and 2
- **SQL-92**
  - Entry/Transitional/Intermediate/Full levels
- **SQL:1999 and SQL:2003**
  - Core SQL, plus named features and packages
- Conformance testing
  - FIPS 127: SQL-92 only
  - NIST (formerly NBS) did conformance testing until 1996; political changes eliminated NIST testing
SQL:1999 and SQL:2003 Parts

1. Framework
2. Foundation
3. Call Level Interface (CLI)
4. Persistent Stored Modules (PSM)
9. Management of External Data (MED)
10. Object Language Bindings (OLB)
11. Information and Definition Schemata
13. Java Routines and Types (JRT)
14. XML Related Specifications (XML)
The Future of ANSI/ISO SQL

• The current expectation is that the SQL language will go into “maintenance mode” after SQL:2003
• Main area of further development: SQL/XML
Questions and Answers