Real Application Testing

Fred Louis
Oracle Enterprise Architect
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.
Operations and Infrastructure Top Concerns

Figure 1. Increasing Rate of Change and Pressure to Move Faster

Managing change effectively is a strategic imperative!

Source: Gartner (December 2006)
Lifecycle of Change Management

- Make Change
- Set Up Test Environments
- Identify Patches & Workarounds
- Test
- Diagnose & Resolve Problems
- Provision for Production
- Diagnose & Manage Problems

Managing change with confidence
Lifecycle of Change Management

- Make Change
- Set Up Test Environments
- Identify Patches & Workarounds
- Test
- Diagnose & Resolve Problems
- Provision for Production
- Diagnose & Manage Problems

Real Application Testing
Real Application Testing

- **Value**
  - Rapid technology adoption
  - Higher testing quality

- **Business Benefit**
  - Lower cost
  - Lower risk

- **Features**
  - Database Replay
  - SQL Performance Analyzer (SPA)

Business Agility through Superior Testing
Database Replay
Testing Today
Production – 1,000s of Real Online Users

PRODUCTION
Testing Today
Test – 1-2 testers trying to be 1,000s of users

PRODUCTION

TEST
Database Replay

Workload for 1,000s of online users captured
Database Replay

Workload for 1,000s of online users replayed
Database Replay

Test your system changes at production levels

PRODUCTION

Capture

Workload

Replay

TEST
Real Application Testing with Database Replay

- Replay production workload in test environment
- Identify, analyze and fix potential instabilities before making changes to production

- Capture Workload in Production
  - Capture full production workload with real load, timing & concurrency characteristics
  - Move the captured workload to test system

- Replay Workload in Test
  - Make the desired changes in test system
  - Replay workload with full production characteristics
  - Honor commit ordering

- Analyze & Report
  - Errors
  - Data divergence
  - Performance divergence
Supported Changes

Changes Supported
• Database Upgrades, Patches
  • Schema, Parameters
  • RAC nodes, Interconnect
• OS Platforms, OS Upgrades
  • CPU, Memory
  • Storage
  • Etc.

Recording of External Client Requests
Database Replay Workflow

Production (9.2.0.8)
- Clients
- Mid-Tier
- Storage

Capture

Process

Replay

Analysis & Reporting

Test (11.1)
- Replay Driver
- Storage
Database Replay
Step 1: Workload Capture

- All external client requests captured in binary files
- System background and internal activity excluded
- Minimal overhead
  - Avoids function call when possible
  - Buffered I/O
- Independent of client protocol
- Can capture on 9.2.0.8 and replay on 11g
- Capture load for interesting time period, e.g., peak workload, month-end processing, etc.
Capture Options

• Workload can be filtered to customize what is captured
  • Inclusion Filters: Specifies which sessions should be captured
  • Exclusion Filters: Specifies which sessions should NOT be captured
  • Filter Attributes: Any of the following session attributes can be used for filtering
    • User
    • Program
    • Module
    • Action
    • Service
    • Session ID

• Workload capture can be run on-demand or scheduled to run at later time
Step 2: Process Workload Files

- Setup test system
  - Application data should be same as production system as of capture start time
  - Use RMAN, Snapshot Standby, imp/exp, Data Pump, etc. to create test system
  - Make change: upgrade db and/or OS, change storage, migrate platforms, etc.
- Processing transforms captured data into replayable format
- Once processed, workload can be replayed many times
- For RAC copy all capture files to single location for processing
Step 3: Replay Workload

- Replays workload preserving timing, concurrency and dependencies of the capture system.

- Replay Client is a special program that consumes processed workload and sends requests to the replay system.

- Clients interpret captured calls into sequence of OCI calls and submit to database.

- For high concurrency workloads, it may be necessary to start multiple clients.

OCIStmtPrepare()  
OCIBindByName()  
OCIAttrSet()  
OCIStmtExecute()
Replay Options

• **Synchronized Replay (Default)**
  - Workload is replayed in full synchronized mode
  - Same concurrency and timing as production workload
  - Transaction commit order is honored
  - Ensures minimal data divergence

• **Synchronization controls**
  - Workload can be replayed in unsynchronized mode
  - Useful for load/stress testing
  - High data divergence
  - Parameters for controlling synchronization
    - Commit order synchronization: `SYNCHRONIZATION`
    - Think time synchronization: `THINK_TIME_SCALE`
    - Connect (logon) time synchronization: `CONNECT_TIME_SCALE`
    - Request rate preservation: `THINK_TIME_AUTO_CORRECT`
Analysis & Reporting

- **Error Divergence**: For each call error divergence is reported
  - New: Error encountered during replay not seen during capture
  - Not Found: Error encountered during capture not seen during replay
  - Mutated: Different error produced in replay than during capture

- **Data Divergence**
  - *Replay*: Number of rows returned by each call are compared and divergences reported
  - *User*: Application level validation scripts

- **Performance Reporting**
  - Capture and Replay Report: Provides high-level performance information
  - ADDM Report: Provides in-depth performance analysis
  - AWR, ASH Report: Facilitates comparative or skew analysis
Current Restrictions

- Database Replay does not support the following features in the current release
  - SQL Loader direct path load, import/export
  - OCI based object navigation (ADTs) and REF binds
  - Streams, non-PL/SQL based AQ
  - Distributed transactions, remote describe/commit operations
  - Flashback queries
  - Shared server
Comparison of LoadRunner & DB Replay

Testing e-Business Suite

<table>
<thead>
<tr>
<th></th>
<th>LoadRunner</th>
<th>DB Replay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install &amp; Setup</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Understand Application Usage</td>
<td>24</td>
<td>0</td>
</tr>
<tr>
<td>Identify Key Transactions</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Generate Workload</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Run Test</td>
<td>80</td>
<td>5</td>
</tr>
</tbody>
</table>

Total Testing Time

DB Replay: ½ month

LoadRunner: 7 ½ months
Database Replay Comparison

Technology that can only be built by Oracle

<table>
<thead>
<tr>
<th></th>
<th>3rd Party Load Testing Tools</th>
<th>Oracle Database Replay</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Workload</strong></td>
<td>Artificial simulated workload</td>
<td>Real application workload</td>
</tr>
<tr>
<td><strong>Test Coverage</strong></td>
<td>1-5% of workflow</td>
<td>100% of workflow</td>
</tr>
<tr>
<td><strong>Test Effort</strong></td>
<td>• Simple app: weeks&lt;br&gt;• Complex app: months</td>
<td>• Simple app: days&lt;br&gt;• Complex app: days</td>
</tr>
</tbody>
</table>
SQL Performance Analyzer (SPA)
Why SQL Performance Analyzer (SPA)?

- Businesses want systems that are performant and meet SLA’s
- SQL performance regressions are #1 cause of poor system performance
- Solution for proactively detecting *all* SQL regressions resulting from changes not available
- DBA’s use ineffective and time-consuming manual scripts to identify problems

SPA identifies all changes in SQL performance before impacting users
SQL Performance Analyzer

- Test impact of change on SQL query performance
- Capture SQL workload in production including statistics & bind variables
- Re-execute SQL queries in test environment
- Analyze performance changes – improvements and regressions
SQL Performance Analyzer Workflow

Production (9.2) → Clients

Mid-Tier

Capture SQL

Transport SQL

Execute SQL Pre-change

Execute SQL Post-change

Compare Perf

Test (10.2 or 11.1)
Step 1: Capture SQL Workload

- SQL Tuning Set (STS) used to store SQL workload
  - For Oracle9i release, SQL*Trace can be used
- STS includes:
  - SQL Text
  - Bind variables
  - Execution plans
  - Execution statistics
- Incremental capture used to populate STS from cursor cache over a time period
Step 2: Move SQL Workload to Test System

- For Oracle Database 10g or higher, export STS to test system
- Transport staging table to test system (datapump, db link, etc.)
- Copy SQL tuning set from staging table (“unpack”)
Step 3: Execute SQL Before Making Change

- Establishes SQL workload performance baseline
- SQL execution plan and statistics captured
- SQL executed serially (no concurrency)
- Each SQL executed only once
- DDL/DML skipped
- Option to do Explain Plan only analysis
Step 4: Execute SQL After Making Change

- **Manually implement the planned change**
  - Database upgrade, patches
  - Optimizer statistics refresh
  - Schema changes
  - Database parameter changes
  - Tuning actions, e.g., SQL Profile creation

- **Re-execute SQL after change**
  - Gathers new SQL execution plans and statistics
Step 5: Compare & Analyze Performance

- Compare performance using different metrics, e.g.,
  - Elapsed Time
  - CPU Time
  - Optimizer Cost
  - Buffer Gets

- SPA Report shows impact of change for each SQL
  - Improved SQL
  - Regressed SQL
  - Unchanged SQL

- Fix regressed SQL using SQL Tuning Advisor or SQL Plan Baselines
DEMONSTRATION

SQL Performance Analyzer
SPA Report

SQL Performance Analyzer Task Result: SYS.SYSTEMCHANGES1

- Task Name: SYSTEMCHANGES1
- Task Owner: SYS
- SQL Tuning Set Name: HR_WORKLOAD
- STS Owner: APPS
- Replay Trial 1: Before
- Replay Trial 2: After
- Total SQL Statements: 50
- SQL Statements With Errors: 0

Global Statistics

Projected Workload Execute Elapsed Time

- Before
- After

Replay Trial

Improvement Impact: 85% ↑
Regression Impact: -5% ↓
Overall Impact: 85% ↑

SQL Statement Count

- Improved
- Regressed
- Unchanged

Change in Execute Elapsed Time

Recommendations:

Run SQL Tuning Advisor to tune regressed SQL statements.
Schedule SQL Tuning Advisor.
To:

- Automated workload capture
- Production workload
- Automated analysis in minutes
- Complete workload
- Low risk

From:

- Manual workload creation
- Synthetic workload
- Months of manual analysis
- Partial workload
- High risk

SQL Performance Analyzer Benefits
Real Application Testing for pre-11g Databases

• More details on how to use Database Replay and SQL Performance Analyzer in earlier releases can be found on OTN

• Real Application Testing for Earlier Releases

• Testing Performance Impact of an Oracle 9i to Oracle Database 10g Release 2 Upgrade with SQL Performance Analyzer
"Oracle Real Application Testing reduces the time required to test changes by as much as 80%, lower testing costs by as much as 70%, mitigate risks by reducing the number of unexpected outages, and improve the quality of service for their IT operations."

Source: Oracle Real Application Testing, business agility through superior testing, Jan 2008

David Mitchell
Senior Vice President, OVUM
Lifecycle of Change Management

Make Change

Set Up Test Environments

Diagnose & Resolve Problems

Provision for Production

Identify Patches & Workarounds

Diagnose & Manage Problems

Test

Provisioning Automation
# Provisioning Automation with Cloning

## Simple, Accurate, Reliable Capacity on Demand

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple, Accurate, Reliable rolling out of images</td>
<td>Gold image based cloning of software bits</td>
</tr>
<tr>
<td>Reduce possibility of errors and improve reliability</td>
<td>Image can be pre-patched to any level</td>
</tr>
<tr>
<td>Low TCO through automation</td>
<td>Image is sourced from the host itself or from the Software Library</td>
</tr>
<tr>
<td>Higher quality of service</td>
<td>Optionally performs post cloning configuration e.g., database creation</td>
</tr>
<tr>
<td></td>
<td>Several homes can be cloned in parallel</td>
</tr>
<tr>
<td></td>
<td>Database cloning uses fast, proven RMAN technology</td>
</tr>
</tbody>
</table>
Deployment Cycle

1. Create reference system
2. Stage gold images
3. Provision Production system
4. Scale out (RAC cluster)
5. Scale back

Testing and Staging

Production

ORACLE
Automated Patch Management

Simple, Accurate, Reliable Maintenance of Distributed Systems

• Benefits
  • Simple, Accurate, Reliable maintenance of Oracle systems
    • Reduce possibility of errors and improve reliability
  • Fewer errors and faster patching
  • Patch and maintain systems while providing higher quality of service

• Functionality/Capability/What’s new
  • Out-of-box best practice driven patching of single instance and RAC databases, AS midtier and custom (BPEL and SOA Apps)
    • Support for interim patches, patchsets and CPUs
    • Metadata driven patch prerequisites
  • Rolling support for RAC/ASM/Clusterware patching
  • Continued Live update of best practices from Metalink
  • Support for sudo and PAM based authentication
Summary
Use Real Application Testing to Manage Change with Confidence and …

- **LOWER** change risk
- **LOWER** testing time without compromising quality
- **LOWER** unplanned outages
- **IMPROVE** application reliability
- **IMPROVE** system performance
- **IMPROVE** end-user experience
- **IMPROVE** quality of service
- **ADOPT** new technology faster