

ORACLE®

**ORACLE**  
DATABASE **11<sup>g</sup>**




**ORACLE®**

## **Secrets to a Successful Upgrade: Database Replay to the Rescue**

Mughees A. Minhas  
Director of Product Management  
Database and Systems Management  
Oracle USA





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.

# Agenda

- Motivation for Database Replay
- Technology deep-dive
- Usage best practices
- Case study
- Conclusion
- Q & A

# Causes of Unsuccessful Upgrades

- Inadequate testing
  - ... despite extensive testing and validation
    - Many issues go undetected
    - System availability and performance negatively impacted
- Cause of low success rate
  - Current tools provide inadequate testing
    - Simulate synthetic workload instead of replaying actual production workload
    - Provide partial workflow coverage



**DB Replay provides realistic testing for real-world systems**

# What is Database Replay?

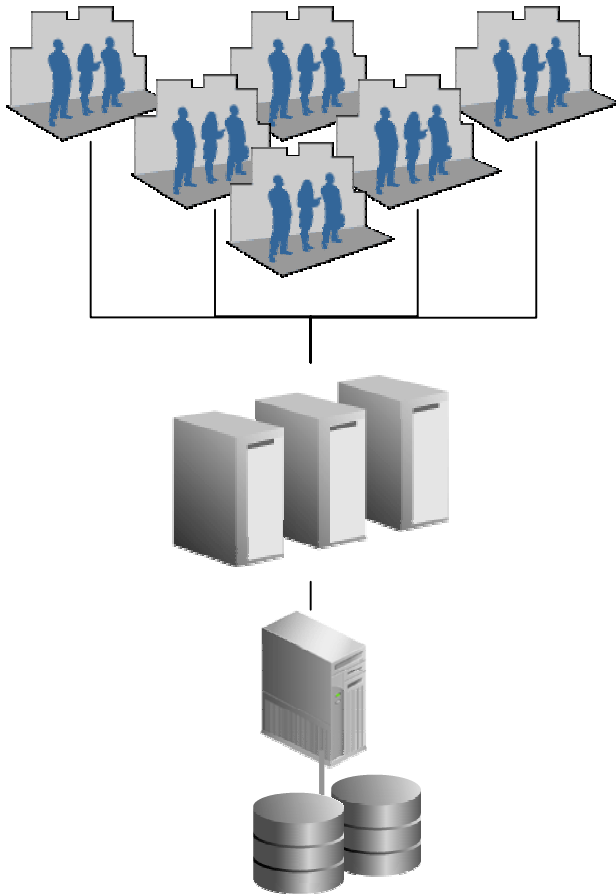
- New change assurance feature introduced in Oracle Database 11g
- Offered in conjunction with SQL Performance Analyzer (SPA)
- Goal is to enable successful adoption of new technology, specifically new releases of the database at low-cost and low-risk



# Database Replay

# Testing Today

**Production – 1,000s of Real Online Users**

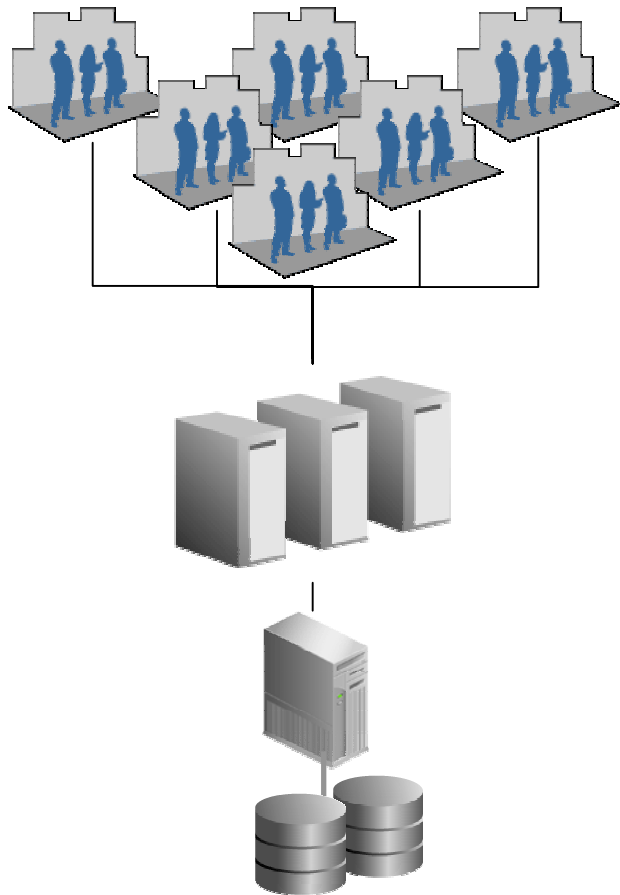


**PRODUCTION**

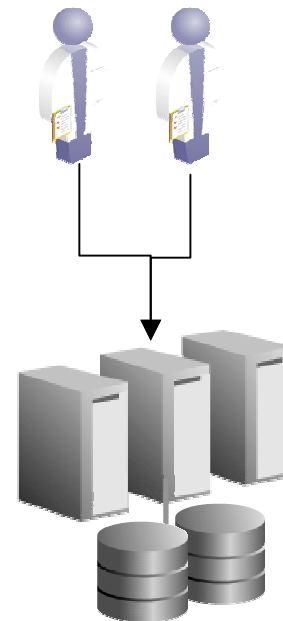


# Testing Today

**Test – 1-2 testers trying to be 1,000s of users**



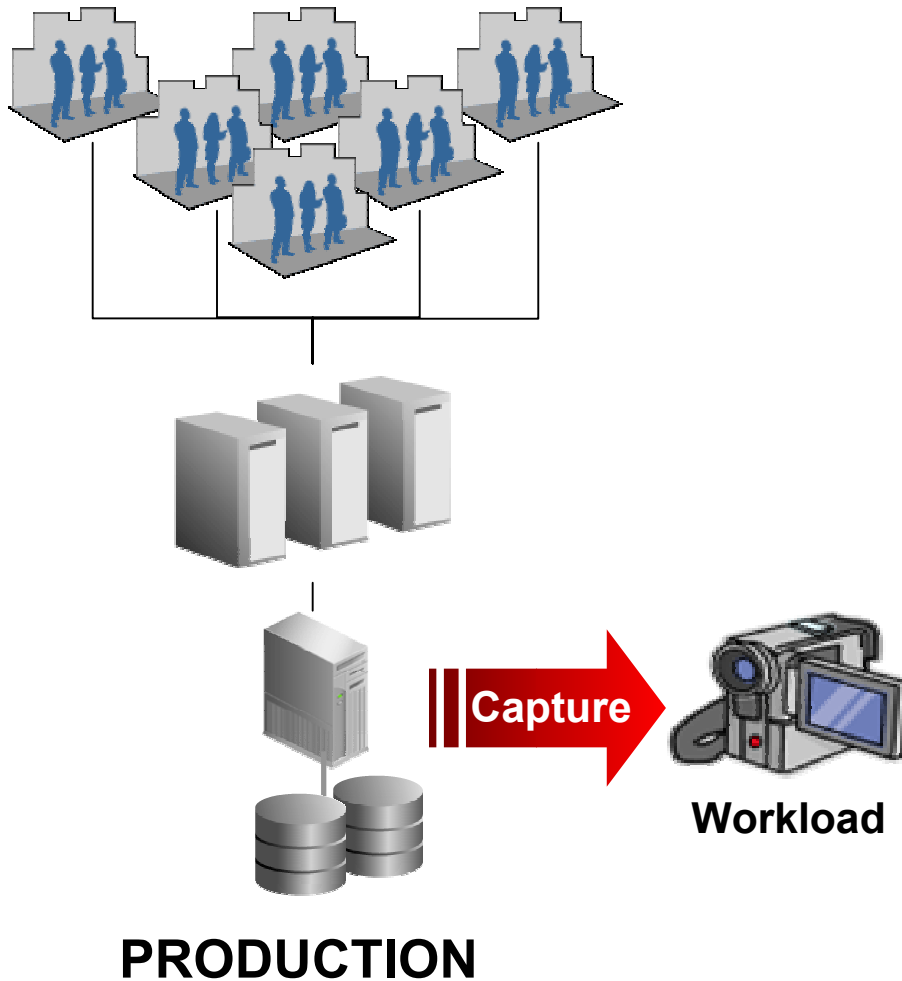
**PRODUCTION**



**TEST**

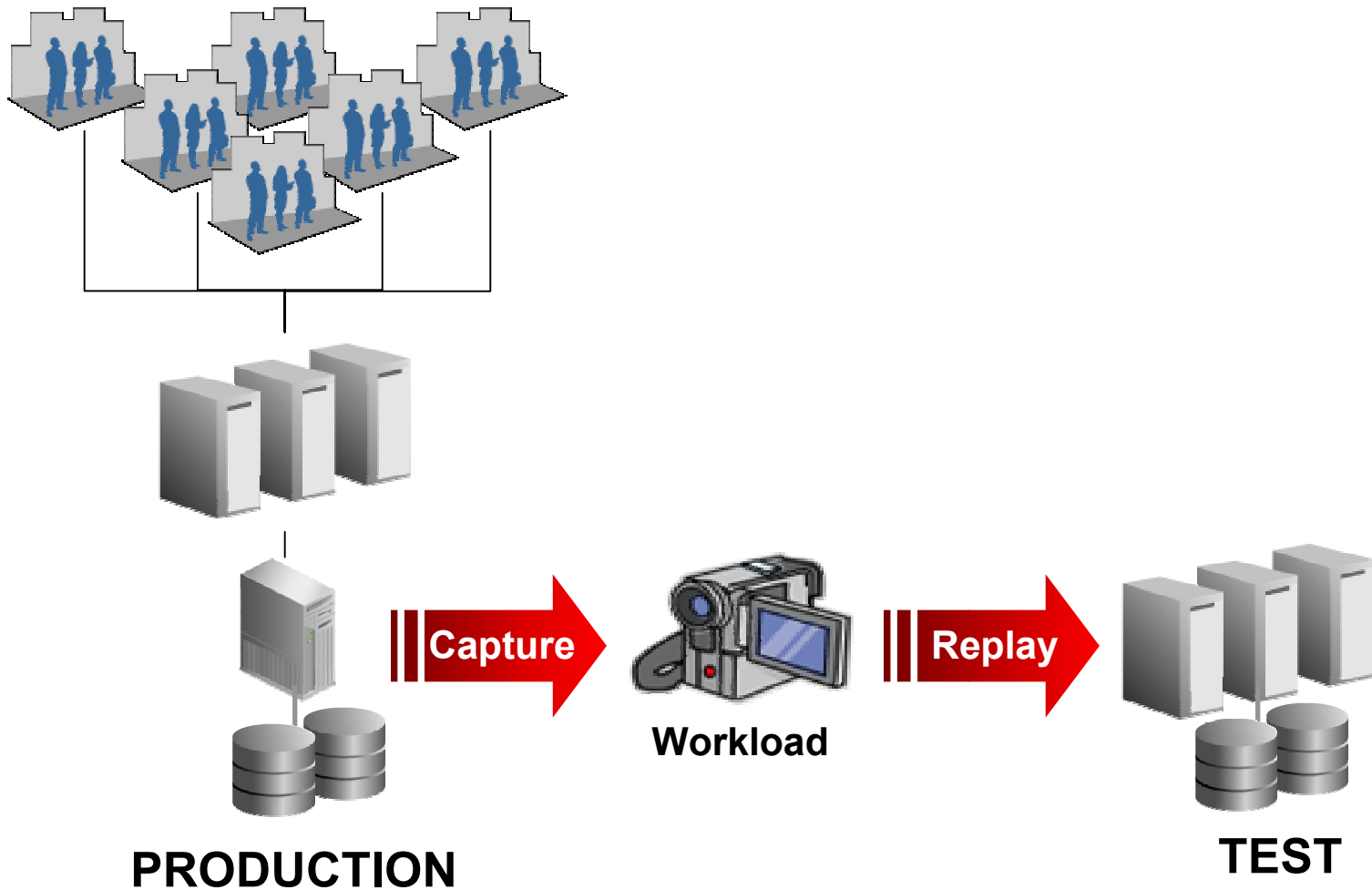
# Database Replay

Workload for 1,000s of online users captured



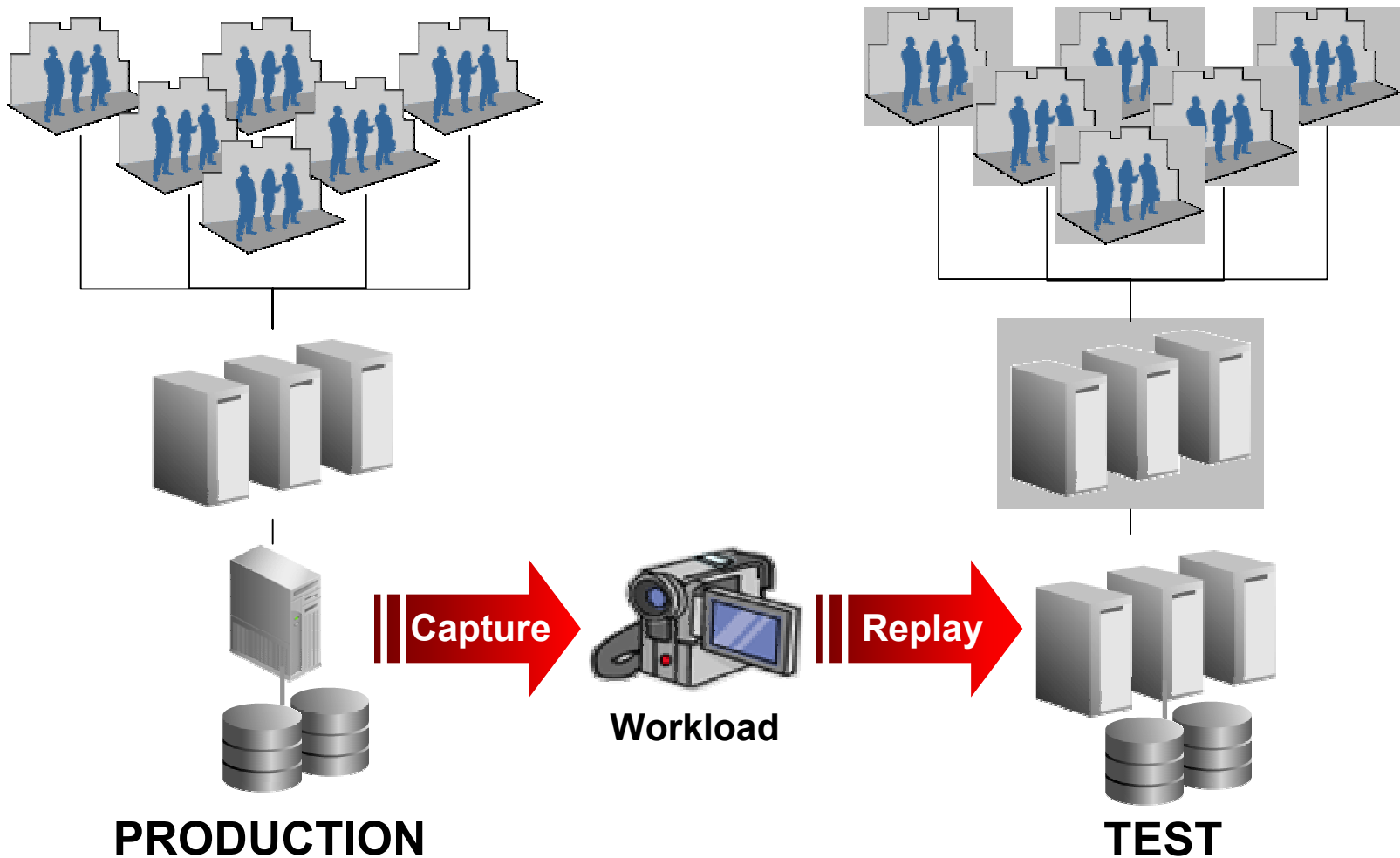
# Database Replay

Workload for 1,000s of online users replayed



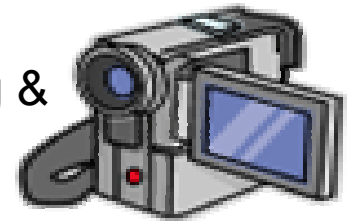
# Database Replay

Test your system changes at production levels



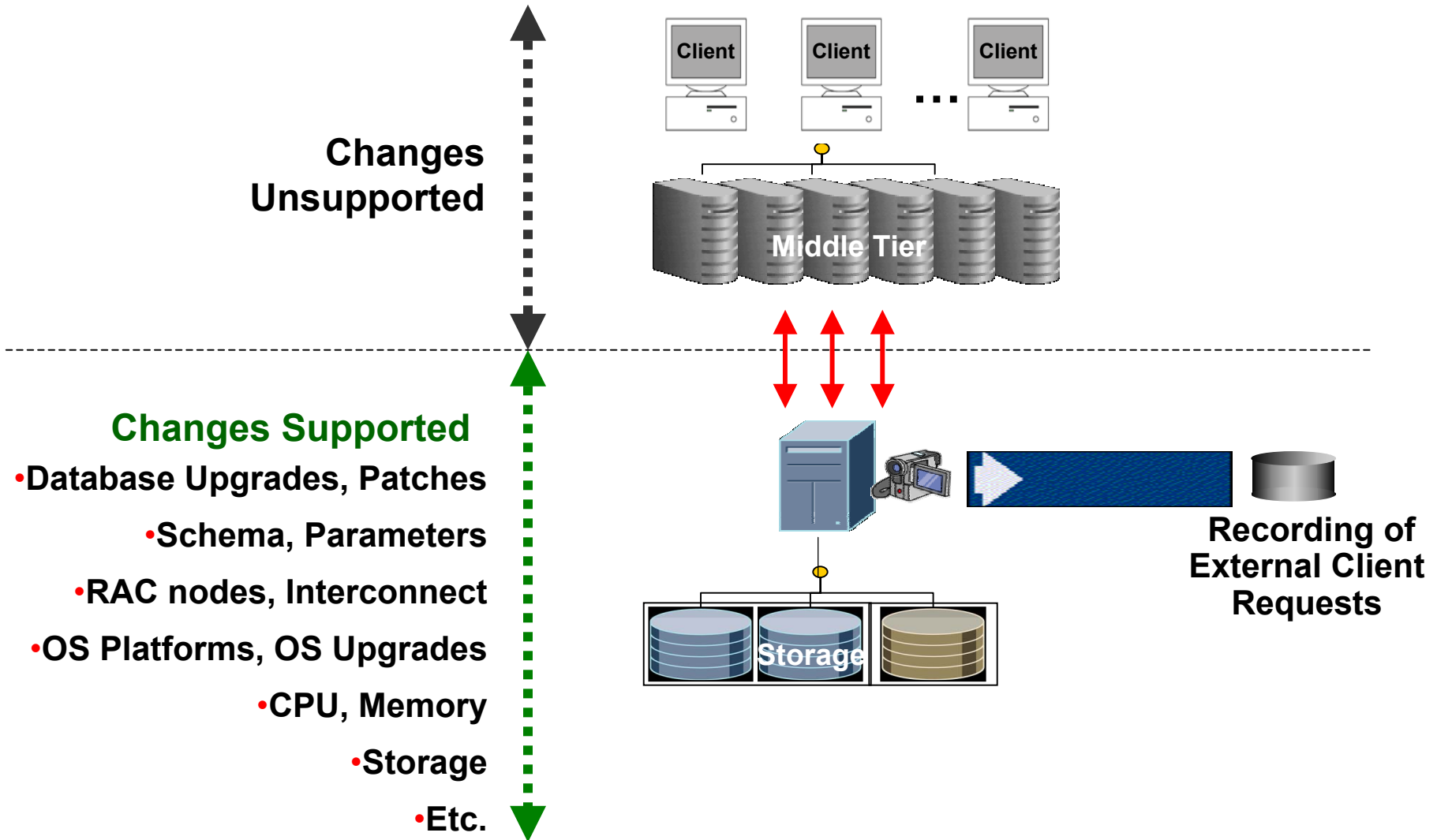
# Database Replay Overview

- Replay actual production database 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

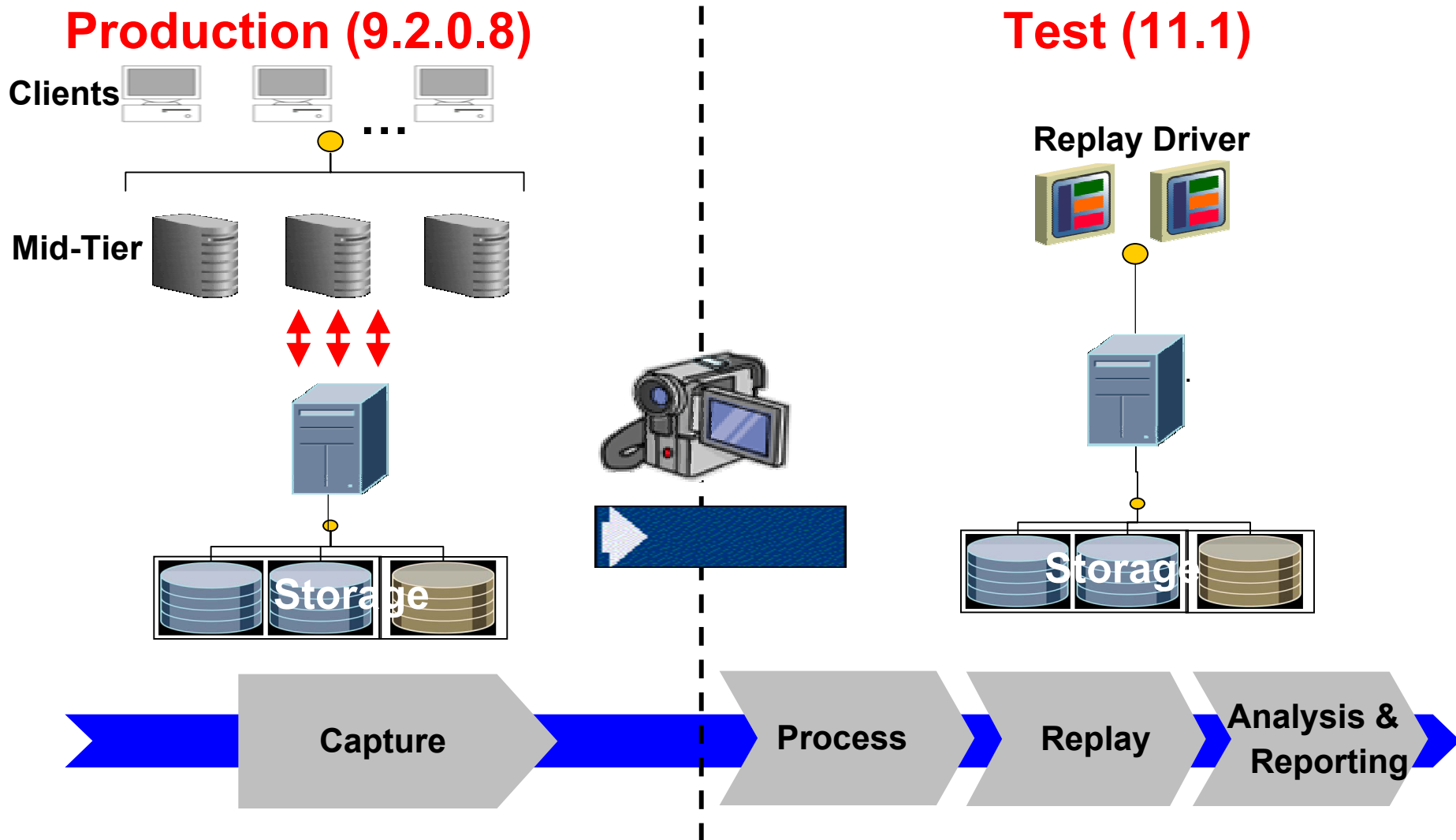


**Analysis & Reporting**

# Supported Changes

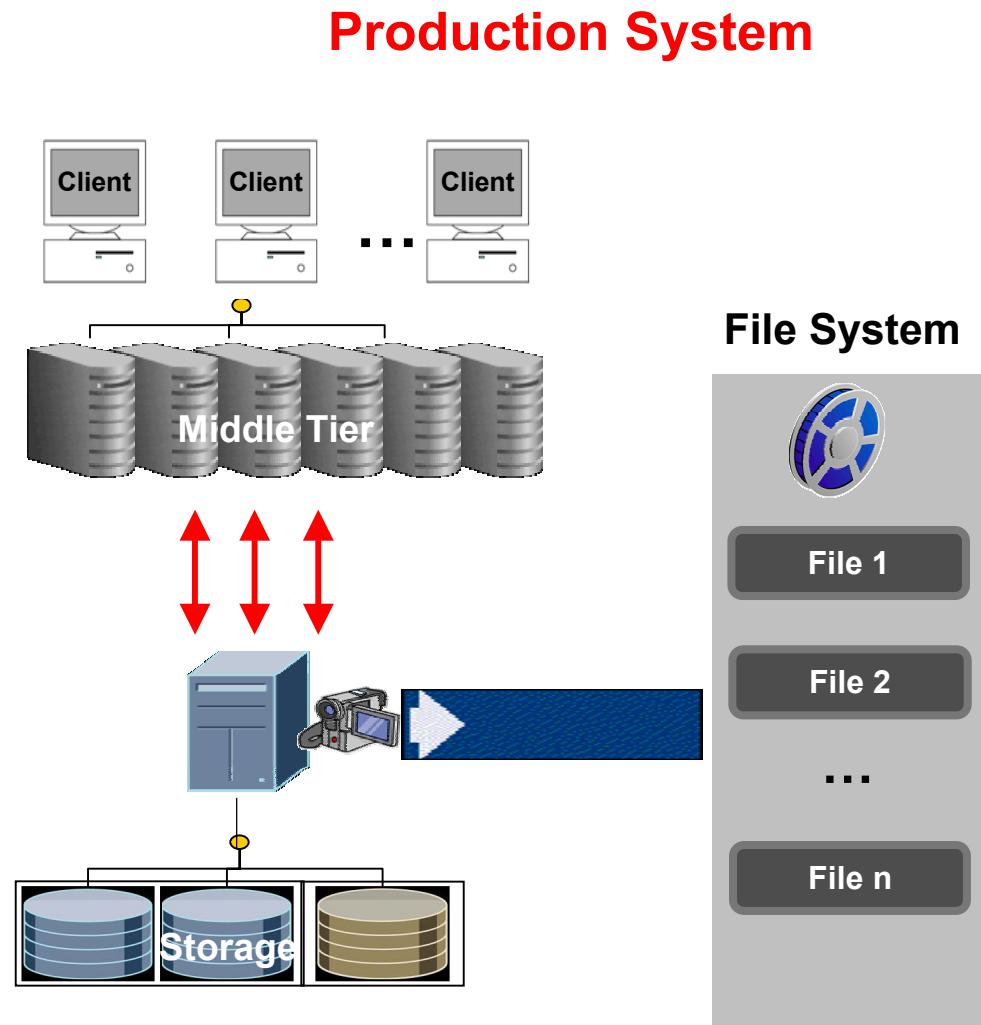


# Database Replay Workflow



# Step 1: Workload Capture

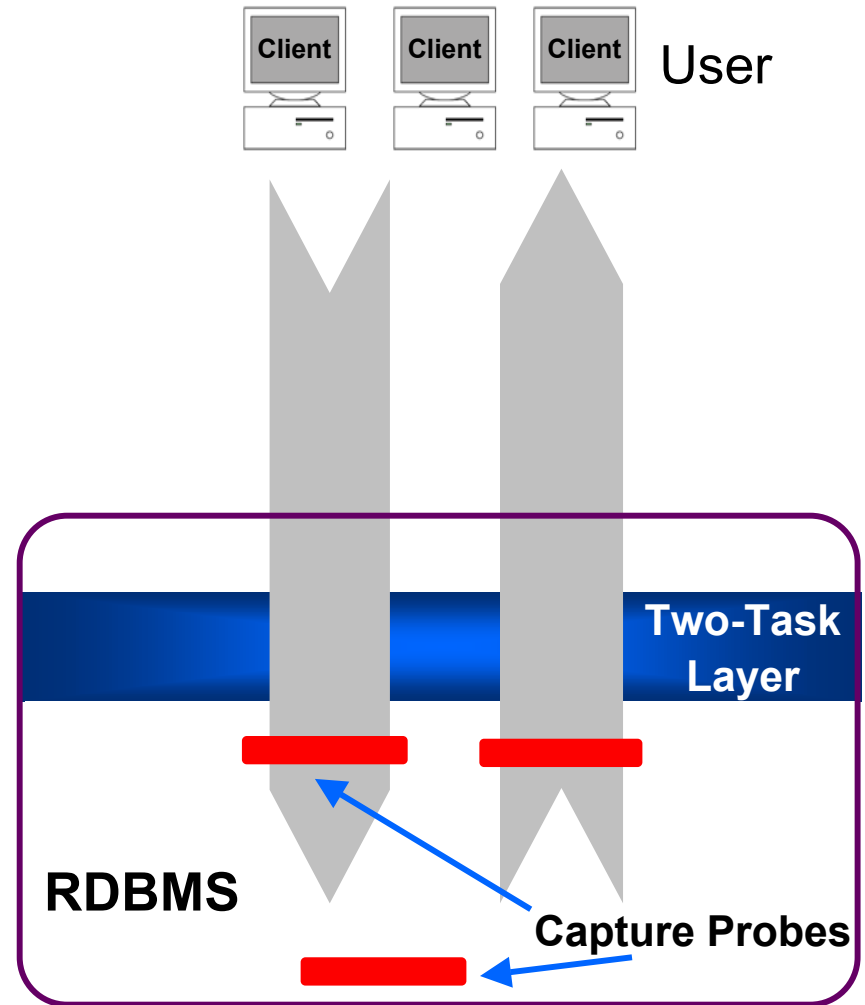
- All external client requests captured in binary files
- System background and internal activity excluded
- Minimal overhead
  - Avoids function call when possible
  - Buffered I/O
- Independent of client protocol
- Can capture on 10.2.0.4 and replay on 11g
- Capture load for interesting time period, e.g., peak workload, month-end processing, etc.





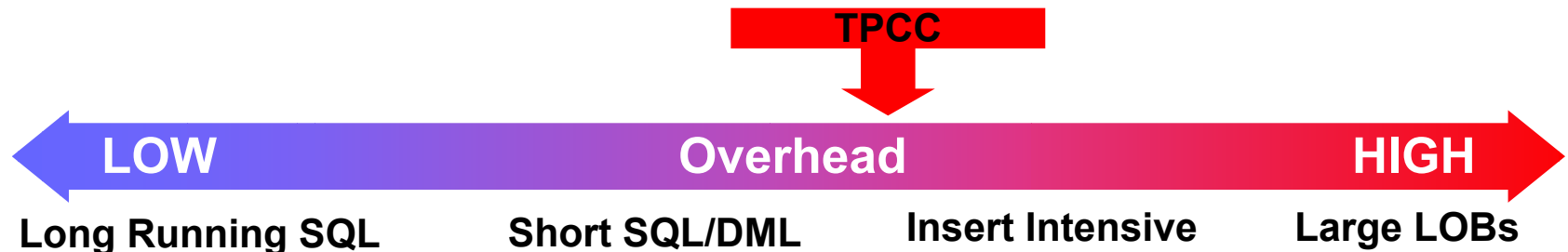
# Workload Capture Data Classes

- Client → RDBMS
  - SQL and PL/SQL Text
  - Binds
  - OCI Interface calls
- RDBMS → Client
  - ROWIDS, Lob locators in out-binds and select lists
  - Row counts for queries and DML
  - Error Codes
- System Data
  - Sequences
  - SCN
  - Timing information



# Capture Overhead

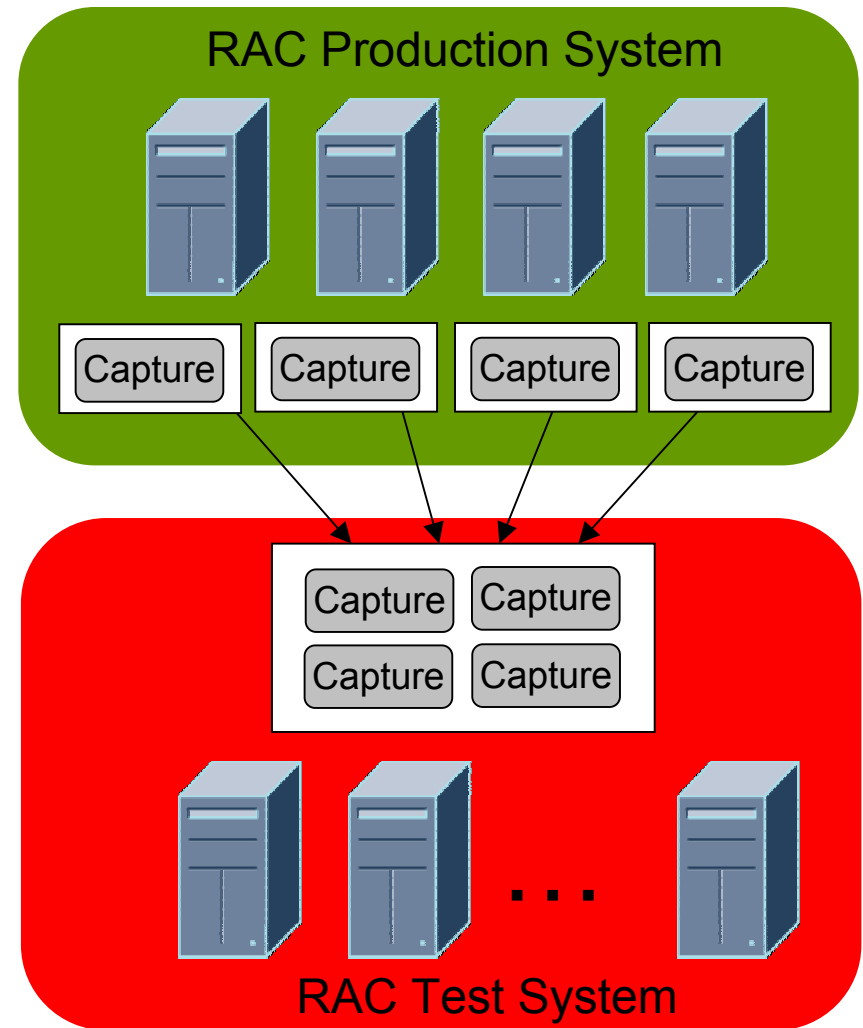
- Performance Overhead
  - Workload dependent
  - TPCC throughput degradation about 4.5%
  - Proportional to database call size



- Memory Overhead
  - Each captured process allocates 64KB in PGA for buffering captured data

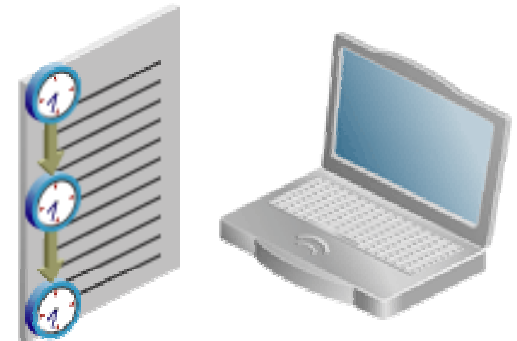
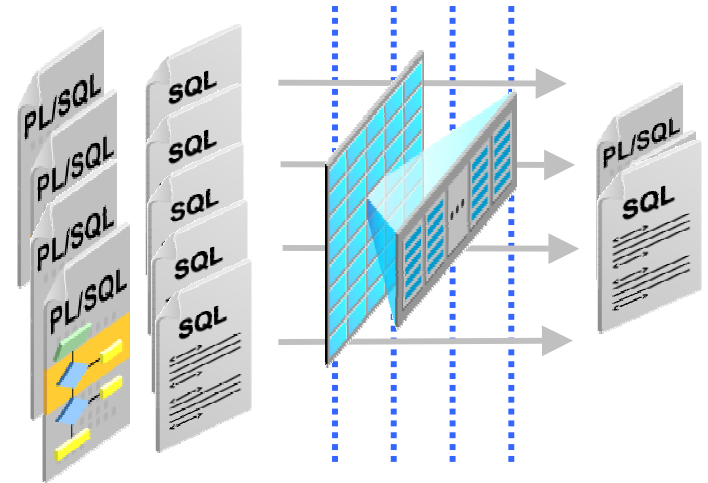
# Workload Capture on RAC

- Shared and local file system supported
- Production and test system can have different number of nodes
- Shared file system (recommended)
  - One directory shared across all nodes
  - Captures entire workload
- Local file system
  - Each node with separate capture directory
  - Directory name and path must be same on all nodes
  - For replay, workload files must be consolidated into single directory



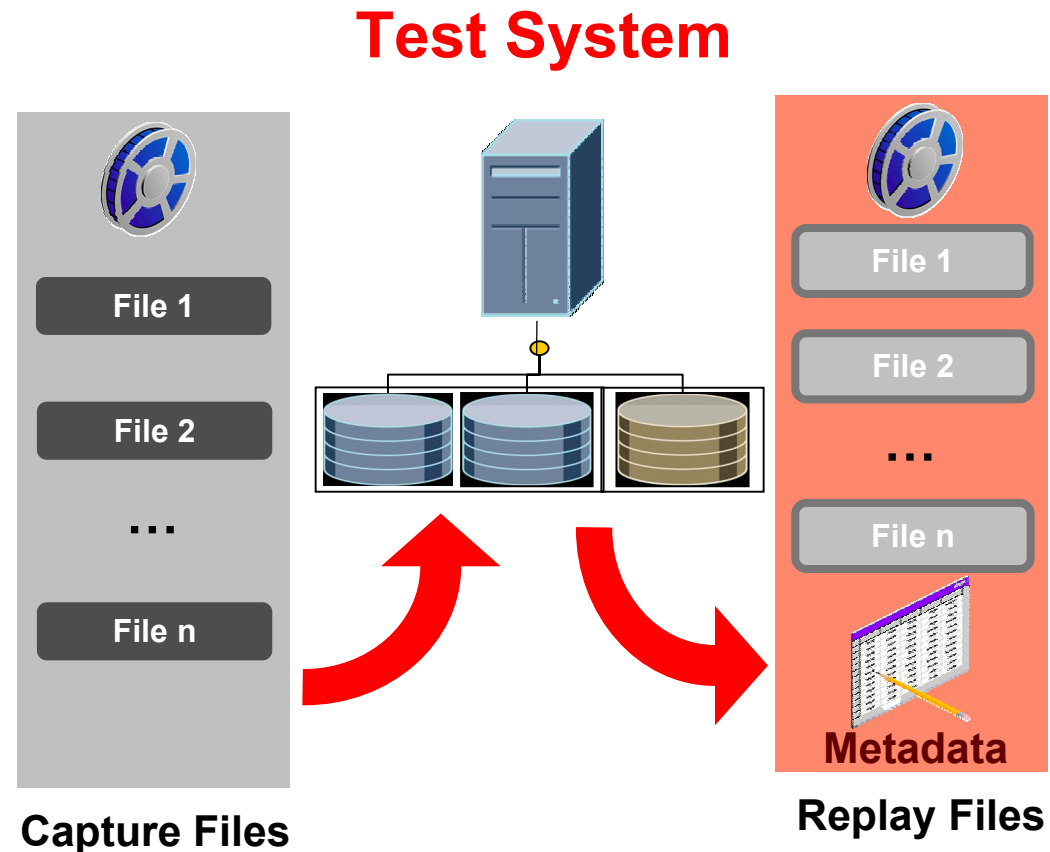
# Capture Options

- 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



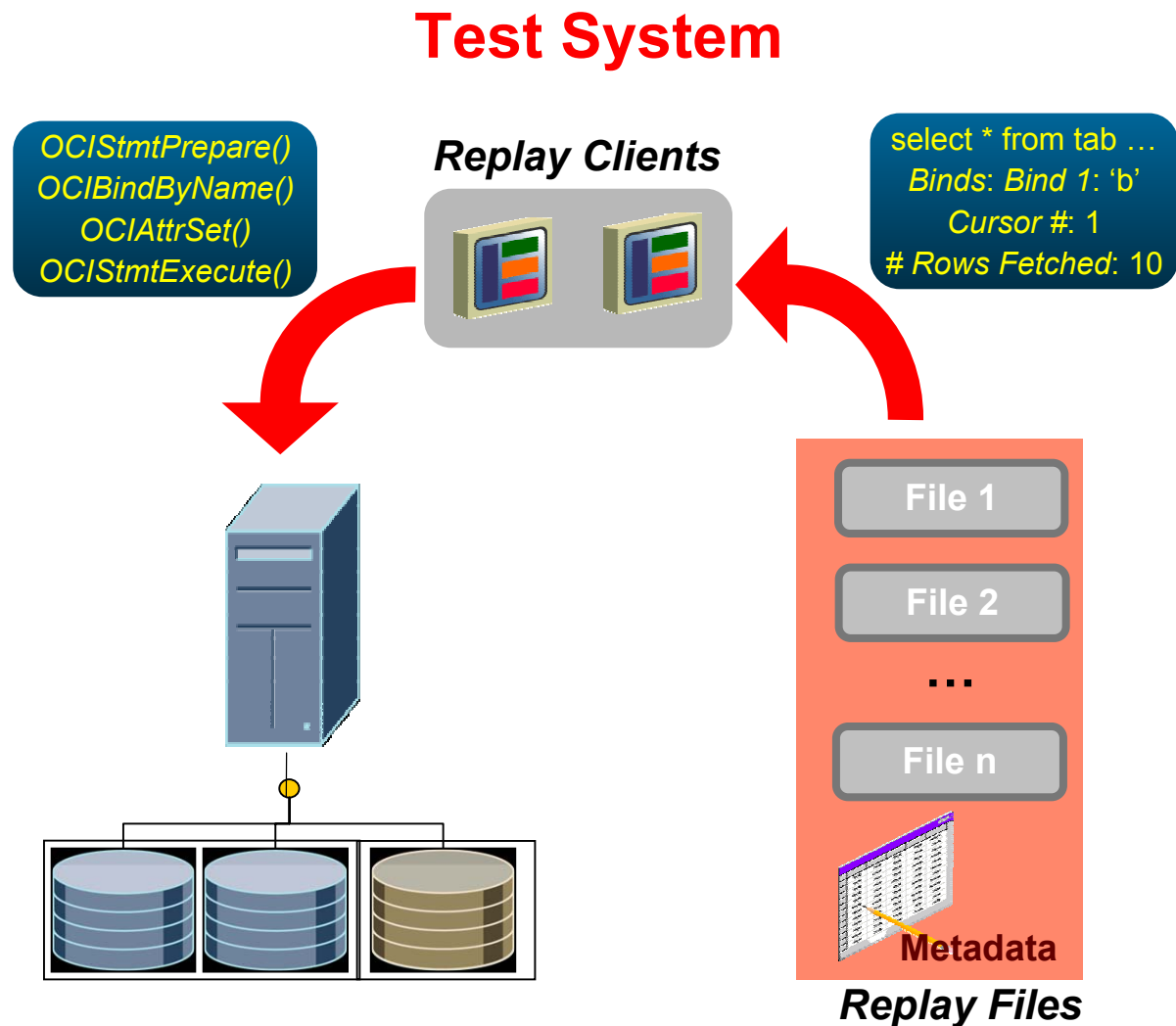
# Step 2: Process Workload Files

- 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

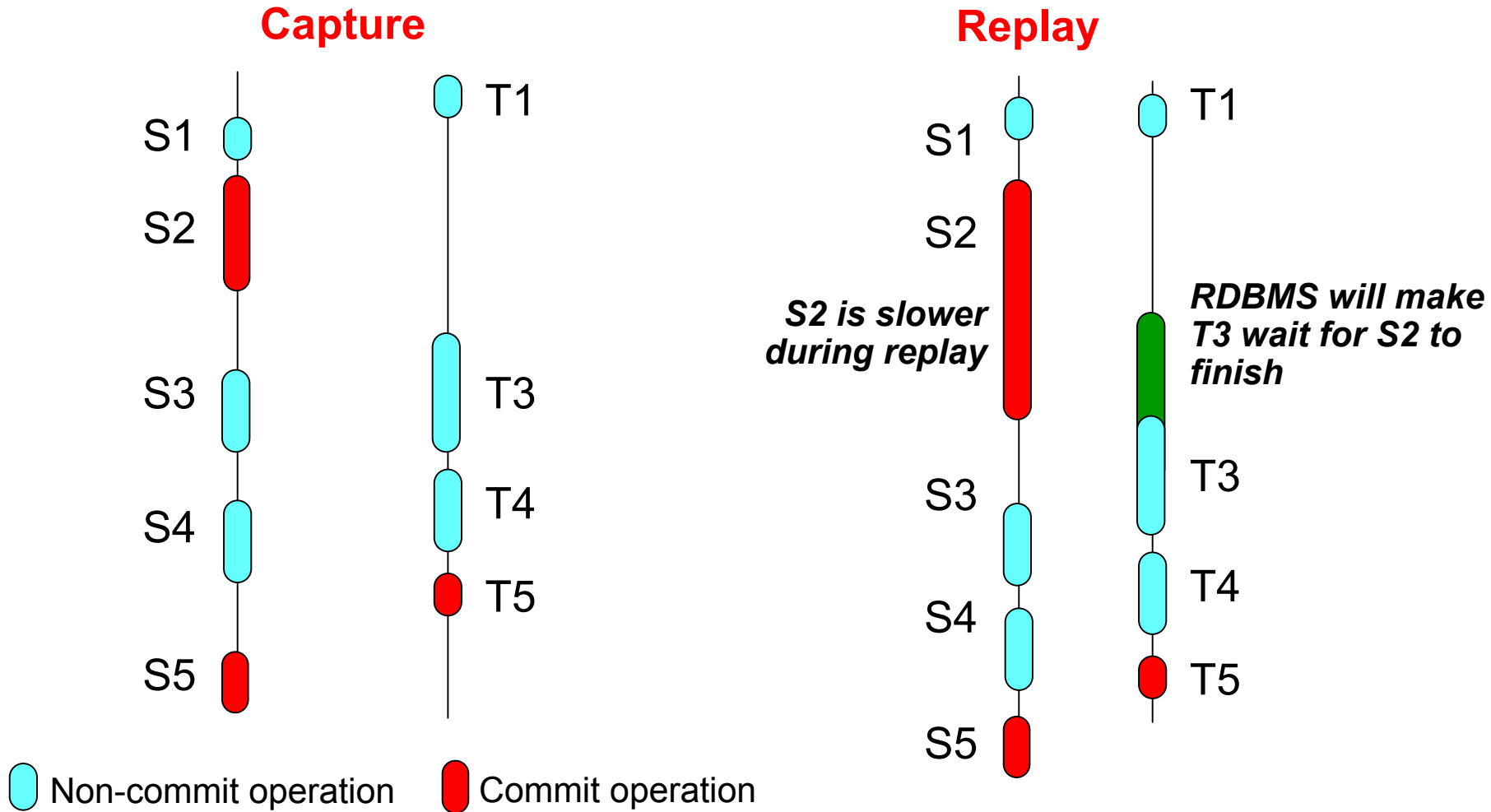


# Step 3: Replay Workload

- 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



# Workload Replay: Synchronization



# Workload Replay: Physical Locator Remapping

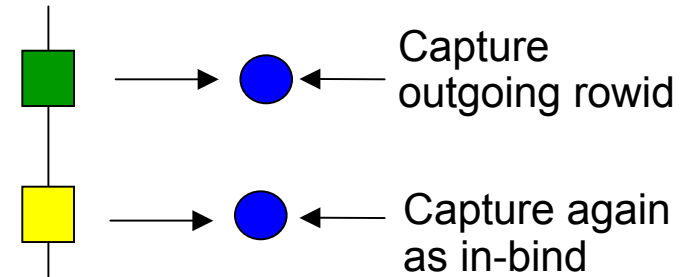
- **Scenario:**

■ Select rowid from emp where  
ename = 'Smith';

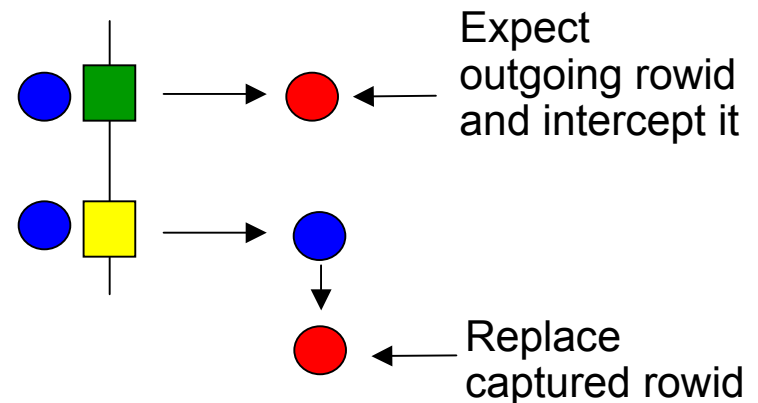
...  
■ Update emp set sal=2\*sal where  
rowid=:1 ■

- Captured bind value does not help with replay
- Update will fail unless remapped
- Rowids automatically remapped with valid runtime values
- Physical locators auto-remapped include, rowids, LOB locator, Ref cursors

## Capture



## Replay

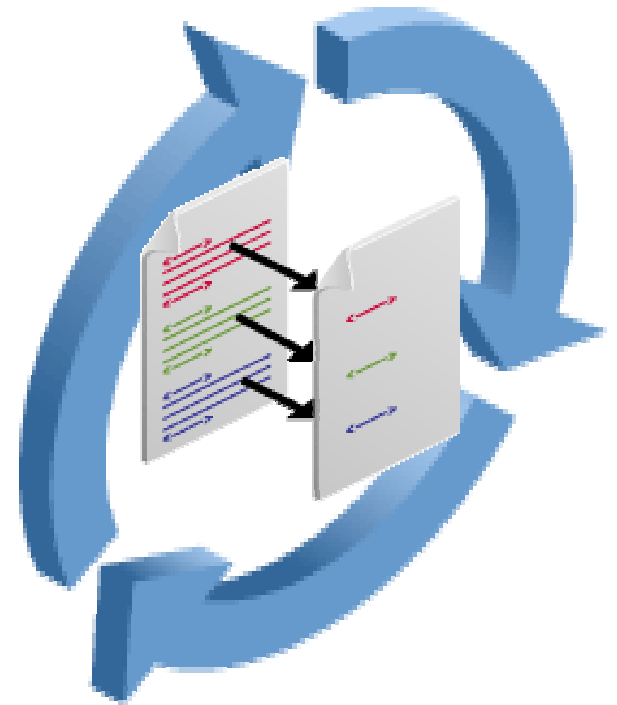




# Workload Replay:

## Causes of Data Divergence

- Certain user actions not synchronized during replay
  - Calls to `dbms_pipe`
  - Commits within PL/SQL
  - User locks
- External data dependencies maintained in application logic
- Use of non-repeatable functions
  - `RANDOM()`, `SYSDATE()`
- External interactions via urls, dblinks and BFILEs
- In-flight sessions at start of capture



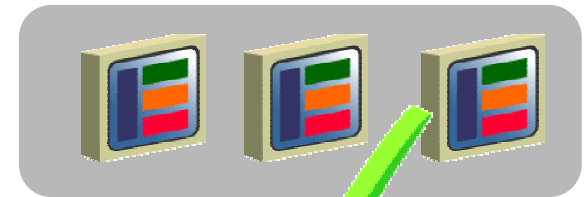
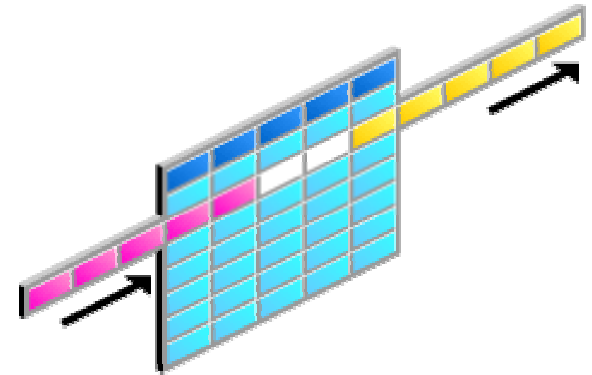
# Replay Options

- 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**



# Replay Options

- Connection Remapping
  - Capture-time connection string should be remapped for replay time
  - Examples
    - One-to-One: Allows simple instance-to-instance re-mapping
    - Many-to-One: Maps several connection strings to a service in the test system (e.g., load balancing listener)
- Number of Replay Clients
  - Configurable by user
  - Calibration mode recommends number of replay clients needed for specific workload
  - Replay clients are multithreaded clients that can drive multiple workload sessions



# 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



# Database Replay Best Practices



# Best Practices

- Pre-capture Planning
  - Storage Overhead
    - Provide adequate disk space for workload capture files
    - Space depends on size and type of workload
    - Extrapolate storage needed based on running capture for few minutes
    - TPC-C Benchmark: 1.2 GB for 100 users for approx. 20min
  - System Overhead
    - Ensure system has spare capacity (CPU, memory, I/O) for capture
- Choose the capture period
  - Interesting workload
  - Peak period

# Best Practices

- Workload Capture
  - Database restart (optional): Recommended to minimize divergence
  - File system for RAC: Use shared file system
  - AWR Data Export
    - Export AWR Data to enable in-depth replay performance analysis
    - Consider impact on production system before exporting
- Test System Setup
  - Application data in test system must be identical to production to minimize replay data divergence
  - Have strategy in place to duplicate production data on test



# Best Practices

- Workload Processing
  - Processing workload has performance overhead and can possibly take a long time
  - Process workload on test system instead of production
- Workload Replay
  - Isolate test system
    - Modify DB Links, directory objects pointing to production systems
    - Isolate test system LAN (optional)
  - System clock setting
    - Reset system clock to same time as production if application logic involves SYSDATE usage

# Database Replay Case Study



# Case Study

- Evaluate benefits of Oracle 11g Advanced Compression feature
- System profile
  - Linux 32-bit Red Hat
  - 1 CPU hyperthreaded
  - 2 GB RAM
- Workload Profile
  - Mixed – OLTP (30%) and DSS (70%)
  - 10 TPS

# Case Study: Assessing OLTP Compression

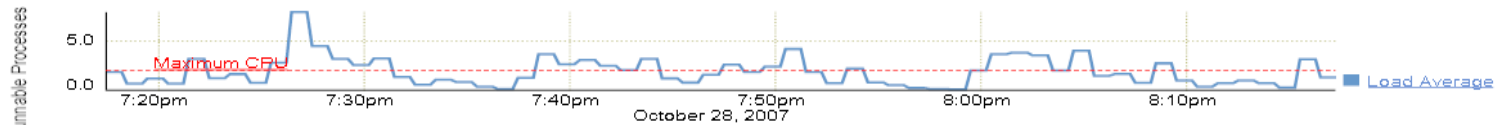
Database Instance: v3f.us.oracle.com

[Home](#) [Performance](#) [Availability](#) [Server](#) [Schema](#) [Data Movement](#) [Software and Support](#)

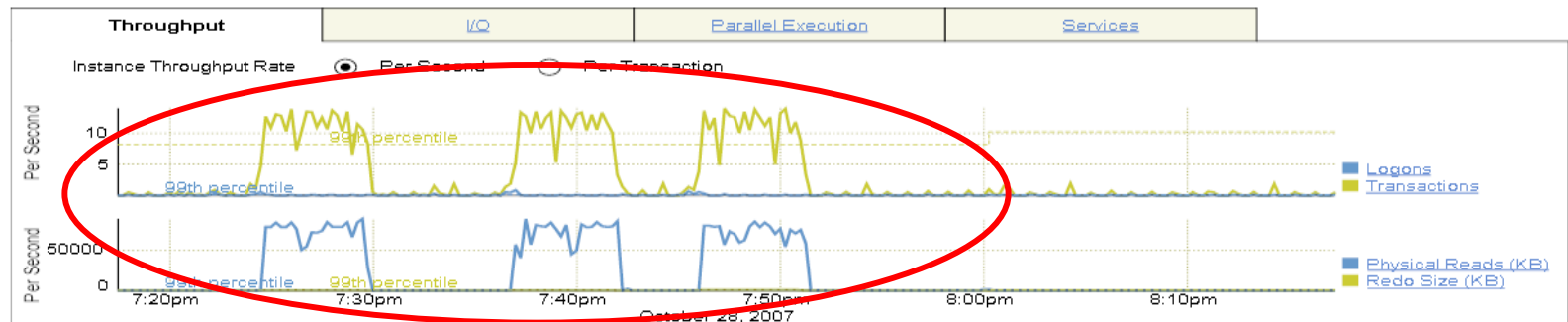
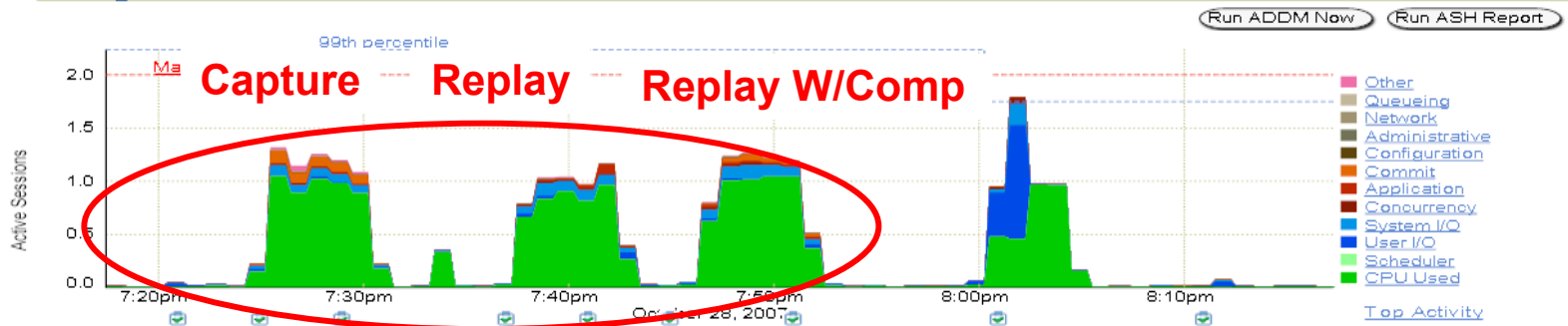
Baseline Name SYSTEM\_MOVING\_WINDOW

[Settings](#) [View Data](#) Real Time: 15 Second Refresh

Host



Average Active Sessions

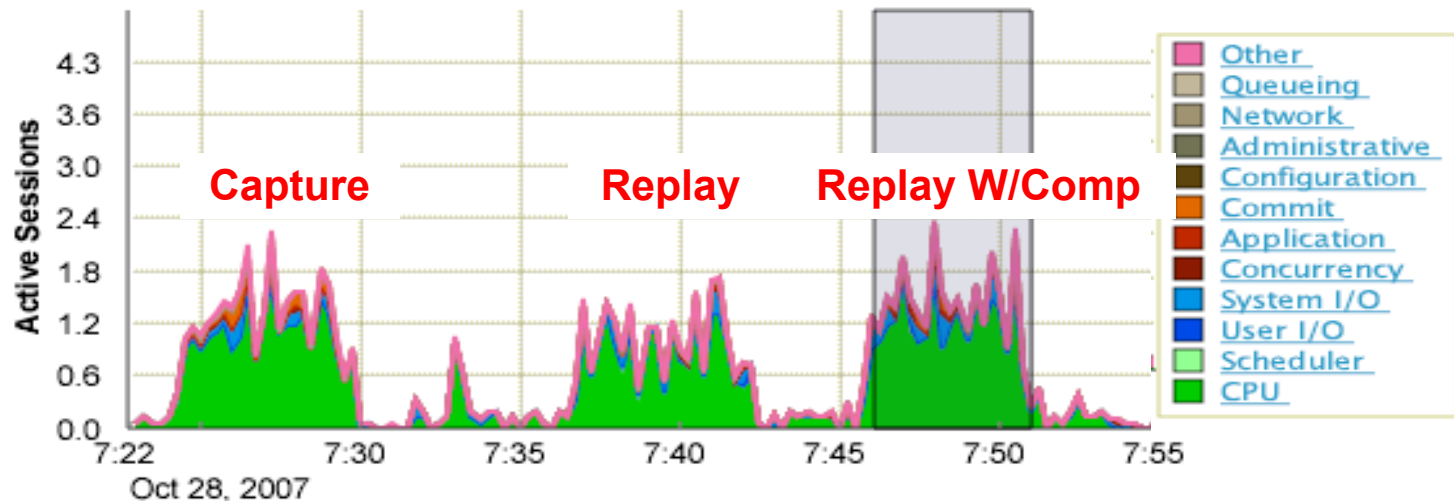


# Case Study: Assessing OLTP Compression

Database Instance: v3f.us.oracle.com >

## Top Activity

Drag the shaded box to change the time period for the



## Detail for Selected 5 Minute Interval

Start Time Oct 28, 2007 7:46:01 PM PDT

# Case Study: Assessing OLTP Compression

Table Size (Blocks)	9714
Table Size with Compression (Blocks)	1000
Compression Ratio	<b>9.71</b>

## Load Profile

	1st per sec	2nd per sec	%Diff	1st per txn	2nd per txn	%Diff
DB time:	1.08	1.23	13.89	0.10	0.12	↑ 20.00
CPU time:	0.80	0.93	16.25	0.08	0.09	↑ 12.50

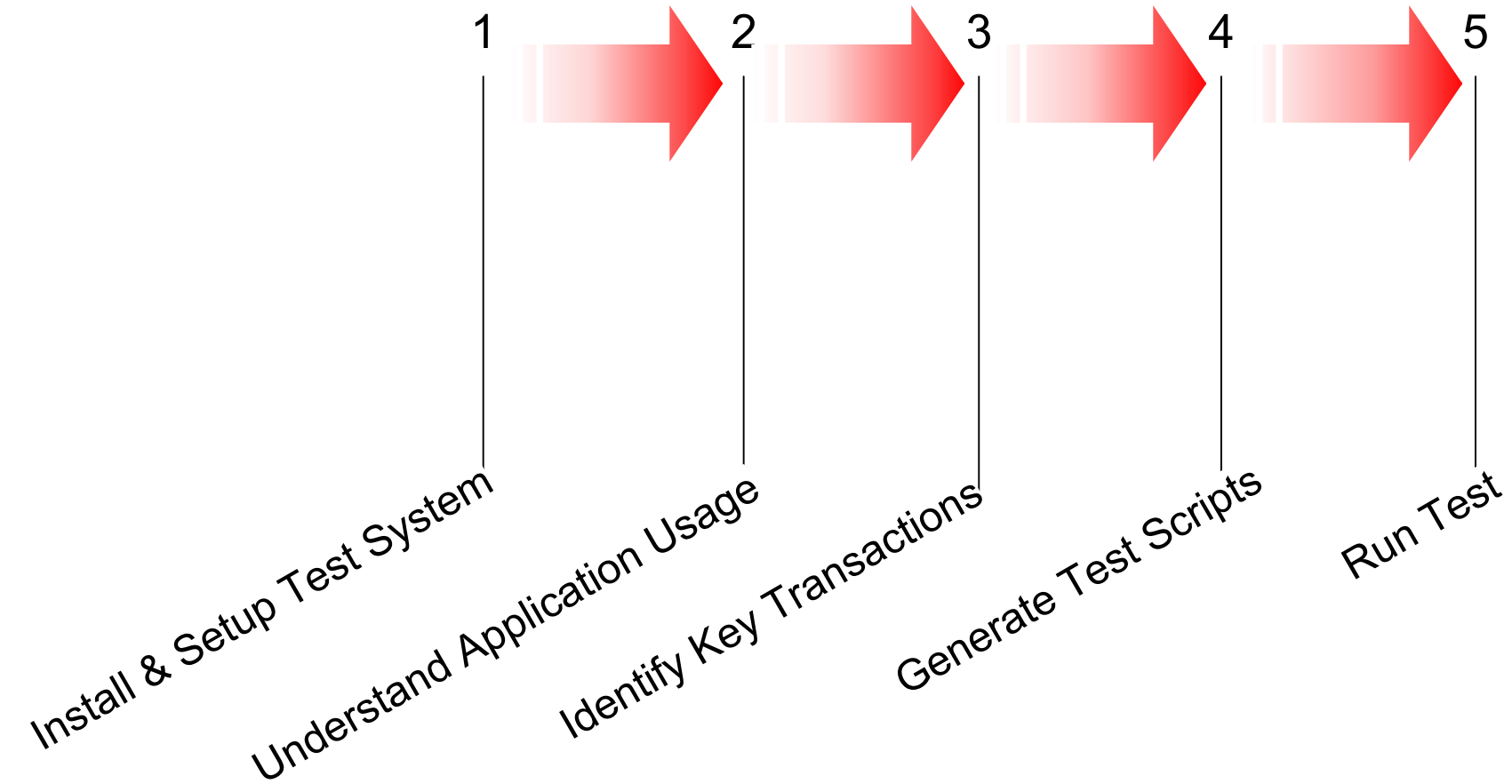
## Divergence

	Number of Calls
<b>Error Divergence:</b>	
Session Failures Seen During Replay	0
<b>Data Divergence:</b>	
DMLs with Different Number of Rows Modified	0
SELECTs with Different Number of Rows Fetched	0

# Database Replay Conclusion

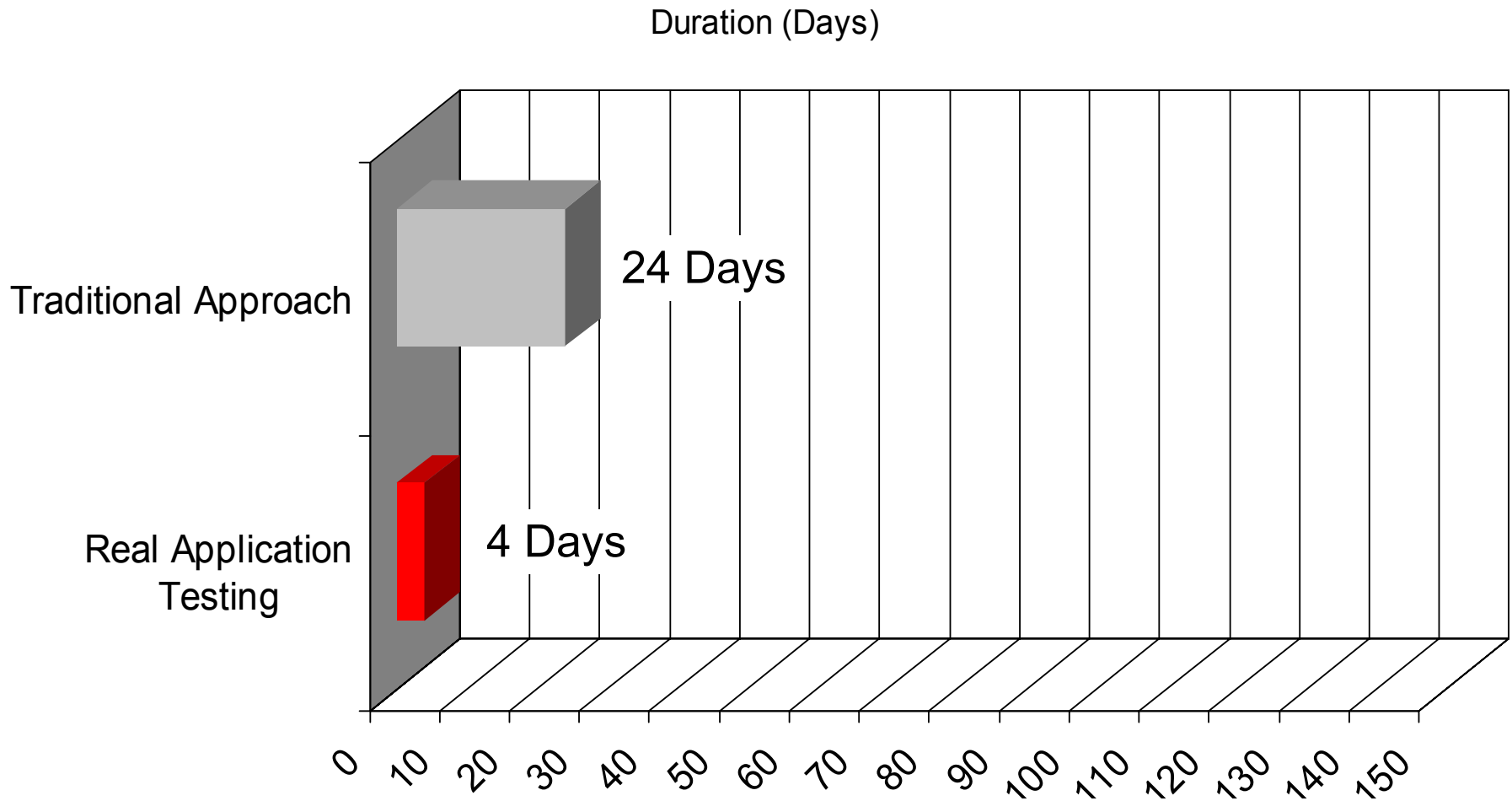


# Typical Steps in Test Phase

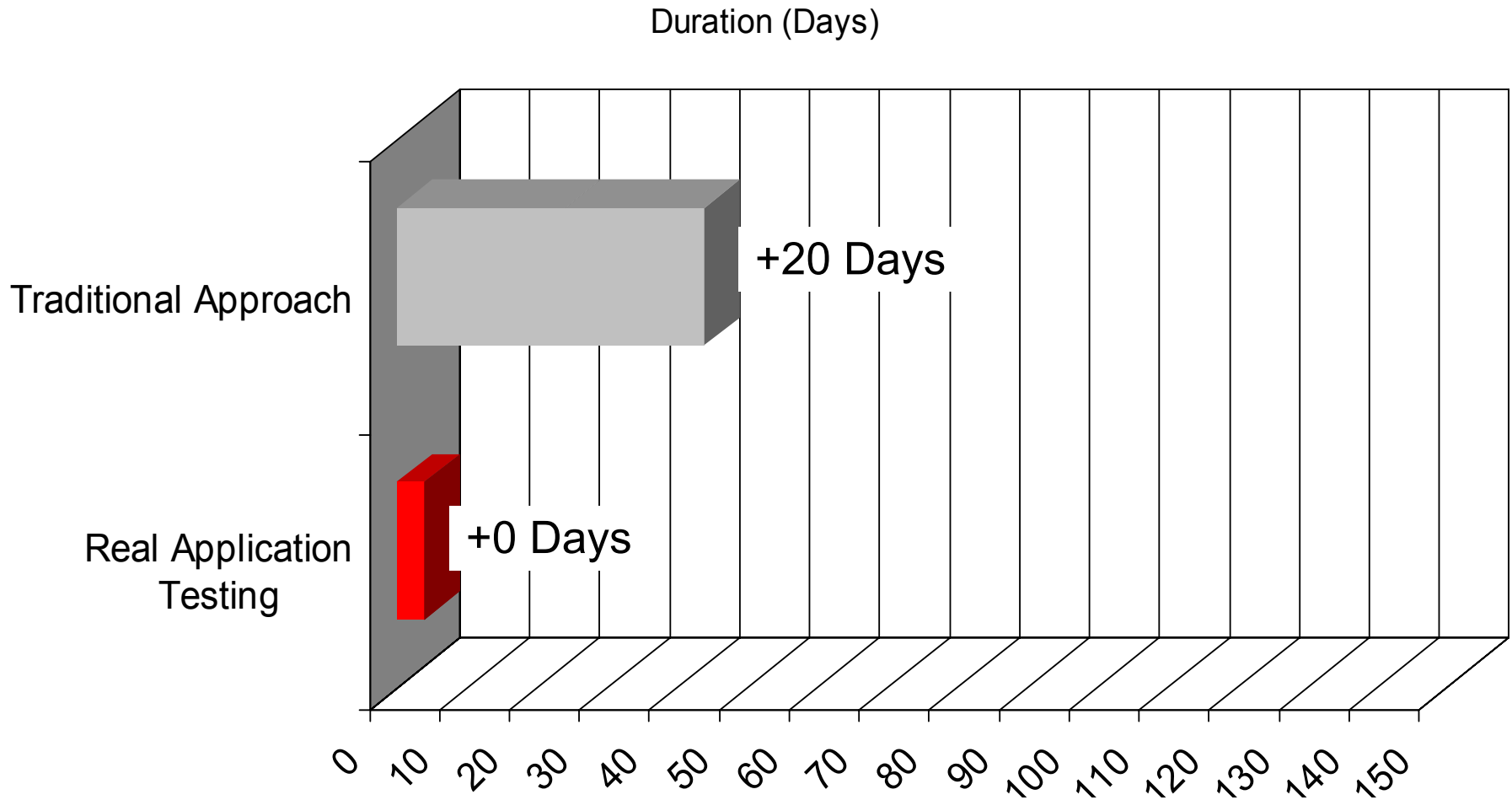




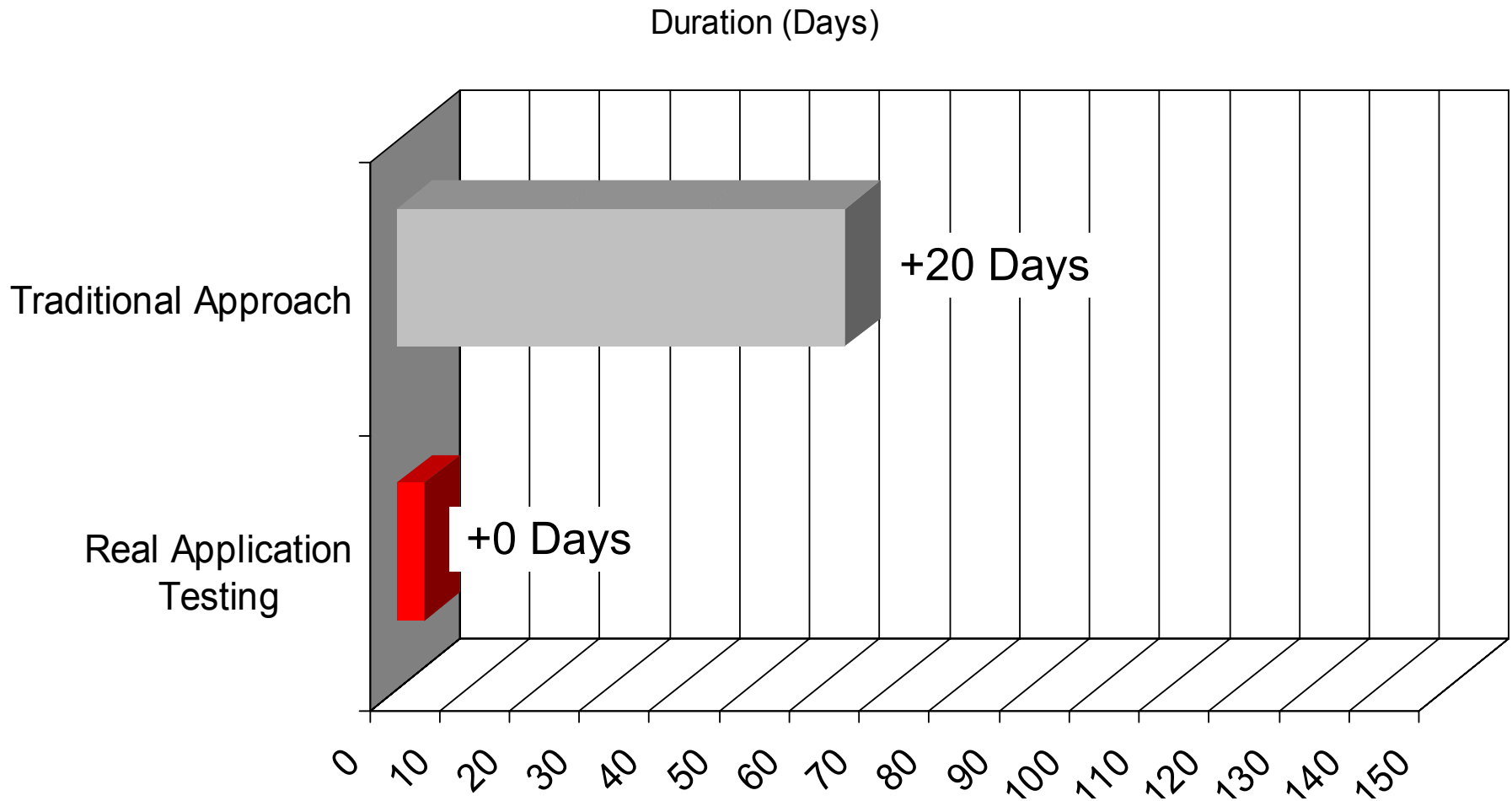
# 1: Install & Setup Test System



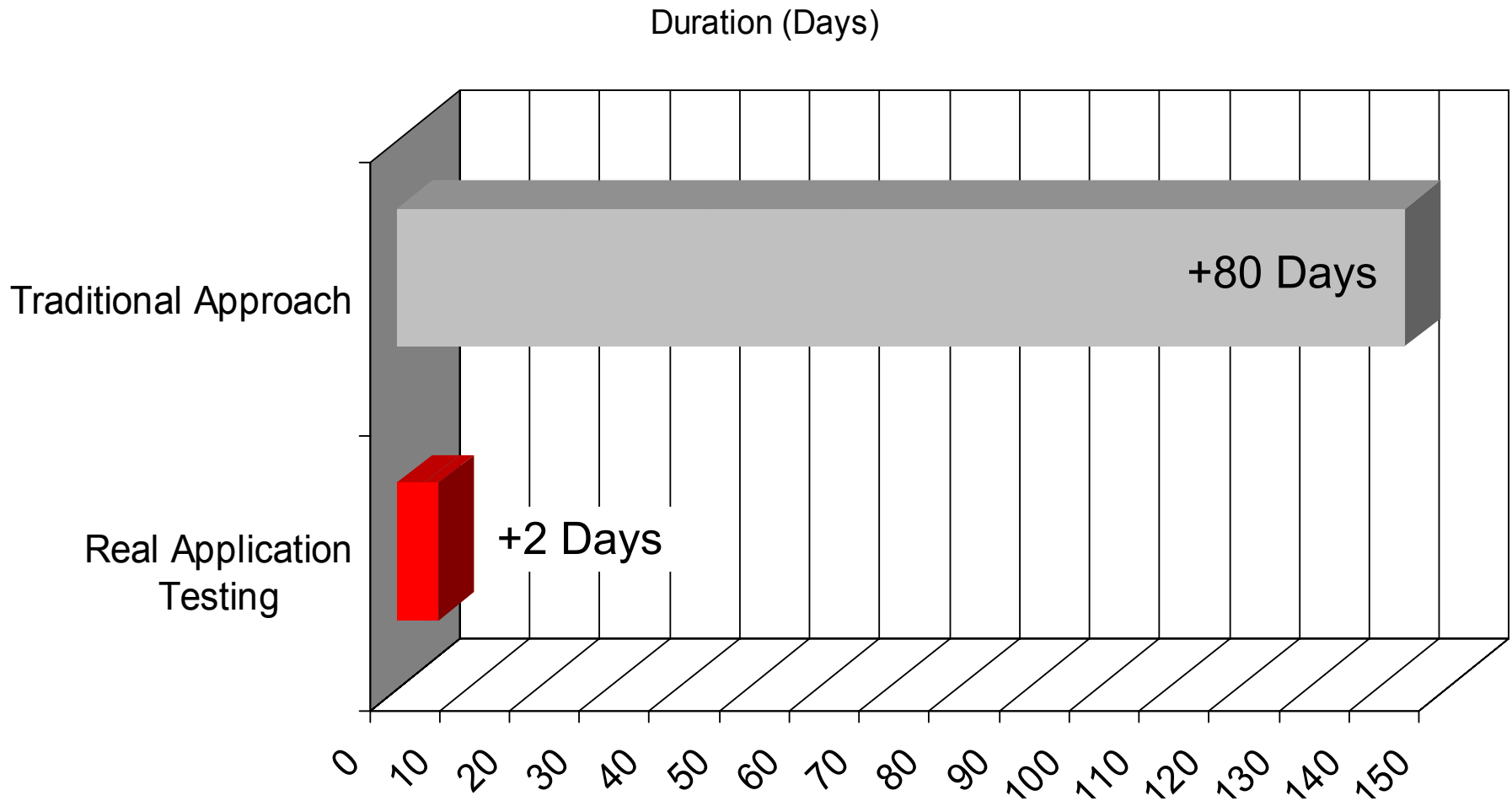
## 2: Understand Application Usage



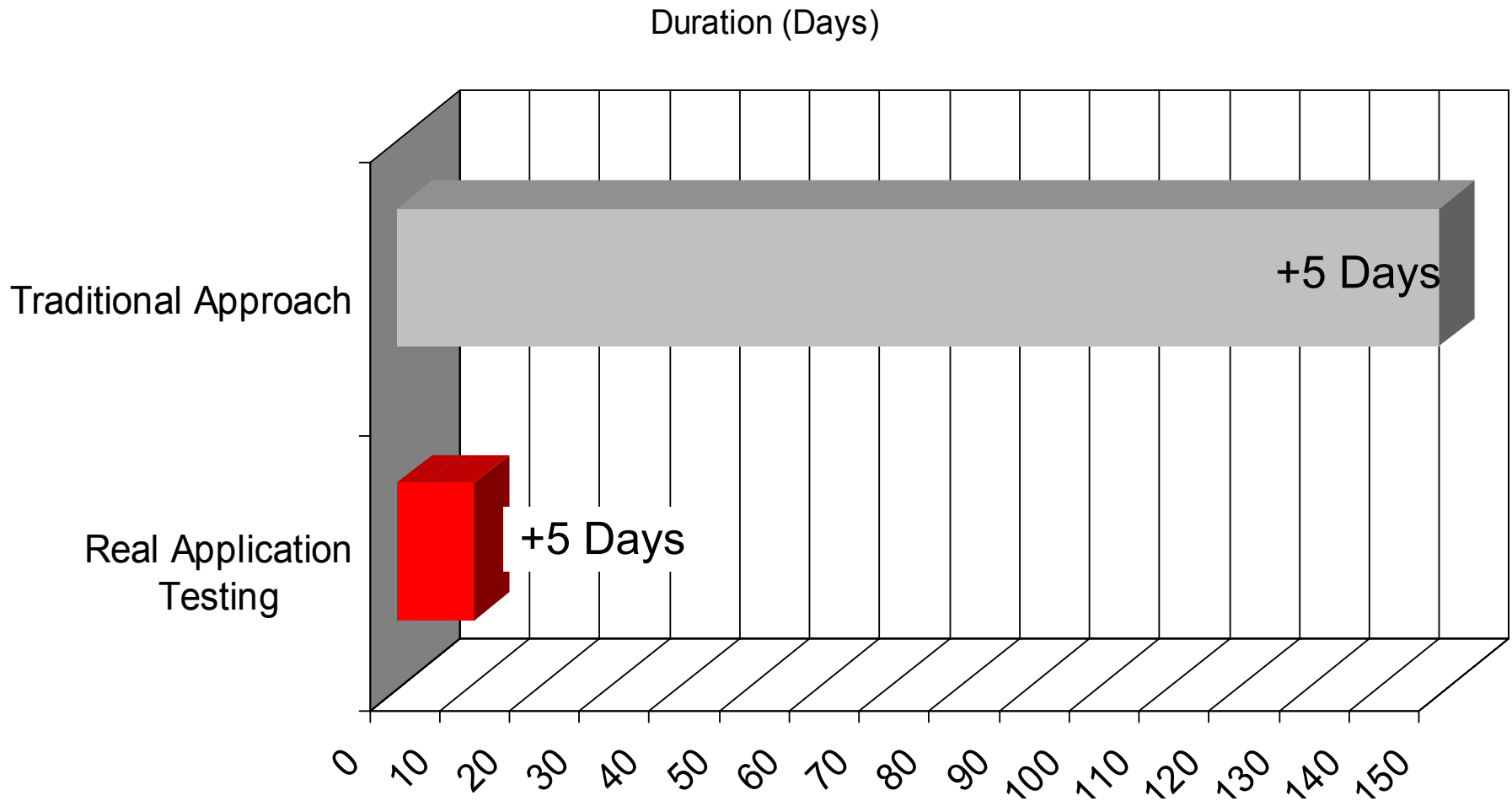
# 3: Identify Key Transactions



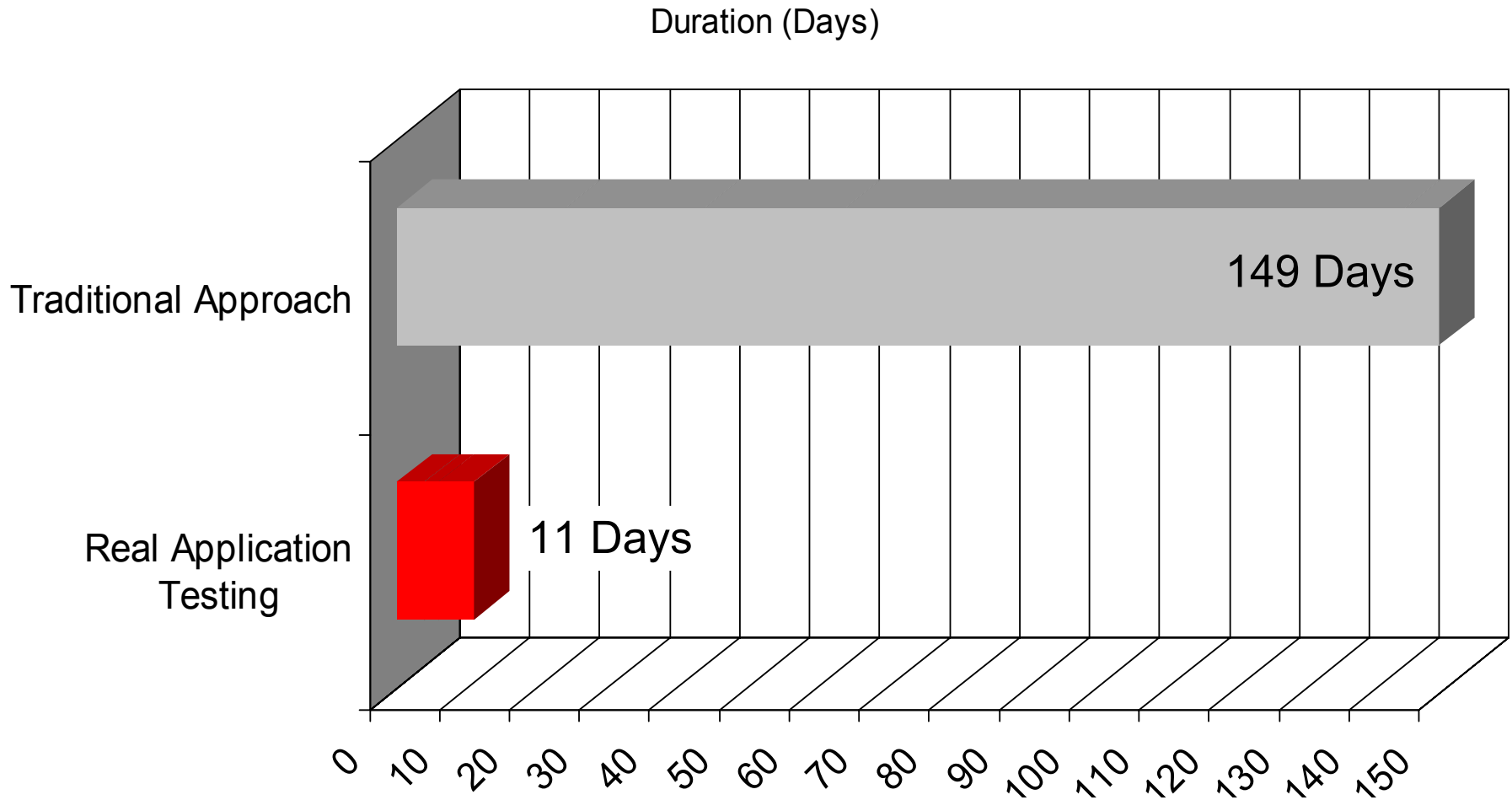
# 4: Generate Test Scripts



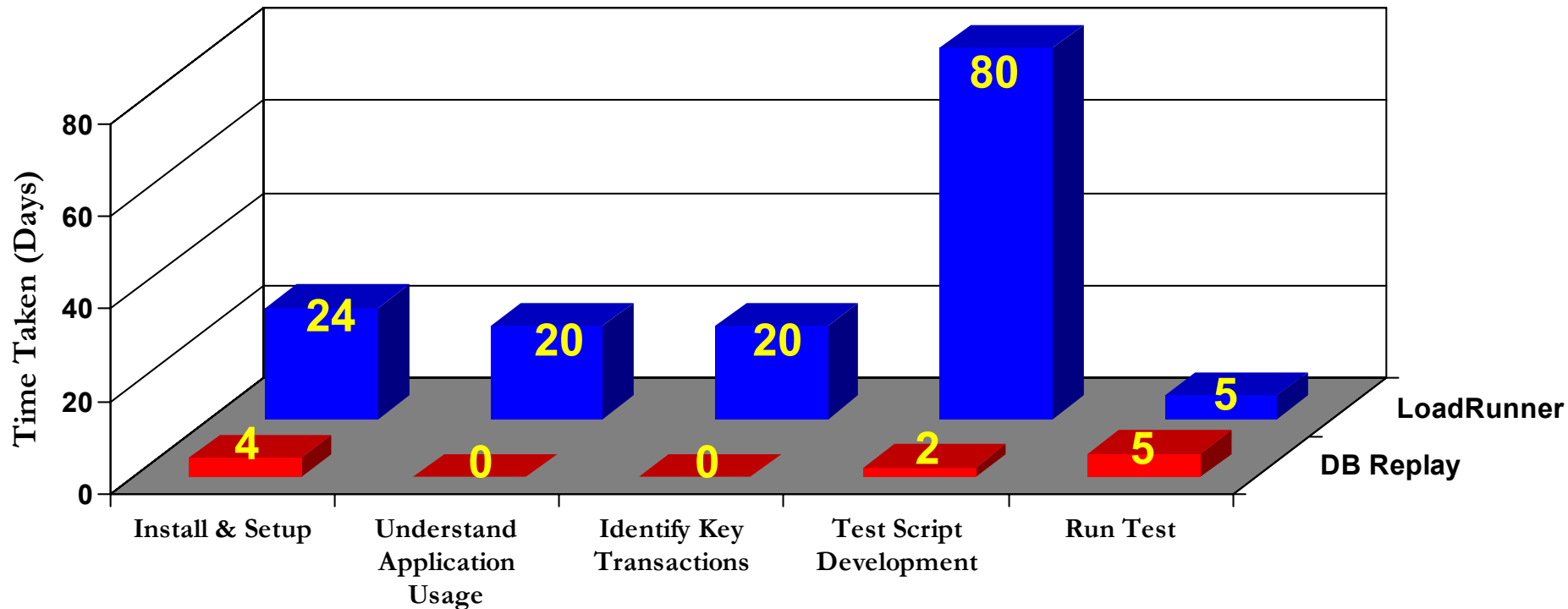
# 5: Run Test Workload



# Innovate Faster

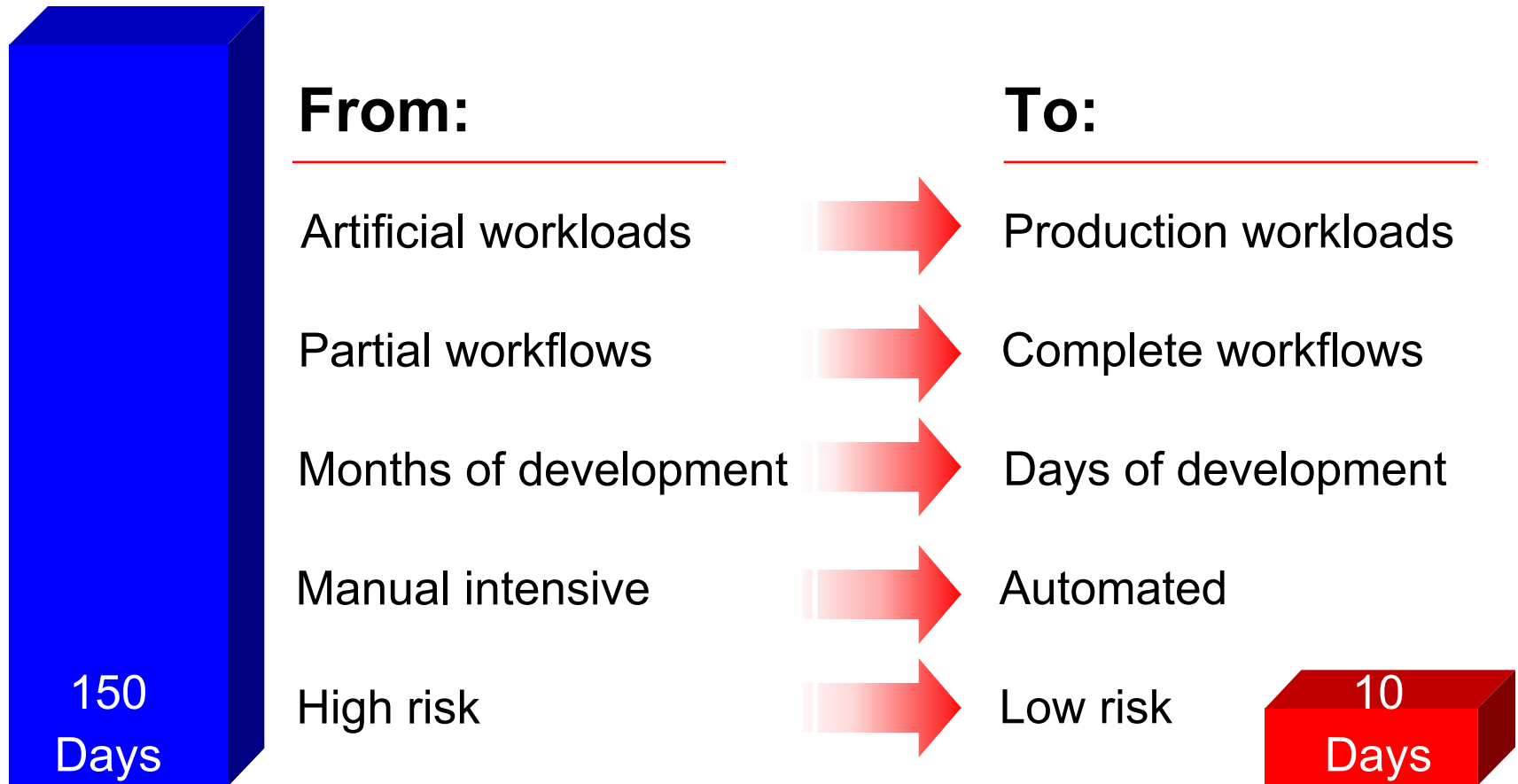


# Comparison of Traditional Solution & DB Replay: Testing Oracle e-Business Suite



Total Testing Time
DB Replay: 2 weeks
Traditional Solution: 30 weeks

# Why Database Replay?







David Mitchell  
Senior Vice President, OVUM

*“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.”*

# Summary

- Provides solution of assessing impact of change on real-world production systems
  - Holistic and actual workload testing results in lower risk
  - Brings down testing cycle from months down to days
  - Lowers hardware costs by eliminating need for mid-tier and application setup on test system
  - Maximizes ROI by leveraging Diagnostics and Tuning Pack to remediate regressions
- With Database Replay businesses can
  - Stay competitive
  - Improve profitability
  - Be compliant



ORACLE®