Data Caching Application Deployment Use Cases for Oracle TimesTen In-Memory Database

Susan Cheung
Director Product Management – TimesTen
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.
When You Think “Database…”

RDBMS + network connectivity

This may NOT be fast enough for some response-time-critical applications

Typical solutions: Build a home-grown, application-specific, in-memory buffer ‘cache’…..
What if you have...

- Full capabilities of a relational database
- Memory-optimized speed & latency
- Persistent, recoverable, highly available

The Cache and the Database in ONE product with

- Full capabilities of a relational database
- Memory-optimized speed & latency
- Persistent, recoverable, highly available
Oracle TimesTen In-Memory Database
Memory-Optimized RDBMS for Real-Time Enterprises

- Delivers real-time responsiveness and very high throughput
- Deployed in the middle tier, close to the application
- Operates as standalone database, or as a read/write cache for Oracle Database
Oracle TimesTen In-Memory Database

- **In-memory RDBMS**
  - Entire database in memory
  - Standard ODBC/JDBC, SQL 92
  - Compatible with Oracle Database
  - Embeddable

- **Exceptional performance**
  - Very low response time → high throughput

- **Persistence and recoverability**
  - Database persists to disk
  - Transactions with ACID properties

- **On-line, non-blocking real-time operations**
Why Is TimesTen So Fast?

• In-Memory Database
  • Entire database is always in memory
  • Designed and optimized for memory layout
  • No buffer cache management overhead
  • Shorter code path = faster performance
  • Fewer CPU instructions = fewer processors required

• Application can embed TimesTen DBMS into its process address space
  • Eliminate network and inter-process communication overhead
  • Very low response time (like calling a procedure)
Replication – TimesTen to TimesTen

- Real-time transactional replication
  - Between TimesTen databases
- Flexible configuration
  - Active-standby, Active-active, N-way
- High performance
  - Asynchronous replication
  - Synchronous replication
- Robust and reliable
  - Auto recovery, master catch-up
  - Conflict detection and resolution
- Support online upgrade
  - No downtime, cross version replication
Cache Connect to Oracle

- Improves application response time
- Cache tables from Oracle database
  - Cache individual tables and related tables
  - Cache all or subset of rows and columns
- Automatic data synchronization
  - TimesTen to Oracle
  - Oracle to TimesTen
- Access cache tables like regular database tables using SQL
  - Joins/search, insert/update/delete
Caching Data from Oracle Database

**Cache Group** – describes the Oracle data to be cached in the TimesTen in-memory database, via a SQL WHERE clause

**Example:** Caching name and address of premier customers who have had more than 500 purchase orders

CREATE CACHE GROUP PremierUsers
FROM CUSTOMER ( 
    NAME VARCHAR2(100) NOT NULL, 
    ADDR VARCHAR2(100) 
) 
WHERE CUSTOMER.ORDER > 500;
Read-only Cache Groups

- Read-only Cache Groups
  - Updates disallowed in TimesTen
  - Optionally updates can be sent to Oracle (via the Pass-through feature)
- Updates in Oracle automatically refreshed to the TimesTen Cache Group
- High availability and resilience to failures
Updatable Cache-Groups

- Transactions performed in TimesTen
- Updates write-through to Oracle
  - Synchronous or asynchronous
- Synchronous write-through transactions synchronously committed in Oracle
- Asynchronous write-through transactions batch-committed in Oracle
- High availability and resilience to failures
On-Demand Loading of Cache Data

- Some applications need to have user data loaded on-demand
  - The entire set of data is in the backend Oracle database
  - Load cache data from Oracle database if data not found in TimesTen
  - This feature can be turned ON/OFF by the application

- Load specific cache instance
  - SELECT query uses primary key or foreign key
  - Automatically load the related parent and child rows from the Oracle database
Automatic Data Aging

- Two types of data aging support
  - Time-based aging – based on timestamp values
  - Usage-based aging – based on LRU algorithm

- User configurable aging policy
  - Set policy on selected tables or cache groups

- Enable the application to keep only the most recent data in memory
  - Regulation compliance
  - Sliding window data caching
  - Database size control
Types of TimesTen Caches

- **Read-only Cache**
  - Updates in Oracle automatically refreshed to TimesTen

- **Updatable Cache**
  - Transactions committed in TimesTen and write-through to Oracle database
  - Asynchronous and synchronous write-through options

- **Static Cache**
  - Entire content of cache is pre-loaded prior to doing any work

- **Dynamic Cache**
  - Data loaded transparently from Oracle database if not found in TimesTen
  - Automatic data aging from TimesTen: LRU and timestamp-based
Multiple Read-Only Caches

Examples

**Oracle TimesTen (ref-data):**
- User profile
- Ticket symbols
- Service policies
- Flight schedule

**Oracle Database:**
- Flight reservation
- Reference data
- Historical data

Reference-data is heavily accessed by applications, “read-only”, and an ideal candidate for real-time caching
Multiple Updatable Caches

Examples

Oracle TimesTen:
- Pre-paid billing data
- Portfolio/positions
- Account balance
- Flight reservation

Oracle Database:
- DB of record
- Provisioning
- Order history
- Message storage
Dynamic Database Caching
Call Center Application Example

- Transparent loading of customer data from Oracle Database
  - Load customer data dynamically at the time of the call
  - Improve database responsiveness for subsequent operations
- Automatic data aging
  - Remove old or least-recently-used data to make room for new callers
Sliding Window Caching

Cache data from a specific time window for real-time management, monitoring and optimization

- Cache 5 days of shipments for real-time delivery status
- Cache last 15 minutes of RFID data for real-time process monitoring
- Cache last 90 days of orders to speed searching by call agents or self-service portals
- Cache last 30 days of market data for analytics & simulation
Benefits of Caching in TimesTen

- Reduced application response time
  - Read and Write transactions are completed more quickly in TimesTen
- Very high throughput
  - Multiple caches provide horizontal scaling
- Reduced workload on the Oracle Database
  - More computing resources are available for other workloads on the database tier
- Run on commodity hardware
  - TimesTen is available on 20 popular platforms
Cross-Tier High Availability

Integration with Oracle RAC

- Automatic recovery from Oracle Database RAC node failures using TAF and FAN
  - Automatic reconnection to the cluster
  - Automatic resumption of data refresh from Oracle to TimesTen
  - Automatic resumption transaction propagation from TimesTen to Oracle
  - No loss of transactions
Cross-Tier High Availability

Read-Only Caches

- Active-Standby Pair configured with 3 read-only subscribers
- Read-only cache groups
- Updates from Oracle database are
  - Refreshed to the TimesTen Active node
  - Replicated to the TimesTen Standby node
  - Replicated to the read-only subscribers by the standby
Cross-Tier High Availability
Read-Only Caches

• Failure of TimesTen Active node:
  • Standby becomes the new Active
  • No loss of transactions
Cross-Tier High Availability
Read-Only Caches

- Failure of TimesTen Active node:
  - Standby becomes the new Active
  - No loss of transactions

- Failure of TimesTen Standby node:
  - Active assumes the additional role of the standby
  - No loss of transactions for TimesTen read-only nodes
Cross-Tier High Availability

Updatable Caches

- Active-Standby Pair configured with 3 read-only subscribers
- Updatable cache groups
- TimesTen transactions committed in the Active node are:
  - Replicated to the Standby
  - Write-through to the Oracle database, and
  - Replicated to the read-only subscribers by the standby
Cross-Tier High Availability

Updatable Caches

- Failure of TimesTen Active node:
  - Standby becomes the new Active
  - No loss of transactions
Cross-Tier High Availability

Updatable Caches

- Failure of TimesTen Active node:
  - Standby becomes the new Active
  - No loss of transactions

- Failure of TimesTen Standby node:
  - Active assumes the additional role of the standby
  - No loss of transactions for TimesTen read-only nodes
Oracle TimesTen

Customer Use Cases
Proven in Real-Time Deployments
Thousands of Companies Use Oracle TimesTen
Electronic Ticketing Java Application

- Largest ticketing company in Japan
  - Concerts, sporting events
- Existing home-grown solution
  - Expensive to maintain
  - Need better performance to keep up with business growth
- Need to improve performance during peak loads
  - First day of ticket sales for an event
- Solution must be
  - High performance
  - Lower cost
Electronic Ticketing Java Application

1. Search for vacant seats
2. Hold the vacant seats
3. Collect payment info
4. Complete the reservation

Operators, Phones, Websites, Devices at stores...

EJBs

Nightly Batch Load of Seating Data

Oracle RAC

JSP / Servlets
Electronic Ticketing Application

• Configuration
  • IBM Blade servers (x86), AIX 5.3
  • Oracle TimesTen 7.0 with Cache Connect to Oracle
  • Oracle RAC 10g
  • Oracle Application Server
  • Java 1.5, 64-bit application

• Why TimesTen
  • Throughput performance in the application-tier
  • > 300 round-trip reservations per second
  • Easy development using standard interface (Java)

• Application deployment
  • Planned for January 2008
Pre-Paid Real-Time Billing
T-Mobile Germany

- Customer is one of the largest mobile operator in Germany with over 80 million subscribers

Business Requirements
- Standards based infrastructure software
- Ease of development and maintenance
- Fast time to market
- High performance
Pre-Paid Real-time Billing
Pre-Paid Real-Time Billing

• **Why TimesTen**
  • Performance – low response time and high throughput
  • Standards based solution
  • Easy development and maintenance

• **TimesTen usage – in production since 2004**
  • Event capture (active prepaid sessions & volume status)
  • Reference data lookups (volume & content charging)
  • Balance management (prepaid authentication/charging)

• **Configuration**
  • Use TimesTen as a cache in front of Oracle Database
  • Transactional updates for volume discounts and content charging in TimesTen and then propagated to Oracle Database
Service Analysis and Routing Platform
SMART Communications

- Leading wireless services provider in Philippines with 28M subscribers (~58% market share)
- Competitive business climate – need to differentiate
  - Flexibility and speed in offering compelling services
  - Ability to customize/personalize

- Oracle TimesTen used in multiple applications
  - Pre-paid billing
  - SMS database to manage storage & delivery of SMS messages
  - Service analysis to determine customer service type for rating/charging
Service Analysis and Routing

- Retrieve subscriber profile
- Perform service analysis to determine service type
- Route to service engine based on rating/charging rules

Configuration

- Oracle TimesTen
  - 64-bit Red Hat Linux, Sun Opteron 8-way server
  - TimesTen database: 5-15GB of database per node
  - 6 pairs of replicated nodes for HA and redundancy
- Oracle RAC
  - 3-node cluster, 12-way Sun Sparc servers, Solaris 10
  - EMC DMX3 storage
Service Analysis and Routing Platform

- Six pairs of TimesTen replicated databases, caching data from a central Oracle RAC Database
Service Analysis and Routing

- **TimesTen Cache Configuration**
  - 3 cache groups per TimesTen data store, 1 cache group for each Oracle base table
  - Incremental auto refresh

- **Why TimesTen**
  - Low and predictable response time
  - Geographically distributed scalable architecture for data caching
  - Flexible caching options
  - Tight integration with Oracle Database
Mutual Fund Trading
Global Broker-Dealer

- Customer is a global broker/dealer with operations in UK, USA, and Tokyo
- Program trading serving institutional clients (hedge fund, mutual funds)

Challenges
- A combination of disk-based RDBMS plus Java/J2EE architecture could not sustain the high order volume of program trading
- J2EE caching of full objects was too slow
- Homegrown Java object cache too expensive for in-house staff
**Mutual Fund Trading**

- Institutional Investors
- Message Bus
- Orders, Inquiries, Notifications
- Pre-Trade Validation
- Order Prep and Release
- Post-Trade Allocation
- Active
- BEA WLS
- ORACLE TIMESTEN
- Standby
- ORACLE TIMESTEN
- XLA Application
- RDBMS Direct Connect (ODBC)
- Aged Order Data
- Shared Global Data
- Historical Reporting
- Reference Data
- Global Order Repository

*Other applications are also running on this server*
Mutual Fund Trading

• Configuration
  • TimesTen is the trading application database
  • Application in Java using BEA WebLogic Application Server
  • TimesTen Replication for HA
  • IBM MQ for orders, inquiries, and notifications

• Why TimesTen?
  • Standards based, commercial product
  • Order of magnitude performance improvement
  • Can sustain high volume orders

• TimesTen usage – in production since 2003
  • Reference data lookup
  • Order transaction processing
  • Event publishing
Meet SLA for Routing Application
Deutsche Börse – German Stock Exchange

**Requirement**
- Meet SLA of 80ms round trip for 95% of trade orders for active retail customers

**Solution**
- Use Replicated TimesTen Cache with Oracle Database

**TimesTen Value**
- Bounded response time
- HA with no data loss
- Ease of implementation
Subscriber-Centric Policy Management

Bridgewater Systems

- Policies must be enforced in the path of a call
- High performance achieved by using TimesTen for caching data

• Database requirements:
  - Low latency
  - Scalability in size and throughput
  - No data loss or corruption
  - Geographic distribution
  - Online upgrades

State Transaction Routing Function

Policy Decision Function

Policy Decision Function

Policy Decision Function
Summary

• Oracle offers industry-unique in-memory technology to enable Real-time business
  • Proven technology with thousands of deployed customers

• Oracle TimesTen
  • Offers real-time performance for enterprise and embedded applications
  • Extends Oracle Database to improve application performance
  • TimesTen offers predictable response-time
For More Information

Oracle TimesTen Product Center on OTN:
http://oracle.com/technology/products/timesten

• Technology white papers
• Quick Start Guide and tutorials
• Discussion Forum
• And more..