

# ORACLE IN-MEMORY DATABASE CACHE

## OVERVIEW

### ORACLE<sup>®</sup> 11g DATABASE

- Real-time performance for Oracle data
- Unlimited read/write caching
- Concurrency, Persistence, and Durability
- Oracle Database Integration
- Flexible “cache group” definitions
- Automated data loading and refresh
- Transactional update propagation
- Transparent SQL pass through
- Dynamic data caching and data aging
- High availability
- Automatic resynchronization
- Web-based configuration utility

*Oracle In-Memory Database Cache is an Oracle Database product option ideal for caching performance-critical subsets of an Oracle database for improved response time in the application tier. Cache tables can be read-only or updatable. Applications read and update the cache tables using standard SQL, and data synchronization between the cache and the Oracle Database is performed automatically. Oracle In-Memory Database Cache offers applications the full generality and functionality of a relational database, the transparent maintenance of cache consistency with the Oracle Database, and the real-time performance of an in-memory database.*

### Real-Time Performance for Oracle Data

The majority of data in corporate databases is historical and infrequently accessed. But, buried within this data are pockets of information that should be instantly accessible when requested. Your best customers, open orders, recent transactions and product catalog are examples. Oracle In-Memory Database Cache enables such information to be cached and managed in Oracle TimesTen in-memory databases for real-time access with automatic data synchronization between the in-memory cache and the Oracle database.

By managing data in memory, and optimizing data structures and access algorithms accordingly, database operations execute with maximum efficiency, achieving dramatic gains in performance, even compared to a fully cached disk-based RDBMS.

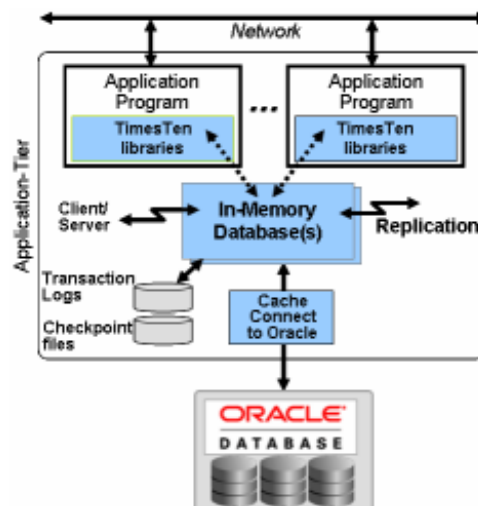


Figure 1. Oracle In-Memory Database Cache Components

Real-time data management has two performance dimensions – response time and throughput. With Oracle TimesTen In-Memory Database, a transaction that reads a database record can take less than 5 microseconds (a microsecond is one millionth of a second), and transactions that update or insert a record can take less than 15 microseconds (measured with Red Hat Linux on AMD Opteron 1.8Ghz processor). Consequently, throughput is measured in tens to hundreds of thousands of transactions per second, using commodity hardware.

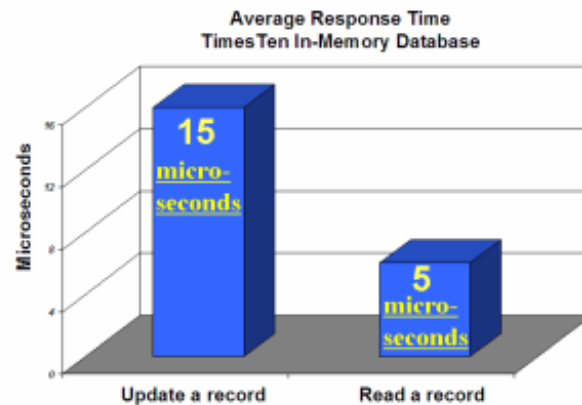


Figure 2. TimesTen In-Memory Database Average Response Times

### Unlimited Read/Write Caching

Most caching products are read-only, and thus limited to a narrow set of functions. Because Oracle In-Memory Database Cache supports full read/write SQL operations, it can be used for a wide range of applications. For example, capturing and processing data at network speeds, online commerce, securities trading systems, billing, real-time business process monitoring, real-time data analysis, and other transactional systems for which a read-only cache isn't applicable. Other examples include reference data, such as subscriber profiles, and look-up tables for authorizations and network configurations, where the cache is loaded at the start of each processing cycle then updated as changes occur.

### Multi-User Concurrency, Persistence and Durability

Oracle In-Memory Database Cache is deployed in the application tier for multi-user and multi-threaded applications using row-level locking and committed-read isolation. The in-memory cache tables are persistent and recoverable. Durability is achieved through a combination of transaction logging and database checkpointing to disk by the Oracle TimesTen In-Memory Database component.

### Oracle Database Integration

Oracle In-Memory Database Cache is specifically designed to use Oracle Database features and interfaces. This product option supports the same recommended database character sets and mainstream data types as the Oracle Database, ensuring that data can be stored and processed with compatible semantics and behaviors. To enable maximum application transparency, In-Memory Database Cache supports an SQL "pass through" feature, which allows the applications to transparently send SQL operations that cannot be satisfied by the in-memory cache data to the Oracle

Database for execution. With the pass-through feature, applications only need to establish one connection to the In-Memory Database Cache as the connection to the Oracle Database is managed transparently. Users can configure different settings to specify what types of statements are to be passed through and under what circumstances.

### Flexible “Cache Group” Definitions

In-Memory Database Cache uses the concept of a “cache group”, which describes a collection of in-memory database tables that map to all or a subset of the tables in an Oracle Database using extended SQL syntax. A cache group can consist of all or a subset of the rows and columns in these tables. Multiple cache groups can be used to cache different sets of related tables in the Oracle Database. There are two basic categories of cache groups:

**SYSTEM-MANAGED** cache groups provide pre-determined caching behaviors that are managed entirely by In-Memory Database Cache. Data synchronization between the Oracle Database and the cache tables are automatically managed without application intervention.

**USER-MANAGED** cache groups allow the applications to select from all of the attributes and SQL statements to define customized caching behaviors. The applications are responsible for manually initiating data loading, propagation, and refresh mechanisms that best meet their application requirements.

### Automated Data Loading and Refresh

In-Memory Database Cache provides a number of options for determining how and when data from the Oracle database is loaded into the in-memory cache tables, depending on the type of cache group created. SQL operations, such as LOAD and REFRESH, are used to populate a cache group and periodically update the contents, either in whole or incrementally.

### Transactional Update Propagation

Updates to the in-memory cache tables are asynchronously or synchronously written through to the Oracle Database, depending upon the desired trade off between performance and consistency. Synchronous write-through will ensure that, if the Oracle Database cannot accept the update(s), the transaction is rolled back from the in-memory database. Whereas asynchronous write-through leverages the speed of Oracle TimesTen In-Memory Database by first committing the transactions locally, and then using asynchronous write-through to send the updates to the Oracle Database. Asynchronous write-through cache groups provide better application response time and transaction throughput.

For read-only cached groups, incremental updates in the Oracle Database are asynchronously refreshed to the in-memory cache tables in the application-tier, at user-specified intervals.

### Dynamic Data Caching and Data Aging

For some applications, the set of cached data cannot be pre-determined. These applications require that the cached data be loaded on-demand when the data is needed. For example, a Call Center application may choose to load the caller’s

**IN-MEMORY DATABASE  
CACHE****REAL-TIME DATA  
MANAGEMENT FOR  
PERFORMANCE-CRITICAL  
APPLICATIONS:**

- Telecom and networking
- Capital markets
- Defense and intelligence
- Travel and reservations
- Service-Oriented Architectures (SOA)
- Middle-tier data caching
- Operational data stores
- Business intelligence
- Business activity monitoring
- Self-service portals
- Call centers

**RELATED PRODUCTS  
AND SERVICES:**

- Oracle TimesTen In-Memory Database

current billing statement when routing the call to a qualified call agent, rather than pre-loading the entire set of billing statements for all customers into the in-memory cache. A complementary feature for dynamic data loading is the automatic data aging feature to remove old data or least-recently-used (LRU) data. Data aging enables the applications to free up space for new cache data while keeping the “hot” data available in memory.

**High Availability**

Availability is an essential requirement for most real-time applications. High availability for in-memory cache tables is provided through real-time transactional replication.

- Transaction-log based replication scheme enables high efficiency and low overhead. The replication agent on the master database reads the records from its transaction log and forwards any relevant changes to the agent on the subscriber database, which then applies the updates to its local database. If the subscriber agent is not running, the master retains the log records until they can be applied at the subscriber.
- Asynchronous replication provides maximum performance, and the application is completely decoupled from the receipt process of the replicated elements on the subscriber.
- Synchronous replication provides higher level of confidence for data consistency between the master and subscriber databases, and the application is blocked until replication confirms that the update has been received and possibly committed on the subscriber

**Automatic Resynchronization**

In-Memory Database Cache is designed to continue running even after the Oracle Database server or network connection has been lost. Committed transactions to the in-memory cache tables are tracked and, once the connection is restored, propagated to the Oracle Database. Similarly, committed transactions to the cached tables in the Oracle Database are tracked and refreshed to the in-memory cache tables, once the connection to the Oracle Database is restored.

**Web-based Configuration Utility**

In-Memory Database Cache provides a Web-based GUI, the Cache Administrator, to create and manage cache groups.

For more information about Oracle In-Memory Database Cache, visit [http://www.oracle.com/technology/products/timesten/imdb\\_cache](http://www.oracle.com/technology/products/timesten/imdb_cache)

Copyright 2008, Oracle. All Rights Reserved.

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor is it subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.