ORACLE

Accelerate your Applications with True Cache

Self-Refreshing, Consistent, High-Performance, Scalable, Secure and Transparent SQL and Object Cache

Shailesh Dwivedi

Vice-President, Product Management and Cloud Engineering

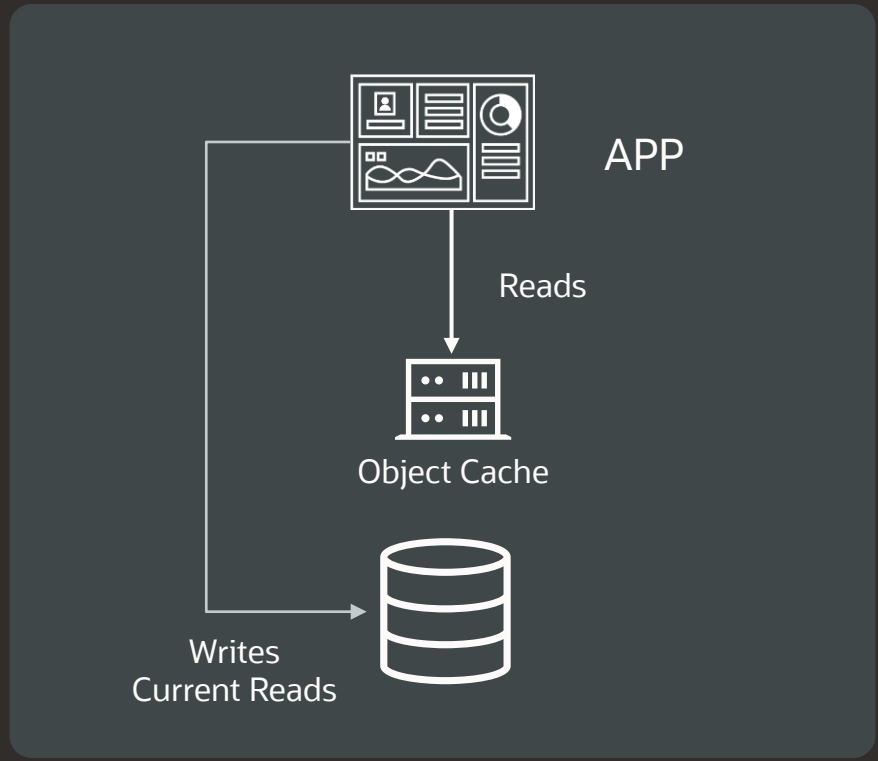
Cache use in Applications

To accelerate performance, applications use a separate cache cluster

- Examples include Redis & Memcached

Reads that do not need the most current data are directed to the cache

Writes and current reads go to the backend database since that is the source of truth



Shortcomings of Conventional Caches such as Redis or Memcached

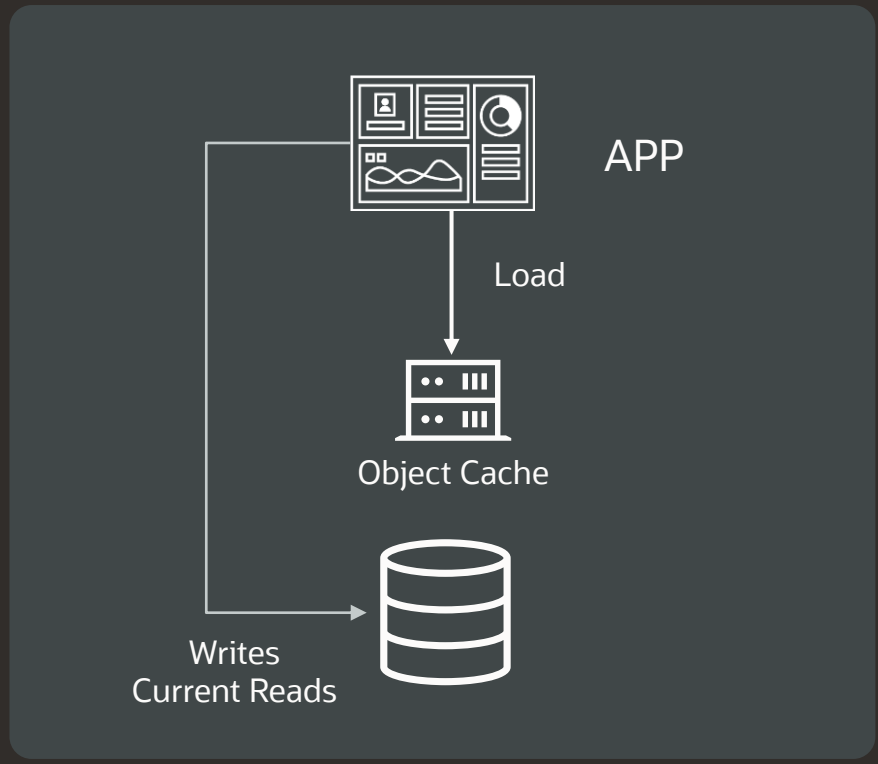
“We had a performance problem, and now we also have a Caching Problem”

Developers are responsible for loading the cache

When application starts, it reads the data from the database and loads it in the cache

Subsequently, on each cache miss, the application reads the data from the database, loads it in the cache, and then returns the data to the application

Developers have to write custom code for loading the data in the cache



Developers are responsible for cache consistency with the Database

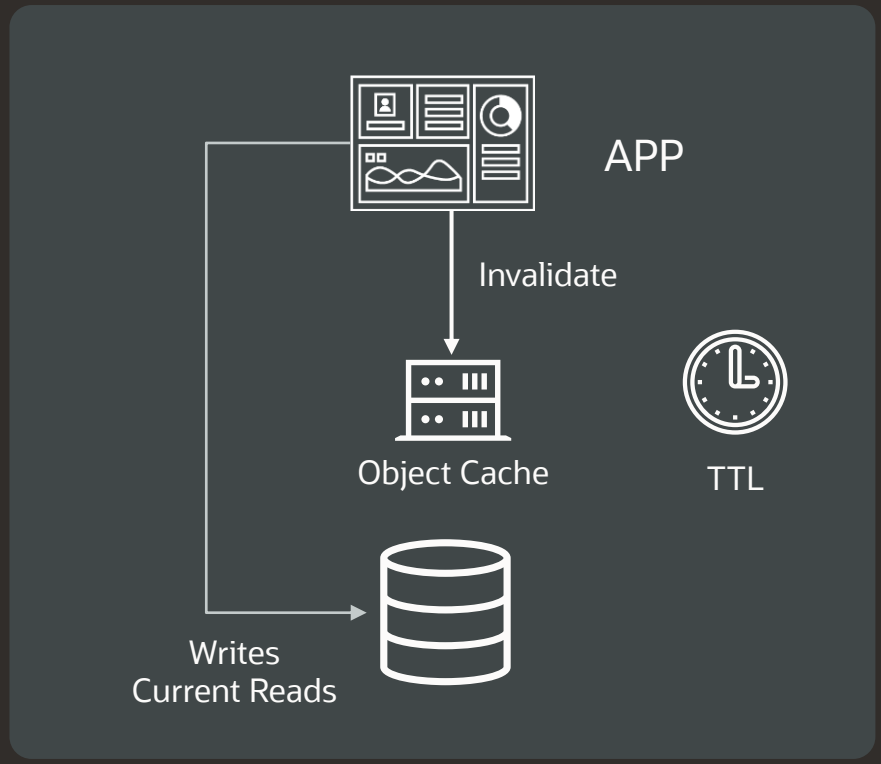
Data in the cache becomes stale as the data in the database is updated

- Developers have to write custom code to keep the cache in sync with the database

Some caches work around this by allowing applications to configure “Time to Live” for the data in the cache

- This doesn't work well: Frequent invalidations reduce cache effectiveness and add load to the backend database, less frequent invalidations result in a higher chance of getting stale data

Schema changes in the database could invalidate entire cache content



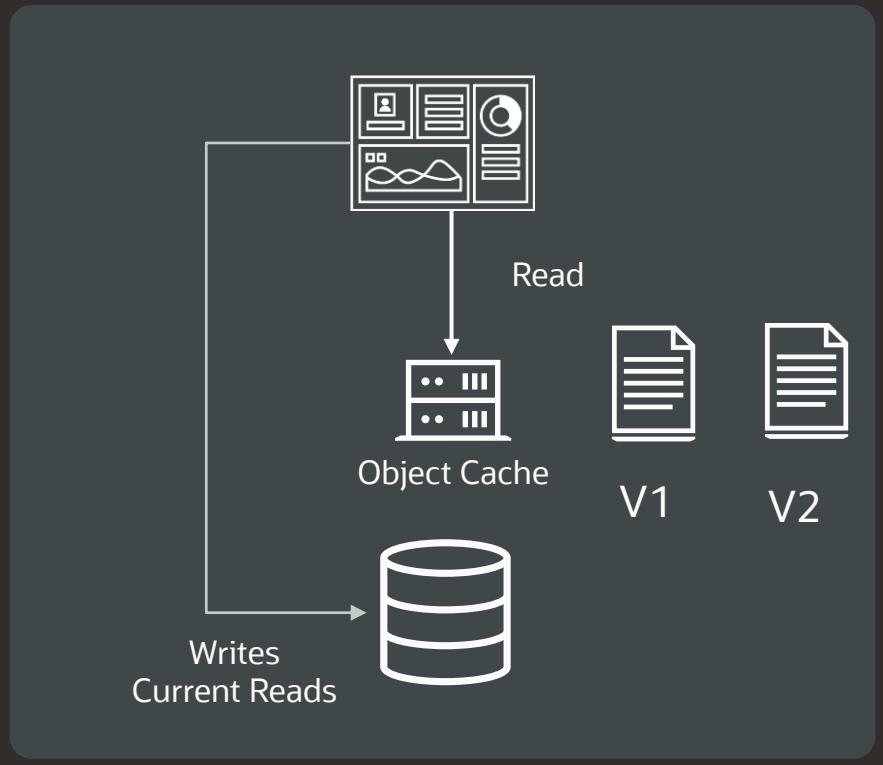
Developers are responsible for cache consistency with other caches

In Object Caches, each object is independently maintained and is likely to be from different points in time & hence inconsistent (your accounts won't balance)

- Developers are responsible for maintaining data consistency and integrity in the cache.

It is hard to keep the data in the cache correct. Hence, certain types of data are not cached

- For example, complex objects with nested relationships are not cached



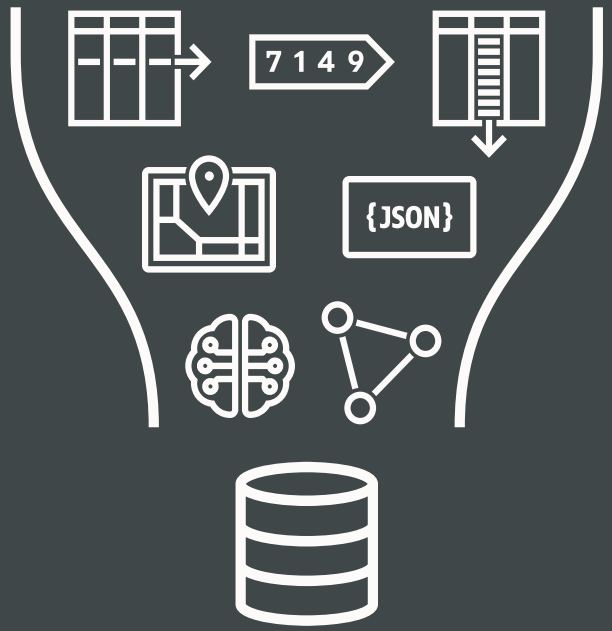
Limited data type and data format support

Object caches support only scalar data types like string, integer, JSON

Data cannot be accessed in Columnar formats for Analytics/Reporting purposes

Oracle database supports a rich variety of data types like relational, JSON, text, spatial, graph, and vector and allows access in a row as well as columnar format

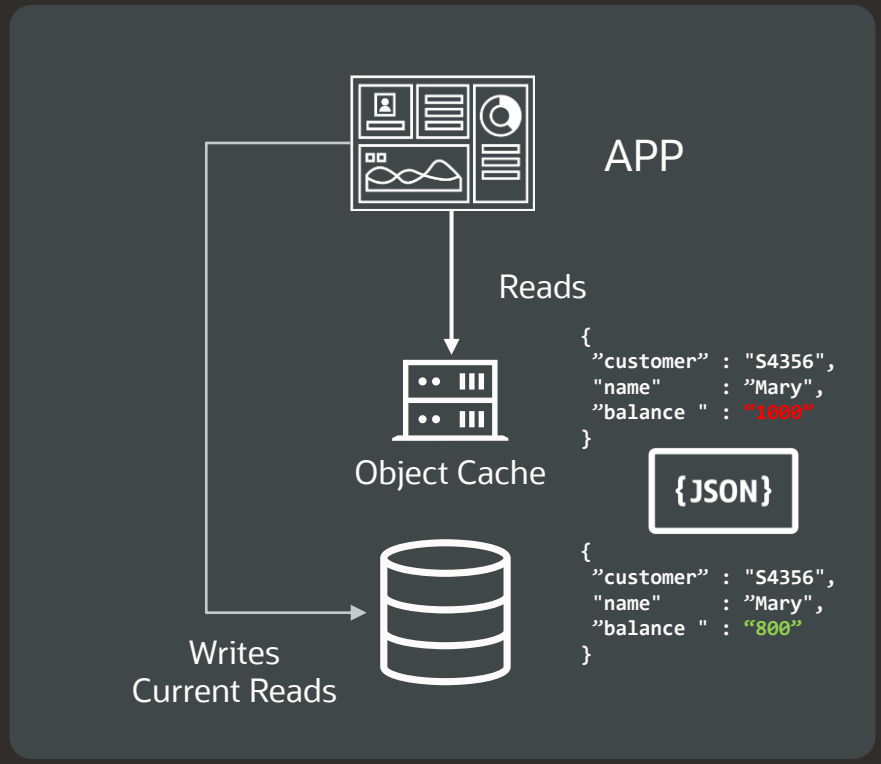
Developers must handle the conversion between the rich datatypes and formats supported by the Oracle database and the simple types supported by object caches



JSON Document exacerbates Caching Problem

Full JSON document has to be reloaded in the cache even if only one attribute is changed in the database

- This results in increased I/O and load on the backend database



Missing Enterprise-Grade Features

Object caches lack performance features like multi-threading, parallel processing, sub-partitioning, secondary indices

- Often, large cache clusters are deployed to compensate for it

Object caches do not have advanced security features like user and privilege management, strong encryption, object-level security, row-level security, disallowing admins access to data in the cache

- Hence, sensitive data cannot be stored in the cache

Object caches also lack enterprise-grade availability, manageability, and observability, which results in lower application SLAs

Oracle True Cache

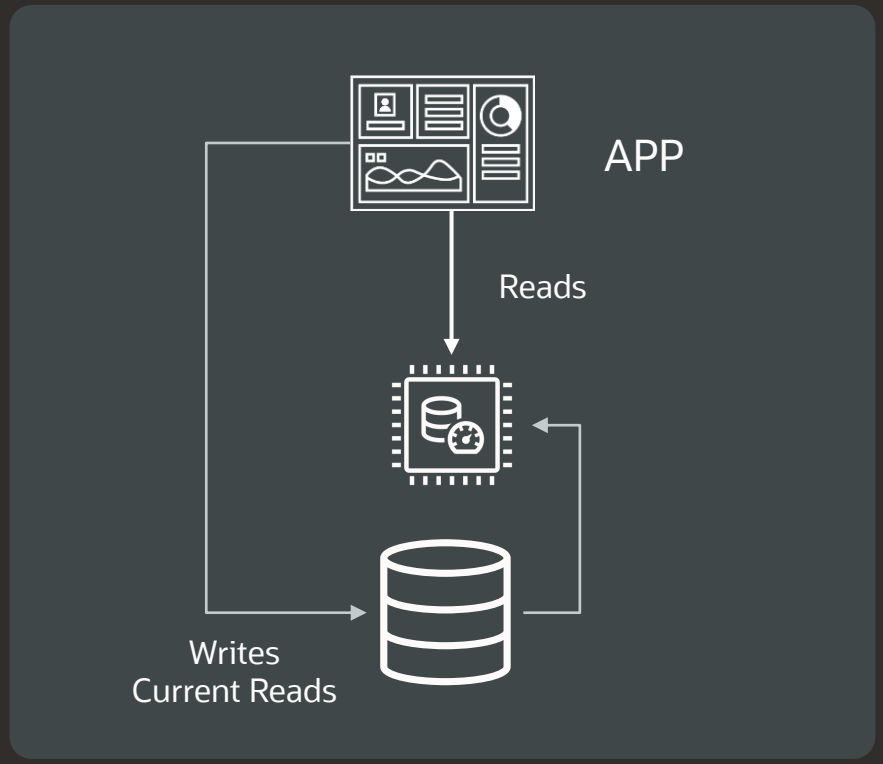
What is Oracle True Cache ?

In-memory, consistent, and automatically refreshed SQL and object cache deployed in front of an Oracle database

- It handles all read SQLs like Oracle database
- Conceptually it is like a diskless Active Data Guard standby

Data in True Cache is always current, and consistent across multiple tables and objects

Schema changes in the database are automatically propagated to the cache

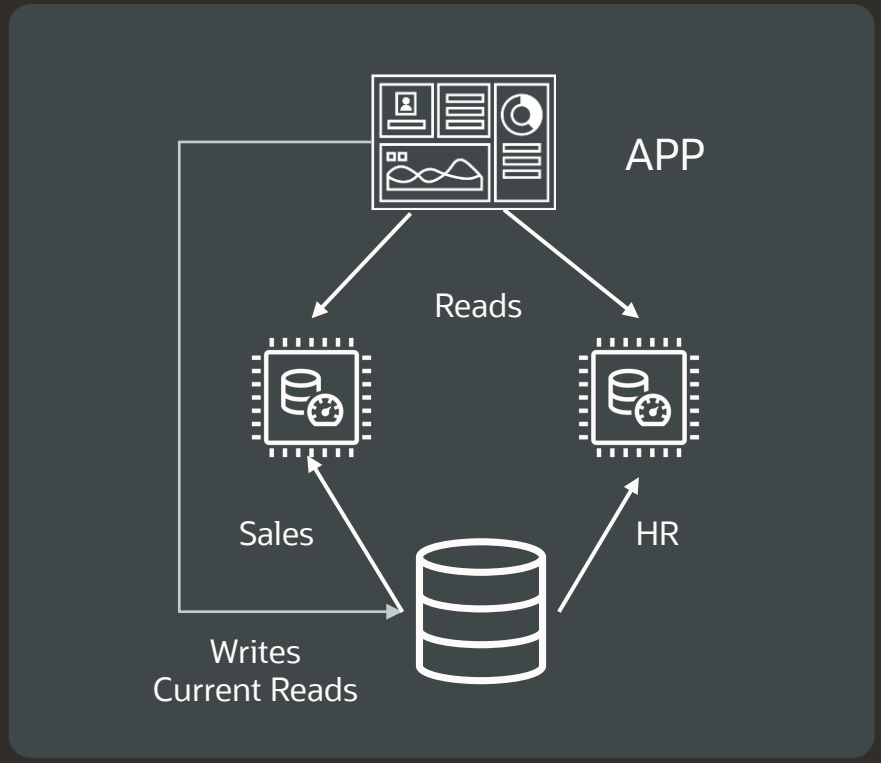


Scaling True Cache with Partitioned Configuration

Multiple True Cache instances can be deployed in front of a database for availability, scalability, and caching larger data volume

True Cache also supports spooling data to disk which doesn't fit in the memory

- This allows us to cache much more data

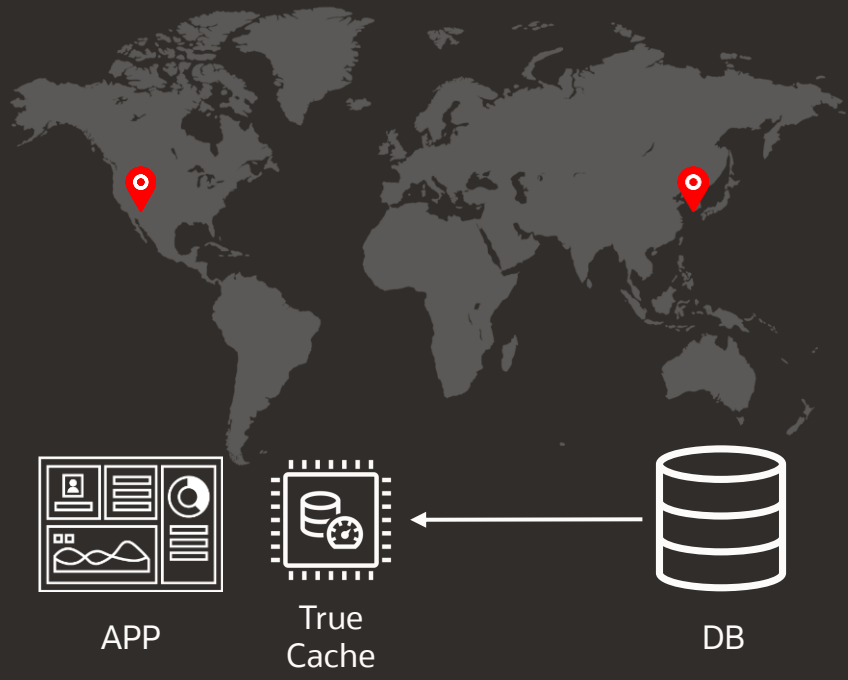


Global Data Proximity to Application

For use cases like data residency, user proximity, and device proximity, applications are deployed far from the database

In such cases, True Cache can be deployed near the application to boost the read performance significantly

True Cache does not store any data on the disk by default; hence, it is a data-processing tier that Data Residency regulations allow



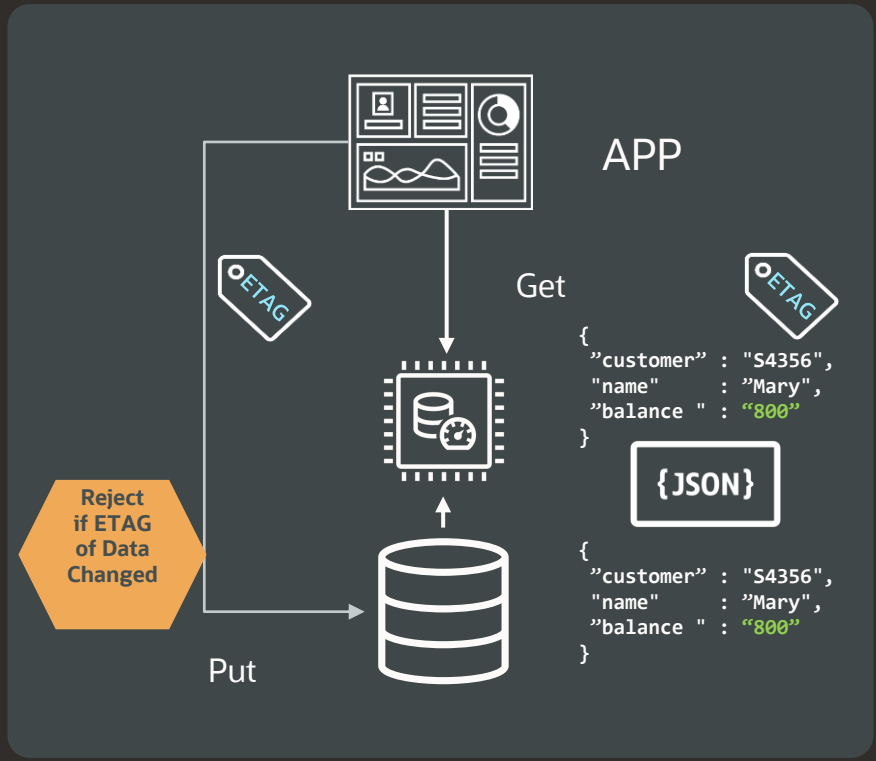
True “Document and Relational” Cache with Lock Free Concurrency

JSON documents stored in the database can be cached in True Cache

- Documents are automatically refreshed at attribute level granularity hence reducing the I/O

Concurrency is automatically maintained with versioning

- When a document is read from True Cache, a value-based ETAG is embedded in it. When the same document is updated in the database, the database verifies the ETAG in the document to ensure its recency
- If ETAGs don’t match, an error is reported



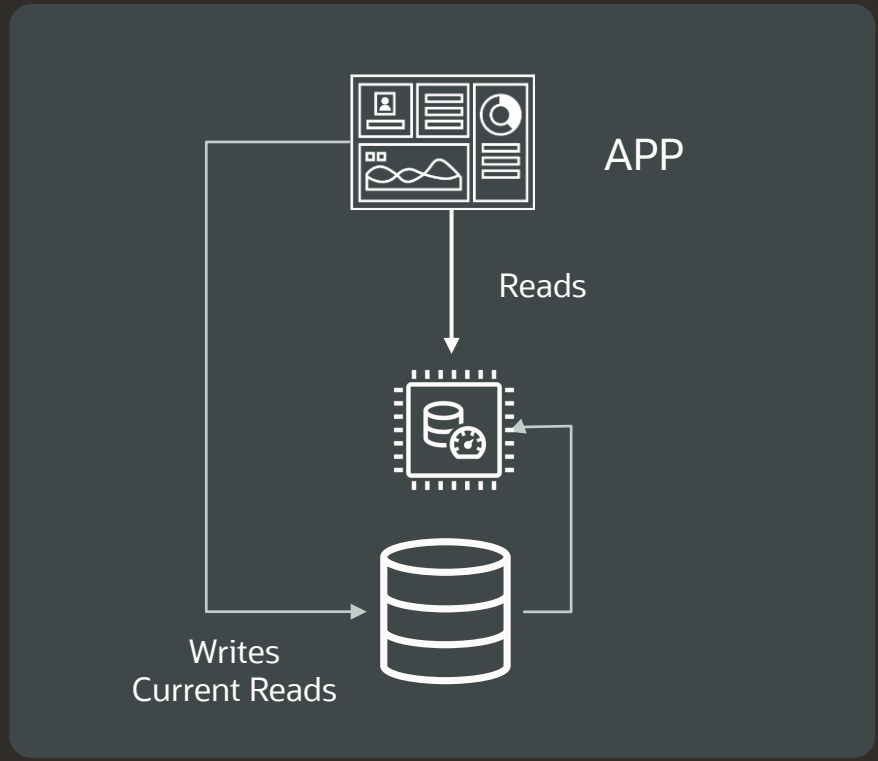
True Cache mitigates deficiencies of conventional caches

Operation	Conventional Caches	True Cache
Loading the cache	Developer responsibility	Automated
Cache consistency with DB	Developer responsibility	Automated
Cache consistency with values in the same cache	Developer responsibility	Automated
Cache consistency with other caches	Developer responsibility	Automated
Complex data type support	Developer responsibility	Automated
Full JSON support	Developer responsibility	Automated
Comprehensive security	Developer responsibility	In-Built
Parallel processing	Developer responsibility	In-Built
High Availability	Developer responsibility	In-Built

How To Use True Cache

The application can query data from True Cache in the following two modes

1. The application maintains two connections:
   - A read-only connection to True Cache and a read-write connection to the database and uses the read-only connection for offloading queries to True Cache
2. The application maintains one connection.
   - The JDBC driver maintains two connections internally and does the read-write split under application control
   - The application uses setReadOnly() calls to mark some sections of code as “read-only”
   - This is an existing JDBC API which MySQL already uses for similar purposes
   - JDBC driver will send read-only queries to True Cache and writes and DDLs to the primary



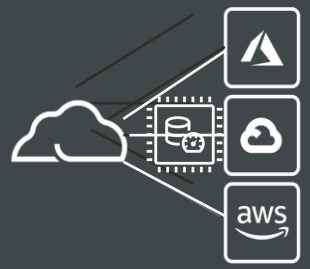
True Cache Use Cases

True Cache can be deployed as a:

Mid-Tier Cache

Edge Cache

Cross-Region Cache



Cross-Cloud Cache

Benefit To The Business

Cost Savings

- Improve application performance without rewriting the applications
- No investment in additional caching products and skills
- Single cache for different data types and formats (document, relational, row, columnar)
- Offload caching to commodity hardware from the database (which might be on more expensive hardware)

Stronger security, which comes with Oracle database

Leverage the full power of Oracle Database

Customer Feedback

Stock Exchange – Offload read queries for stock tickers to True Cache with 10 TB of cached data for a 200 TB database.

- 6 True Cache instances (for load balancing) are being deployed, which would have required a 200-node Redis cluster

Mobile Phone Manufacture – Offload read-only queries to True Cache as the primary is maxed out with vertical scaling

Financial Institution – Offload fraud detection application's AI Model Inferencing to True Cache as models are trained once a day, and most fraud detection queries are read-only.

Marketing – Would like to use True Cache for real-time marketing campaigns. Data in conventional caches is stale and does not allow for real-time tweaking of marketing campaigns.

Oracle Fusion Applications – Cache for Business Object layer for objects that do not change as often

Oracle Banking Apps – Testing to offload Reads to Cache for the core banking application, which is a read-intensive application

Example Performance Test Results

Linear Throughput Improvement by adding low-cost True Cache Nodes

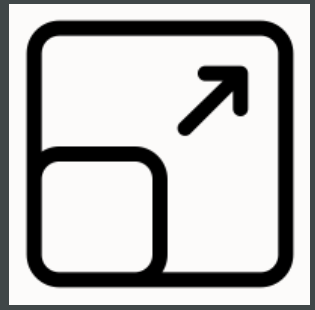
Workload: 84% selects, 16% inserts/updates OLTP JDBC workload modeling online shopping and e-commerce

The primary and each True Cache instance all have 48 cores

Primary SGA: 236G; each True Cache instance SGA: 128G

True Cache nodes are cheaper: diskless, less memory, can be commodity hardware

Close to linear speedup



Linear Throughput Scalability

# of True Cache Instances	Throughput Improvement
1	1.8x
2	2.6x

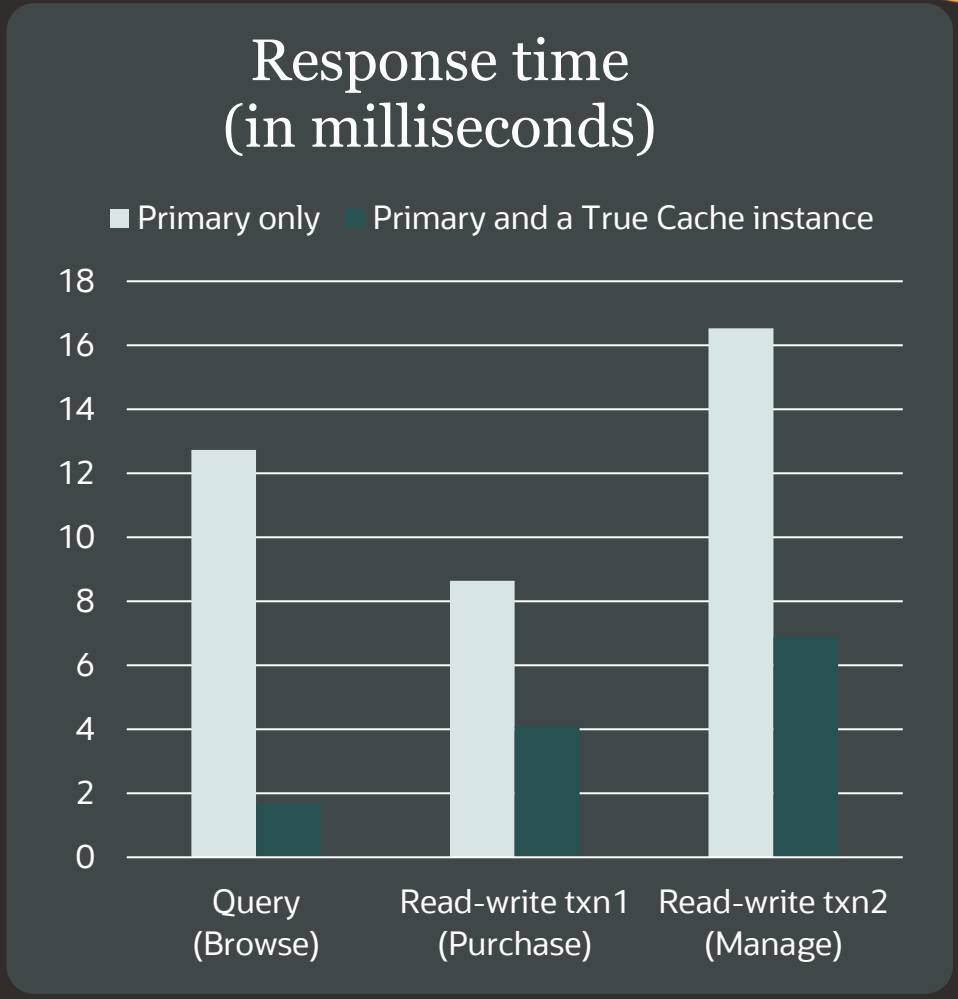
Response Time Improvement by adding one True Cache instance

Same OLTP JDBC workload

Primary only: long response time due to queuing effect when CPU is maxed out

Adding a True Cache instance reduces the queuing effect because the same workload is distributed to two instances

Adding a True Cache instance can reduce response time significantly for the same workload



10X Throughput improvement with bigger aggregate cache

With partitioned configuration, the aggregate cache size of all True Cache instances can be much larger than that of the database

Simulation:

- Database: 10G SGA
- True Cache instance: 100G SGA
- Database size: 116G
- Workload: based on Swingbench

10X

True Cache speed up for cache hit Indexed OLTP query

Response time improvement due to cache proximity

Primary: Mumbai, Application: Phoenix, SQL roundtrip: 600 ms due to distance

Adding a True Cache instance in Phoenix reduces SQL round trip to 2 ms for cache hit queries, a 300x speedup

One can eliminate cache miss for tables that fit in memory:

- Allocate a keep buffer pool on True Cache
- Assign the table to keep the pool on the primary
- Perform an initial table scan on True Cache

300X

True Cache speed up for cache hit for OLTP query

Scenario	Response Time
Primary only	600 ms
Primary + True cache, cache hits only	2 ms

Key Takeaways

Oracle True Cache

- Most fully-featured SQL and Object Cache
- Automated Data Refresh
- Full Oracle Database Capabilities
- Supports leading-edge AI such as Vector Search

A True Cache with less effort

Why settle for less?

For More Information

oracle.com/database/truecache