ORACLE

Caching Oracle Data for Performance: an eBay perspective

Doug Hood
Product Manager
Oracle

Santosh Karnik
Director Data Infrastructure
eBay

Bin Zhang
Lead TimesTen DBA
eBay

Tim Robison
Lead Data Access Layer Developer
eBay

Contents

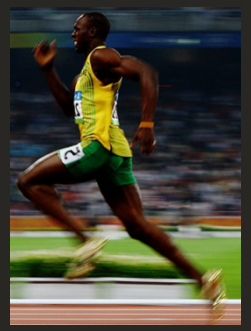
- Caching Concepts
- Caching with Oracle TimesTen
- How eBay uses TimesTen for Caching

Safe Harbor

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

Characteristics of Performance



Latency

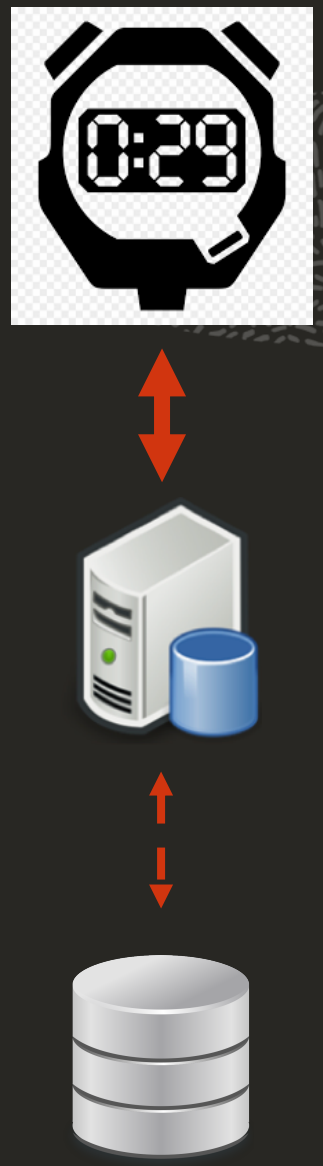
Throughput

Scalability

Why Cache Data?

Caching can enable

- Lower Latency
- Higher Throughput
- Scalability
- Database Offload
- High Availability

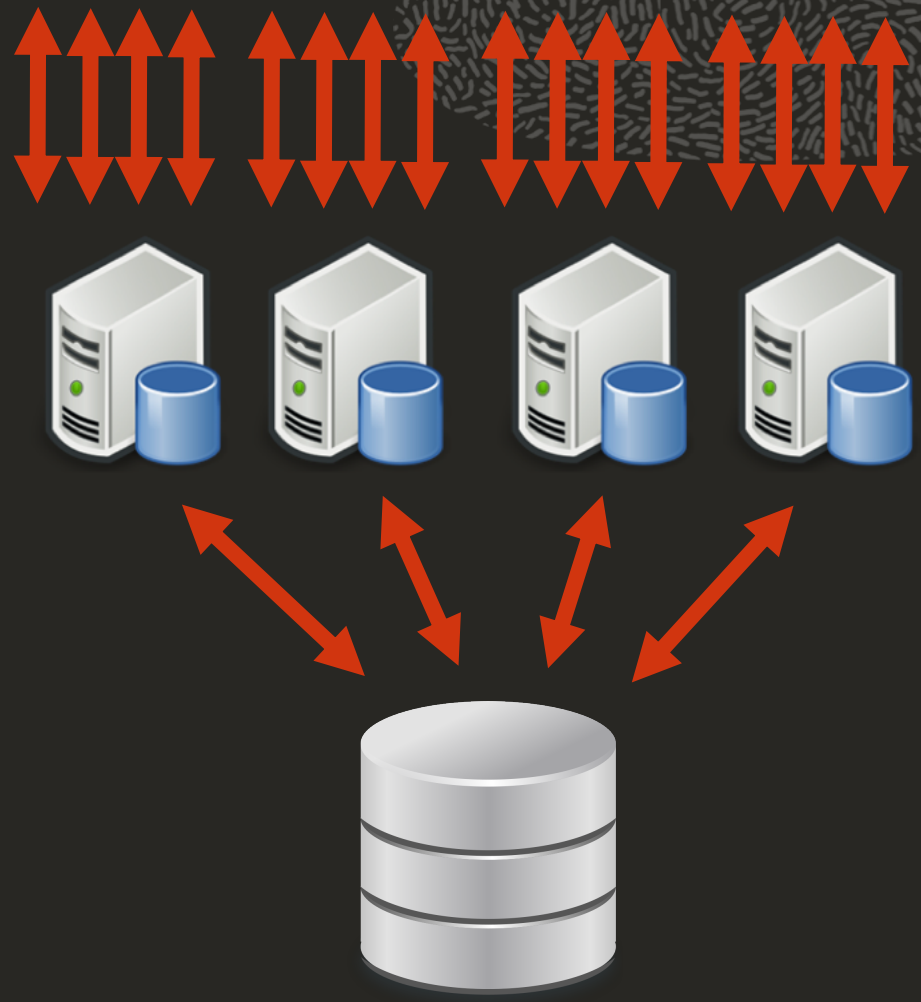


Caching can enable

- Lower Latency
- Higher Throughput
- Scalability
- Database Offload
- High Availability

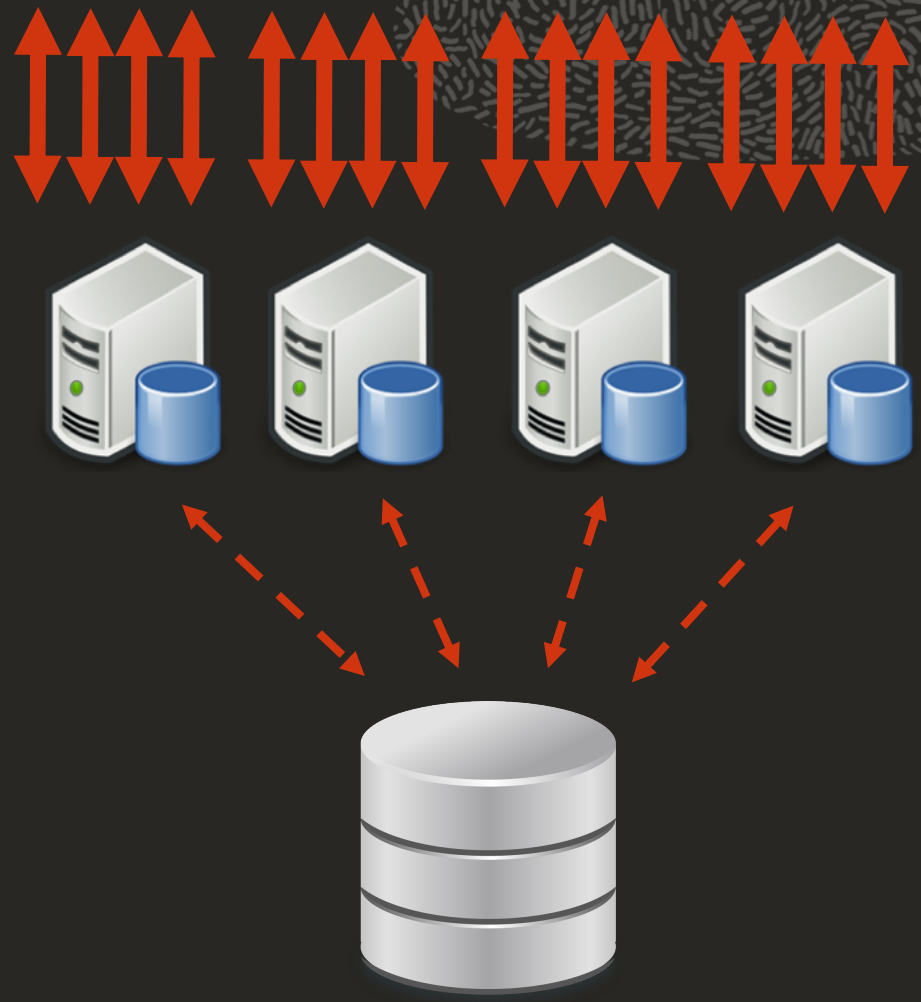
Caching can enable

- Lower Latency
- Higher Throughput
- Scalability
- Database Offload
- High Availability



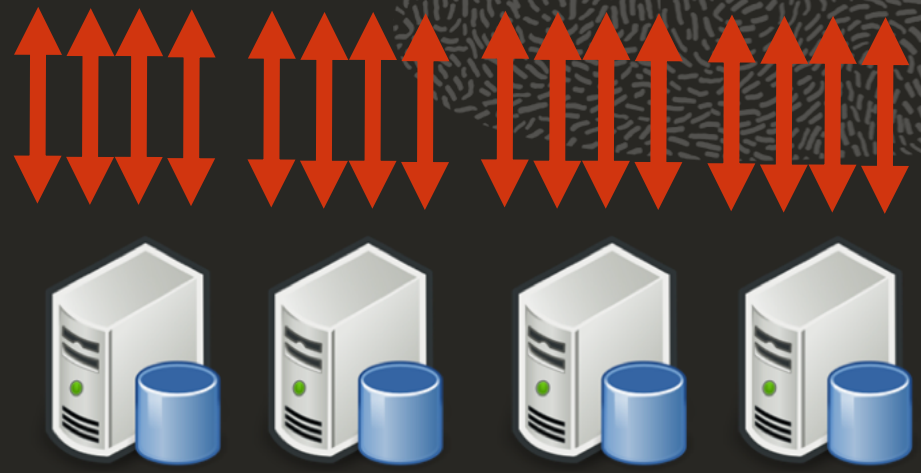
Caching can enable

- Lower Latency
- Higher Throughput
- Scalability
- Database Offload
- High Availability

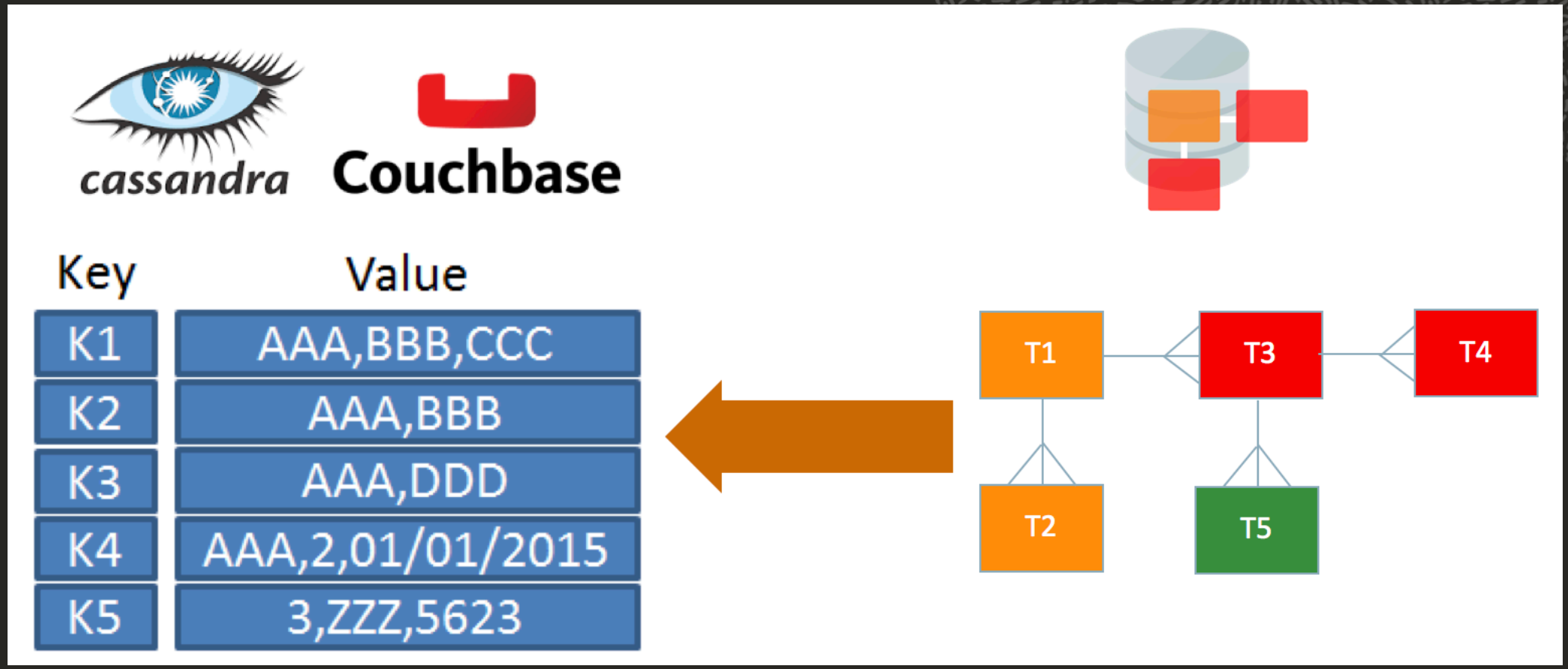


Caching can enable

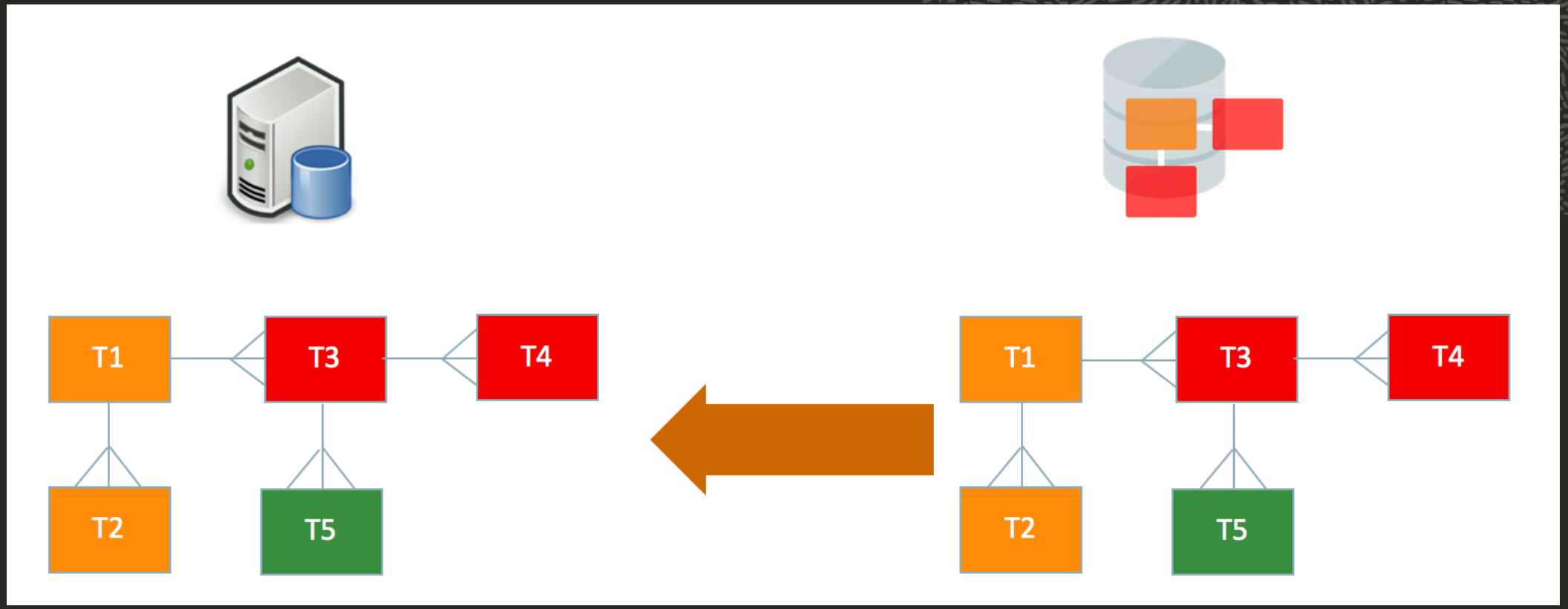
- Lower Latency
- Higher Throughput
- Scalability
- Database Offload
- High Availability



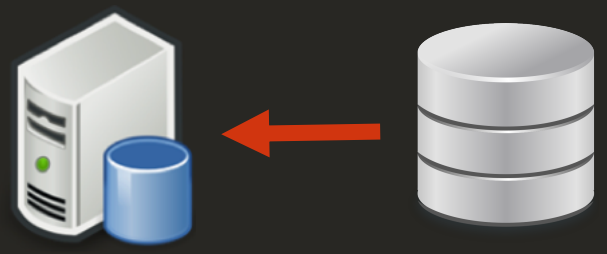
ResultSet Caching with a NoSQL DB



SQL Table Data Caching



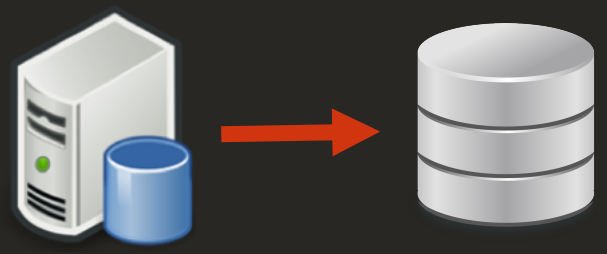
Read Caching



Cache Consistency

- How do you do the initial cache load
- What about automatically refreshing the cache
  - Committed Inserts, Updates & Deletes
  - How often do you refresh the cache
- How do you un-load and re-load the cache
- Can or should you load missing data on demand
  - What about dependent data
- Is the cache data persistent
- Is the cache data highly available

Write Caching



Cache Consistency

- How do you do the initial cache load
- What about automatically writing the data
  - Committed Inserts, Updates & Deletes
  - Sync or async writes
- How do you un-load and re-load the cache
- Can or should you load missing data on demand
  - What about dependent data
- Is the cache data persistent
- Is the cache data highly available

Metadata Driven Caching

- Should you use custom code or SQL DDL to define your caches?

Metadata Driven Caching

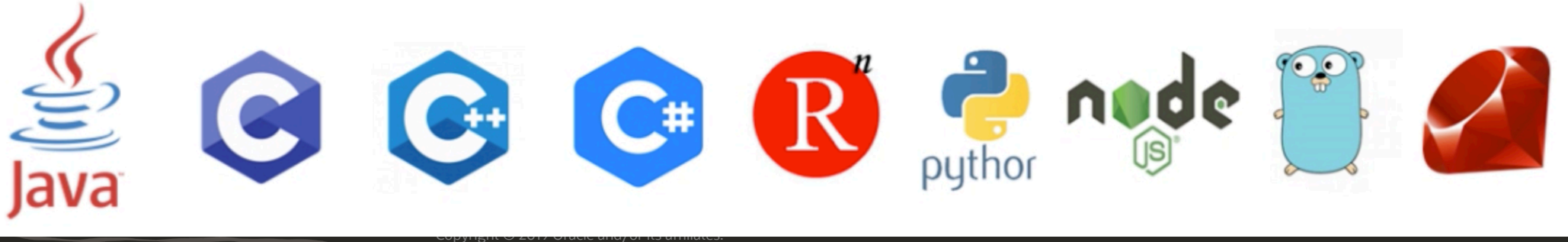
```
CREATE dynamic READONLY CACHE GROUP customers
  autorefresh mode incremental interval 1 millisecond
FROM foo.customer (
  cust_num NUMBER(12) NOT NULL,
  region VARCHAR2(10),
  dob DATE,
  PRIMARY KEY(cust_num)
);
```

Metadata Driven Caching

```
CREATE asynchronous WRITETHROUGH CACHE GROUP orders
FROM foo.order (
  ord_num NUMBER(12) NOT NULL,
  cust_num NUMBER(12) NOT NULL,
  description VARCHAR2(200),
  when_placed DATE,
  when_shipped DATE,
  PRIMARY KEY(cust_num)
);
```

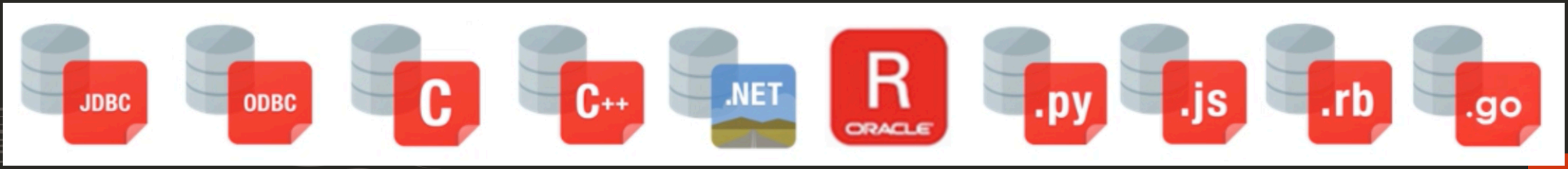
What is TimesTen

- TimesTen is an In-Memory RDBMS
- It is very fast
- It supports SQL and PLSQL
- It supports familiar SQL APIs
- It supports ACID transactions with persistent storage
- It runs on many platforms and in the Cloud

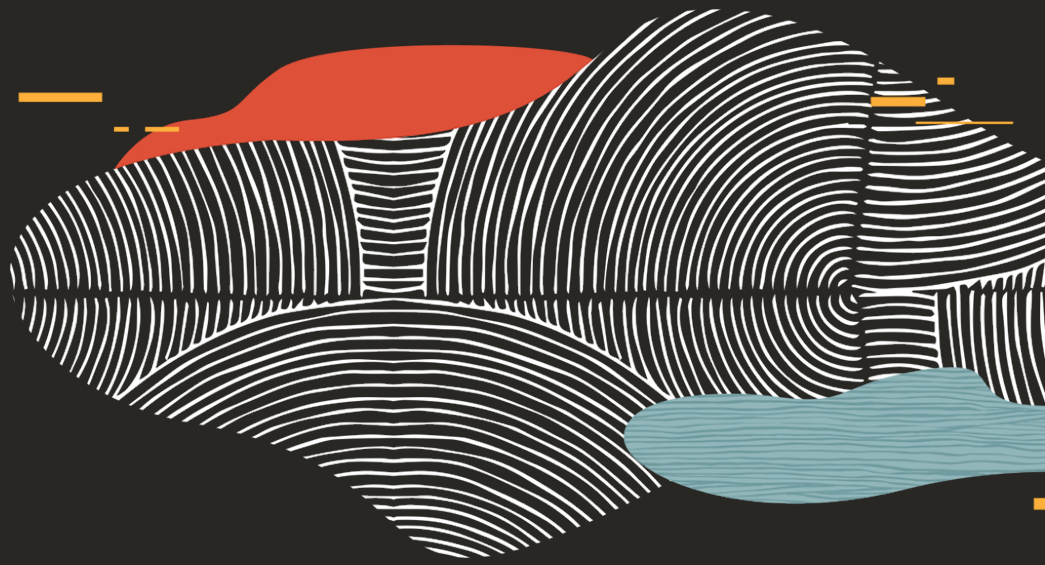


What is TimesTen

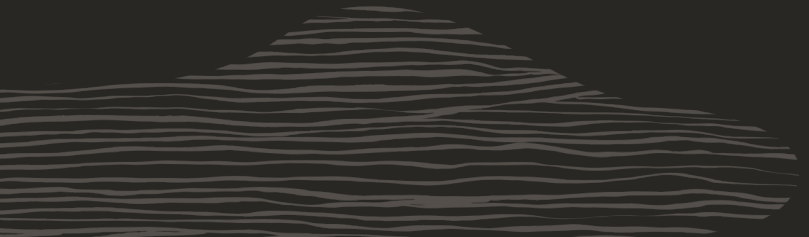
- TimesTen is an In-Memory RDBMS
- It is very fast
- It supports SQL and PLSQL
- It supports familiar SQL APIs
- It supports ACID transactions with persistent storage
- It runs on many platforms and in the Cloud



What is TimesTen

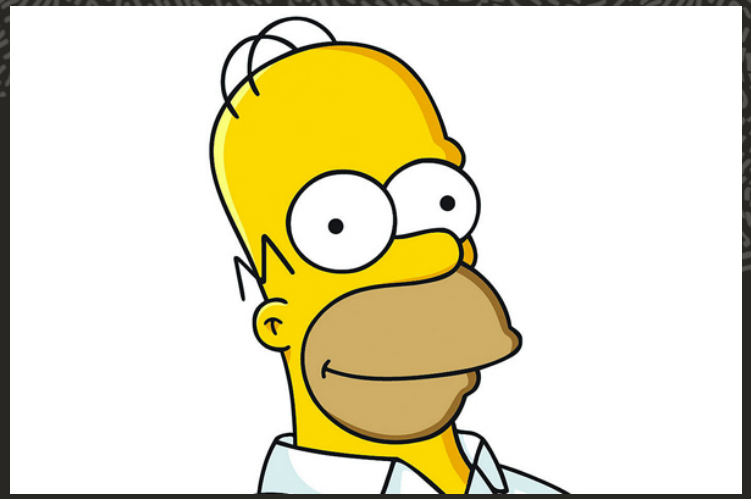


- TimesTen is an In-Memory RDBMS
- It is very fast
- It supports SQL and PLSQL
- It supports familiar SQL APIs
- It supports ACID transactions with persistent storage
- It runs on many platforms and in the Cloud

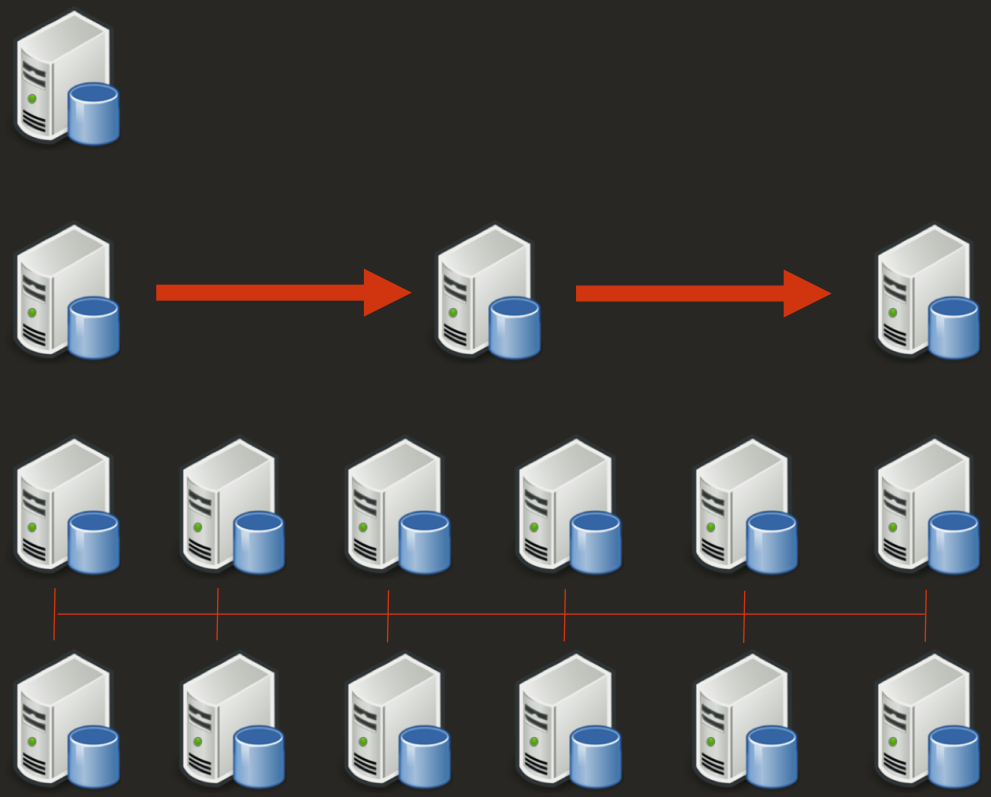


TimesTen Cache Features

- Read + Write Caching
- SQL Table Caching (can join cache tables)
- Metadata Driven Caching
- Automatic Cache synchronization
- High Availability (replication + Active DataGuard)



TimesTen Deployment Options



Standalone Database

Replication (sync/async, active/standby)

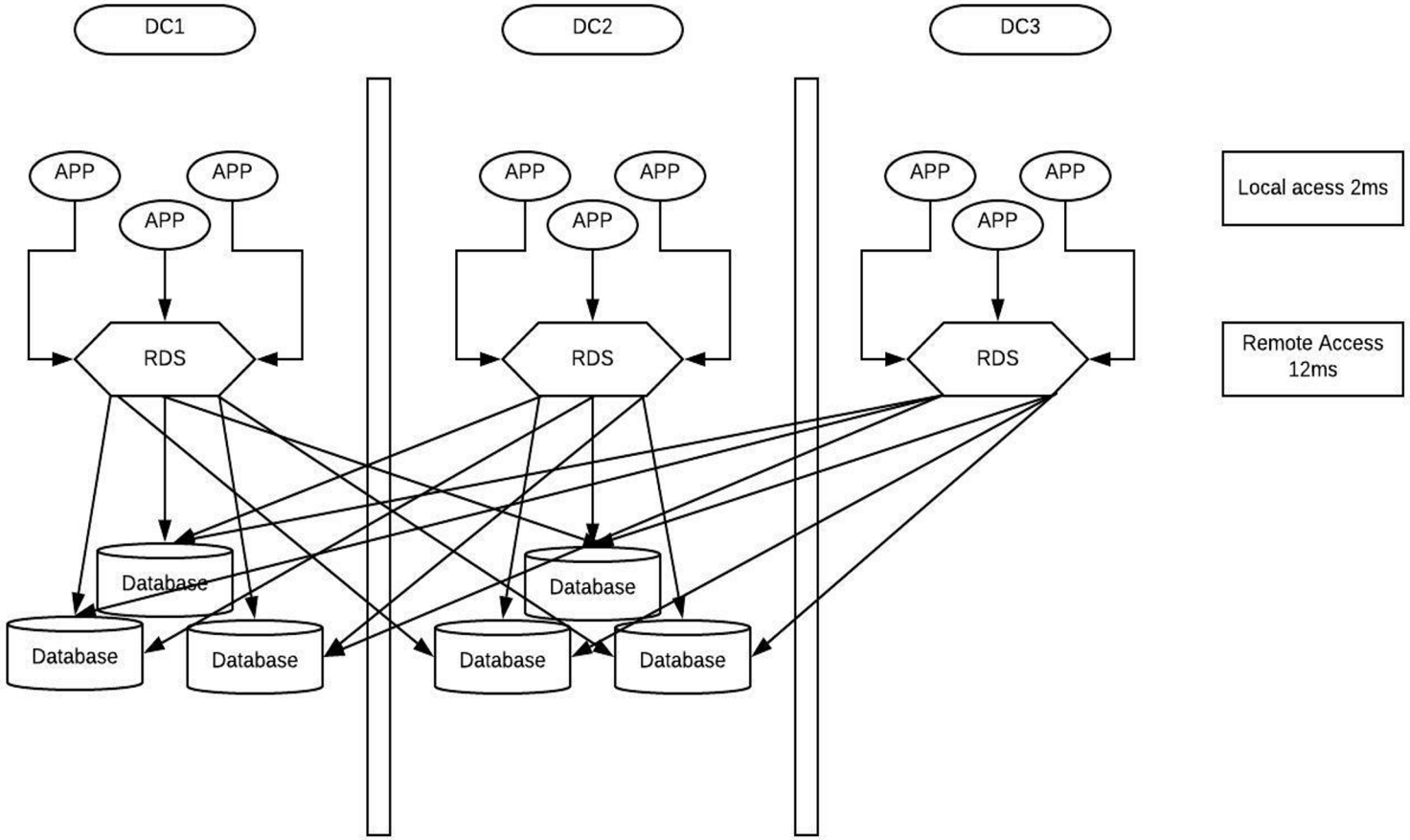
Scaleout (active/active)

Data Frameworks & Services TimesTen End-To-End

Santosh Karnik and Tim Robison

App/DB Architecture & Current Challenges

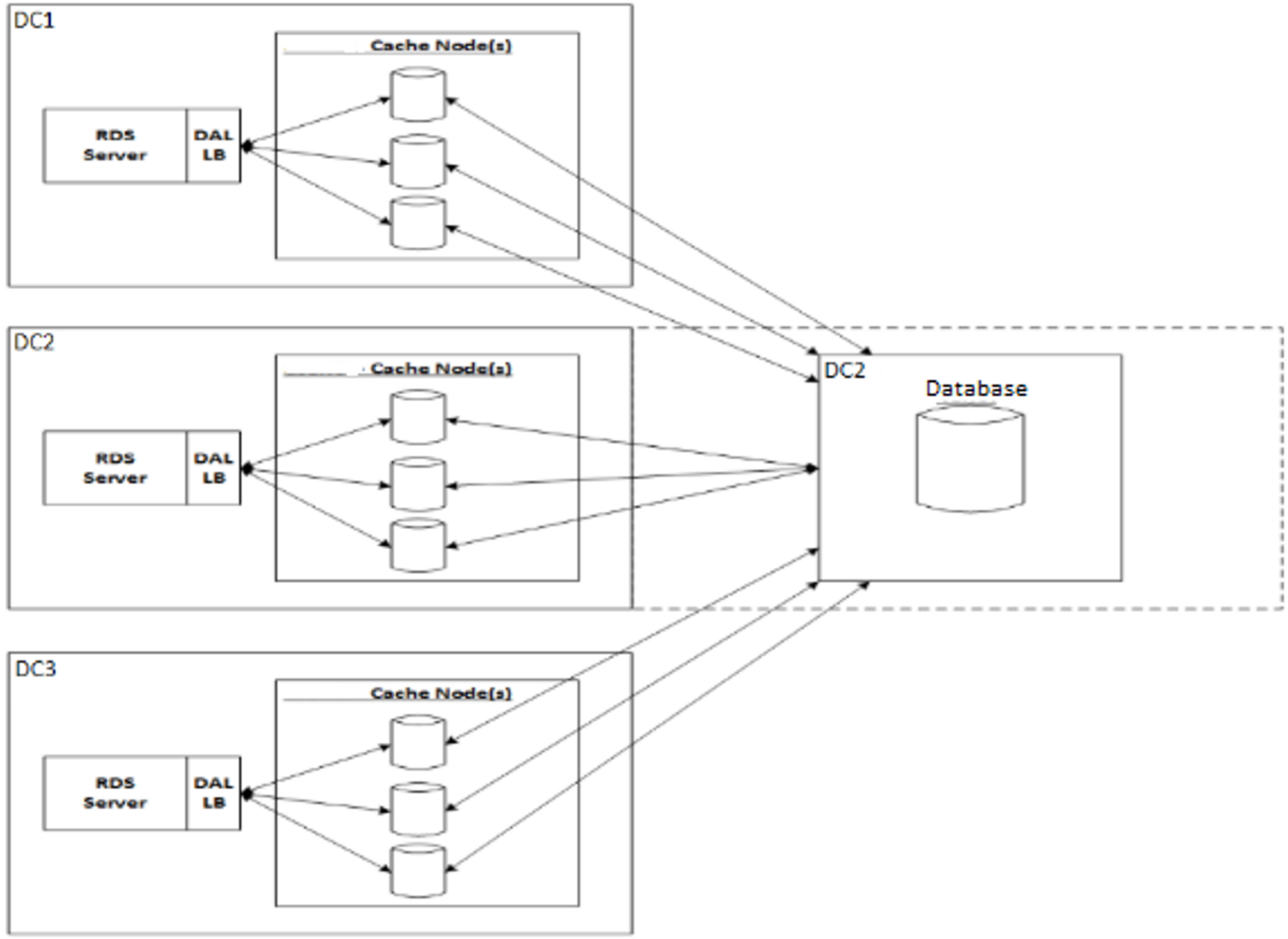
Multi DC Deployment for DB nodes and Applications
50% Local calls with Avg Latency 3ms
50% Remote calls with Avg latency 12ms
180B calls/day workload across the DB fmly
CBC latch contention halting all operations causing business impact
Oracle overloading SAN with high IOPS



System Overview

High Level System Diagram

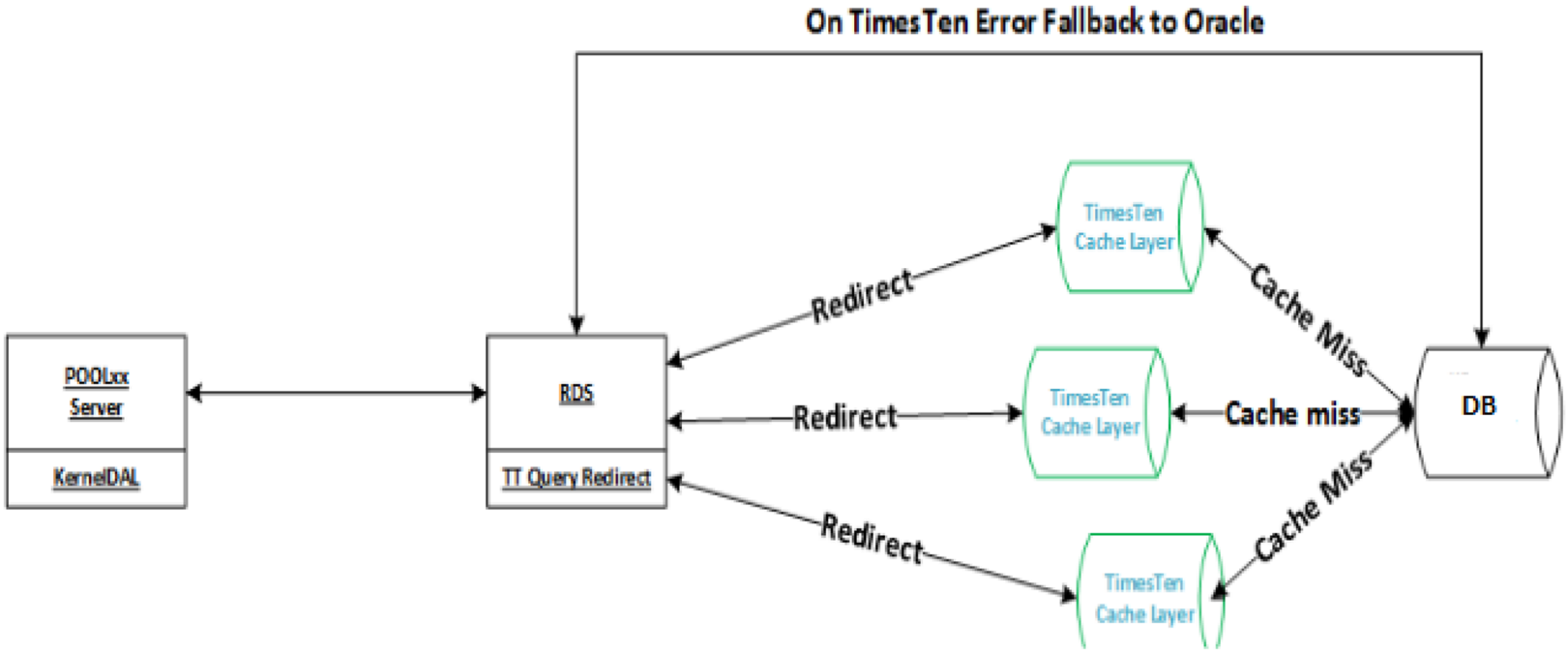
- RDS(Relational Data Service) uses DAL(Data Access layer) Load Balancer and the RDS Data Source Redirect Feature to route SQL requests to the Oracle caching layer.
- A TimesTen Cluster for each User database will be deployed to each Data Center providing Application Users with local reads and low latency.



System Overview

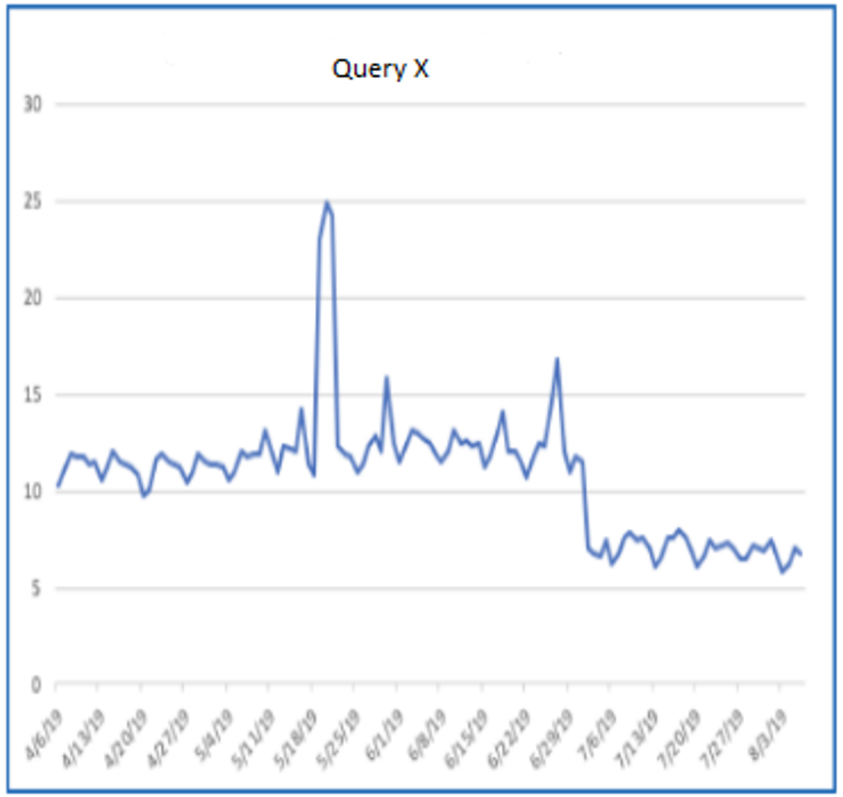
RDS Caching Layer - Data Center Diagram

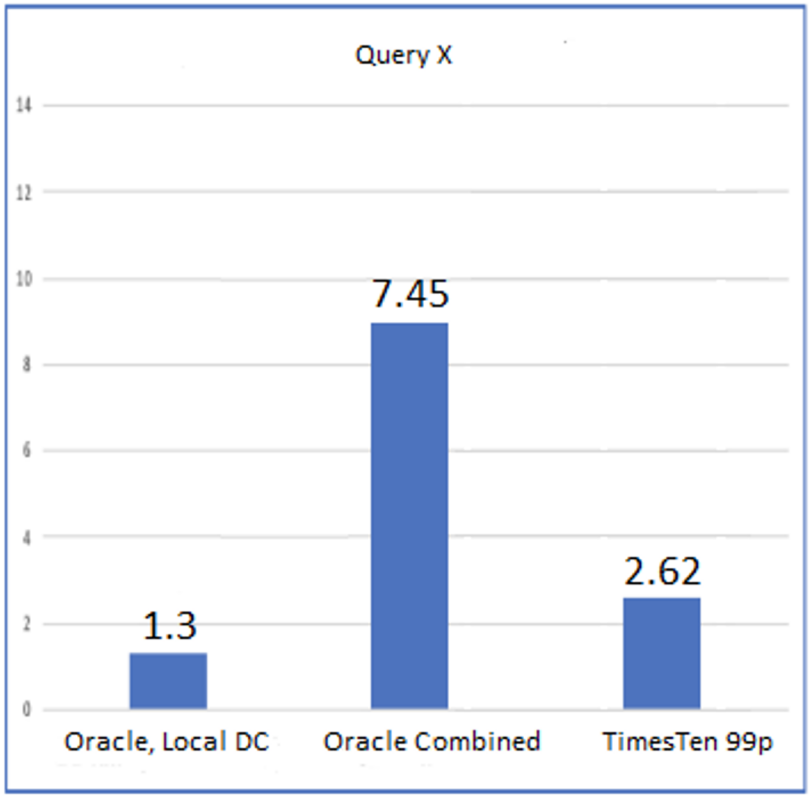
- RDS Redirect Feature has been designed to fail back to using the primary Oracle data source if any exception/error occurs during a SQL read request.
- If a TimesTen data source cluster is marked-down the SQL redirect operation will be ignored and the request sent directly to Oracle.
    - If at least one TimesTen data source within the DAL load balancer is enabled all traffic will be routed to it until either all data sources are marked down or the down data sources are marked back up.



Oracle TimesTen Caching Benefits

Reduces DB Hits
Avoids DB Read/Write Contentions
Reduces Sessions
Increases Speed (All Local Reads)
Current User Cache Capable of 140 B calls
No Cache invalidation pipeline





TimesTen Best Practices for eBay

- Enable huge page in linux ---- It can double the throughput and reduce server process memory usage
- Disable autocommit in jdbc driver ---- It can reduce number of network round trip and improve TimesTen stability
- Warm-up before enable traffic ---- load pre-defined user id info into cache, app can have 99.9%+ high cache hit ratio
- Place data file and log file into separate disk volume ---- avoid disk contention between CKPT and LGWR (by cache group refresh/dynamic read)
- Limit CKPT IO rate if only one volume ---- if LGWR IO is slow, it would delay cache group refresh and bring lock contention
- Reduce LRU aging activity ---- Aging would bring log buffer spike and CKPT blocks written spike. Choose LOWUSAGETHRESHOLD and HIGHUSAGETHRESHOLD based on IO capacity
- Enable CacheCommitDurable from Timesten 11.2.2.8.33 -- improve to dynamic load performance by making autorefresh txns non-durable. The TT6003 errors at sys.cache_groups can be eliminated by this
- Different cache refresh interval for different cache group
- Monitor long transaction on Oracle base table, it's blocking writes on base tables for very short time when creating dynamic cache group
- 2k connections max-limit in 11.2. Looking forward to 18c

ebay

Future Steps

LOB caching

Improve latencies with caching heavy accessed LOBs

Write Thru Caching

Built a true fault-tolerant database layer

TimesTen Using Persistent Memory

Looking forward to TimesTen Scaleout

ebay