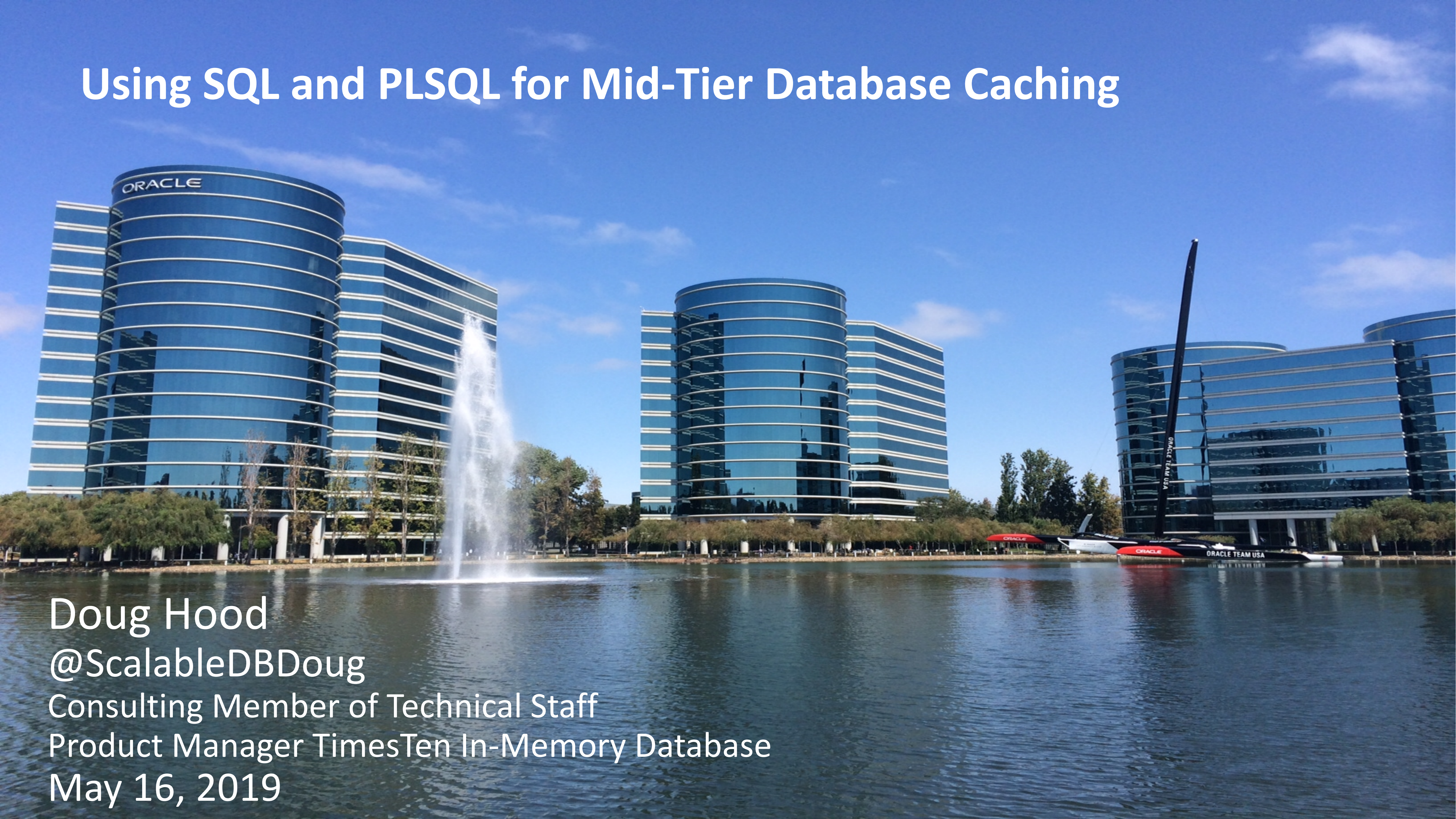
Using SQL and PLSQL for Mid-Tier Database Caching



Doug Hood

@ScalableDBDoug

Consulting Member of Technical Staff

Product Manager TimesTen In-Memory Database

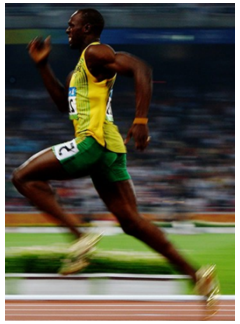
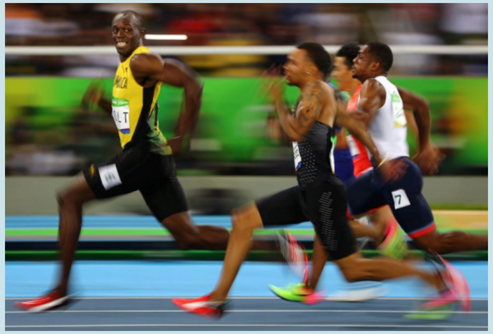
May 16, 2019

Agenda

1. Business Problem
2. DevOps challenge
3. Architectures and technical challenges
4. How to do it
5. Demo, summary and Q & A

ORACLE

Latency, Throughput and Scalability

Latency		**How quickly can one operation complete** **One sprinter in 9.58 seconds** **~ 40 km/h for 100M [2009]**
Throughput		**How quickly can many operations complete** **Ten sprinters in under 11 seconds** **~ 40 km/h for 100m [2009]**
Scalability		**By adding more resources can throughput keep increasing** **33 cars on 2.5 mile oval track** **~250 km/h for 804 km [Indy 500, 2017]**

Lower Latency with TimesTen Cache

Query	Oracle	Cache
Q1	43	3
Q2	69	6
Q3	105	8
Q4	121	20
Q5	140	18
Q6	163	19
Q7	231	18

Oracle 11.2.0.4 RAC
RAC nodes were Oracle Sun X7-2L
NVMe Storage
Over 50 Million Users

Application Tier Database Cache (TimesTen)
Ran on the same nodes as the production RAC
5 table joins for 100s of millions of rows of data

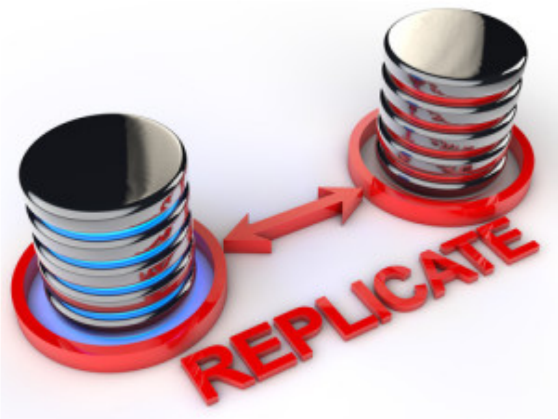
Latency is in Micro Seconds …

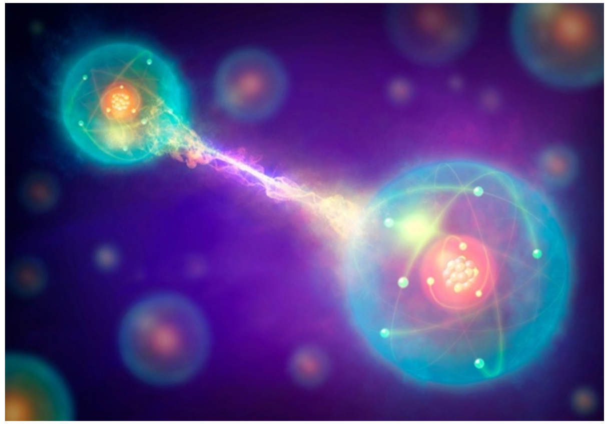
Why Cache Data?

Why Cache Data?

- To get lower latency for SQL statements
- To get more throughput
- To get more scalability

Caching Challenges





The Business Problem

- DB apps, make them faster and cheaper
- Do more with less

The DevOps Challenge

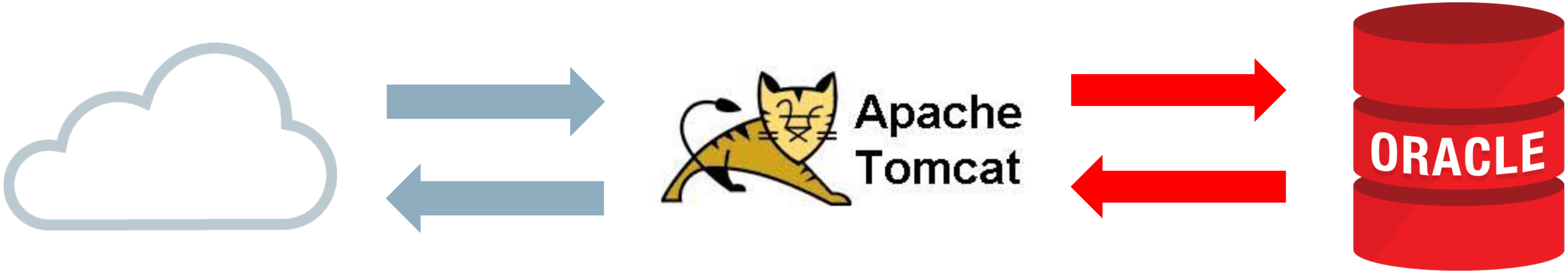
- Make it boring, no surprises
- Standardized, run in VMs or containers
- **Everything** is **automated**
- Figure out everything that can go wrong and **be well behaved**

docker

Xen™

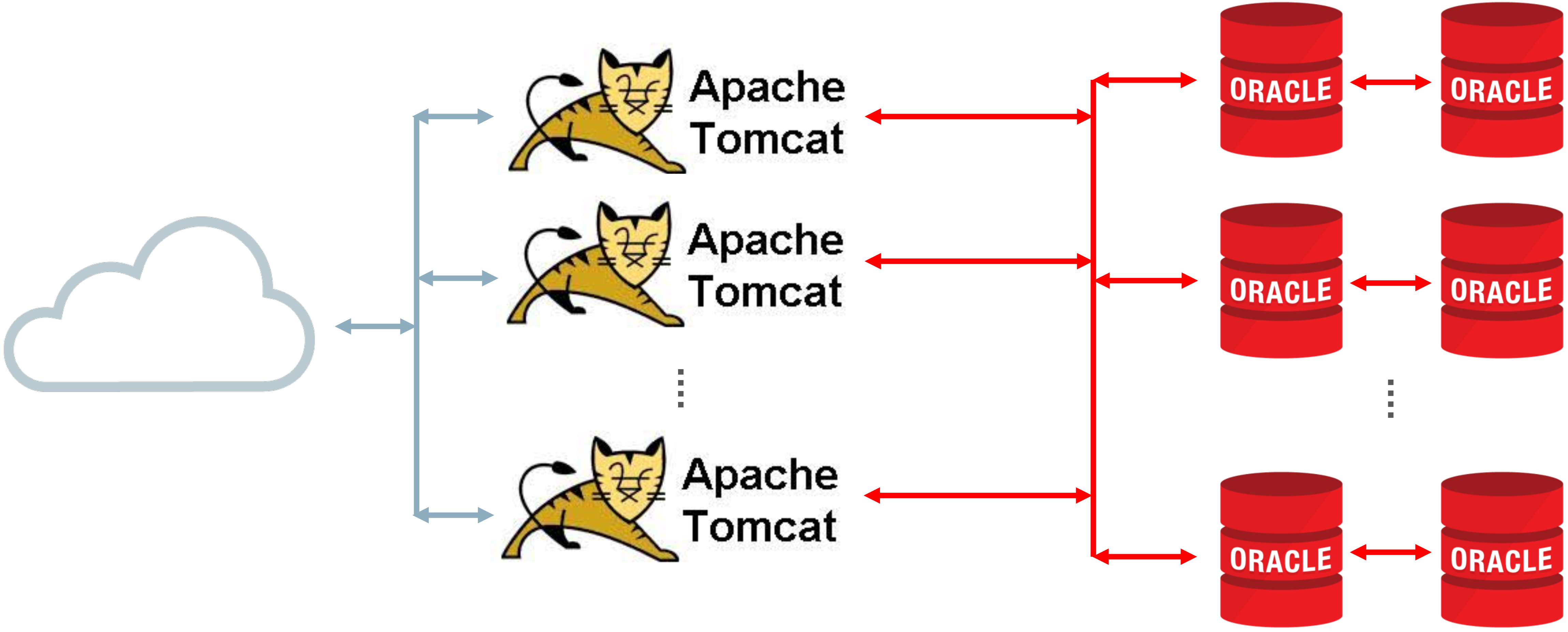
Linux KVM

Architectures and technical challenges



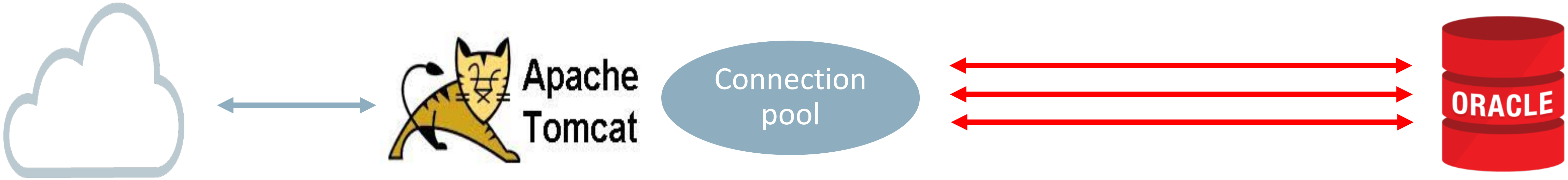
- How many users? How many concurrent users?
- How many concurrent Oracle DB connections?
- How many concurrent Application Server connections?
- What about HA? What about DR?
- What about latency?

Architectures and technical challenges



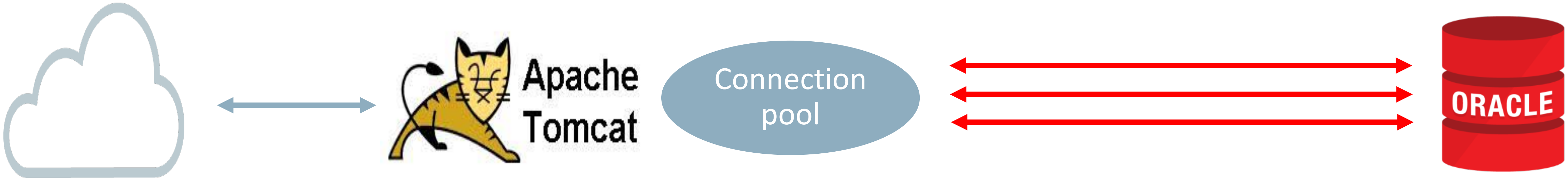
Can we do better?

Architectures and technical challenges



Can we do better?

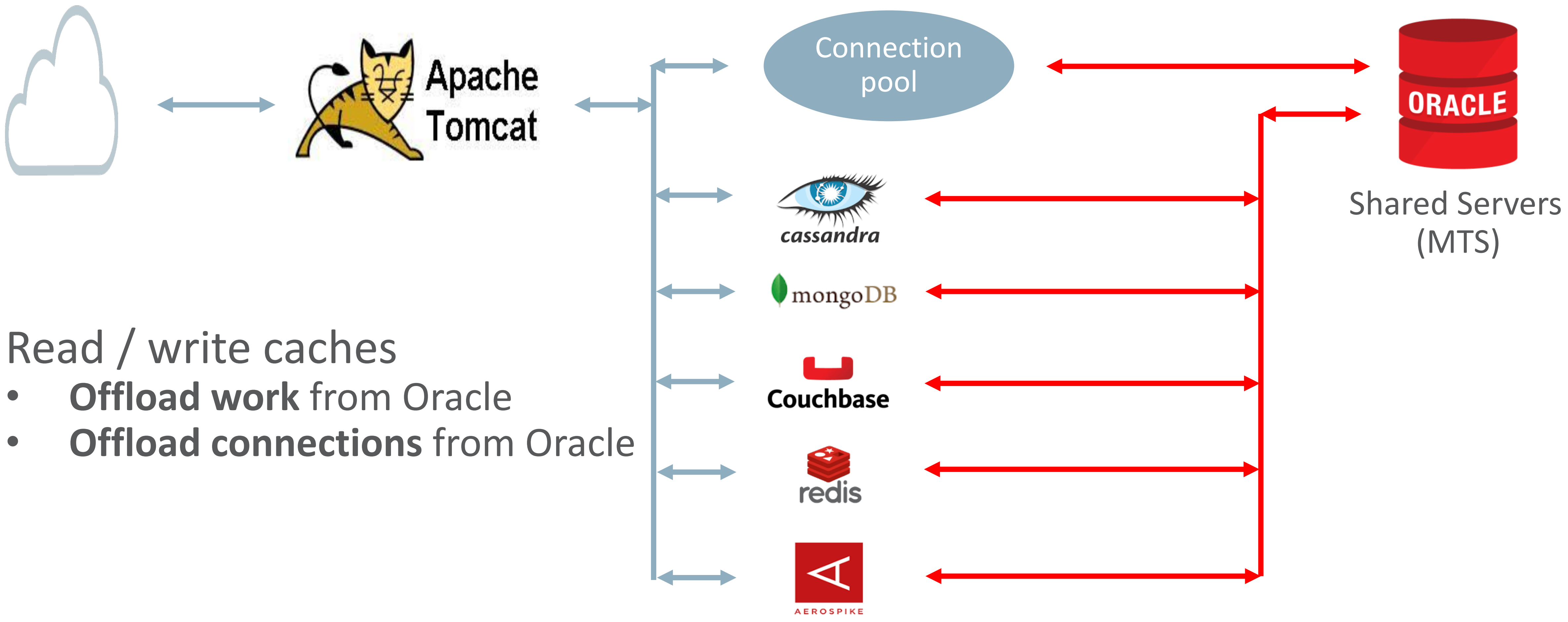
Architectures and technical challenges



Shared Servers (MTS)

Can we do better?

Architectures and technical challenges



Read / write caches

- **Offload work** from Oracle
- **Offload connections** from Oracle

Can we do better?

Architectures and technical challenges

- **Go faster** than NoSQL solutions & **minimize** the **hardware cost**
  - Lower latency at the **99th percentile**

cassandra

mongoDB

Couchbase

redis

VOLTDB

AEROSPIKE

ORACLE

Architectures and technical challenges

- **Go faster** than NoSQL solutions & **minimize** the **hardware cost**
  - Lower latency at the **99th percentile**

cassandra

mongoDB

Couchbase

redis

VOLTDB

AEROSPIKE

- Simple and scalable is good,
  but how many machines do you need?
- What about the 95th and 99th percentile for latency?
- What about cache misses?
- How is the cache kept up to date?

99th percentile latency can be as high as 324 ms …

100th percentile latency is many seconds, timeout > 100 ms

YCSB Workload B (95% read, 5% write)

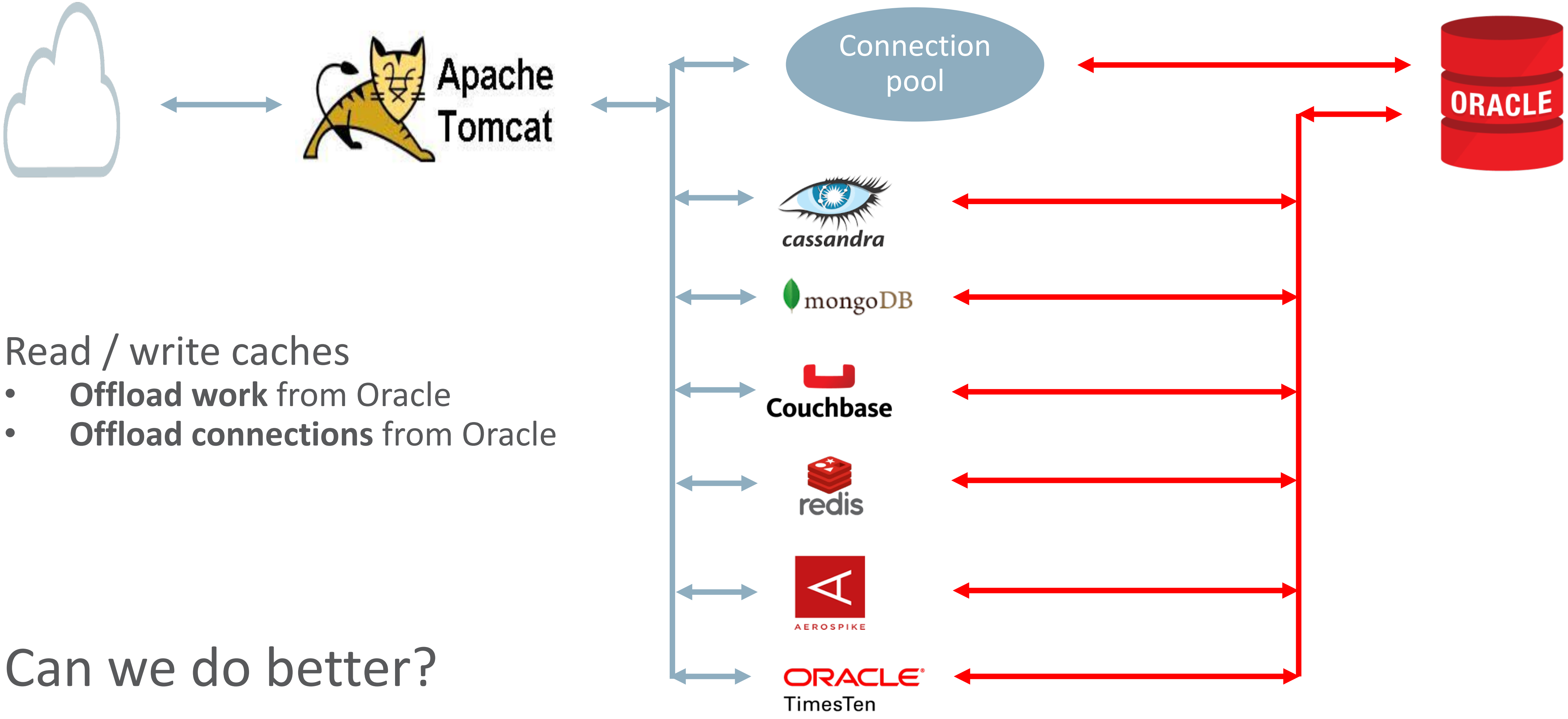
Database	TPS	Nodes
Cassandra	221K	32
MongoDB	260K	2
CouchBase	454K	9
Redis	1M	3
VoltDB	1.5M	6
AeroSpike	1.6M	1

YCSB Workload B (95% read, 5% write)

Database	TPS	Nodes
Cassandra	221K	32
MongoDB	260K	2
CouchBase	454K	9
Redis	1M	3
VoltDB	1.5M	6
AeroSpike	1.6M	1
TimesTen	2.8M	1

← Lets talk

Architectures and technical challenges



Read / write caches

- **Offload work** from Oracle
- **Offload connections** from Oracle

Can we do better?

Customer measured TimesTen Cache latency

- 97% cache hit ratio
  - 3% of the time the data was not in the cache, so needed a round trip to Oracle
- 99th percentile latency = 1ms

Customer measured TimesTen Cache latency

- 97% cache hit ratio
  - 3% of the time the data was not in the cache, so needed a round trip to Oracle
- 99th percentile latency = 1ms

Percentile	Latency in ms
87	0.016
98	0.125
99.7	1
99.99	8
100	423

← NoSQL was 324 ms

Customer measured TimesTen Cache latency

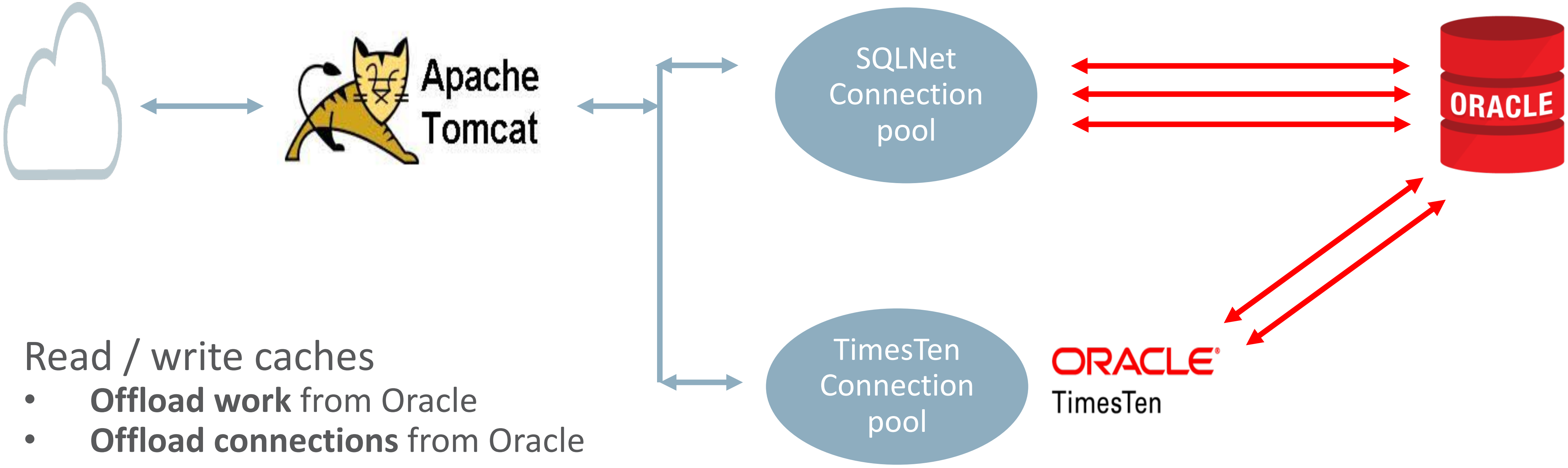
- 97% cache hit ratio
  - 3% of the time the data was not in the cache, so needed a round trip to Oracle
- 99th percentile latency = 1ms

Percentile	Latency in ms
87	0.016
98	0.125
99.7	1
99.99	8
100	423

Working to improve this

- Goal is < 50ms for 100th percentile
- Also define a 300 ms timeout

Improving the 100th percentile [max latency]



Read / write caches

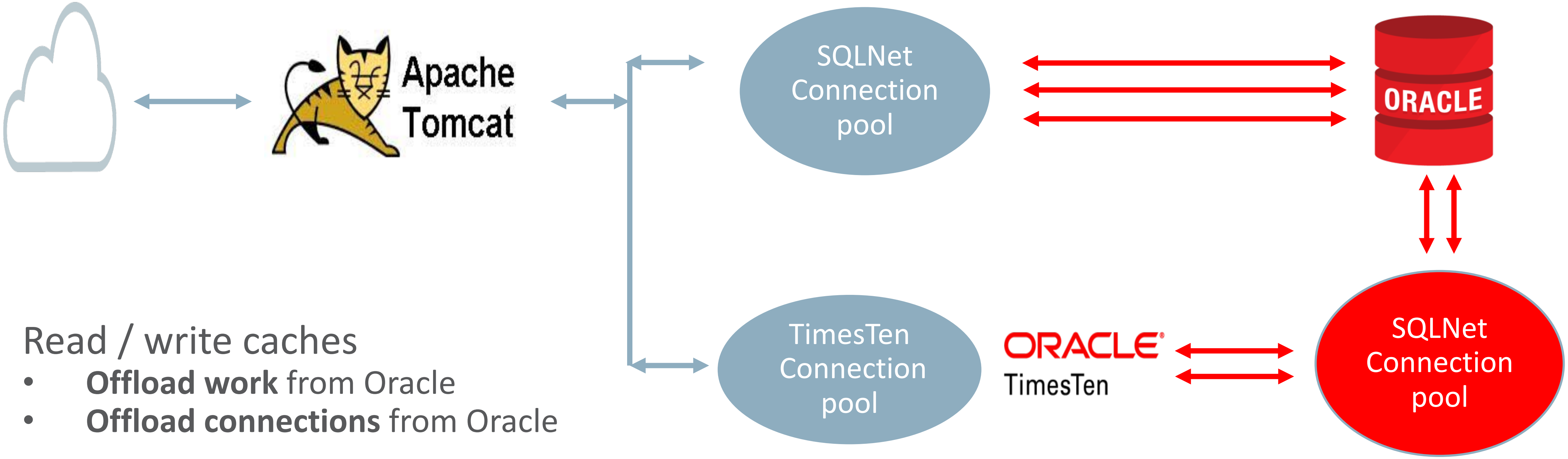
- **Offload work** from Oracle
- **Offload connections** from Oracle

For cache misses,
create a new SQLNet connection!

This can be **really slow …**

Can we do better?

Improving the 100th percentile [max latency]



Read / write caches

- **Offload work** from Oracle
- **Offload connections** from Oracle

For cache misses,
get connection from TT OCI CP
rather than the shared pool

Tune the pool for the expected min/max

Can we do better?

How is it possible

How is it possible

- Use a **really fast** In-memory SQL RDBMS

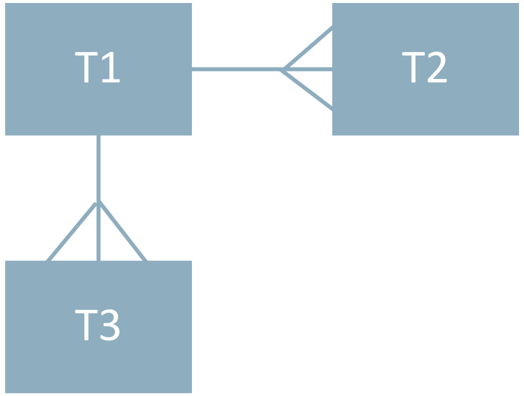
ORACLE
TimesTen

How is it possible

- Use a **really fast** In-memory SQL RDBMS
- Use a **Cache DB** that just requires configuration

ORACLE

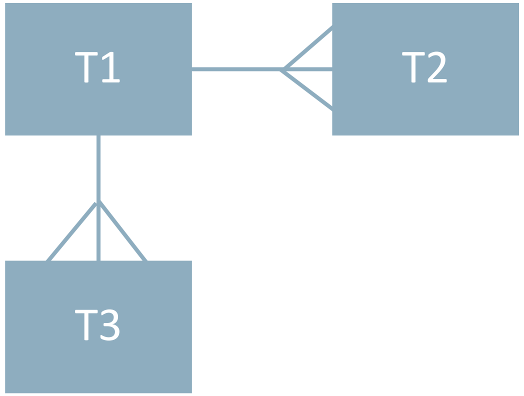
TimesTen

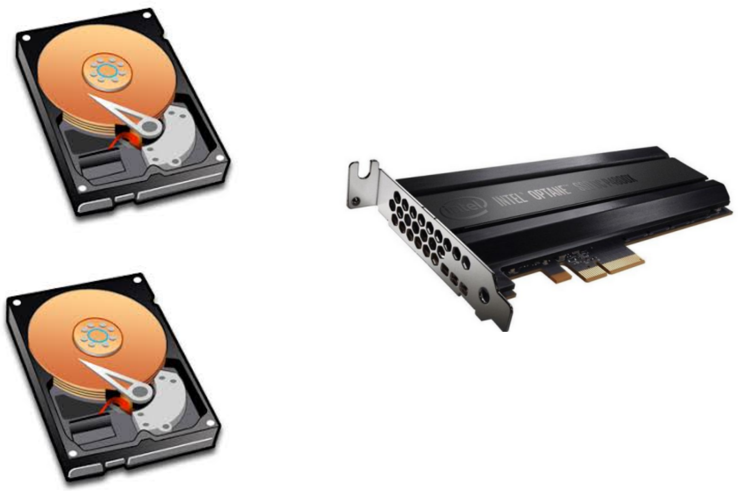


How is it possible

- Use a **really fast** In-memory SQL RDBMS
- Use a **Cache DB** that just requires configuration
- Some hardware and software tuning

ORACLE
TimesTen





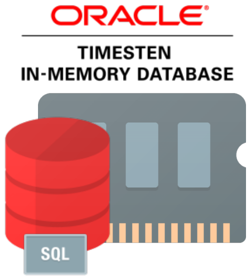
- Prepare & bind SQL statements
- Use the relevant indexes
- Do sensible joins
- Update statistics for SQL optimizer
- Check TimesTen SQL stats for SQL and IO bottlenecks

Most Widely Used Relational In-Memory Database

Deployed by Thousands of Companies

ERICSSON
CIMA
Designing The Future KDDI
UNITED STATES POSTAL SERVICE
KODIAK
中国移动 China Mobile
aspect
vodafone
中国平安 PINGAN
Bank of America
COMCAST
broadsoft
amdocs embrace challenge experience success
FCA FINANCIAL CONDUCT AUTHORITY
CISCO
hp
香港賽馬會 The Hong Kong Jockey Club
verizon
SMART
JAL JAPAN AIRLINES
T · · Mobile ·
koscom
NEC
WIND
Alcatel·Lucent
中国电信 CHINA TELECOM
DEUTSCHE BÖRSE GROUP
Zagrebačka banka UniCredit Group
rti Your systems. Working as one.
TTNET
松井証券
BSE EXPERIENCE THE NEW
金融界 JRJ.COM
J.P.Morgan
Rakuten Travel
intel
KTF
NYFIX
Nable Communications
Fed Centric TECHNOLOGIES, LLC
SHTECH
LOCKHEED MARTIN
TIM

Oracle TimesTen In-Memory Database



Relational Database

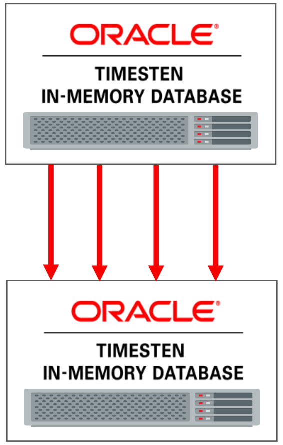
- Pure in-memory
- ACID compliant
- Standard SQL / PLSQL
- Entire database in DRAM

Persistent and Recoverable

- Database and Transaction logs persisted on local disk or flash storage
- Replication to standby and DR systems

Extremely Fast

- Microseconds response time
- Very high throughput



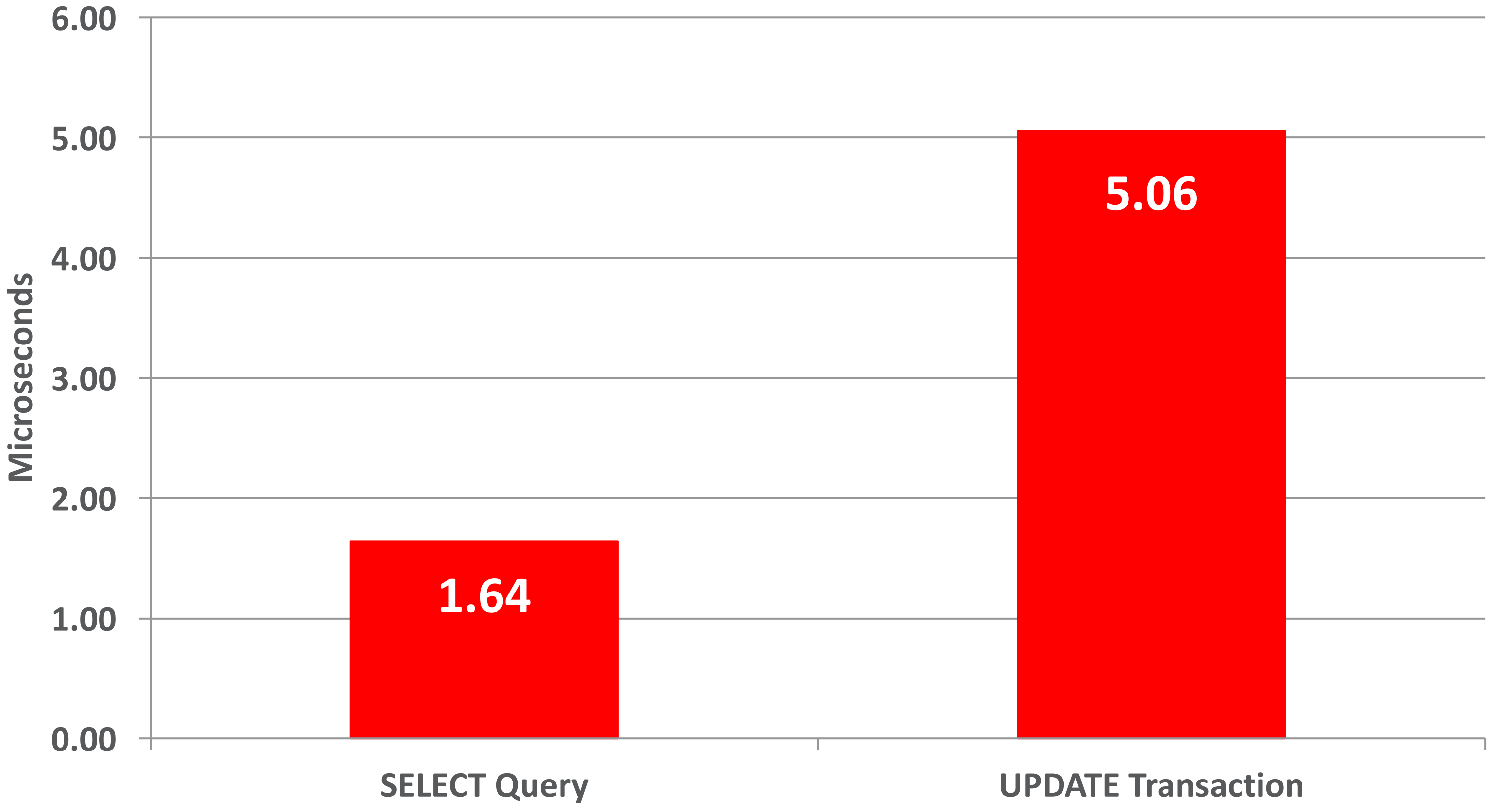
Highly Available

- Active-Standby and multi-master replication
- Very high performance parallel replication
- HA and Disaster Recovery

Performance – Response Time

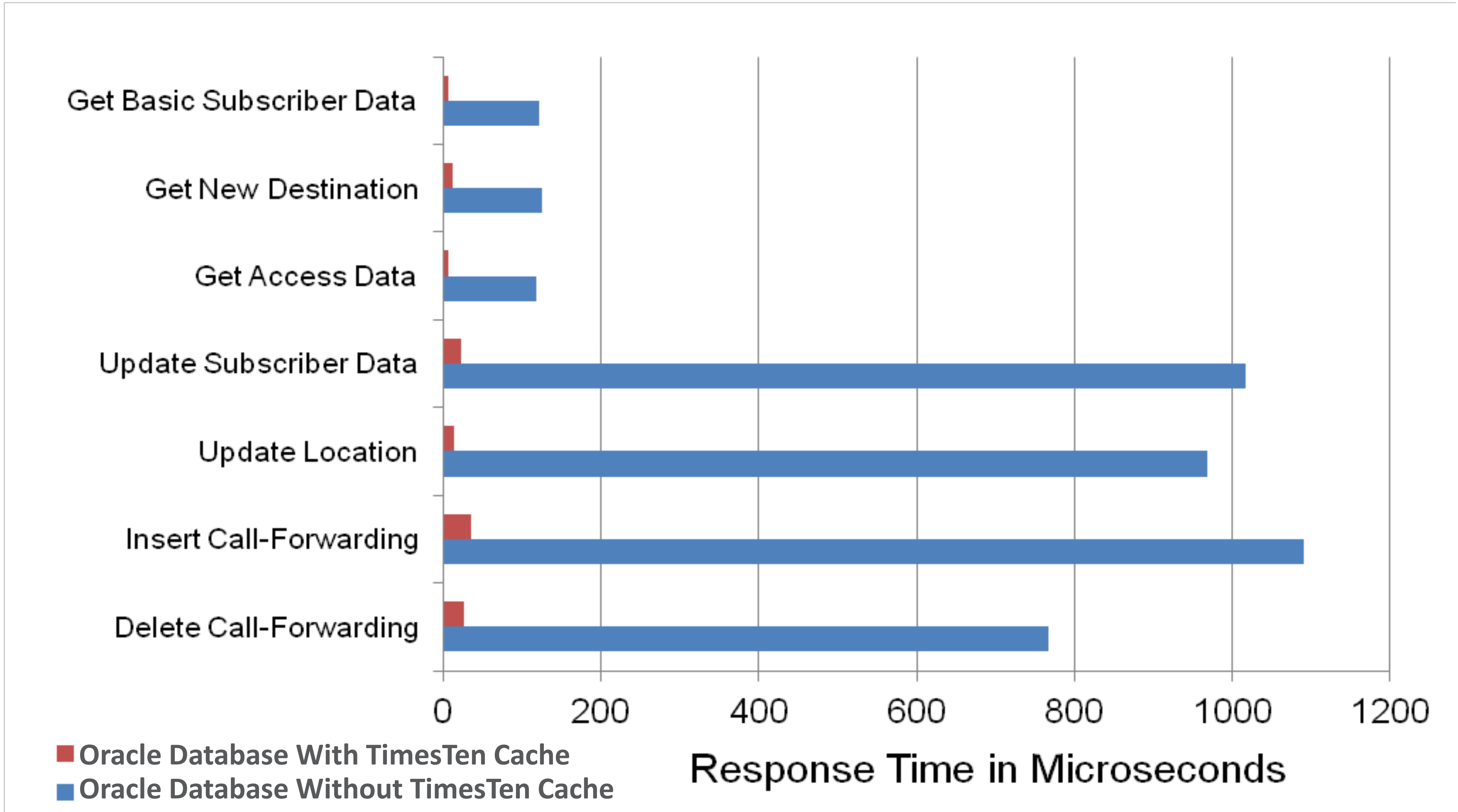
Low Latency - Microseconds Response Time

TPTBM Read and Update
E5-2699 v4 @ 2.20GHz
2 socket, 22 cores/socket,
2 threads/core
TimesTen 11.2.2.8.0
(100M rows, 17GB data)



HLR Mobile Transactions Response Time

Response Time Improvement With TimesTen Application-Tier Database Cache

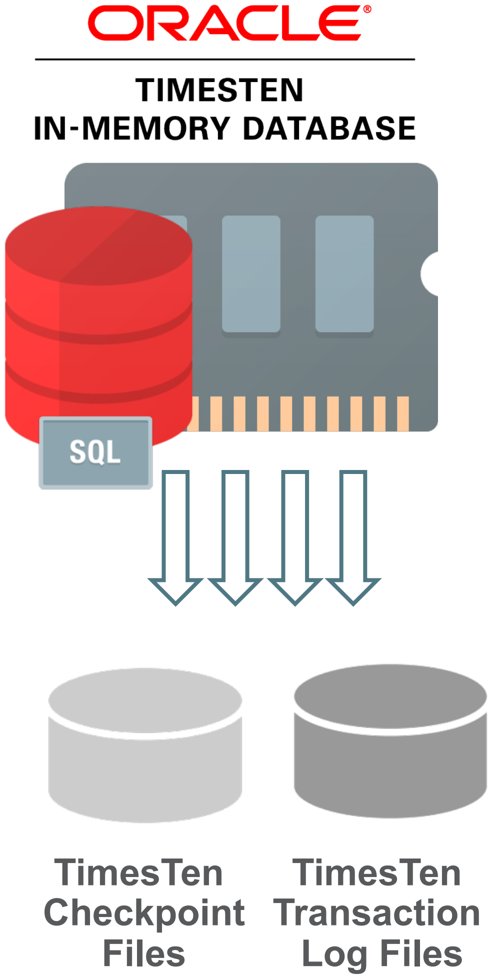


Speedup
17x
10x
17x
44x
69x
31x
29x

Intel® Xeon CPU E5-2680 @2.7GHZ 2 sockets 8 cores/socket 2 hyper-threads/core 32 vCPU

TimesTen In-Memory Database

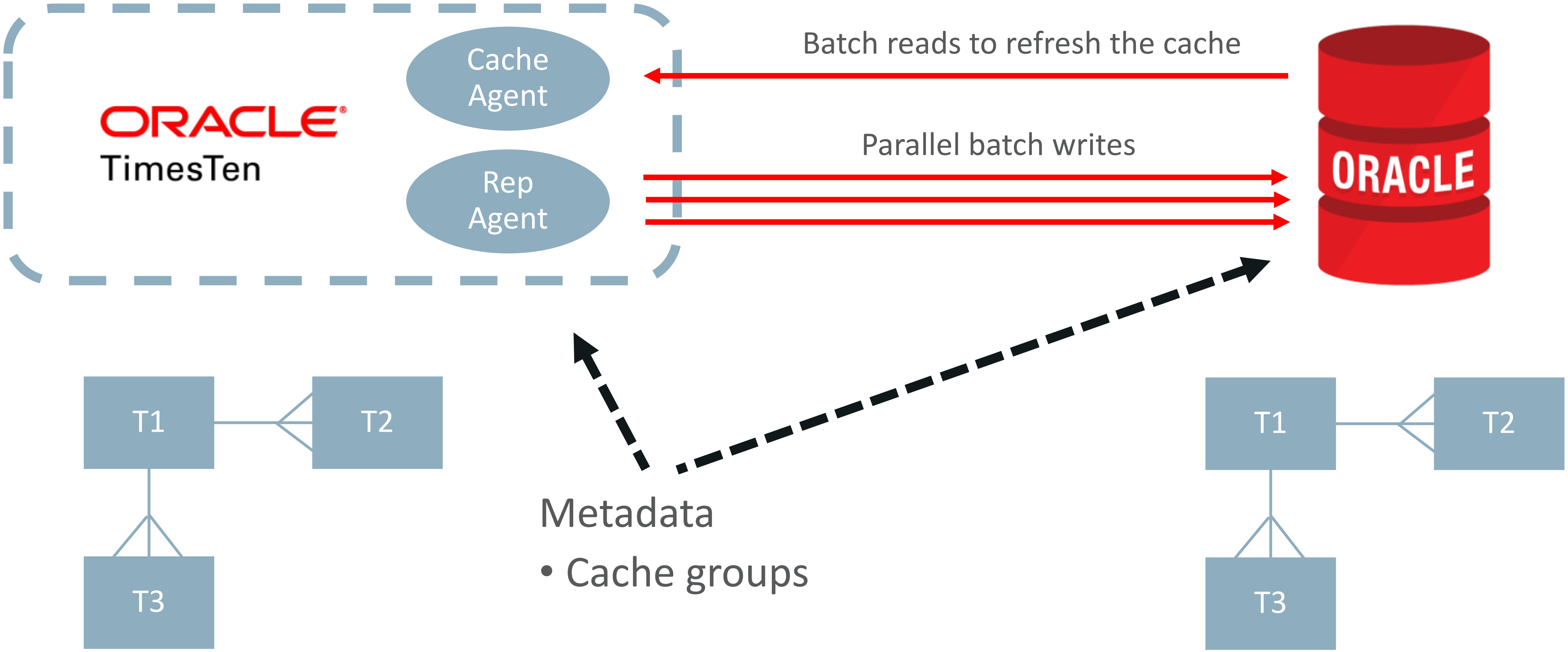
Persistence and Recovery



- TimesTen database persistence can be configured on
  - Flash, SSD, hard disk storage
- All transactions are logged and persisted
  - Redo, undo, and recovery
- Dual database Checkpoint files
  - Database restart
  - Database recovery after failures

Cache DB = Oracle Application Tier Database Cache

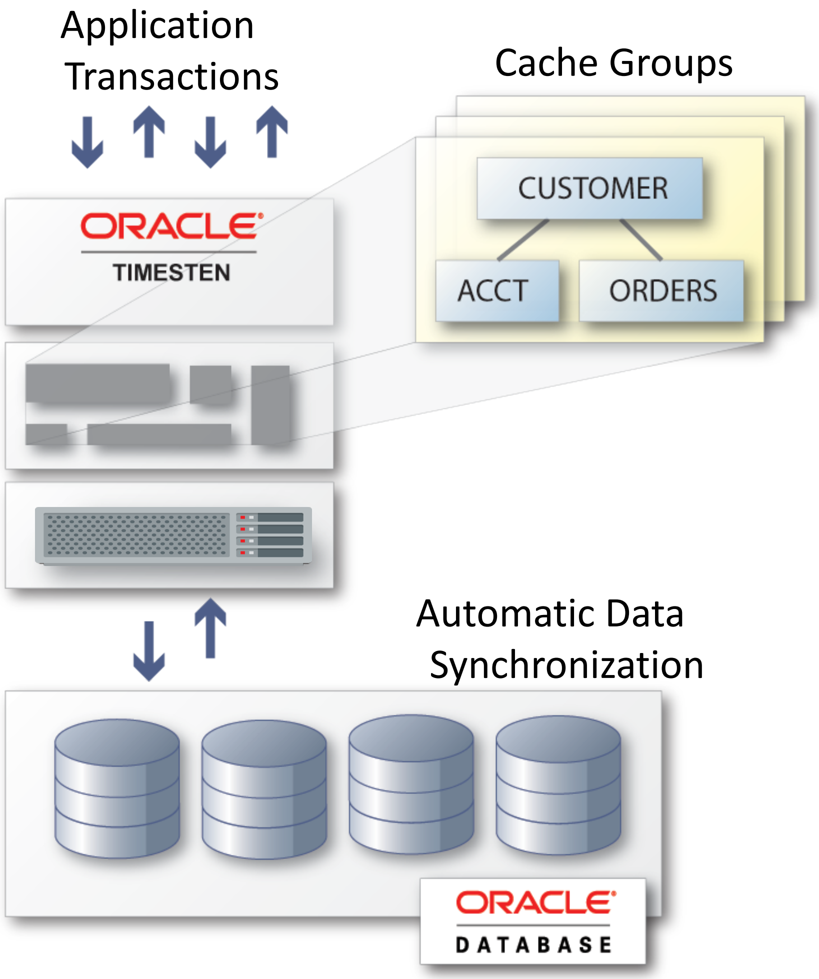
A feature of Oracle Enterprise Edition



Metadata

- Cache groups

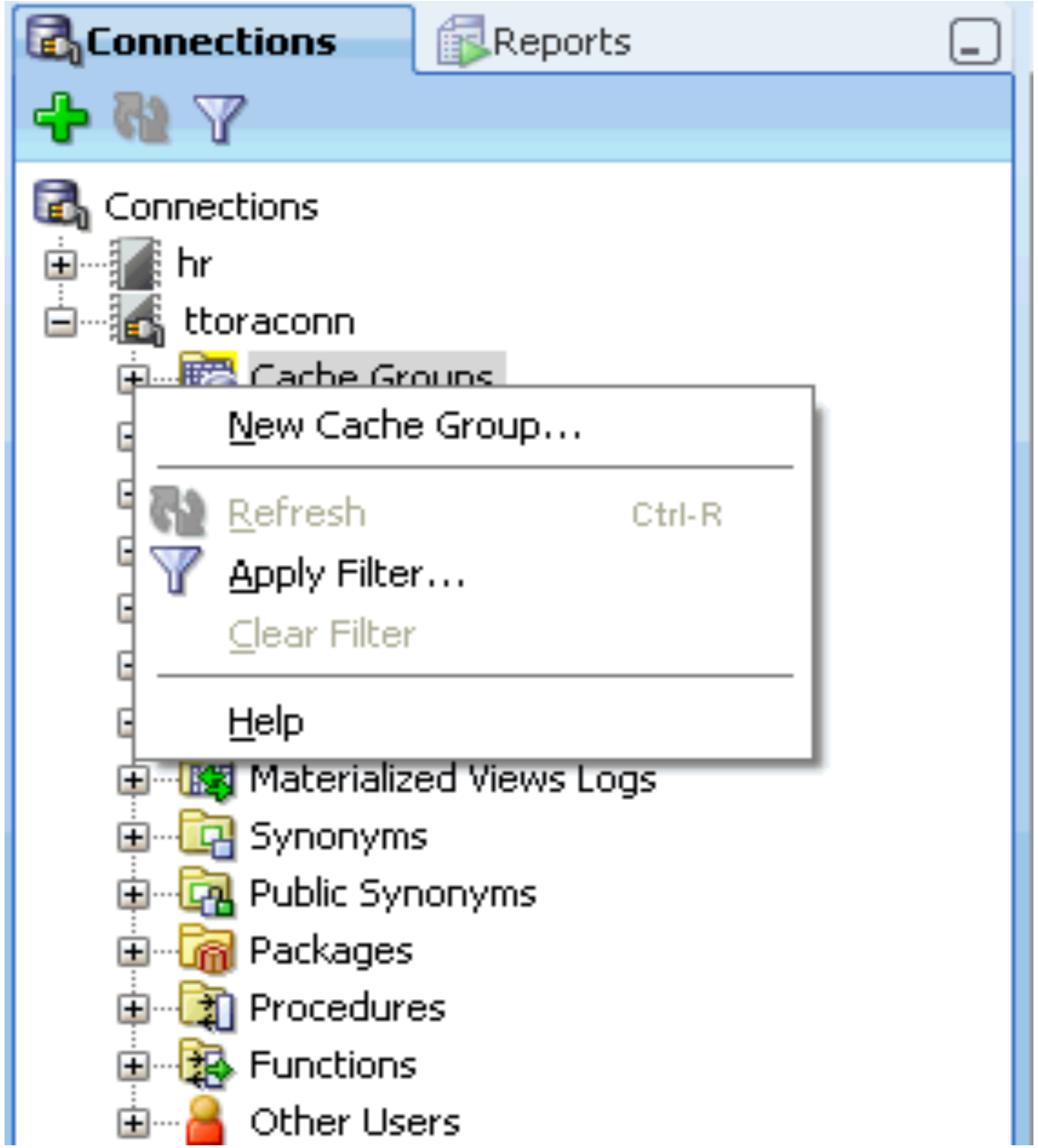
Flexible Cache Group Configurations

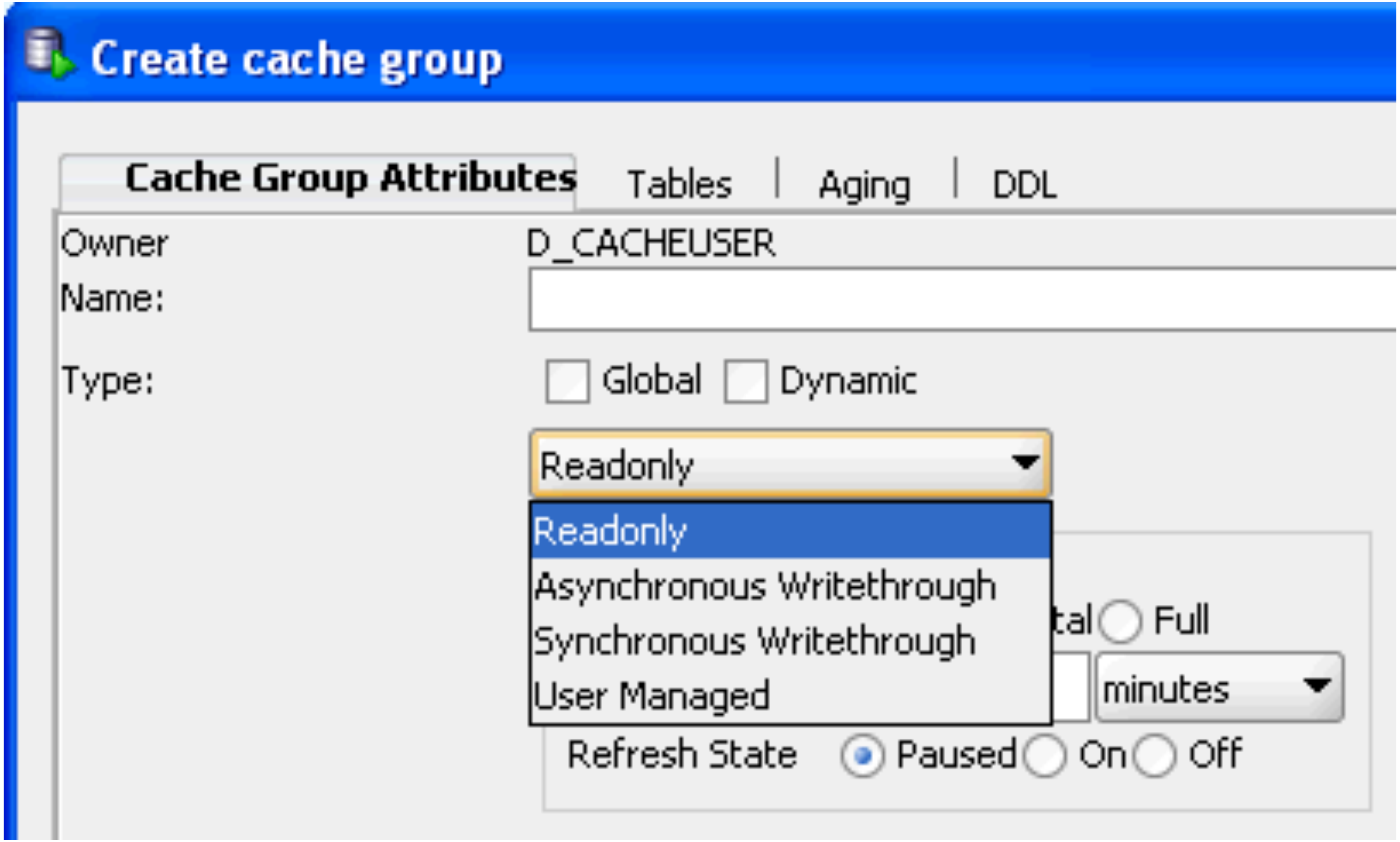


- Cache Group describes the Oracle Database tables to cache
  - All or subset of rows and columns
  - Defined using SQL

```
CREATE CACHE GROUP PremierUsers
   FROM OE.CUSTOMER (
       NAME VARCHAR2(100) NOT NULL,
       ADDR VARCHAR2(100)
  )
 WHERE OE.CUSTOMER.ORDER > 500;
```

- Cache tables are regular tables in TimesTen
  - Joins/search, insert/update/delete


Connections
Reports
Connections
hr
ttoraconn
Cache Groups
New Cache Group...
Refresh Ctrl-R
Apply Filter...
Clear Filter
Help
Materialized Views Logs
Synonyms
Public Synonyms
Packages
Procedures
Functions
Other Users



Create cache group

Cache Group Attributes | Tables | Aging | DDL

Owner D_CACHEUSER

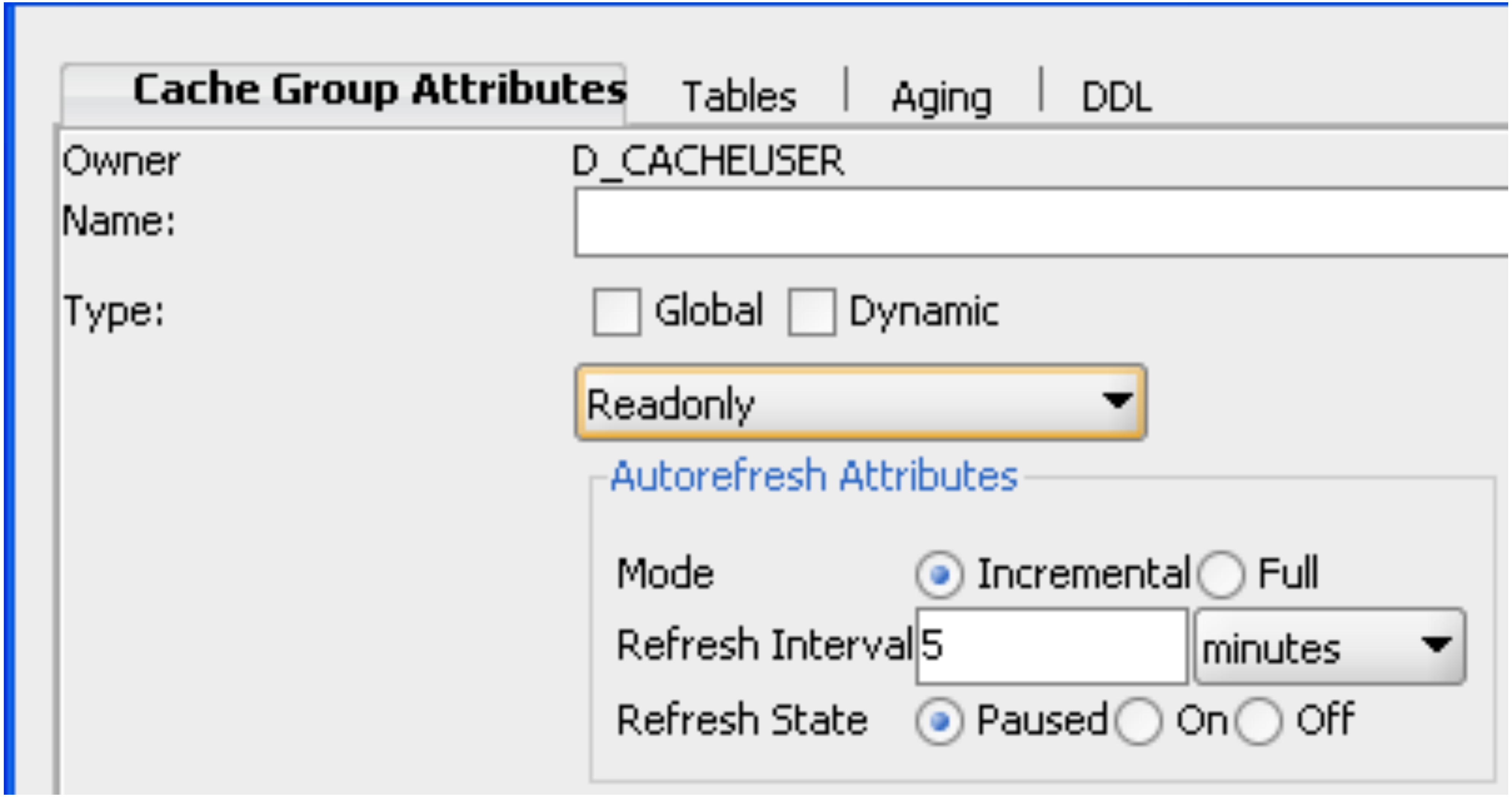
Name:

Type: □ Global □ Dynamic

Readonly

- Readonly
- Asynchronous Writethrough
- Synchronous Writethrough
- User Managed

Refresh State ● Paused ○ On ○ Off



Owner D_CACHEUSER

Name:

Type: □ Global □ Dynamic

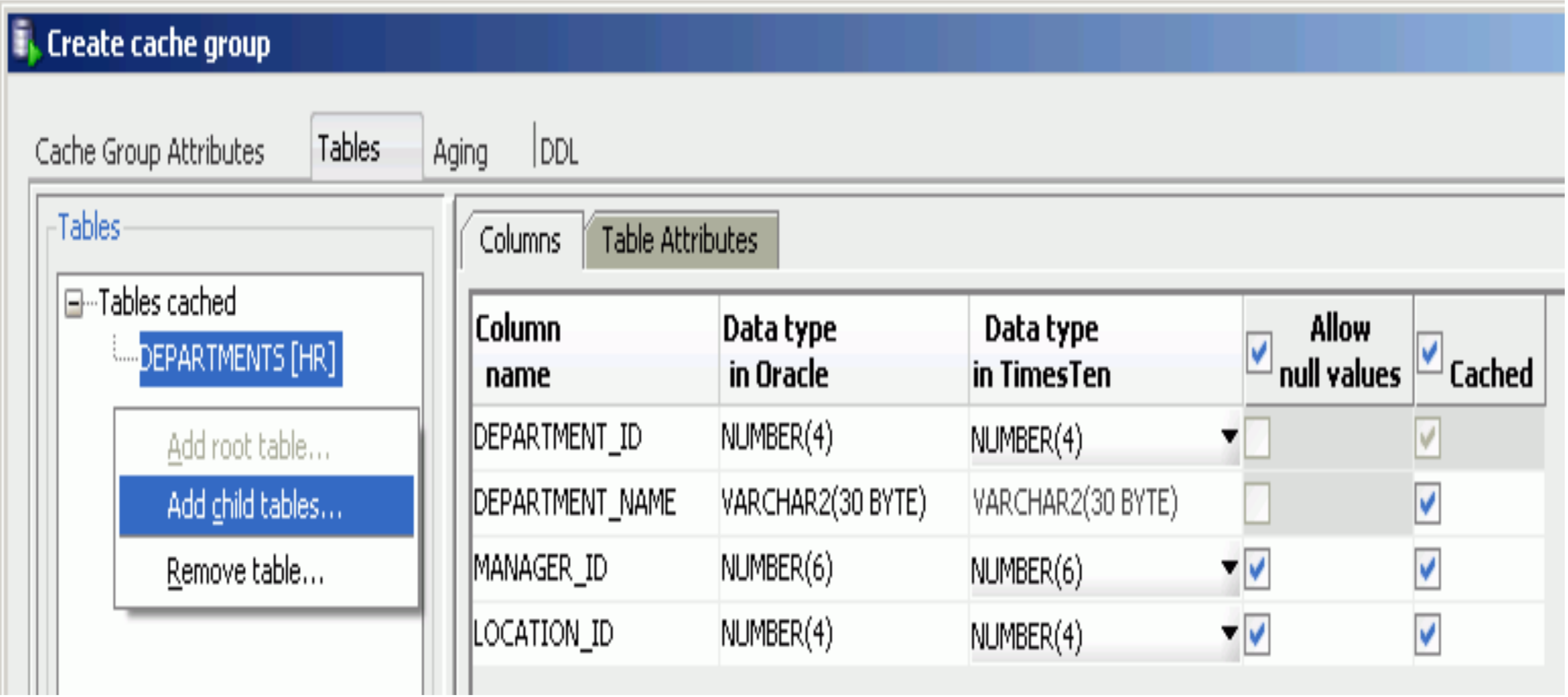
Readonly

Autorefresh Attributes

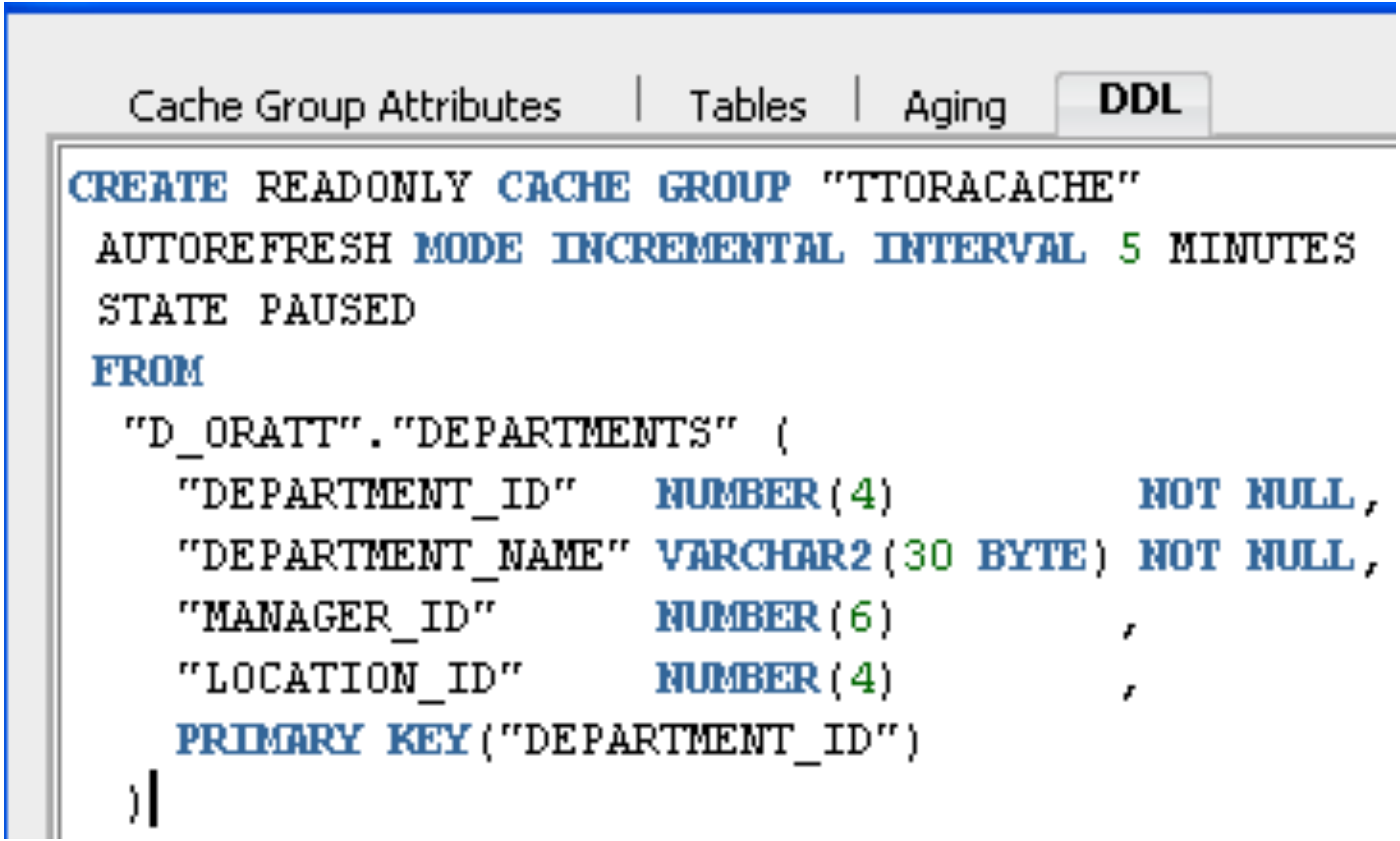
Mode ● Incremental ○ Full

Refresh Interval 5 minutes

Refresh State ● Paused ○ On ○ Off



Column name	Data type in Oracle	Data type in TimesTen	☑ Allow null values	☑ Cached
DEPARTMENT_ID	NUMBER(4)	NUMBER(4) ▼	□	☑
DEPARTMENT_NAME	VARCHAR2(30 BYTE)	VARCHAR2(30 BYTE)	□	☑
MANAGER_ID	NUMBER(6)	NUMBER(6) ▼	☑	☑
LOCATION_ID	NUMBER(4)	NUMBER(4) ▼	☑	☑



```
CREATE READONLY CACHE GROUP "TTORACACHE"
 AUTOREFRESH MODE INCREMENTAL INTERVAL 5 MINUTES
 STATE PAUSED
 FROM
  "D_ORATT"."DEPARTMENTS" (
    "DEPARTMENT_ID"   NUMBER(4)          NOT NULL,
    "DEPARTMENT_NAME" VARCHAR2(30 BYTE) NOT NULL,
    "MANAGER_ID"      NUMBER(6)         ,
    "LOCATION_ID"     NUMBER(4)         ,
    PRIMARY KEY("DEPARTMENT_ID")
  )
```

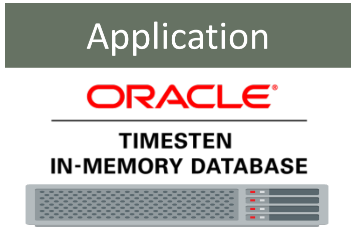
```
create dynamic read only cache group myReadCache1
autorefresh mode incremental interval  100 millseconds state on
from hr.employees ( employee_id   number (6) not null,
          first_name               varchar2(20),
          last_name                varchar2(25) not null,
          hire_date                date not null,
          job_id                   varchar2(10) not null,
          salary                   number (8,2),
          manager_id               number (6),
          department_id            number(4),
      primary key (employee_id)),
hr.job_history (employee_id number(6) not null,
          start_date date not null,
          end_date date not null,
          job_id varchar2(10) not null,
          department_id number(4),
 primary key (employee_id, start_date),
 foreign key (employee_id)
 references hr.employees (employee_id));
```

Oracle TimesTen

- One technology, two products, three deployment modes

Oracle TimesTen In-Memory Database

Standalone In-Memory Database for OLTP and analytics

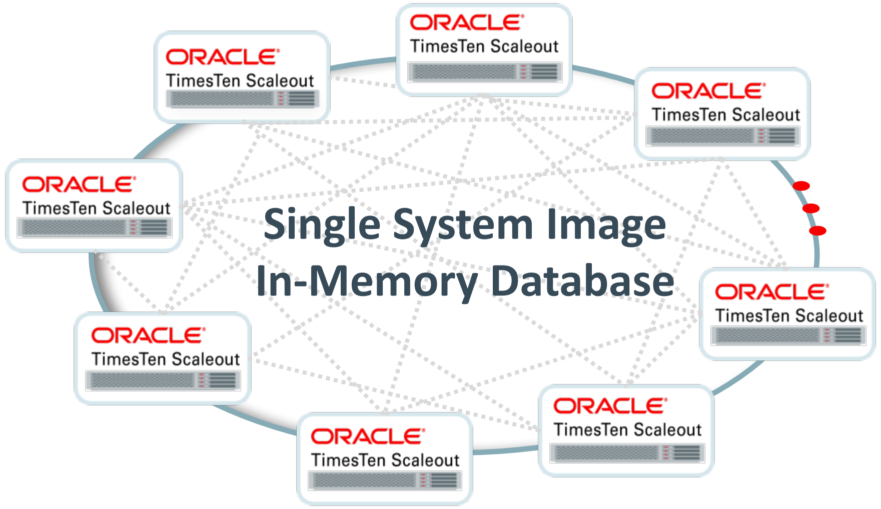


Application

ORACLE

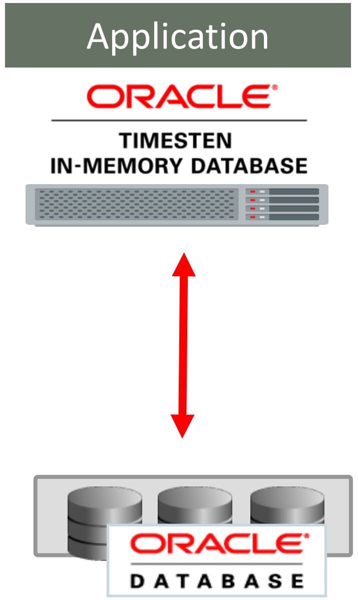
TIMESTEN
IN-MEMORY DATABASE

TimesTen Scaleout



Single System Image
In-Memory Database

Application Tier Database Cache



Application

ORACLE

TIMESTEN
IN-MEMORY DATABASE

ORACLE DATABASE

Summary

- Cloud Scale read/write caching with 1ms latency at 99^{th} percentile
- Oracle TimesTen is faster than any NoSQL DB
- Caching is configuration, not coding
- Use TimesTen PLSQL stored procedures to minimize network hops

- Write TimesTen apps as if it were an Oracle DB*
  - SQL, JDBC, PLSQL, OCI, ODBC, ODP.Net, Pro*C, Pro*COBOL
  - R, Python, Node.js, Go, Ruby & PHP

Subset of the Oracle SQL and PLSQL