Database In-Memory 2
Under The Hood

Tirthankar Lahiri
Vice President
Data Technologies and TimesTen
October 28, 2015

#DBIM12c
A Brief History of Time

Before OOW 2013

Miscellaneous Events

• Big Bang
• Dinosaurs
• Battlestar Galactica
• Relational Databases
• Back to the Future 1,2,3
• etc.
Relational Databases: **Row Format or Column Format?**

- **Transactions** run faster on row format
  - Example: Insert or query a sales order
  - Fast processing for few rows, many columns

- **Analytics** run faster on column format
  - Example: Report on sales totals by region
  - Fast accessing few columns, many rows

State of the Art: Choose One Format, Enjoy Benefits, Live with Drawbacks
A Brief History of Time

Before 2013

- Big Bang
- Dinosaurs
- Battlestar Galactica
- Relational Databases
- Back to the Future 1,2,3
- etc.

OOOW 2013

- Industry first dual-format in-memory database
- Trivial to deploy

Miscellaneous Events
Oracle Database In-Memory: Dual Format Architecture

- **BOTH** row and column formats for same table
- Simultaneously active and consistent
- OLTP uses existing row format
- Analytics uses new In-Memory Column format
In-Memory Columnar Format

- Pure in-memory column format
  - Enable for subset of database
  - Cheap to maintain – no logging or IO
  - Allows efficient OLTP
  - No change to disk format
- Built **seamlessly** into Oracle Database
  - **Transparent** to Applications
  - **All** Enterprise Features work ..
  - Availability – RAC, Flashback, DataGuard, etc.
  - Security – Encryption, Auditing, etc.
A Brief History of Time

Before OOW 2013

- Big Bang
- Dinosaurs
- Battlestar Galactica
- Relational Databases
- Back to the Future 1,2,3
- etc.

OOOW 2013

- Industry first dual-format inmemory database
- Trivial to deploy

July 2014

- Oracle Database 12.1.0.2
- Orders of magnitude faster analytic workloads
- Faster Enterprise mixed-workload OLTP

Miscellaneous Events

- etc.
Superfast In-Memory Scans

- Each CPU core scans only required columns
- SIMD vector instructions used to process multiple values in each instruction
  - E.g. Intel AVX instructions with 256 bit vector registers
- Billions of rows/sec scan rate per CPU core
  - Row format is millions/sec

Example:
Find all sales in state of CA
Superfast In-Memory Joins

**Example:** Find total sales in outlet stores

- Bloom filter created on dimension scan
- Bloom filter pushdown:
  - Filtering pushed down to fact scan
  - Returns only rows that are likely to be join candidates
- Joins tables **10x** faster

**Example: **

<table>
<thead>
<tr>
<th>Store ID</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>100</td>
</tr>
<tr>
<td>38</td>
<td>200</td>
</tr>
<tr>
<td>64</td>
<td>300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Store ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>15, 38, 64</td>
</tr>
</tbody>
</table>
Superfast In-Memory Reporting

Example: Report sales of footwear in outlet stores

- Create (empty) in-memory report outline during dimension scan
- Push down report outline aggregation to fact scan
- Reduces complex aggregations to series of fast in-memory scans
- Reports run 10x faster
  - Without predefined cubes
Much faster Mixed Workloads

- Complex OLTP is Slowed by Analytic Indexes
  
  Inserting one row into a table requires updating 10-20 analytic indexes: *Slow!*

- Column Store Replaces Analytic Indexes
  
  - Fast analytics on *any* columns
  - Column Store not persistent so DML is $2x-5x$ faster
**Scale-Out** In-Memory Database to Any Size

- Scale-Out across servers to grow memory and CPUs
- DISTRIBUTE clause: by Partition, Sub-Partition, or Rowid Range
- In-Memory **queries parallelized** across servers to access local column data
- **Direct-to-wire** InfiniBand protocol speeds messaging
A Brief History of Time

Before 2013
- Miscellaneous Events
  - Big Bang
  - Dinosaurs
  - Battlestar Galactica
  - Relational Databases
  - Back to the Future 1,2,3
  - etc.

Sept 2013
- Database In-Memory Announced
  - Industry first dual-format inmemory database
  - Trivial to deploy

July 2014
- Database In-Memory General Availability
  - Oracle Database 12.1.0.2
  - Orders of magnitude faster analytic workloads
  - Faster Enterprise mixed-workload OLTP

7/2014 | OOW 2015
- Database In-Memory Adoption
  - Fastest growing option in the history of Oracle Database
  - Further enhancements across interim update releases

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
Fastest Customer Adoption Of All Time

Oracle Database In-Memory and Multitenant Options
Customer **In-Memory** Sessions here at Openworld...

Monday 2:45pm
Room 301

Monday 4:00pm
Room 310

Tuesday 4:00pm
Room 307

Wednesday 3:00pm
Room 254

Wednesday 1:45pm
Room 103

Wednesday 1:45pm
Room 103

Wednesday 1:45pm
Room 103

Wednesday 1:45pm
Room 103
“We expected these reports in 10 seconds, but never sub-second.”

– Dan Huls
Senior Technical Director
AT&T Wi-Fi

IDC White Paper: Memory Optimized Transactions and Analytics in One Platform
Mixed Workloads: Improvement to date from GA*

<table>
<thead>
<tr>
<th>Metric</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Utilization</td>
<td>337%</td>
</tr>
<tr>
<td>Overall Throughput</td>
<td>285%</td>
</tr>
</tbody>
</table>

Synthetic mixed workload:
- Analytic scans (1%)
- OLTP queries (30%)
- Update (30%)
- Insert (24%)
- Delete (15%)

(*) Disclaimer: Results for illustrative purpose only. Your Mileage May Vary (workload, queries, data)
A Brief History of Time

Before OOW 2013

- Miscellaneous Events
  - Big Bang
  - Dinosaurs
  - Battlestar Galactica
  - Relational Databases
  - Back to the Future 1, 2, 3
  - etc.

OOOW 2013

- Database In-Memory Announced
  - Industry first dual-format inmemory database
  - Trivial to deploy

July 2014

- Database In-Memory General Availability
  - Oracle Database 12.1.0.2
  - Orders of magnitude faster analytic workloads
  - Faster Enterprise mixed-workload OLTP

7/2014 | OOW 2015

- Database In-Memory Adoption
  - Fastest growing option in the history of Oracle Database
  - Further enhancements across interim releases
  - Oct 21, 2015: Marty McFly arrives in Hill Valley

2016

- Database In-Memory Release 2

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.
Back to the Future
Database In-Memory Release 2
Faster In-Memory Joins: In-Memory Join Groups

Example:
Find total sales in outlet stores

Create Join Group store_sales_jg (STORES, STORE_ID), (SALES, STORE_ID);

- **Today:** Joins run on decompressed data
  - Data populated in-memory is compressed
  - Qualifying data is decompressed before join

- **12.2:** Join Groups
  - Users specify which columns used for joins across tables as a **Join Group**
  - Columns specified in a group use same dictionary for encoding
  - Joins occur on symbols rather than data
  - **3x-5x** faster in-memory joins
Faster Expression Evaluation: **In-Memory Expressions**

**Example:**
```
Select PRICE * TAX
From   SALES;
```

- **Today:** Commonly used expressions are recomputed every time

- **12.2: In-Memory Expressions**
  - Two modes:
    - Manual: User defined virtual columns
    - Automatic: Frequently-evaluated expressions
  - Many uses – e.g. Arithmetic Expressions, Type Conversions, In-Memory optimized JSON format
  - Maintained consistently with source columns
  - All In-Memory query performance optimizations apply - Vector Processing, Storage Indexes, Bloom Filter joins etc.
In-Memory Expressions: Performance Gain Example

Analytic Data Set (TPC-H Schema)

- Explicitly declared in-memory expressions
  - $price \times (1 - \text{discount})$
  - $Price \times (1 - \text{discount}) \times (1 + \text{tax})$
- Shows major performance gains for analytic queries
In-Memory Expressions: Performance Gain Example

- JSON automatically stored in memory optimized format
- Allows much faster processing of JSON_TABLE operations
- Explicitly created IME on JSON_VALUE provides massive speedup for extracting scalar values from JSON fields
SQL in Silicon: **10x** Acceleration of Database In-Memory

**SPARC M7**

- **Today:** We use standard SIMD vector instructions designed for Graphics and HPC, not for Databases
  - Translating database query operations to SIMD vector instructions is complex and expensive

- **SPARC M7:** New M7 chip has 32 Database Acceleration Engines (DAX), like having 32 specialized cores for Database In-Memory
  - Directly runs basic database query primitives
    - E.g. find all values that match ‘California’
    - **2-10x** speedup – up to 170 Billion Rows per second
Capacity in Silicon: Double In-Memory Capacity

- **TODAY:** Compress FOR QUERY uses value compression
  - Fast since queries run directly on compressed data
  - Lightweight compression, lower compression ratio

Compress for CAPACITY uses bit pattern compression
- Uses Oracle Zip (OZIP): 2-3x better compression than QUERY
- Slower since data must be decompressed prior to access

- **SPARC M7:** Compress FOR QUERY HIGH uses OZIP on SPARC M7
  - DAX includes specialized OZIP decompression engine
  - Runs OZIP decompress at full memory speed, > 120 GB/sec
  - Pipelines decompression and data processing in hardware
  - Doubles memory capacity with negligible performance penalty

32 OZIP Decompressors
In-Memory Column Store on Active Data Guard

- **Today**: In-Memory queries possible only on Primary Database
  - Also on Logical Standby

- **12.2**: In-Memory queries also possible on Active Data Guard (Physical Standby)
  - Analytic reporting can be offloaded to standby
  - Completely different data can be populated into IM column store on standby
  - Different standbys can have different data in their IM column stores
  - Increases capacity and improves availability for inmemory column store
Columnar Flash Cache: In-Memory Columnar on Flash

- **Today:** IM column format only in DRAM of Compute Node
  - Each database node has its own IM column store
  - Data can be distributed or duplicated across nodes

- **12.2:** IM column format on Flash Cache of Storage Node
  - Populate Flash Cache with IM columnar format
  - Smart Scans leverages all IM optimizations:
    - SIMD vector processing
    - Storage index pruning
    - Predicate / Aggregate processing optimizations
  - Multiplies effective Columnar Capacity by **10-100x**
Faster Restore of In-Memory Column Store: **Fast-Start**

- **Today**: IM column store is always rebuilt on startup
  - Recreated from row format (populate)

- **12.2**: IM column format persisted to storage
  - IM column store contents checkpointed to SecureFile Lob on populate
  - When DB restarts population is faster as population process reads the column format directly from storage
  - Faster restore (3-5x) of column store since no need to reformat data
Automatic Data Optimization in 12.1: Recap

- An in-memory heat map tracks disk based block and segment access
  - Heat map is periodically written to storage
  - Data is accessible by views or stored procedures
- Users can attach policies to tables to compress or tier data based on access
  - Tables, Partitions or Sub-partitions can be moved between storage tiers and compression levels
  - Online, no impact to data availability
  - Allows automatic data tiering
Automatic Data Optimization in 12.1: Recap

- An in-memory heat map tracks disk based block and segment access
  - Heat map is periodically written to storage
  - Data is accessible by views or stored procedures

- Users can attach policies to tables to compress or tier data based on access
  - Tables, Partitions or Sub-partitions can be moved between storage tiers and compression levels
  - Online, no impact to data availability
  - Allows automatic data tiering
Automatic Data Optimization with Database In-Memory

- **Today**: The user must specify the contents of the inmemory column store

- **12.2**: IM column store is managed automatically as a new data tier
  - *Two levels* of automation
    - **Policy Mode** - Supports user policies to populate and evict segments
    - **Fully Automatic Mode** - Segment heat map used to add & evict segments based on memory pressure
Automatic Data Optimization with Database In-Memory

- **Today**: The user must specify the contents of the in-memory column store.

- **12.2**: IM column store is managed automatically as a new data tier.
  - *Two possible levels* of automation:
    - **Policy Mode** - Supports user policies to populate and evict segments.
    - **Fully Automatic Mode** - Segment heat map used to add & evict segments based on memory pressure.

In-Memory Column Store

- PRODUCTS
- SUPPLIERS
- SALES
- COSTS
- CUSTOMERS

- SALES_p1
- SALES_p2
- SALES_p3

- Populate
- Evict
Enhanced In-Memory Advisor

**Today:** In-Memory Advisor is a standalone utility
- Available as an OTN download

**12.2:** In-Memory Advisor fully integrated with Enterprise Manager and RDBMS
- Accessible from Advisors Home and In-Memory Central
- PL/SQL interface: DBMS_INMEMORY_ADVISOR
- Interactive interface: $ORACLE_HOME/rdbms/admin
- Improved recommendations and performance estimates
Resizeable In-Memory Area

Today: In Memory Area is a static memory pool. If out of memory, need to bounce instance to increase its size.

12.2: In Memory Area can be dynamically resized on a running instance.

- Can shrink buffer cache and grow inmemory area if running low on space.
- No downtime to increase size.

ALTER SYSTEM SET INMEMORY_SIZE = 80G
Database In-Memory Release 2: Summary

**Faster Performance**
- In-Memory Expressions
- In-Memory Join Groups
- In-Memory Ordering
- SQL In Silicon

**Greater Capacity**
- Capacity in Silicon
- In-Memory Columnar Flash Cache
- In-Memory on Active Data Guard

**Easier to Manage**
- Enhanced IM Advisor
- Automatic In-Memory Data Tiering
- Resizeable In-Memory Area

**Improved High Availability**
- In-Memory on Active Data Guard
- In-Memory Fast Start
In-Memory Database Technology Across Tiers

**In-Memory Row Store**
- Application

**TimesTen In-Memory Database**
- Embeddable In-Memory Database for Application Tier
- Primary Usecase: Latency-critical custom OLTP
  - **Microsecond** Response Time
- Standalone Database or as Application-Tier Cache for Oracle Database

**Oracle Database In-Memory**
- Dual-Format In-Memory Database
- Primary Usecase: Real Time Analytics on any source
  - **Billions of Rows/Sec** analytic data access
- Faster mixed-workload enterprise OLTP
- Storage-Tiering: Combines best of memory, flash, disk
- Transparent: packaged apps run with no changes
Additional Resources

Join the Conversation
- https://twitter.com/db_inmemory
- https://www.facebook.com/OracleDatabase

Related White Papers
- Oracle Database In-Memory White Paper
- Oracle Database In-Memory Aggregation Paper
- When to use Oracle Database In-Memory
- Oracle Database In-Memory Advisor

Related Videos
- In-Memory YouTube Channel
- Managing Oracle Database In-Memory
- Database In-Memory and Oracle Multitenant
- Industry Experts Discuss Oracle Database In-Memory
- Software on Silicon

Any Additional Questions
- Oracle Database In-Memory Blog
# Oracle Database In-Memory Schedule for Oracle Open World

<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Location</th>
<th>Speaker</th>
</tr>
</thead>
</table>
| Tuesday Oct 27th
11:00 – 11:45   | Best Practices for Getting Started with Oracle Database In-Memory    | Moscone South Room 104 | Maria Colgan Product Manager, Oracle         |
| Tuesday Oct 27th
17:00 – 17:45   | Oracle Database In-Memory—What’s New and What’s Coming               | Moscone South Room 103 | Juan Loaiza Senior Vice President, Oracle    |
| Wednesday Oct 28th
13:45 – 14:30  | Oracle Database In-Memory Customer Panel                              | Moscone South Room 102 | Maria Colgan with 4 In-Memory Customers      |
| Wednesday Oct 28th
16:15 – 17:00  | Oracle Database In-Memory: Under the Hood                             | Moscone South Room 103 | Tirthankar Lahiri – Vice President, Oracle   |
| Monday          | Oracle Database In-Memory Boot Camp: Everything You Need to Get Started | Hotel Nikko Room Peninsula | Andy Rivenes & Andy Yao                  |
| Tuesday        | 08:45                                                                 |                        |                                              |
| Wednesday      | 08:45                                                                 |                        |                                              |
| Thursday       | 09:30                                                                 |                        |                                              |
If you have more questions later, feel free to ask