In-Memory Analytics: A comparison between Oracle TimesTen and Oracle Essbase
Agenda

• Introduction
  ‣ Why In-Memory?
  ‣ Options for In-Memory in Oracle Products
    - Times Ten
    - Essbase

• Comparison - Essbase Vs Times Ten
  ‣ Architecture
  ‣ Data Size Handling (Volume)
  ‣ BI EE Native Support - Aggregates
  ‣ Ease of Querying
  ‣ Real-Time loads
  ‣ Reporting Performance
  ‣ Incremental Updates
  ‣ Integration with Other Systems

• Essbase-Times Ten
  ‣ Use Case Scenarios
Why In-Memory?
Memory is Faster, Cheaper, & has more Capacity Today

64X More Capacity
25X Cheaper
50,000X Faster

Faster Analysis, Faster Reporting, Faster Planning
Better Interactivity, Better Visualizations, Better Intelligence
More Users, More Data, More Calculations

* - Thomas Kurian Presentation
In-Memory Options - Oracle Stack

- **Transaction Processing**
  - In-Memory Database Cache
    - Times Ten in Oracle Database
    - In-Memory Transaction Processing
  - Coherence
    - In-Memory Transaction Processing

- **Analytical Processing**
  - Times-Ten
    - In-Memory Analytical Processing
    - Exalytics
  - Essbase
    - In-Memory Analytical Processing
    - Multi-Dimensional Database
In-Memory Database Cache

- Uses Times-Ten
- Used for speeding up Transaction Processing
- Caches frequently used tables in-memory
- Supports sql & pl/sql grammar
- Out of the box sync with the Oracle Database
- Very little latency/IO
  - Extremely fast transaction updates
Coherence

• Formerly Tangosol
• In-Memory Data Grid
• Distributed Caching at Application Tier
• Full support for Java & Non-Java
• Not a database cache - Application cache
  ‣ Can cache data
• Focuses on Application Tier
  ‣ Eg. Stock Trading Apps
• Inherent part of WLS App Server
Times Ten for Exalytics

- Built with original Times-Ten Codebase
- Extensive additional features to support analytic functions
  - Work like in-memory Oracle Database
  - All Major Oracle DB Analytical functions function-shipped
- Special release for Exalytics
- Native support from BI EE
- Supports only SQL grammar
- Columnar Compression
Essbase

- Multi-Dimensional Analytical Database
- 2 types of Applications
  - Block Storage (in-memory Kernel - With Control)
  - Aggregate Storage (in-memory Kernel - limited control)
- Comprehensive MDX support
- Native Support for BI EE
- Native Support from Excel - Adhoc querying
Exalytics
Oracle TimesTen and Oracle Essbase - A Comparison
Parameters of Comparison

- Architecture
- BI EE Native Support - Aggregates
- Ease of Querying
- Real-Time loads
- Reporting Performance
- Incremental Updates
- Integration with Other Systems
Oracle TimesTen Vs Oracle Essbase

Architecture
Times Ten
Times Ten

Data loads into memory during startup

All Data loaded to Memory
Times Ten
Essbase
Traditional Non-In Memory Databases

- Very good from storage standpoint
  - Performance degrades due to IO
- Common Reasons behind Performance Bottlenecks
  - Large IO to retrieve data
  - Lack of Memory to hold all data in-memory
  - Multiple random queries
    - Not all can fit in memory (frequent memory swapping)
  - Not all databases are optimized to push everything into memory
Times Ten - Use Case - Architecture
Source Data

<table>
<thead>
<tr>
<th>Products</th>
<th>Time</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOB</td>
<td>Year</td>
<td>Country</td>
</tr>
<tr>
<td>Brand</td>
<td>Quarter</td>
<td>City</td>
</tr>
<tr>
<td>Product</td>
<td>Month</td>
<td>Street</td>
</tr>
</tbody>
</table>

Source Data → SALES

Saturday, September 29, 12
Required Analysis

- Products: LOB, Brand, Product
- Time: Year, Quarter, Month
- Customer: Country, City, Street
- Sales:

T: +44 (0) 8446 697 995 or (888) 631 1410 (USA)  E: enquiries@rittmanmead.com  W: www.rittmanmead.com
Times Ten - Typical Usage

Times Ten - In Memory Tables

- Pre-Summarized &Loaded
  - Brand
  - Year
  - City
  - Sales

- Pre-Summarized &Loaded
  - LOB
  - Quarter
  - Country
  - Sales

- Loaded Directly from Source
  - Product
  - Month
  - Street
  - Sales

Sales
Times Ten - Memory Size

1. Easier to gauge the size of the Cache
2. Optimal usage of Cache size
3. Easy to judge the number of possible rows before hand
## Times Ten - Run-Time Queries

### Pre-Summarized & Loaded

<table>
<thead>
<tr>
<th>Brand</th>
<th>Year</th>
<th>City</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOB</td>
<td>Quarter</td>
<td>Country</td>
<td>Sales</td>
</tr>
</tbody>
</table>

### Loaded Directly from Source

<table>
<thead>
<tr>
<th>Product</th>
<th>Month</th>
<th>Street</th>
<th>Sales</th>
</tr>
</thead>
</table>

### All Aggregated at Run-Time

<table>
<thead>
<tr>
<th>LOB</th>
<th>Year</th>
<th>City</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOB</td>
<td>Year</td>
<td>Country</td>
<td>Sales</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Brand</th>
<th>Month</th>
<th>Country</th>
<th>Sales</th>
</tr>
</thead>
</table>

### Run-Time Aggregations

---

### Times Ten - In Memory Tables

---

T: +44 (0) 8446 697 995 or (888) 631 1410 (USA)  E: enquiries@rittmanmead.com  W: www.rittmanmead.com
Times Ten - Run-Time Queries

Pre-Summarized & Loaded

<table>
<thead>
<tr>
<th>Brand</th>
<th>Year</th>
<th>City</th>
<th>Sales</th>
</tr>
</thead>
</table>

Pre-Summarized & Loaded

<table>
<thead>
<tr>
<th>LOB</th>
<th>Quarter</th>
<th>Country</th>
<th>Sales</th>
</tr>
</thead>
</table>

Loaded Directly from Source

<table>
<thead>
<tr>
<th>Product</th>
<th>Month</th>
<th>Street</th>
<th>Sales</th>
</tr>
</thead>
</table>

All Retrieved at Run-Time

<table>
<thead>
<tr>
<th>Brand</th>
<th>Year</th>
<th>City</th>
<th>Sales</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>LOB</th>
<th>Quarter</th>
<th>Country</th>
<th>Sales</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Month</th>
<th>Street</th>
<th>Sales</th>
</tr>
</thead>
</table>

No run-time aggregations

Times Ten - In Memory Tables
Times Ten - As in-memory Engine

- Aggregated data stored in tables
- Non-existing aggregated data has to be retrieved through SQL
- Comprehensive analytical functions supported
- Size of the in-memory cache
  - Easy to judge
  - Can be planned
- Possible Performance Issues
  - When retrieving aggregated non-existing data
### Times Ten - Source Data Change

#### Pre-Summarized & Loaded
- **Brand**
- **Year**
- **City**
- **Sales**

#### Pre-Summarized & Loaded
- **LOB**
- **Quarter**
- **Country**
- **Sales**

#### Loaded Directly from Source
- **Product**
- **Month**
- **Street**
- **Sales**

1. **Source Data Changes**
   - 1. Incremental Updates
   - 2. New Data

---

#### Times Ten - In Memory Tables

T: +44 (0) 8446 697 995 or (888) 631 1410 (USA)  
E: enquiries@rittmanmead.com  
W: www.rittmanmead.com

---

Saturday, September 29, 12
Times Ten - Source Data Change

- **Pre-Summarized & Loaded**
  - Brand
  - Year
  - City
  - Sales

- **Pre-Summarized & Loaded**
  - LOB
  - Quarter
  - Country
  - Sales

- **Loaded Directly from Source**
  - Product
  - Month
  - Street
  - Sales

Source Data Changes
- Reload Source

Reload & Recalculate Aggregates
- Complete Reload

Times Ten - In Memory Tables
Times Ten - As in-memory Engine

- Whenever Source Data Changes
  - All aggregates need to be repopulated
  - Depending on the size of aggregates
    - Can take a long time
  - No trickle feed incremental update for aggregates
  - Trickle feed incremental update for source data possible
    - ODI + Golden Gate supported
    - Aggregates still need to be recreated
Essbase - Use Case - Architecture
Source Data

<table>
<thead>
<tr>
<th>Products</th>
<th>Time</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOB</td>
<td>Year</td>
<td>Country</td>
</tr>
<tr>
<td>Brand</td>
<td>Quarter</td>
<td>City</td>
</tr>
<tr>
<td>Product</td>
<td>Month</td>
<td>Street</td>
</tr>
</tbody>
</table>

Source Data

SALES

Saturday, September 29, 12
Required Analysis

- Products: LOB, Brand, Product
- Time: Year, Quarter, Month
- Customer: Country, City, Street
- Sales

T: +44 (0) 8446 697 995 or (888) 631 1410 (USA)  E: enquiries@rittmanmead.com  W: www.rittmanmead.com
Essbase - Typical Usage

- Brand
- Year
- Quarter
- Month
- City
- Country
- LOB
- Product
- Street
- Sales

Essbase In-Memory

Aggregated Data

- Year
- Quarter
- Month

T: +44 (0) 8446 697 995 or (888) 631 1410 (USA)  E: enquiries@rittmanmead.com  W: www.rittmanmead.com

Saturday, September 29, 12
Essbase - Memory Size

1. Hard to Gauge the complete Memory Size
2. Can control the memory settings of BSO Essbase through Index Cache Size & Data Page Cache Size
Essbase - Run-Time Queries

Limited Run-Time Aggregations

- LOB
- Year
- City
- Sales

- LOB
- Year
- Country
- Sales

- Brand
- Month
- Country
- Sales

All Pre-Aggregated Data
Essbase - As in-memory Engine

- Aggregated data stored
  - BSO - Index Cache & Data Cache
  - ASO - Limited Control (not completely in-memory)
- Limited runtime aggregations
- Not straightforward to calculate memory required
  - Can be controlled through the cache sizes
- Size of the in-memory cache
  - Requires careful planning & design
Essbase - Source Data Change

1. Incremental Updates
2. New Data
Essbase - As in-memory Engine

- Whenever Source Data Changes
  - All aggregates automatically aggregated by Essbase
    - Complete control on aggregation
  - Depending on the size of aggregates
    - ASO/BSO - depending on performance requirement - Control
  - Trickle feed incremental update for aggregates
  - Trickle feed incremental update for source data possible
    - Aggregates still need to be recreated
    - But done automatically by Essbase
    - ODI Supported
## Architecture - Summary

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Times Ten</th>
<th>Essbase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Database</td>
<td>Relational</td>
<td>Multi-Dimensional</td>
</tr>
<tr>
<td>Storage Structure</td>
<td>Tables</td>
<td>Index File &amp; Data File (BSO)</td>
</tr>
<tr>
<td>Read from Disk</td>
<td>Only during startup</td>
<td>Disk Retrieval when data not in memory</td>
</tr>
<tr>
<td>Read from Memory</td>
<td>Full read from memory. No disk swap.</td>
<td>Read from Memory, if relevant data available</td>
</tr>
<tr>
<td>Run-Time calculations</td>
<td>Extensive SQL Support</td>
<td>Extensive MDX support</td>
</tr>
<tr>
<td>Incremental Updates</td>
<td>Supported. All tables to be updated separately.</td>
<td>Supported. Automated calculations &amp; aggregations.</td>
</tr>
<tr>
<td>Maintenance Overhead</td>
<td>Limited</td>
<td>More - tuning required</td>
</tr>
<tr>
<td>Compression</td>
<td>Columnar Compression (Exalytics Only)</td>
<td>Bitmap, Zlib, RLE</td>
</tr>
<tr>
<td>Partitioning Support</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Ease of Querying

• Most important factor for in-memory analytical databases
• Times Ten
  ‣ Uses SQL
  ‣ BI EE - primary reporting interface
• Essbase
  ‣ Uses MDX
  ‣ Lot of reporting interfaces
    - Excel - Smart View
    - BI EE
    - Other reporting tools like HFR etc
Ease of Querying

<table>
<thead>
<tr>
<th></th>
<th>New York Actual</th>
<th>Budget</th>
<th>Variance %</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>200</td>
<td>400</td>
<td>100</td>
<td>200</td>
<td>300</td>
<td>400</td>
</tr>
<tr>
<td>Qtr1</td>
<td>Sales</td>
<td>1998</td>
<td>1778</td>
<td>1896</td>
<td>1890</td>
<td>1710</td>
<td>1570</td>
</tr>
<tr>
<td>Qtr3</td>
<td>Sales</td>
<td>2612</td>
<td>1879</td>
<td>2413</td>
<td>2470</td>
<td>1810</td>
<td>1850</td>
</tr>
<tr>
<td>Qtr4</td>
<td>Sales</td>
<td>1972</td>
<td>2293</td>
<td>2010</td>
<td>1720</td>
<td>2180</td>
<td>1540</td>
</tr>
<tr>
<td>First Half Total</td>
<td>Sales</td>
<td>4356</td>
<td>3767</td>
<td>4091</td>
<td>4110</td>
<td>3640</td>
<td>3520</td>
</tr>
<tr>
<td>Second Half Total</td>
<td>Sales</td>
<td>4584</td>
<td>4172</td>
<td>4423</td>
<td>4190</td>
<td>3990</td>
<td>3390</td>
</tr>
<tr>
<td>First/Second Variance</td>
<td>Sales</td>
<td>4584</td>
<td>4172</td>
<td>4423</td>
<td>4190</td>
<td>3990</td>
<td>3390</td>
</tr>
<tr>
<td>Year</td>
<td>Sales</td>
<td>8940</td>
<td>7939</td>
<td>8514</td>
<td>8300</td>
<td>7630</td>
<td>6910</td>
</tr>
</tbody>
</table>

* - Source Google Images
Ease of Querying - Times Ten

• Common Issues with using SQL for building such reports
  ‣ SQL - can become complex
  ‣ Requires a separate tool to build interfaces
    - BI EE (Exalytics)
    - Not possible directly with SQL alone
      - Multiple SQLs to be generated & joined together

• BI EE native support for Times Ten
  ‣ Works really well
  ‣ BI EE PS enhanced
    - Complex Pivot Queries possible
    - Without Performance Issues
Ease of Querying - Times Ten

- Possible for Times Ten to leverage power of MDX
  - Using Essbase XOLAP
  - Times Ten metadata loaded into Essbase outline structure
  - Excel native querying
    - With member selection

Saturday, September 29, 12
Ease of Querying - Essbase

• MDX - Very flexible for common Pivot Type reports
  ‣ Supports Member Selection
  ‣ Supports dynamic calculations
  ‣ Supports axis type queries
    - Easy swap of rows<->columns

• BI EE Support for Essbase
  ‣ Native Support
  ‣ Common known issues
    - Multiple MDXs for 1 report
    - Generates SQL type MDX
BI EE Aggregates Support

- With BI EE 11.1.1.6.2 BP1
  - Aggregates Supported for Times Ten (only Exalytics)
  - Aggregates Supported for Essbase
    - Only ASO supported
    - BSO to be manually created (if needed, for in-memory)
- Times Ten
  - Summary Advisor supported
  - Exalytics Only
- Essbase
  - No Summary Advisor
  - Technically possible to use (license restrictions)
Real Time Loads & Incremental Updates

- **Times Ten - Base Data**
  - Supports load through BI EE Aggregate Persistence Wizard
  - Supports load through ODI & Golden Gate
    - Trickle Feed
  - External Utilities available to load data from Flat files

- **Times Ten Aggregates**
  - Aggregates need to be rebuilt
  - No trickle feed aggregate update
    - Hard to determine the changes and update the aggregates
Real Time Loads & Incremental Updates

- **Essbase - Base Data**
  - Supports load through BI EE Aggregate Persistence Wizard
  - Supports load through ODI
    - Parallel loads & simultaneous multi-thread updates
  - External Utilities available to load data from Flat files
  - EAS/Essbase Studio for external loads

- **Essbase Aggregates**
  - Aggregates need to be rebuilt
  - Native to Essbase
    - Faster than times ten when it comes to aggregate update
Essbase & Times Ten - Use Cases

- Times Ten
  - Very good for DW type environments
  - Aggregate reloads can be tied to ETL process
  - Not suited for
    - Finance type data - ragged hierarchies
    - Source data containing Parent-Child hierarchies
  - For sources that change every hour (on incremental basis)
    - Use Times Ten Federated tables
      - Incremental data in one table
      - Historical data along with aggregates in another set
Essbase & Times Ten - Use Cases

- **Essbase**
  - Very good for environments where source data changes frequently
    - Historical data changes as well
  - Aggregates - natively created - Fast - can be tied to ETL process
  - Not suited for
    - Environments where metadata hierarchies are not unique
    - ASO - Suited for DW style environments
    - BSO - Suited for environments where control on memory is required