Leveraging Oracle Database In-Memory to accelerate Oracle Business Intelligence Analytics Applications

Kai Yu
Oracle Solutions Engineering
Dell EMC
Linkedin: https://www.linkedin.com/in/kaiyu1/
Twitter: @ky_austin1
Kai Yu

- Senior Principal Engineer
- Dell/EMC Oracle Solutions Engineering
- 25+ years working with IT Industry
- Specializing in Oracle Database, Cloud, Virtualization
- Author (35 articles/book) and Speaker (135+ sessions)
- Oracle ACE Director
- 2011 OAUG Innovator of Year
- 2012 Oracle Excellence Award - Technologist of the Year: Cloud Architect by Oracle Magazine
Related Work: Dell Integrated Systems for Oracle Business Analytics - Ready Infrastructure

Diagram showing the components:
- Database Server (R920)
- OBIEE Server (R730)
- Management Server (R320)
- (1 x S60) 1GbE Switch (Management)
- Top of Rack (2 x S6000) 40GbE Switches (Public & Private)
- Oracle Database 12c (2 x R920)
- 2 x Brocade 6510 16Gbps Fibre Channel SAN Switches
- Active/Active DAAD-HA

Additional notes:
- Related Work: Dell Integrated Systems for Oracle Business Analytics - Ready Infrastructure
Agenda

- Oracle 12c In-Memory Database (IMDB) Option
- Oracle 12cR2 IMDB new features
- Oracle Database and BI services in Oracle Cloud
- Leveraging Oracle IM Memory Advisor
- Case Studies of IMDB for Oracle IBEE
- Questions
Oracle 12c In-Memory Option
Oracle 12c In Memory Option

- Oracle 12c Database Introduced Database In-Memory option:
  - Accelerates analytics by orders of magnitude.
  - Speeding up mixed-workload OLTP.
  - Transparent to applications.
- Dual-Format of Architecture in Oracle 12
  - Oracle traditional row based:
    - Row format data stored in storage
    - Row format data stored in buffer cache in SGA
    - Good for OLTP (insert/update/delete) operations
  - Oracle 12c introduced In-memory option
    - Introduced with Oracle 12.1.0.2
    - Column format In-Memory column storage in SGA
    - A New component of Oracle Database SGA.
    - Coexist with database buffer cache (row format)
    - Good for OLAP applications
Oracle 12c In-Memory Option

• The Dual Format Architecture can be illustrated as

• The In-Memory Column Store:
  – A new component called In-Memory Area in SGA
    SQL> alter system set inmemory_size =  100G scope=spfile;

```
SQL> SQL> SQL> startup
ORACLE instance started.
Total System Global Area 2.6521E+11 bytes
Fixed Size 7662672 bytes
Variable Size 2.7380E+10 bytes
Database Buffers 1.2992E+11 bytes
Redo Buffers 529207296 bytes
In-Memory Area 1.0737E+11 bytes
Database mounted.
Database opened.
```

Alter SYSTEM SET INMEMORY_QUERY=DISABLE
Alter SYSTEM SET INMERMORY_QUERY=ENABLE

• Help Analytical processing through reading data from the In memory column store
• Help OLTP by allowing you drop indexes that were created for reporting
Oracle 12c In-Memory Option

- Select contents to populate the In-Memory column store:
  - Tablespace level: alter tablespace data MEMORY;
  - Table level: alter table sales INMEMORY PRIORITY CRITICAL;
  - background process to populate in-memory store:

| oracle 14737 | 1 0 14:30 ? | 00:00:17 ora_w004_pocdb1 |
| oracle 14759 | 1 0 14:30 ? | 00:00:15 ora_w005_pocdb1 |
| oracle 14763 | 1 0 14:30 ? | 00:00:12 ora_w006_pocdb1 |
| oracle 14765 | 1 0 14:30 ? | 00:00:12 ora_w007_pocdb1 |
| oracle 17515 | 1 0 14:38 ? | 00:00:06 ora_w008_pocdb1 |
| oracle 19344 | 1 0 14:43 ? | 00:00:06 ora_w009_pocdb1 |
| oracle 19346 | 1 0 14:44 ? | 00:00:00 ora_w00a_pocdb1 |
| oracle 112632 | 1 0 13:26 ? | 00:00:22 ora_w000_pocdb1 |
| oracle 112634 | 1 0 13:26 ? | 00:00:22 ora_w001_pocdb1 |

- Features to accelerate query execution: In-Memory Scan, In-Memory Storage Index, SIMD Vector Processing, In-Memory Joins, in Memory Aggregation
- In Memory Option: Application transparent, no need to modify application.
- How to determine if In-Memory option takes effect. Look the INMEMORY key word in query plan such as:

```
PLAN_TABLE_OUTPUT
  | 17 | TABLE ACCESS INMEMORY FULL |
  | 18 | PARTITION LIST JOIN-FILTER |
  | 19 | TABLE ACCESS INMEMORY FULL |
  | 20 | PARTITION LIST JOIN-FILTER |
  | 21 | TABLE ACCESS INMEMORY FULL |
  |      | EDAPIHDR_BASE               |
  |      | EDAPILIN_BASE               |
  |      | EDAPIQ_BASE                 |
```
Oracle 12cR2 IMDB New Features
Oracle 12cR2 IMDB New Features

• New Features Summary:
  – **In-Memory Column Store dynamic resizing**
    The size of the In Memory can be dynamically increased without reopening the database.
  – **In-Memory Expressions**
    Frequently used expression for population in the IM column store.
  – **In FastStart**
    Database reads data from the FastStart area and populate IM column store.
  – **Object-level support for service**
    Control the population of an object for the database instances where a service runs.
  – **In column storage on a standby database**
    Enable an IM column store in an Oracle Active Data Guard standby database.
  – **ADO support for the IM column store**
    ADO policies to evict objects from IM column store based on Heat Map statistics.
  – **Join groups**
    List two joined columns and help eliminate the performance overhead of decompressing and hashing column values during the join operation.
Oracle 12cR2 IMDB New Features

• **In-Memory Column Store dynamic resizing**
  Prerequisites: the column store enabled, the comparability level 12.2.0 or higher, db instance started with spfile, new size at least 128M bigger (if smaller, use scope=spfile)
  sqlplus>alter system set inmemory_size = 60000M scope=both

• **In-Memory expressions (IME)**
  “Pre-compute” frequently evaluated expressions
  IME can be created for:
  - Virtual columns
  - Automatic capture
    . Frequently evaluated query expression
    . Other useful internal computation (join hash values, predicate evaluations, data conversion)
  . Reduce computationally expensive repeated evaluations
  . Significant performance increases
  . Example: Select price*Tax_ratio from sales where state=’TX’
Oracle 12cR2 IMDB New Features

• **Identify IM-memory expression**
  
  DBMS_INMEMORY_ADMIN.IM_CAPTURE_EXPRESSIONS identifies “hot” expression, called IM-Memory Expressions (IM Expression)
  
  - auto-detected : hot expression
  
  - One or more columns of a single row if a table, possible some constrains
  
  - Have a 1 to 1 mapping with rows in a table
  
  select employ_name, Round(Salary*12)/52,2) as “weekly_sal
  from employees
  
  Round(Salary*12)/52,2) is frequently and computationally intensive
  a good candidate for IM expression.

• **Populate IM-memory expression**

  The INMEMORY_EXPRESSIONS USAGE determines which type of IM expression is populated:
  
  - Enable, Static_only, Dynamic only, Disable modes.
Oracle 12cR2 IMDB New Features

- **In Memory Virtual Columns**
  - The value on an IM virtual column derived by an expression.
    - Example, in Sales table: `sale_price=price * (1+tax_ratio)`, the value is pre-calculated and is stored in the IM column store to improve the query performance.
  - IM expression and IM virtual column: same underlying mechanism.
    - IM virtual columns are user created and exposed.
    - IM expressions are database created and hidden.
      - Set to Manual: need to explicitly add the column into IM columns store:
        ```sql
        alter table sales add (sale_price AS price * (1+tax_ratio);
        ```
  - Populate virtual columns
    - `INMEMORY_VIRTUAL_COLUMNS = (MANULA, ENABLE, DISABLE)`
  - Example:
    ```sql
    alter table sales add (sale_price AS price * (1+tax_ratio);
    ```
    ```sql
    Alter table set INMEMORY_VIRTUAL_COLUMNS = ENABLE SCOPE=SPFILE;
    ```
Oracle 12cR2 IMDB New Features

• **Join groups**
  
  – The IM columns store enhances the performance of joins when the two join tables are stored in Memory
  
  – Join Group: list two joined columns and help eliminate the performance overhead of decompressing, hashing column values during the join operation
  
  – Create join groups:
    
    Example: create a join group between `part` and `lineitem` on the `partkey` create inmemory join group `jgrp_lo_part(lineitem(l_partkey), part(p_partkey))`

---

Compare the performance with or without join group:

without Join group:

```sql
SELECT /*+ no_inmemory no_vector_transform */
    count(*),
    count(l.l_orderkey),
    count(p.p_type)
FROM Lineitem l, part p
WHERE l.l_partkey = p.p_partkey
AND  1.l_discount > 0;
```

With Join group

```sql
SELECT /*+ no_vector_transform */
    count(*),
    count(l.l_orderkey),
    count(p.p_type)
FROM Lineitem l, part p
WHERE l.l_partkey = p.p_partkey
AND  1.l_discount > 0;
```
Oracle Database and BI Services in Oracle Cloud
Oracle Database Cloud Service in Oracle Cloud

- Oracle Database Cloud service in Oracle Cloud
  - Oracle Database Cloud Service
  - Oracle Exadata Express Service
- Two levels of Oracle Database Cloud Service
  - Virtual Images level: Virtual OS, customers to install Oracle
  - Oracle Database Cloud Service Level
    Oracle Database service already installed. Oracle RAC not supported
    Two Oracle versions supported: 12.1.0.2 and 12.2.0.1: You can try 12.2.0.1 now
Oracle Business Intelligences in Oracle Cloud

- **Oracle Business Intelligences in Oracle Cloud**
  - Offer the full array of intuitive BI tools
  - Intuitive Cloud Experience
    Friendly interactive interface has built-in guidance and tutorials to get users productive quickly
  - Advanced Analysis and Visualizations
    Select interactive visualization and easy create advanced calculations to reveal the insights in your data
  - Interactive Dashboards
    Configurable dashboards that allow you to quick analyze and manage activity across the entire system.
  - Products:
    Business Intelligence Cloud Service
    Oracle Database Schema Service
    Oracle Database Cloud Service
Leverage In-Memory Advisor
Leverage In-Memory Advisor

- **Oracle In-Memory Advisor**
  - Help to answer these questions:
    - Which tables and/or partitions should be marked for In-Memory column store
    - How to size the In memory.
  - An Oracle new feature, licensed as part of the Database Tuning pack
  - MOS note: 1965343.1 Oracle In-Memory Advisor (include `twp_oracle_database_in_memory_advisor.pdf` whitepaper)

- Two whitepapers: Oracle Database In-Memory Advisor and Oracle Database In-Memory Advisor Best practices published in February 2015

- How it works:
  1. Differentiates analytics processing from other database activity based upon SQL plan cardinality, Active Session History (ASH), use of parallel query, and other statistics.
  2. Estimates analytic processing performance improvement factors based upon the following:
     - Eliminating user I/O waits, cluster transfer waits, buffer cache latch waits,
     - Certain query processing advantages related to specific compression types.
     - Decompression cost heuristics per specific compression types.
     - SQL plan selectivity, number of columns in the result set, etc.
Leverage In-Memory Advisor

- Download and Install In-Memory Advisor
  - Download imadvisor.zip from Oracle, copy to DB server and unzip it
    
    ```
    [oracle@inmem1 in_memory] $ ls
    imadvisor_clone_view.sql  imadvisor_DataPump.sql  instimadv.zip
    imadvisor_export.sql  imadvisor_fetch_recommendations.sql  prvtimadv.zip
    imadvisor_clone_view.sql  imadvisor_fetch_temp.sql  prvtimadv.plb
    imadvisor_awr_augment_export.sql  imadvisor_load_report_templates.sql  schmimadv.sql
    imadvisor_awr_augment_import.sql  imadvisor_spool_debug.sql
    imadvisor_awr_augment_tables.sql  imadvisor_version.sql
    dbmsimadv.sql  instimadv.sql
    imadvisor_analyze_and_report.sql  prvtimadv.plb
    dbmsimadv.plb
    ```

  - Installed in SQLPLUS with sysdba privilege
    
    SQL> @instimadv.sql
    
    - Do you currently have a valid Oracle Tuning Pack license with this database (Y/N)?
    - Create a new user called IMADVISOR and schema
    - Create DBMS_INMEMORY_ADVISOR package
    - Need to provide the connection string (from TNSNAME entry)
    - Provide the Oracle directory object IMADVISOR_DIRECTORY directory that In-Memory Advisor uses
    - Need to specify the users that will use this tool for tuning:
    - It will GRANT EXECUTE ON DBMS_INMEMORY_ADVISOR to the users
    - You can add more users by granting EXECUTE ON DBMS_INMEMORY_ADVISOR to additional users later
Leverage In-Memory Advisor

- Running In-Memory Advisor
  - Run script `imadvisor_analyze_and_report.sql` as a user with the privilege to execute the `DBMS_INMEMORY_ADVISOR` package:
    ```sql
    SQL> @imadvisor_analyze_and_report
    ```
  - Specify the IM task name
  - The IM Advisor generates a report as `imadvisor_<taskname>.html` file in the current working directory
  - The sql file is generated as `imadvisor_sql_<taskname>.sql`
  - Enter value for `im_task_name`: test
  - IM Task name Specified: test
  - Enter begin time for report: ...
  - Enter value for `begin_time`: -1:30
  - Report begin time specified: -1:30
  - Enter duration in minutes starting from begin time:
  - Defaults to `SYSDATE - begin_time`
  - Enter value for duration: 60
  - Report duration specified: 60

  Using 2016-Jan-14 09:33:13.000000000 as report begin time
  Using 2016-Jan-14 10:33:13.000000000 as report end time

  IM Advisor: Adding Statistics..
  IM Advisor: Adding Statistics..
  IMADVISOR: Finished Adding Statistics
  IMADVISOR: Finished Executing the task
  IM Advisor: Generating Recommendations..

  `imadvisor_cmpldaad.html`
  `imadvisor_sql_cmpldaad.html`
  `imadvisor_object_cmpldaad.html`
Leverage In-Memory Advisor

- Output of In-Memory Advisor
  - `imadvisor_taskname.html`
  - Summary of the total database time analyzed
  - Percentage for Database Time for Analytics Processing
  - In-Memory sizes vs the estimated benefit

<table>
<thead>
<tr>
<th>Percentage of Maximum Recommended In-Memory Size</th>
<th>Percentage of Current SGA Size (39GB)</th>
<th>In-Memory Size</th>
<th>Estimated Analytics Processing Time Reduction (Seconds)</th>
<th>Estimated Analytics Processing Performance Improvement Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>116%</td>
<td>45GB</td>
<td>4013</td>
<td>2.9X</td>
</tr>
<tr>
<td>95%</td>
<td>110%</td>
<td>43GB</td>
<td>1562</td>
<td>1.3X</td>
</tr>
<tr>
<td>90%</td>
<td>104%</td>
<td>41GB</td>
<td>1562</td>
<td>1.3X</td>
</tr>
</tbody>
</table>

- Recommending the top objects to place in memory and compression type and estimated benefit

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Object</th>
<th>Compression Type</th>
<th>Estimated In-Memory Size</th>
<th>Analytics Processing Seconds</th>
<th>Estimated Reduced Analytics Processing Seconds</th>
<th>Estimated Analytics Processing Performance Improvement Factor</th>
<th>Benefit / Cost Ratio (Reduced Analytics Processing/In-Memory Size)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>CMPLUSER.DISTRICT</td>
<td>Memory compress for query low</td>
<td>1MB</td>
<td>110</td>
<td>81</td>
<td>3.8X</td>
<td>4489 : 1</td>
</tr>
<tr>
<td>TABLE</td>
<td>CMPLUSER.ORDERS</td>
<td>Memory compress for query low</td>
<td>2GB</td>
<td>1823</td>
<td>1481</td>
<td>5.3X</td>
<td>42 : 1</td>
</tr>
<tr>
<td>TABLE</td>
<td>CMPLUSER.STOCK</td>
<td>Memory compress for query low</td>
<td>36GB</td>
<td>3314</td>
<td>2451</td>
<td>3.8X</td>
<td>4 : 1</td>
</tr>
</tbody>
</table>
Oracle Exalytics In-Memory vs 12c In-Memory Database
Oracle Exalytics In-Memory Machine

- Oracle Engineered System for Extreme Analytics: Delivers extreme in-memory analytics performance, two main components together
  - Optimized Oracle Business Intelligence Foundation Suite
  - Oracle TimesTen In-Memory Database for Exalytics
Oracle Exalytics In-Memory Machine

• Oracle Exalytics In-Memory Machine features
  – Single X86-64 server: 4 X Intel Xenon E7-4800 processors, 2 TB RAM, 2 QDR 40Gb/s Infiniband Ports, 2X 10Gbps Ethernet ports, 6 X 400G Flash PCI-e
  – Oracle Business Intelligence Foundation Suite including Oracle Essbase
  – Oracle TimesTen In-Memory Database for Exalytics
  – Exalytic In-Memory Software

• Difference between TimesTen In-Memory Database vs Oracle 12c In-Memory
  – TimesTen In-Memory Database for Exalytics is a full memory database designed to run Analytics.
  – TimesTen In-Memory Database runs on the same server as OBIEE
  – Tightly connected between BI and TimesTen In-Memory Database
  – Oracle 12c In-Memory is a feature added to Oracle Database
  – Oracle 12c In-Memory works for both OLAP and OLTP mixed workloads
Oracle OBIEE with Oracle 12c In-Memory Database
Oracle Business Intelligence Enterprise edition 11g

- Oracle OBIEE: Business intelligence and *Analytics Platform and* common infrastructure for reports, scorecards, dashboards, ad-hoc analysis, OLAP analysis
  - OBIEE 11g Interactive Dashboards solution for Interactive Dashboards
  - Ad hoc Analysis and Interactive Reporting
  - Oracle BI Mobile for Mobile Analytics
Oracle OBI EE with Oracle 12c In-Memory Database

- Oracle BI server Architecture
  - Oracle BI server connects to Oracle Database through ODBC/JDBC
  - Oracle BI present a logic schema view independent of physical database
  - BI server translates the logic SQL to physical SQL
  - Oracle BI Administration tools shows the three layers: Presentation, Business Model and Mapping, Physical
POC: Accelerates BI with Oracle 12c In-Memory

• Basic Idea:
  – On physical level BI reports usually involve a large full table scan and complex join operation.
  – Full table scan operation is very expensive in large storage IO operation.
  – Load the partially or the entire table to In-Memory store to reduce the storage IO for the full table scan.

• How to identify the tables to load into In Memory store: in manual way
  – Start with the slow report and find the presentation layer the report reads.
  – Through the mapping from presentation layer to the physical layer to identify the physical SQL for the report.
  – Through the physical SQL to identify the underneath full table scan operation.

  The rest presentation use the EDI Queue report as an example to use the process.
An Example: Accelerate BI Report with Oracle 12c

- Identify Physical SQL layer for the report:
  - From the Dashboard report definition to identify the presentation layer Fact EDI Queue.
  - Through the presentation layer to find the Business Model and mapping on Fact EDI Queue and identify the physical database view: FACT_EDI_QUEUE_V as shown below:
An Example: Accelerate BI Reports with Oracle 12c

• Review the definition of the physical View:
  – View name: FACT_EDI_QUEUE_V and found underneath physical tables
  – Identified four large tables:
    EDAPIHDR_BASE, EDAPIQ_BASE, EDAPIQ_BASE
    VEN_LOC_BASE

• Populate In-Memory Column store with these four tables:
  – SQL> alter table APD_BASE.EDAPILIN_BASE inmemory priority high;
  – SQL> alter table APD_BASE.EDAPIQ_BASE inmemory priority high;
  – SQL> alter table APD_BASE.EDAPIHDR_BASE inmemory priority high;
  – SQL> alter table APD_BASE.VEN_LOC_BASE inmemory priority high;

• Check size of the segments in the In-Memory
  SQL> select SEGMENT_NAME, INMEMORY_SIZE from v$im_segments;

<table>
<thead>
<tr>
<th>SEGMENT_NAME</th>
<th>INMEMORY_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VEN_LOC_BASE</td>
<td>1279648</td>
</tr>
<tr>
<td>EDAPIQ_BASE</td>
<td>291168512</td>
</tr>
<tr>
<td>EDAPIHDR_BASE</td>
<td>961496576</td>
</tr>
<tr>
<td>VEN_LOC_BASE</td>
<td>1279648</td>
</tr>
<tr>
<td>EDAPILIN_BASE</td>
<td>930710528</td>
</tr>
<tr>
<td>VEN_LOC_BASE</td>
<td>1279648</td>
</tr>
</tbody>
</table>
An Example: Accelerates BI Reports with Oracle 12c

• Compare the Query plans on: FACT_EDI_QUEUE_V
  Not In-Memory
  ![Query Plan 1](image1)
  ![Query Plan 2](image2)

  ![Query Plan 3](image3)
An Example: Accelerate BI Reports with Oracle 12c

- Compare the Query plans on FACT_EDI_QUEUE_V

**Not In-Memory**

```sql
SQL> explain plan for select count(*) from APD_STAR.FACT_EDI_QUEUE_V;

explained.

SQL> select plan_table_output
2  from table(dbms_xplan.display('plan_table',null,'basic'));

PLAN_TABLE_OUTPUT
plan hash value: 1571279316

<table>
<thead>
<tr>
<th>id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SORT AGGREGATE</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>HASH JOIN</td>
<td>FACT_EDI_QUEUE_V(Not In-Memory)</td>
</tr>
<tr>
<td>3</td>
<td>PART JOIN FILTER CREATE</td>
<td>:BF0000</td>
</tr>
<tr>
<td>4</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TABLE ACCESS FULL</td>
<td>DIM_PERIOD</td>
</tr>
</tbody>
</table>

PLAN_TABLE_OUTPUT

<table>
<thead>
<tr>
<th>id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TABLE ACCESS FULL</td>
<td>DIM_REGION_CCN</td>
</tr>
<tr>
<td>8</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PART JOIN FILTER CREATE</td>
<td>:BF0001</td>
</tr>
<tr>
<td>10</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PART JOIN FILTER CREATE</td>
<td>:BF0002</td>
</tr>
<tr>
<td>12</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>TABLE ACCESS FULL</td>
<td>DIM_VENDOR</td>
</tr>
<tr>
<td>14</td>
<td>PARTITION LIST ALL</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>TABLE ACCESS FULL</td>
<td>VEN_LOC_BASE</td>
</tr>
<tr>
<td>16</td>
<td>PARTITION LIST JOIN-FILTER</td>
<td></td>
</tr>
</tbody>
</table>

PLAN_TABLE_OUTPUT

<table>
<thead>
<tr>
<th>id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>TABLE ACCESS FULL</td>
<td>EDAPIHDR_BASE</td>
</tr>
<tr>
<td>18</td>
<td>PARTITION LIST JOIN-FILTER</td>
<td>EDAPIHDR_BASE</td>
</tr>
<tr>
<td>19</td>
<td>TABLE ACCESS FULL</td>
<td>EDAPITN_BASE</td>
</tr>
<tr>
<td>20</td>
<td>PARTITION LIST JOIN-FILTER</td>
<td>EDAPITN_BASE</td>
</tr>
<tr>
<td>21</td>
<td>TABLE ACCESS FULL</td>
<td>EDAPIQ_BASE</td>
</tr>
</tbody>
</table>
```

**In-Memory**

```sql
SQL> explain plan for select count(*) from APD_STAR.FACT_EDI_QUEUE_V;

explained.

SQL> select plan_table_output
2  from table(dbms_xplan.display('plan_table',null,'basic'));

PLAN_TABLE_OUTPUT
plan hash value: 1571279316

<table>
<thead>
<tr>
<th>id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SELECT STATEMENT</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SORT AGGREGATE</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>HASH JOIN</td>
<td>FACT_EDI_QUEUE_V(In-Memory)</td>
</tr>
<tr>
<td>3</td>
<td>PART JOIN FILTER CREATE</td>
<td>:BF0000</td>
</tr>
<tr>
<td>4</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TABLE ACCESS FULL</td>
<td>DIM_PERIOD</td>
</tr>
</tbody>
</table>

PLAN_TABLE_OUTPUT

<table>
<thead>
<tr>
<th>id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TABLE ACCESS FULL</td>
<td>DIM_REGION_CCN</td>
</tr>
<tr>
<td>8</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PART JOIN FILTER CREATE</td>
<td>:BF0001</td>
</tr>
<tr>
<td>10</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PART JOIN FILTER CREATE</td>
<td>:BF0002</td>
</tr>
<tr>
<td>12</td>
<td>HASH JOIN</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>TABLE ACCESS FULL</td>
<td>DIM_VENDOR</td>
</tr>
<tr>
<td>14</td>
<td>PARTITION LIST ALL</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>TABLE ACCESS INMEMORY FULL</td>
<td>VEN_LOC_BASE</td>
</tr>
<tr>
<td>16</td>
<td>PARTITION LIST JOIN-FILTER</td>
<td></td>
</tr>
</tbody>
</table>

PLAN_TABLE_OUTPUT

<table>
<thead>
<tr>
<th>id</th>
<th>Operation</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>TABLE ACCESS INMEMORY FULL</td>
<td>EDAPIHDR_BASE</td>
</tr>
<tr>
<td>18</td>
<td>PARTITION LIST JOIN-FILTER</td>
<td>EDAPIHDR_BASE</td>
</tr>
<tr>
<td>19</td>
<td>TABLE ACCESS INMEMORY FULL</td>
<td>EDAPITN_BASE</td>
</tr>
<tr>
<td>20</td>
<td>PARTITION LIST JOIN-FILTER</td>
<td>EDAPITN_BASE</td>
</tr>
<tr>
<td>21</td>
<td>TABLE ACCESS INMEMORY FULL</td>
<td>EDAPIQ_BASE</td>
</tr>
</tbody>
</table>
```
An Example: Accelerate BI Reports with Oracle 12c

• Compare the Query plans execution time on FACT_EDI_QUEUE_V
  Not In-Memory

```
SQL> set timing on
SQL> set time on
12:35:36 SQL> select count(*) from APD_STAR.FACT_EDI_QUEUE_V;

COUNT(*)  
---------
51638519
Elapsed: 00:04:02.80
```

In-Memory

```
14:51:55 SQL> select count(*) from APD_STAR.FACT_EDI_QUEUE_V;

COUNT(*)  
---------
51638519
Elapsed: 00:02:15.11
```

• Compare the Dashboard report execution:
  Not In-Memory: 9 minutes 31 seconds
  In-Memory: 7 minutes 50 seconds

Next step: Test the performance gains by using the join group in 12cR2
Case Study 2: Use IMDB for Business Analytics Apps

- POC Background
  Dell Statistica Analytic Application

  Database queries: form dataset by querying 32 columns of 100M rows in a single select statement and computed various stats with these columns:

  Example:

<table>
<thead>
<tr>
<th>statement</th>
<th>Cost without In Memory</th>
<th>Cost with In memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>8781</td>
<td>474</td>
</tr>
<tr>
<td>S2</td>
<td>8781</td>
<td>474</td>
</tr>
<tr>
<td>S3</td>
<td>8762</td>
<td>357</td>
</tr>
<tr>
<td>S4</td>
<td>9084</td>
<td>754</td>
</tr>
</tbody>
</table>

  Not much difference in Elapsed time:

  Why: Query Statistics on in memory: huge number of data sent on network
  17554520327 bytes sent via SQL*Net to client
  73333877 bytes received via SQL*Net from client
  6666668 SQL*Net roundtrips to/from client
  CPU cost comparisons of four major queries:
Case Study 2: Use IMDB for Business Analytics Apps

• Example 2: Statistics Aggregation/Computation on large data set

Took the dataset (100M rows x 32 columns) and computed various stats for columns in a single select statement with in-memory option on/off (table was configured for parallel execution). The results are as follows:

<table>
<thead>
<tr>
<th>Stats computed for 32 columns</th>
<th>No In-memory</th>
<th>In-memory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>Cost</td>
</tr>
<tr>
<td>Sum</td>
<td>6.313</td>
<td>8781</td>
</tr>
<tr>
<td>Sum Avg</td>
<td>6.328</td>
<td>8781</td>
</tr>
<tr>
<td>Sum Avg Count</td>
<td>6.266</td>
<td>8781</td>
</tr>
<tr>
<td>Sum Avg Count StdDev</td>
<td>19.564</td>
<td>8781</td>
</tr>
<tr>
<td>StdDev</td>
<td>14.314</td>
<td>8781</td>
</tr>
</tbody>
</table>

Why:

For more complex aggregations like STDDEV, all of the data in the column is scanned, decompressed and sent to SQL execution layer where the STDDEV calculation is conducted. STDDEV calculation takes more time than scanned, decompressed, majority of the spend on STDDEV. The time saving by In memory is not significant compared the elapsed time for STDDEV operation.
How to get most benefits from IMDB

IMDB speeds up data access for Business anaclitic application
. Not to improve data transfer on Network between BA servers and DB server
. Not to improve the data processing
. Data Load such as ETL, staging data
. Complex PL/SQL, procedures and functions

Good Use Cases:
. OLTP; real-time reporting on OLTP data, Reduce extra indexes for reporting
And improve the OLTP performance
Thank You and QA
Contact me at kai_yu@dell.com or visit my Oracle Blog at http://kyuoracleblog.wordpress.com/