A journey from SAP HANA to Oracle database In-Memory

Kirill Loifman
adidas Group AG
Herzogenaurach, Germany

Keywords

Oracle, database, In-Memory, SAP, HANA, migration, performance

Introduction

Oracle database In-Memory (IM) - what's in it for me, for my application and for my company? How to achieve the maximum value of Oracle In-memory? How to migrate from different platform? We explore these topics based on a migration project from SAP HANA to Oracle database In-Memory. The presentation will cover architectural comparison between SAP HANA and Oracle Database In-Memory, best practices of Oracle In-Memory, data migration techniques, data model adjustments. At the end we will review test results and make conclusions. The presentation will follow the project workflow:

Oracle In-memory - why to use it?
Project goal and scope
Functional tests – does it work?
Build a comparable test environment
Data and data model migration
Performance and acceptance tests
Oracle IM best practices and useful SQL
Project test result and lessons learned

The goal of this draft document is a high level introduction to the presentation topics. All the details will be provided in the presentation itself accompanied with detail procedures.

Oracle In-memory - why to use it?

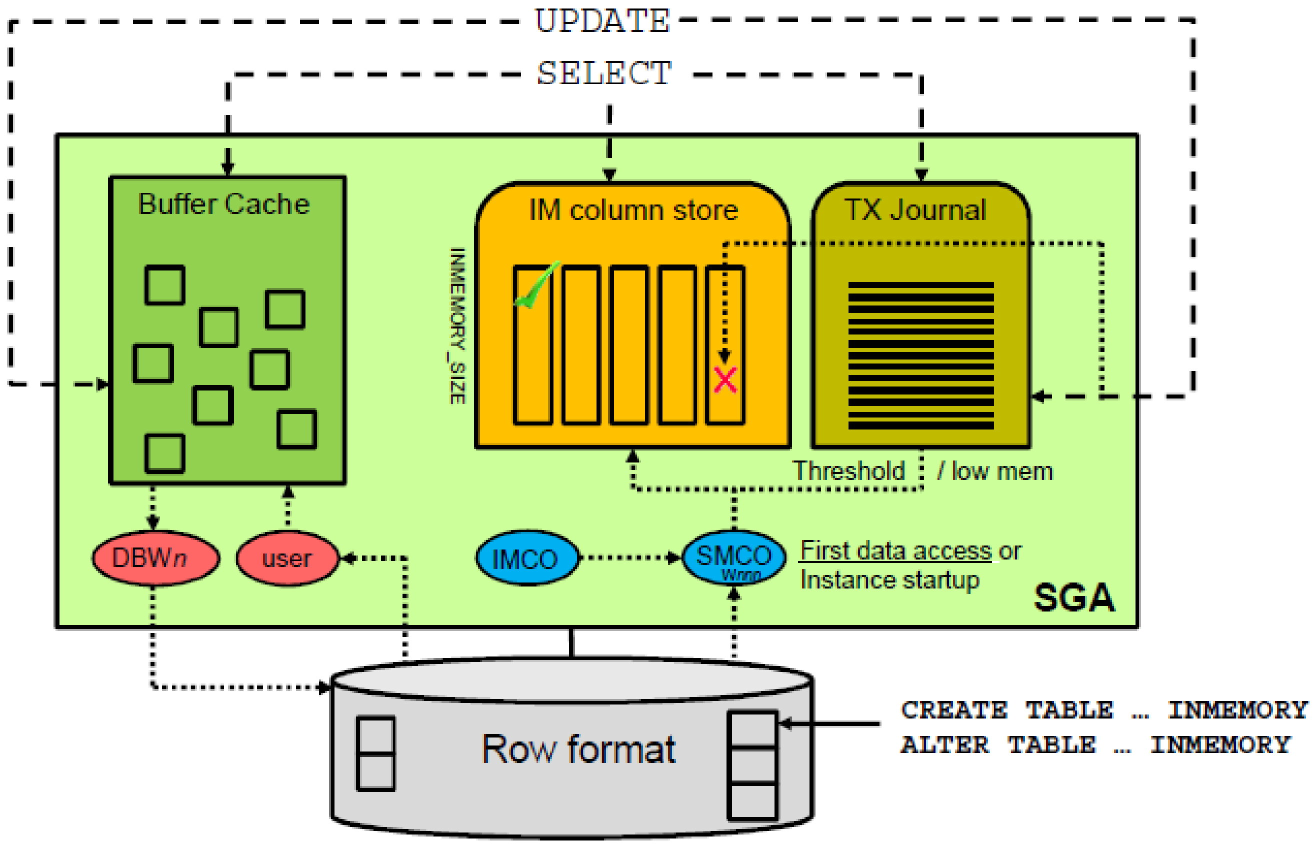
Why do I need Oracle In-Memory option? We cache anyway almost everything in my Oracle database. I have my large Buffer Cache and I can additionally use following technics to improve the database caching:

- KEEP cache & _small_table_threshold parameter

- 12c Big Table Caching

- 12c Full DB-Caching

The key point of Oracle IM is not "What to cache" but "How". The Oracle In-Memory Column Store enables individual database objects to be loaded into memory in the compressed columnar format. This technic enables scans, joins, and aggregates to perform much faster than the traditional on-disk formats, providing fast analytical and reporting performance for both OLTP and DW environments. See below high level Oracle IM architecture diagram.



When to use Oracle IM store? Below I recommend a simple decision tree to start considering Oracle IM:

1. If your end-users have a “real” performance problem

2. If your performance problem is query related

3. If it can not be solved with “classic” database tuning

4. It’s mostly useful for analytic and reporting applications/queries that operate on few columns returning many rows rather than for OLTP operating on few rows returning many columns.

5. Useful not only for huge systems. Smaller systems can get independent from I/O.

6. The performance benefit can outperform the additional Oracle IM license costs.

SAP HANA – the key differences

will be presented at DOAG

Project goal and scope

The project goal was to move a particular application from SAP HANA platform to Oracle IM store. Since this was a pilot we decided to shape the project scope on the way based on the test results at every project step.

Functional Tests

Before diving into the project itself we did some functional Oracle IM tests on a sandbox. We created manually a few simple use cases (objects and reporting queries) on Oracle side based on the current application data model. Then loaded initial data from HANA to Oracle and run some tests. And indeed we could do it quickly and the tests results were promising (more details in the presentation). Below is our Sandbox configuration and a procedure how to enable Oracle IM feature on your regular Oracle EE 12.1.0.2+ instance:

- Sandbox: CPU: 2 Sockets / 12 cores / X5650 @ 2.67GHz / RAM: 186Gb

- Software: OEL 6.6 (150G HugePages) / Oracle database server EE 12.1.0.2 + PSU 12.1.0.2.160119

- Ensure DB minimum compatibility: COMPATIBLE = 12.1.0.0.0

- Adjust DB parameters:

```
ALTER SYSTEM SET sga_target=100000M SCOPE=SPFILE;
ALTER SYSTEM SET inmemory_size=60000M SCOPE=SPFILE;
ALTER SYSTEM SET inmemory_max_populate_servers=6;
ALTER SYSTEM SET parallel_max_servers=64;
ALTER SYSTEM SET use_large_pages=ONLY SCOPE=SPFILE;
```

- redo logs: 2G / disable Audit / gather System stats

Setup comparable test environment

After an analysis of the target HANA platform and the application candidate we prepared the following Big test box for our project.

- HW: HP DL580 Gen9 Intel® Xeon® E7-8880v3 (4 sockets/18-core/2.3GHz)
  RAM: 2TB
  800GB SAS WI 2.5in SC SSD (for temp tablespace)
  NAS = 4TB (direct NFS mount)
  4 NICs
  Important: BIOS adjustments required (more details in the presentation)

- Create File systems:
  - Oracle binaries on local disk
  - Temp TS on local SSD mirrored

- oradata on NAS storage

- Install Oracle EE Server 12.1.0.2

- Important: Patch DB with the latest Bundle Patch (not only PSU)

- Enable Oracle direct NFS

- Enable Linux HugePages

- Create DB and adjust DB parameters:

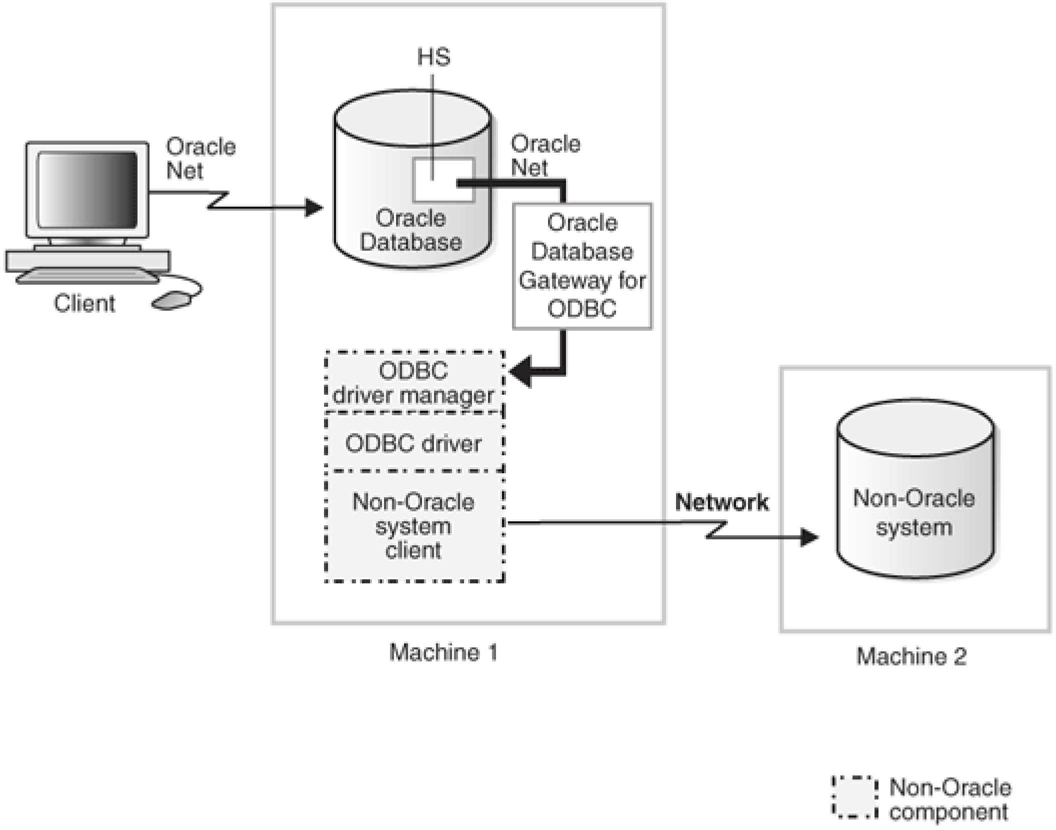
```
ALTER SYSTEM SET sga_target=1700G SCOPE=SPFILE;
ALTER SYSTEM SET pga_aggregate_target=100G;
ALTER SYSTEM SET pga_aggregate_limit=120G;
ALTER SYSTEM SET inmemory_size=1500G scope=spfile;
ALTER SYSTEM SET inmemory_max_populate_servers=32;
ALTER SYSTEM SET use_large_pages=ONLY scope=spfile;
```

Database model migration

After some analysis there was a project decision adjust the data model. See details in the presentation.

Data load

One of the requirements of the project was to have continuous data loads from SAP HANA to Oracle IM DB. After evaluating some options such as CSV file export/import, ETL tools, SAP SLT, Data Services, etc, we opted for Oracle ODBC Gateway. Oracle Gateway allows establishing a direct connection to non-Oracle data stores. Yet another advantage is that it’s included in the DB license. See below the Oracle ODBC Gateway architecture



In our case *Non-Oracle system* is SAP HANA environment and the *Non-Oracle system client* is HANA client. The complexity of connecting HANA and Oracle via Oracle Gateway was in the lack of documentation on that. With some research and many attempts we found the way. See below a short procedure (more details in the presentation):

Step 1: Install unixODBC drivers

Step2: Install SAP HANA Client

Step3: Install Oracle Gateway for ODBC

Step4: Configure Oracle gateway for ODBC

Step5: Point the gateway to SAP HANA DB

Step6: Configure and Start Oracle Net Listener for the Gateway

Step7: Setup connection from Oracle DB to Gateway listener

Step8: Create database link on Oracle DB to HANA

During our testing a few issues occurred during data transfers. One of them is a SQL predicate truncation for NVARCHAR columns on HANA side. We could find a workaround by using bind variables (more details in the presentation).

Populate DB objects into Oracle IM store

Oracle IM store enables you to populate individual objects (not necessary the entire DB) into Oracle IM cache. Every existing on-disk data format for permanent heap tables, individual columns and all object types (partitions, materialized views, materialized join views, materialized view logs, and inline LOBs) are supported by IM column store.

There are few steps required to achieve that.

1) Identifying required objects for IM store

There are different options to identify best candidates for IM store:

- Based on application logic and developers experience:
    - Very hot data, large segments (not less than 1MB)
    - Segments frequently accessed by analytical queries, operating on many rows returning few columns
- Based on Oracle statistics for segments, column scans, AWR
- Based on Oracle IM Advisor: command line (Doc ID 1965343.1) or OEM 12c-13c

Note, Oracle will use still Buffer Cache based on CBO decisions, usually for:

a) segments accessed by OLTP style queries, operating on few rows returning many columns

b) Fetch-by-rowid

c) DML

d) Queries with 1 row processing

2) Define DB objects to be loaded in IM store

There are 2 important aspects in defining DB objects for IM population:

- compression level
- population priority

Below are a few examples how to create objects candidates for Oracle IM store:

```
CREATE TABLE large_tab (c1 …) INMEMORY;

ALTER TABLE | PARTITION  …  INMEMORY;

ALTER TABLE sales NO INMEMORY;

CREATE TABLE countries_part ... PARTITION BY LIST ..
```

```
( PARTITION p1 .. INMEMORY, PARTITION p2 .. NO INMEMORY);

CREATE TABLE myimtab (c1 NUMBER, c2 CHAR(2), c3 DATE)
INMEMORY NO INMEMORY (c1);

ALTER TABLE tab INMEMORY MEMCOMPRESS FOR CAPACITY LOW (c2, c3)
NO INMEMORY (c1, c4);

ALTER TABLE myimtab NO INMEMORY (c2);

CREATE TABLE test (…) INMEMORY PRIORITY CRITICAL;
```

Notes:

- It is recommended to enable all columns for IM at once rather than using ALTER TABLE for each column, as it is more efficient.

- Changing the compression level of columns with the ALTER TABLE statement results in an update of the dictionary, but does not force a repopulation of existing data into the IM column store

- General recommendation is to use compression level: FOR QUERY LOW. Why? See example…

3) Populate objects in IM store

New background processes IMCO, SMCO, Wnnn populate tables in IM store fully (or partially if there is not enough space) at the following occasions:

- On database startup (by default)

- With delays based on population priorities (CRITICAL, LOW)

- At first object access (populated on demand; usually when priority=NONE)

- Manual object population (see below):

```
BEGIN
  DBMS_INMEMORY.POPULATE(schema_name => user, table_name => 'SALES');
  DBMS_INMEMORY.POPULATE(schema_name => user, table_name => 'SALES_CACHE');
  DBMS_INMEMORY.POPULATE(schema_name => user, table_name => 'CUSTOMERS');
  DBMS_INMEMORY.POPULATE(schema_name => user, table_name => 'PRODUCTS');
  DBMS_INMEMORY.POPULATE(schema_name => user, table_name => 'TIMES');
END;
/
```

Another SQL to see what tables are populated in the IM store:

```
SELECT owner, table_name, inmemory, inmemory_compression, inmemory_priority
priority, inmemory_distribute RAC, inmemory_duplicate DUP
FROM dba_tables
WHERE inmemory = 'ENABLED'
/
```

More topics will be presented

- Oracle IM best practices and useful SQL

- Project test results and lessons learned

Contact address:

Kirill Loifman
Olympia Ring 100
91074, Herzogenaurach

Phone: +49(0)9132 84 2541
Email : dadbm1@gmail.com
Internet: dadbm.com
twitter.com/loifmkir
facebook.com/dadbm
google.com/+dadbm
youtube.com/dadbman