



Using SAP NetWeaver with Oracle Database In-Memory

Key Guidelines

ORACLE WHITE PAPER | AUGUST 2015



ORACLE®



Table of Contents

Preface	3
Prerequisites	4
Restrictions	5
Index Design	5
Database Buffer Cache	5
SAP Dictionary Support	5
SAP BW Aggregates	5
Approach 1: Oracle Database In-Memory Advisor	6
Prerequisites	6
Getting the IM Advisor Package and Documentation	7
Installation	7
Using IM Advisor	7
AWR Statistics and SQL Tuning Set	7
Running IM Advisor	9
Transport of AWR Statistics, SQL Tuning Set and IM Advisor Augment Data from your Live System to another System for further Analysis	9
IM Advisor Parameters with SAP	10
Fetching and refining IM Advisor results	10
Load identified SAP tables into the IM Column Store	12
Check Status of SAP tables in the IM Column Store	12
Approach 2: Knowing which SAP table(s) to be placed into the IM Column Store	14



Installation of SAP_IM_ADV package	14
How to use SAP_IM_ADV package	14
Recommended Database In-Memory related init.ora Parameters	17
References	18
SAP Notes	18
MOS Notes	18
White Papers	18
Documentation	18



Preface

This document describes mandatory steps for using Oracle Database In-Memory with SAP Applications and SAP Solutions based on SAP NetWeaver Technology. Therefore this white paper is valid for SAP Products like SAP BW, SAP ERP, SAP CRM or SAP XI or SAP solutions like SAP IS-U, SAP IS-Banking or SAP IS-Retail. Oracle Database In-Memory is certified and supported for both Single Instance and Real Applications Clusters (RAC) on all UNIX/Linux, Windows and Oracle Engineered platforms certified by SAP using SAP Products based on SAP NetWeaver Technology.

Two approaches are described on how to organize SAP tables and table partitions in the In-Memory Column Store (IM column store). The first approach provides a systematic way to identify SAP tables and table partitions which are useful to be placed in the IM column store. It provides you with information whether the SAP application can benefit from using Database In-Memory. The second approach provides you with information on how much additional memory you need on your system for placing a specific SAP table/partition or a set of SAP tables/partitions into the IM column store.

Guidelines and recommendations given in this document are SAP specific and may differ from guidelines and recommendations given in other Non-SAP specific or general purpose papers.



Prerequisites

To use Oracle Database In-Memory with SAP NetWeaver the following technical and business prerequisites must be met:

- Oracle Database 12c Release 1 Patch Set 1 (12.1.0.2) plus
 - UNIX/Linux: Oracle Database SAP Bundle Patch June 2015 (SAP1202P_1506) or newer
 - Strongly Recommended Oracle Database SAP Bundle Patch August 2015 (SAP1202P_1508)
 - Windows: Windows DB Bundle Patch 12.1.0.2.6 or newer
 - Strongly Recommended Windows DB Bundle Patch 12.1.0.2.8
- SAP NetWeaver 7.x Version with minimum SAP Kernel 7.21_EXT
- Proper Licensing of Oracle Database In-Memory Option
 - For Oracle Full Use Customers the license for the Database In-Memory Option is required
 - For ASFU Customers (Customers who have licensed Oracle Software through SAP) the license from SAP for the “Oracle In-Memory License Package” is required
- Oracle Database In-Memory (IM) Advisor
 - Requires Licenses of Oracle Tuning and Diagnostics Pack for Oracle Full Use Customers if Database In-Memory option has been deployed. Free of charge for ASFU Customers.

SAP Note 2178980 will be updated on a regular base with future changes. Always check SAP Note 2178980 before using Database In-Memory with SAP NetWeaver based products.

Restrictions

Index Design

It is not allowed to make any changes to the standard index design of the SAP installations. SAP Applications highly depend on indexes to deliver good throughput and response times. Therefore it is not supported to drop any index of the standard SAP index design or set any standard SAP index invisible or unusable. Therefore it is forbidden to make any changes to indexes which are not part of the Y or Z index namespaces (these namespaces are reserved for customer specific changes). Violation of this rule will create an unsupported SAP installation.

SAP customers are allowed to make changes to their customer specific index design i.e. all indexes which belong to the Y or Z namespaces can be changed.

Examples:

- Index BSIS~0 (or any primary key index) may NOT be dropped or set invisible/unusable as it is a standard index.
- Index GLFUNCA~2 may NOT be dropped or set invisible/unusable as it is a standard index.
- Index VBAP~Z03 may be dropped or set invisible/unusable as it is a custom index (namespace Z (~Z)) and not a standard index.

Individual Columns

It is not supported to load individual columns of an SAP table or partition into the IM column store. It is also not supported to exclude individual columns from an SAP table or partition from the IM column store. An SAP table is a database table used by an SAP application.

All columns of an SAP table or partition must be loaded into the IM column store.

Database Buffer Cache

It is not allowed to reduce the size of the database buffer cache and assign the memory to the IM column store. When using Database In-Memory additional memory is needed on the system to accommodate both the database buffer cache and the IM column store.

SAP Dictionary Support

Full SAP Dictionary (DDIC) Support of in-memory attributes at the table level starts with the support package SAP_BASIS 7.40 SP12. When using older SAP kernels in-memory attributes will be lost during system copies, upgrades, table reorganization or other SAP maintenance operations.

SAP BW Aggregates

It is not recommended to drop all aggregates from an existing BW system as aggregates are designed for providing good throughput in concurrent user environments.

Approach 1: Oracle Database In-Memory Advisor

The Oracle Database In-Memory Advisor (IM Advisor) is a tool that tries to provide you with recommendations which database tables could be placed into the IM column store and tries to estimate the performance benefits of using the IM column store.

The IM Advisor relies on data stored in the automatic workload repository (AWR) as well as on some heuristics. For better results it also allows the use of captured SQL tuning sets.

For SAP applications it is strongly recommended to use a reasonable time window of collected AWR data. So at least 2-3 days of AWR data should be used for the IM Advisor. It absolutely makes no sense to use data from a 1-2 hour time window.

For SAP applications the more AWR data is used for the IM Advisor the better the recommendations of the IM Advisor will be as in general the SAP workload changes during different times of the day. For instance during the night batch jobs are running which do not run during the day generating different database workloads. Also AWR data during month-end closing, year-end closing or fiscal quarter/year closing should be considered as well.

Note that the IM Advisor has no knowledge about SAP specific tables where in-memory placement is not useful. Therefore additional steps are required which are described later in this paper to define the in-memory table set when running SAP applications.

Prerequisites

The database where you want to run IM Advisor must have XDB component installed as IM Advisor relies on functions provided by XDB.

In Oracle Database 12c XDB is installed by default. In Oracle Database 11g Release 2 Version 11.2.0.3 or 11.2.0.4 XDB may have to be installed if not already available before installing IM Advisor.

Run the following query to determine if XDB is available in your database:

```
select comp_name, version, status from dba_registry where comp_id = 'XDB';
```

In order to install XDB run @?/rdbms/admin/catqm.sql with the following parameters:

```
XDB user password -> welcome1 (customer specific)
XDB user default tablespace -> SYSAUX
XDB user temporary tablespace -> PSAPTEMP
YES or NO (SecureFile or LOBs) -> YES
```

Example:

```
sqlplus / as sysdba
spool xdb_install.log
set echo on;
connect / as sysdba
shutdown immediate;
startup;
@?/rdbms/admin/catqm.sql welcome1 SYSAUX PSAPTEMP YES
@?/rdbms/admin/utlrlp.sql
spool off
```

For further details refer to My Oracle Support (MOS) note 1292089.1

Getting the IM Advisor Package and Documentation

The IM Advisor is contained in the SAP Bundle Patch as patch 21231656. The SAP Bundle Patch places the IM Advisor in the directory <ORACLE_HOME>/sap/ora_imadvisor.

The IM Advisor can also be downloaded from Oracle as described in My Oracle Support (MOS) note 1965343.1.

Furthermore the IM Advisor white paper can be downloaded from SAP Support Note 2178980 or from My Oracle Support (MOS) Note 1965343.1

The IM Advisor uses AWR statistics and – optionally – a SQL Tuning Set to generate its recommendations. Because of this it can run either on the database where the AWR statistics (and SQL Tuning Set) have been collected or on another database where the AWR statistics (and SQL Tuning Set) have been loaded into using AWR and STS export/import methods as well as IM Advisor's augment export and import scripts (check scripts imadvisor_awr_augment_export.sql and imadvisor_awr_augment_import.sql).

Installation

Follow the IM Advisor documentation and start the installation of IM Advisor either by unzipping the package if downloaded from the MOS note or by running the imadvisor_install.sql installation script if IM Advisor was installed with the SAP Bundle Patch (SBP).

When installed with SBP execute:

```
cd $ORACLE_HOME/sap/ora_imadvisor
sqlplus / as sysdba "@imadvisor_install"
```

It is recommended to create the IM Advisor schema objects in the SYSAUX tablespace instead of the SYSTEM tablespace.

Using IM Advisor

AWR Statistics and SQL Tuning Set

IM Advisor can work with AWR statistics only. AWR statistics are collected automatically on a regularly basis and kept until they expire. It is not recommended to rely on AWR statistics only because relevant SQL statements may have already been aged out from the SQL area during snapshot intervals. Therefore adding a SQL Tuning Set that contains all the statements run over a representative period of time can help to improve the results of IM Advisor significantly.

AWR statistics and SQL Tuning Set can be either collected on the live system during normal workload phases or if you are using the Oracle Real Application Testing Option (RAT) on a copy of the live system during workload replay.

The following script gives an example how to collect proper data for the IM Advisor in a live system. At the beginning it creates an AWR snapshot then it starts collection of SQL Tuning Set data every minute for the given amount of time and at the end it creates a second AWR snapshot. The AWR snapshot range and the captured SQL Tuning Set will then be used in the IM Advisor run.

Note: If you do not wish to use a SQL Tuning Set then you do not need to execute the following script.

```
ACCEPT sts_name PROMPT "Name of SQL Tuning Set:";
ACCEPT sts_owner PROMPT "Owner of SQL Tuning Set:";
ACCEPT sts_schema PROMPT "SAP schema for capture of sql area:";
ACCEPT sts_seconds PROMPT "How many seconds should be captured:";
```

```

SET SERVEROUTPUT ON;
DECLARE
    snapid_start number;
    snapid_stop number;
BEGIN
    snapid_start:=DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT();
    DBMS_OUTPUT.PUT_LINE('start snapshot id = '||snapid_start);
    DBMS_SQLTUNE.CREATE_SQLSET(
        sqlset_name =>'&sts_name',
        sqlset_owner =>'&sts_owner');

    DBMS_SCHEDULER.CREATE_JOB(
        job_name      => 'STSSAPWORKLOAD',
        job_type      => 'PLSQL_BLOCK',
        job_action    =>
            'BEGIN dbms_sqltune.capture_cursor_cache_sqlset(
                sqlset_name      => '&sts_name',
                time_limit       => &sts_seconds,
                repeat_interval  => 60,
                sqlset_owner     => '&sts_owner',
                basic_filter     => 'parsing_schema_name = ''&sts_schema''');
            END;');
    DBMS_SCHEDULER.RUN_JOB('STSSAPWORKLOAD');
    snapid_stop:=DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT();
    DBMS_OUTPUT.PUT_LINE('stop snapshot id = '||snapid_stop);
END;
/

```

We recommend to create the SQL Tuning Set in the same schema you are planning to run IM Advisor later i.e. “SAPWORK” as your SQL Tuning Set name, “SAPSR3” or “SYS” as your SQL Tuning Set owner, “SAPR3” or “SAPSR3” as your SAP schema. The script will print the AWR snapshot id at the start and the end of SQL Tuning Set collection. Please remember and write down those two snapshot ids as they will be needed later.

If you are planning to collect AWR statistics and SQL Tuning Set during workload replay on a test system you do not need to create the SQL Tuning Set with the script above. Instead just specify to collect SQL Tuning Set when calling DBMS_WORKLOAD_REPLAY.PREPARE_REPLAY.

Example:

```

begin
    dbms_workload_replay.prepare_replay
    (
        capture_sts => true,
        sts_cap_interval => 60
    );
end;

```

To determine the system generated name of the SQL Tuning Set check the view DBA_WORKLOAD_REPLAYS after the workload replay has completed.

Running IM Advisor

IM Advisor can run on any Oracle 12c database that contains the collected AWR statistics and SQL Tuning Set data. In addition it can run on any Oracle 11g database version 11.2.0.3 or higher that has XDB installed which is mandatory for the IM Advisor.

This means that you could collect AWR statistics and SQL Tuning Set in a live system and run the IM Advisor either in the same system or transport it from the live system to another system where you want to run the IM Advisor.

Transport of AWR Statistics, SQL Tuning Set and IM Advisor Augment Data from your Live System to another System for further Analysis

Export/Import of AWR statistics and SQL Tuning Sets is described in in the Oracle 12c Database Documentation. Export/Import of AWR augment data for the IM Advisor is described in the IM Advisor documentation.

Here an overview of the steps for exporting AWR statistics, SQL Tuning Set and IM Advisor augment data:

- Create an export directory on local filesystem
Example: `mkdir /oracle/C11/sapdata1/imadv_data`
- Create a directory object for that export directory
Example: `create or replace directory IMADV_DATA as '/oracle/C11/sapdata1/imadv_data'`
- Export AWR statistics within the relevant AWR snapshot range
Example: `sqlplus / as sysdba @?/rdbms/admin/awrextr`
- Export IM Advisor augment data
Example: `sqlplus / as sysdba @imadvisor_awr_augment_export.sql`
- If you have created an SQL Tuning Set, create a staging table, load the SQL Tuning Set data into the staging table and export the staging table using 'expdp' (datapump).
Example:

```
exec DBMS_SQLTUNE.CREATE_STGTAB_SQLSET(table_name=>'STSSAPWORKLOAD');
exec DBMS_SQLTUNE.PACK_STGTAB_SQLSET(sqlset_name=>'STSSAPWORKLOAD',
sqlset_owner=>'%', staging_table_name=>'STSSAPWORKLOAD');
expdp / DIRECTORY=IMADV_DATA DUMPFILE=stssapworkload.dmp
TABLES=STSSAPWORKLOAD
```
- Drop the SQL Tuning Set
Example: `exec DBMS_SQLTUNE.DROP_SQLSET(sqlset_name => 'STSSAPWORKLOAD');`
- Drop the staging table
Example: `drop table STSSAPWORKLOAD;`

Now copy the files in the export directory to your import directory on the system where you want to run the IM Advisor and import them.

- Create a directory object for that import directory
Example: `create or replace directory IMADV_DATA as '/oracle/C11/sapdata1/imadv_data'`
- Import AWR statistics
Example: `sqlplus / as sysdba @?/rdbms/admin/awrload`
- Check the dbid of the AWR import
Example: `select distinct dbid from dba_hist_snapshot;`
- Import IM Advisor augment data
Example: `sqlplus / as sysdba @imadvisor_awr_augment_import.sql`
- If you have created and exported an SQL Tuning Set, import the staging table using 'impdp' (datapump), copy the SQL Tuning Set from the staging table into the SQL Tuning Sets schema.
Example:

```
impdp / DIRECTORY=IMADV_DATA=stssapworkload.dmp TABLES=STSSAPWORKLOAD
exec DBMS_SQLTUNE.UNPACK_STGTAB_SQLSET(sqlset_name => 'STSSAPWORKLOAD',
replace => TRUE, staging_table_name => 'STSSAPWORKLOAD');
```
- Drop the staging table
Example: `drop table STSSAPWORKLOAD;`

IM Advisor Parameters with SAP

Beside AWR statistics and SQL Tuning Set the IM Advisor provides a number of parameters that take influence on the heuristics used to calculate and build the list of recommendations.

The following IM Advisor parameters have to be used for SAP applications. These parameters may slightly change over time or additional parameters may be added in the future.

IM Advisor Parameter Name and Value:

- WRITE_DISADVANTAGE_FACTOR = 0.7
- LOB_BENEFIT_REDUCTION = 1.2
- MIN_INMEMORY_OBJECT_SIZE = 1024000
- READ_BENEFIT_FACTOR = 2

IM Advisor can optimize the recommendations based on a specified amount of memory (RAM). This parameter should be used with SAP applications when generating the recommendations.

The script below is an example how to run the IM Advisor and fetch the results. It assumes that AWR statistics and an SQL Tuning Set have been collected as described in the above chapter AWR Statistics and SQL Tuning Set. It creates an IM Advisor task, sets the necessary parameters for proper heuristics, adds the SQL Tuning Set and the AWR statistics by specifying a start- and end- snapshot id and finally executes the task and generates the recommendations based on the available memory target value (inmemory_size) of **64GB**.

```
set serveroutput on
/* drop eventually pre-existing task */
exec dbms_inmemory_advisor.drop_task('SAPWORK',true);
exec dbms_inmemory_advisor.create_task('SAPWORK');
exec dbms_inmemory_advisor.set_parameter('WRITE_DISADVANTAGE_FACTOR', 0.7,
'SAPWORK');
exec dbms_inmemory_advisor.set_parameter('LOB_BENEFIT_REDUCTION', 1.2,
'SAPWORK');
exec dbms_inmemory_advisor.set_parameter('MIN_INMEMORY_OBJECT_SIZE', 1024000,
'SAPWORK');
exec dbms_inmemory_advisor.set_parameter('READ_BENEFIT_FACTOR', 2, 'SAPWORK');
/* Only needed when using a SQL Tuning Set */
exec dbms_inmemory_advisor.add_sqlset('SAPWORK', 'STSSAPWORKLOAD');
exec dbms_inmemory_advisor.add_hist_statistics('SAPWORK', 1306, 1590);
exec dbms_inmemory_advisor.execute_task('SAPWORK');
/* Memory Target Value 64GB = 68719476736 Bytes */
exec dbms_inmemory_advisor.generate_recommendations('SAPWORK', NULL, 68719476736);
```

Fetching and refining IM Advisor results

The generated recommendations are stored in the database and need to be fetched and written into local files.

```
SQL> @imadvisor_fetch_recommendations.sql
Enter value for im_task_name: SAPWORK
Fetching recommendation files for task SAPWORK
Enter value for recommendation_directory: /data/awr_2_20150605_im_12864/imadv
Placing recommendation files in directory /data/awr_2_20150605_im_12864/imadv
```

This will generate the files below:

```
[oracle@myhost imadv]$ ls -la *SAPWORK*
-rw-r--r--. 1 oracle oinstall 47673 Jun 11 11:09 imadvisor_auxiliary_SAPWORK.html
-rw-r--r--. 1 oracle oinstall 205698 Jun 11 11:09 imadvisor_SAPWORK.html
-rw-r--r--. 1 oracle oinstall 6151 Jun 11 11:09 imadvisor_SAPWORK.sql
-rw-r--r--. 1 oracle oinstall 259247 Jun 11 11:09 imadvisor_object_SAPWORK.html
-rw-r--r--. 1 oracle oinstall 453202 Jun 11 11:09 imadvisor_sql_SAPWORK.html
```

NOTE: Please check the value of the Analytics Processing Percentage from the Workload Database Usage section in the file [imadvisor_SAPWORK.html](#). If the value of Analytics Processing Percentage is less than 40% then using Database In-Memory will not improve the overall throughput of the SAP system. In such case Database In-Memory cannot help as there are not enough analytical queries produced by the SAP application.

In above example [imadvisor_SAPWORK.sql](#) contains the SQL statements to place the tables recommended by the IM Advisor into the IM column store. As the IM Advisor is not SAP aware it may have recommended tables which should never be loaded into the IM column store when running an SAP Application.

Because of this two scripts are provided which correct the IM Advisor output. These scripts are included in the SAP Bundle Patch (and not available from MOS).

The first script called [sapgenexclusions.sql](#) produces a text file (exclusions.txt) with all SAP tables that should not be placed into the IM column store. The second script called [sapmodimadvscript.pl](#) will generate the final list of recommendations in the file [finalimtabs.sql](#) based on the output of the IM Advisor and the file [exclusions.txt](#). In addition it will add the required "PRIORITY HIGH" option for every in-memory table used by the SAP application. It is not supported to use any other priority for in-memory tables with SAP as it will affect the overall throughput of the SAP application.

Follow the steps below to generate the final SQL script that can be used to place the remaining tables into the IM column store.

1. Go to directory \$ORACLE_HOME/sap/ora_imadvisor
2. Generate "exclusions.txt"

```
[oracle@myhost imadv]$ sqlplus / as sysdba @sapgenexclusions
SQL*Plus: Release 12.1.0.2.0 Production on Mon Jun 15 15:27:39 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

Enter the owner of your SAP tables (e.g. SAPR3 or SAPSR3):SAPR3

3. Generate final SQL script

On Unix/Linux platforms run

```
$ $ORACLE_HOME/perl/bin/perl sapmodimadvscript.pl \
    imadvisor_SAPWORK.sql \
    exclusions.txt > finalimtabs.sql
```

On Windows platforms run

```
C:\Users\oracle>%ORACLE_HOME%\perl\bin\perl sapmodimadvscript.pl \  
imadvisor_SAPWORK.sql \  
exclusions.txt > finalimtabs.sql
```

Load identified SAP tables into the IM Column Store

In order to load the tables into the IM column store run the generated SQL file ([finalimtabs.sql](#)) as user sys in sqlplus.

Example: sqlplus / as sysdba @finalimtabs.sql

Check Status of SAP tables in the IM Column Store

There are two database performance views v\$im_segments and v\$immemory_area which have to be checked to verify that all data of the identified SAP tables/partitions have been successfully loaded into the IM column store. For a detailed description of the database views for the IM column store refer to the Database Administration Reference Manual of the Oracle Database 12c Documentation, <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>

First check the status of the column POPULATE_STATUS of v\$immemory_area or gv\$immemory_area (RAC). The status should NEVER be OUT OF MEMORY which indicates that the size of the IM column store is too small. Not all data of all identified SAP tables/partitions can be loaded in the IM column store. In this case the size of the IM column store (init.ora: inmemory_size) needs to be increased.

Example 1: IM column store is too small

```
sqlplus / as sysdba  
SQL> select * from v$immemory_area;
```

POOL	A LLOC_BYTES	USED_BYTES	POPULATE_STATUS	CON_ID
1MB POOL	267386880	267386880	OUT OF MEMORY	0
64KB POOL	50331648	13107	DONE	0

Example 2: IM column store sized correctly

```
sqlplus / as sysdba  
SQL> select * from v$immemory_area;
```

POOL	A LLOC_BYTES	USED_BYTES	POPULATE_STATUS	CON_ID
1MB POOL	267386880	228589568	DONE	0
64KB POOL	503316	1310720	DONE	0

Second check the number of rows in v\$im_segments. The number of rows has to match the number of lines in the file [finalimtabs.sql](#). If the number of rows in v\$im_segments is less than the number of lines in [finalimtabs.sql](#) then not all tables or partitions have been loaded in the IM column store and this needs to be corrected. Depending on the number of selected SAP tables/partitions for the IM column store it may take several hours to get all tables loaded into the IM column store. So please check the contents of v\$im_segments accordingly.

Note: With Oracle RAC and using the performance view gv\$im_segments you may see multiple entries of the same table.

Third check the entry for each individual SAP table/partition in the IM column store using the view v\$im_segments. Important is the column BYTES_NOT_POPULATED ("bytes NOT in IM"). If this column has a value greater than zero not all data of an SAP table or an SAP table partition is loaded in the IM column store.

Example 1: Not all data is loaded in the IM column store

```
sqlplus / as sysdba
SQL> select owner, segment_name, partition_name, inmemory_size "bytes in IM",
bytes "bytes on disk", bytes_not_populated "bytes NOT in IM" from v$im_segments;
```

OWNER	SEGMENT_NAME	PARTITION_NAME	bytes in IM	bytes on disk	bytes NOT in IM
SAPSR3	CE4LA80		229703680	152043520	0
SAPSR3	BSIS		34799616	1.9461E+10	1.9354E+10

Note: With Oracle RAC the value of BYTES_NOT_POPULATED can be greater than zero as data of a single table will be distributed across the IM column stores of the RAC instances.

Example 2: All data is in IM column store

```
SQL> select owner, segment_name, partition_name, inmemory_size "bytes in IM",
bytes "bytes on disk", bytes_not_populated "bytes NOT in IM" from v$im_segments;
```

OWNER	SEGMENT_NAME	PARTITION_NAME	bytes in IM	bytes on disk	bytes NOT in IM
SAPSR3	CE4LA80		229703680	152043520	0
SAPSR3	MARC		90505216	23068672	0

Approach 2: Knowing which SAP table(s) to be placed into the IM Column Store

This approach should be taken if you already have identified certain SAP reports which run long and where you have identified that the long runtimes of these reports are caused by queries in the database which have to process a lot of data. This may be full table scans or complex joins in the database which read a lot of data and cannot efficiently be processed by using indexes in the database.

In such situations it makes sense to put a single table or a set of tables used by these long running queries or joins into the IM column store. For sizing purposes you need to know how much memory (init.ora parameter: `inmemory_size`) you need to allocate for the IM column store.

Note: Please verify that the identified tables are not included in the SAP exclusion list i.e. in the list produced by the script [sapgenexclusions.sql](#). Tables in the SAP exclusion list are not candidates for the IM column store.

To calculate the required amount of memory for a single table or a set of tables the `SAP_IM_ADV` package is provided. This package is part of SAP Bundle Patch and will be installed in the directory `<ORACLE_HOME>/sap/ora_imadvisor`.

Installation of SAP_IM_ADV package

1. Create an output directory for the sizing information (you can specify any directory here):

On Windows:

create or replace directory "~IM_ADV_DIR" as 'c:\TEMP';

On UNIX:

create or replace directory "~IM_ADV_DIR" as '/tmp';

2. Start the script `sapimadv.sql` in `sqlplus` as user `sys`:

```
sqlplus / as sysdba
```

```
SQL> start sapimadv
```

How to use SAP_IM_ADV package

The package `SAP_IM_ADV` can be called using the owner of the table and the table name:

```
sqlplus / as sysdba
```

```
SQL> set serveroutput on
```


```
SQL> exec sap_im_adv.in_memory_check ('SAPSR3', 'TABLE');
```

You can also provide a list of tables:

```
sqlplus / as sysdba
```

```
SQL> set serveroutput on
```

```
SQL> exec sap_im_adv.in_memory_check ('SAPSR3', 'TABLE1', 'TABLE2', ..., 'TABLEn');
```



The first parameter specifies the owner of the SAP schema and the second parameter specifies one or more tables separated by “,”.

The procedure has also two optional parameters. As the third parameter you may specify a working tablespace. This working tablespace must have enough free space as half the size of the largest table. The fourth parameter specifies the sample count. The default is 1,000,000 rows. The output is written to the screen and to a file in the directory which was created in the beginning. The file name is for single table operation <table>_IM_ADV.txt and for multiple tables it is MULTI_TABLE_IM_ADV.txt.

Note: For large tables it may take a considerable amount of time (i.e. 1-2 hours) to compute the in-memory size. Should you have a test system where the database is an exact copy of the production database you should use the SAP_IM_ADV package on the test system.

Example 1:

```
sqlplus / as sysdba
SQL> exec sap_im_adv.in_memory_check ('SAPSR3', '/BIC/B0001275000');
```

```
Output from /tmp/_BIC_B0001275000_IM_ADV.txt
rem
rem In-Memory Advisor Information Version 1.0
rem Space calculation for table /BIC/B0001275000
rem Size on Disk          : 14 GB
rem Required IM space (IM_QLOW): 1 GB
alter table "SAPSR3"."/BIC/B0001275000" inmemory memcompress for query low priority high;
```

So in this example you need an IM column store size of 1GB (init.ora: inmemory_size = 1GB) to load the partitioned SAP BW table /BIC/B0001275000 into the IM column store. Memcompress for query low is chosen as SAP tables are never read only and memcompress for query low provides good DML performance and excellent query performance for in-memory tables. Priority high is chosen to have the table immediately loaded and maintained in the IM column store.

Example 2:

```
sqlplus / as sysdba
SQL> exec sap_im_adv.in_memory_check ('SAPSR3', 'AFRU');
```

```
Output from file /tmp/AFRU.txt:
rem
rem In-Memory Advisor Information Version 1.0
rem Space calculation for table AFRU
rem Size on Disk          : 2108 MB
rem Required IM space (IM_QLOW): 290 MB
alter table "SAPSR3"."AFRU" inmemory memcompress for query low priority high;
```

So in this example you need an IM column store size of 290MB (init.ora: inmemory_size = 290M) to load the standard SAP R/3 table AFRU into the IM column store.

Example 3:

```
sqlplus / as sysdba
```

```
SQL> exec sap_im_adv.in_memory_check ('SAPSR3', 'COEP, EKPO');
```

Output from /tmp/MULTI_TABLE_IM_ADV.txt:

```
rem
rem In-Memory Advisor Information Version 1.0
rem Space calculation for table COEP
rem Size on Disk          : 74 GB
rem Required IM space (IM_QLOW): 16 GB
alter table "SAPSR3"."COEP" inmemory memcompress for query low priority high;
rem Space calculation for table EKPO
rem Size on Disk          : 5329 MB
rem Required IM space (IM_QLOW): 574 MB
alter table "SAPSR3"."EKPO" inmemory memcompress for query low priority high;
rem Total required IM space (IM_QLOW): 17 GB
```

So in this example you need an IM column store size of 17GB (init.ora: inmemory_size = 17G) to load the standard SAP R/3 tables EKPO and COEP into the IM column store.

Recommended Database In-Memory related init.ora Parameters

All SAP systems produce always a mixed workload of transactions and queries. To make sure that Database In-Memory will not impact critical transactions in the SAP system it is recommended to limit the resources used by Database In-Memory. For SAP systems therefore the following init.ora parameters should be used:

`inmemory_max_populate_servers = 4`

Limit the resources for column store background processing on a system

`inmemory_clause_default = "PRIORITY HIGH"`

Makes sure that tables/partitions are loaded into the IM column store as quickly as possible and data is in the IM column store before a query accesses the data and new inserted data of tables/partitions in the IM column store can be used by queries as quickly as possible.

`inmemory_size = <Size of the IM Column Store>`

Default Value of 0, which means that the IM column store is not used

Set to the value (+ ~20% for metadata and journals) used in the generate recommendation step of the IM Advisor (In above IM Advisor example used in this document the value would be 76G (64G + ~20%))

or

set to value calculated by the SAP_IM_ADV Package for all tables/partitions to be loaded into the IM column store

Note: `inmemory_size` is a static parameter. To make a change of this parameter effective you have to restart the database instance(s). Therefore if you already know the growth rate of the tables/partitions to be stored in the IM column store size this parameter accordingly. As a rule of thumb calculate with 20% of the growth rate. 20% because all tables/partitions are stored in compressed columnar format in the IM column store.

Example: If the growth rate for all tables/partitions in the IM column store is 10GB per month and you can restart the database only once a year then increase `inmemory_size` by 24GB (20% of 10GB x 12 months).



References

SAP Notes

2178980 – Using Oracle Database In-Memory with SAP NetWeaver

MOS Notes

1292089.1 – Master Note for Oracle XML Database (XDB) Install / Deinstall

1965343.1 – Oracle Database In-Memory Advisor

White Papers

<http://www.oracle.com/technetwork/database/manageability/info/twp-in-memory-advisor-bp-2430474.pdf>

Documentation

<http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>



CONNECT WITH US



blogs.oracle.com/oracle

facebook.com/oracle

twitter.com/oracle

oracle.com

Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

Hardware and Software, Engineered to Work Together

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.0115

Using SAP NetWeaver with Oracle Database In-Memory
August 2015



Oracle is committed to developing practices and products that help protect the environment