How To Size SAP BW on HANA

Thomas Becker – TIP In-Memory Platform BW P&QE
October 2012
DISCLAIMER

Sizing recommendations apply for certified hardware only. Please contact hardware vendor for suitable hardware configuration.

Note that HANA is constantly being optimized. This might have impact on sizing recommendations, which will be reflected in this document. Therefore, check for the latest version of relevant sizing notes.

Note that the sizing guideline in this document refers to SAP BW on HANA (aka Orange) only. Additional applications running on top of HANA are not covered in this document. Please refer to application specific sizing guidelines for details in that case.
SAP BW on HANA Sizing Elements

SAP BW on HANA sizing consists of

- Memory sizing for column store data
- Memory sizing for row store data
- Memory sizing for objects created during runtime (data load and query execution)
- Disk Sizing
Sizing Approaches

Depending on project status, there are three sizing approaches available for HANA:

• **QuickSizer Approach**
  • Suitable for customer that start „from scratch“
  • Some details on data models known (such as largest cubes, DSOs, user numbers, etc.)
  • Not suitable for customers who plan to migrate an existing BW

• **DB specific scripts**
  • Analyze data volumes of an existing BW system, based on database dictionary and statistics information
  • Scripts available for Oracle, DBx, MSSQL, MaxDB
  • OS access / DB admin required to run scripts
  • Fast (runtime: < 5 mins)

• **New: ABAP based sizing report**
  • Analyzes data volume of an existing BW system, based on data samples
  • DB independent report coming with Support Tools Plug-in
  • Much more accurate than script based sizing

  **Strongly recommended, preferred sizing tool**
QuickSizer For SAP HANA

1. Start QuickSizer tool: https://websmp103.sap-ag.de/quicksizer
   – QuickSizer general help

2. Create new project
3. Choose ‘SAP NetWeaver BW powered by SAP HANA’ questionnaire
4. Read help ‘How to fill in questionnaire’

5. Fill out the relevant sections
   - Table 5: Definition of InfoCubes for HANA (relevant for HANA sizing)
   - Table 6: Definition of DataStore Objects on HANA (relevant for HANA sizing)

   All other sections are used for BW Application Server Sizing!
DB specific scripts and sizing information provided by note 1637145

- Unix/Linux and Windows scripts draw information from database dictionary tables to determine size of **tables** in database (excluding indexes, temp table spaces, etc.)

- Distinction between row store tables and column store tables: different compression factors → Scripts deliver two values: **row store tables footprint** and **col store tables footprint**. These values need to be fed into **sizing formula** (see next slide)

- Scripts for DB2 family (DB2, DB4, DB6) take into account source database compression, other scripts don’t (compression factor has to be applied manually)

- To execute scripts, operating system access with DB administrator privileges is required

- Scripts rely on **up-to-date database statistics**

- Note: after a system cleanup (deletion of unnecessary data) the database system might not reflect the freed space before table reorganization has been performed

**Important: ABAP application server sizing to be done separately!**
BW on HANA Sizing Formula

**RAM:**

\[
\text{RAM} = \left( \text{colstore tables footprint} \times \frac{2}{4} + \frac{\text{rowstore tables footprint}}{1.5} \right) \times c^{1)} + 50 \text{ GB}
\]

**Disk:**

\[
\text{DISK}_{\text{persistence}} = 4 \times \text{RAM}
\]

\[
\text{DISK}_{\text{log}} = 1 \times \text{RAM}
\]

1) \(c\) = source database specific compression factor (only required if source database compression is used and source database is other than any of the DB2 family)
BW on HANA Sizing Formula: Remarks

Explanation of constants:

\[ \text{RAM} = \left( \frac{\text{colstore tables footprint} \times 2}{4} + \frac{\text{rowstore tables footprint}}{1.5} \right) \times c^1 + 50 \text{ GB} \]

- Uplift for runtime objects
- Average compression factor for col store
- Average compression factor for row store
- Source DB compression factor
- Uplift for HANA caches, additional services, etc.
BW on HANA Sizing Formula: Procedure and Example

1. Download archive „get_size.zip“ from SAP note 1637145 attachments
2. Extract script suitable for your DB and Operating system
3. Extract file „load_RowStore_List.sql“ into same folder
4. Execute script, result will be stored in file „Extent.txt“
5. Read sizing figures from script output:

<table>
<thead>
<tr>
<th>Table Type</th>
<th>GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RowStore Tables</td>
<td>49.56</td>
</tr>
</tbody>
</table>
| ColumnStore Tables| 672.35| Total Size: 701.91 GB

6. Feed figures into Sizing formula:

Assumption: source DB compressed by factor 1.8

\[
\text{RAM} = \left( \text{colstore tables footprint} \times \frac{2}{4} + \frac{\text{rowstore tables footprint}}{1.5} \right) \times \text{DB compression} + 50 \text{ GB} \\
= (672 \text{ GB} \times \frac{2}{4} + 50 \text{ GB} / 1.5) \times 1.8 + 50 \text{ GB} = 715 \text{ GB}
\]

\[
\text{Disk}_{\text{persistence}} = 715 \text{ GB} \times 4 = 2,860 \text{ GB} \\
\text{Disk}_{\text{log}} = 715 \text{ GB}
\]
ABAP Based Sizing

ABAP report developed to overcome deficiencies of DB scripts:
- DB independence
- Easier deployment
- Distinct size information for different types of tables (InfoCubes, DSOs, PSAs, …)
- Extrapolation of memory consumption based on future growth
- More sophisticated sizing approaches (e.g. non-active data) coming soon!

Availability
- Prerequisite: Support Tools Plug-in ST-PI 2008_1_7xx SP7
- Preliminary version based on ST-PI 2008_1_7xx SP6 available with note 1736976
- Report: /SDF/HANA_BW_SIZING for full system sizing
- Function Module: /SDF/HANA_TABLE_SIZE for single tables or list of tables

Documentation
- Please read documentation attached to note 1736976 for details and tips how to run the report
ABAP Based Sizing: How it works …

Sizing based on sample data
- For each DB table read small set of sample records into ABAP → source DB compression can be neglected!
- Determine average record length
- Calculate table size based on row count from DB statistics (need to be up to date!)
- Automatic distinction between row store and column store tables

Versatile parameterization
- Parallel execution: List of tables distributed across multiple processes to speed up runtime
- Different levels of precision (resulting in different sample sizes) – ‘Low’ good enough in most cases!
- Future growth projection

Distinct results
- Total sizes for InfoCubes, DSOs, Change Logs and PSA tables are determined
- Sizes of individual tables help identify potential for clean-up operations
- Automatically applies sizing formula: result contains data + runtime memory
Future growth: calculate future HANA requirements for next few years, based on relative or absolute yearly growth

Store output to file directly. If not set, a result screen will be displayed (which still can be saved as a file at a later point of time).

File name for output (relative to SAP work directory, or absolute path name)

Number of parallel work processes used for sampling. Note: free dialog processes required for each process!

In detailed list of tables, suppress those with sizes lower than 1 MB.

Precision setting: decreases / increases data sample rate. Note: lower precision means less run time!

Enter number of years and amount of growth (absolute in GB or relative in %)
Total source size and number of tables relevant for row store

Total size and number of tables relevant for column store

Size and number of tables corresponding to InfoCubes, DSOs, Change Logs, PSAs and Master Data, number of aggregate tables

Total number of tables processed

Minimum amount of physical memory for HANA server (including runtime memory, memory for additional services, etc.), and disk space.

Share of resources for row store and column store (data only and data including dynamic runtime memory)
Sizing recommendations considering future growth with detailed sizing requirements for each year (only if future growth was selected)

System information, including report run time, parameter settings, operating and DB system information and SAP BW release
Detailed size information for each row store table (sorted by size descending). Large tables may be good candidates for data cleansing to reduce memory requirements. Report shows ABAP size and size in HANA memory.

Detailed size information for each column store table (sorted by size descending). Helps to identify good candidates for data cleansing to reduce memory requirements (e.g. very large PSA tables, etc.).

Distinguishes between PSA (P) and Change Log (C) tables.
## Outlook: Not-Active Data in HANA

### Non-active data concept available with NetWeaver BW SPS8 and HANA SP5

| Hot | Data is read/written frequently  
     | In Memory, additional memory required for dynamic objects (merge, intermediate results, etc.) |
|-----|------------------------------------------------------------------------------------------|
| Warm| Mostly read access – lookups, transformations, etc.  
    | In Memory, no additional memory required for dynamic objects |
| Cool| Infrequent access – no need to keep in memory all the time  
    | On disk, loaded to memory only on demand, good candidate for displacement if memory runs short |

### Will be reflected in next version (V1.3) of Sizing Report – coming soon!
- Optimized usage of HANA memory resources
- Modeling of LSA concepts (Corporate Memory, …)

### NOTE: Due to the complexity of the sizing algorithm, **only the ABAP report will be able to consider non-active data.** Database scripts and QuickSizer will NOT be changed to reflect this feature!
If a single HANA node cannot accommodate data due to limited memory, data has to be distributed across multiple nodes (scale-out).

- **Symmetric solution**: 1 master node, n slave nodes, all on identical hardware

- **Master node** will handle system load and transactional load: ABAP system tables and general operational data of the BW are stored on the master node. DDL statements are executed on this node, global locks are acquired here.

- **Slave nodes** will handle OLAP queries as well as loading/staging/activation/merging. BW data (master data + cubes/DSOs/PSAs) is distributed across all slaves. This ensures a balanced utilization of the available CPU and memory resources.

- **Easy extensibility**: When a new slave is added, BW data can quickly be reorganized to fit the new system. More information here: [http://help.sap.com/hana/hana_db_part_en.pdf](http://help.sap.com/hana/hana_db_part_en.pdf)

- Optional: **stand-by node(s)** can take over in case of node failure
HANA BW Scale Out Concept: Overview 1/2 – Setup with 512 GB Nodes

Master Node
- 512 GB of memory
- holds system tables, all row tables
- DDL execution, global locks, database metadata

Up to 14 Slave Nodes
- 512 GB of memory each
- master data tables will be distributed across slave nodes
- fact, DSO and PSA tables are partitioned and distributed

Standby Node(s) (optional)
- 512 GB of memory
- can take over role of master node or of one of the slaves for failover
HANA BW Scale Out Concept: Overview 2/2 – Setup with 1 TB Nodes*

Master Node
- 1 TB of memory
- holds system tables, all row tables
- DDL execution, global locks, database metadata

Up to 14 Slave Nodes
- 1 TB of memory each
- master data tables will be distributed across slave nodes
- fact, DSO and PSA tables are partitioned and distributed

Standby Node(s) (optional)
- 1 TB of memory
- can take over role of master node or of one of the slaves for failover

* For Hardware vendors that support it
HANA BW Scale Out Concept: Pure High Availability Solution with a Standby Node

Worker Node
- 512 GB or 1 TB of memory
- single worker node

Standby Node
- same amount of memory as worker node
- can take over role of worker node for failover
HANA BW Scale Out Concept: Exception for systems with a single slave node

Please see note 1702409 why this setup is not recommended.

Master Node
- holds system tables, all row tables
- DDL execution, global locks, database metadata

Slave Node
- holds master data, fact, DSO and PSA tables

Standby Node (optional)
- can take over role of master or slave node for failover

For special requirements, master data tables may optionally be moved to the master node. This can lead to performance degradation. Master data tables may be moved manually (there is no official tool support) or with the help of a consulting note.
Thank You!

Contact information:

Dr. Thomas Becker
Quality Manager
TIP In-Memory Platform
BW Performance & Quality Engineering
Th.Becker@sap.com