SAP HANA Administration Guide for SAP HANA Service

Generated on: 2024-11-25 08:48:23 GMT+0000

SAP HANA Service for SAP BTP in AWS Regions | Cloud

PUBLIC

Original content: https://help.sap.com/docs/HANA_SERVICE_CF/6a504812672d48ba865f4f4b268a881e?locale=en-US&state=PRODUCTION&version=Cloud

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the https://help.sap.com/docs/disclaimer.

6 Data Compression in the Column Store

The column store allows for the efficient compression of data. This makes it less costly for the SAP HANA database to keep data in main memory. It also speeds up searches and calculations.

Data in column tables can have a two-fold compression:

- Dictionary compression

  This default method of compression is applied to all columns. It involves the mapping of distinct column values to consecutive numbers, so that instead of the actual value being stored, the typically much smaller consecutive number is stored.

- Advanced compression

  Each column can be further compressed using different compression methods, namely prefix encoding, run length encoding (RLE), cluster encoding, sparse encoding, and indirect encoding. The SAP HANA database uses compression algorithms to determine which type of compression is most appropriate for a column. Columns with the PAGE LOADABLE attribute are compressed with the NBit algorithm only.

Note

Advanced compression is applied only to the main storage of column tables. As the delta storage is optimized for write operations, it has only dictionary compression applied.

Compression is automatically calculated and optimized as part of the delta merge operation. If you create an empty column table, no compression is applied initially as the database cannot know which method is most appropriate. As you start to insert data into the table and the delta merge operation starts being executed at regular intervals, data compression is automatically (re)evaluated and optimized.

Automatic compression optimization is ensured by the parameter `active` in the `optimize_compression` section of the `indexserver.ini` configuration file. This parameter must have the value `yes`.

Note

If the standard method for initiating a delta merge of the table is disabled (AUTO_MERGE_ON column in the system view TABLES is set to FALSE), automatic compression optimization is implicitly disabled as well. This is the case even if the AUTO_OPTIMIZE_COMPRESSION_ON column is set to TRUE in the system view TABLES. It is necessary to disable auto merge if the delta merge operation of the table is being controlled by a smart merge triggered by the application. For more information, see the section on merge motivations.

Compression Factor

The compression factor refers to the ratio of the uncompressed data size to the compressed data size in SAP HANA.

The uncompressed data volume is a database-independent value that is defined as follows: the nominal record size multiplied by the number of records in the table. The nominal record size is the sum of the sizes of the data types of all columns.

The compressed data volume in SAP HANA is the total size that the table occupies in the main memory of SAP HANA.

Example

You can retrieve this information for a fully-loaded column table from the monitoring view M_CS_TABLES by executing the statement:

```
select SCHEMA_NAME, TABLE_NAME, MEMORY_SIZE_IN_TOTAL from PUBLIC.M_CS_TABLES where SCHEMA_NA
```

The compression factor achieved by the database depends on your SAP HANA implementation and the data involved.

For more information see *Cost Functions*

Cost Functions for Optimize Compression

The cost functions for optimize compression are in the optimize_compression section of the service configuration (e.g. indexserver.ini)

- **auto_decision_func** - if triggered by MergeDog
- **smart_decision_func** - if triggered by SmartMerge

Default Cost Function Configuration	Meaning
`MMU > 0.010240 and if(OCRC, max(MRC, OCRC) / min(MRC, OCRC) >= 1.75, 1) and (not RP or (RP and TMD > 86400))`	Optimize compression runs if • The table contains more than 10240 rows AND • (Optimize compression was never run before OR • The number of rows increase or decrease by factor of 1.75) • AND - if range partitioned, the last delta merge happened more than 24 hours ago.

Related Information

Merge Motivations
M_CS_TABLES System View
TABLES System View
Cost Functions
Compress a Column Table Manually

Check the Compression of a Column Table

For column-store tables, you can check the type of compression applied to table columns, as well as the compression ratio.

Prerequisites

To check the compression status of a table accurately, ensure that it is first fully loaded into main memory.

Procedure

1. To check the type of compression applied to table columns, execute the following SQL statement:

   ```
   SELECT SCHEMA_NAME, TABLE_NAME, COLUMN_NAME, COMPRESSION_TYPE, LOADED from
   PUBLIC.M_CS_COLUMNS where SCHEMA_NAME='<your_schema>' and TABLE_NAME='<your_table>'
   ```

   The columns of the selected table are listed with the type of compression applied. The following values are possible:

   - DEFAULT
   - SPARSE

  - PREFIXED
  - CLUSTERED
  - INDIRECT
  - RLE

  i Note

  Even if the column is not loaded into memory, the compression type is indicated as DEFAULT. This is because there will always be some level of dictionary compression. However, unless the column is loaded, the database cannot determine the type of compression actually applied. The LOADED column indicates whether or not the column is loaded into memory.

2. Check the compression ratio of table columns, that is, the ratio of the column's uncompressed data size to its compressed data size in memory.

   ```
   SELECT SCHEMA_NAME, TABLE_NAME, COLUMN_NAME, COMPRESSION_RATIO_IN_PERCENTAGE, LOADED
   from PUBLIC.M_CS_COLUMNS where SCHEMA_NAME='<your_schema>' and TABLE_NAME='<your_table>'
   ```

Related Information

Load/Unload a Column Table into/from Memory

M_CS_ALL_COLUMNS System View

Compress a Column Table Manually

The SAP HANA database decides which columns in a column table to compress and which compression algorithm to apply for each column. It does this as part of the delta merge operation. It is normally not necessary that you interfere with this process. However, you can trigger compression manually.

Prerequisites

You have the UPDATE privilege for the table.

Context

We do not recommend that you interfere with the way in which the SAP HANA database applies compression. However, if a table is not compressed and you think it should be, you can request the database to reevaluate the situation.

Before you do this, consider the reasons why the table may not be compressed, for example:

- The table is very small.
- The table's delta storage has never been merged with its main storage.
- The table was created and filled using an old version of the SAP HANA database that did not compress data automatically. No further data loads, and consequently no delta merge operations, have taken place.
- The auto merge function has been disabled for the table (AUTO_MERGE_ON column in the system view TABLES is set to FALSE). Deactivating auto merge for a columnstore table implicitly disables the automatic compression optimization as well. This is the case even if the AUTO_OPTIMIZE_COMPRESSION_ON column is set to TRUE in the system view TABLES.

Procedure

1. Request the database to reevaluate compression by executing the SQL statement:

   ```
   UPDATE "<your_table>" WITH PARAMETERS ('OPTIMIZE_COMPRESSION'='YES')
   ```

The database checks all of the table's columns and determines whether or not they need to be compressed, or whether or not existing compression can be optimized. If this is the case, it compresses the data using the most appropriate compression algorithm. However, note the following:

- The database will only reevaluate compression if the contents of the table have changed significantly since the last time compression was evaluated.
- Even if the database does reevaluate the situation, it may determine that compression is not necessary or cannot be optimized and so changes nothing.

2. Check the compression status of the table.
3. Optional: If compression has not changed, force the database to reevaluate compression by executing the following SQL statement `UPDATE "<your_table>" WITH PARAMETERS ('OPTIMIZE_COMPRESSION'='FORCE')`.

   The database checks all of the table's columns and determines whether or not they need to be compressed, or whether or not existing compression can be optimized. If this is the case, it compresses the data using the most appropriate compression algorithm. Note that the database may still determine that compression is not necessary or cannot be optimized and so changes nothing.
4. Check the compression status of the table.

Related Information

Check the Compression of a Column Table

The Delta Merge Operation

UPDATE Statement (Data Manipulation)

TABLES System View