

# Digital Tables Partitioning in SAP HANA 4: A Novel Method for Handling Huge Data Sets

Rajesh Azmeera

Technology Professional, Department of Information and Technology, Stryker, USA  
(<https://orcid.org/0009-0005-4643-1599>)

**Abstract:** *The growing demand for handling huge data sets has presented enterprises with significant challenges associated with storage capacity, cost management, and operational efficiency. Data processing of higher volumes of data at higher speeds by the CPU was something that couldn't be attained using conventional methods for storing data in tables. This study explores the trajectory of data growth, its impact, and how digital tables partitioning in SAP HANA System could help overcome the aforementioned challenges.*

**Keywords:** big data analytics, digital tables partitioning, SAP HANA big data solutions, big data with SAP

## 1. Introduction

SAP HANA 4 has a time-tested and proven solution for dealing with large tables using SAP HANA, an in-memory RDBMS that is 10-1000 X faster than traditional database systems like Oracle for most data processing operations.

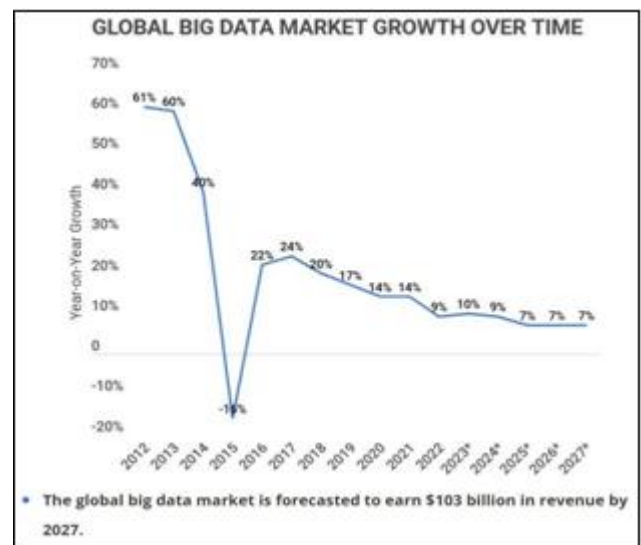
The speed of data processing of large data sets by the CPU depends on how the data is distributed among the tables and how the tables are managed and where they are stored.

Regardless of the RDBMS that's being used for processing data in tables, storing huge data in a single table will result in reduced processing speed and increased processing time, making an expensive statement while reporting.

## 2. Literature Review

The global big data analytics market is all set to explode, with projected revenues touching \$68 billion by 2025. This represents a substantial upswing of US\$53 billion from the market's US\$15 billion in 2019—a result of a robust projected Compound Annual Growth Rate (CAGR) of 30% during this time frame.

Demographics indicate that the world will generate, consume, and store a whopping 180 zettabytes of data by 2025. This figure was 94 zettabytes in 2022, commemorates a significant increase from the time the COVID-19 pandemic first broke out in March 2021. This can be attributed to an increase in the usage of cloud-based storage, remote WFH jobs and higher usage of digital home entertainment powered by OTT providers like Netflix during the pandemic. For reference, one zettabyte equates to one trillion gigabytes.



What is obvious here is the world is transcending towards creating, consuming and storing larger data sets with the passage of time.

```
REPLICATION_FALLBACK_ROUTING:"FALSE",TEMPORAL_APPLICATION_TIME_
AS_OF:"",TEMPORAL_SYSTEM_TIME_AS_OF:"") TRUE FALSE FALSE TRUE
1042814(6) SAPHANADB.ICFSERVICE(6) 1042814(6) SAPHANADB.ICFSERVICE(6)
(
0, 3) COLUMN GLOBAL -1 8,054,270,003 48,651 0 1 0 1,101
210,740 191 151 804 151,125 137 104 683 131,794 119 89 641 9,083 8 6 261 6,073 5
3 71 2 20,686 10,333 8,217 12,449 1,101 0 0 2023-02-24 12:00:01 927 2023-02-24
11:13:29 687 0 0 354,480 0 0 0 0 <Null>
30,003 3 SELECT [ICFPARGUID] ,RSSMREQ_OTP"
"ICFALTNAME" ,ICFALTNAME_ORIG" FROM "ICFSERVICE" WHERE "ICFNODGUID"
= ? LIMIT 1 1d6c745b8773c0035717e8f6cd248bfa SAPHANADB SAPHANADB
{ATTACHED_DEBUG_SESSION:"NO","CASE_SENSITIVE":"TRUE","DATETIME_FO
```

```
RMAT":0,"ENCRYPTED_SQLSCRIPT":"FALSE","RANGE RESTRICTION":",TABLE
REPLICATION_FALLBACK_ROUTING:"FALSE",TEMPORAL_APPLICATION_TIME_
AS_OF:"",TEMPORAL_SYSTEM_TIME_AS_OF:"") TRUE FALSE FALSE TRUE
1042814(6) SAPHANADB.ICFSERVICE(6) 1042814(6) SAPHANADB.ICFSERVICE(6)
30003, 0, 3) COLUMN GLOBAL -1 8,054,230,003 48,235 0 1 0 365
91,793 251 163 591 71,277 195 114 542 62,973 172 99 489 3,093 8 6 50 3 480 9 3 158
2 19,626 9,813 9,239 10,386 365 0 0 2023-02-24 12:00:01
```

Below is an infographic of an expensive stamen of Table (RSBMREQ\_DTP) populated with data.

Table RSBMREQ_DTP	
Reported as expensive statements	Yes
Is this portioned	No
CPU taking time to process this statement	15%
Memory to execute this statement	600GB (crossed statement limit which was set as 500GB)
OOO dumps	10 at OS

SAP HANA has two (2) robust solutions for distributing data in tables in a single distributed system.

### 3. Table Distribution in SAP HANA

#### Database Partitioning

Different tables are assigned to different index servers that run on different hosts.

#### Table Partitioning

A single table is split into multiple partitions (containing rows) which are then assigned to multiple index servers. In a distributed database system, a non-partitioned table can only be assigned to one index server.

By default, new tables are assigned to the available hosts in a round robin fashion for balancing the load evenly.

It is possible to override the default round robin algorithm for table distribution by explicitly specifying commands for a table or a partition to be created on a specific index server.

In the event an out-of-memory (OOM) dump on one of the hosts in the landscape is encountered, then one needs to verify whether the load is balanced evenly across all the hosts in the distributed system or not and resolve accordingly.

If the load is unevenly balanced, then SAP HANA has two scenarios for resolving the load balancing problem: Scale Up and Scale Out.

### 4. Case Study

#### Two Scenarios in SAP HANA for Load Balancing

In this scenario, a robust single HANA DB server is used to process the workload by upgrading the hardware of the server with better resources.

In this scenario, new resources are added across one or more hosts to process the workload, giving a business the benefit of scaling beyond the hardware capabilities of a single host.

Table Distribution is possible only in Scale Out scenario.

#### Privileges:

To implement a table redistribution, SAP HANA requires you to have a system privilege RESOURCE ADMIN and additionally at least the object privilege ALTER for all the involved schemas.

To implement a manual table distribution from one host to

another, SAP HANA requires you to have the following system privileges.

System Privilege 'DATA ADMIN' or System Privilege 'CATALOG READ' and SQL privilege 'ALTER'

You must ensure the following before adding or removing a host for redistribution. Optimize existing table distribution Optimize existing table partitioning

#### Parameters for table redistribution

global. ini-- > [table\_placement]-- > same\_num\_partitions  
By default this parameter is set to FALSE. It means that all the tables in the same group have the same number of level 1 partitions. The largest table in the group is taken into consideration for determining the number of partitions.

#### Indexserver. ini-- > [table\_redist]-- > all\_moves\_physical

By default this parameter is set to FALSE. Whenever a table redistribution is done in the default mode, tables with their operational memory part are only moved to the new host without their persistence part. The persistent part gets moved to the new host at a later time during the next delta merge.

Set this parameter to TRUE If you wish to move the persistent part along with the working memory part of the tables. However, this will come at a cost of a significant increase in the runtime of table redistribution.

#### Indexserver. ini-- > [table\_redist]-- > force\_partnum\_to\_splitrule

Set this parameter to TRUE if you wish to alter the number of level 1 partitions in a table. For instance, if the number of level 1 partitions in a table are 3 but you have a custom requirement to set them to 2, then by default the table redistribution cannot adjust this, you must activate this option before doing the custom settings for the adjustment.

#### global. ini-- > [table\_placement]-- > method

Set this parameter to "2" if you want to ensure that the row store tables are always created on the master node or moved there. By default, the classification of row store tables is not taken into account.

### 5. Performing Backup for Tables

SAP highly recommends that you always keep a backup of the tables you wish to redistribute both before and after running the table redistribution in the distributed system.

For backing up the existing table distribution, follow the procedure outlined below. In the Administrator Editor, navigate the path: Landscape-> Redistribution->Save. Upon hitting the Save option, a backup of the existing table distribution gets done.

For restoring the table distribution from a point in time, follow the procedure outlined below.

In the Administrator Editor, navigate the path: Landscape--

>Redistribution→Executed Operations.

Upon hitting the Executed Operations option, you must locate the operation area that corresponds to the table distribution that you are planning to restore.

Restore Table Distribution plan by performing the steps outlined below. call REORG\_GENERATE (ALGORITHM\_ID”), While in the same session, execute Select \* from REORG\_PLAN

Export the results, you could send the backup of the restored table to SAP for review.

## 6. Redistribution of Tables in SAP HANA

### Redistribute Tables Before Removing a Host

#### Before Removing a Host

Before removing a host in the distributed SAP HANA system, you must redistribute the tables in the index server of the host (that is being removed) to the other available index servers in the landscape.

#### Prerequisites

The prerequisites to be met for being able to redistribute tables in the distributed system are you must possess a) the system privilege RESOURCE ADMIN and b) at least the object privilege ALTER for all the schemas involved.

#### Backup Existing Tables Data

SAP highly recommends that you make a backup of the existing tables data in the index server of the host to be removed before running table redistribution so that you can recover it in the event the operation fails.

#### Procedure

In the Administration Editor navigate the path: Landscape – Hosts. Hover over the host that you want to remove in the landscape and go to the context menu of that host and single click the option Remove Host with the mouse.

You will be interacting with a Remove Host Dialog Box. Single click the option Yes with the mouse. The HANA System will execute the redistribution of tables operation after marking the host for removal.

#### Result

The end result is all the tables on the index server of the host in question to be removed are moved to the other available index servers of the remaining hosts in the landscape.

The host marked for removal gets removed in the landscape.

The redistribution operation will appear in the list of executed operations on the Redistribution tab.

### Redistribute Tables After Adding a Host

#### Prerequisites

The prerequisites to be met for being able to redistribute tables in the distributed system are you must possess a) the system privilege RESOURCE ADMIN and b) at least the object privilege ALTER for all the schemas involved.

### Backup Existing Tables Data

SAP highly recommends that you make a backup of the tables in the new host to be added before running table redistribution so that you can recover them in the event the operation fails.

#### Procedure

In the Administrator Editor navigate the path: Landscape→Redistribution

In the Redistributions Operations area, hover over the option Redistribute tables after adding host (s) and single click the option Execute with the mouse.

You will be interacting with a Table Redistribution dialog box. Go with the option Next.

The HANA system will evaluate the existing table distribution on the host to be added and propose a new redistribution plan that clearly states which tables will be moved to which available index servers on the landscape.

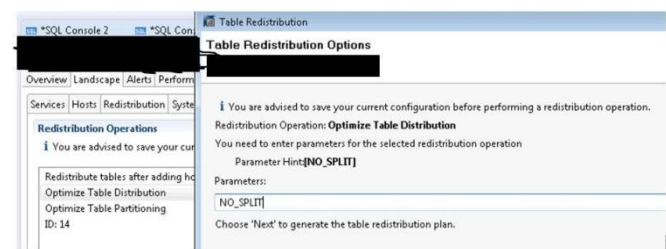
Review the redistribution plan and then single click the option Execute with your mouse.

#### Result

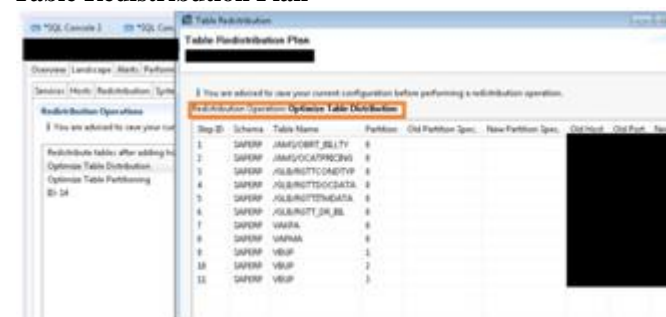
The end result is all the tables on the host to be added are moved to the different index servers in the landscape. The redistribution operation will appear in the list of executed operations on the Redistribution tab.

## 7. Below are Infographics for Table Redistribution

### Table Redistribution Method



### Table Redistribution Plan



Executed Operations

Executed Operations

You can see how tables are distributed in the [Table Distribution](#) editor.

ID	Status	Start Time	End Time	Duration (Mins)	Finished (Failed)/All
4	FINISHED	3/15/17 6:45 PM	3/15/17 6:45 PM	<1	11 (0) / 11
3	FINISHED	3/15/17 1:50 PM	3/15/17 1:50 PM	<1	1 (0) / 1
2	FINISHED	3/15/17 12:57 PM	3/15/17 12:58 PM	<1	73 (0) / 73
1	CANCELED	3/15/17 10:10 AM	3/15/17 10:10 AM	<1	213 (0) / 213

8. Table Partition

Row count: for partitioned tables, the row count can be no greater than 2 billion rows per table or partition

Statement Execution Performance: split the tables into partitions and assign them to different index servers for parallel processing. Scale Up: very specific performance indication

BI Scale Out: Distribution of the workload across multiple nodes must be the fundamental concern for high performance and speed. ERP Scale Out: DON'T ever distribute partitions of a single table across multiple nodes.

Delta Merge & Compression Performance: always ensure that the number of rows per partition and the disk size are small to facilitate smaller volume delta merge and optimize data compression.

9. Range Hash, and Round Robin Algorithms

For the purpose of illustration, here's an e, g. of a vividly used Hash Partition—a scenario where HANA Partitioning was used.

The problem: the BW Table RSBMREQ\_DTP reached the maximum allowed threshold value of 2 billion records per table and eventually the data collection process ended abruptly without a success.

The solution: perform table partitioning.

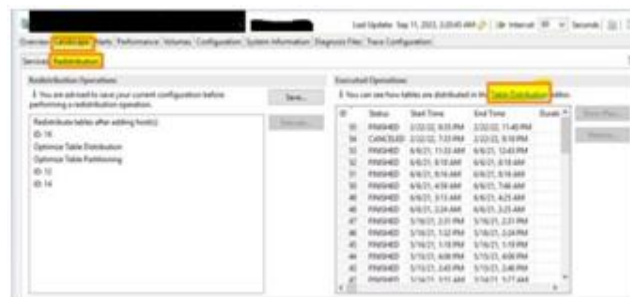
Here are the infographics for the above solution: Log into tenant database

Table Name:	Schema:	Type:
RSBMREQ_DTP		Column Store

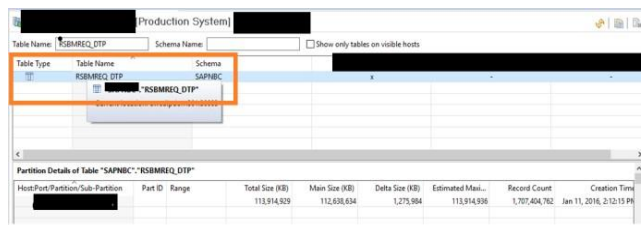
  

Columns   Indexes   Further Properties   Runtime Information			
General			
Total Memory Consumption (KB):	103,825,301	Memory Consumption in Main Storage (KB):	102,596,198
Number of Entries:	1,726,383,042	Memory Consumption in Delta Storage (KB):	1,227,419
Size on Disk (KB):	91,900,856	Estimated Maximum Memory Consumption (KB):	103,825,376

Navigate the path: Landscape→Redistribution→Table Distribution



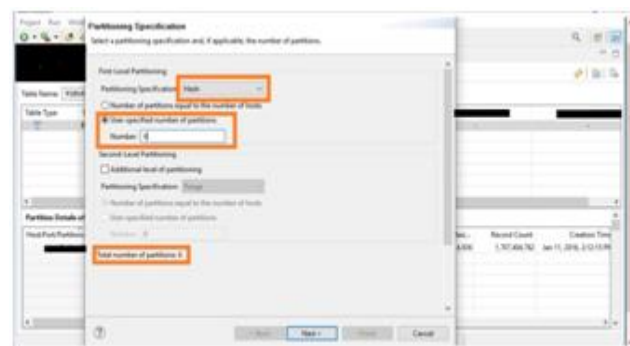
You may want to save the existing table distribution before starting over. Search tables need to be partitioned to overcome the above problem.



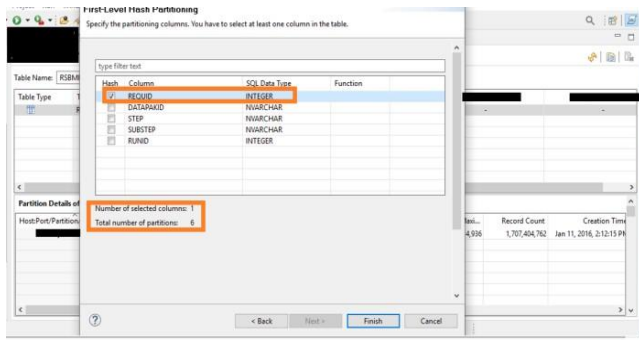
Single click on the option table with the mouse and then select Partition Table.



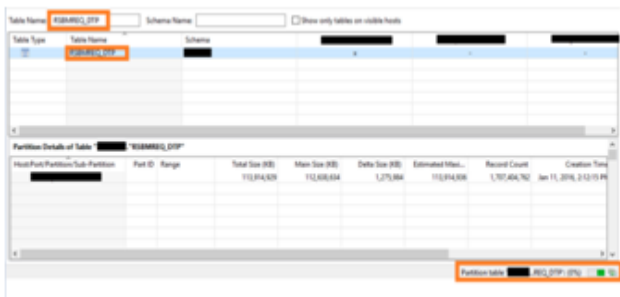
Single click the option type of partition. You will be interacting with a Partitioning Specification pop up. key in the value “#” for the field Number in the pop up and single click Next with the mouse.



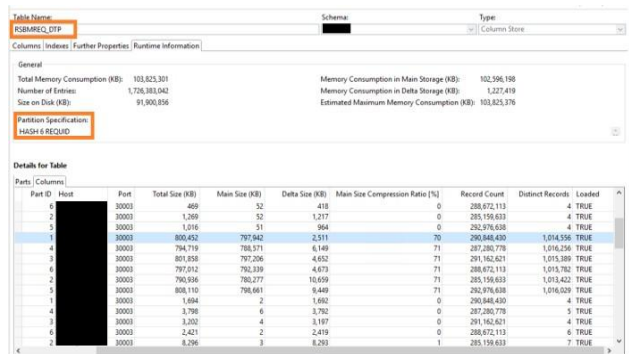
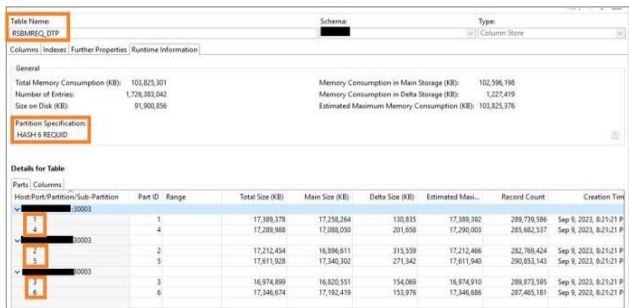
Single click the option Partitioning Column with the mouse.



You will get a message “Partition is in progress.”



That's it the HANA Partitioning is done and the problem is resolved.



**Results**

Table RSBMREQ_DTP	
Reported as expensive statements	No
Is this partitioned	Yes
CPU taking time to process this statement	2%
Memory to execute this statement	100GB (reduced from 500GB)
OOO dumps	No

**10. Conclusion**

We ensured that the data is distributed evenly across all the nodes for unleashing the power of faster data processing by the CPUs of the host nodes.

Additionally, the table is partitioned using HANA Partitioning to overcome the malfunction of the data collection operation due to the utilization of the maximum threshold value of 2 billion rows per table by Table RSBMREQ\_DTP.

Some table partitions are assigned to multiple threads for parallel processing.

For this use case, the total rows for the table are 1726383042. We have partitioned the above table into 6 partitions, each partition containing 287730507 rows.

As we relied on the power of multiple processors (CPUs) and multi-threading, we were able to process the data at higher speeds.

**Declarations:**

**Ethical Approval: Not Applicable Funding: Not Applicable**

**Availability of data and materials: Not Applicable**

**References**

- [1] HA 200, SAP HANA Installing and administering. SAP Training Materials. [Online]. Available at: <https://training.sap.com/course/ha200-sap-hana-installing-and-administering-classroom-019-g-en/>
- [2] Help Portal Documentation, SAP HANA Platform. SAP Help. [Online]. Available at: [https://help.sap.com/docs/SAP\\_HANA\\_PLATFORM](https://help.sap.com/docs/SAP_HANA_PLATFORM)
- [3] Table Partitioning, SAP HANA Docs, SAP Help. [Online]. Available at: <https://help.sap.com/docs/hana-cloud-database/sap-hana-cloud-sap-hana-database-administration-guide/table-partitioning>
- [4] SAP NOTE - 2044468 - FAQ: SAP HANA Partitioning. [Online]. Available at: <https://userapps.support.sap.com/sap/support/knowledge/en/2044468>
- [5] MD NAUSHAD ANSARI, Table partition in HANA, SAP Blogs. [Online]. Available at: <https://blogs.sap.com/2019/12/18/table-partition-in-hana/>
- [6] Jens Gleichmann, HANA Partitioning – 2 billion rows limitation – Part II: BSoH / S/4HANA, SAP Blogs. [Online]. Available at: <https://blogs.sap.com/2020/02/09/hana-partitioning-2-billion-rows-limitation-part-ii-bsoh-s-4hana/>
- [7] Gilbert Wong, Data Management for Technical Tables for SAP S/4HANA, SAP Blogs. [Online]. Available at: <https://blogs.sap.com/2021/02/09/data-management-for-technical-tables-for-sap-s-4hana/>
- [8] Richard Bremer (Author), Lars Breddemann (Author), SAP HANA Administration, SAP PRESS [Online]. Available at: [https://www.sap-press.com/sap-hana-administration\\_3506/](https://www.sap-press.com/sap-hana-administration_3506/)
- [9] SAP NOTES - 2081591 - FAQ: SAP HANA Table

Distribution. [Online]. Available at:  
<https://userapps.support.sap.com/sap/support/knowledge/en/2081591>

- [10] Big Data, Google Trends. [Online]. Available at:  
<https://trends.google.com/trends/explore?date=all&q=Big%20Data>