Microsoft

Data Migration Jumpstart

Getting SQL DW table size (Guide from PG)

Prepared by

Paula Berenguel

Solution Architect – DM Jumpstart

08-Aug-2018

Context:

During a typical DM Jumpstart POC, one of the key tasks that an architect must execute is getting metrics about the new environment post-migration: collect load times from on-premises to cloud, collect Polybase load times, etc. Amongst these tasks, one of the most important is the Storage size in SQL DW as opposed to the current platform of the customer.

It positively plays in Microsoft's favour as a success (savings) factor and, it contributes with the sizing/pricing of the to-be architecture usually conducted by account teams.

In a DM Jumpstart engagement, the storage size occupied by Azure SQL DW was considerably bigger than Netezza. In conversation with Azure SQL DW Engineering team, we solved the problem and that's why this IP was created.

Getting accurate table sizes in Azure SQL DW:

recommended by Matt Usher – SQL DW PM

Let's assume we are getting metrics from table DIM_EMPLOYEE. Assuming DIM_EMPLOYEE is a round_robin, cci index table.

The guide cover the default config for tables in Azure SQL DW (round_robin, cci) however, this guide can be used for any cci index on a table.

Step 1 – CTAS the table

```
CREATE TABLE [dbo].[DIM_EMPLOYEE_new]
WITH
(
    DISTRIBUTION = ROUND_ROBIN
,   CLUSTERED COLUMNSTORE INDEX
)
AS
SELECT  *
FROM    [dbo].[ DIM_EMPLOYEE];
```

Step 2 – Run Reorganize command on the new table

After performing loads of any type, you can have multiple small rowgroups in the deltastore. You can use `ALTER INDEX REORGANIZE` *to force all of the rowgroups into the columnstore, and then to combine the rowgroups into fewer rowgroups with more rows. The reorganize operation will also remove rows that have been deleted from the columnstore.*

```
ALTER INDEX ALL ON [dbo].[DIM_EMPLOYEE_new] REORGANIZE ;
```

Step 3 – Collect full statistics on the new table

Script to create statistics on a table with default sampling (20%):

select 'create statistics stats_col'+ convert(varchar(3), b.column_id)+' ON '+ c.name +'.'+ a.name+'('+b.name+');'

from sys.tables a

inner join sys.columns b on a.object_id = b.object_id

inner join sys.schemas c on c.schema_id = a.schema_id

where a.NAME = 'DIM_EMPLOYEE_new'

CREATE STATISTICS stats_col1 ON dbo.DIM_EMPLOYEE_new (EmployeeKey);

CREATE STATISTICS stats_col2 ON dbo.DIM_EMPLOYEE_new (EmployeeName);

Etc…

The default sampling rate of 20 percent is sufficient for most situations. However, you can adjust the sampling rate.

To sample the full table, use this syntax:

select 'create statistics stats_col'+ convert(varchar(3), b.column_id)+' ON '+ c.name +'.'+ a.name+'('+b.name+') WITH FULLSCAN;'

from sys.tables a

inner join sys.columns b on a.object_id = b.object_id

inner join sys.schemas c on c.schema_id = a.schema_id

where a.NAME = 'DIM_EMPLOYEE_new'

CREATE STATISTICS CustomerStats1 ON dbo.DIM_EMPLOYEE_new (EmployeeKey) `WITH FULLSCAN;`

CREATE STATISTICS CustomerStats2 ON dbo.DIM_EMPLOYEE_new (EmployeeName) `WITH FULLSCAN;`

Etc..

Step 4 – Check the row groups for the new/old tables using the query below. It should have 200-250K rows per index segment. Note that we are looking for any rowgroup that has a status of "Open"

```
SELECT       *
FROM sys.dm_pdw_nodes_db_column_store_row_group_physical_stats
--WHERE object_id  = object_id('SalesLT.Customer')
WHERE [state] = 1 OR [state_desc] = 'OPEN'
ORDER BY row_group_id;
```

Step 5 – Get the table size by running dbcc pdw_showspaceused

DBCC PDW_SHOWSPACEUSED("dbo.DIM_EMPLOYEE_new")

- Look for columns Data Space and Index Space
- Sum all the 60 rows for Data Space and Index Space

Suggestion: Paste it in excel and SUM(Data Space) + SUM(Index Space)

Real-World Example:

A real-world case from a DMJ engagement. Note how greatly those storage numbers differ from before and after following this guide from PG:

Before:

Table Name	Poc Netezza (GB)	Azure SQL DW (GB)
DIM_BILL_SUBS_INST_HISTORY	5.65	6.00
DIM_CUSTOMER_ACCOUNT	0.87	0.98
DIM_CUSTOMER_ACCOUNT_HISTORY	3.84	3.36
DIM_SUBSCRIPTION_INSTANCE	1.58	1.80
FCT_INVOICED_CHARGE	93.43	21.85
ODS_CH_PORTFOLIOITEMPRODUCT	4.64	1.80
ODS_SV_CUSTOMER_NODE_DA_ARRAY	13.75	2.59
ODS_SV_CUSTOMER_NODE_HISTORY	9.52	5.24
ODS_SV_SERVICE_TRE_V	6.63	0.06

After:

Table Name	Poc Netezza (GB)	Azure SQL DW (GB)
DIM_BILL_SUBS_INST_HISTORY	5.65	6.30
DIM_CUSTOMER_ACCOUNT	0.87	0.93
DIM_CUSTOMER_ACCOUNT_HISTORY	3.84	3.51
DIM_SUBSCRIPTION_INSTANCE	1.58	1.75
FCT_INVOICED_CHARGE	93.43	11.04
ODS_CH_PORTFOLIOITEMPRODUCT	4.64	4.51
ODS_SV_CUSTOMER_NODE_DA_ARRAY	13.75	2.41
ODS_SV_CUSTOMER_NODE_HISTORY	9.52	4.46
ODS_SV_SERVICE_TRE_V	6.63	5.42

More information:

https://docs.microsoft.com/en-us/sql/relational-databases/indexes/reorganize-and-rebuild-indexes?view=sql-server-2017

https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-defragmentation?view=sql-server-2017

Feedback and suggestions

If you have feedback or suggestions for improving this data migration asset, please contact the Data Migration Jumpstart Team (askdmjfordmtools@microsoft.com). Thanks for your support!

Note: For additional information about migrating various source databases to Azure, see the Azure Database Migration Guide.