**Power BI + Synapse Synergies – Demo**

**Composite Models/Dual Mode/Aggregations + Direct Query**

**Last updated 10/04/2021**

**Overview/ Demo Goals**

This is meant to be quick and easy to set up demo to show the Synergy of Power BI + Synapse Dedicated Pool for large data sets to enable huge performance gains and cost savings. This demo is based on the GA available features, but will include advise about using the Automated Aggregations preview. In particular,

* Demonstrate fast performance of many cases on Terabytes to Petabytes without the cost/time of importing all data into Power BI Service by using Aggregations/Dual Tables in combination with Direct Query.
* Demonstrate how refreshing the data very quickly using materialized views.
* Show how additional tuning is possible with Materialized Views and Resultset caching.
* Show new Auto Agg Feature
* Discuss and show how Power BI Accelerator (auto materialized views) will help

**Note: for setup of demo please see separate document. All documents and files needed are in github:**

[**https://github.com/stevemoss-microsoft/Power-BI---Synapse-Demo.git**](https://github.com/stevemoss-microsoft/Power-BI---Synapse-Demo.git)

1. **Demo Outline**
   1. **Show size of data and size of model**
   2. **Fast response of 1st level and 2nd level Agg**
      1. Show in UI
      2. Show in Monitor
      3. Show how it is set up: with Agg table connected to
   3. **Direct Query visual performance**
      1. Show in UI
      2. Show in Monitor
   4. **Accelerate: direct Query**
      1. Capture SQL and then do Explain with\_recommendations
      2. Implement Materialized view
      3. Test direct query again
      4. Turn on resultset caching and try
   5. **Preview Features**
      1. Automated Aggregations
      2. Accelerator (automated materialized views).
2. **Size of data and size of file**

This is on an over 1 billion row fact table.

Connect in SQL Server Management Studio or Azure Data Studio and run:

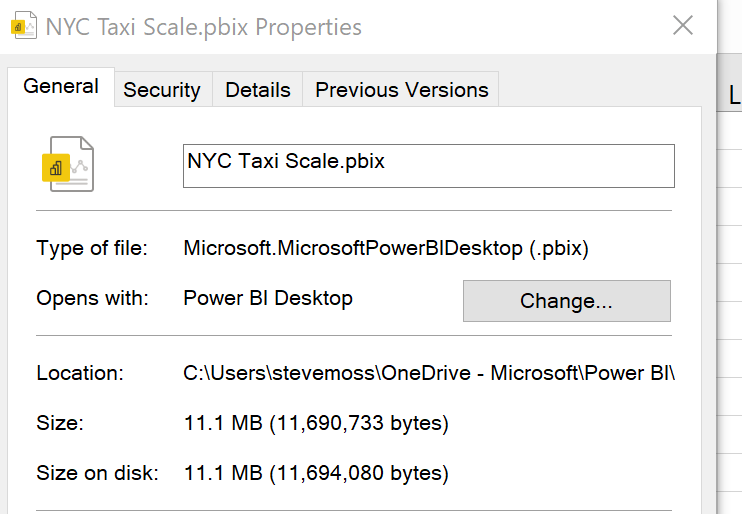
Sp\_spaceused trip

The result shows 1.7 billion rows.

Graphical user interface, text

Description automatically generated

But Size of model in MBs is small (right click and show file properties in file explorer):

****

1. **Fast response of 1st level and 2nd level Agg**
   1. Show in UI

In the Power BI Visualization Aggtest, add a DayName selection. The response time is very fast:

Chart, bar chart

Description automatically generated

Looking at the Profiler in Dax Studio, it can be seen that the duration is milliseconds 18 in this case.

Table

Description automatically generated

Also, that all the work was done in Aggregate Tables: there are Aggregate Table rewrite events, but no Direct Query events:

Graphical user interface, text, application, email

Description automatically generated

How can it be this fast on a 1.7 billion row table where the data isn’t imported (or even it is!)?

The reason is a new type of table called an Aggregate table. This can be viewed in the model/diagram view:

Graphical user interface, diagram

Description automatically generated

You can see that the dimensions are not only related to the main Trip fact table, but also to a table called vw.tripaggbig. This is an aggregate table, basically presummarized data is imported.

`Graphical user interface, application, Word

Description automatically generated

In this case the aggregations are only a sum on the trip distance by date, but they can be multiple aggregations at different levels of granularity.

Graphical user interface, application

Description automatically generated

Example of multiple aggregation tables.

Diagram

Description automatically generated

Also note that the Date table has broken blue lines: this indicates that this dimension table is in Dual Mode, which means that depending on whether the query can be satisfied by an aggregation table it will act as a table in Import Mode, but when the query needs a Direct Query it will act as in Direct Query mode.

**Accelerating refresh time of Aggregate tables using Materialized Views**

Note that the setup scripts included materialized views. In this case, the aggregate tables are based on these materialized views. The benefit of this is that refresh times for the aggregate table can be very fast because the view is materialized as a table, and any updates are automatically added to it.

Graphical user interface, application, Word

Description automatically generated

In a test with Power BI Desktop, it took about 60 seconds to refresh an aggregation table even with 11 million rows.

Text, table

Description automatically generated

**Direct Query visual performance**

Show in UI

On the visual titled Direct Query, select different hourly buckets. It takes a bit longer than the previously visual.

Check the Profiler in Dax Studio and it shows Direct Query events, rather than only aggregate events. This is the magic in Power BI Dual Mode/Composite Model: it automatically uses an Aggregate table when it is possible and when it is not it will automatically generate a Direct Query.

Table

Description automatically generated

In looking at the DirectQuery End event the details pane shows the actual query:

Text

Description automatically generated

**Accelerate Direct Query: using Materialized Views and Resultset Caching.**

We can further accelerate the Power BI + Synapse Direct queries by determining Materialized Views that would. This is easy to determine by capturing the direct query sql, and then running it with Explain With\_Recommendations.

To show this,

In looking at the DirectQuery End event the details pane shows the actual query:

Text

Description automatically generated

Capture this query, then put it in SSMS or Azure Data Studio

Graphical user interface, text, application

Description automatically generated

Next, copy the Explain plan XML and put it in your favorite text editor, you will note that there is a section for Materialized View Candidates:

Scatter chart

Description automatically generated

This generates one or more scripts to create materialized views that may increase the performance of the query. It is advised to collect several queries and determine a materialized view that will assist with many direct queries.

Another example of Synapse Synergies is that it can create Materialized Views that join to multiple tables, as opposed to Snowflake which can only create materialized views on a single table.

Graphical user interface, text, application

Description automatically generated

* + 1. Resultset Caching

Another feature available in Synapse to enhance performance is Resultset Caching.

By turning on this feature, if the direct query issued by Power BI has been issued previously within time and the data hasn’t changed, it will be cached on SSD disks and won’t require any query processing.

-- run in master

ALTER DATABASE dedpoolmoss SET RESULT\_SET\_CACHING ON;

The cache will be preserved even if the Synapse Dedicated Pool is paused/resumed or scaled up or down, unlike Snowflake where the cache will be reset during scaling and pausing.

**Automated Aggregations - Preview:**

With the above techniques, aggregations must be created manually so the patterns of use must be known where they will benefit. With a the new Power BI Automated Aggregations feature in preview, the aggregations will be created automatically.

“Automatic aggregations use state-of-the-art machine learning (ML) to continuously optimize DirectQuery datasets for maximum report query performance. Automatic aggregations are built on top of existing [user-defined aggregations](https://docs.microsoft.com/en-us/power-bi/transform-model/aggregations-advanced) infrastructure first introduced with composite models for Power BI.” Unlike user-defined aggregations, automatic aggregations don’t require extensive data modeling and query-optimization skills to configure and maintain. Automatic aggregations are both self-training and self-optimizing. They enable dataset owners of any skill level to improve query performance, providing faster report visualizations for even the largest datasets. “ [Automatic aggregations overview - Power BI | Microsoft Docs](https://docs.microsoft.com/en-us/power-bi/admin/aggregations-auto#:~:text=%20Automatic%20aggregations%20%28Preview%29%20%201%20Requirements.%20Automatic,report%20or%20interacts%20with%20a%20report...%20More%20)

The process can be diagrammed as below:

Diagram, schematic

Description automatically generated

The aggregations can be set up in the Power BI workspace dataset, by Settings then Optimize Performance.

Graphical user interface, text, application, email

Description automatically generated

Chart, line chart

Description automatically generated

Setting the percentage of queries that will use the aggregations and the target query performance time will influence how many aggregations are created. Note that this is not synchronous isn’t applied immediately.

Also, one should be careful as setting these higher may cause the refresh to use excessive resources on the server:

*“Training and refresh operations are process and resource intensive for both the Power BI service and the data source systems. Increasing the percentage of queries that use aggregations means more aggregations must be queried and calculated from data sources during training and refresh operations, increasing the probability of excessive use of system resources and potentially causing timeouts.”*

**Power BI Performance Accelerator (with Synapse) - Preview**

The basic idea of Power BI Performance Accelerator is to create materialized views automatically based on the power bi usage pattern. In a sense it’s like the auto aggregations in Power BI, but done in Synapse. While this feature has much promise, the preview of this feature is closed to new customers, and that it will only be implemented GA on Synapse Gen3, so won’t be available for current Synapse Gen2 users.

[Powering data exploration and data warehousing with new features - Microsoft Tech Community](https://techcommunity.microsoft.com/t5/azure-synapse-analytics/powering-data-exploration-and-data-warehousing-with-new-features/ba-p/1695416#:~:text=As%20Power%20BI%20users%20run%20their%20queries%20and,improving%20query%20response%20times%20over%20the%20latest%20data.)