**Power BI + Synapse Synergies: Enabling fast performance over huge data sets for less cost.**

**Overview**: Power BI can handle Terabytes or Petabytes of Data by using features of Power BI and features of Synapse Dedicated pool, without the cost of storing all the data in Power BI, or overly long refresh times. The features that can accomplish this are GA available right now if implemented manually and there are 2 features that are in preview that will automate and optimize them. Some of the features are on the Power BI side, and some on the Synapse Dedicated Pool side, and some work best together.

Power BI Features in GA are: Aggregations/Composite models used with Direct Query.

Synapse Dedicated Features in GA are: Materialized Views and Resultset Caching.

Power BI Feature in Preview is: Automated Aggregations.

Synapse + Power BI feature in preview is: Power BI Accelerator (auto materialized views for Power BI).

This is meant to be an overview of the concepts and how they are implemented, rather than a step-by-step demo: for a turn-key demo of these features you can use, please see [stevemoss-microsoft/Power-BI---Synapse-Demo (github.com)](https://github.com/stevemoss-microsoft/Power-BI---Synapse-Demo)

Graphical user interface, text, application

Description automatically generated

**Aggregations + Dual/Composite Models with Direct Query in Power BI**

Aggregations in Power BI are just what they sound like: instead of storing raw data, they store aggregations, sums, averages, grouped by different dimensions. To use them with large datasets, they are added to the dimensional design and joined to the dimension.

For example, in Power BI, the tables are best modeled in a dimensional design/star schema such as this example where Trip is a Fact table, and Date, Time and Geography are dimensions that join and filter the facts.

Graphical user interface

Description automatically generated

For very large tables (Terabytes to Petabytes and/or billions or trillions of rows), it would be too large to import pull all the data into the fact table, so Direct Query technique is used, that will generate queries to the database (Synapse in this case). In other words, a pass through. This can be workable, especially if Synapse is designed properly and features such as Clustered Columnstore Indexes, Materialized Views, Resultset caching and others are used.

But to make performance even better, pre-aggregated tables imported into Power BI memory can be added to the model. Since these are at a higher level of granularity, they won’t use as much memory as bringing in the lowest level of granularity.

The magic is that when the visualizations can use the pre-aggregations, they automatically do, and when the visualization needs additional data from the full data set, it will automatically issue a direct query. The process diagram:

Diagram

Description automatically generated

[User-defined aggregations - Power BI | Microsoft Docs](https://docs.microsoft.com/en-us/power-bi/transform-model/aggregations-advanced)

While this is not a step-by-step how-to article, the basic technique is to create aggregation table(s), and set up relationship to dimensions, then set up the aggregations through the manage aggregations option on the aggregation table model.

Graphical user interface, application

Description automatically generated

This brings up this screen to manage the aggregations:

Graphical user interface, application, table

Description automatically generated

Note that several Aggregation tables may be created at different levels of granularity, which can satisfy different queries, and the Aggregations can be prioritized as to which could be tried first..

**Populated/Refreshing the Aggregation Table – using Synapse Materialized Views:**

One method to create the aggregation table is by creating a materialized view in Synapse Dedicated Pool, and using that as the source of the aggregation table. The advantage to this is that the aggregation table can be refreshed very quickly.

For example,

Create materialized view vw.tripaggbig WITH (DISTRIBUTION=round\_robin) AS

SELECT dbo.trip.[DateID],pickupgeographyid,

SUM(TripDistanceMiles) as SumTripDistance,

count\_big(TripDistanceMiles) as cb

FROM dbo.trip

GROUP BY dbo.trip.[DateID],pickupgeographyid

This is used as the datasource which you can see in the edit query/Power Query Editor.

Graphical user interface, application, Word

Description automatically generated

**Competitive advantages of Synapse Materialized Views and Resultset Caching**

Note that Materialized Views in Synapse dedicated pool are more powerful and useful than the equivalent in Snowflake, because in Snowflake a materialized view can only query a single table, no joins are permitted. In addition, in some cases Synpase Resultset caching can provide another level of performance in that if the data has not changed since the aggregation table has been refreshed, the results will be cached on disk and returned with no computation or concurrency slots needed. Resultset Cache in Synapse have advantages over the implementation in Snowflake because they are maintained when the DW is paused or scaled.

In addition, the direct queries generated by Power BI can be captured and it can be determined if materialized views can be used. Adding additional Materialized Views to speed up Direct Queries can be a very powerful technique, and again Synapse’s ability to create them with joins to multiple tables makes them much more useful and powerful than Snowflake’s implementation.

**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.)

**Summary**

While Power BI features of manual and auto aggregations and direct query enable analysis/visualizations over massive data sets, combining Power BI with the power and performance features of Synapse Dedicated Pool can improve performance, refresh time and scalability.