Azure Synapse Analytics

How to tune Azure Synapse Analytics SQL Dedicated Pool performance and enable its full potential

Tiago Balabuch

Cloud Solution Architect
@Microsoft

10+ years of experience in data related roles

Lived in 4 different countries

Enthusiast of Azure SQL Database, Azure Synapse Analytics, Cosmosdb, Data Lake and AI

From SQL to Analytics

Traveler and Beach addict

linkedin.com/in/tiagobalabuch

Filipa Lobão

Data Cloud Solution Architect
@Microsoft

5+ years of experience in data related roles

Enthusiast of Azure Synapse Analytics, Power BI, Azure Data Factory, Azure SQL Database, AI&ML and Security

Biology > Cabin Crew > BI Dev&Ops > Azure D&AI Architect

Love speaking, outdoors and experiencing new cultures

linkedin.com/in/filipalobao
github.com/filipalob

Agenda

Business Case Overview

Synapse SQL Dedicated Pool architecture

Table Distributions & Indexes

Data Movement

Materialized View & Workload Management

Monitoring

Common Issues

Demo

Resources

Business Case Overview

Poor query performance

- Difficulty in achieving a performance benchmark
- On-premises to cloud migration scenarios

Dedicated SQL Pool

Dedicated clusters optimized for mission-critical data warehouse workloads

Dedicated SQL Pool Architecture

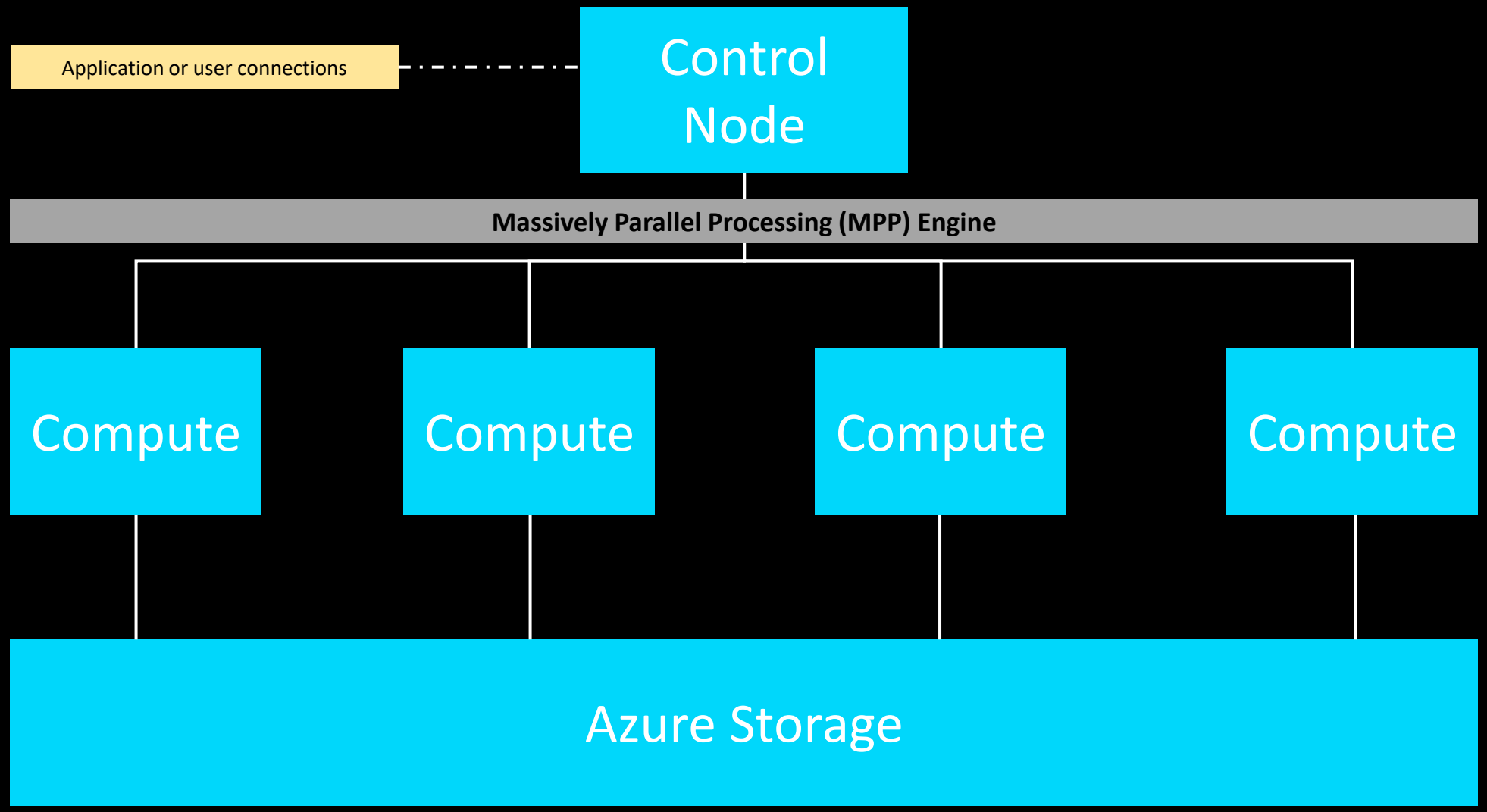
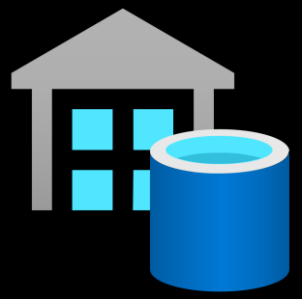


Table Distributions



Round Robin

Distributes table rows evenly across all distributions at random

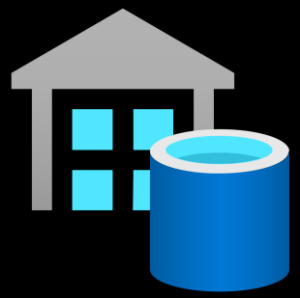
Hash

Distributes table rows across the Compute nodes by using a deterministic hash function to assign each row to one distribution.

Replicate

Full copy of table accessible on each Compute node.

Round-robin table – Best Practices



Round-robin is the default distribution style - **be careful**

The assignment of rows to distributions is random

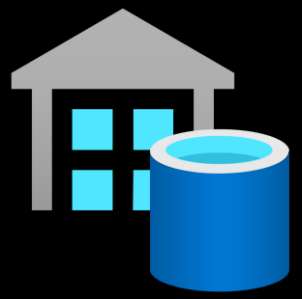
Rows with equal values are not guaranteed to be assigned to the same distribution

Most of the cases it invokes a data movement operation

A simple starting point since it is the default

- If there is no obvious joining key
- If there is no good candidate column for hash distributing the table
- If the table does not share a common join key with other tables
- If the join is less significant than other joins in the query
- When the table is a temporary staging table

Hash-distributed table – Best Practices



Minimize data skew

- Means the data is not distributed evenly across the distributions
- Processing skew means that some distributions take longer than others when running parallel queries

Has many unique values (at least > 600 minimum distinct values)

Does not have NULLs or has only a few NULLs

Minimize data movement by:

- Using Columns in JOIN, GROUP BY, DISTINCT, OVER, and HAVING clauses
- Avoid Columns used in WHERE clauses
- Avoid Column that are a date column

Choose a column with data that distribute evenly

- All the distributions should have approximately the same number of rows

Replicated table – Best Practices

Replicated table is fully copied to a distribution database on each compute node

- Queries run fast because joins on replicated tables don't require data movement
- Table size is less than 2 GB compressed
- If the data is static and does not change, you can replicate larger tables
- Use replicated tables instead of round-robin tables in most cases
- Avoid table that has frequent insert, update, and delete operations

Indexes

Indexes

Clustered Columnstore (Default)

Highest level of data compression

Best overall query performance

Clustered

Performant for looking up a single to few rows

Heap

Faster loading and landing temporary data

Best for small lookup tables

Nonclustered

Enable ordering of multiple columns in a table

Allows multiple nonclustered on a single table

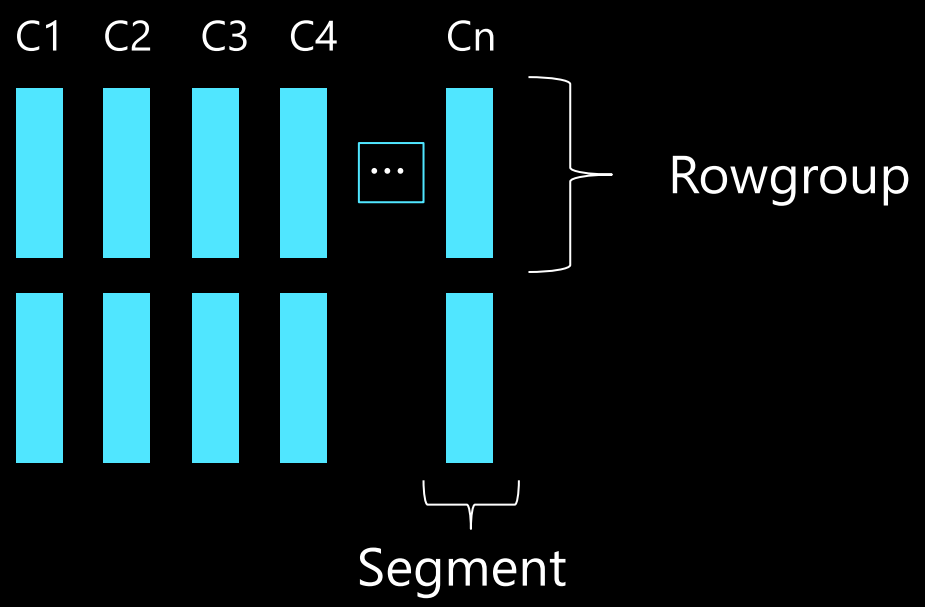
Can be created on any of the above primary indexes

More performant lookup queries

Clustered Columnstore Indexes

Data store as rows

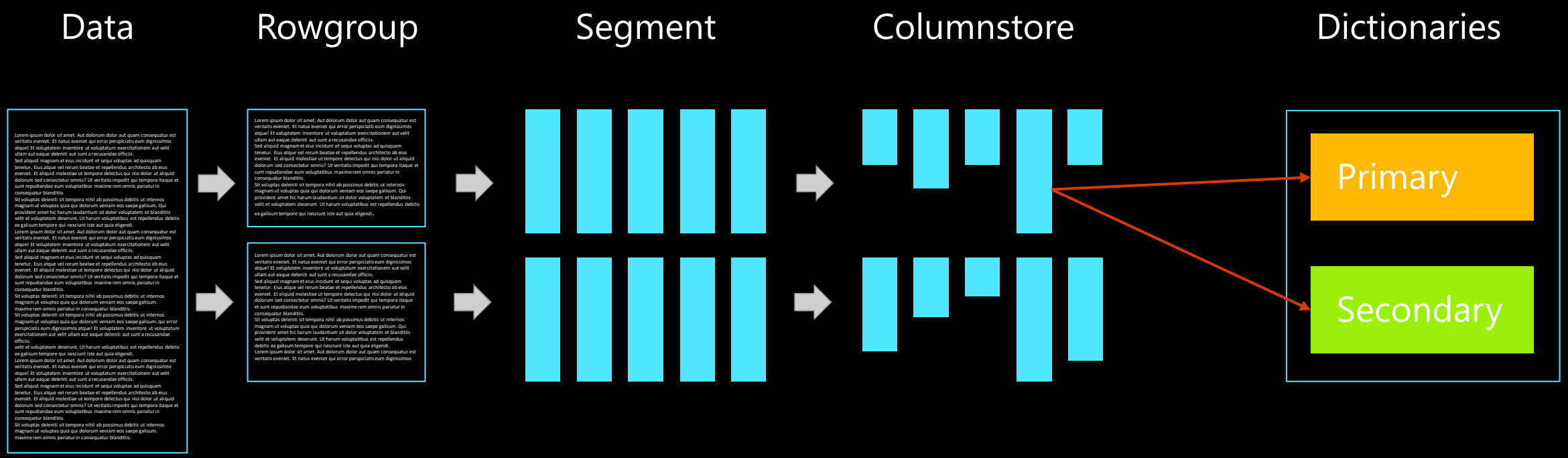
Data store as columns



Frequent read & writes
Small set of rows

Improved compression
Reduce I/O
Improved Performance
Batch mode execution

Terminology



Rowgroup

Group of rows into batches of ~1M rows

Segment

Values from one column of the rowgroup

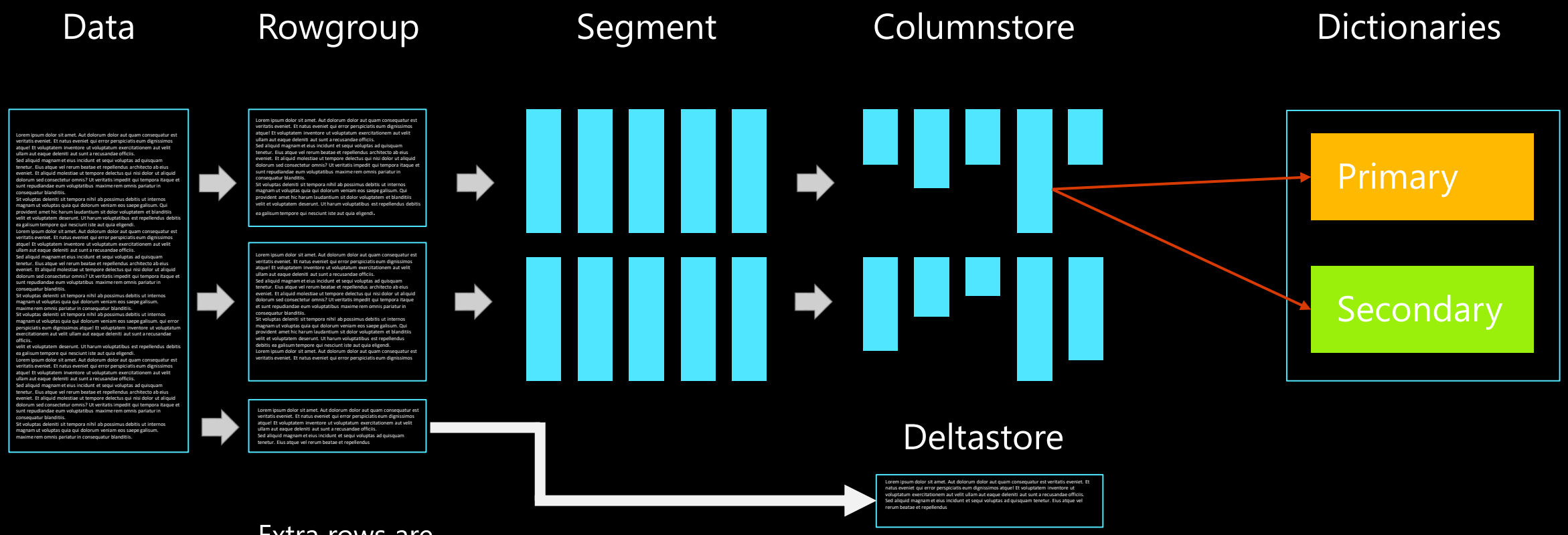
Dictionaries

Used to encode some but not all data types, therefore not all columns

Primary - for all segments

Secondary - used for a subset of the column's segments

Terminology



Rowgroup states

Open	Accepting new rows
Closed	Contains the maximum number of rows waiting for a background task to compress it into the columnstore
Compressed	A row group that is compressed with columnstore compression and stored in the columnstore

Rowgroup Health – Trimming types

NO TRIM - The row group was compressed with the maximum of 1,048,576 rows

BULKLOAD - The bulk-load batch size limited the number of rows (102,400 rows)

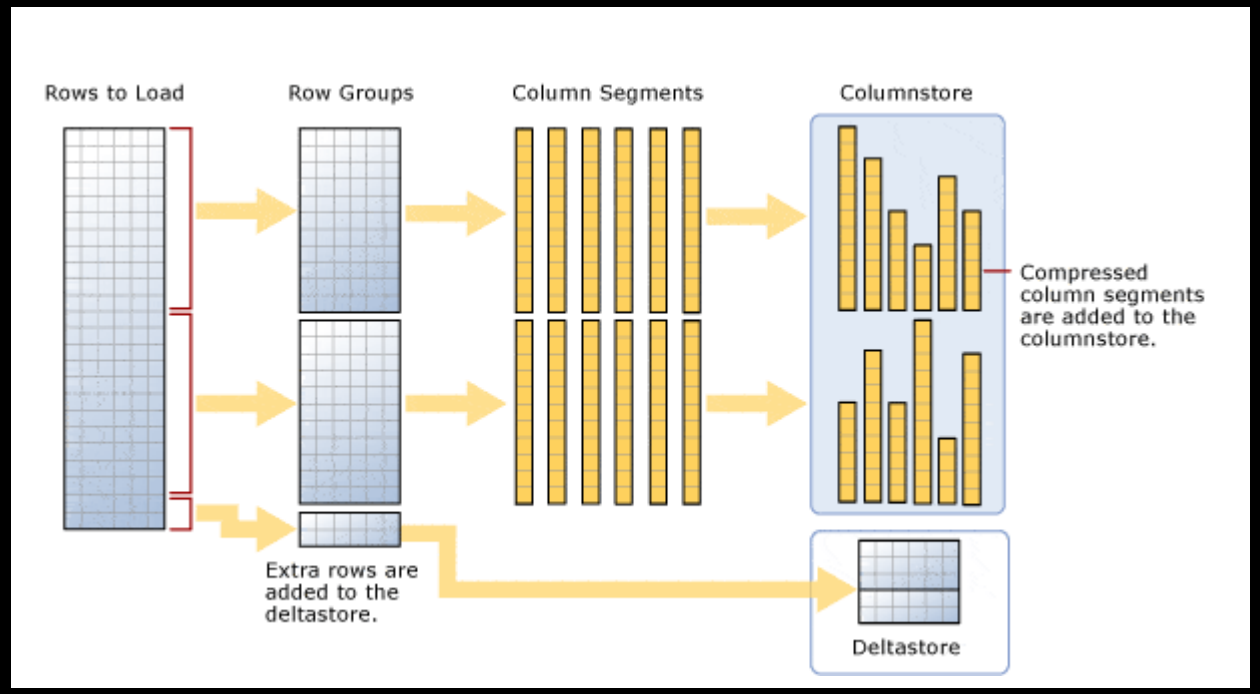
MEMORY_LIMITATION - Not enough available memory to compress all the rows together

DICTIONARY_SIZE - Dictionary size grew too large (exceeded 16MB) to compress all of the rows together

Ordered Clustered Columnstore Indexes

Overview

- Segments can take advantage of improved segment elimination
- Reduce the time needed to service a query
- Requires manual rebuild if data is inserted, updated, or deleted




```sql
-- Create Table with Ordered Columnstore Index
CREATE TABLE sortedOrderTable
(
    OrderId   INT NOT NULL,
    Date      DATE NOT NULL,
    Name      VARCHAR(2),
    Country   VARCHAR(2)
)
WITH
(
  CLUSTERED COLUMNSTORE INDEX ORDER (OrderId)

-- Create Clustered Columnstore Index on existing table
CREATE CLUSTERED COLUMNSTORE INDEX cciOrderId
ON dbo.OrderTable ORDER (OrderId)
```

What determines performance?

Size of compressed row group – a.k.a. row group quality

Compression quality decreases with fewer rows

1M (1,048,576) rows per rowgroup is ideal but < 500K is not desired

<10K is abysmal

Memory pressure when the index was built

Number of rows in deltastore

Deltastore is scanned row-by-row

Is a clustered B-tree index and page compressed

Small batch inserts/deletes/updates end up in deltastore

Singleton inserts/deletes/updates end up in deltastore

Too many partitions

Data movement

Data movement

Overview

Is responsible for the biggest performance impact

Primary goal of the optimization is to reduce the amount of data that needs to be moved to satisfy the query column

Why data is moved

- Incompatible Joins
- Incompatible Aggregations
- Re-distribute Data
- Query Syntax
- Round-robin tables

Common data movement

DMS Operation	Description
Shuffle Move	Redistributes data for compatible join or aggregation *This is the most common move seen in query plans
Broadcast Move	Convert a Distributed table to a Replicated table
Partition Move	Data move from compute to the control node
Round Robin Move	Data gets redistributed as round-robin
Trim Move	Convert a Replicated table to a Distributed table

Materialized View

Materialized View

Overview

A materialized view pre-computes, stores, and maintains its data like a table

Benefits

Automatic and synchronous data refresh with data changes in base tables. No user action is required.

EXPLAIN WITH_RECOMMENDATIONS - provides query plan with recommendations to optimize the SQL statement performance.

```
EXPLAIN WITH_RECOMMENDATIONS
select count(*)
from ((select distinct c_last_name, c_first_name, d_date
       from store_sales, date_dim,
       where store_sales.ss_sold_date_sk = date_dim.d_date_sk
         and store_sales.ss_customer_sk = customer.c_customer_sk
         and d_month_seq between 1194 and 1194+11)
       except
      (select distinct c_last_name, c_first_name, d_date
       from catalog_sales, date_dim, customer
       where catalog_sales.cs_sold_date_sk = date_dim.d_date_sk
         and catalog_sales.cs_bill_customer_sk =
customer.c_customer_sk
         and d_month_seq between 1194 and 1194+11)
) top_customers
```

Workload Management

Workload Management

Classification

To assign a request to a workload group and setting importance levels.

Importance

To influence the order in which a request gets access to resources.

Isolation

To reserve resources for a workload group.

Monitoring

Monitoring

SQL Pool Dynamic Management Views (DMVs)

sys.dm_pdw_sql_requests

sys.dm_pdw_exec_requests

sys.dm_pdw_request_steps

sys.dm_pdw_exec_sessions

sys.dm_pdw_workers

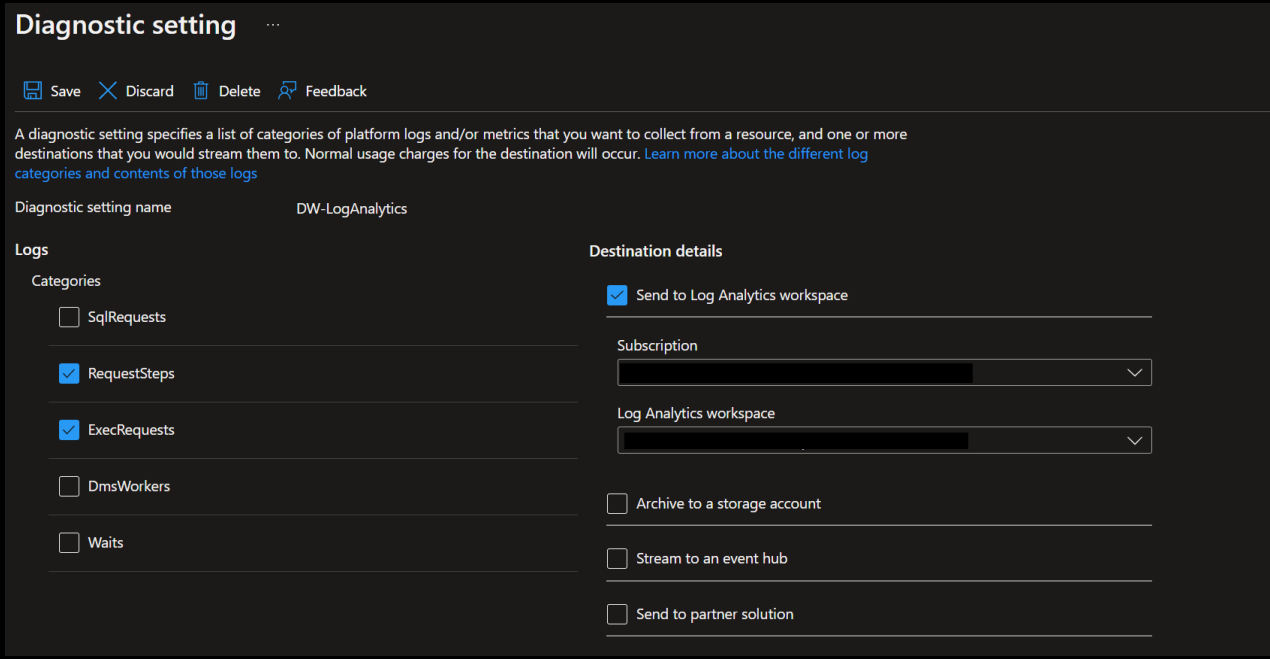
sys.dm_pdw_wait

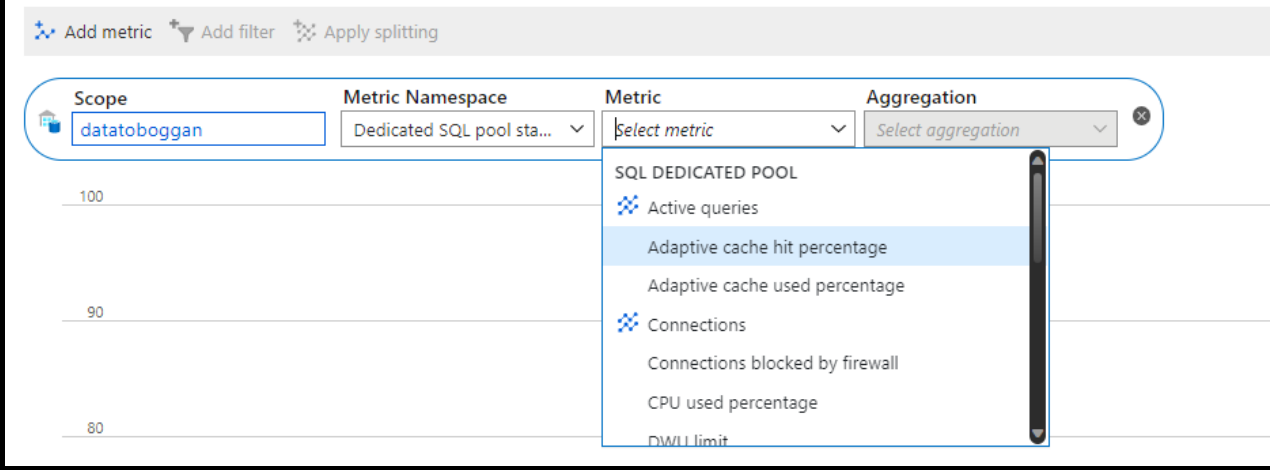
Log Analytics

Ideal for request steps e exec requests with history, cost controlled

Azure Metrics

Cost free





Monitoring

Power BI

Microsoft

Synapse analyzer

Tables with issues | All Tables

Schema

dbo

Distribution Type

ROUND_ROBIN

6

Tables with issues

7

Total tables

85.7%

of tables have issues

Table count by issue

Missing Stats 6

Outdated Statistics 6

Row groups trimmed by table

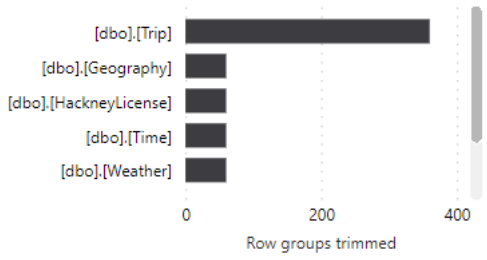
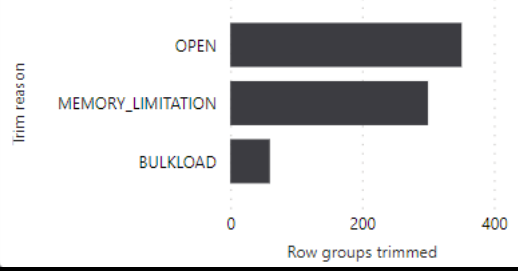


Table count by distribution policy

Dist. Policy	CLUSTERED COLUMNSTORE	Total
ROUND_ROBIN	6	6
Total	**6**	**6**

Row groups by trim reason



Tables with issues - Right click to drill through for more detail

Schema.Table	Index Type	Distribution Policy	Rows	Size (MB)	Data Skew %	Issue	More Info
[dbo].[Geography]	CLUSTERED COLUMNSTORE	ROUND_ROBIN	305K	29.16	2.9%	Missing Stats	
[dbo].[Geography]	CLUSTERED COLUMNSTORE	ROUND_ROBIN	305K	29.16	2.9%	Outdated Statistics	
[dbo].[Weather]	CLUSTERED COLUMNSTORE	ROUND_ROBIN	526K	28.15	3.4%	Missing Stats	
[dbo].[Weather]	CLUSTERED	ROUND_ROBIN	526K	28.15	3.4%	Outdated	

Common Issues

Common Issues

Issue	Action
Data movement	Look at table geometry & statistics
Tempdb usage	Look at queries with excessive data movements
Query queueing	Look at Workload Management configuration
High CPU usage	Optimize queries
Build Replicated table	Check table size
CCI Bad Quality	Check minimum row number, use Ordered CCI

Taxi NYC
Demo

Q & A

Resources

Documentation

Azure Synapse Analytics dedicated SQL Pool documentation

What is dedicated SQL pool (formerly SQL DW)? - Azure Synapse Analytics | Microsoft Docs

Best practices for dedicated SQL pools - Azure Synapse Analytics | Microsoft Docs

Online Free Training

Microsoft Learn

https://docs.microsoft.com/en-us/learn/

Quickstarts

Tutorial: Get started analyze data with dedicated SQL pools - Azure Synapse Analytics | Microsoft Docs

Azure Synapse Analytics Github repo

GitHub - microsoft/MCW-Azure-Synapse-Analytics-and-AI: Microsoft Cloud Workshop Azure Synapse Analytics and AI

GitHub - Azure-Samples/Synapse: Samples for Azure Synapse Analytics

Thank you!

Register for more Data Toboggan Cool Runnings '22 sessions

@datatoboggan.co.uk/coolrunnings2022