Azure Synapse Analytics

========================

Need for a datawarehouse

In our databases - oracle db, mysql

we use database for our day today transactions

Database - day today transactions

- reporting (analysis)

why we should not perform analysis of our data in database

1. performing analysis of data can overburden the database

2. A datawarehouse is specially build for analysis

3. keeping historical data in database can be a big challenge

Azure synapse analytics - it is much more than a datawarehouse

Every synapse workspace should be associated with one storage account - adls gen2

Azure synapse - session 2

===========================

what is azure synapse analytics?

unified analytics service which brings together

data integration

enterprise level datawarehousing

big data analytics

Ingestion - synapse pipeline, mapping dataflow

external source -> INGEST -> ADLS gen2 -> COMPUTE -> ADLS gen2 ->

INGESTION - synapse pipeline, mapping dataflow

COMPUTATION - dedicated sql pool, serverless sql pool, apache spark pool, mapping dataflows dedicated sql pool is more like your redshift in aws

serverless sql pool is more like your athena in aws - it charges us $5 for 1 TB of data scanned.

apache spark spool - it will process on a spark cluster

mapping dataflows - we already know this.

Connected services - Power BI

Azure synapse - session 3

==========================

serverless sql pool

dedicated sql pool

apache spark pool

serverless sql pool

=====================

how to upload a file using synapse studio - ADLS gen2 account

we have seen how to query this file using OPENROWSET where we need not create any entity - table

External table & a normal table

External table

===============

where your metadata is stored in DWH

and data is stored in ADLS gen2

metadata (DWH) + data (Datalake)

Normal table

=============

DWH stores the metadata and data both...

In case of serverless SQL pool we can have only an external table and we can never have a normal table

Azure synapse - session 4

===========================

SQL Pool

1. dedicated sql pool

2. serverless sql pool

there is a already available sql pool named

built-in - serverless sql pool

$5 per tb of data scanned...

even if you are scanning less than 10 mb it will atleast charge you for 10 mb

if you are scanning 5kb of data -

(10mb/1tb) * $5

how to create an external table

==================================

create external table orders (

order_id int,

...

...

...

)

with (

Location = "/xyz/orders.csv"

DATA_SOURCE = ""

FILE_FORMAT = ""

)

ADF -

linked service

dataset

1. create a datasource

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Xyz@123#$';

CREATE DATABASE SCOPED CREDENTIAL SasToken

WITH IDENTITY='SHARED ACCESS SIGNATURE'

, SECRET =

'sv=2020-08-04&ss=b&srt=sco&sp=rwdlacx&se=2022-05-27T02:39:58Z&st=2
022-05-26T18:39:58Z&spr=h

ttps&sig=cFumoIzqc0FeIWeIlLuayon6EO5pfWs7uWV12M4iJ2c%3D';

CREATE EXTERNAL DATA SOURCE ExtDataSrc

WITH (LOCATION = 'https://trendytechsa101.dfs.core.windows.net/data',

CREDENTIAL = SasToken)
```

2. create a file format

```
CREATE EXTERNAL FILE FORMAT TextFileFormat WITH (

FORMAT_TYPE = DELIMITEDTEXT,

FORMAT_OPTIONS (

FIELD_TERMINATOR = ',',

FIRST_ROW = 2))
```

3. create external table

```
CREATE EXTERNAL TABLE orders (

order_id bigint,

order_date nvarchar(4000),

customer_id bigint,

order_status nvarchar(4000)

)

WITH (
LOCATION = 'orders.csv',

DATA_SOURCE = ExtDataSrc,

FILE_FORMAT = TextFileFormat

)

GO
```

Azure synapse - session 5

===========================

Serverless sql pool - external tables

dedicated sql pool - external tables, internal tables

=============

use cases for serverless sql pool

for adhoc data exploration

logical datawarehouse

$5 per tb of data scanned - minimum charges for 10 mb.

csv file

2 tb - 200 columns

select col1, col2 from this table

we should opt for parquet format so that we scan less data...

T-SQL for querying

we do not have to perform an ETL to bring the data to your datawarehouse.. rather we can directly

process the data at the source location...

federated queries...

in serverless SQL pool we saw 2 things

1. OPENROWSET - this is to directly query a file without creating any entity on top...

2. External table

Azure synapse - session 6

===========================

Need for a datawarehouse

is azure synapse a datawarehouse?

ingestion - synapse pipelines

computation - dedicated sql pool, serverless sql pool, apache spark cluster, synapse

dataflows

serving layer - dedicated sql pool

connected services - power bi

Serverless SQL Pool

======================

how to query the data which is stored in our datalake (adls gen2)

2 ways

=======

OPENROWSET

external tables -

1. create data source

2. external file format

3. external table

faster data exploration

logical datawarehouse

cost effective way

Data Lakehouse Architecture

=============================

or referred as modern datawarehouse architecture

Datawarehouse

===============

highly curated and structured data

data governance

security aspects

challenge

============

we have a small % of data which is structured

machine learning cannot be done

DataLake

==========

store data in open file formats

scalable and are inexpensive

enables ML and datascience

challenges

=============

data governanace and reliablity of data

we do not get granular access control

not suitable for BI workloads

Datalake + Datawarehouse

various data source -> Ingest -> Datalake -> Datawarehouse -> Power BI

but there is an overhead - we have to manage 2 systems independently

this leads to duplication of data and additional etl activity

data lakehouse

================

brings the best of both systems together

datawarehouse

Datalake

objectives of data lakehouse

================================

1. quick data discovery
2. handle all types of data
3. reducing the etl activity
4. there should not be multiple copies
5. csv, parquet - open file formats
6. storage and compute should be decoupled
7. Integrated security and governance
8. handle BI, machine learning and other use cases
9. should be cost effective
10. acid transactions (Delta format)

synapse

========

through external tables

we can store all kinds of data in adls gen2

databricks, snowflake, aws redshift, azure synapse

a single solution which can handle our computational needs and can be acting as a

serving layer.

Azure synapse - session 7

============================

Dedicated SQL pool

SQL datawarehouse

internally dedicated SQL pool uses a distrbitued query engine.

for what purpose should we use a dedicated sql pool?

should we use it as a processing/computing layer

or we should use as a serving layer

we can use Dedicated SQL pool for both the use cases..

MPP engine (massively parallel processing)

Architecture

==============

1 control node and multiple compute nodes

the underlying data is distributed in various distribution - 60 distributions

DW100c - 1 compute node , 60 gb of ram

DW200c - 1 compute node, 120 gb of ram

DW300c - 1 compute node, 180 gb of ram

DW400c - 1 compute node, 240 gb of ram

DW500c - 1 compute node, 300 gb of ram

DW1000c - 2 compute node, 600 gb of ram

DW2000c - 4 compute nodes, 1200 gb of ram

DW30000c - 60 compute nodes, 18000 gb of ram

DWU - datawarehousing units

each DWU is nothing but some amount of compute, memory and IO'ps bundled together

we get dedicated internal storage for dedicated sql pool (60 fixed distributions)

1 compute node - 60 distributions

2 compute nodes - each compute node should handle 30 distribution

60 compute nodes - each compute node handles 1 distribution

in this case we get the maximum parallelism

Azure synapse - session 8

==========================

Dedicated SQL Pool

it is more like a traditional datawarehouse

facts and dimensions

orders can be the fact table

customers - dimension table

products - dimension table

star schema

=============

customers

|

catergories - orders - products

|

department

fact table will be a large table

dimension table will be smaller table

no concept of a primary key and foreign key

in dedicated sql pool we have 3 types of table distributions

==
==

1. round robin

2. hash

3. replicate

D1 1,4,7

D2 2,5,8

D3 3,6,9

```
1,2013-07-25 00:00:00.0,11599,CLOSED
2,2013-07-25 00:00:00.0,256,PENDING_PAYMENT
3,2013-07-25 00:00:00.0,12111,COMPLETE
4,2013-07-25 00:00:00.0,8827,CLOSED
5,2013-07-25 00:00:00.0,11318,COMPLETE
6,2013-07-25 00:00:00.0,7130,COMPLETE
7,2013-07-25 00:00:00.0,4530,COMPLETE
8,2013-07-25 00:00:00.0,2911,PROCESSING
9,2013-07-25 00:00:00.0,5657,PENDING_PAYMENT
10,2013-07-25 00:00:00.0,5648,PENDING_PAYMENT
11,2013-07-25 00:00:00.0,918,PAYMENT_REVIEW
12,2013-07-25 00:00:00.0,1837,CLOSED
13,2013-07-25 00:00:00.0,9149,PENDING_PAYMENT
14,2013-07-25 00:00:00.0,9842,PROCESSING
15,2013-07-25 00:00:00.0,2568,COMPLETE
16,2013-07-25 00:00:00.0,7276,PENDING_PAYMENT
17,2013-07-25 00:00:00.0,2667,COMPLETE
18,2013-07-25 00:00:00.0,1205,CLOSED
19,2013-07-25 00:00:00.0,9488,PENDING_PAYMENT
20,2013-07-25 00:00:00.0,9198,PROCESSING
```

round robin

============

easy to distribute

but when running query it has to do shuffling when we look to do aggregations or joins..

hash

=====

60 distributions

Hash(customer_id)

1

2

3

4

1

3

5

6

1

7

8

9

1

2

2

hash(1) = 4

hash(1) = 4

takes little time to distribute the data

query time is faster

Replicate

===========

the copy is made on all 60 distributions

small dimension tables

orders table

order_customer_id

customers table

customer_id

D1

orders table

1 1

2 2

3 3

customers table can be replicated accross all distributions

orders table can be hash distribution

map side join, broadcast join

Azure synapse - session 9

===========================

1. upload the data in adls gen2 - orders

2. we want to create a order table in dedicated sql pool

in orders table we want to load the data from datalake

2 options to load the data from datalake to dedicated sql pool

1) polybase

datalake -> control node

cn1 cn2 cn3...............

step 1) first we should create an external table in dedicated sql pool

data source

external file format

external table

```
IF NOT EXISTS (SELECT * FROM sys.external_file_formats WHERE name =
'SynapseDelimitedTextFormat')
CREATE EXTERNAL FILE FORMAT [SynapseDelimitedTextFormat]
WITH ( FORMAT_TYPE = DELIMITEDTEXT ,
FORMAT_OPTIONS (
FIELD_TERMINATOR = ',',
FIRST_ROW = 2,
USE_TYPE_DEFAULT = FALSE
))
GO
IF NOT EXISTS (SELECT * FROM sys.external_data_sources WHERE name
=
'raw_ttsynapsesa_dfs_core_windows_net')
CREATE EXTERNAL DATA SOURCE
[raw_ttsynapsesa_dfs_core_windows_net]
WITH (
LOCATION = 'abfss://raw@ttsynapsesa.dfs.core.windows.net',
TYPE = HADOOP
)
GO
CREATE EXTERNAL TABLE orders_ext (
[order_id] bigint,
[order_date] VARCHAR(4000),
[customer_id] bigint,
[order_status] nvarchar(4000)
)
WITH (
LOCATION = 'orders.csv',
DATA_SOURCE = [raw_ttsynapsesa_dfs_core_windows_net],
FILE_FORMAT = [SynapseDelimitedTextFormat]
)
GO
SELECT TOP 100 * FROM dbo.orders_ext
GO
```

step 2) CTAS

```
create table internal_table as
select * from external table
CREATE table orders_internal_roundrobin
WITH
(
DISTRIBUTION = ROUND_ROBIN
)
AS
select * from orders_ext
select customer_id, count(*) as total_orders from orders_internal_roundrobin group by
customer_id order by total_orders DESC ;
DBCC PDW_SHOWSPACEUSED('orders_internal_roundrobin')
```

2) copy command

```
CREATE table orders_internal_HASH
```

WITH

(

DISTRIBUTION = HASH(CUSTOMER_ID)

)

AS

select * from orders_ext

DBCC PDW_SHOWSPACEUSED('orders_internal_HASH')

select customer_id, count(*) as total_orders from orders_internal_HASH group by

customer_id order by total_orders DESC ;

Azure synapse - session 10

==============================

Dedicated SQL Pool

Azure SQL Datawarehouse

60 distributions

1 control node, and multiple compute nodes

DWU's

2 ways to load the data in Dedicated SQL pool table

1. using polybase

step 1: we create external table

step 2: CTAS

create table as select

create table orders_internal

WITH

(

DISTRIBUTION = ROUND_ROBIN

)

AS

select * from orders_ext

in case of polybase we can move data 2 ways

from dedicated sql pool to data lake - CETAS

data lake to dedicated sql pool - CTAS

2. copy command

Just like polybase it use mpp architecture

can perform even better than polybase

no external objects required

we can use wild card characters in the file path

Distribution types

===================

1. round robin

2. hash

3. replicate

load time query time when to use

round robin - quick high staging tables

hash - high quick for fact tables

replicate - high quick for small dimension tables

customers orders

D1 - 1001,1002,1003 1002

D2 - 1001,1002,1003 1001

D3 - 1001,1002,1003 1003

D4 - 1001,1002,1003

in this case we could have created the customers table as a replica

Azure synapse - session 11

=============================

serverless - compute

dedicated sql pool - compute / serving layer

spark pool - compute

number of nodes - 3

node size - medium (8 vcores, 64 gb RAM)

The cluster is configured but started yet

24 vcores , 192 GB RAM

driver

executor

8 cores and 56 ram is usable from each node

1 gb ram goes for each core as part of overhead

small - 5 (4vcores, 28 GB RAM)

medium - 2

we have 3 medium size nodes , we can have total 6 small size executors and we can have 3 medium

size executors

1 out of executors will go for driver

Notebook

=========

Languages, visualization, a set of resources are attached to the notebook, collaborate, monitoring

Remember to delete the cluster to avoid any charges

Azure synapse - session 12

=============================

1. stop the session

2. terminate the cluster

3. spark tables -

metadata (hive metastore)

+

Data (adls gen2)

```
%%pyspark

df =
spark.read.load('abfss://raw@ttsynapsesa.dfs.core.windows.net/orders.csv',
format='csv'

, header=True

)

df.write.mode("overwrite").saveAsTable("orders_spark_new")
```

4. you have a table in dedicated sql pool

how to read/process this data with spark

spark cluster is taking the data from dedicated sql pool using polybase

driver will ask the control node that it needs the data

control node will pass the instruction to compute node and then compute node writes the data in

parallel in a datalake

from the datalake spark will take the data...

spark scala way of doing

dedicated sql pool -> datalake -> spark -> spark table

hive metastore

5. read/process spark table through serverless sql pool

serverless sql pool cannot access hive metastore

a copy of this metadata is created

spark table -> a copy of metadata will be created

Azure synapse - session 13

==============================

Azure Synapse summary

1. serverless sql pool

openrowset

external tables

2. dedicated sql pool

control, compute

distribution

polybase, copy

3. spark sql pool

spark tables

dedicated sql

reading data in spark tables using serverless sql pool