Oscar Zamora
@ZamoraO
linkedin.com/in/ozamora
ozamora.com

PASS SQLSATURDAY

Azure SQL Data Warehouse Redefining MPP

How it stacks against Snowflake & RedShift

Microsoft CERTIFIED
AZURE DATA ENGINEER
ASSOCIATE

Agenda

Agenda

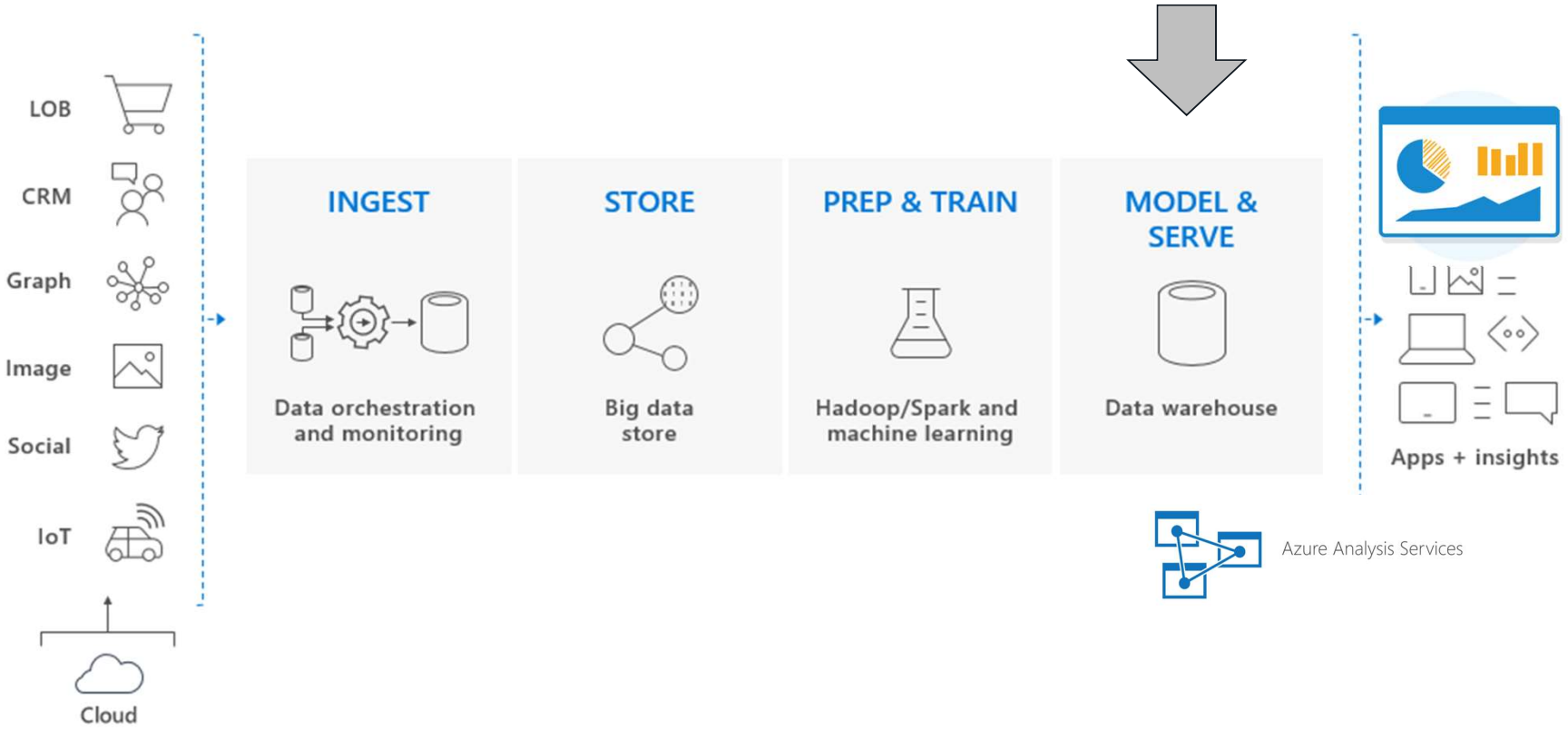
- What is Azure SQL Data Warehouse
- Azure SQL Data Warehouse Architecture
- Key differences with SQL Server
- Compare against Snowflake
- Compare against Redshift
- Azure SQL DW T-SQL Sample for loading data
- Q & A Session

What is Azure SQL Data Warehouse

What is Azure SQL Data Warehouse

- A cloud iteration of APS (formerly PDW)
- A cloud-based Enterprise Data Warehouse (EDW) that uses Massively Parallel Processing (MPP)
- Allows running complex SQL across terabytes of data leveraging multiple nodes
- Accepts fast ingestion of data using Polybase
- With scalable compute and limitless storage

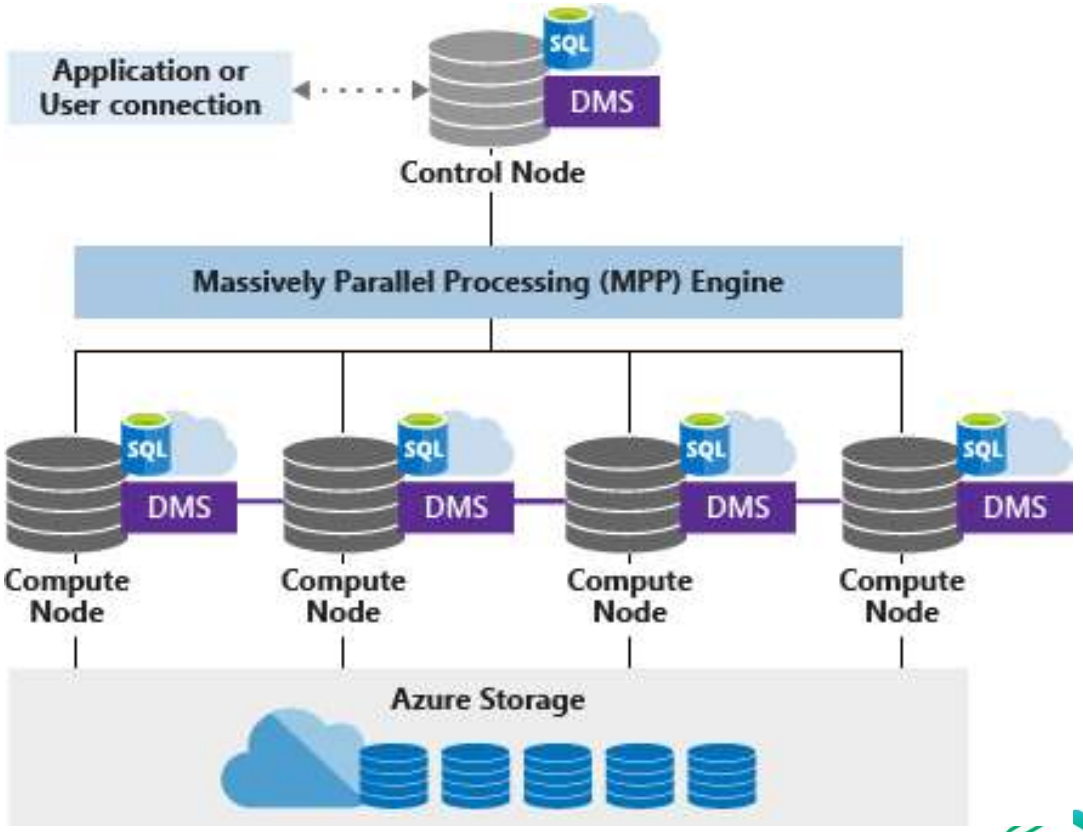
What is Azure SQL Data Warehouse



Azure SQL Data Warehouse Architecture

Azure SQL Data Warehouse Architecture

- Node based
  - Control Node
  - Compute Nodes
- Decoupled Storage
  - Distributed Datafiles
  - Blob Storage
  - Local NVMe caching



Azure SQL Data Warehouse Compute

- Control Node:
  - Single Master funnels queries and returns data
  - Governed by AAD for authentication and RBAC
  - Creates query plan and distributes queries
  - Aggregates/combines results and sends back to client
- Compute Nodes:
  - Perform parallel work (1 to 60 nodes).
  - Interact with dedicated databases.
  - Caches data on local NVMe Storage (gen. 2)

Azure SQL Data Warehouse Storage

- Independent to the control and compute nodes
- Data located in 60 distributions
- Distributions attached to 1 to 60 compute nodes
- Leverages Azure BLOB behind the scenes
- Data is Geo-replicated
- 3 types of data distributions: Hash, Round-Robin, Replicated

Azure SQL DW Latest Optimizations

- Result-set caching
- Materialized Views
- Ordered Clustered Columnstore Index
- Workload Importance (priorization)
- Read Committed Snapshot Isolation
- JSON support
- Dynamic Data Masking

Key differences with SQL Server

Key Differences with SQL Server

SQL Server	Azure SQL DW
Single Node, Multi-processor architecture	Massively Parallel Processing architecture
Requires attached storage or SAN	Storage is decoupled from the node(s)
Supports advanced development (CLR, R)	Most (not all) T-SQL capabilities
Data does not need to conform to distributions	Data needs to be distributed to operate efficiently
Scale Up capabilities (CPU clock, RAM)	Scale Out Capabilities (up to 60 nodes)
Caching limited by RAM	Caching increases as node count is increased
Highly operational DBA tasks (on premises), for backups, maintenance, etc	PaaS platform minimizes operational work

Key Differences with SQL Server

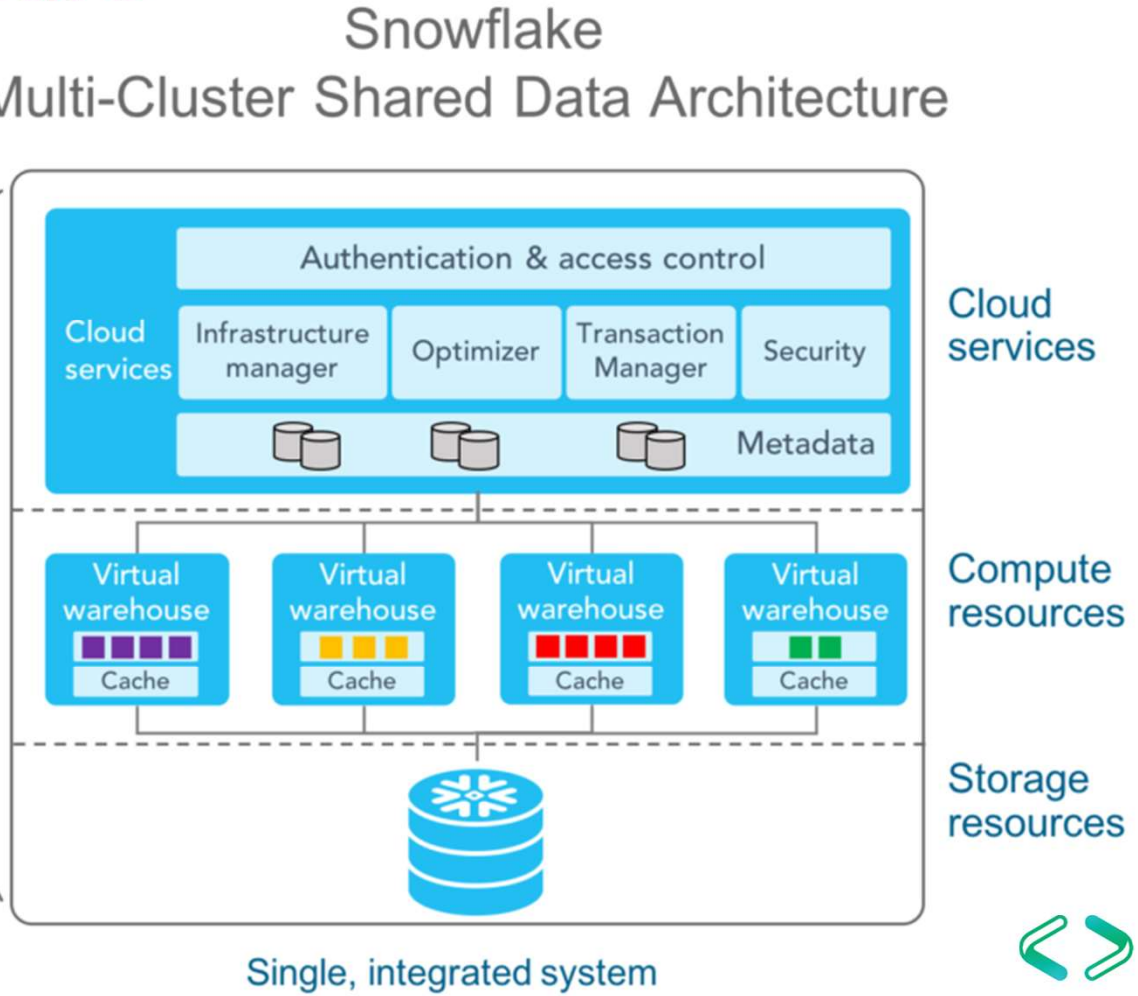
SQL Server	Azure SQL DW
OLTP and OLAP implementations (limited by resources)	OLAP based implementations with multi terabyte/petabyte scale data storage
Highly configurable (partitioning, compression, indexed views, computed, columns, columnar, etc.)	Techniques around data distribution and storage, only.
Available Integration, Reporting and Analytical Engines	Just an MPP engine. Additional services require Azure offerings.
Best for application based services, and Reporting of medium-sized data	Best for very large data analytics, large scale ELT, and data mining services.

Azure SQL DW vs. Snowflake

snowflake

Snowflake Architecture

- 3 layers
  - Cloud Services
  - Compute
  - Storage



Architecture Side by Side

Azure SQL DW	Snowflake
Multiple Compute Nodes 1 Control Node 1 Cluster (up to 60 nodes) Decoupled Storage	Multiple Compute Nodes Multi-Master Multi-Cluster (up to 128 nodes / 10 clusters) Decoupled Storage
50+ Regions on Azure Geo-Replication included by default	7 Regions Under Azure 6 Regions Under AWS Geo Replication is now supported (option)
Increase/decrease Compute Nodes: manual	Increase/decrease Compute Nodes: manual Spin up/down clusters: automatic (Enterprise+)
Database Files accessed by Compute Node(s) Database Files on Blob Storage	Shared Data Architecture Database Objects on Blob Storage
Insert, Update, Delete operations append and flag data within data files. Index maintenance cleans.	Insert, Update, Delete operations create new Blob objects, and updates metadata
Compression: Heap and Column-Store Indexes	Compressed data and Columnar store by default No rowstore option

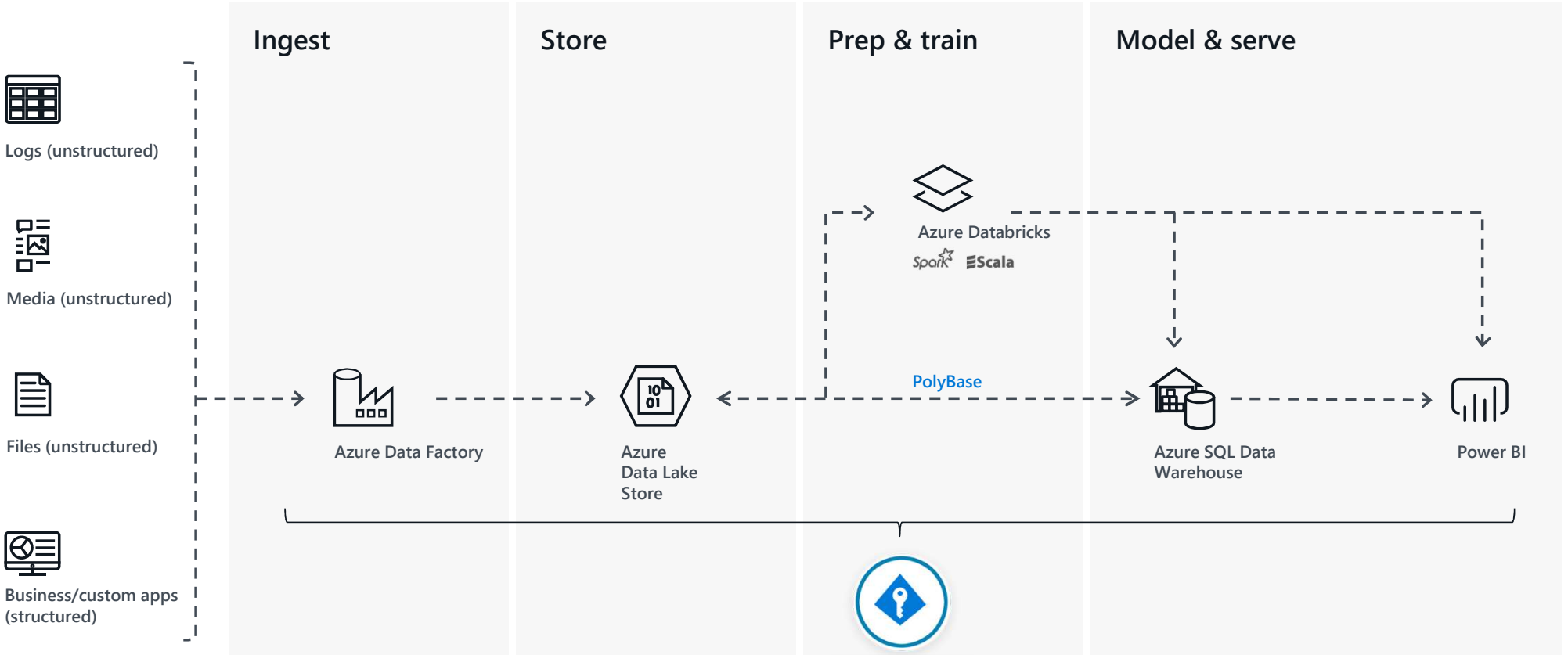
Architecture Side by Side

Azure SQL DW	Snowflake
Automatic Distribution of Data. 3 options: • Hash • Round-Robin • Replicate	No Distribution Keys
Indexing support	No Indexes supported
Customized Partitioning Support	Automatic micro-partitions (not user-configurable)
Statistics maintenance is possible and suggested	No Statistics maintenance
Clustered index column possible on heap and column store indexed tables.	Unmanaged Clustered data. Re-clustering optionally and strongly suggested for very large tables/heavy DML
Per Cluster Limit: Up to 125 queries running concurrently (gen 2).	Per Cluster Limit: Up to 80 queries running concurrently: • 8 queries running concurrently per Cluster • 10 Clusters per Virtual Warehouse allowed

Pricing Models

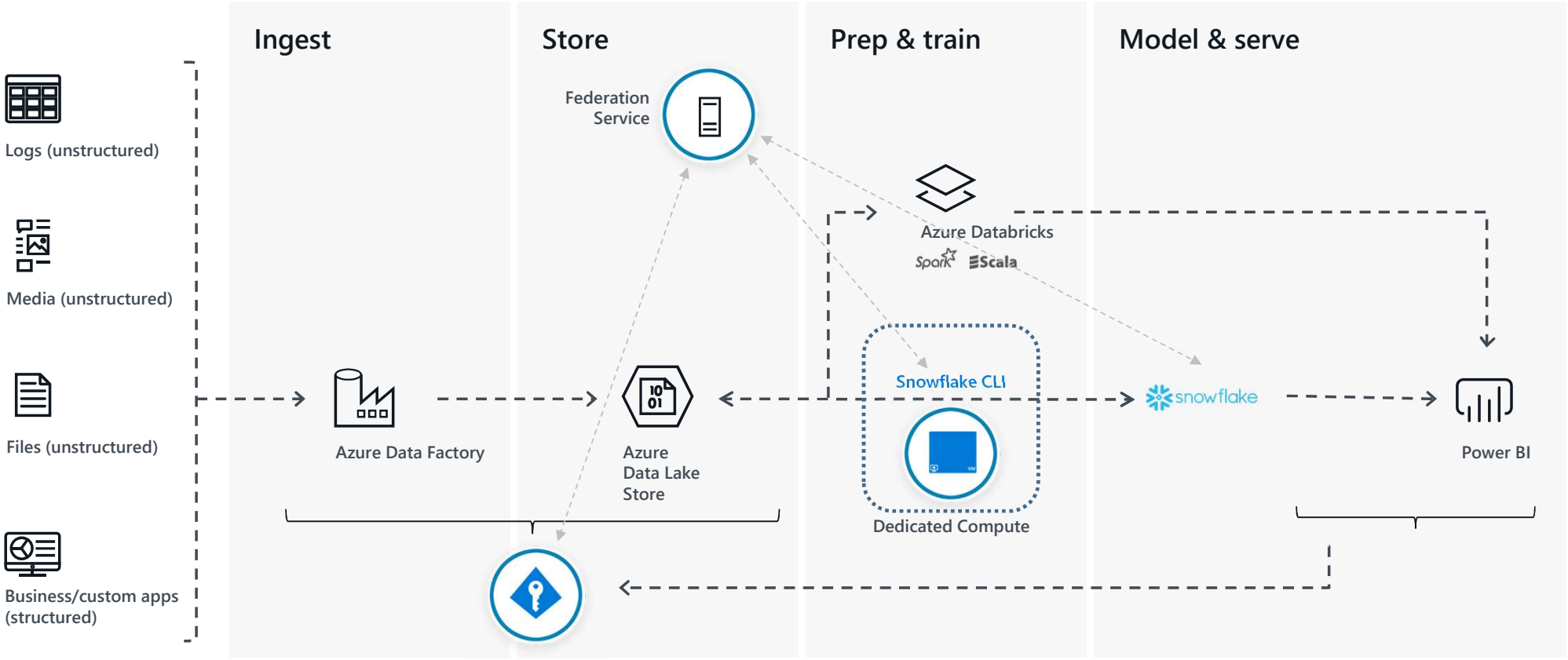
Azure SQL DW	Snowflake
Per-Hour Billing	Per-Second Billing
Per-Terabyte pricing: • Data (compressed) • 7-day incremental snapshots • Geo-replicated Data	Per-Terabyte pricing: • Data (compressed), • Fail-Safe (7 days), • Time-Travel (default 1 day, up to 90)
Data Ingress unlimited, no-cost	Data Ingress unlimited, no-cost
Data Egress Charged per GB per Region	Data Egress Charged per GB per Region

Reference Architecture – Azure SQL DW



Microsoft Azure also supports other Big Data services like Azure HDInsight and Azure Data Lake to allow customers to tailor the above architecture to meet their unique needs.

Reference Architecture - Snowflake

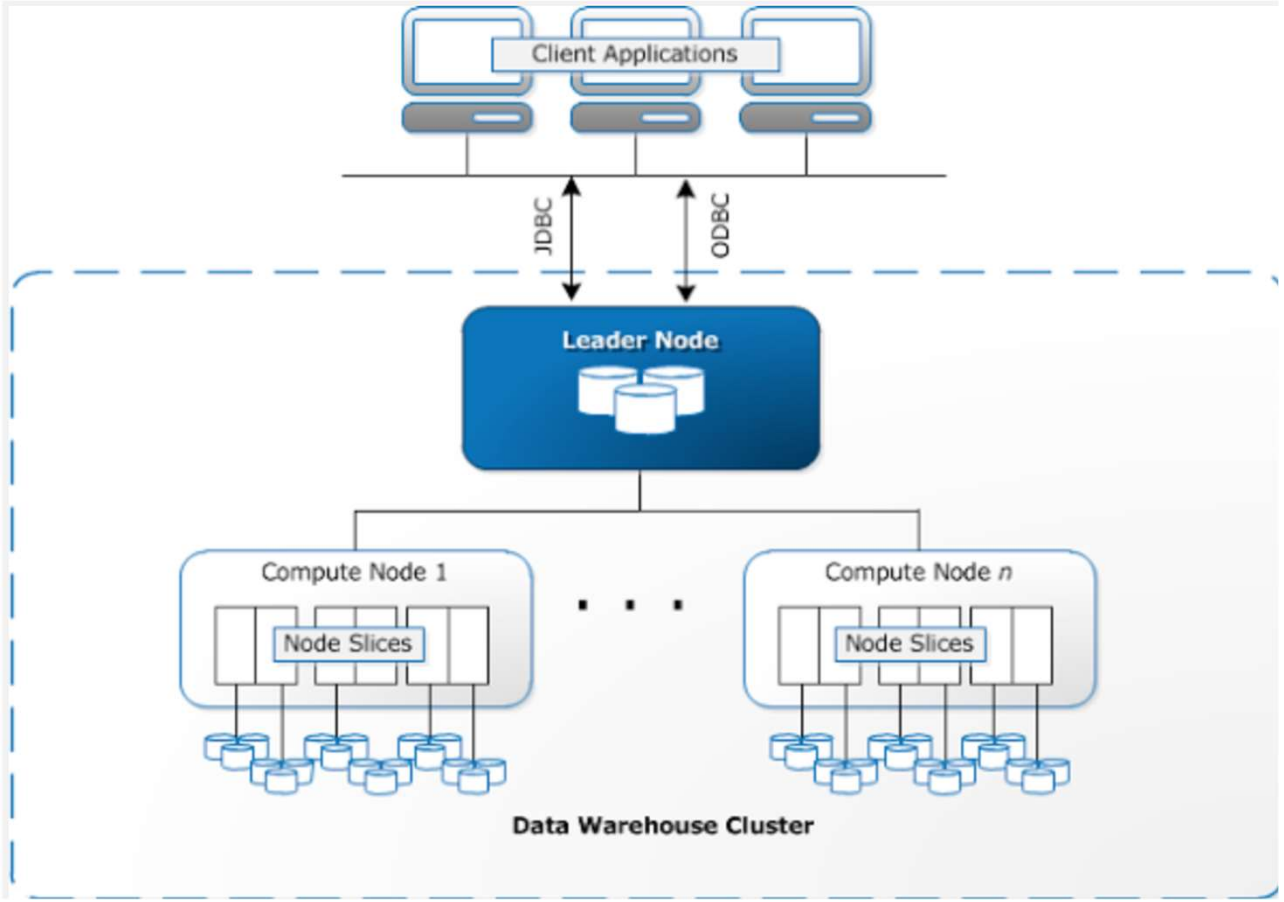


Microsoft Azure also supports other Big Data services like Azure HDInsight and Azure Data Lake to allow customers to tailor the above architecture to meet their unique needs. Federation Trust needs to be enabled for AAD accesss to Snowflake, for login purposes. At the user level, only.

Azure SQL DW vs. Redshift

Redshift Architecture

- 2 layers
  - Leader Node
  - Compute Node + attached storage



Architecture Side by Side

Azure SQL DW	Redshift
Multiple Compute Nodes 1 Control Node 1 Cluster (up to 60 nodes) Decoupled Storage	Multiple Compute Nodes 1 Control Node 1 Cluster (up to 32 nodes) Attached Storage
50+ Regions Geo-Replication included	22 Regions Multiple AZs replication included
Increase/decrease Compute Nodes: manual Compute can be paused anytime	Node resizing is manual. Data requires redistribution @ 1 TB per hour. Reserve instance pricing does not allow pause
Database Files accessed by Compute Node(s) Database Files on Blob Storage	Database objects attached to the Compute Node(s) Database objects are not decoupled from compute
Insert, Update, Delete operations append and flag data within data files. Index maintenance cleans.	Update and deletes flag data. Vacuum required Updates and upserts are not trivial operations.
Compression: Heap and Column-Store Indexes	Columnar data compression. Several algorithms available depending on type of data.

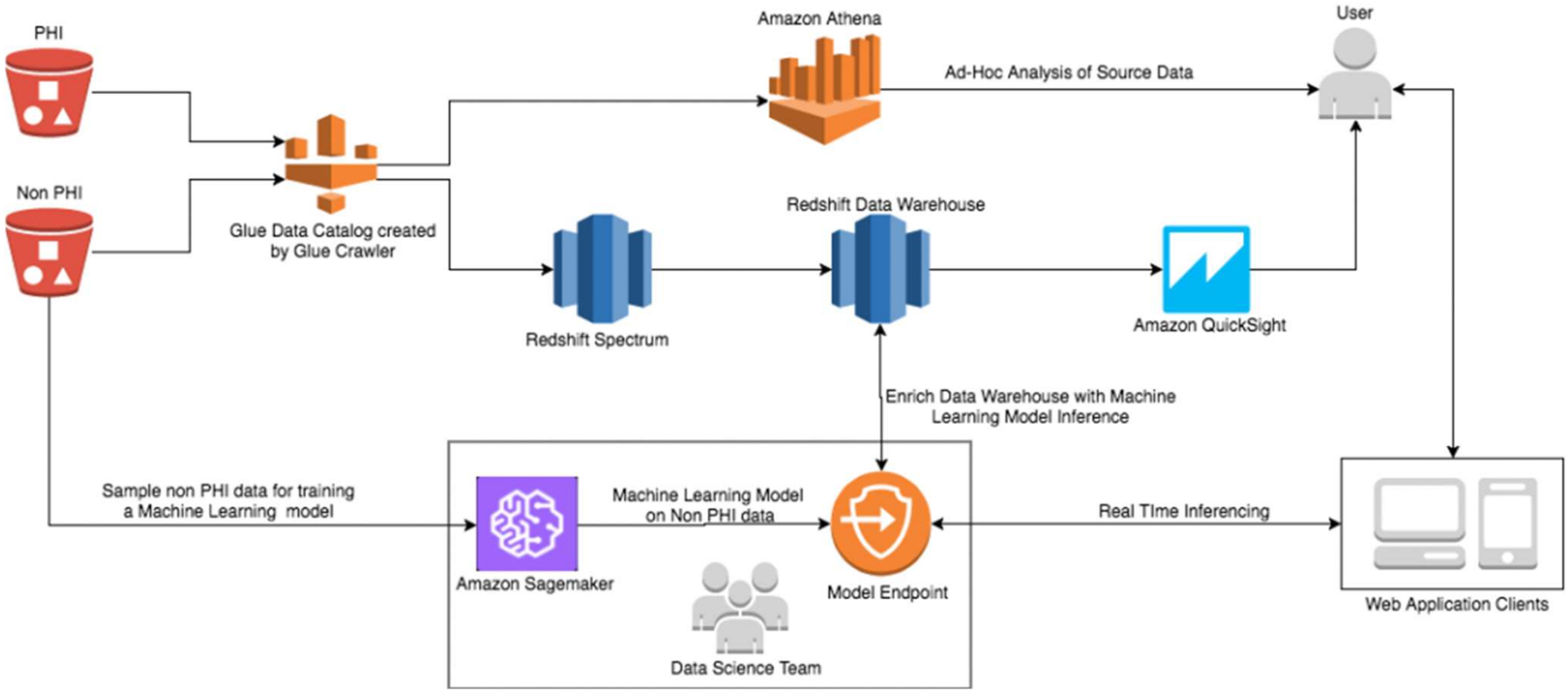
Architecture Side by Side

Azure SQL DW	Redshift
Automatic Distribution of Data. 3 options: • Hash • Round-Robin • Replicate	4 options: • Key (like Hash) • Even (like Round-Robin) • All (like Replicate) • Auto (selected based on size of data)
Indexing support	1 sort key supported
Customized Partitioning Support	No partitioning
Statistics maintenance is possible and suggested	Vacuuming is required to keep performance
Clustered index column possible on heap and column store indexed tables.	No data clustering support
Per Cluster Limit: Up to 125 queries running concurrently (gen 2).	Per Cluster Limit: Up to 50 queries running concurrently

Pricing Models

Azure SQL DW	Redshift
Per-Hour Billing	Per-Hour Billing
Per-Terabyte pricing: • Data (compressed) • 7-day incremental snapshots • Geo-replicated Data	Data is included per node Dense compute: 0.16TB / 2.56TB Dense Storage: 2TB HDD / 16TB HDD Data is replicated within AZs
Data Ingress unlimited, no-cost	Data Ingress unlimited, no-cost
Data Egress Charged per GB per Region	Data Egress Charged per GB per Region

Reference Architecture - Redshift



Price Performance Comparison

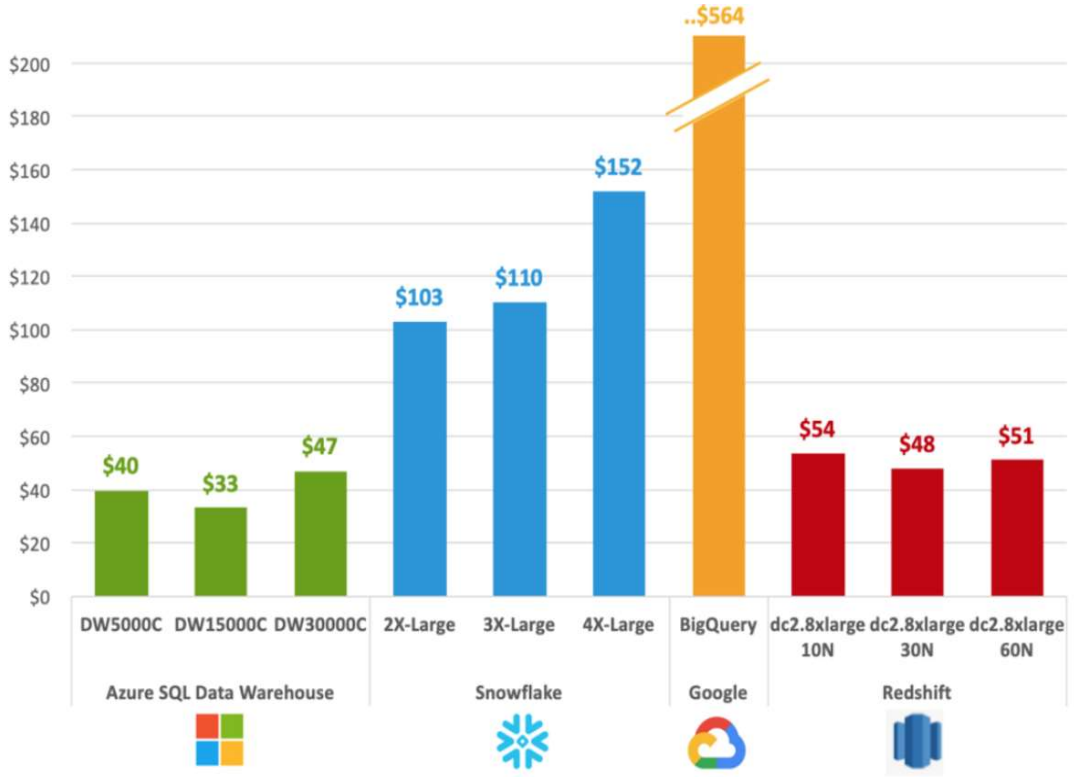
1 TB Query Response Comparison

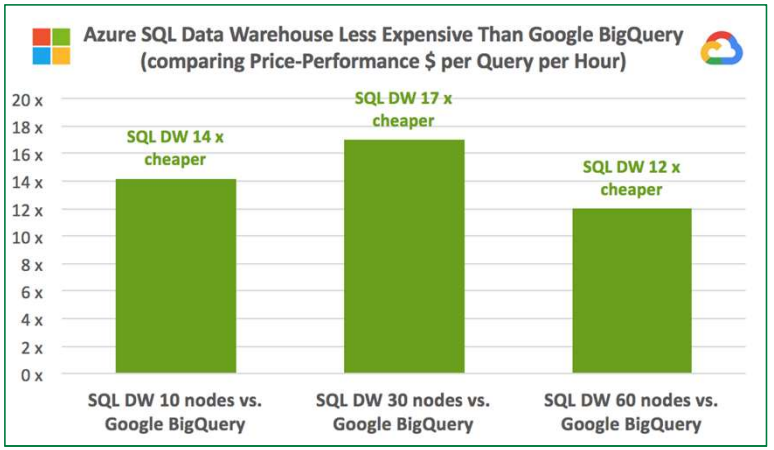
Source:
https://fivetran.com/blog/warehouse-benchmark

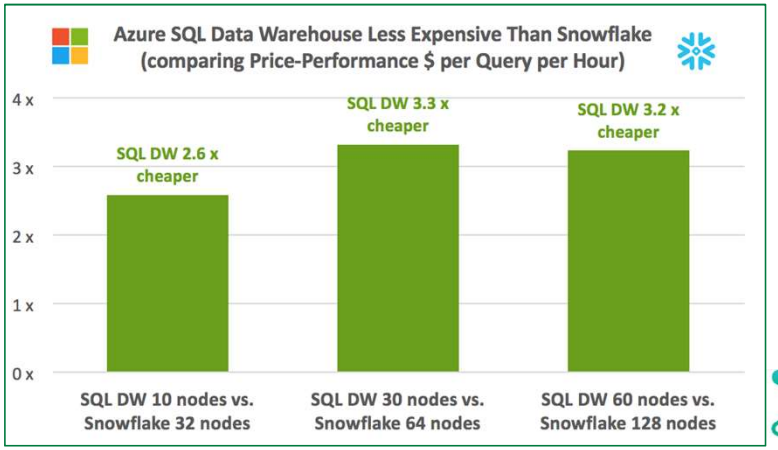
Industry-leading price performance

Source: GigaOm TPC-H 30TB Cloud DW Benchmark (February 2019)

Price-Performance @ 30TB ($ per Query per Hour)
(Lower is Better)



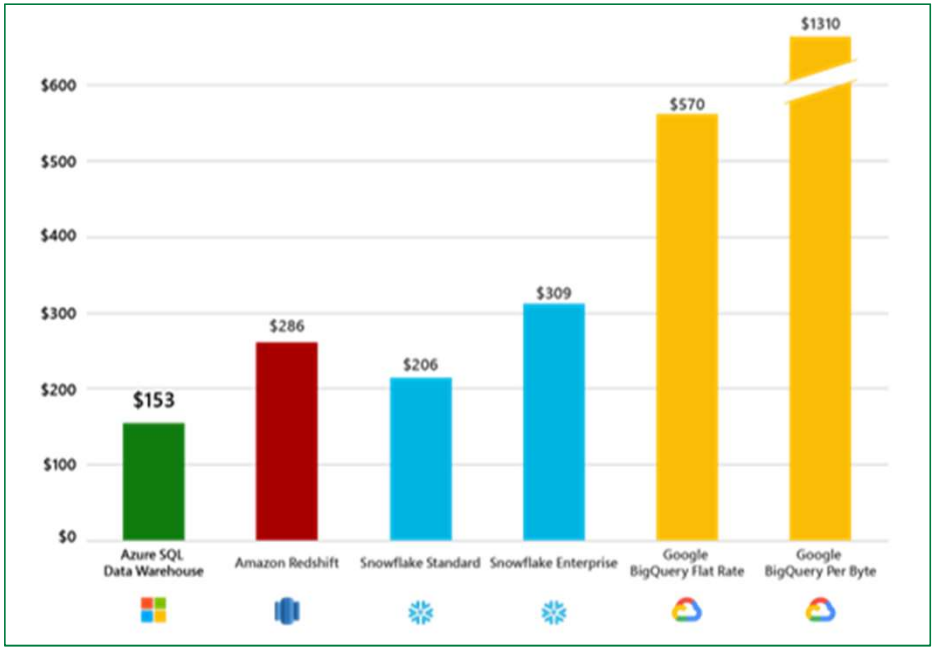




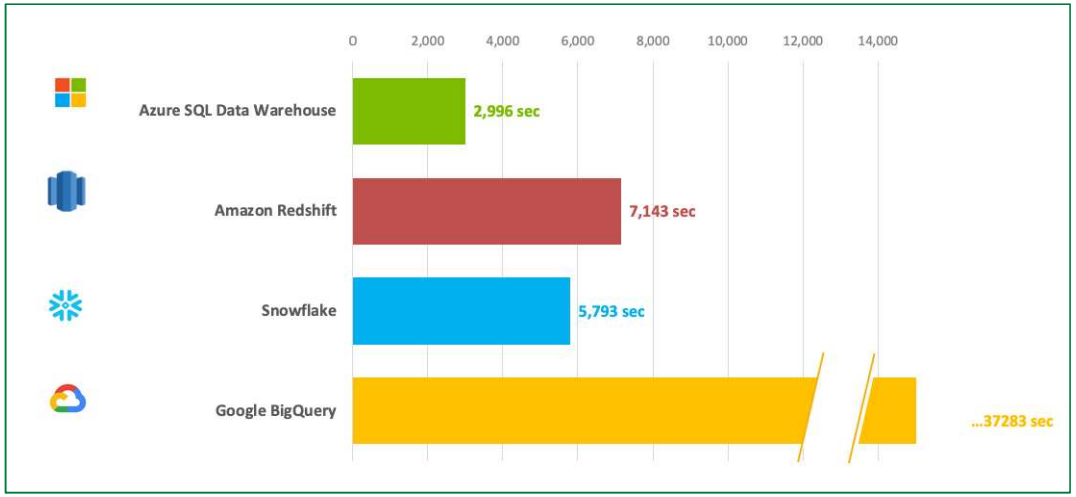
Industry-leading price performance (TPC-DS)

Source: GigaOm TPC-DS Cloud DW Benchmark (April 2019)

Price-Performance @ 30TB ($ per Query per Hour)
(Lower is Better)



Aggregate Execution Time for TPC-DS Queries
(Lower is Better)



Why Azure SQL Data Warehouse?

Enterprise Grade – Multi-Petabyte Scale

- ADW has the suite of features needed to handle an enterprise-grade workload
- ADW is a SQL-based fully managed, petabyte-scale cloud data warehouse.
- It is highly elastic, enabling provision in minutes and scale capacity in seconds.
- Independent scaling for compute and storage, allows you to burst compute for complex analytical workloads or scale down your warehouse for archival scenarios.
- It integrates easily with products such as Azure Analysis services, HDInsight and Data Lake, offering an end-to-end solution for all your data warehousing needs.
- Market leading 99.9% SLA combined with innovative security features provides additional reasons why enterprises should select SQL DW as their go-to cloud based DW product.

Azure SQL Data Warehouse Sample T-SQL Data Loading

Azure SQL DW – Create User for Data Load

```
-- Execute on Master Database
CREATE LOGIN ozamora WITH PASSWORD = 'iujYTG855w0rd!';
CREATE USER ozamora FROM LOGIN ozamora;  -- To create a SQL Server user based on a SQL Server authentication login

-- Execute on target database [DBNAME]
CREATE USER ozamora FROM LOGIN ozamora;  -- To create a SQL Server user based on a SQL Server authentication login
GRANT CONTROL ON DATABASE::[DBNAME] to ozamora;
--EXEC sp_droprolemember 'xlargerc', 'ozamora';
--EXEC sp_addrolemember 'staticrc50', 'ozamora';
--EXEC sp_droprolemember 'staticrc50', 'ozamora';
EXEC sp_addrolemember 'staticrc60', 'ozamora';
GRANT VIEW DATABASE STATE TO ozamora;

-- Login as user
```

Azure SQL DW – Create Table

```
CREATE TABLE [FINANCE].[TRXN_LINE_DIST]
(
   [DIST_KEY_ID] BIGINT  NULL,
   [DIST_DISCRIM_TYP] varchar(30)  NULL,
   [GL_PERIOD_START_DT] datetime2(0)  NULL,
   [ENTERED_AMT] numeric(28, 7)  NULL,
   [ENTERED_CURRENCY_CD] varchar(15)  NULL,
   [ACCOUNTED_AMT] numeric(28, 7)  NULL,
   [DW_CREATE_DT] datetime2(0)  NULL,
   [DW_LAST_MODIFY_DT] datetime2(0)  NULL
)
 WITH ( CLUSTERED COLUMNSTORE INDEX, DISTRIBUTION = HASH(DIST_KEY_ID),
    PARTITION ( GL_PERIOD_START_DT  RANGE RIGHT FOR VALUES
           ( '2018-01-01', '2018-04-01', '2018-07-01', '2018-10-01' )
           )
 )
```

Azure SQL DW – Enable Polybase

```
CREATE MASTER KEY;

CREATE DATABASE SCOPED CREDENTIAL MyKey
 WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
 SECRET = 'A4uZYbv8xaYL+RNdAHSl6y/GpyuhvF4h8D7IUHPfcAeKlWwKoAch3Sn2etDaZu7gGtY/KoOhVF99Loh6yBhj0w==';

CREATE EXTERNAL DATA SOURCE EXTLLD
WITH
(
    TYPE = HADOOP,
    LOCATION = 'wasbs://cilenstage@stage169887.blob.core.windows.net',
     CREDENTIAL = MyKey
);

CREATE EXTERNAL FILE FORMAT TextFileFormat
WITH  (FORMAT_TYPE = DELIMITEDTEXT, FORMAT_OPTIONS
                (FIELD_TERMINATOR = '0x01', STRING_DELIMITER = '"', Encoding = 'UTF8',-- DATE_FORMAT
= 'yyyy-MM-dd HH:mm:ss.fff',
                USE_TYPE_DEFAULT = FALSE),
                DATA_COMPRESSION = 'org.apache.hadoop.io.compress.GzipCodec'
          );
GO
```

Azure SQL DW – Create External Table

```
CREATE EXTERNAL TABLE [FINANCE].[EXT_TRXN_LINE_DIST]
(
    [DIST_KEY_ID] BIGINT  NULL,
    [DIST_DISCRIM_TYP] varchar(30)  NULL,
    [GL_PERIOD_START_DT] datetime2(0)  NULL,
    [ENTERED_AMT] numeric(28, 7)  NULL,
    [ENTERED_CURRENCY_CD] varchar(15)  NULL,
    [ACCOUNTED_AMT] numeric(28, 7)  NULL,
    [DW_CREATE_DT] datetime2(0)  NULL,
    [DW_LAST_MODIFY_DT] datetime2(0)  NULL
)
WITH ( LOCATION='/FINANCE/LINE_DIST/', DATA_SOURCE = EXTLLD,
FILE_FORMAT = TextFileFormat,  REJECT_TYPE = percentage, REJECT_VALUE
= 5,  REJECT_SAMPLE_VALUE = 100 );
```

Azure SQL DW – Load Data

```
INSERT INTO [FINANCE].[TRXN_LINE_DIST]
SELECT * FROM [FINANCE].[EXT_TRXN_LINE_DIST];

-- CTAS Alternative
CREATE TABLE [FINANCE].[TRXN_LINE_DIST]
WITH ( CLUSTERED COLUMNSTORE INDEX, DISTRIBUTION = HASH(DIST_KEY_ID),
    PARTITION ( GL_PERIOD_START_DT  RANGE RIGHT FOR VALUES
          ( '2018-01-01', '2018-04-01', '2018-07-01', '2018-10-01' )
          )
 )
AS
SELECT * FROM [FINANCE].[EXT_TRXN_LINE_DIST];
```

Q & A Session

References

- https://docs.microsoft.com/en-us/azure/sql-data-warehouse/release-notes-10-0-10106-0
- https://azure.microsoft.com/en-us/blog/adaptive-caching-powers-azure-sql-data-warehouse-performance-gains/
- https://docs.aws.amazon.com/general/latest/gr/rande.html
- https://docs.microsoft.com/en-us/sql/t-sql/statements/create-external-file-format-transact-sql?view=sql-server-2017
- https://azure.microsoft.com/en-us/blog/azure-sql-data-warehouse-releases-new-capabilities-for-performance-and-security/
- https://docs.microsoft.com/en-us/azure/sql-database/sql-database-features
- http://microsoft-bitools.blogspot.com/2017/07/azure-sql-database-vs-azure-sql-data.html


Thank You