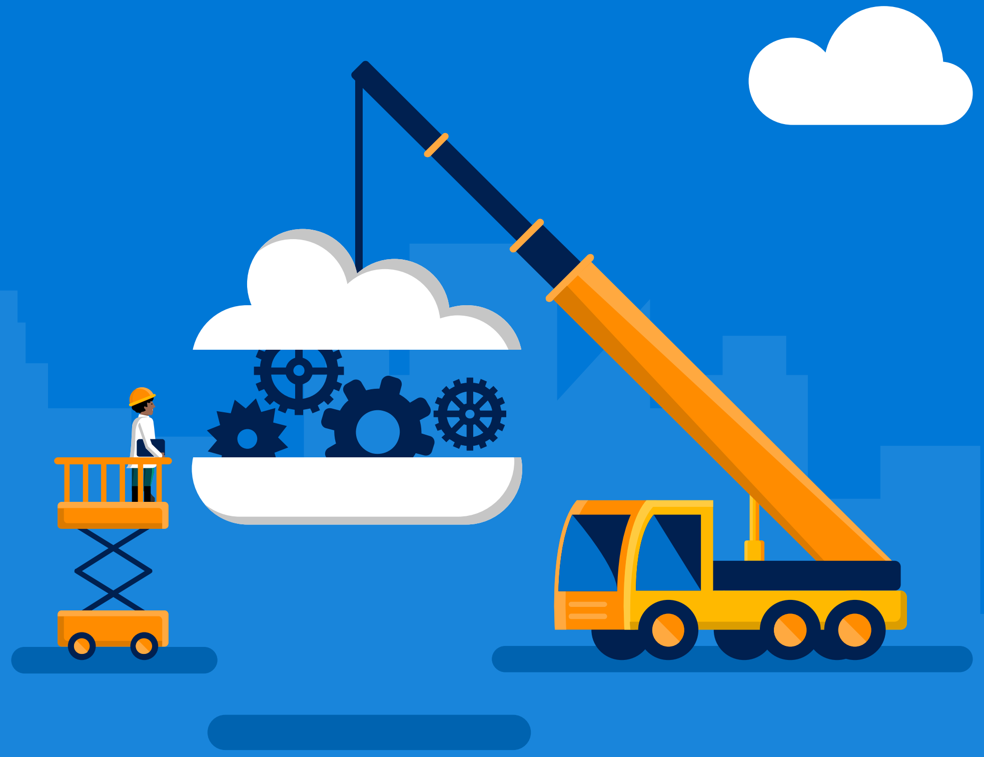
Microsoft

Tulsa Tech Fest & Tulsa School of Dev

July 20 & 21 2017

Tulsa OK



Please help us!
Thank our Sponsors:

Premier Sponsors

amazon web services
WORK READY OKLAHOMA

Gold Sponsors

mozilla
GABLE GOTWALS COUNSEL
NUTANIX
ROAD MAP TECHNOLOGIES
J.B. HUNT

Complete An Evaluation Form & Win

Your input is important!

You can access Evaluation Forms at:

http://TulsaSchoolofDev.com
http://TulsaTechFest.com

Fill them out!
You can win additional prizes!

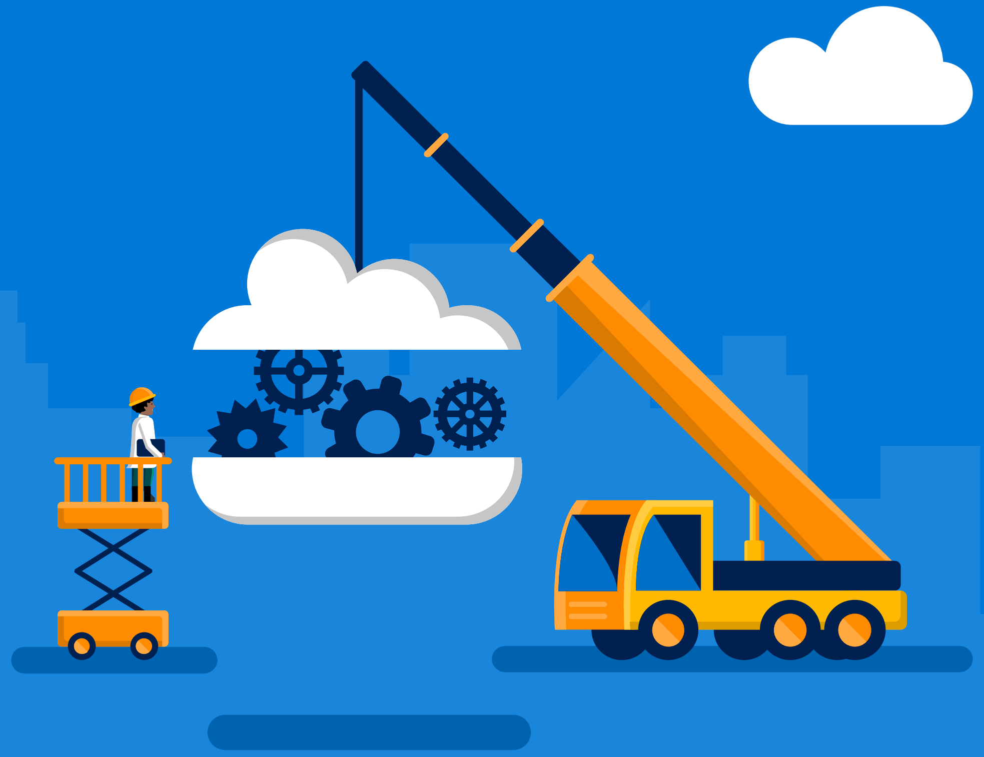
Like a $50 Best Buy Gift Card!!

Winner drawn – Midnight, Sun Jul 23rd!

Microsoft

Intro to Azure SQL Data Warehouse

Shawn Weisfeld



About Me

Shawn Weisfeld

Cloud Solution Architect

Microsoft – One Commercial Partner Technical Team

sweisfel@microsoft.com

Austin, TX

MVP Microsoft® Most Valuable Professional

dallasgivecamp

DRN Digital Recognition Network

woot!

Florida Institute of Technology

LOCKHEED MARTIN

ONETUG Orlando .NET User Group

Watch User Group presentations for FREE online!
We now have over 625 presentations online

- Miss a User Group meeting?
- Forget something that you learned?
- Want to see content from a User Group not in your area?
- Want to share with a buddy?

We know you cannot make
it to every session,
that is why we post them
online for you!

Presentations from the thought leaders on the topics you care about including:

- Agile
- Azure
- C#
- Entity Framework
- HTML5
- MVC
- SQL
- jQuery
- and Much More!

New Content added all the time!

For new content announcements

follow us on twitter

@UserGroupTV

http://www.UserGroup.tv

UserGroup.TV

Session Objective

In this session we will introduce Azure SQL Data Warehouse and provide the basics you need to get started. Azure SQL Data Warehouse combines the SQL Server relational database with Azure cloud scale-out capabilities. Built on our massively parallel processing (MPP) architecture, SQL Data Warehouse can handle your enterprise workload.

Agenda

- Big Data Options in Azure
- SQL Data Warehouse Basics
- Data Migration
- Table Distribution
- Common Architecture Patterns

Big Data Options in Azure

Relational

- IaaS
  - SQL Server
  - You name it
- PaaS
  - SQL Database
  - Analysis Services
  - SQL Data Warehouse

Other

- IaaS
  - You name it
- PaaS
  - Data Late Store/Analytics
  - HDInsight
  - Document DB
  - Redis Cache

Data Tools

- Machine Learning
- Stream Analytics
- Data Catalog
- Data Factory
- Power BI Embedded

SQL Data Warehouse Basics

What is a Data Warehouse?

DWs are central repositories of integrated data from one or more disparate sources. They store current and historical data in one single place and are used for creating analytical reports for knowledge workers throughout the enterprise.

Source: https://en.wikipedia.org/wiki/Data_warehouse

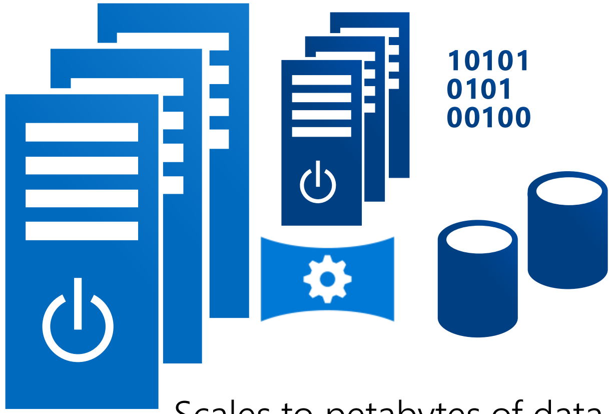
What is Azure SQL Data Warehouse?

Azure SQL Data Warehouse is a massively parallel processing (MPP) cloud-based, scale-out, relational database capable of processing massive volumes of data.

Azure SQL Data Warehouse

A relational **data warehouse-as-a-service**, fully managed by Microsoft.
Industries first **elastic** cloud data warehouse with proven SQL Server capabilities.
Support your **smallest to your largest** data storage needs.

Elastic scale & performance



Scales to petabytes of data

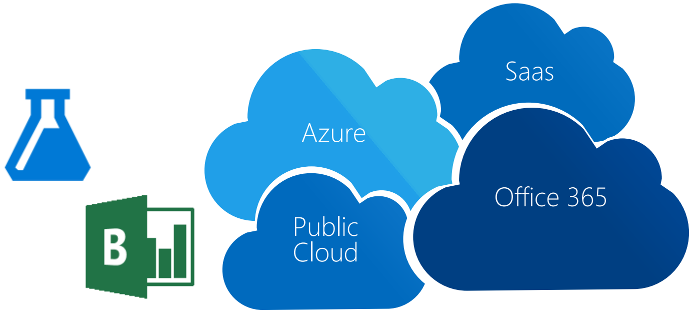
Massively Parallel Processing

Instant-on compute scales in seconds

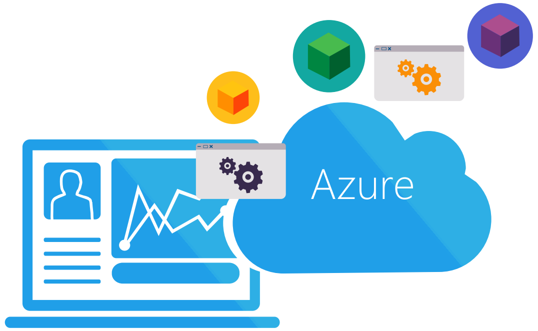
Powered by the Cloud

Get started in minutes

Integrated with Azure ML, PowerBI & ADF



Market Leading Price & Performance



Simple billing compute & storage

Pay for what you need, when you need it
with dynamic pause

What is Azure SQL Data Warehouse?

Azure SQL Data Warehouse is a massively parallel processing (MPP) cloud-based, scale-out, relational database capable of processing massive volumes of data.

- Utilizes SQL Server Transact-SQL (T-SQL) and tools

What is Azure SQL Data Warehouse?

Azure SQL Data Warehouse is a massively parallel processing (MPP) cloud-based, scale-out, relational database capable of processing massive volumes of data.

- Divide and conquer loads and complex queries across many compute nodes.

What is Azure SQL Data Warehouse?

Azure SQL Data Warehouse is a massively parallel processing (MPP) cloud-based, scale-out, relational database capable of processing massive volumes of data.

- No expensive equipment to buy, configure, maintain, upgrade, etc.
- Pay for what you need when you need it.
- Get started in minutes

What is Azure SQL Data Warehouse?

Azure SQL Data Warehouse is a massively parallel processing (MPP) cloud-based, scale-out, relational database capable of processing massive volumes of data.

- Grow or shrink storage size independent of compute.
- Grow or shrink compute power without moving data.
- Pause compute capacity while leaving data intact, only paying for storage.
- Resume compute capacity during operational hours.

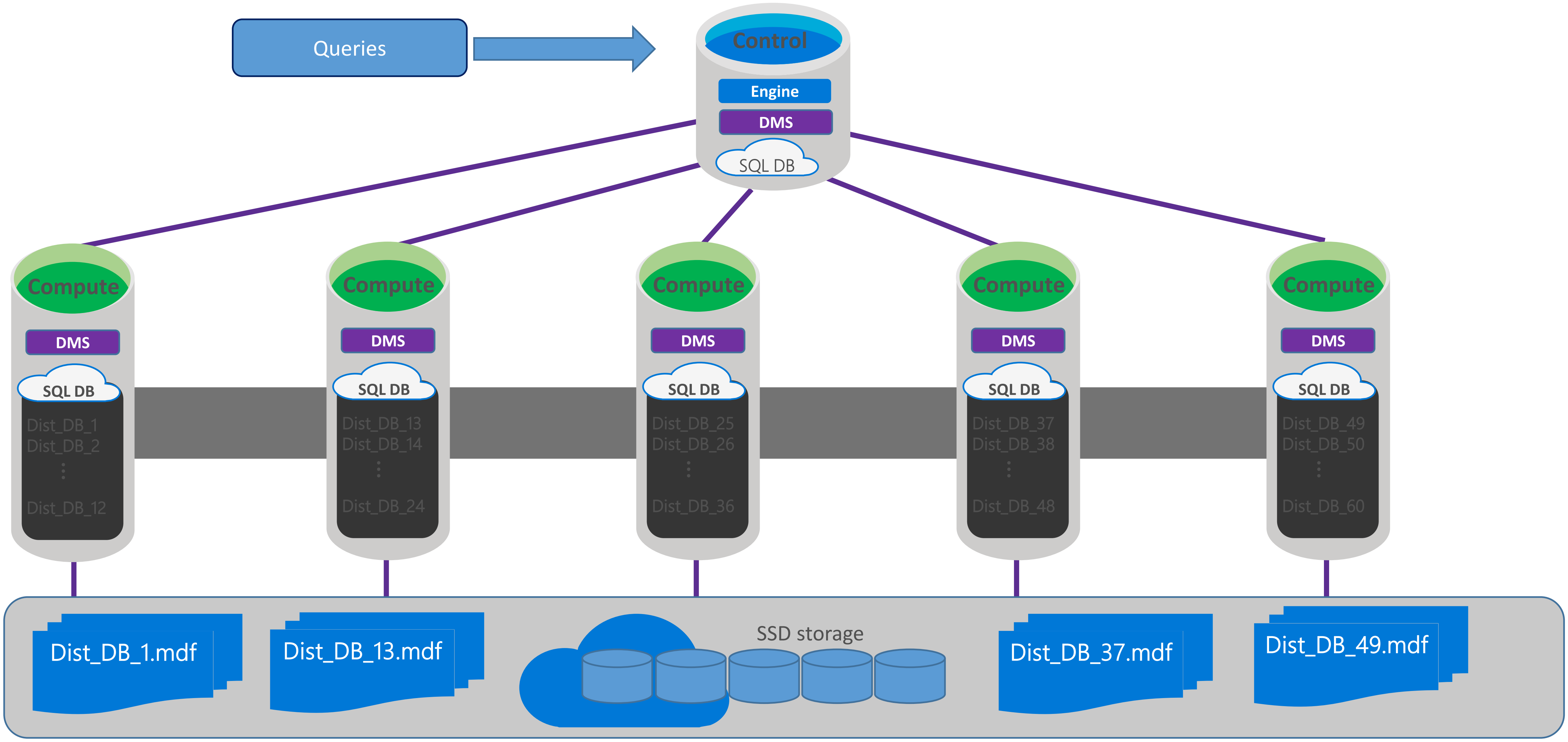
What is Azure SQL Data Warehouse?

Azure SQL Data Warehouse is a massively parallel processing (MPP) cloud-based, scale-out, relational database capable of processing massive volumes of data.

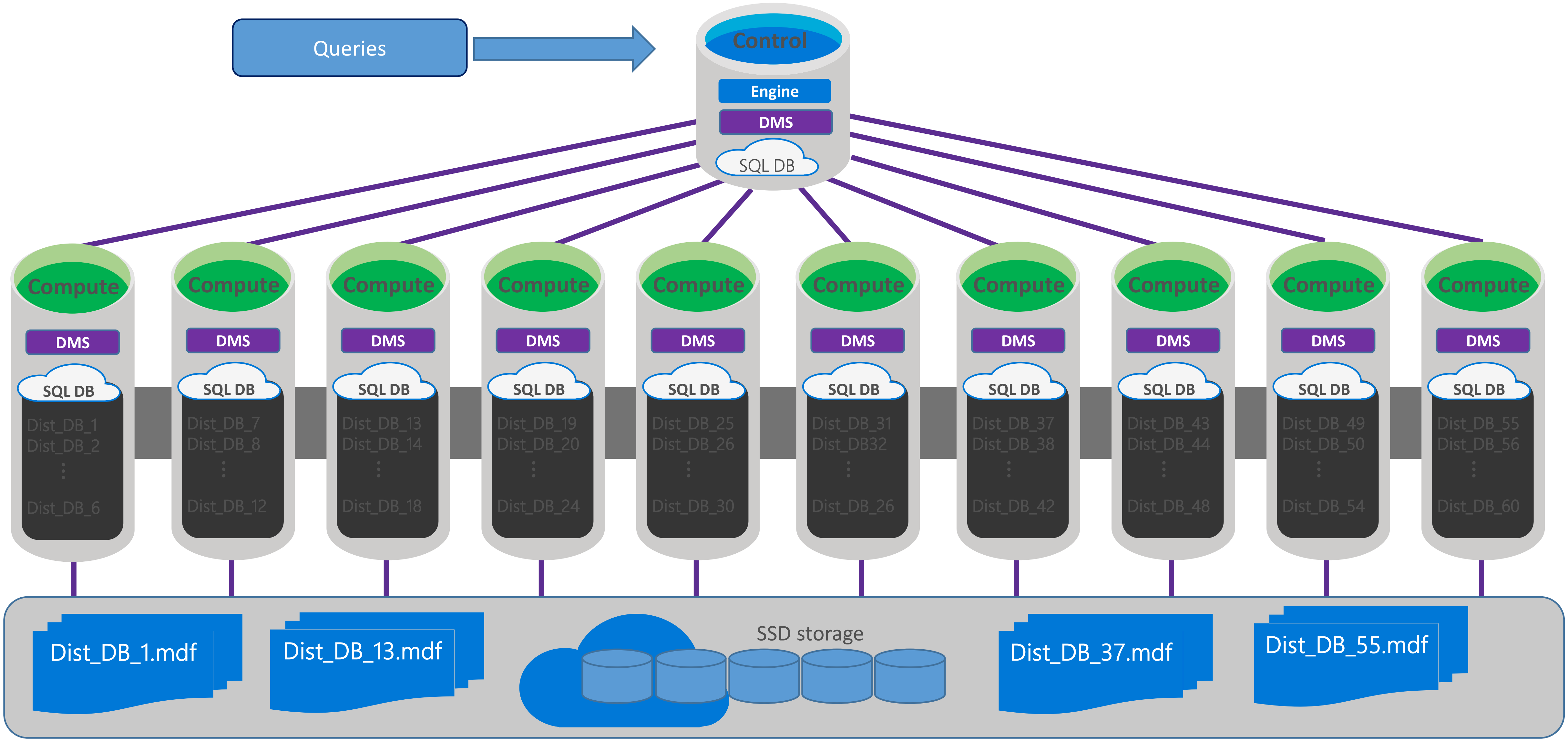
- 240 TB on disk
- Up to approximately 1 PB uncompressed when all tables are clustered columnstore

Source: https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-service-capacity-limits

SQL DW Architecture



SQL DW Architecture



Data Warehouse Unit - DWU

- 100 DWU to 6,000 DWU
- Liner Scale

Resource classes

- Control memory allocation and CPU cycles given to a query
- 4 sizes
  - smallrc, mediumrc, largerc, and xlargerc
- Trade off between power allocated to a user and total number of concurrent queries

More Info: https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-develop-concurrency

Demo

- Create
- Scale
- Query
- Load

Data Migration

Common Loading Options

PolyBase	BCP	SQLBulkCopy API	SSIS
Fastest and preferred load option. Use CTAS for initial load. Use INSERT/INTO for incremental load or CTAS into stage table and partition switch into final table. Load speed increases as you add DWUs	Use only for small files < 10 GB. Limited retry logic. Does not scale as you increase DWU (single thread, single CPU on client). Increase parallel threads to improve performance.	Greater control with error trapping and retry logic. Increase parallel threads to improve performance. Slight performance improvement & greater reliability if run on VM.	Increase client timeout at least 10 min, default 30 sec. Increase parallel threads to improve performance. Slight performance improvement & greater reliability if run on VM.

General Best Practices

- Local Disk Performance on export
- Choose the right region
  - Close to you, close to your customers, close to your other Azure services
- Data Warehouse Migration Utility (Preview)
  - https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-migrate-migration-utility
- Batch DML operations (Insert, Update, Delete)
- Avoid fully logged operations
  - Create Table as Select for historical load
  - Insert Into for Incremental Load

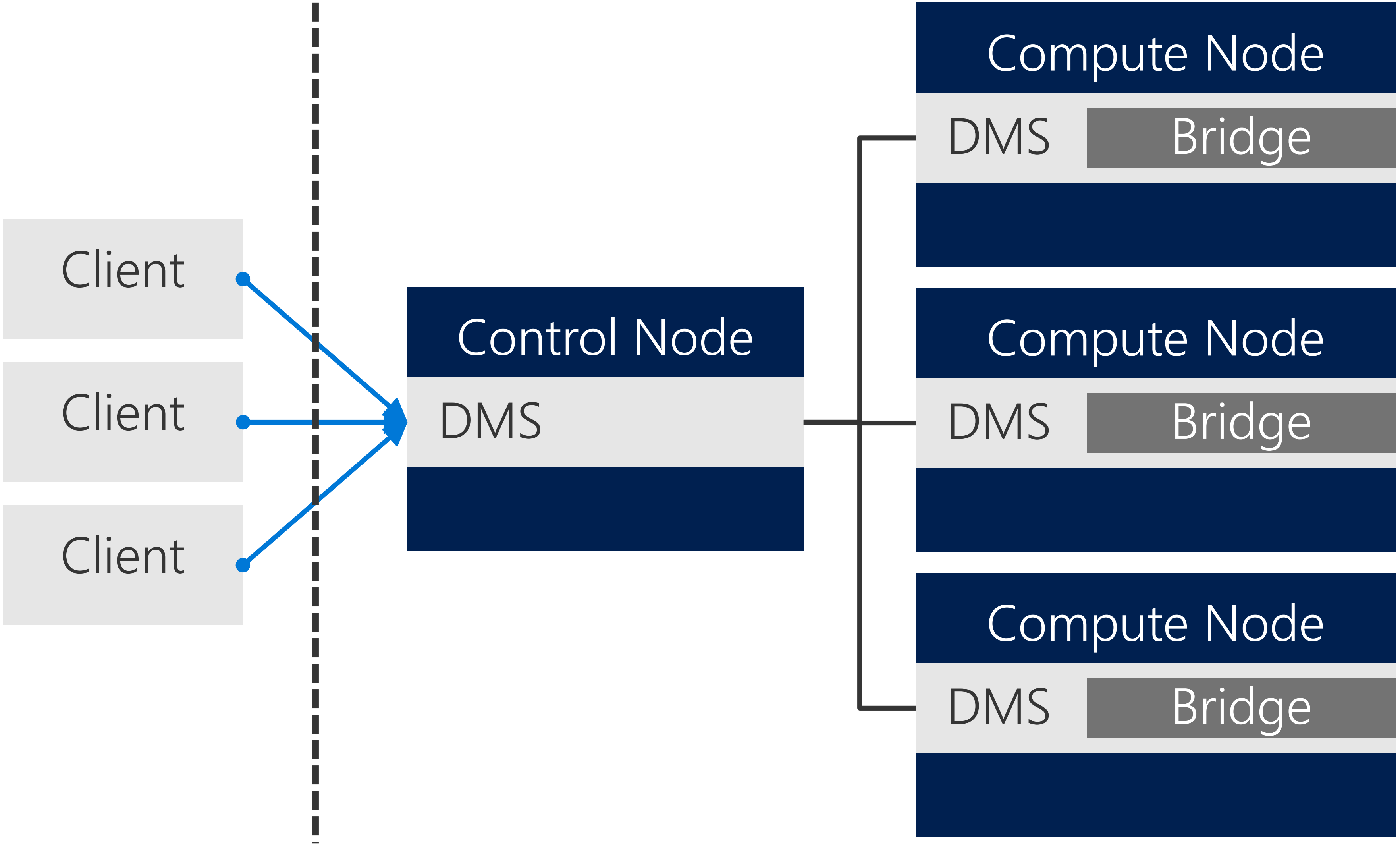
General Best Practices (continued)

- Test loads with small data files through entire pipeline before moving all your data
- Very large amounts of data
  - Look at Express Route and Import/Export service
- Push files to Azure with AZCopy
  - http://aka.ms/AZCopy
- Generate IDs in Source system or during ETL
- Use Left Outer Joins instead of merge command
- Include retry logic

PolyBase Best Practices

- Consolidate on a single date format when exporting from the OLTP system
- Use field delimiters that are not contained in the source
  - Multiple character field delimiters are supported
- Gzip Compression limits you to one reader per zip file
- Create one folder with multiple files for large tables
- UTF-8, UTF-16 & Azure DLS is supported

Data Loading: Single Gated Client



Data Loading: Parallel Loading with PolyBase

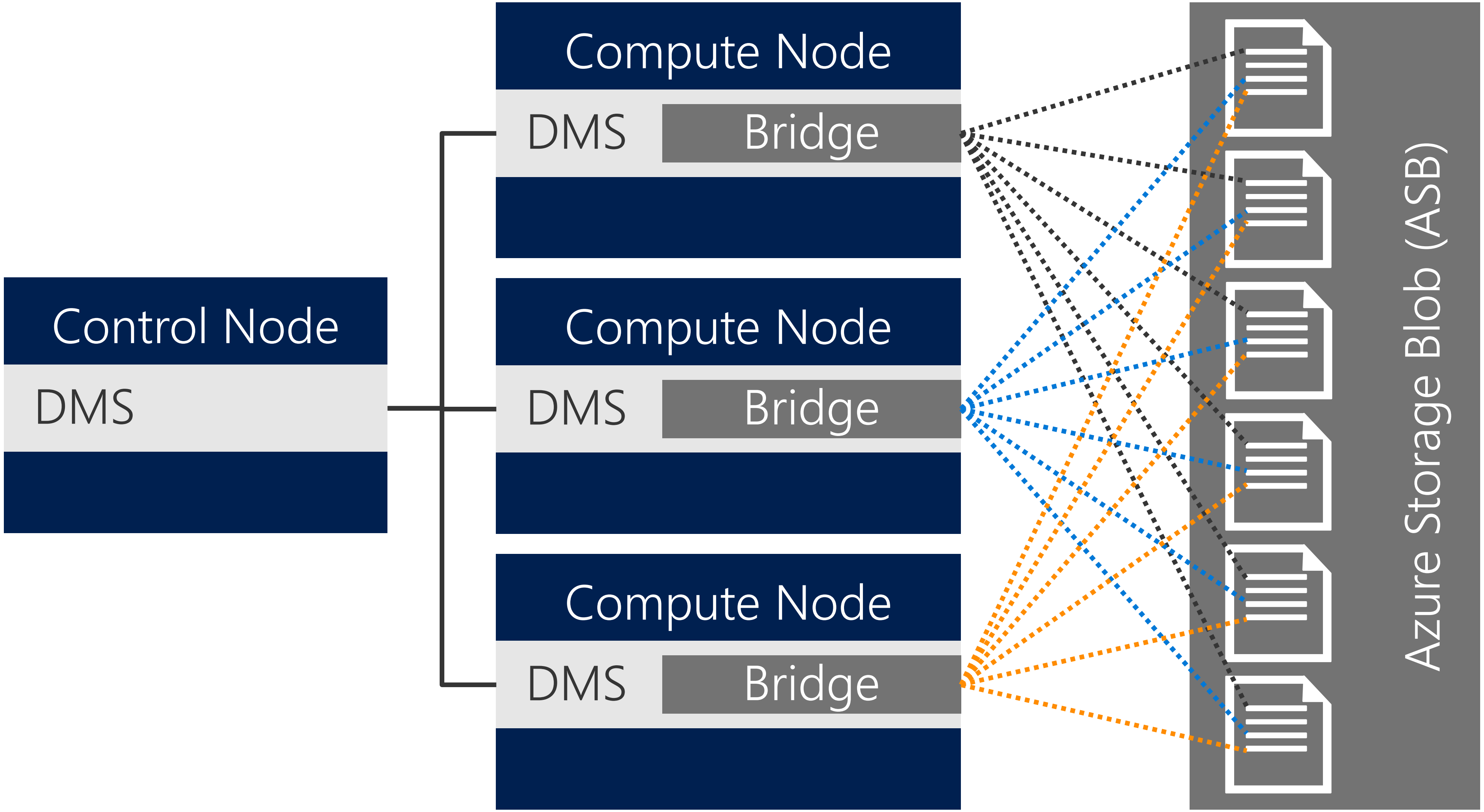


Table Distribution

Table Distribution Options

Hash Distributed

Data divided across nodes based on hashing algorithm

Same value will always hash to same distribution

Single column only

Check for Data Skew, NULLS, -1

Round Robin

(Default)

Data distributed evenly across nodes

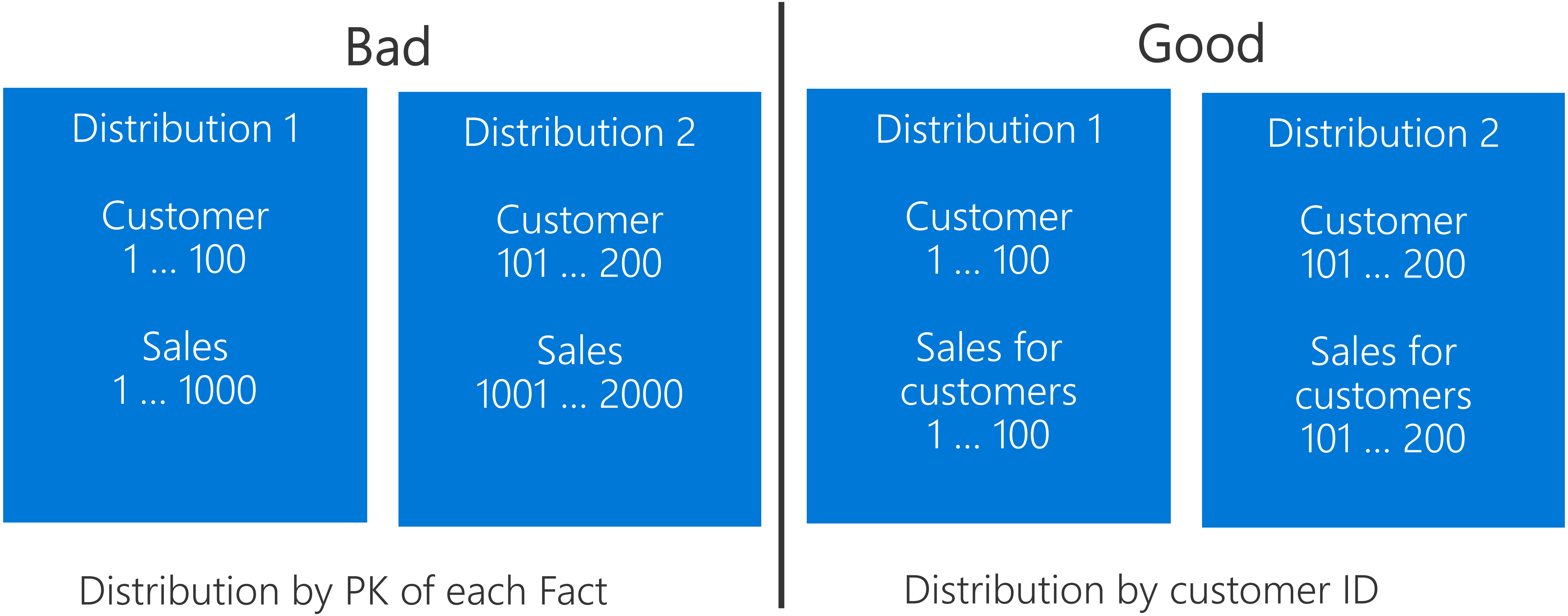
Easy place to start, don't need to know anything about the data

Simplicity at a cost

Will incur more data movement at query time

Table Distribution Example

- Data must be located on the same distribution to join…



Bad

Distribution 1

Customer
1 … 100

Sales
1 … 1000

Distribution 2

Customer
101 … 200

Sales
1001 … 2000

Distribution by PK of each Fact

Good

Distribution 1

Customer
1 … 100

Sales for
customers
1 … 100

Distribution 2

Customer
101 … 200

Sales for
customers
101 … 200

Distribution by customer ID

Fact Table Best Practices

- Hash Distribute by columns used to join to other fact tables
- Keep in Mind
  - Hash column should have highly distinct values (Minimum >60 distinct values)
  - Avoid distributing on a date column
  - Avoid distributing on column with high frequency of NULLs and default values (e.g. -1)
  - Distribution column is NOT updatable
  - For compatible joins use the same data types for two distributed tables
- Use Round Robin as a last resort

Dimension Table Best Practices

- Small
  - Less than 60 Million Rows
    - DW has 60 distributions, need 1 million rows per columnstore
  - Use clustered indexes instead of columnstore
  - Use Round Robin
- Large
  - See Fact Table Best Practices

Common Data Movement Types

DMS Operation	Description
ShuffleMoveOperation	Redistributes data for compatible join or aggregation
PartitionMoveOperation	Data moves from compute to control node (i.e. Average)
BroadcastMoveOperation	Table needs to become replicated for join compatibility

Common Data Movement Types

DMS Operation	Description
ShuffleMoveOperation	Redistributes data for compatible join or aggregation
PartitionMoveOperation	Data moves from compute to control node
BroadcastMoveOperation	Table needs to become replicated for join compatibility

Optimizing with Indexes

Clustered ColumnStore
(SQL DW Default)

- Optimal choice for **large tables**
- Limits scans to columns in the query
- Optimal compression
- Slower to load than Heap
- Keep partitions large enough to compress (> 1 million rows)

Heap

- Optimal choice for **temporary or staging tables**
- Fastest load performance

Clustered Index

- Optimal for tables < 60M rows
- Sorting operation slows down load

Nonclustered Indexes

- **Use sparingly**
- Optimize single row lookups
- Will slow down load

Partitioning Best Practices

- By date – improves performance by partition elimination
- Granularity depends on workload - target 1 million rows per distribution/partition
- Utilize partition switching to Optimize load performance
- Index by partion

DDL Example

```
CREATE TABLE FactFinance
(
            FinanceKey int NOT NULL,
            Date datetime2 NOT NULL,
            OrganizationKey int NOT NULL,
            DepartmentGroupKey int NOT NULL,
            ScenarioKey int  NULL,
            AccountKey int  NULL,
            Amount float NOT NULL)
WITH (clustered columnstore index, DISTRIBUTION = HASH(FinanceKey),
    PARTITION (Date RANGE RIGHT FOR VALUES
   (N‘2016-01-01T00:00:00.000', N‘2016-02-01T00:00:00.000', N‘2016-03-01T00:00:00.000'))
);
```

Statistics

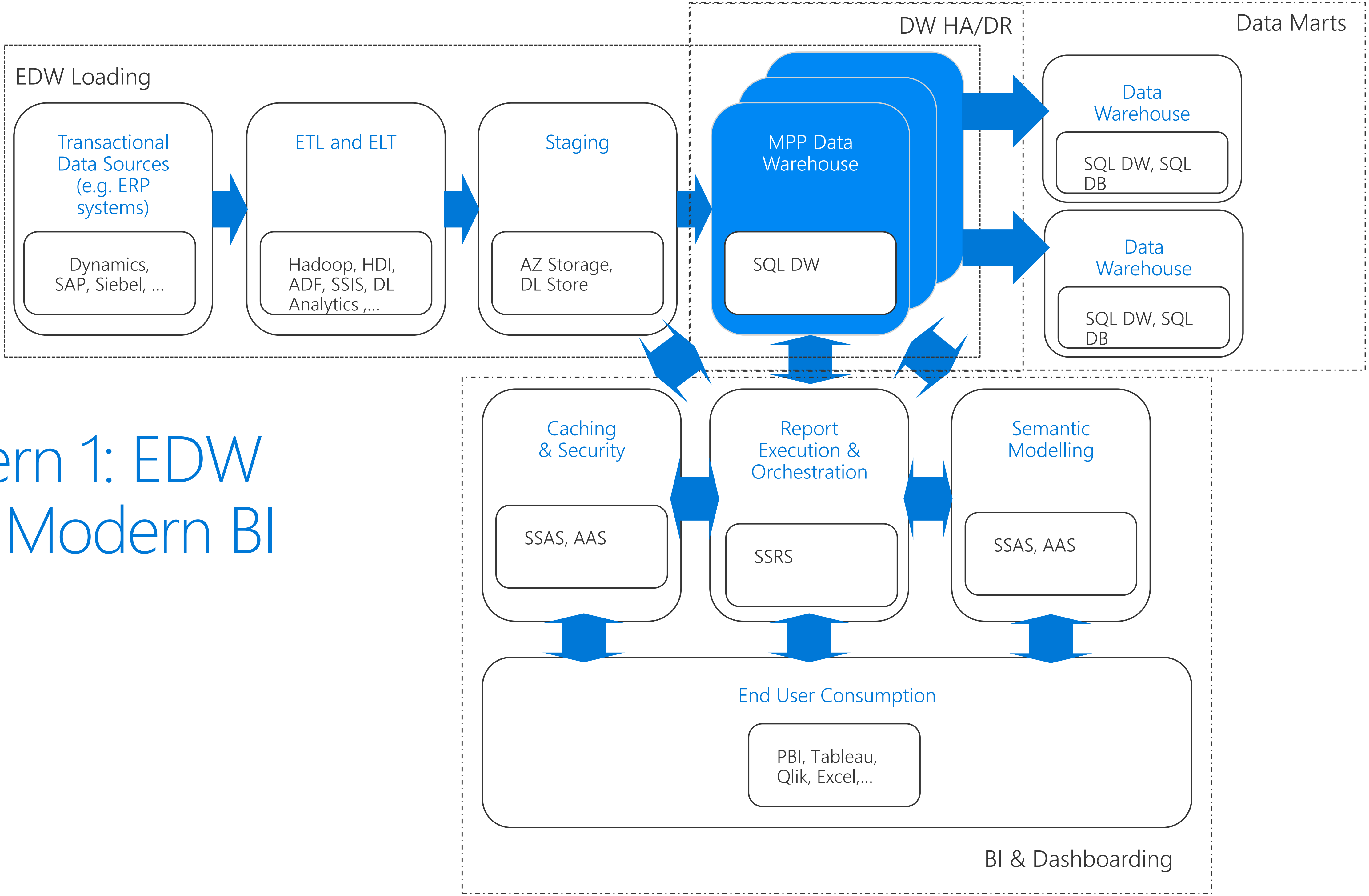
- Manual today
- Cost Based Query Optimizer needs statistics
- Sampled stats are usually fine
- Create statistics for all columns used in JOINs, GROUP BY, WHERE
- Update statistics after incremental load
- If needed, use multi-column statistics on join and group by

Common Architecture Patterns

Pattern 1: EDW with Modern BI

- Data: Usually human-born data, e.g., orders, sales, financial, customers, marketing
- Modeling: Dimensional models (e.g. Kimball)
- Queries: Star joins with grouping and aggregation
- Loading: Periodic incremental load batches
- Workload: Thousands of users, hundreds of user sessions, tens of concurrent queries

Pattern 1: EDW with Modern BI



EDW Loading

Transactional Data Sources (e.g. ERP systems)

Dynamics, SAP, Siebel, …

ETL and ELT

Hadoop, HDI, ADF, SSIS, DL Analytics ,…

Staging

AZ Storage, DL Store

DW HA/DR

MPP Data Warehouse

SQL DW

Data Marts

Data Warehouse

SQL DW, SQL DB

Data Warehouse

SQL DW, SQL DB

Caching & Security

SSAS, AAS

Report Execution & Orchestration

SSRS

Semantic Modelling

SSAS, AAS

End User Consumption

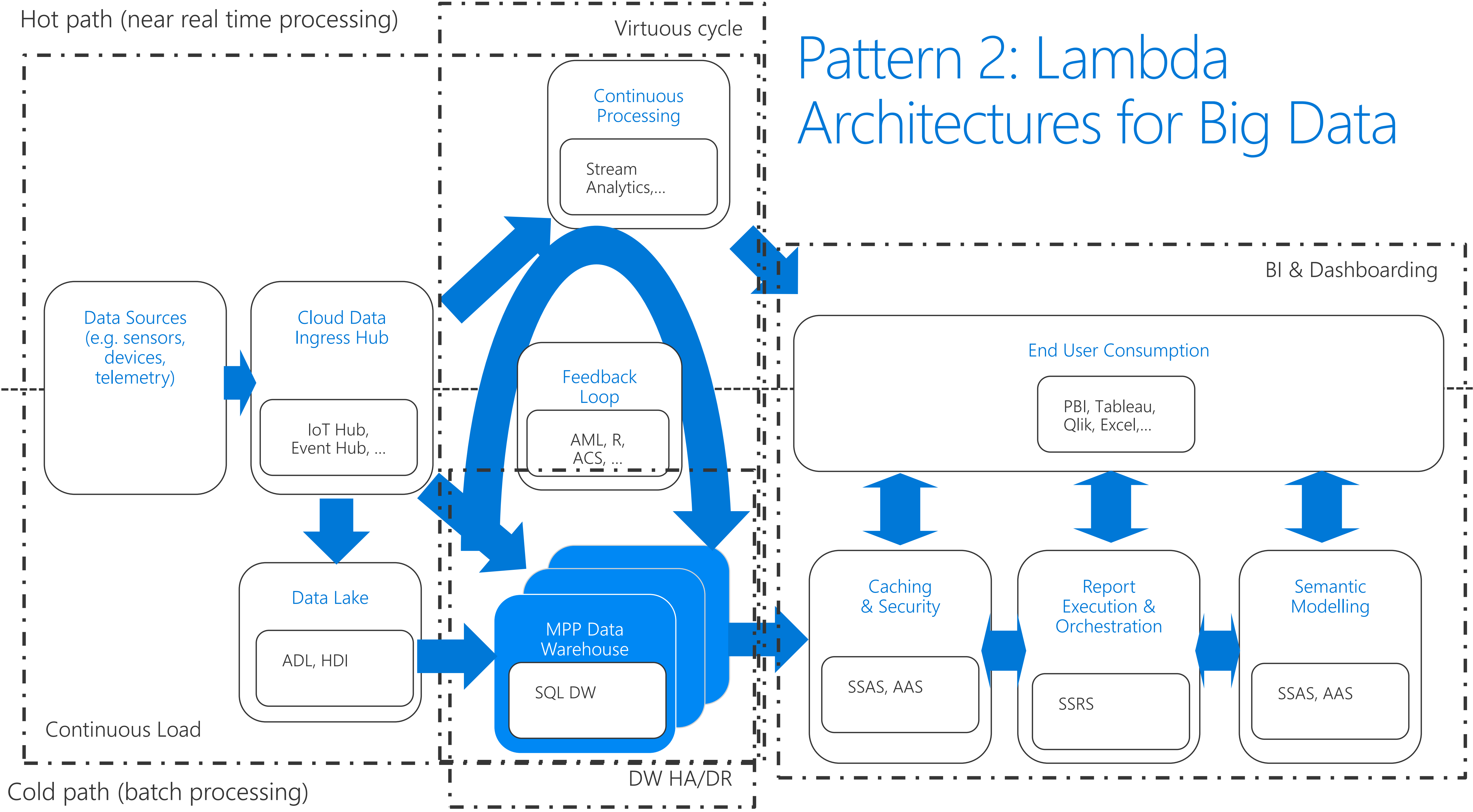
PBI, Tableau, Qlik, Excel,…

BI & Dashboarding

Pattern 2: Lambda Architectures for Big Data

- Data: Usually machine-born data, e.g., device telemetry, log records, IoT data
- Modeling: Oftentimes de-normalized
- Queries: Data ranges combined with string and pattern search
- Loading: Ideally continuous loads
- Workload: Hundreds of users, tens of ongoing user sessions, limited query concurrency

Pattern 2: Lambda Architectures for Big Data



Wrap up

Resources

- SQL CAT Blog
  - http://aka.ms/SQLCAT
- SQL DW Free Trial
  - https://azure.microsoft.com/en-us/services/sql-data-warehouse/extended-trial

Call to Action

- Spin up a SQL Data Warehouse in Azure
- Kick the tires
- Let us know what you think
- Evaluate workloads that you have that this would help with

Thank you! Your Feedback is Important

- Rate My Talk & Download Slides!

http://bit.ly/RateShawnsTalk

(case sensitive)

- Contact Information
  - Email: sweisfel@microsoft.com
  - Blog: http://www.shawnweisfeld.com
  - Twitter: @shawnweisfeld

Microsoft