SQL Data Warehouse
Loading Data

Chris Testa-O'Neill
Features Engineer
Analytics and Data Science Team

Microsoft

Agenda

- The “load user”
- Loading Data
- Using PolyBase
- Importing and Exporting Data Loads
- Monitoring data loads
- Azure Data Factory Integration
- Recommendations

The “load user”

Why create a dedicate user for data loading?

Post Provisioning

1 Login

Service admin

Full “sa” permissions

Fixed memory assignment

What benefits do I get?

More granular permissions model

Flexible memory management

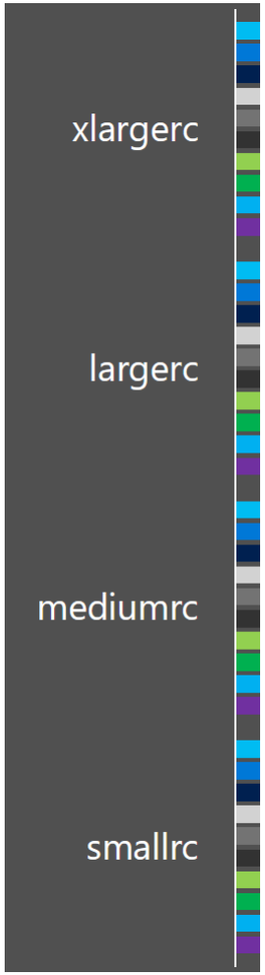
Easier to identify requests

Create Login (master)

```
CREATE LOGIN LoginName WITH PASSWORD = 'SQLB1ts!';

CREATE USER UserName for LOGIN LoginName;
EXEC sp_addrolemember 'loginmanager','UserName';
EXEC sp_addrolemember 'dbmanager','UserName ';
```

Resource class roles




```sql
SELECT   ro.[name]              AS [db_role_name]
FROM     sys.database_principals ro
WHERE    ro.[type_desc]         = 'DATABASE_ROLE'
AND      ro.[is_fixed_role]     = 0
;
```

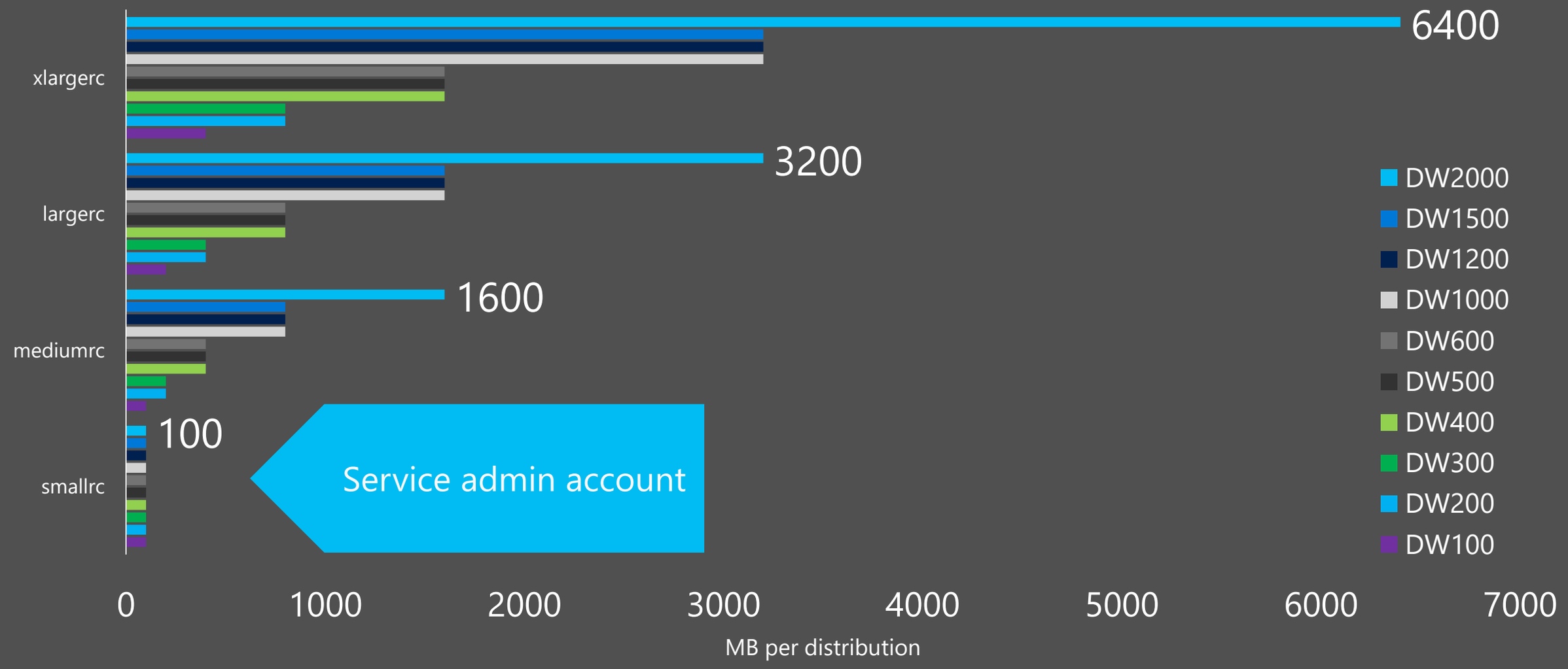
Create user (user db)

```
CREATE USER UserName for LOGIN LoginName
;
GRANT CONTROL ON DATABASE::MySQLDW TO UserName
;
SELECT  r.[name] AS role_principal_name
,       m.[name] AS member_principal_name
FROM    sys.database_role_members rm
JOIN    sys.database_principals AS r    ON rm.[role_principal_id]     = r.[principal_id]
JOIN    sys.database_principals AS m    ON rm.[member_principal_id]   = m.[principal_id]
WHERE   r.[name] IN ('mediumrc','largerc', 'xlargerc')
;
EXEC sp_addrolemember 'mediumrc',UserName'
;
```

Identifying users with elevated requests

```
SELECT  r.[request_id]                                  AS Req_ID
,       r.[command]                                     AS Req_command
,       r.[status]                                      AS Req_Status
,       r.[submit_time]                                 AS Req_SubmitTime
,       r.[start_time]                                  AS Req_StartTime
,       DATEDIFF(ms,[submit_time],[start_time])         AS Req_WaitDuration_ms
,       r.[resource_class]                              AS Req_resource_class
FROM    sys.dm_pdw_exec_requests r
WHERE   [session_id] <> session_id()
;
```

Memory Management (MB per distribution)



Loading

Loading options

Parallel

PolyBase

Azure Data Factory

Single Gated Client

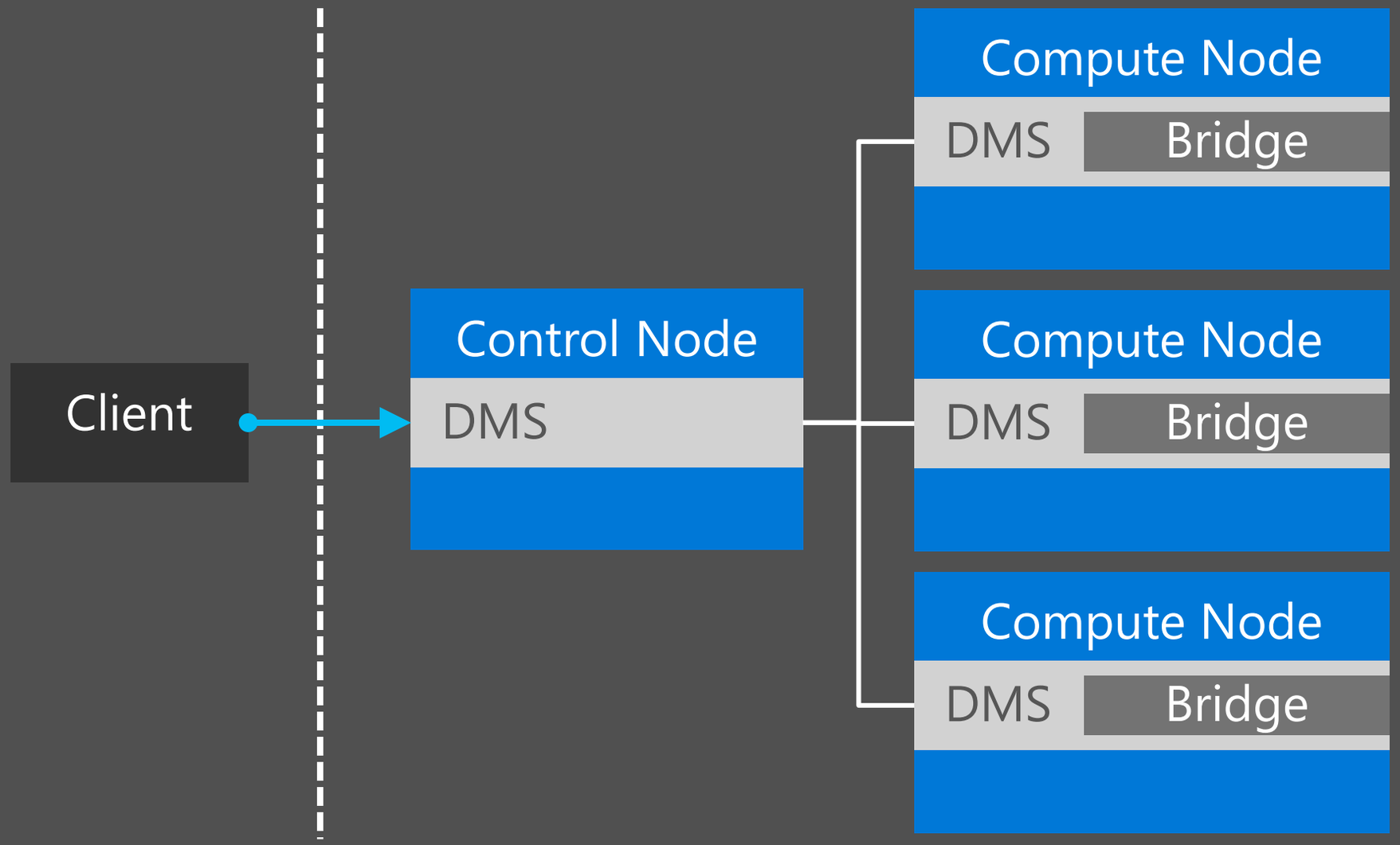
bcp / Insert Bulk

SQLBulkCopy

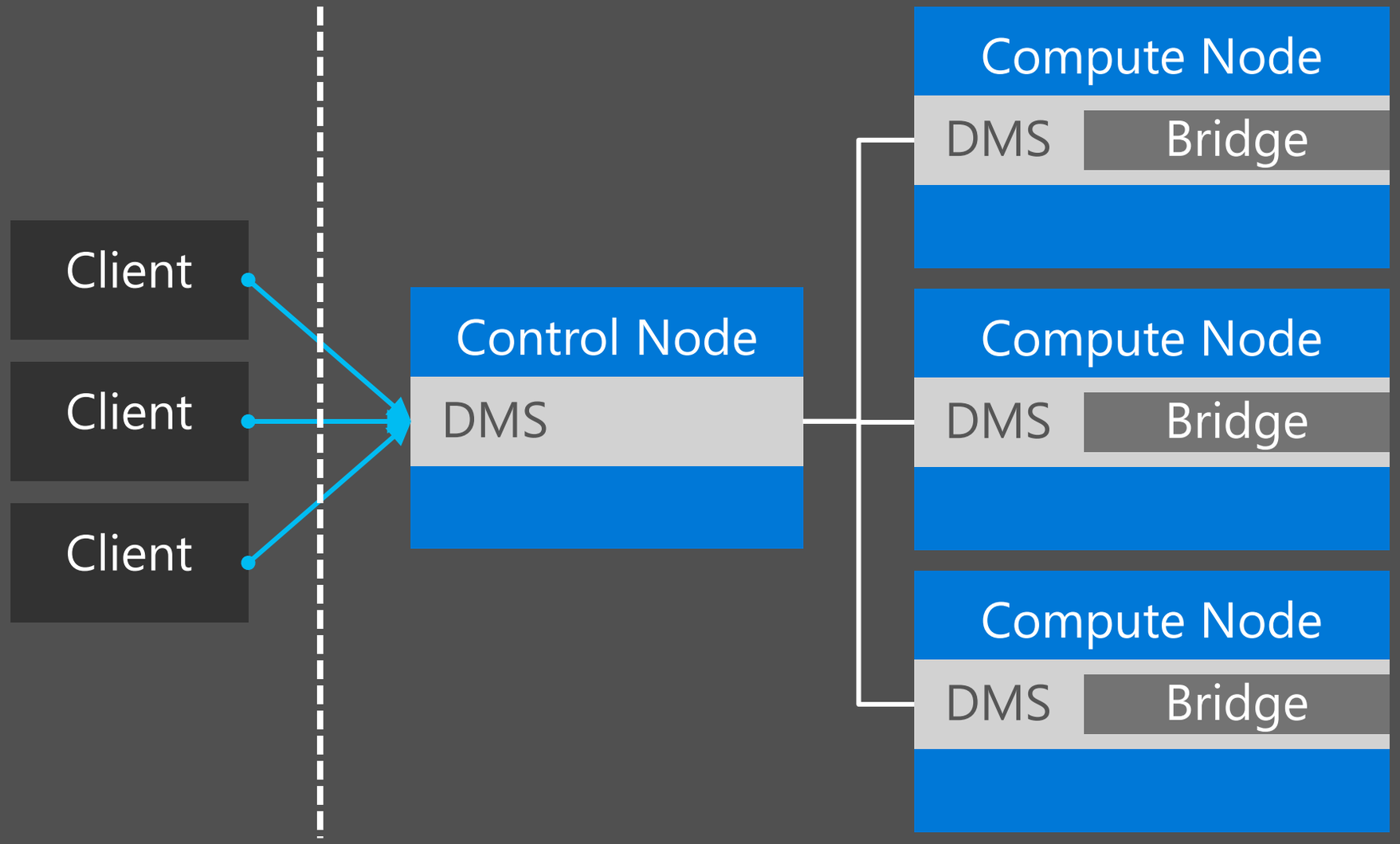
SSIS (data flow)

Azure Data Factory

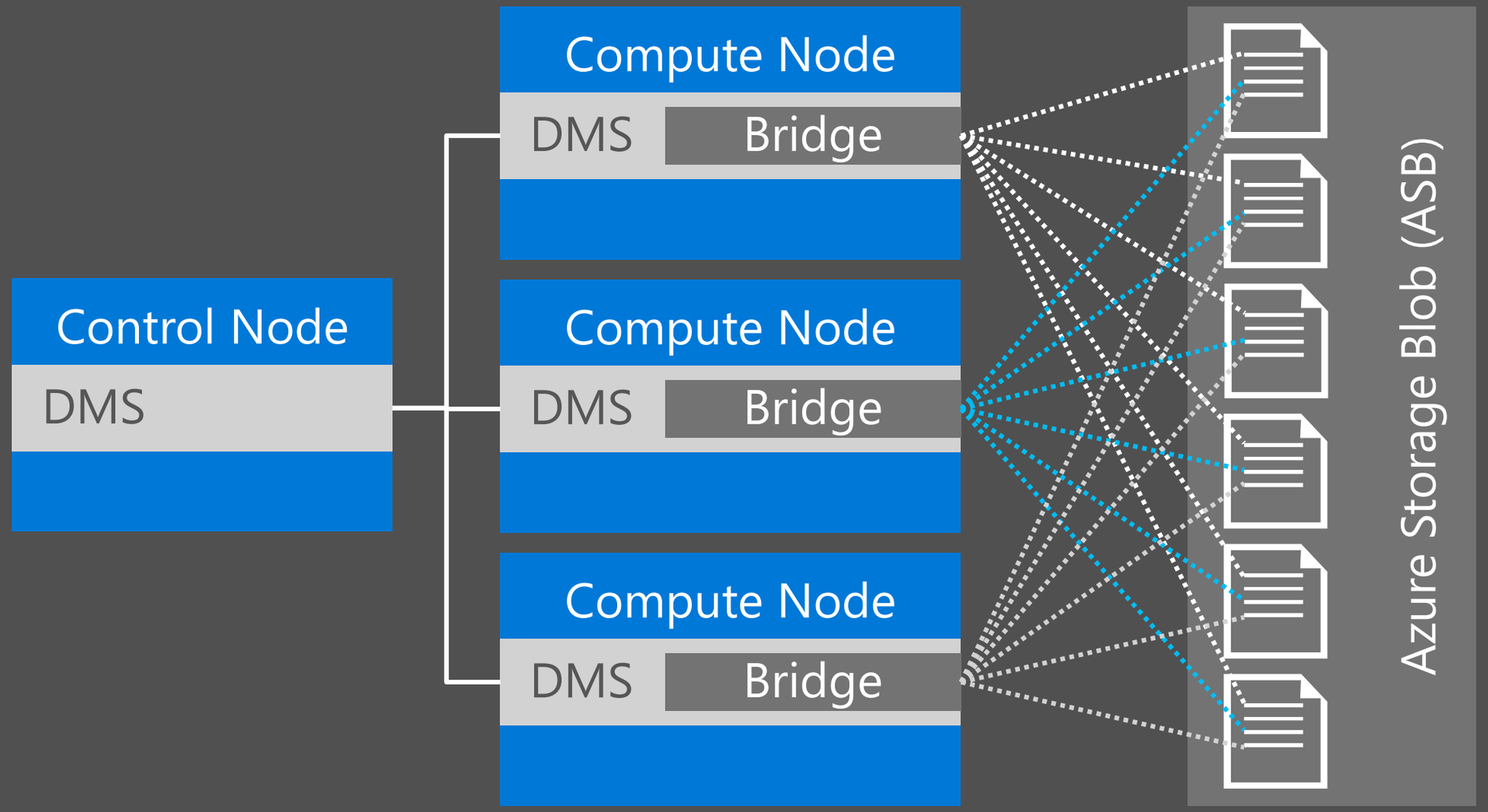
Single Gated Client



Single Gated Client Parallelised



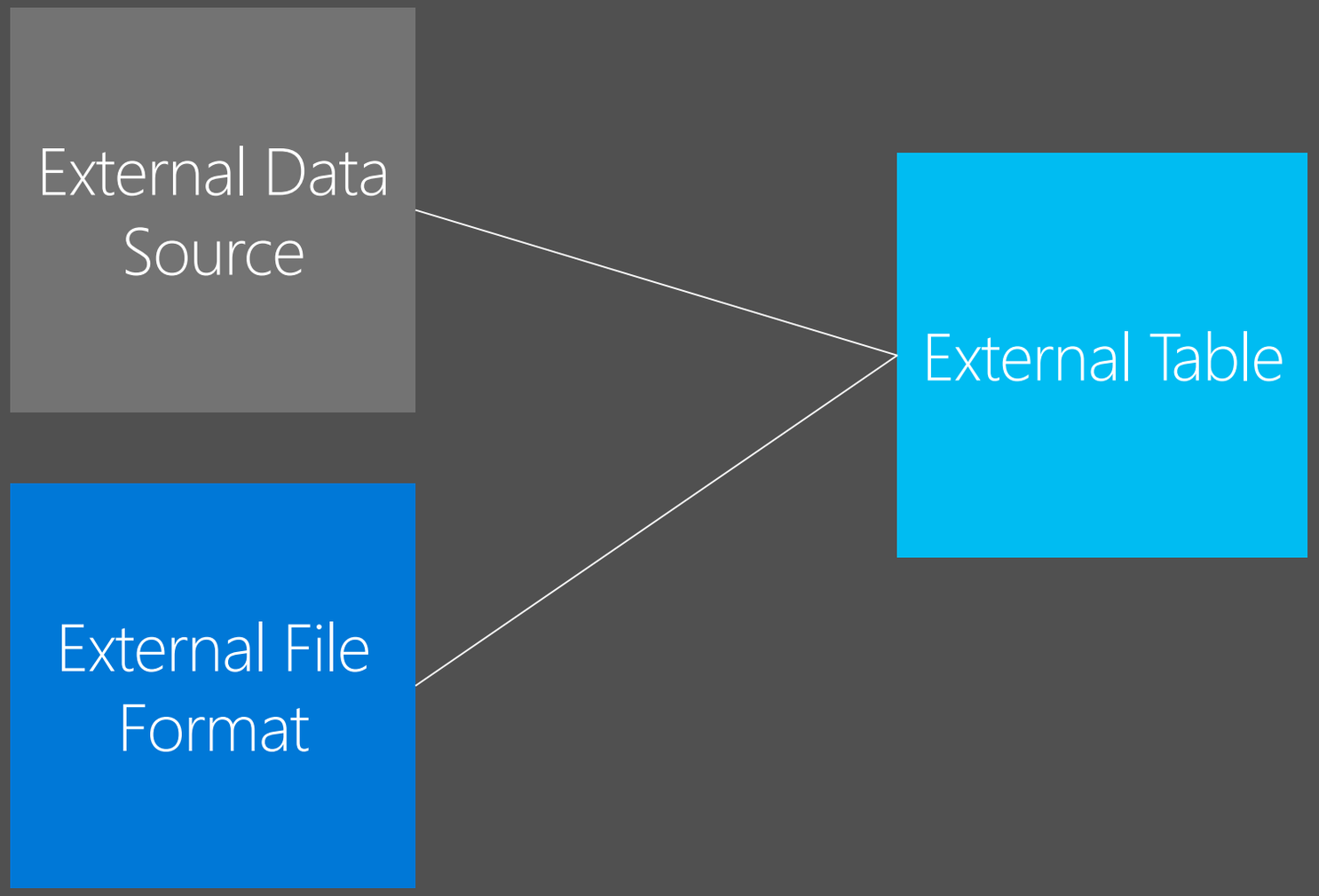
Parallel Loading with PolyBase



Demo:
Loading data with a single gated client.

PolyBase

Core PolyBase objects



External Tables

Metadata used to describe external data

Enables data access outside the database

Never holds data

Does not delete data when dropped

Behaviour of an external table is very similar to Hive external tables

External table metadata

sys.external_tables

sys.tables

Create External Table

```
CREATE EXTERNAL TABLE [asb].[FactOnlineSales]
([ProductKey]    int      NOT NULL
,[StoreKey]      int      NOT NULL
,[DateKey]       int      NOT NULL
,[CustomerKey]   int      NOT NULL
,[PromotionKey]  int      NOT NULL
,[SalesQuantity] int      NOT NULL
,[UnitPrice]     money    NOT NULL
,[SalesAmount]   money    NOT NULL
)
```

External Tables (cont)

```
WITH
(LOCATION='wasbs://filepath_or_directory'
,DATA_SOURCE              = MyDataSourceName
,FILE_FORMAT              = MyFileFormatName
,REJECT_TYPE              = VALUE
,REJECT_VALUE             = 0
,REJECT_SAMPLE_VALUE      = 1000
)
;
```

External Data Source

```
CREATE EXTERNAL DATA SOURCE MyAzureDataSource
WITH
(TYPE          = HADOOP
,LOCATION      =
'wasb[s]://[container@]account_name.blob.core.windows.net/path'
)
;
```

External File Format - ORC

```
CREATE EXTERNAL FILE FORMAT ORCFileFormat
WITH
(FORMAT_TYPE        =    ORC
,DATA_COMPRESSION   =
'org.apache.hadoop.io.compress.DefaultCodec'
| 'org.apache.hadoop.io.compress.SnappyCodec'
)
;
```

External File Format - Parquet

```
CREATE EXTERNAL FILE FORMAT ParquetFileFormat
WITH
(FORMAT_TYPE        =    PARQUET
,DATA_COMPRESSION   =
'org.apache.hadoop.io.compress.SnappyCodec'
| 'org.apache.hadoop.io.compress.GzipCodec'
)
;
```

Hive Data Type Mapping

Missing Types in ORC / Parquet

SQL Type	Recommendation
DATE	Use TIMESTAMP

Different Ranges

Hive Type	Hive	SQL
TINYINT	-128 to +127	0 to 255
TIMESTAMP	1970 to 2039	0001-01-01 to 9999-12-31

Reference:
https://cwiki.apache.org/confluence/display/Hive/Language Manual+Types

External File Format – Delimited Text

```
CREATE EXTERNAL FILE FORMAT MyTextFileFormat
WITH
(FORMAT_TYPE        =   DELIMITEDTEXT
,FORMAT_OPTIONS     (   FIELD_TERMINATOR= '|'
                    ,       STRING_DELIMITER= ','
                    ,       DATE_FORMAT= 'yyyy-MM-dd'
                    ,       USE_TYPE_DEFAULT= TRUE
                    )
,DATA_COMPRESSION   =
'org.apache.hadoop.io.compress.DefaultCodec'
| 'org.apache.hadoop.io.compress.GzipCodec'
)
;
```

Delimited text guidance

UTF-8 encode your files

Row delimiter is not configurable

No row delimiters in strings

GZIP not Winzip for compression

Delimiter	Description
\r	Carriage return {CR}
\n	Line Feed {LF}
\r\n	Carriage return linefeed {CR}{LF}

DATE_FORMAT

No DATE_FORMAT in EFF

DateTime: 'yyyy-MM-dd HH:mm:ss'

SmallDateTime: 'yyyy-MM-dd HH:mm'

Date: 'yyyy-MM-dd'

DateTime2: 'yyyy-MM-dd HH:mm:ss'

DateTimeOffset: 'yyyy-MM-dd HH:mm:ss'

Time: 'HH:mm:ss'

DATE_FORMAT in EFF

Same format used for all date typed fields

Cannot specify multiple date formats in the same EFF

One external file = one file format

Demo:
Loading data with PolyBase

Lab:

Loading Data with Polybase

Microsoft Azure

Importing and exporting data

Importing with CTAS

```
CREATE TABLE [tmp].[FactOnlineSales]
WITH
(    DISTRIBUTION = HASH([ProductKey])
,    CLUSTERED COLUMNSTORE INDEX
)
AS
SELECT    *
FROM      [asb].[FactOnlineSales]
OPTION
(LABEL =  'CTAS : Import [cso].[FactOnlineSales]'
)
;
```

Creating a partitioned table with CTAS

```
CREATE TABLE [cso].FactOnlineSales_PTN
WITH
(   CLUSTERED COLUMNSTORE INDEX
,   DISTRIBUTION = HASH([ProductKey])
,   PARTITION
    (
        [DateKey] RANGE RIGHT FOR VALUES
        (
               '2007-01-01 00:00:00.000','2008-01-01 00:00:00.000'
         ,     '2009-01-01 00:00:00.000','2010-01-01 00:00:00.000'
        )
    )
)
AS
SELECT  *
FROM    [cso].[FactOnlineSales]
;
```

Exporting with CETAS

```
CREATE EXTERNAL TABLE [out].[dimProduct]
WITH(LOCATION = '/export/FactOnlineSales/'
,DATA_SOURCE = AzureStorage
,FILE_FORMAT = TextFileFormat
)
AS
SELECT *
FROM [cso].[dimProduct]
OPTION
(LABEL = 'CETAS : Export [cso].[FactOnlineSales]'
)
;
```

Labelling your code

Supported operations:

Select

Insert

Update

Delete

CTAS

CETAS

```
SELECT  *
FROM    sys.dm_pdw_exec_requests
WHERE   [label] = 'SQLBits'
```

Demo:
Exporting data with CTAS

Lab:

Exporting Data with CTAS

Microsoft Azure

Monitoring data loads

Monitoring execution requests

```
SELECT  'sys.dm_pdw_exec_requests'                          AS DMV
,       [label]                                             AS operation
,       NULL                                                AS location_type
,       NULL                                                AS step_index
,       DATEDIFF(ms ,MIN(req.[submit_time])
                    ,MAX(req.[end_time]))/1000.0            AS duration_sec
,       MIN(req.[submit_time])                              AS min_start_time
,       MAX(req.[end_time])                                 AS max_end_Time
,       MIN(req.[total_elapsed_time])/1000.0                AS min_duration_sec
,       MAX(req.[total_elapsed_time])/1000.0                AS max_duration_sec
,       AVG(req.[total_elapsed_time])/1000.0                AS avg_duration_sec
,       NULL                                                AS row_count
,       [resource_class]                                    AS resource_class
,       LEFT(command,50)                                    AS command
FROM    sys.dm_pdw_exec_requests  AS req
WHERE   [request_id] = @req
GROUP BY [label]
,        [resource_class]
,        [command]
;
```

Monitoring execution request steps

```
SELECT      'sys.dm_pdw_request_steps'                AS DMV
,           step.[operation_type]                     AS operation_type
,           step.[location_type]                      AS location_type
,           step.[step_index]                         AS step_index
,           DATEDIFF(ms ,MIN([start_time])
                        ,max([end_time]))/1000.0      AS duration_sec
,           MIN([start_time])                         AS min_start_time
,           MAX([end_time])                           AS max_end_Time
,           MIN([total_elapsed_time])/1000.0          AS min_duration_sec
,           MAX([total_elapsed_time])/1000.0          AS max_duration_sec
,           AVG([total_elapsed_time])/1000.0          AS avg_duration_sec
,           SUM([row_count])                          AS row_count
,           NULL                                      AS resource_class
,           LEFT(step.[command],50)                   AS command
FROM        sys.dm_pdw_request_steps step
WHERE       [request_id] = @req
GROUP BY step.[operation_type]
,        step.[location_type]
,        step.[step_index]
,        step.[command]
;
```

Bringing execution requests and steps together

```
SELECT      'sys.dm_pdw_sql_requests'                    AS DMV
,           step.[operation_type]                        AS operation_type
,           step.[location_type]                         AS location_type
,           step.[step_index]                            AS step_index
,           DATEDIFF(ms ,MIN(sreq.[start_time])
                        ,MAX(sreq.[end_time]))/1000.0    AS duration_sec
,           MIN(sreq.[start_time])                       AS min_start_time
,           MAX(sreq.[end_time])                         AS max_end_Time
,           MIN(sreq.[total_elapsed_time])/1000.0        AS min_duration_sec
,           MAX(sreq.[total_elapsed_time])/1000.0        AS max_duration_sec
,           AVG(sreq.[total_elapsed_time])/1000.0        AS avg_duration_sec
,           SUM(sreq.[row_count])                        AS row_count
,           NULL                                         AS resource_class
,           LEFT(step.[command],50)                      AS command
FROM        sys.dm_pdw_sql_requests  sreq
JOIN        sys.dm_pdw_request_steps step   ON  sreq.[step_index]      = step.[step_index]
                                            AND sreq.[request_id]      = step.[request_id]
WHERE       step.[request_id] = @req
GROUP BY    step.[operation_type]
,           step.[location_type]
,           step.[step_index]
,           step.[command]
;
```

Monitoring worker activity

```
SELECT      'sys.dm_pdw_dms_external_work'          AS DMV
,           [type]                                  AS worker
,           DATEDIFF(ms ,MIN([start_time])
                        ,max([end_time]))/1000.0    AS duration_sec
,           MIN([start_time])                       AS min_start_time
,           MAX([end_time])                         AS max_end_Time
,           SUM([bytes_processed])/1000000000.0     AS sum_GB_processe
,           NULL                                    AS AVG_throuphput_MB_sec
,           NULL                                    AS SUM_throuphput_MB_sec
,           MIN([total_elapsed_time])/1000.0        AS min_duration_sec
,           MAX([total_elapsed_time])/1000.0        AS max_duration_sec
,           AVG([total_elapsed_time])/1000.0        AS avg_duration_sec
FROM        sys.dm_pdw_dms_external_work
WHERE       [request_id] = @req
GROUP BY    [type]
;
```

Monitoring data movement workers

```
SELECT      'sys.dm_pdw_dms_workers'                AS DMV
,           [type]                                  AS worker
,           DATEDIFF(ms ,MIN([start_time])
                        ,max([end_time]))/1000.0    AS duration_sec
,           MIN([start_time])                       AS min_start_time
,           MAX([end_time])                         AS max_end_Time
,           SUM([bytes_processed])/1000000000.0     AS sum_GB_processed
,           AVG([bytes_per_sec])/1000000.0          AS AVG_throuphput_MB_sec
,           SUM([bytes_per_sec])/1000000.0          AS SUM_throuphput_MB_sec
,           MIN([total_elapsed_time])/1000.0        AS min_duration_sec
,           MAX([total_elapsed_time])/1000.0        AS max_duration_sec
,           AVG([total_elapsed_time])/1000.0        AS avg_duration_sec
FROM        sys.dm_pdw_dms_workers
WHERE       [request_id] = @req
GROUP BY    [type]
;
```

Azure Data Factory Integration

ADF components

Linked services
Datasets
Activities

} Pipeline

PolyBase Pre-requisites: Linked Service

Azure Storage source only

No SAS authentication

Azure Storage Linked Service

```
{
    "name": "<ASBLinkedServiceName>"
,   "properties":
    {   "hubName": "DWfactory_hub"
    ,   "type": "AzureStorage"
    ,   "typeProperties":
        {
            "connectionString":
"DefaultEndpointsProtocol=https;AccountName=jrjtrip2015;Accoun
tKey=*********"
        }
    }
}
```

SQLDW Linked Service

```
{
    "name": "<SQLDWLinkedServiceName>"
,   "properties":
    {   "description": ""
    ,   "hubName": "DWfactory_hub"
    ,   "type": "AzureSqlDW"
    ,   "typeProperties":
        {
            "connectionString": "Data
Source=<server>.database.windows.net;Initial Catalog=<db>;Integrated
Security=False;User ID=<user>;Password=**********;Connect
Timeout=30;Encrypt=True"
        }
    }
}
```

PolyBase Pre-requisites: Input Dataset

Azure Blob Properties:

Input Dataset: Azure Blob

Type: TextFormat

rowDelimiter: \n

nullValue: ""

encodingName: utf-8 (default)

escapeChar: not specified in activity

quoteChar: not specified in activity

Input Dataset

```
 "typeProperties":
{   "folderPath": "<blob_path>"
,   "format":
    {
        "type": "TextFormat"
    ,   "columnDelimiter": "<any delimiter>"
    ,   "rowDelimiter": "\n"
    ,   "nullValue": ""
    ,   "encodingName": "utf-8"
    }
,   "compression":
    {
        "type": "GZip"
    ,   "level": "Optimal"
    }
}
```

PolyBase Pre-Requisites: Copy Activity

Blob Source Properties:

skipHeaderLineCount: not specified

SqlDWSink:

sliceIdentifierColumnName: not specified

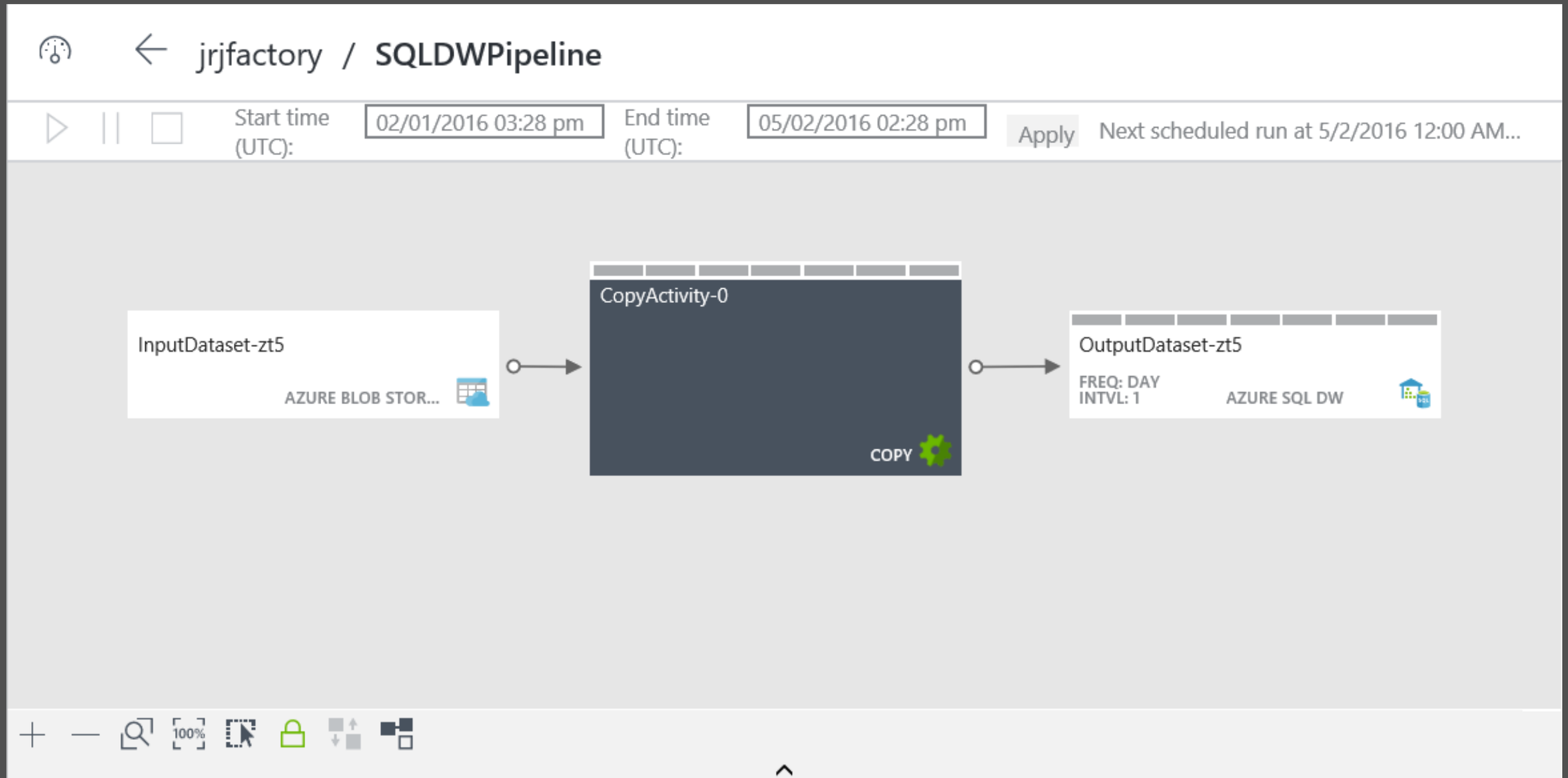
Copy Activity:

columnMapping: not specified

Copy Activity

```
 "sink":
{ "type": "SqlDWSink"
, "writeBatchSize": 1000000
, "writeBatchTimeout": "00:05:00"
, "allowPolyBase": true
, "polyBaseSettings":
    { "rejectType": "percentage"
    , "rejectValue": 10
    , "rejectSampleValue": 100
    , "useTypeDefault": true
    }
}
```

Copy Activity Wizard



ADF Limitations

Primary limitations

One time sync can’t be edited

PolyBase can’t be configured in Copy Wizard (today)

File headers must be addressed

ADF validates the data types of the data in the source

Fields must all map to string if headers are present

Use another copy activity (blob to blob) to remove the header from the source

Avoiding column mappings

Input names must equal output names

Data types must match

Demo:
Using ADF

Lab:

Loading Data with ADF

Microsoft Azure

Recommendations

Data preparation

Transfer data to blob storage

One root folder per table

Sub-folders for partitions / subset analysis

Split table data into multiple files: 1 file for each reader

Compress data to optimise transfer

Initial load

CTAS data with PolyBase for max throughput

One external table definition per table

Configure load user

Size the rowgroup for memory grant

Set appropriate resource class

Maximise # readers to accelerate load

DWU1000+ for 60 readers

Multiply #files by readers for balanced throughput (i.e. 60,120,180 etc.)

Lab review

1. What is the purpose of the load user in Azure SQL Data Warehouse?

2. What is the fastest method for loading data in Azure SQL Data Warehouse?

3. What is an external format file, and its' purpose

4. What does CTAS and CETAS stand for? What is the difference?

5. What is the wizard that can be used in Azure Data Factory to export data?

Summary

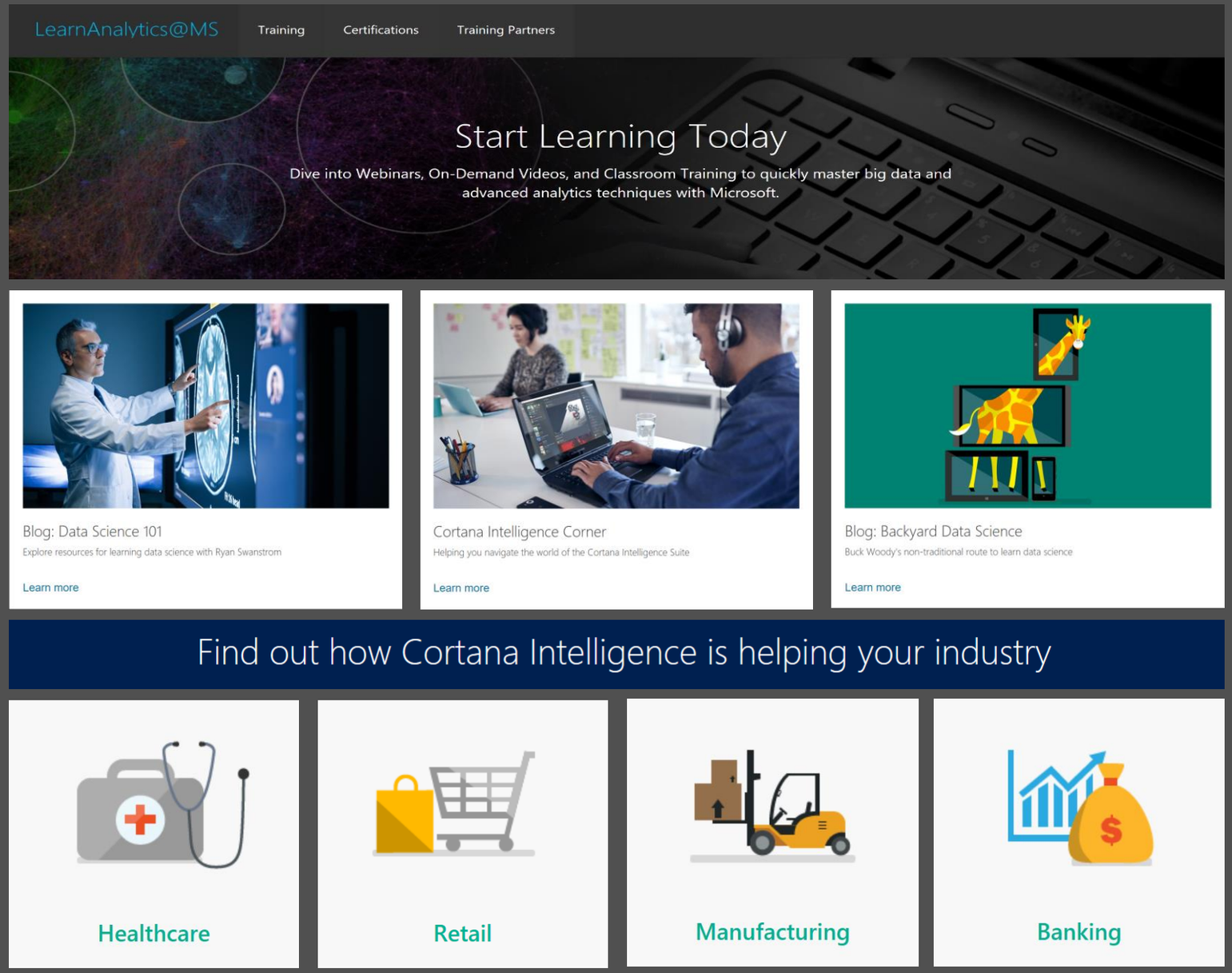
Summary

- The role of the load user.
- The different methods for loading data.
- How to use PolyBase.
- Importing and Exporting Data.
- Monitoring Data Loads.
- Using ADF to load SQL DW.
- Loading recommendations.

Microsoft

There are more learning options as shown in the links on the right, including:

- Online training
- Videos
- Instructor Led training
- Blogs
- Cortana Intelligence Gallery



Course Documentation

Accessing the course materials

1. Click on the picture on the left.
2. Sign in with your Live ID.
3. Look for the SQLW301 item.
4. Click on the course materials link.