



Data Loading Patterns

Agenda

- Environment & Options
- Tools
- Process
- Export
- DMVs
- Best Practices

Environment & Options

Load Server

Reference Architecture

- Customer Domain
- Flexible Configuration

Connect to PDW via

- 1GbE
- 10GbE
- Infiniband 56GbE

Also good location for

- Job scheduling
- File exports
- SMP Functionality

You can always have >1 Loading Server attached to a single PDW. Benefits are performance and / or availability

Loading Goals

To Load Data

- Efficiently
- Unobtrusively
- Resiliently
- Optimally

In other words...

- Fast
- Mixed Workload
- Self healing
- Read Performance

Loading Modes

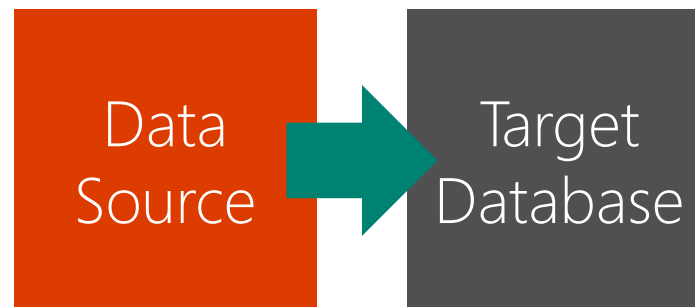
PDW offers three standard bulk loading modes

- Append
- Reload
- Upsert



And one optimised mode

- FastAppend



Loading Options

Command Line
dwloader.exe

File Based

SSIS
2010 & 2012

Heterogeneous
Sources

Polybase
Hadoop
Azure

File Based

3rd Party Loading options

- Informatica PowerCenter (versions up to 9.5.1)
 - Windows environments only
 - Default operation row by row
 - PDW Loader – Bulk functionality
- SAP Business Objects Data Integrator
- Attunity Replicate
 - Trickle loading using dwloader under the hood

Loading Limitations

Maximum 10 concurrent loads across appliance

- SSIS – 10 destination adaptors sending
- dwloader – 10 load commands

Queue

- Up to 40 loads will be queued
- 51st concurrent load - error

Loading Limit is applied across all loading methods

Dwloader and SSIS

dwloader.exe – Parameters

- W Trusted Authentication
- U login
- P password
- S target appliance
- T target table (3 part name)
- d staging database
- D date format
- dt data format file
- e character encoding
- E empty strings to NULL
- i input file location
- s string delimiter
- t field delimiter
- r row_delimiter
- fh header_rows

dwloader.exe Parameters

- se skip empty files
- c remove white space
- rv reject value
- rt reject value or percentage
- rs reject sample size
- M Load Mode
 - Append
 - FastAppend
 - Reload
 - Upsert
- m Multi Transaction Mode

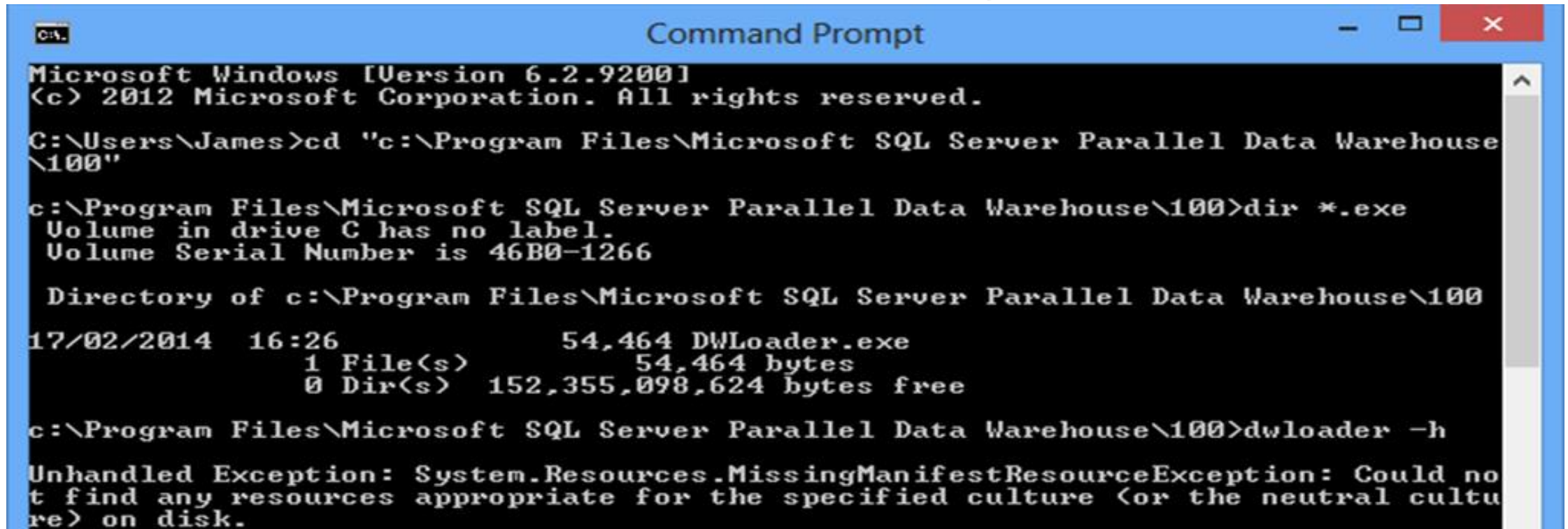
dwloader

Uses some non-Microsoft standard parameters

- -h help
- -W trusted authentication

Dwloader Limitation

- Dwloader has limited locale support
- Server OS must be EN-US region



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\James>cd "c:\Program Files\Microsoft SQL Server Parallel Data Warehouse\100"

c:\Program Files\Microsoft SQL Server Parallel Data Warehouse\100>dir *.exe
Volume in drive C has no label.
Volume Serial Number is 46B0-1266

Directory of c:\Program Files\Microsoft SQL Server Parallel Data Warehouse\100

17/02/2014  16:26                54,464 DWLoader.exe
               1 File(s)                54,464 bytes
               0 Dir(s)  152,355,098,624 bytes free

c:\Program Files\Microsoft SQL Server Parallel Data Warehouse\100>dwloader -h

Unhandled Exception: System.Resources.MissingManifestResourceException: Could not find any resources appropriate for the specified culture (or the neutral culture) on disk.
```

Breaking Change in AU1

- Date Format -D
 - ymd
 - Mdy
 - Myd
 - Ydm
 - Dmy
 - Dym
 - Custom_date_format

SSIS

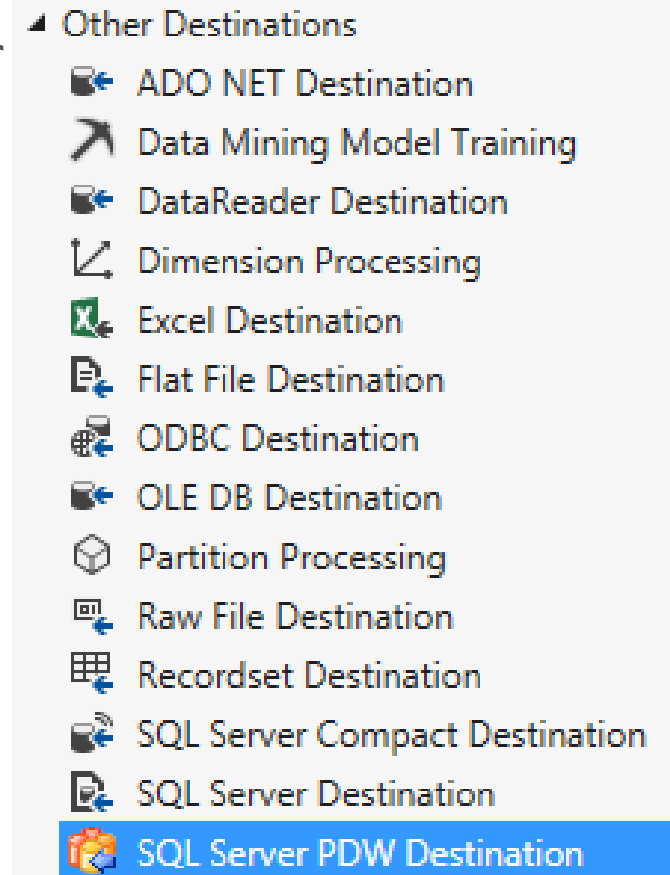
Destination Adaptors available for

- SSDT 2010
- SSDT 2012

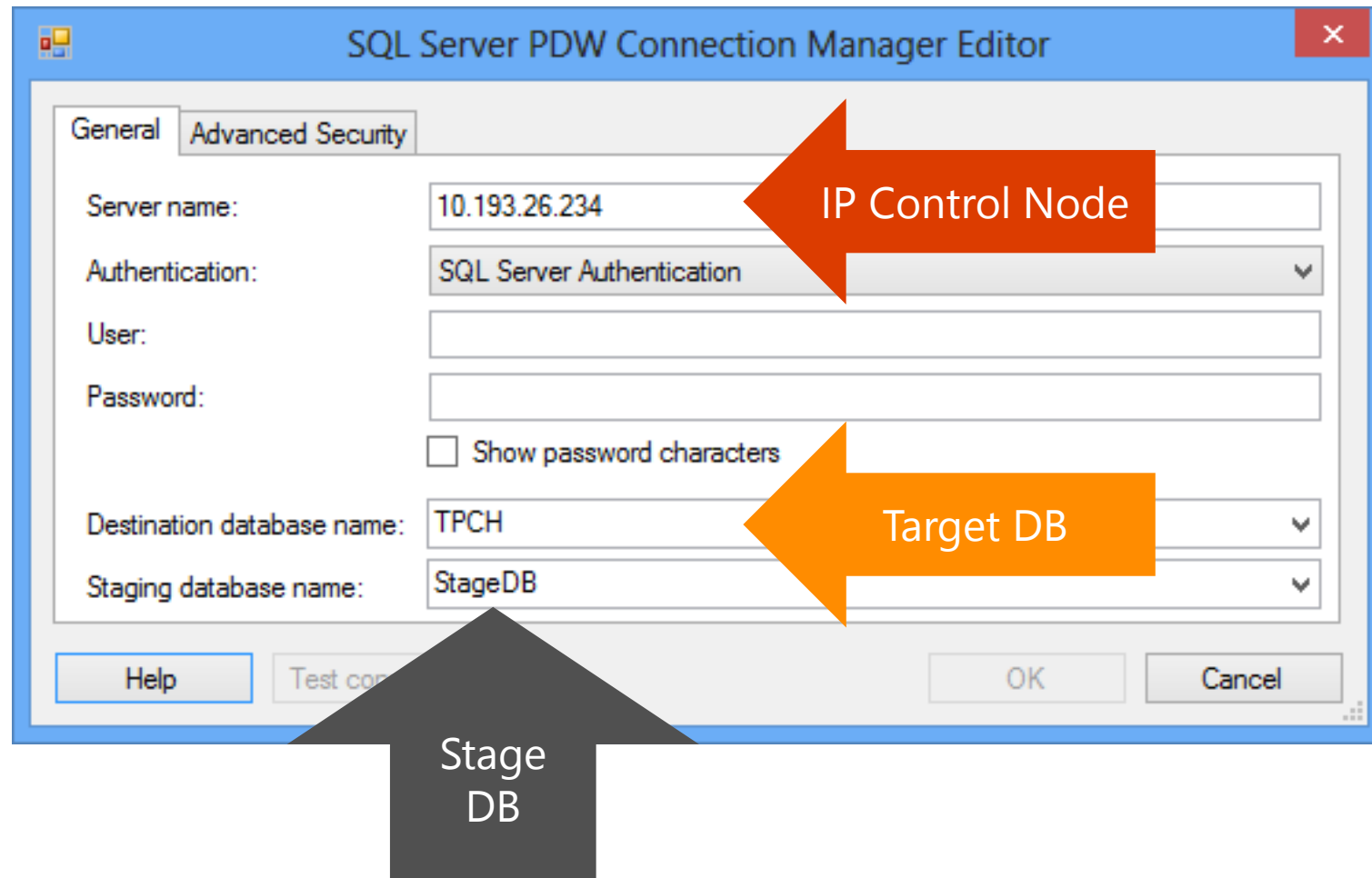
Shipped on the Control Node

- C:\PDWInst\ClientTools

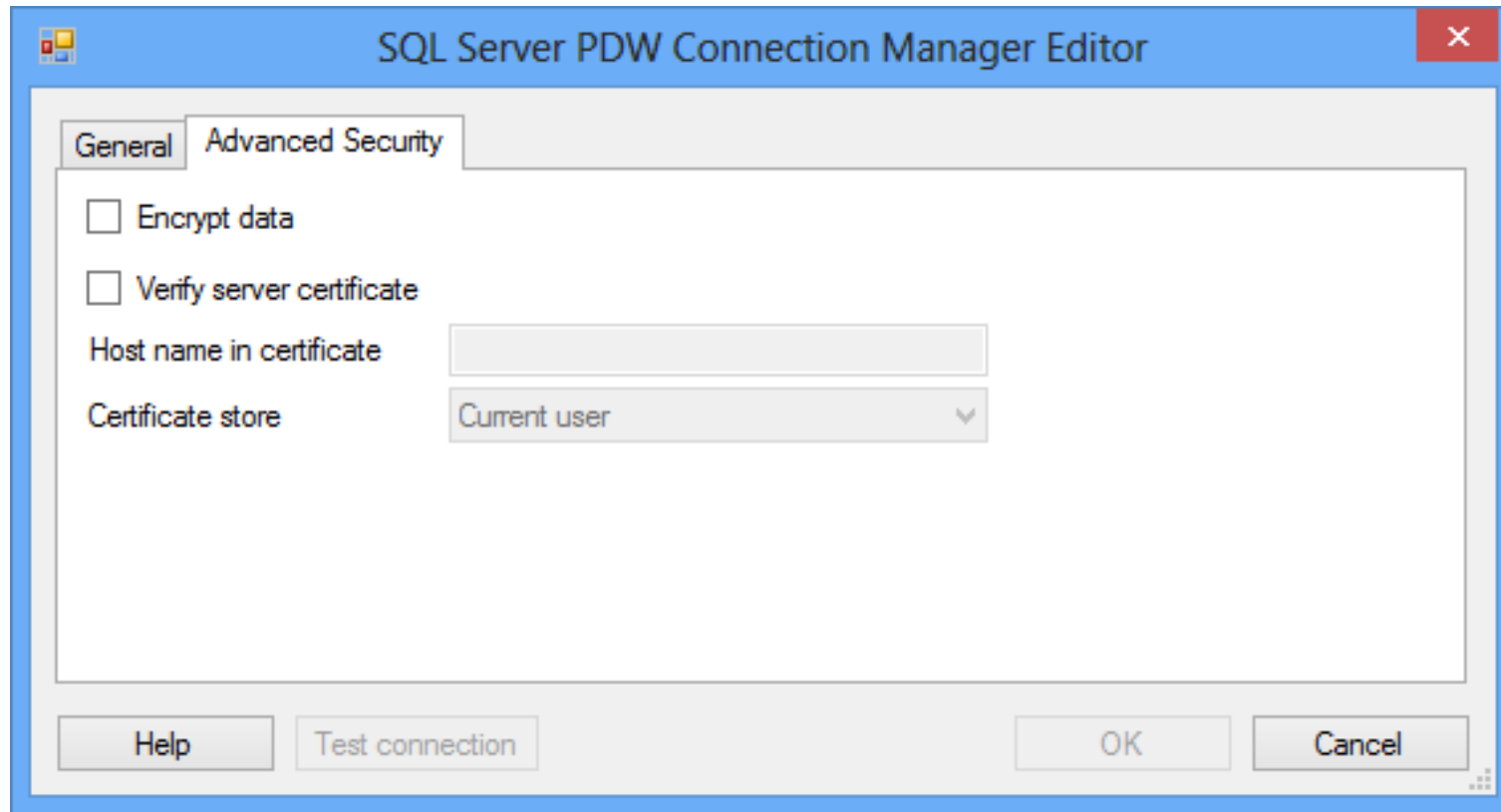
Uninstall previous version first



SSIS Connection Manager



Securing SSIS Connection



Inside the Destination Adapter

The screenshot shows the 'SQL Server PDW Destination Editor' window. The 'Destination' tab is active, showing the following settings:

- Connection manager: PDW
- Destination table: [tpch_orders]
- Loading mode: Append
- Roll-back load on table update or insert failure: ☒

The 'Error Handling' tab is collapsed. The 'Column mapping' tab is active, showing the following columns mapped:

Input	Destination
o_orderkey	o_orderkey
o_custkey	o_custkey
o_orderstatus	o_orderstatus
o_totalprice	o_totalprice
o_orderdate	o_orderdate
o_orderpriority	o_orderpriority
o_clerk	o_clerk
o_shippriority	o_shippriority
o_comment	o_comment

The 'Unmapped Input Columns' list is empty. The 'Unmapped Destination Columns' list is also empty.

-m in dwloader

Load Mode

AU1 Tool RTM Version Numbers

Dwloader

- 10.0.4721.1

SSIS

- 11.0.4721.1



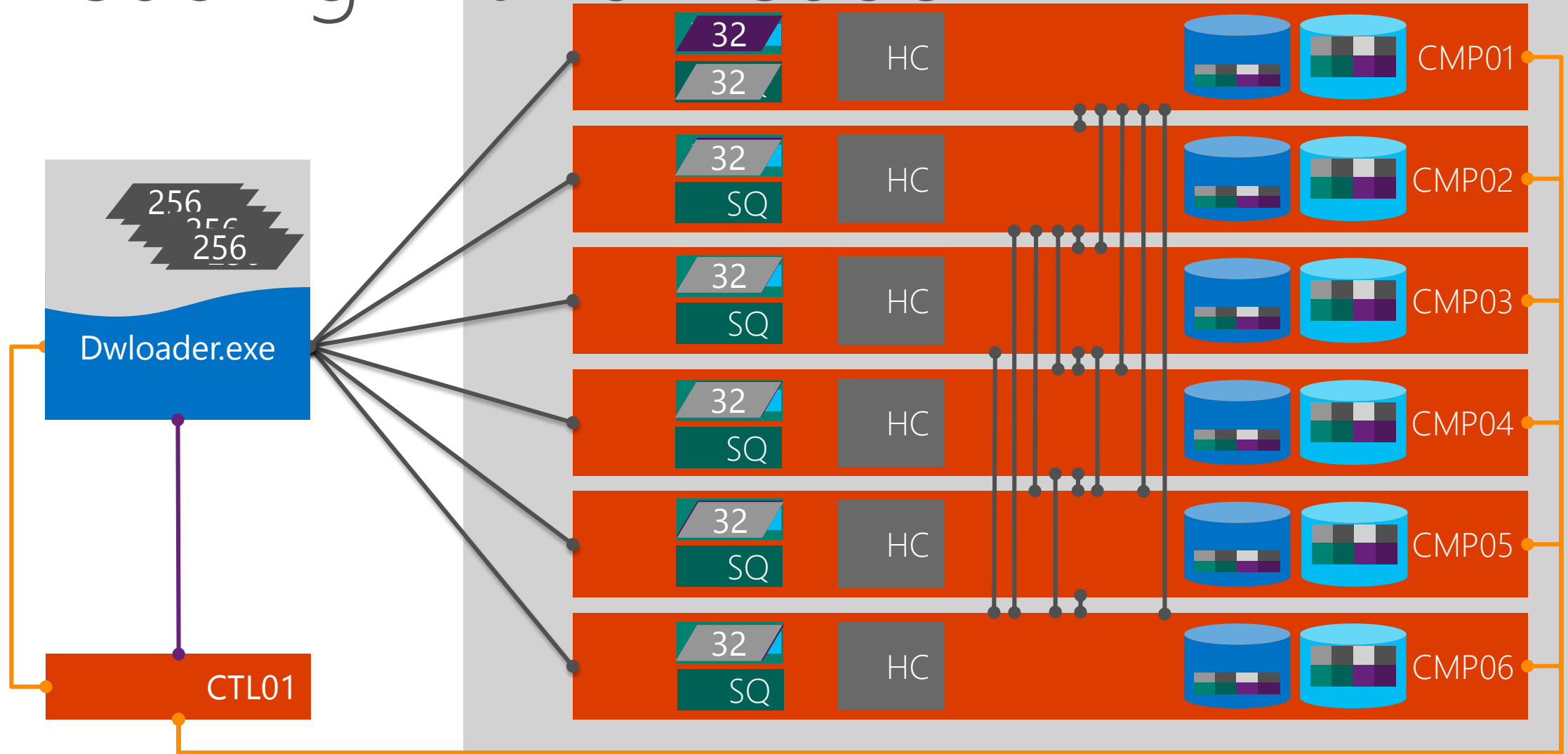
Why
10.0?

PDW uses version numbers as defined interface numbers not for product build versions. Currently the interface supported is Katmai level i.e. 10.0. The APS product build number is the 4721 figure.

Loading Process

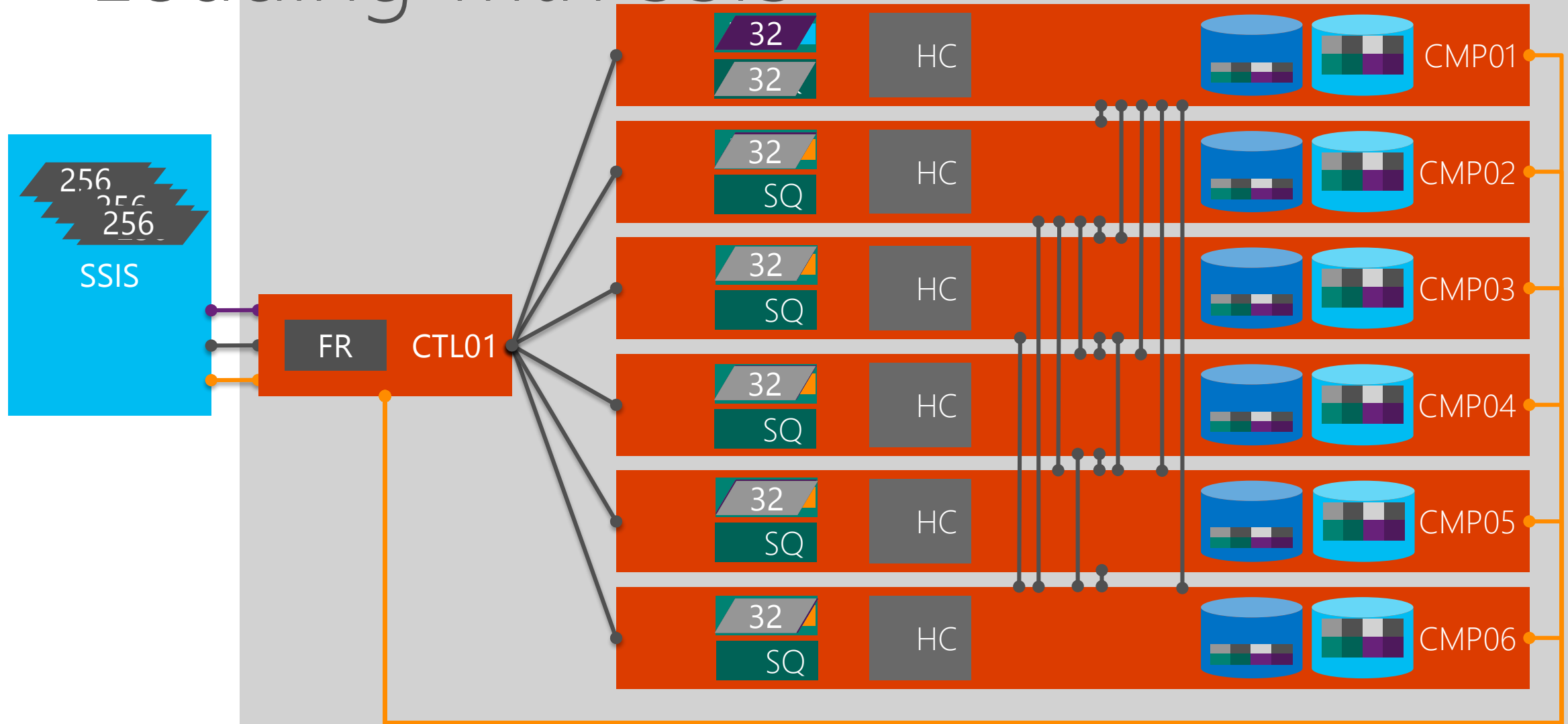
Loading with dwloader

4. Load Cleanup



Loading with SSIS

4. Load Cleanup



When Loading Replicated Tables

Reader Worker Type is different

- Direct_Converter not hash converter
- No hash required

Loading takes longer

- Entire data set is written to each node

Loading Example: 1bn rows 6 nodes

Replicated Table

- 1bn rows * 6 nodes
- 6 target tables
- 1bn rows per node
- 6bn rows in total

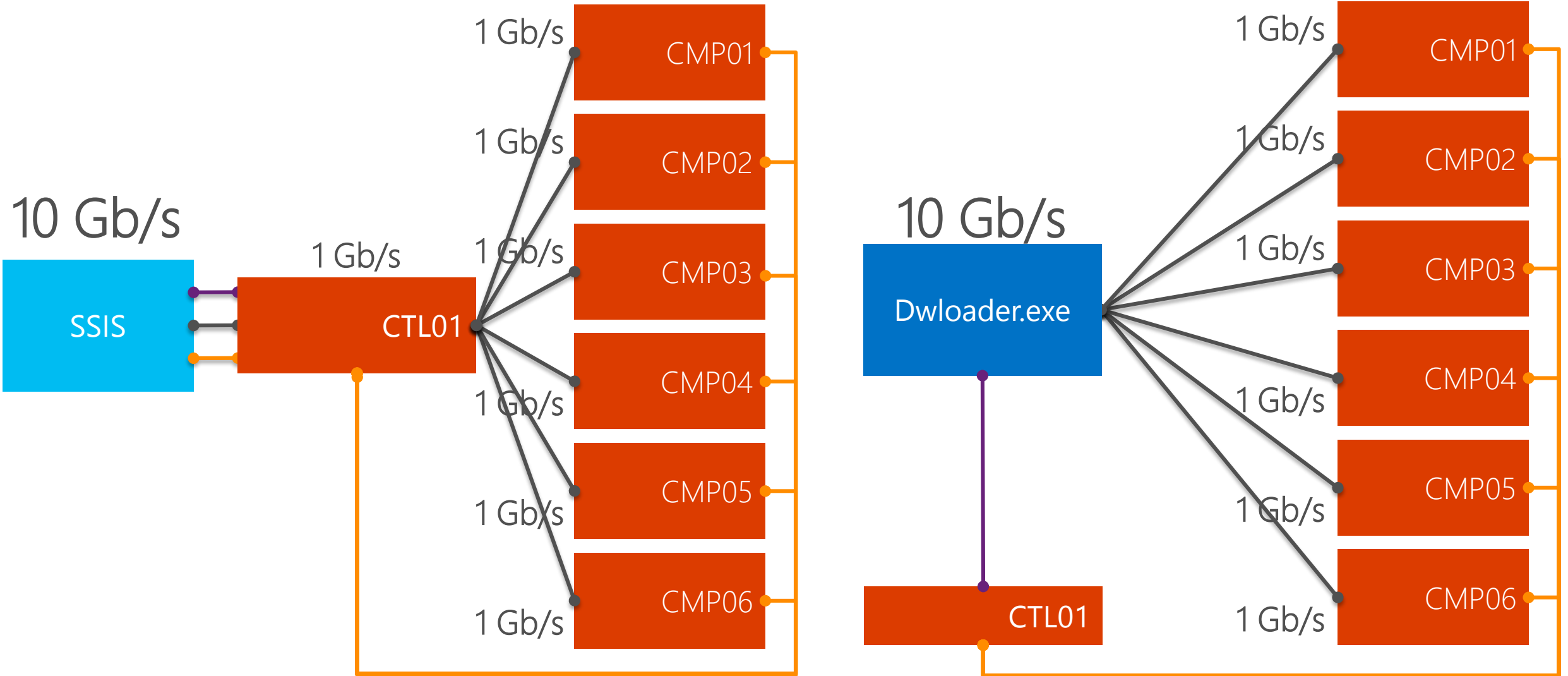
Distributed Table

- 1bn / 6 nodes
- 48 target tables
- ~166.7m rows per node
- ~20.8m rows per distribution

Loading Process Notes

- 1 reader worker per distribution
- 1 writer worker per distribution when target is indexed
- >1 writer worker per distribution used when target is a heap – up to 8x uplift for replicated tables
- 2 buffers are allocated per reader
- 6 node system = 96 buffers

Distributed Data Load & dwloader



dwloader vs SSIS

	Dwloader	SSIS
Initial load target	Compute Nodes	Control Node
Parallel Transfer (Loading to PDW)	Yes	Yes with caveat
Data Access Layer	Native (SQL Native Client)	ADO.NET (SQLClient)
Conversion Location	Compute Nodes	ETL Server
Converter	Hash/Direct Converter	SSIS Destination Adaptor
No of Conversions	1 Text -> Binary (ODBC)	3 Text -> .NET (SSIS) .NET -> Binary (Dist Key) .NET -> Binary (Bulk Load)

SSIS Parallel Buffer Transfer Caveat

Data buffers transferred to PDW on multiple parallel sessions when using

- SQL Server 2012 plus Cumulative Update Package 5 (or later)
- SQL Server 2012 Service Pack 1 plus Cumulative Update Package 2 (or later)

If the CU Packages are not applied:
SSIS 2012 will transfer data to PDW serially

Loading performance - dwloader

	Source File Size (GB)				Throughput (TB /Hour)			
	1	2	4	8	1	2	4	8
	User	User	User	User	User	User	User	User
AU1	22.2	44.4	NULL	177.4	1.9	2.7	NULL	3.6
AU0.5	22.2	44.4	88.7	177.4	1.3	1.7	2.0	1.7

TPCH Lineitem – 8 compute nodes

Loading performance - SSIS

	Source File Size (GB)				Throughput (GB /Hour)			
	1 User	2 User	4 User	8 User	1 User	2 User	4 User	8 User
AU1	7.3	14.6	NULL	58.4	83.7	166.5	NULL	536.2
AU0.5	7.3	14.6	29.2	58.4	83.6	166.6	334.1	515.1

Export

PDW Export options

- CRTAS -> SQL Server
- CETAS -> HDFS, WASB
- SQLCMD -> Flat File
- SSIS -> Anything
- Power Query -> Excel

BCPing out data from SMP

Use views over source tables

- manage column selection
- Transform types
- Hide user defined schemas
- Parallelise export using modulus
- ASCII encoding files faster to load
- Export with UCS2 or UTF-16 for files containing special characters

Microsoft tooling

- Ask your TSP for the latest SMP to MPP migration script
- Schema migration scripts also available

DMVs

Loader DMVs

- Run_ID
- Request_ID is a GUID as opposed to QID

DMVs

- sys.pdw_loader_backup_runs
- sys.pdw_loader_run_stages
- sys.dm_pdw_dms_workers

DMVs for load

```
SELECT *
FROM      sys.pdw_loader_backup_runs r
JOIN      sys.server_principals      p ON r.principal_id      = p.principal_ID
JOIN      sys.pdw_loader_run_stages  s ON r.request_id        = s.request_id
LEFT JOIN sys.dm_pdw_dms_workers     w ON s.request_id        = w.request_id
                                                AND s.start_time    <= w.start_time
                                                AND s.end_time      >= w.end_time

LEFT JOIN sys.dm_pdw_nodes            n ON w.pdw_node_id      = n.pdw_node_id
LEFT JOIN sys.pdw_distributions       d ON w.pdw_node_id      = d.pdw_node_id
                                                AND w.distribution_id = d.distribution_id
```

Recommendations

General Recommendations

- Make sure you're on the latest version
- Always pass in target appliance for enhanced code portability (-S)
- Ignore batch size parameter
 - Size is now fixed to 1048576

More Recommendations

- Always use a Staging Database
 - Isolates staging tables minimizing fragmentation
 - Helps to accurately size target database
- Use wildcards in filenames
 - Consider file naming convention for importing multiple files in one load

Best Practices

Parallel load to one table

Background

- Massive data load to a big table
 - 100s of GB or TB
- Even with 1+Tb/h can take time to load
- Goal
 - Optimize data load to provide best possible performance

Parallel load to one table

Prerequisites

- Table is partitioned
- Load can be split by partition boundary
 - By month
 - By Region

Recommendation

- Stay with 10 concurrent loads

Parallel load to one table

Solution

- Create table per partition boundary
- Load data to each table in parallel
 - Switch partitions to the target table

```
ALTER TABLE claim_parallel_stage_1 SWITCH PARTITION 1 TO claim PARTITION 1
ALTER TABLE claim_parallel_stage_2 SWITCH PARTITION 2 TO claim PARTITION 2
ALTER TABLE claim_parallel_stage_3 SWITCH PARTITION 3 TO claim PARTITION 3
...
```

Hint

- Partition switch is a metadata based operation

Data reloading / partition boundary

Background

- Data partitioned by day in SMP
 - Using “partition elimination” for improved performance
- Customer may re-load data for several days
 - Common retail scenario – “tail”

Data reloading / partition boundary

Goal

- Enhanced flexibility
 - Data re-load
 - Easy maintenance

Data reloading / partition boundary

Solution

- Use month boundary
- Switch data OUT from required partition for modification
- Update or CTAS to modify
- Switch partition back IN

ELT vs ETL

ELT

- Set operations
- Heavy lift on appliance
 - 128 - 144 cores (full rack by vendor)
 - 2 - 2.25 TB RAM (full rack by vendor)
- Simpler framework
- Distributed load by design

ETL

- Row-by-Row process
- Heavy lift on loading server
 - Custom hardware specification
 - Unlikely to match appliance
- Distributed load complex

ELT vs ETL

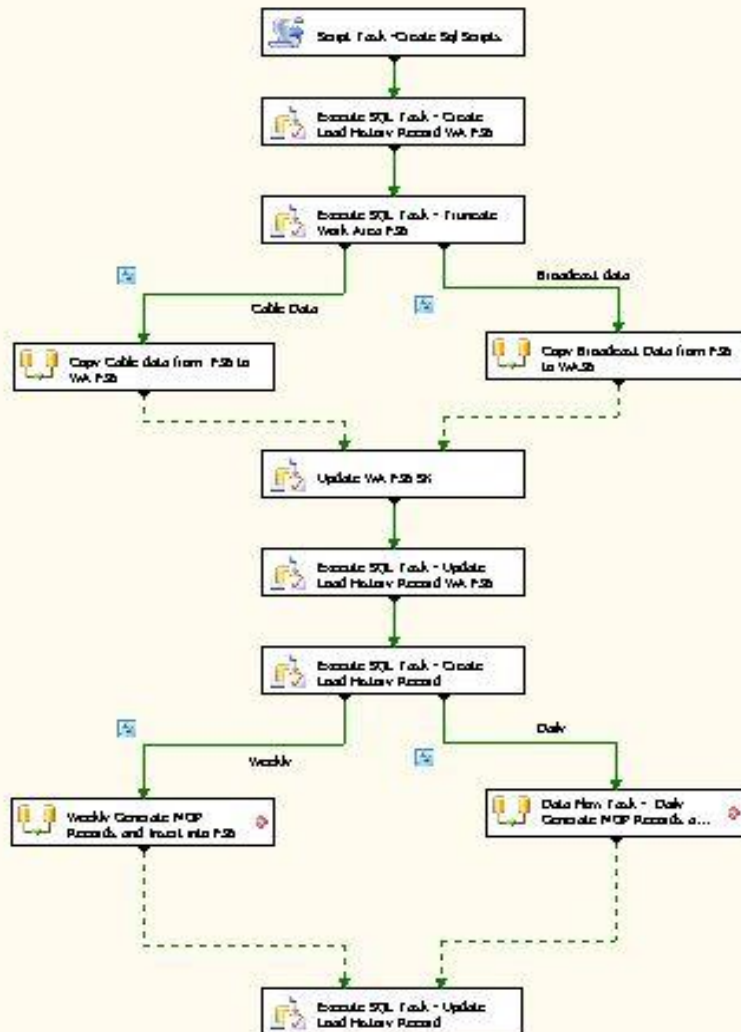
ELT: Partial SMP fidelity

- Optimal load speed
 - Combine multiple steps
 - Multiple surrogate key lookups (full cache lookups)
 - Multiple updates to a single table
 - Re-architect solution for maximum throughput

ETL: Full SMP fidelity

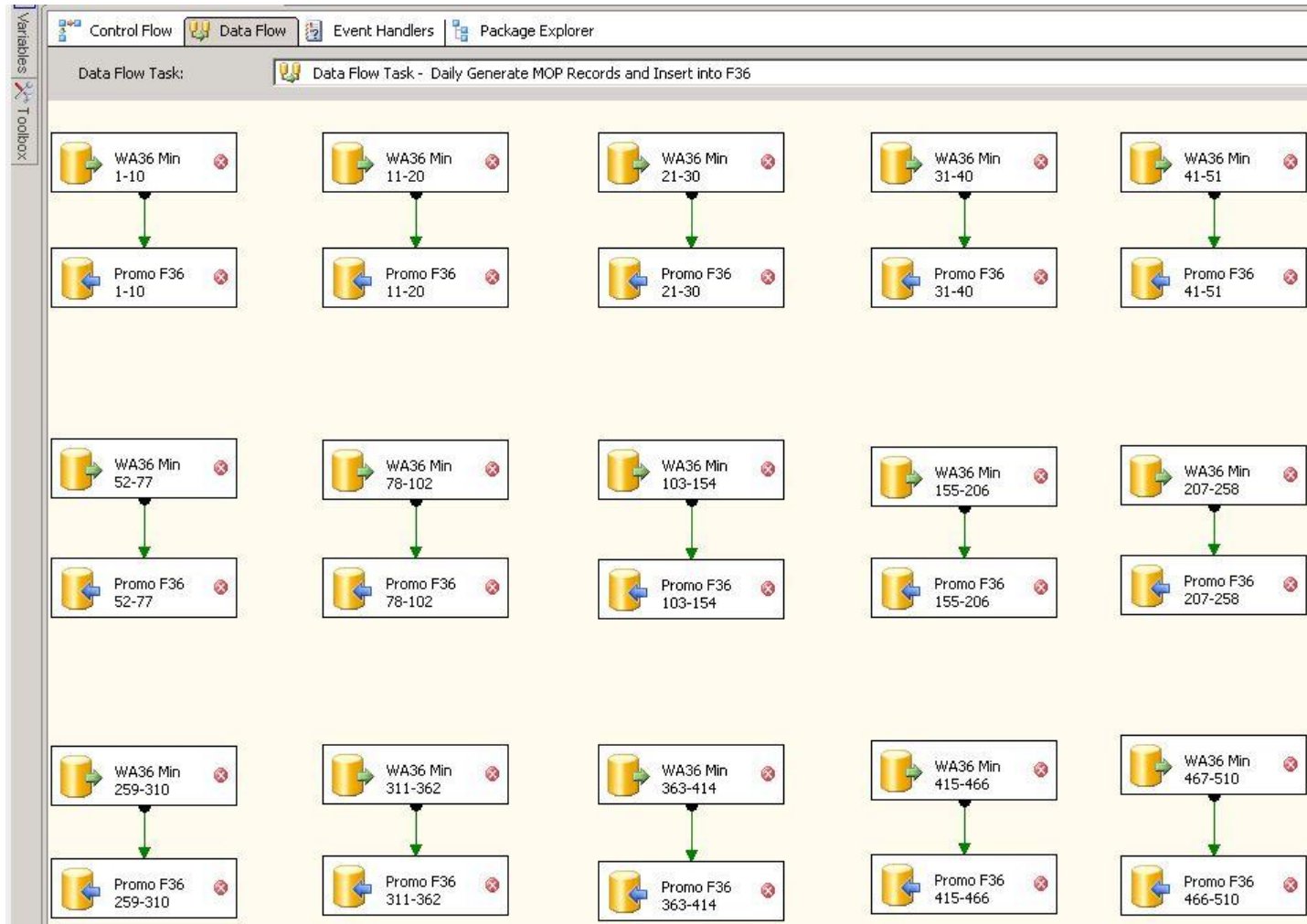
- May still need change
 - Manual Parallelism removal
 - Excessive connections

ELT vs ETL : Before



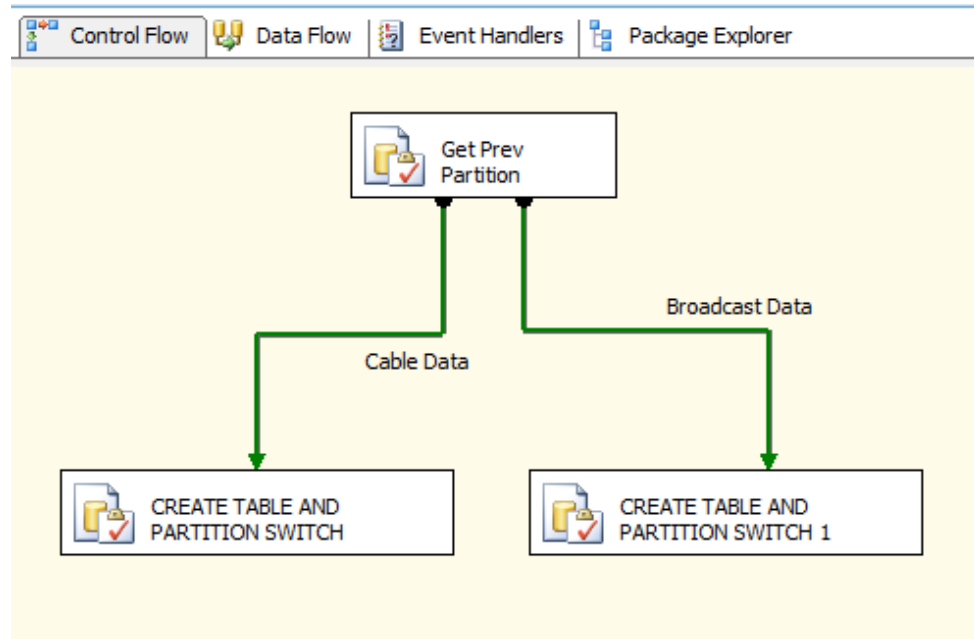
- Step 1 : Copy data to Stage
- Step 2 : Lookup Match
- Step 3 : Update Surrogate Keys
- Step 4 : ??

ELT vs ETL: Before



- Parallelize Process
- Expand for each Minutes

ELT vs ETL: After



Combine steps!

Create table as select (CTAS)

- Combine surrogate key lookups
- Cartesian product for each minute viewed

Switch partitions

- Target destination table

Best Practices : ELT vs ETL : Results

Process	PDW Time	Current Environment
Load to Repository (full package)	11 min 33 sec	1.5 Hours
Load to Target		
Load Ratings (Cable)	3 min 8 sec	1.5 Hours
Load Ratings (Broadcast)	15 sec	
Load Duplication	* 6 min 11 sec	
Load HH Duplication	6 sec	
Load Households	28 sec	
Load Persons	20 sec	
Load Programs	9 sec	
Load UE	3 sec	

ELT vs ETL : Results

Setup	Load and Transform	Remote Table Copy	Cube Processing	Total Processing Time
Current Environment	5hrs +		< 33 min	~ 6 hrs
Target				< 2 hrs
PDW	30 minutes			30 min + Cube

Connection Speed

- If you saw this... should check this...

