Perth MS Cloud User Group
December 2016

Azure SQL DW

Lessons from the Field

Bhavik Merchant
Data Platform Solution Architect
Microsoft Australia
bhmerc@microsoft.com

Microsoft

Session Goals

- Drink beer and eat pizza ;)
- Know what SQL DW is
- Understand real client scenario
  - **Focus on getting data in**
- Follow my journey and lessons learnt
- Get some guidance to take into practice
- Learn about some limits

Overview

What is Azure SQL DW?

- Cloud based, scale out DB
- Designed for **massive** data volumes
  - Approx 5x compression. Support 240 Tb **compressed** user data
- Built on MPP architecture
- Separates compute and storage
- Flexible: Scale out, scale back, pause/resume
- Allows seamless querying over Hadoop
  - Can leverage this with minimal setup

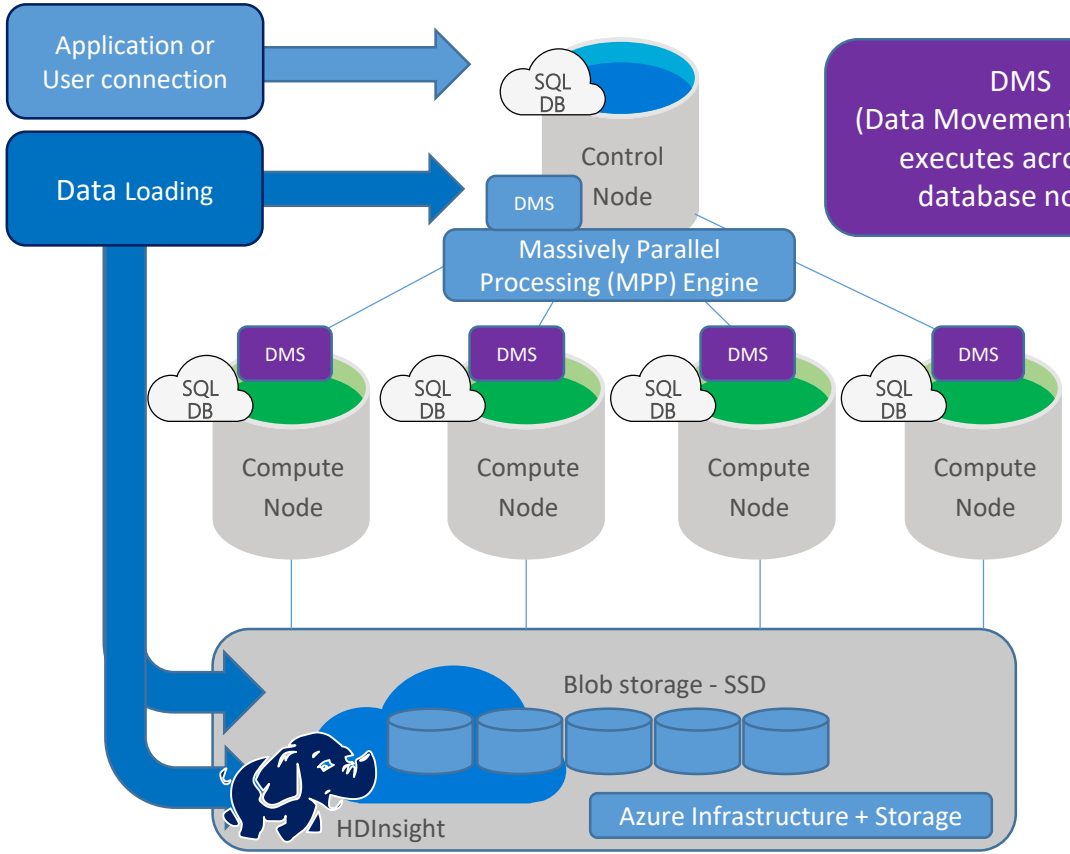
What are the Benefits?

- Deploy within 10 minutes
  - Almost zero architecting, configuring, tuning*
- Use familiar paradigms
  - SQL tables, stored procs, indexes, partitions
  - T-SQL
- Cost control via elasticity – think peak/off-peak
- Hybrid architectures possible (sensitive vs non)
- Familiar analytical tools – Power BI, Excel, SSRS

Internals

What's in the Box?

Storage and Compute are de-coupled, enabling a true elastic service and separate charging for both compute and storage

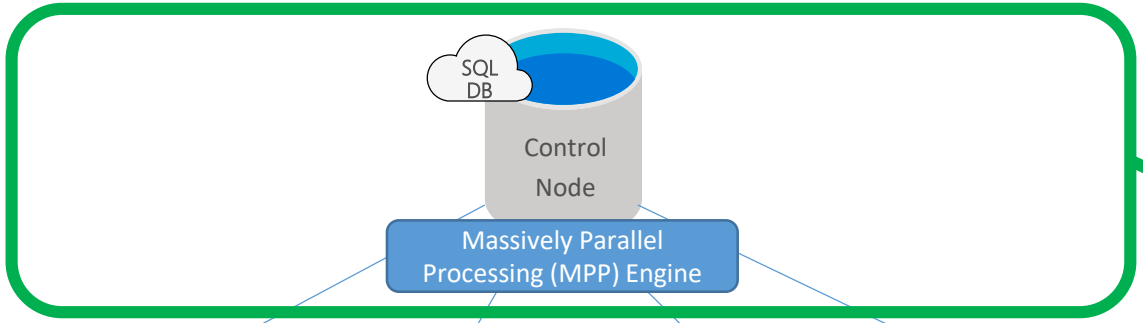


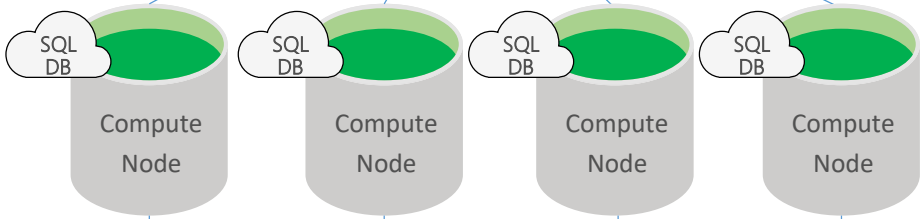
Compute
Scale compute up or down when required

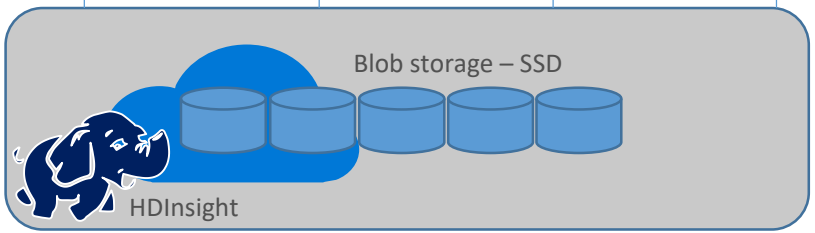
Pause, Restart, Stop, Start.

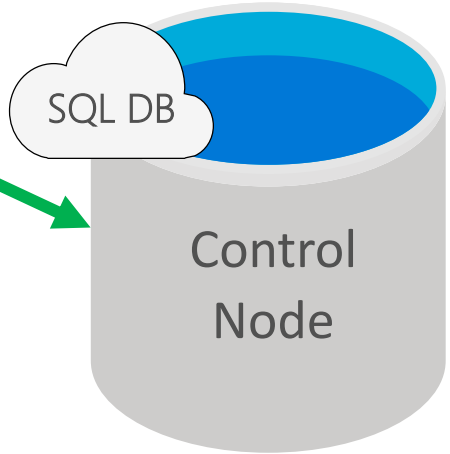
Storage
Add/Load data to WASB(S) without incurring compute costs

Control Node



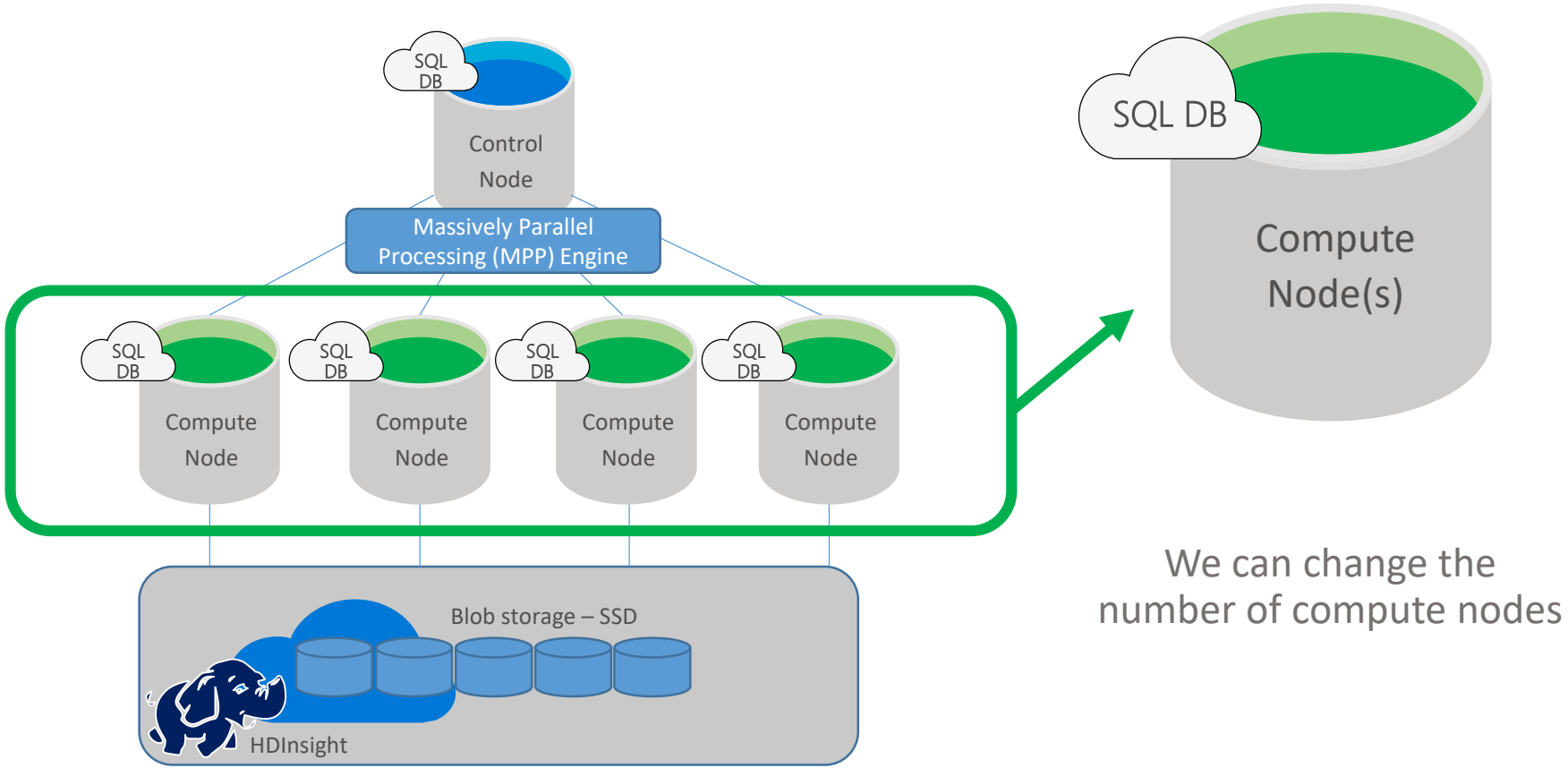






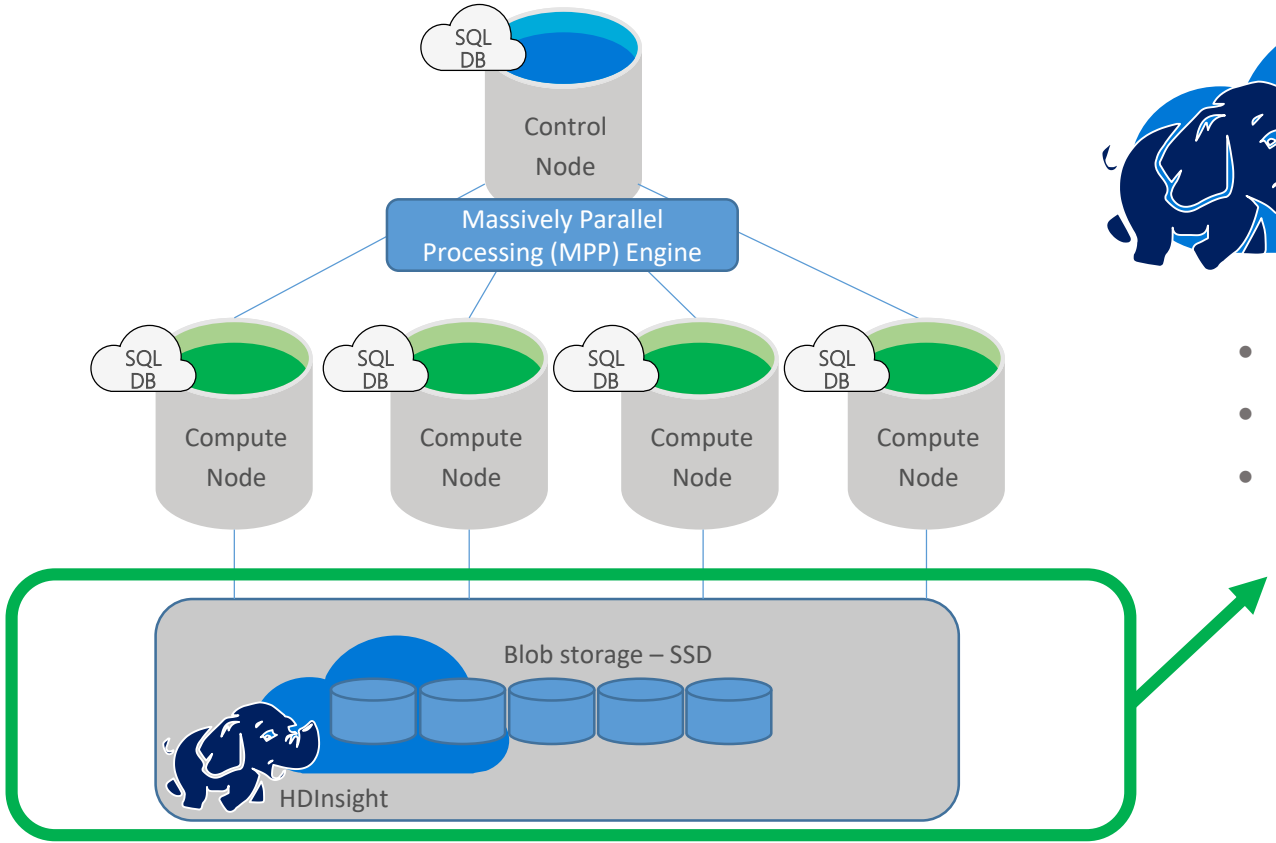
- Endpoint for connections
- Regular SQL endpoint (TCP 1433)
- Coordinates compute activity using MPP

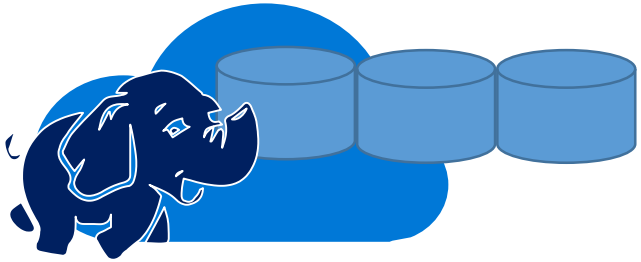
Compute Nodes



We can change the number of compute nodes

Blob storage





- RA-GRS storage
- +PB's of storage
- Load data without incurring compute costs

Customer Scenario

Business Problem

- Anomaly detection on gas plant sensors
- Platform must cater for a range of questions
- Limited self-service capability
- You don’t know what you don’t know
- Queries are ad-hoc and varied - mainly around
  - I want to look at one tag over a range of day/weeks/months
  - I want to look at a group of tags over smaller period
  - I want to be able to average a range of tag values in the same category
  - Etc

Technical Challenges

- We have billions of rows of sensor data
- How to load data to cloud efficiently?
- How to design for performance for wide range of questions?
- How to visualise massive datasets?
  - Can we cater for the inadvertent troublesome user?
  - Even on 4K screen, you can visualise 10s of thousands of points in a trend

My Constraints

- Use existing data formats
- Load the data in a very short time
- Solve the process and visualisation challenges
  - Beyond the scope of this presentation

- The focus of the presentation is getting the data in

Loading Data

Source Data

- Asked the client to dump data to Azure Blob storage
- Millions of gzip files
- Each zip contained ~200-1500 PSV files
- Tens of thousands of “folders”

A Look at the Data

Polybase to the Rescue? In theory…

- Configure credentials and external data source in SQL DW
- Configure external file format in SQL DW
- Define external table with file format
- Load from Blob to SQL DW using CTAS
- Once in DW, get columnstore compression and usual T-SQL benefits

In reality…

- Polybase relies on Hadoop
- Creating external data source killed SQL DW
- Discovered Hadoop limitation
  - 33k files/folders or files per folder

LESSON 1

- Do not overload Polybase/Hadoop with over 33k folders/files per folder

Loading Data... Again

Need a Better Distribution Strategy

- How do I spread the millions of gzip files?
- Look for patterns
- Move more files into less folders
- Try load again

The New Plan

- PowerShell to dump list of files from Blob store
- Bulk load file list to SQL table (SSIS via Wizard)
- Find groups and make a subgroup (pattern) list for each group (T-SQL query)
- Dump patterns to CSV files (SQLCMD via T-SQL)
- Read each pattern in group and execute AZCopy (in parallel via PowerShell)

Visualising the patterns

- 20 Groups -> 100-200 batches -> hundreds of files

Group	Batch (lines in CSV)	Example Filenames that AZCopy would match
EVL01Value.csv	**EVL01Value**1380606	**E01Value1308060**7603720000.xml_17130.psv.gz **E01Value1308060**1972930000.xml_72310.psv.gz **E01Value1308006**7603720000.xml_13049.psv.gz
	EVL01Value1380708	**E01Value1308070**7603720000.xml_17130.psv.gz **E01Value1308070**3849810000.xml_95829.psv.gz **E01Value1308070**8422340000.xml_24524.psv.gz
EVL02Value.csv	**EVL02Value**1380606	**E02Value1308060**7603720000.xml_17145.psv.gz **E02Value1308060**1972930000.xml_84836.psv.gz **E02Value1308006**7603720000.xml_23423.psv.gz
	EVL02Value1380708	**E02Value1308070**7603720000.xml_66239.psv.gz **E02Value1308070**3849820000.xml_16455.psv.gz **E02Value1308070**8422340000.xml_17653.psv.gz

Lets see the Code

More Problems…

- AZCopy is multi-threaded and parallel
  - Cleverly generates storage API calls for you
- BUT
  - It takes forever to do each group!
  - I have limited time
  - I have limited patience

- Can I use faster storage?
- Why not run the groups in parallel?

Modified Approach

- Move to Premium storage (SSD) to make life easier later
- I have 20 groups so..
- Run each group in parallel
  - Do it on an Azure VM
  - Destinations are new premium storage accounts (helps later)
  - 20 CMD windows running PowerShell AZCopy loop to Premium storage
- Sounds good now… right?

Yet More Problems

- PROBLEM 1: Premium storage = only Page Blob
  - AZCopy cant change from Block blob to Page blob on the fly
- PROBLEM 2: “Existing manifest”, “incomplete operation” error in AZCopy
  - AZCopy cant be run in parallel under the same user profile
- Clever parallelisation no more ☹

LESSON 2

- Plan storage account tiers and number beforehand!

LESSON 3

- AZCopy is parallel by default, but cant be run in parallel itself!

Loading Data…
Still Going

Divide and Conquer Again!

- Create 10 windows users on Azure VM
- Run a group under each user in parallel
  - Separation of profile means no manifest error
- Wake up in the morning and kick off the other 10
- Maxxed out a G Series VM with 32 cores for about 10 hours

Loading Data...
Last Step

Finally, Load to SQL DW!

- Set up external location again
- Use CTAS to stage the data

```
CREATE TABLE [staging].[table_name]
WITH (DISTRIBUTION = …)
AS
SELECT … FROM external_table
```

One more Hiccup!

- Initial table load was pretty fast (CTAS)
- Subsequent loads used

```
INSERT INTO [staging].[table_name]
SELECT … FROM external_table
```

LESSON 4

- CTAS is minimally logged and fast. Use it wherever possible. Subsequent inserts are slower.

Where Does SQL DW Fit?

SQL Server VM (Iaas)	Azure SQL Database	Azure SQL Data Warehouse	Azure Data Lake
OLTP / DW workloads Lift and Shift Customer managed Shared features 1TB+	OLTP/ DW workloads Net new development Fully managed service No shared features 1Gb-1Tb GB	DW workloads only Fully managed Dynamic Pause/Scale 250GB – PB+	Non-relational Cheap, flexible Access Processing raw data 1 TB+

Some SQL DW Caveats

- No PK/FK relationships
  - A result of MPP architecture. Workaround – views
- Other unsupported constructs
  - E.g. Constraints, triggers
- Must manually create stats on columns
  - Don’t skip! Choose join/group columns. Regularly update.
- Cant fully use SSMS/SSDT fully at present
- Not all data types supported
  - Guide available online for conversion
- 32k buffer size for Polybase (will be increased)

Final Takeaways

- Have an initial architectural plan
- Consider each step of the data pipeline
  - Initial load could be very different from incremental updates
- Research limits/restrictions on ALL services
- Azure was very agile
  - Killed SQL DW and provisioned a new one
  - Provisioned additional storage as needed
  - Tested load on D series, scaled up VM to G Series when needed
- Got through it all in 2-3 days