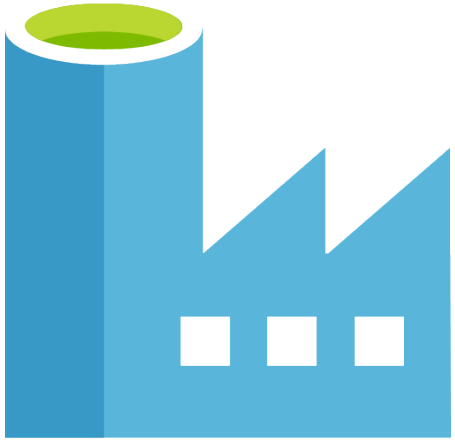
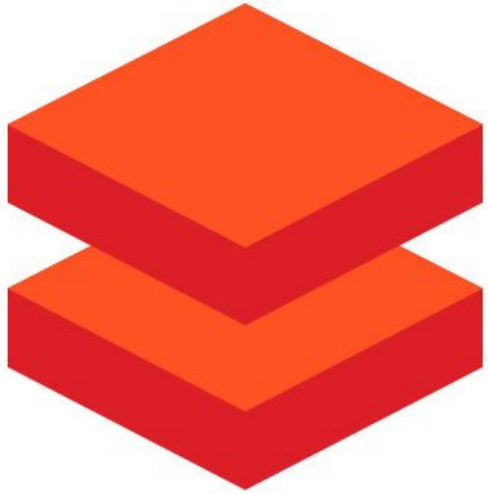
Data Factory Data Flow vs Databricks





Niall Langley

Data Developer / Consultant

11 years experience with SQL Server doing OLTP and data warehousing / BI, now working with Azure data platform

Blog: https://www.sqlsmarts.com

LinkedIn: https://uk.linkedin.com/in/niall-langley

Twitter: @NiallLangley

Introduction

- Introduction to Azure Data Factory and Databricks
- Background to Spark and Hadoop
- Demos
  - Transform data using ADF Mapping Data Flow
  - Transform data using a Databricks notebook
  - Run a Databricks notebook from ADF
- Comparison and conclusions

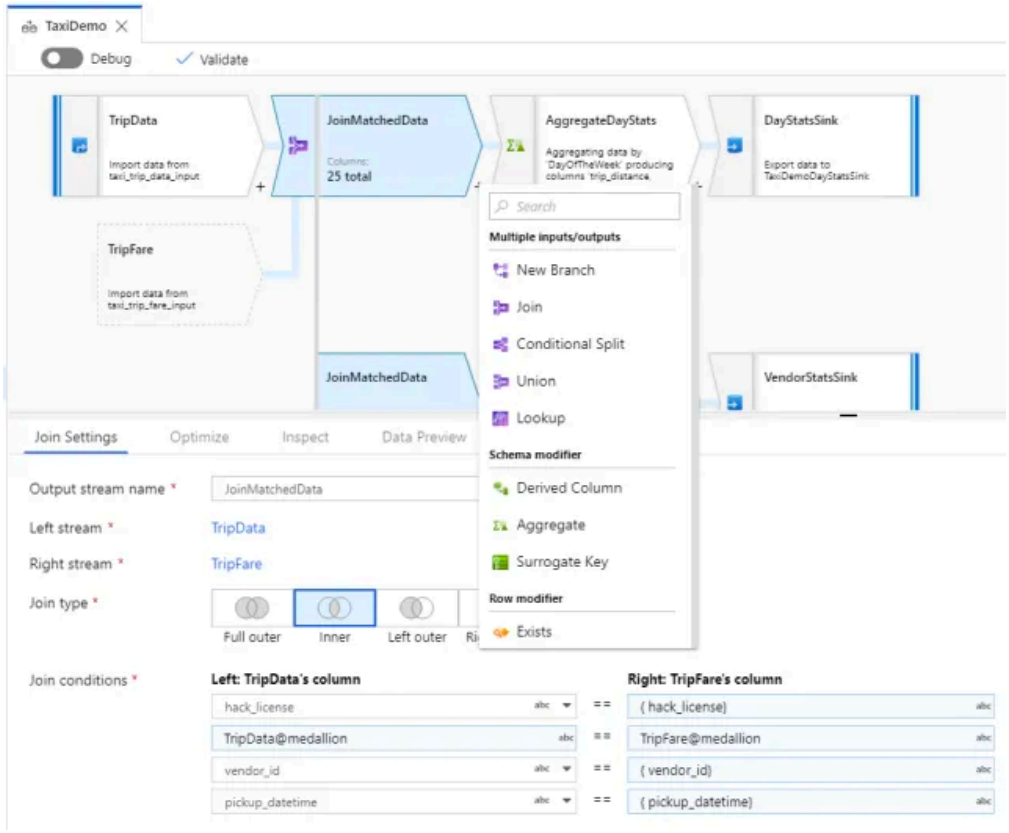
Azure Data Factory

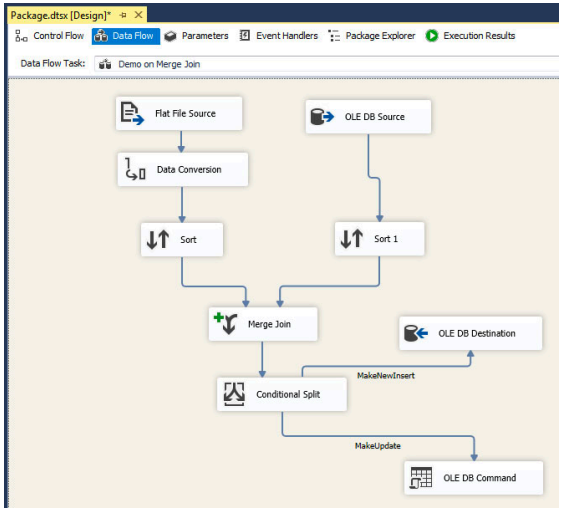
- Visual ELT tool to orchestrate the movement of data
- Data processing is done on an Integration Runtime
- We work with
  - Linked Services
  - Data Sets
  - Pipelines
  - Mapping Data Flows
  - Triggers
- Linked services and datasets can be dynamic

ADF Mapping Data Flow

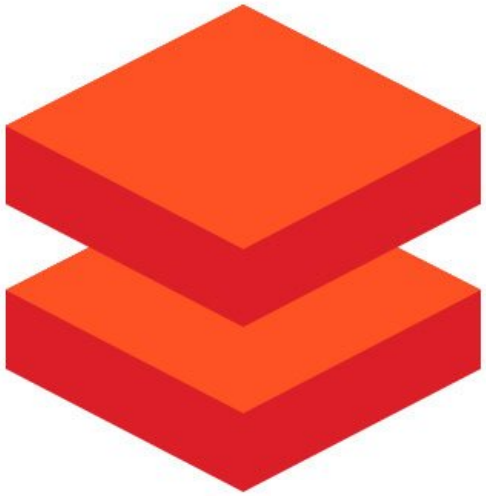
- Mapping Data Flows are like SSIS Data Flow Tasks
- Can build data transforms with no code
- Lots of tools to transform our data
- Limited set of source and sink types
- Cannot use dynamic linked services and datasets
- Runs on Databricks!

ADF Mapping Data Flow





Azure Databricks



- Databricks is a wrapper around Apache Spark to make it simpler to use
- It helps us manage our clusters
- Code can be written in Jupyter notebooks in Python, Scala, SQL and R
- DataFrames and DataSets on top of Spark RDD's
- Mature ecosystem of data processing and connectivity libraries

Hadoop and HDFS

- Started as two Google research papers, the Google File System (2003) and MapReduce (2004)
- Doug Cutting applied these concepts to search engine processing at Yahoo
- The first public release of Hadoop was in 2006
- Designed to use commodity hardware to process petabytes of data
- Takes the processing to the data
- Data structure is defined on retrieval
- Huge ecosystem built around Hadoop

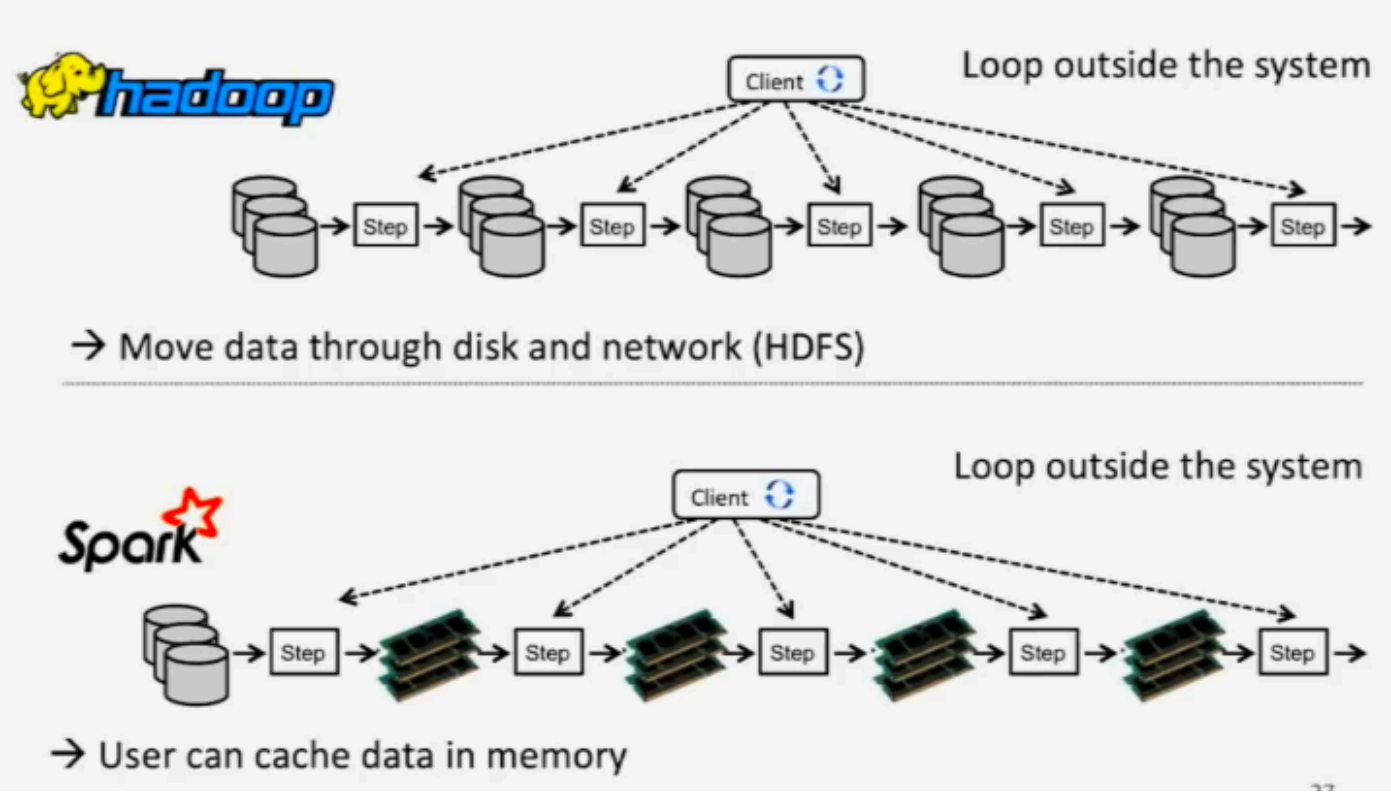
Spark Is Born

- Hadoop faced challenges
  - Lots of reads and writes to disks are slow
  - Hardware changed: SSD & NVMe, lots more RAM, the Cloud
- Spark started as a PHd project at UCL Berkley's AMPLab in 2009
  - In memory cluster computing
  - Improve Hadoops reliance on disk I/O
- First open source release in 2010 with a paper from Matei Zaharia
- A second paper released in 2012 on Resilient Distributed Datasets

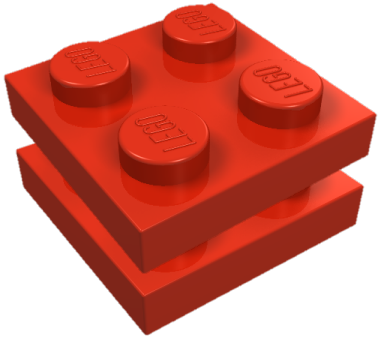
Spark

- Uses Resilient Distributed Datasets (RDD) as it's core concept
  - Immutable
  - Lazy
  - Distributed
  - Resilient
- Data is split into sets small enough to fit into the memory of the Executors doing the work
- If an Executor fails, the Driver sends the work to another executor
- Written in Scala, runs on the Java Virtual Machine
- PySpark for Python, SQL on the Catalyst Optimiser

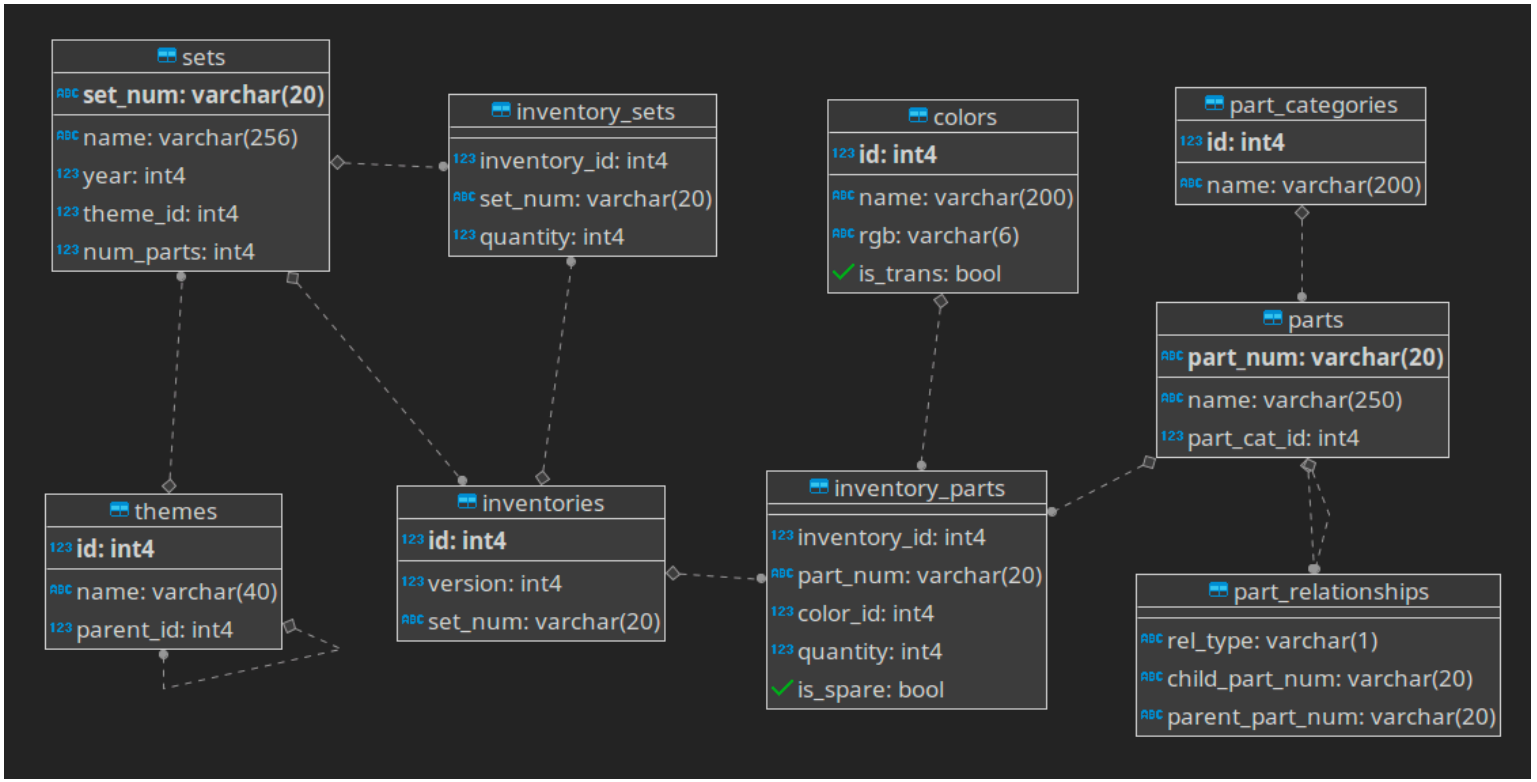
Hadoop vs Spark



DEMO



Demo Dataset Schema



Comparison

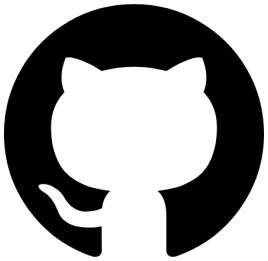
	ADF Mapping Data Flow	Databricks
Code free / GUI development	Yes	No
Data source / sink support	Limited	Lots
Documentation	Visual Tool	Rich Markdown in Notebooks
Extensibility	No	Yes
Bring-your-own Cluster	No	Yes
SQL Support	No	Yes
Learning Curve	Low / Medium	High

Conclusions

- Use ADF Mapping Data Flows…
  - To get the power of Spark without learning Scala or Python
  - Simple to understand for SSIS developers learning Azure
  - You don’t want to manage a Databricks cluster
- Use Databricks…
  - You want to write SQL queries to transform your code
  - You need to transform dynamic schemas
  - You prefer writing code to a GUI
  - You need extensibility beyond what is in Mapping Data Flows
  - You want to reuse existing business logic written in Java, Scala, Python or SQL

Resources

- https://github.com/NJLangley/Azure-Demos



- https://rebrickable.com/downloads/

?

sqlbits

31st March – 4th April 2020 | ExCel Centre, London, UK

Pub after the talks