Add some SPARK to your ETL Pipeline

ANDREW KARCHER

Agenda

- Traditional vs. Modern ETL
- What is Spark?
- What is Azure Databricks?
- Developing with Databricks

What is Traditional ETL?

- Vendors such as Talend, SQL Server Integration Services, Informatica, etc.
- Focused on periodic batch based processes
- Heavily dependent on fixed schemas
- Typically a Scale-Up to handle increased loads
- ETL = Extract, Transform, Load

What has changed?

- Business Intelligence -> Big Data
- Relational Databases -> Files, APIs
- Batch -> Streaming
- Applications -> Microservices
- Daily Updates -> Real Time

What is Spark?

Unified Analytics Engine

Developed at AmpLab at UC Berkeley in 2009

100% Open Source

Multi Language Support

Optimized for Scale Out Processing

Why Spark?

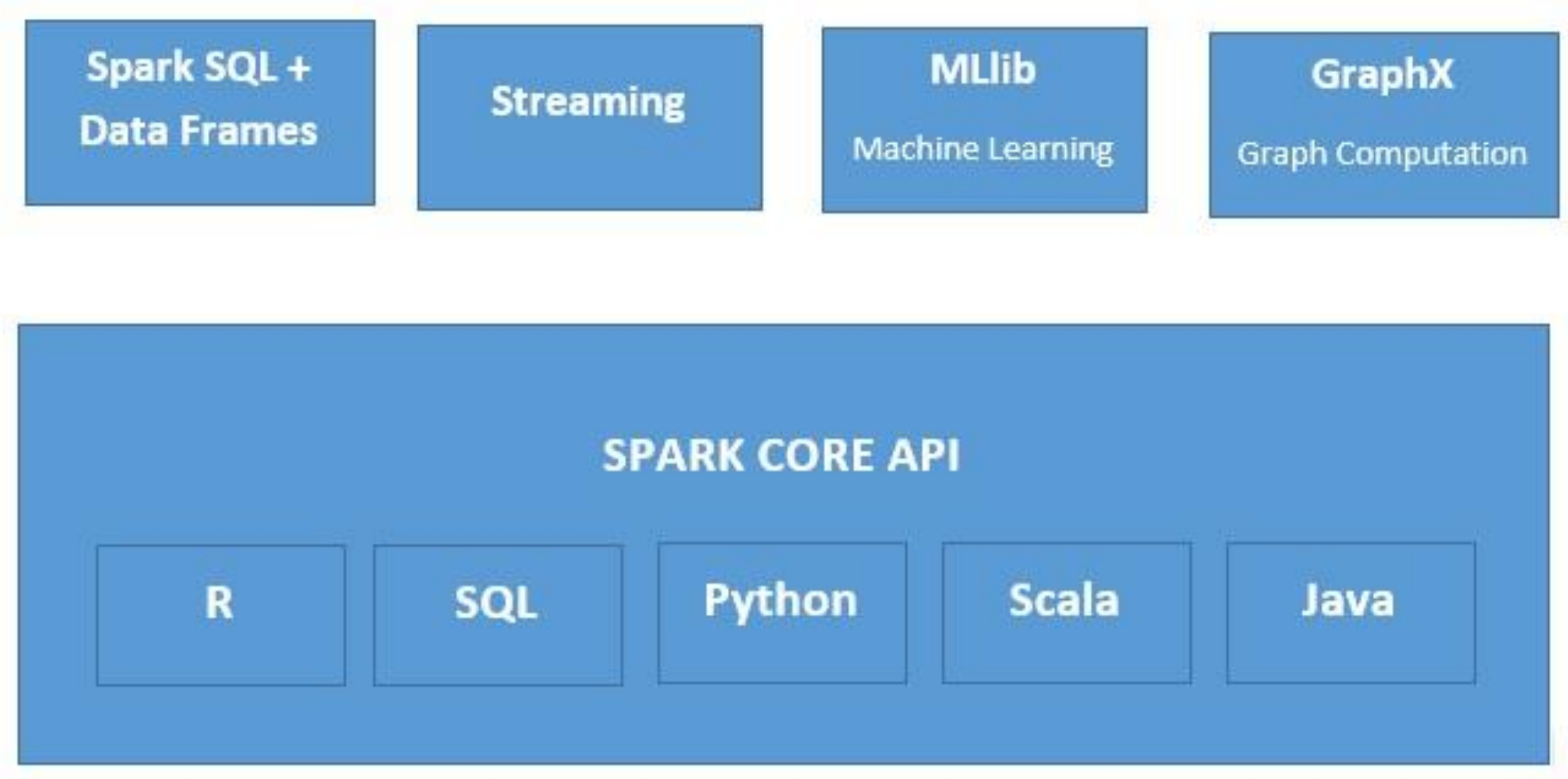
Need for better, faster processing than Hadoop

Industry shift to more text based storage

Need to have scale out data processing technology

Processing layer distinct from the Storage Layer

Spark Ecosystem



Spark SQL + Data Frames

Streaming

MLlib

Machine Learning

GraphX

Graph Computation

SPARK CORE API

R

SQL

Python

Scala

Java

What is Azure Databricks?

Azure Databricks is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud services platform. Designed with the founders of Apache Spark, Databricks is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.

Azure

databricks®

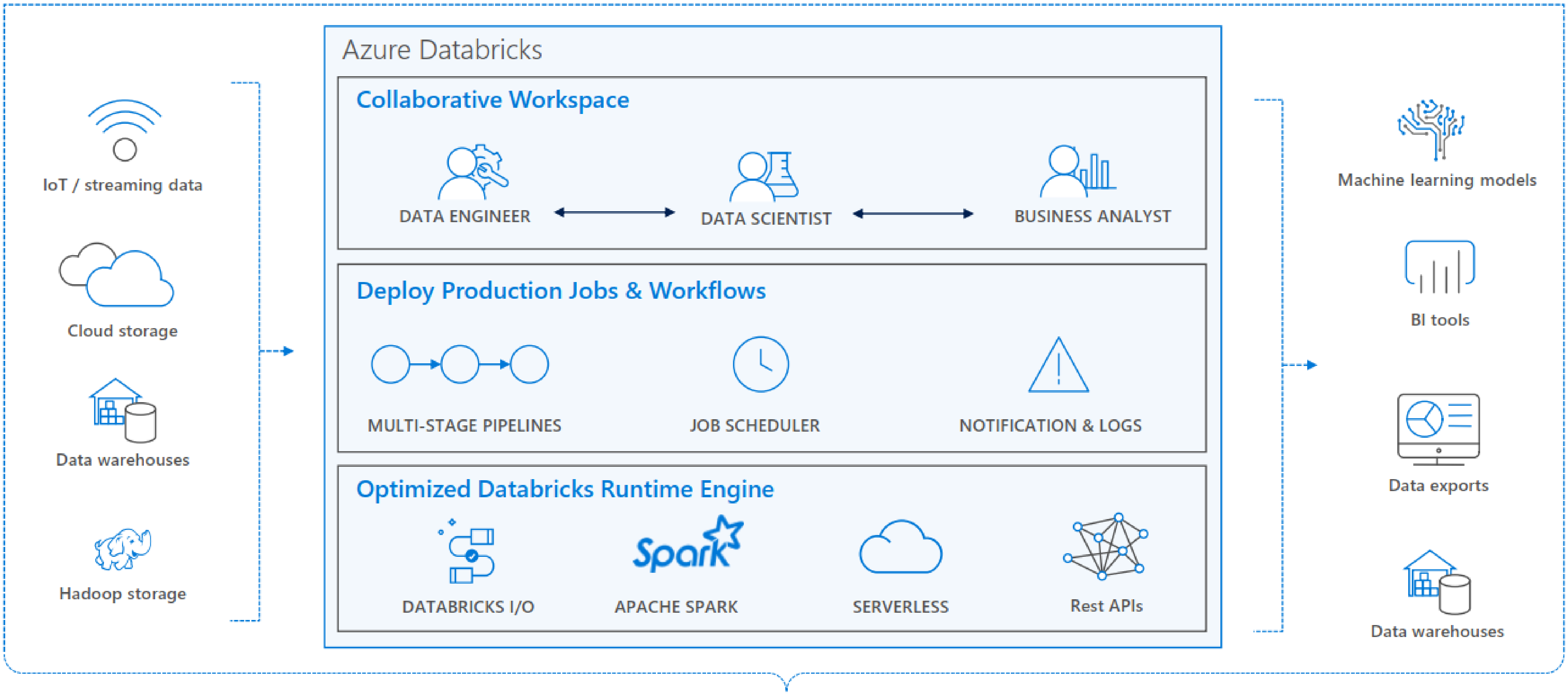
What is Databricks?

Founded in 2013 by the original developers of Spark

Works within the community to help support open source development

Developed Databricks Cloud Hosted Platform

Goal is to enable customers to be successful with Big Data



Enhance Productivity

Build on secure & trusted cloud

Scale without limits

Collaborative Workspace

Single Click to Launch Environment

Interactive Exploration with Notebooks

Collaborate with colleagues as well as integrate with Source Control

In-Notebook Visualizations

Deployment

- Jobs with Ability to Schedule
- Create Multi-Stage Pipelines with Programming Control Structures
- Turn Notebooks or JARs into Resilient Spark Jobs
- Alerts and Audit Logs
- Native Integration with other Azure Services

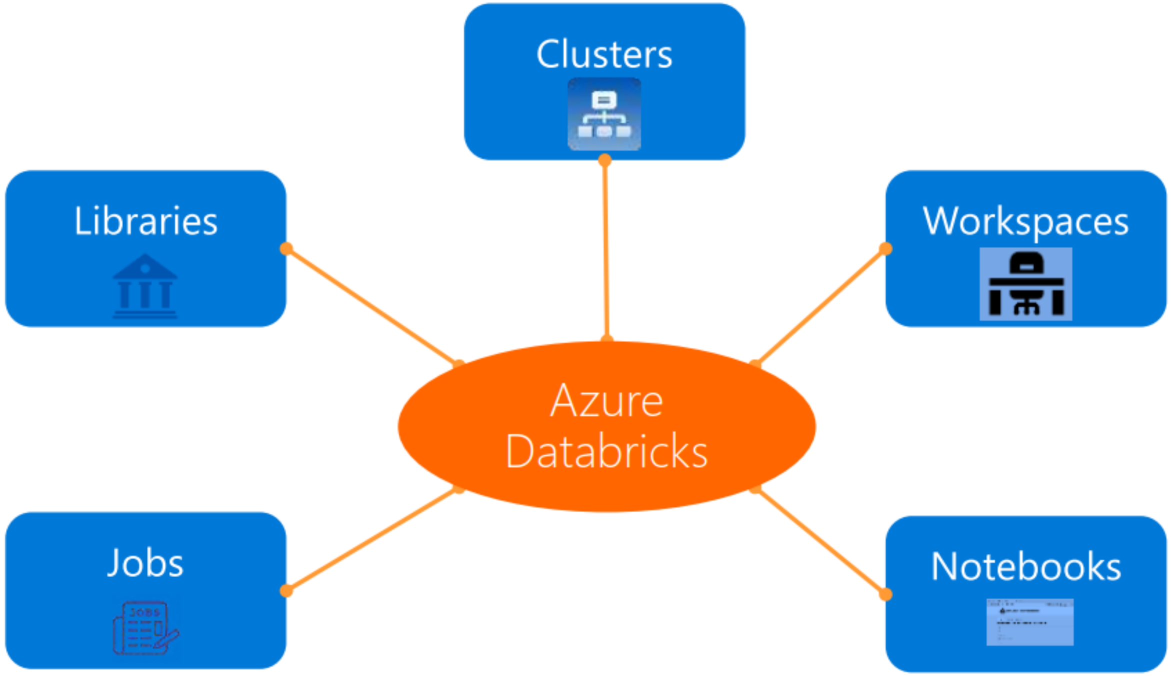
Runtime

Fully Managed Cloud Platform (Azure, AWS)

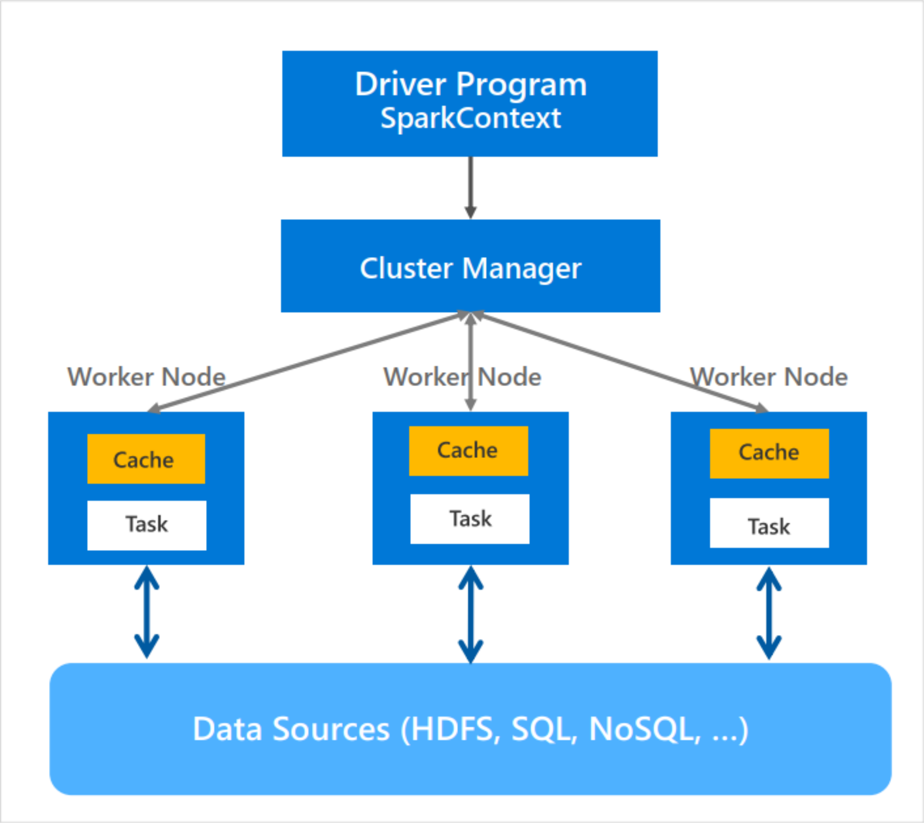
Serverless and Elastic Cloud

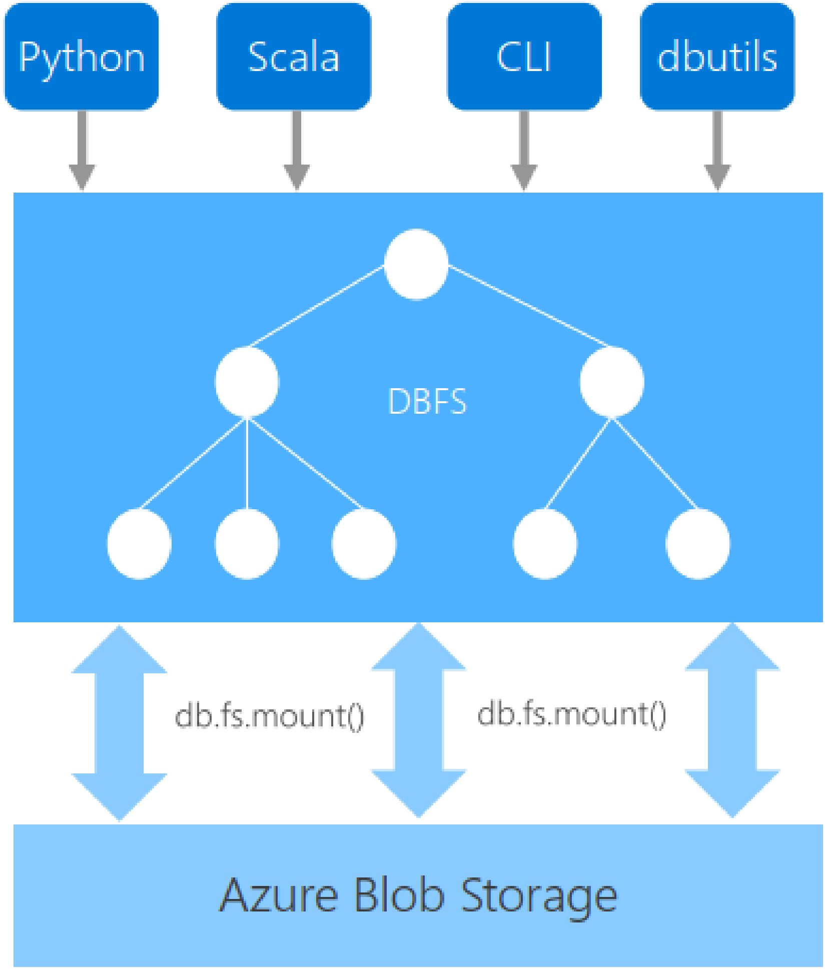
Able to operate and Scale to Massive Scale

Azure Core Artifacts



General Spark Cluster Architecture





Databricks File System

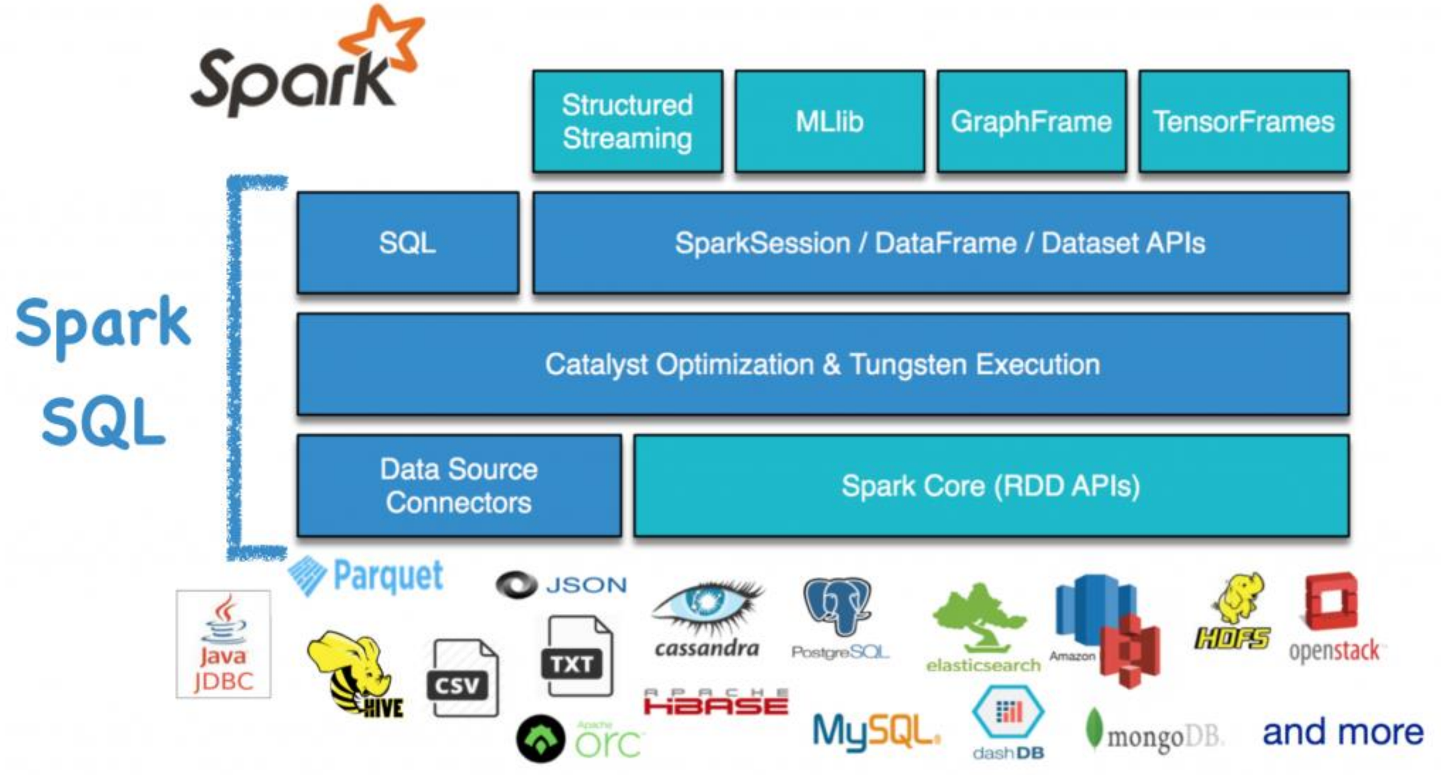
Azure Storage Buckets can be mounted in DBFS for all users to use without keys

Data is persisted in File Storage so data is not lost when cluster is removed

Pre-Installed on Databricks Clusters

Data can be cached on worker nodes on SSDs

Spark SQL Overview



Can Query data in a wide variety of data sources in one query language

Can be queried using SQL or HiveQL

Bindings in Python, Scala, and Java

Built-In Support for Structured Streaming

Spark Databases and Tables

- Databases are a collection of related tables
- Tables are defined using the GUI or programmatically using APIs or Notebooks
- Utilizes the Hive Metastore to manage tables
- Like a Dataframe, any Spark operation can be applied to Tables (Filtering, Caching, etc.)
- Spark SQL is able to dynamically generate partitions at the file storage level

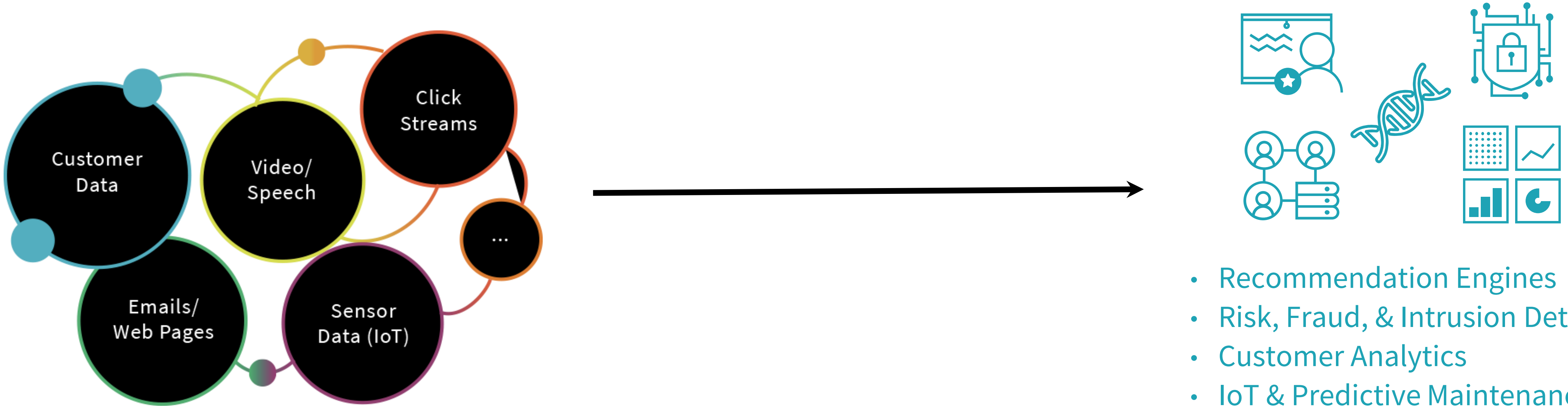
Spark Demo

Databricks Delta

AKA “DELTA LAKE”

Data Lakes - A Key Enabler of Analytics

Data Science and ML



- Recommendation Engines
- Risk, Fraud, & Intrusion Detection
- Customer Analytics
- IoT & Predictive Maintenance
- Genomics & DNA Sequencing

Data Reliability Challenges

- ✕ Failed production jobs leave data in corrupt state requiring tedious recovery
- ⚠ Lack of schema enforcement creates inconsistent and low quality data
- Lack of consistency makes it almost impossible to mix appends ands reads, batch and streaming

Performance Challenges

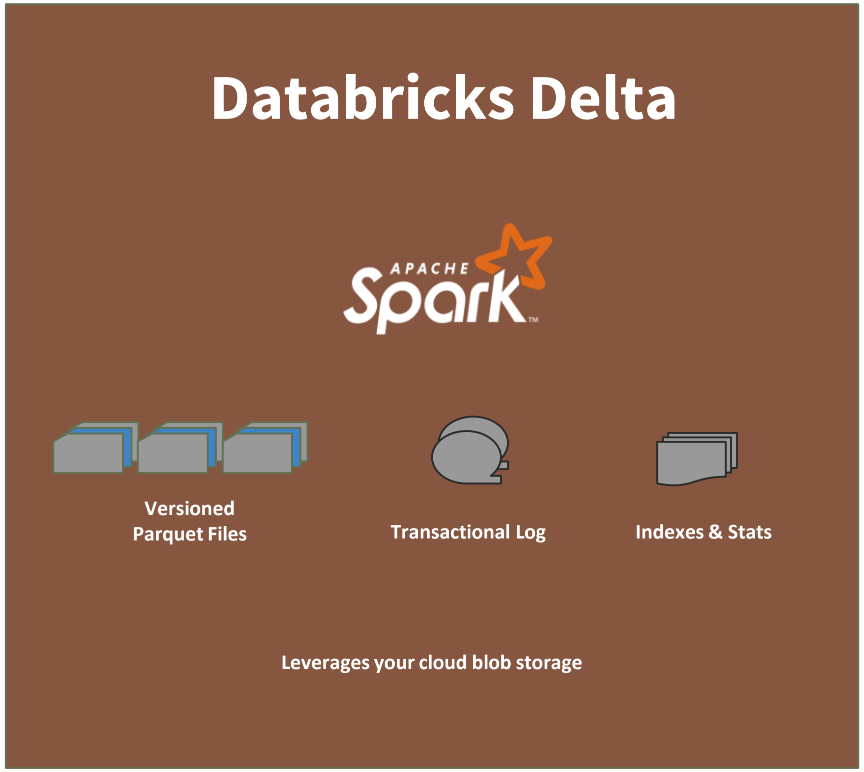
Too many small or very big files - more time opening & closing files rather than reading contents (worse with streaming)

Partitioning aka “poor man’s indexing”- breaks down if you picked the wrong fields or when data has many dimensions, high cardinality columns

No caching - cloud storage throughput is low (S3 is 20-50MB/s/core vs 300MB/s/core for local SSDs)

Databricks Delta

Next-generation engine built on top of Spark



Databricks Delta

Versioned Parquet Files

Transactional Log

Indexes & Stats

Leverages your cloud blob storage

Co-designed compute & storage

Compatible with Spark API’s

Built on open standards (Parquet)

Features of Databricks Delta

ACID TRANSACTIONS

SCHEMA ENFORCEMENT

UPSERTS

DATA VERSIONING

Databricks Delta Performance

- Compaction
- Caching
- Data Skipping
- Z-Order Indexes
- Optimized Parquet

Demo: Delta and Streaming

Conclusion: Should you add Spark to your ETL Pipeline? (Pros)

As you move from a Batch based pipeline to Streaming it is worth a consideration

If you want to tap into ML Pipelines it is definitely something to consider

I don’t think you need to migrate all your existing ETL pipelines to Spark

If you are doing your data work in the cloud then it is definitely something to consider

Conclusion: Should you add Spark to your ETL Pipelines? (Cons)

Do you have the skills in-house? If not, are you willing to invest in training your existing people or hiring new ones?

Is there a business reason that supports investing in a new technology?

Is your data in the cloud or on-premise? (Can you manage a Spark Cluster internally?)

About Me

http://www.andrewkarcher.com

Twitter: @akarcher

LinkedIn, Twitter

Email: akarcher@gmail.com

Company – Data Architect, Becton Dickinson (BD)

Community Work

- Former Organizer for SQL Saturday OC
- President San Diego .Net User Group
- Organizer – Data Engineering San Diego
- Former SQL Server MVP

Questions??

AKARCHER@GMAIL.COM

@AKARCHER

HTTP://WWW.ANDREWKARCHER.COM