Spark (h)and(s)-on Data Bricks

Salvo Nicotra

Data Bricks

https://www.databricks.com

2023 - a Data Lakehouse Platform

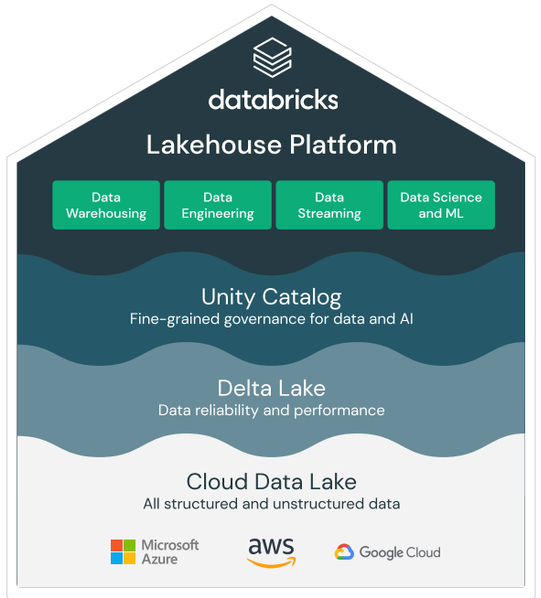
https://www.databricks.com/blog/impact-data-and-ai-modern-business

Simple. Open. Multicloud.

The Databricks Lakehouse Platform combines the best elements of **data lakes** and **data warehouses** to deliver the reliability, strong governance and performance of data warehouses with the openness, flexibility and machine learning support of data lakes.

This unified approach simplifies your *modern data stack* by eliminating the data silos that traditionally separate and complicate data engineering, analytics, BI, data science and machine learning.

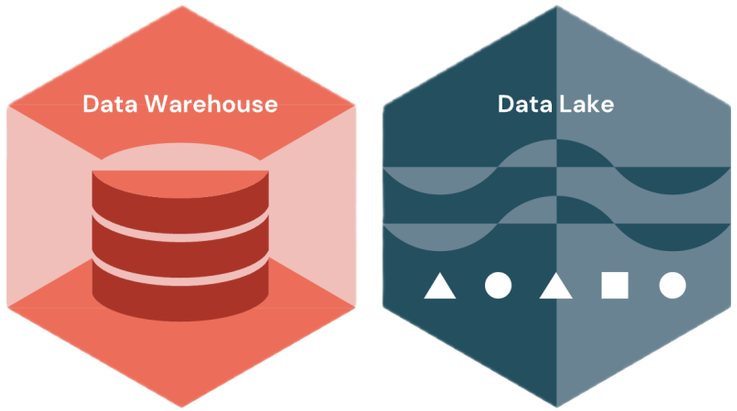
It’s built on open source and open standards to maximize flexibility. And, its common approach to data management, security and governance helps you operate more efficiently and innovate faster



Simple

The unified approach simplifies your data architecture by eliminating the data silos that traditionally separate analytics, BI, data science and machine learning.

With a lakehouse, you can eliminate the **complexity** and **expense** that make it hard to achieve the full potential of your analytics and AI initiatives.

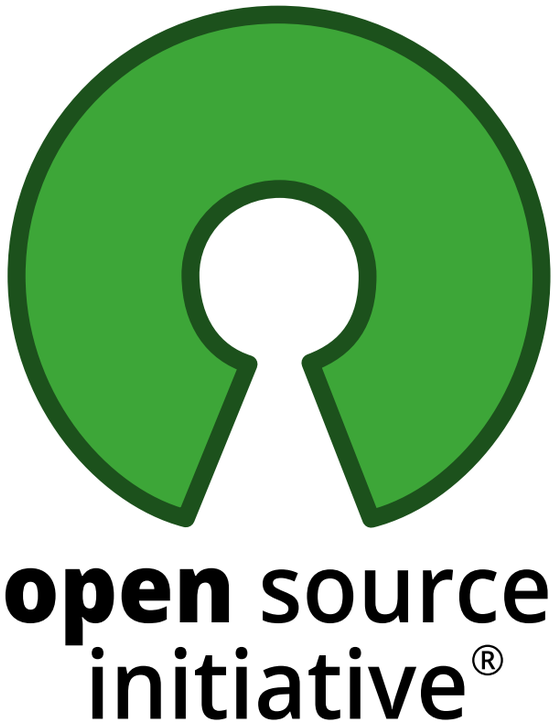


Open

https://www.databricks.com/product/open-source

Delta Lake forms the open foundation of the lakehouse by providing reliability and world-record-setting performance directly on data in the data lake.

You're able to avoid proprietary walled gardens, easily share data, and build your modern data stack with unrestricted access to the ecosystem of open source data projects and the broad Databricks partner network.

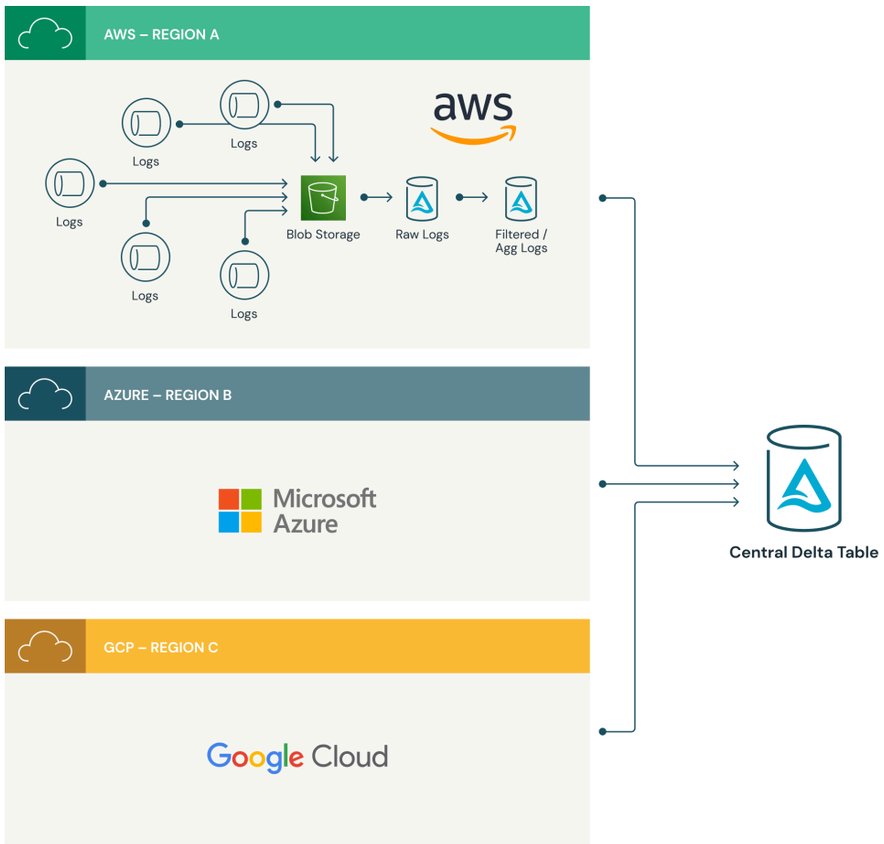


Multicloud

https://www.databricks.com/blog/2021/07/14/petabyte-scale-data-processing-across-multiple-cloud-platforms.html

The Databricks Lakehouse Platform offers you a consistent management, security, and governance experience across all clouds.

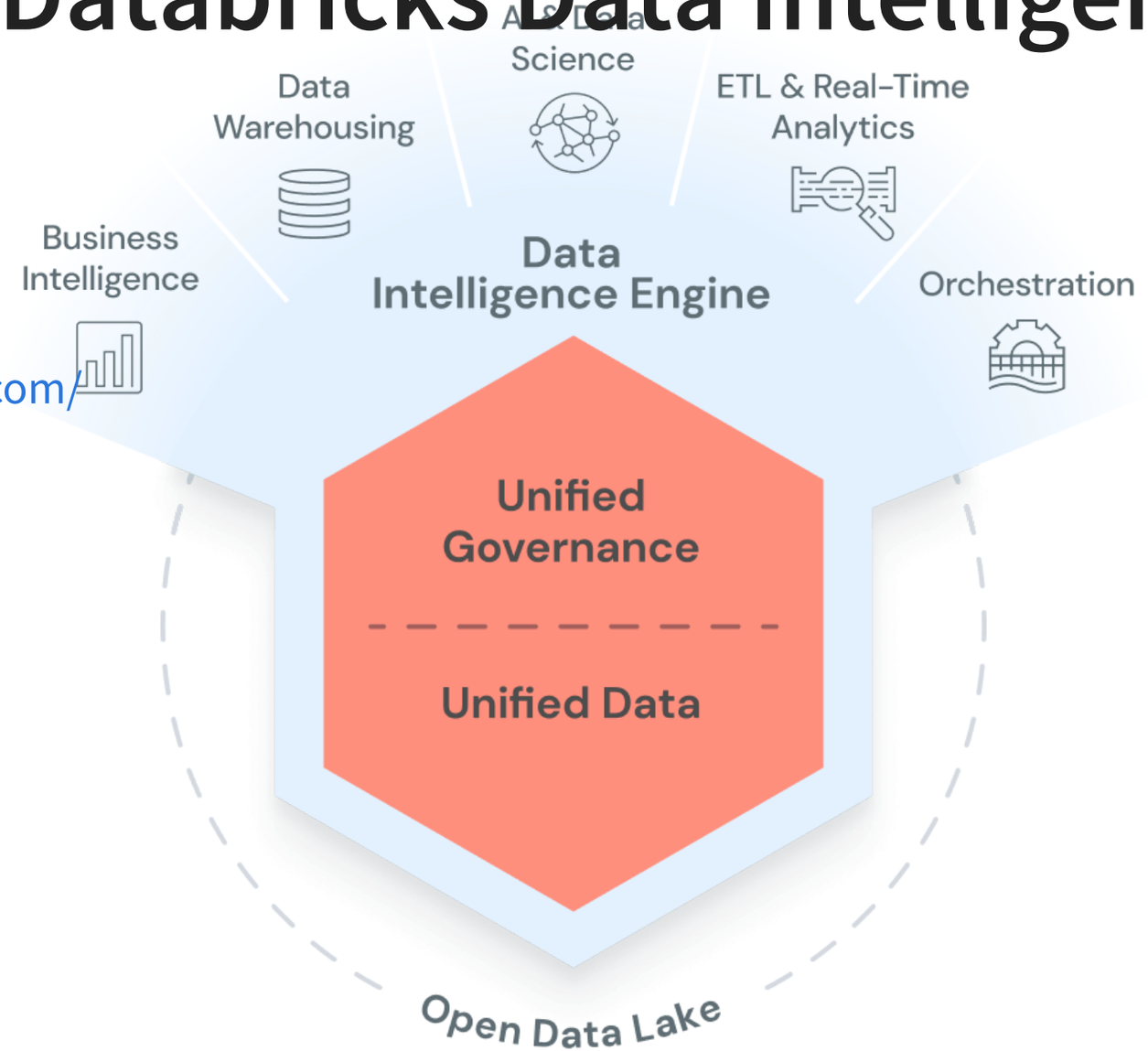
You don’t need to invest in reinventing processes for every cloud platform that you’re using to support your data and AI efforts. Instead, your data teams can simply focus on putting all your data work to discover new insights.



2024 - The Databricks Data Intelligence Platform

> Data and AI for all

https://www.databricks.com/



Community Edition

https://community.cloud.databricks.com/login.html

Create Account

- Sign Up for a new account
- Select “Get started with Community Edition” instead of selecting a Cloud Provider
- Verify the email

Run Tutorial

- Click on Guide: Quickstart tutorial
- Create a Cluster clicking on Connect
- Run Cells in order

Apache Spark: Unified engine for large-scale data analytics

https://spark.apache.org

Apache Spark™ is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters

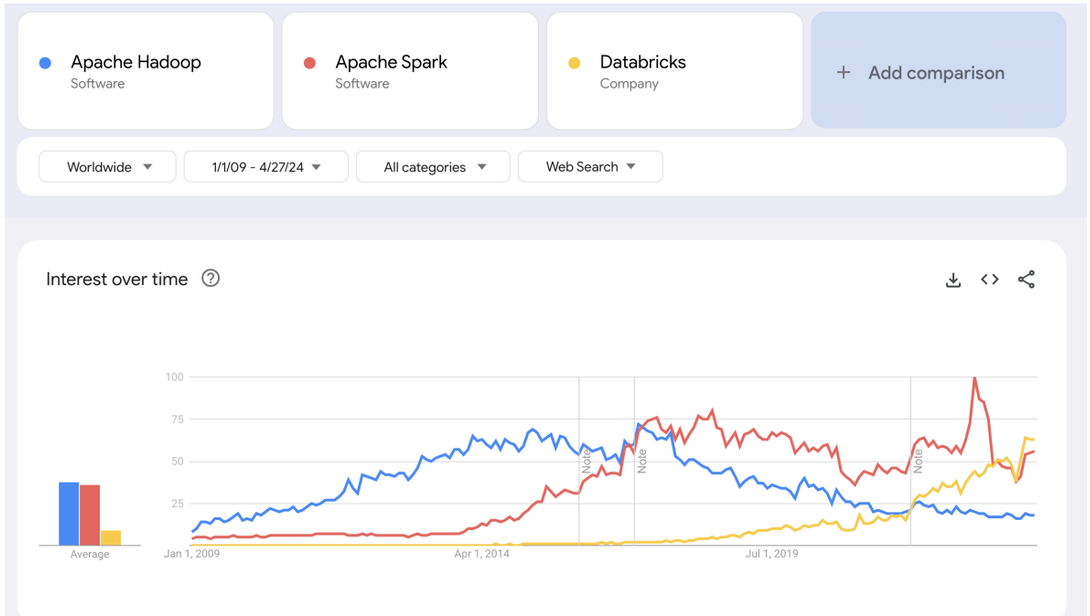
The Apache Spark project's History

> Spark was originally written by the founders of Databricks during their time at UC Berkeley. The Spark project started in 2009, was open sourced in 2010, and in 2013 its code was donated to Apache, becoming Apache Spark. The employees of Databricks have written over 75% of the code in Apache Spark and have contributed more than 10 times more code than any other organization. Apache Spark is a sophisticated distributed computation framework for executing code in parallel across many different machines. While the abstractions and interfaces are simple, managing clusters of computers and ensuring production-level stability is not. Databricks makes big data simple by providing Apache Spark as a hosted solution.

A Gentle Introduction to Apache Spark on Databricks

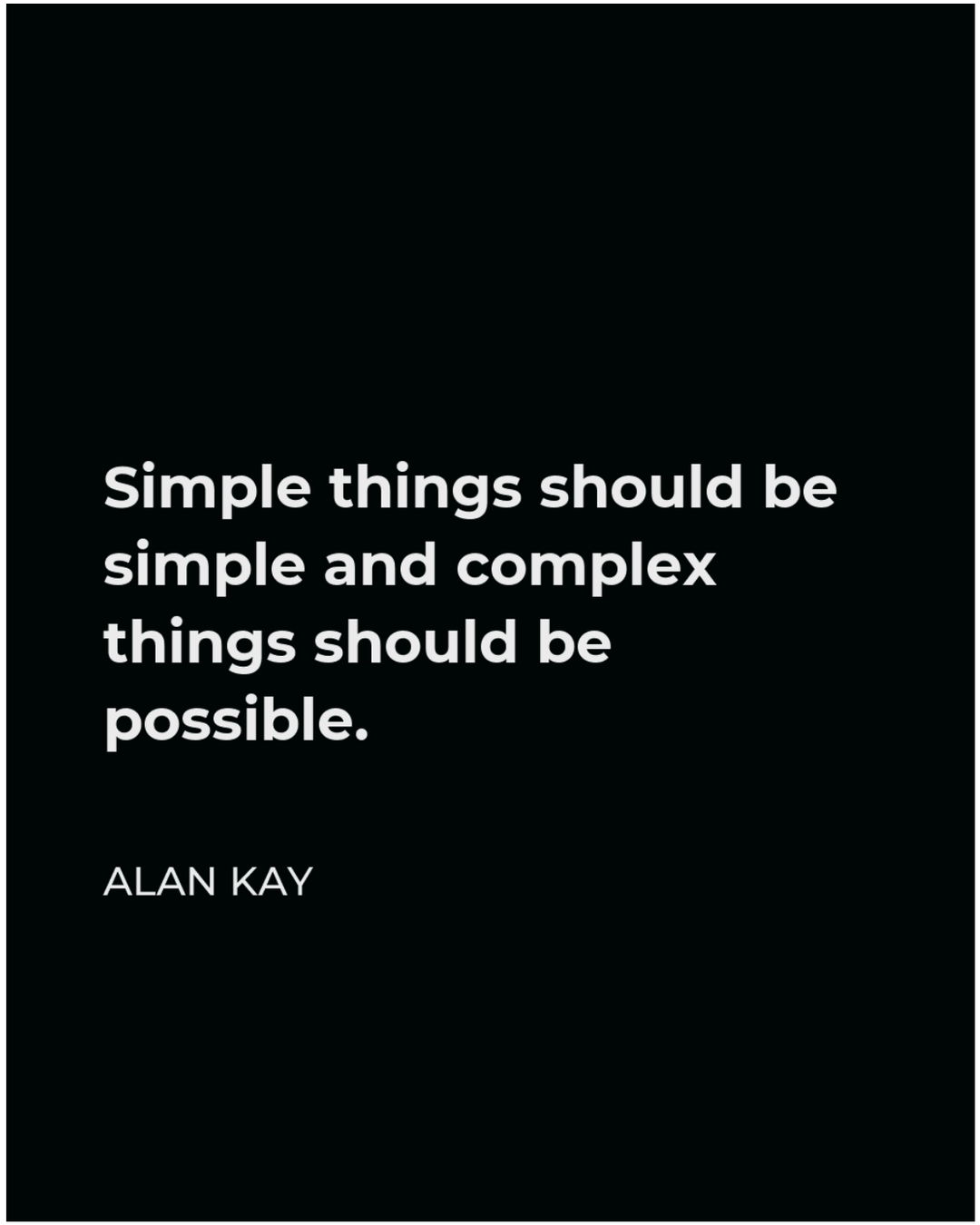
Spark Trends

The Spark project started in 2009, was open sourced in 2010, and in 2013 its code was donated to Apache, becoming Apache Spark.



The Genesis of Spark: from Hadoop

- Big Data and Distributed Computing at Google (2004)
- Hadoop at Yahoo! (2006)
- The question then became: there a way to make Hadoop and MR simpler and faster?



Spark 1.0 and beyond

- **2009** – Spark's early development begins at UC Berkeley's AMPLab
- **2010** – First research paper published: Spark shown to be 10–20x faster than MapReduce
- **2014** – Spark 1.0 released: marks the first major production-ready version
- **2016** – Spark 2.0: Introduces DataFrame and Dataset unification, Structured Streaming
- **2020** – Spark 3.0: Adds Hadoop 3.0 support, Pandas UDFs, and a faster SQL engine
- **2023** – Spark 3.4: Introduces Spark Connect, enabling remote client connectivity
- **2024** – Spark 3.5: Spark Connect goes GA; adds distributed training with DeepSpeed
- **2025** – Spark 3.5.5: Maintenance release with key bug fixes, timestamp enhancements, Kubernetes Dockerfile fixes, and REST API hardening

Design Philosophy

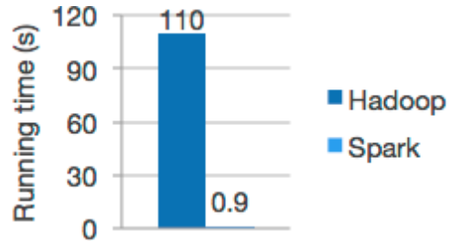
https://www.oreilly.com/library/view/learning-spark-2nd/9781492050032/ch01.html

Spark's design philosophy centers around four key characteristics:

- Speed
- Ease of use
- Modularity
- Extensibility

1. Speed

Run workloads 100x faster



Logistic Regression

Apache Spark achieves

- high performance for both batch and streaming data
- using a state-of-the-art DAG scheduler
- a query optimizer
- a physical execution engine.

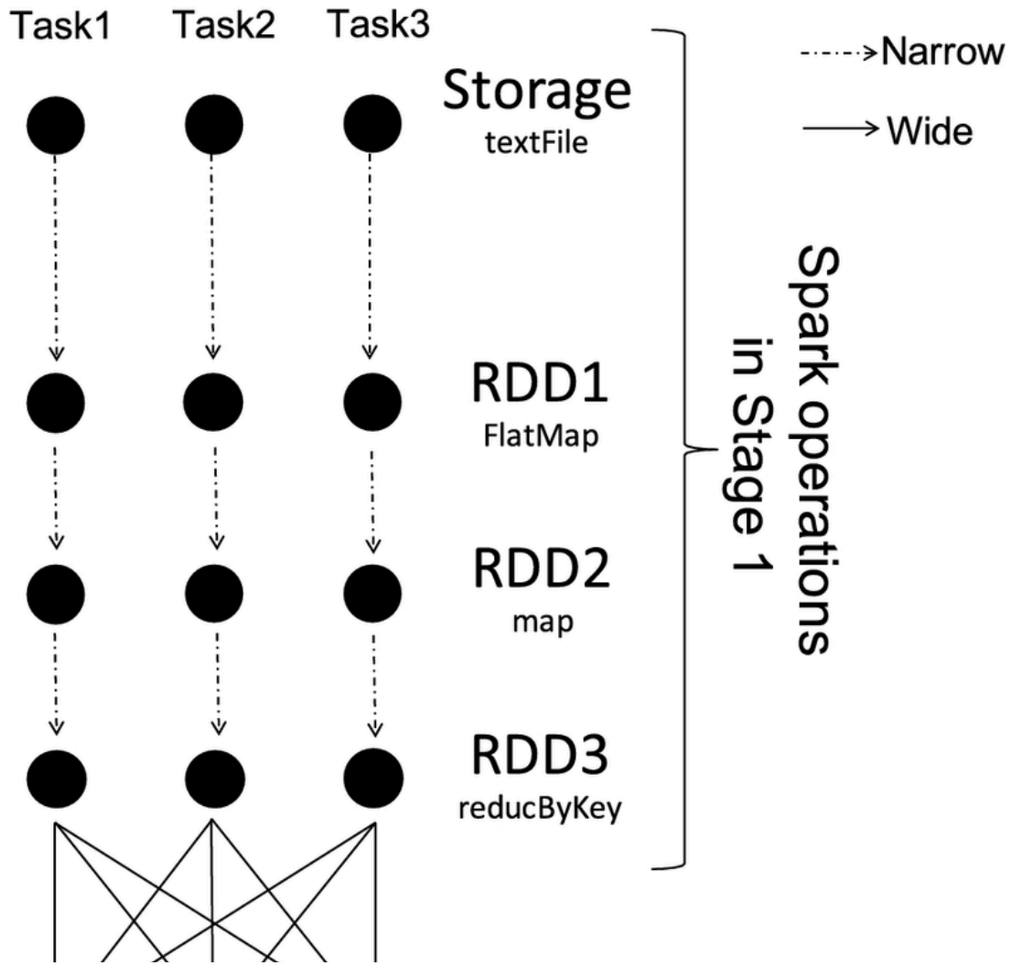
Why Spark is faster ?

1. Hardware improvements: Today's commodity servers come cheap, with hundreds of gigabytes of memory, multiple cores, and the underlying Unix-based operating system taking advantage of efficient multithreading and parallel processing.

RAM

SCRAP

2. Direct Acyclic Graph (DAG) Scheduler and Query Optimizer: Provides an efficient computational graph that can usually be decomposed into tasks that are executed in parallel across workers on the cluster



2. Ease of Use

Key factor that influences the success and adoption of a platform - Ref For Spark

- Unified API: Available in Python, Scala, Java, and R, making it accessible to diverse developers.
- Interactive Shells: Real-time data exploration and debugging in Python (PySpark) and Scala.
- Familiar Data Structures: DataFrame and SQL support simplify data manipulation.
- Built-in Libraries: Includes MLlib for machine learning, Spark Streaming, and GraphX, reducing external dependencies.
- Lazy Evaluation: Optimizes workflows by delaying computation until necessary.
- In-Memory Processing: Speeds up iterative tasks, ideal for data exploration and machine learning.

3. Modularity

- Write applications quickly in Java, Scala, Python, R, and SQL.
- Spark offers over 80 high-level operators that make it easy to build parallel apps.
- And you can use it **interactively** from the Scala, Python, R, and SQL shells.

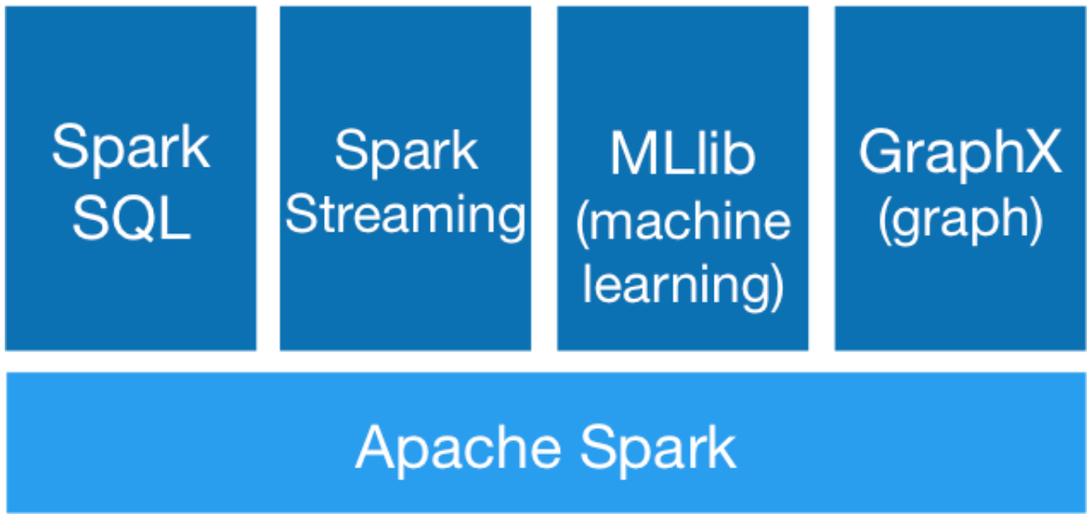
Scala Example

```
df = spark.read.json("logs.json")
df.where("age > 21").select("name.first").show()
```

4 Generality Combine SQL, streaming, and complex analytics.

Spark powers a stack of libraries including:

- **Spark SQL** module for working with structured data
- **Spark Streaming** build streaming applications and pipelines
- **MLlib** scalable machine learning library
- **GraphX** API for graphs and graph-parallel computation New:
- **Pandas API**: Use pandas syntax on Spark
- **Spark Connect**: Client application that communicate with remote Spark server



Runs everywhere

https://www.corriere.it/foto-gallery/esteri/14_ottobre_07/nuovo-attrezzo-fare-sport-ruota-criceti-misura-d-uomo-809cb22a-4e22-11e4-b38c-5070a4632162.shtml

Spark runs on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud. You can run Spark using its standalone cluster mode, on EC2, on Hadoop YARN, on Mesos, or on Kubernetes

It can access diverse external data sources

Analyse

Spark can create distributed datasets from any storage source supported by Hadoop, including your local file system, HDFS, Cassandra, HBase, Amazon S3, etc. Spark supports text files, SequenceFiles, and any other Hadoop InputFormat.

https://spark.apache.org/docs/latest/rdd-programming-guide.html#external-datasets

Query

Spark SQL supports operating on a variety of data sources through the DataFrame interface. A DataFrame can be operated on using relational transformations and can also be used to create a temporary view. Registering a DataFrame as a temporary view allows you to run SQL queries over its data.

https://spark.apache.org/docs/latest/sql-data-sources.html

In short

- Apache Spark is a fast and general-purpose cluster computing system.
- It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs.
- Supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Spark Streaming.

Spark (2024)

Batch/streaming data

Unify the processing of your data in batches and real-time streaming, using your preferred language: Python, SQL, Scala, Java or R.

SQL analytics

Execute fast, distributed ANSI SQL queries for dashboarding and ad-hoc reporting. Runs faster than most data warehouses.

Data science at scale

Perform Exploratory Data Analysis (EDA) on petabyte-scale data without having to resort to downsampling

Machine learning

Train machine learning algorithms on a laptop and use the same code to scale to fault-tolerant clusters of thousands of machines.

Examples

https://spark.apache.org/examples.html

Compute Spark PI

https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/1408031979081866/151400444:

Word Count

https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/1408031979081866/117735199(

Data Frame Text Search

https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/1408031979081866/7042140928

Core Data Abstractions in Spark

- Spark provides several key abstractions:
  RDDs, **DataFrames**, **Datasets**, and **SQL Tables**
- These all represent **distributed collections of data**
  that operate efficiently across clusters.

Immutability in Spark

- Spark data structures are **immutable**
  → Once created, they **cannot be changed** directly.
- So how do you “modify” data?
  You don’t change it — you **describe how to transform it**.

Resilient Distribuited Datasets (RDD)

https://spark.apache.org/docs/latest/rdd-programming-guide.html

https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/1408031979081866/145571796

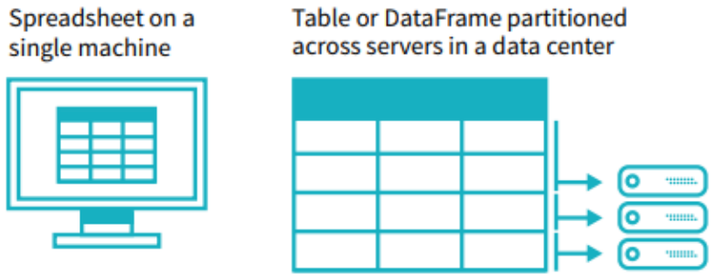
Spark SQL, DataFrames and Datasets

https://spark.apache.org/docs/latest/sql-getting-started.html

What is a DataFrame?

https://www.databricks.com/glossary/what-are-dataframes

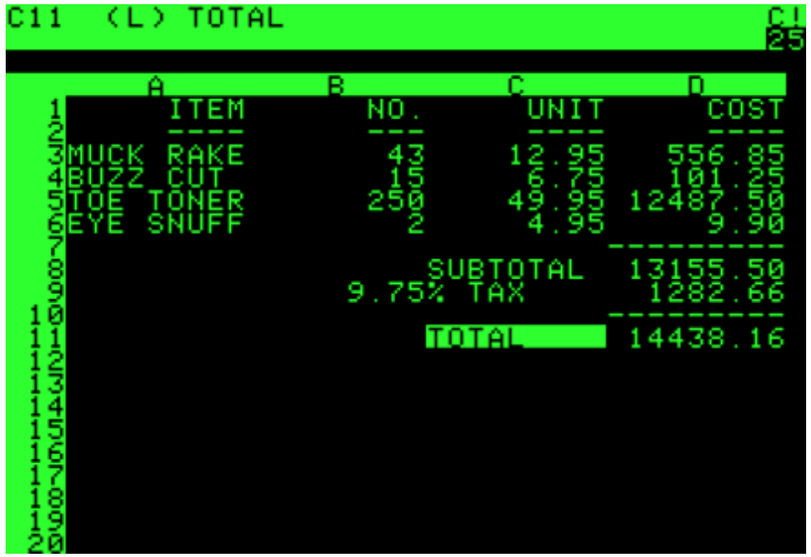
- A DataFrame is a data structure that organizes data into a 2-dimensional table of rows and columns, much like a spreadsheet. DataFrames are one of the most common data structures used in modern data analytics because they are a flexible and intuitive way of storing and working with data.
- Every DataFrame contains a blueprint, known as a schema, that defines the name and data type of each column. Spark DataFrames can contain universal data types like StringType and IntegerType, as well as data types that are specific to Spark, such as StructType. Missing or incomplete values are stored as null values in the DataFrame.



Dataframe representation

Analogy with Spreadsheet

A simple analogy is that a DataFrame is like a spreadsheet with named columns. However, the difference between them is that while a spreadsheet sits on one computer in one specific location, a DataFrame can span thousands of computers. In this way, DataFrames make it possible to do analytics on big data, using distributed computing clusters.



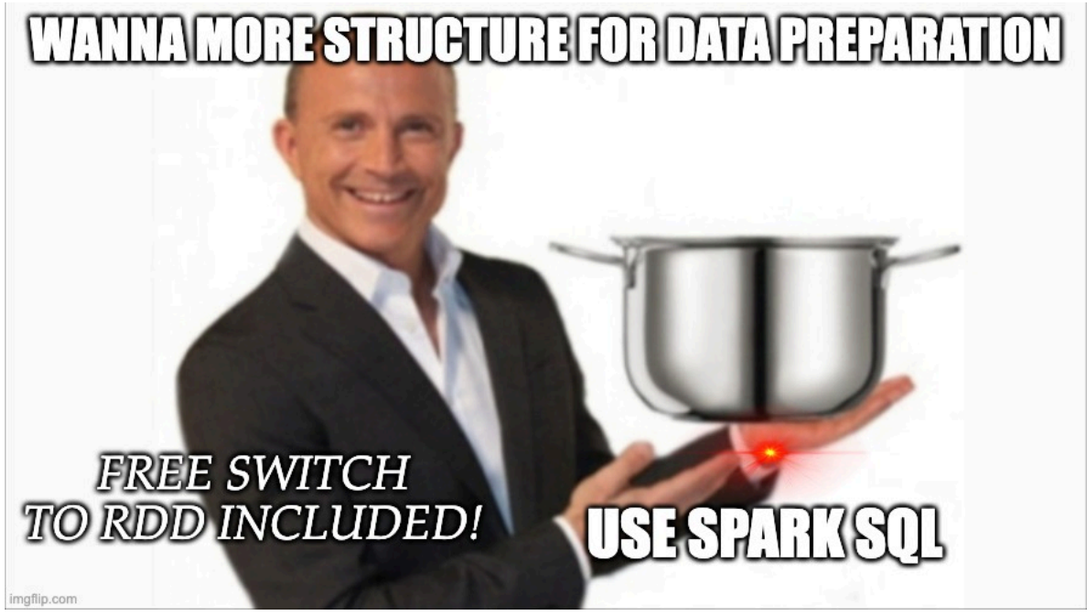
Originally

RDD was the primary user-facing API in Spark since its inception.

At the core, an RDD is:

- an immutable distributed collection of elements of your data,
- partitioned across nodes in your cluster that can be operated in parallel
- with a low-level API that offers transformations and actions.

Is RDD enough ?



NicsMeme

Spark SQL

- Spark SQL is a Spark module for **structured data processing**.
- Unlike the basic Spark RDD API, the interfaces provided by Spark SQL provide Spark with more information about the structure of both the data and the computation being performed. Internally, Spark SQL uses this extra information to perform extra optimizations.
- There are several ways to interact with Spark SQL including **SQL** and the **Dataset API**.
- When computing a result, the same execution engine is used, independent of which API/language you are using to express the computation. This unification means that developers can easily switch back and forth between different APIs based on which provides the most natural way to express a given transformation.

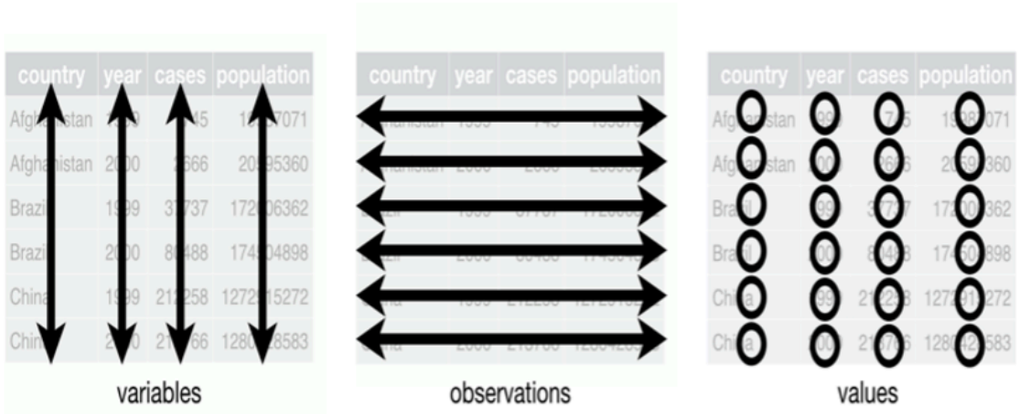
Structured data

Tidy Dataset

> “Happy families are all alike; every unhappy family is unhappy in its own way.” –– Leo Tolstoy

> “Tidy datasets are all alike, but every messy dataset is messy in its own way.” –– Hadley Wickham

https://uta.pressbooks.pub/datanotebook/chapter/1-3-structured-data/



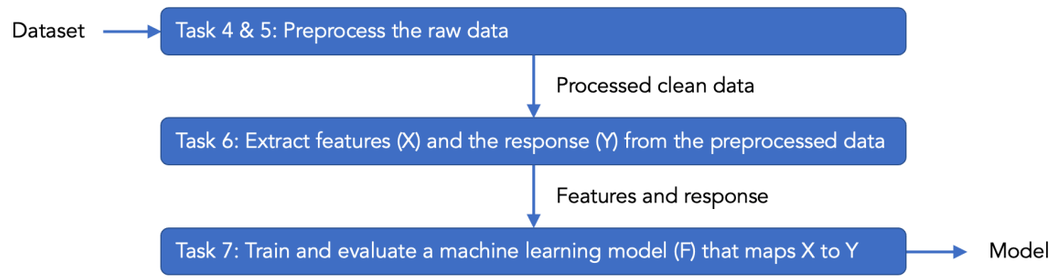
What is structured data?

https://aws.amazon.com/what-is/structured-data/

- Structured data is data that has a standardized format for efficient access by software and humans alike.
- It is typically tabular with rows and columns that clearly define data attributes.
- Computers can effectively process structured data for insights due to its quantitative nature.

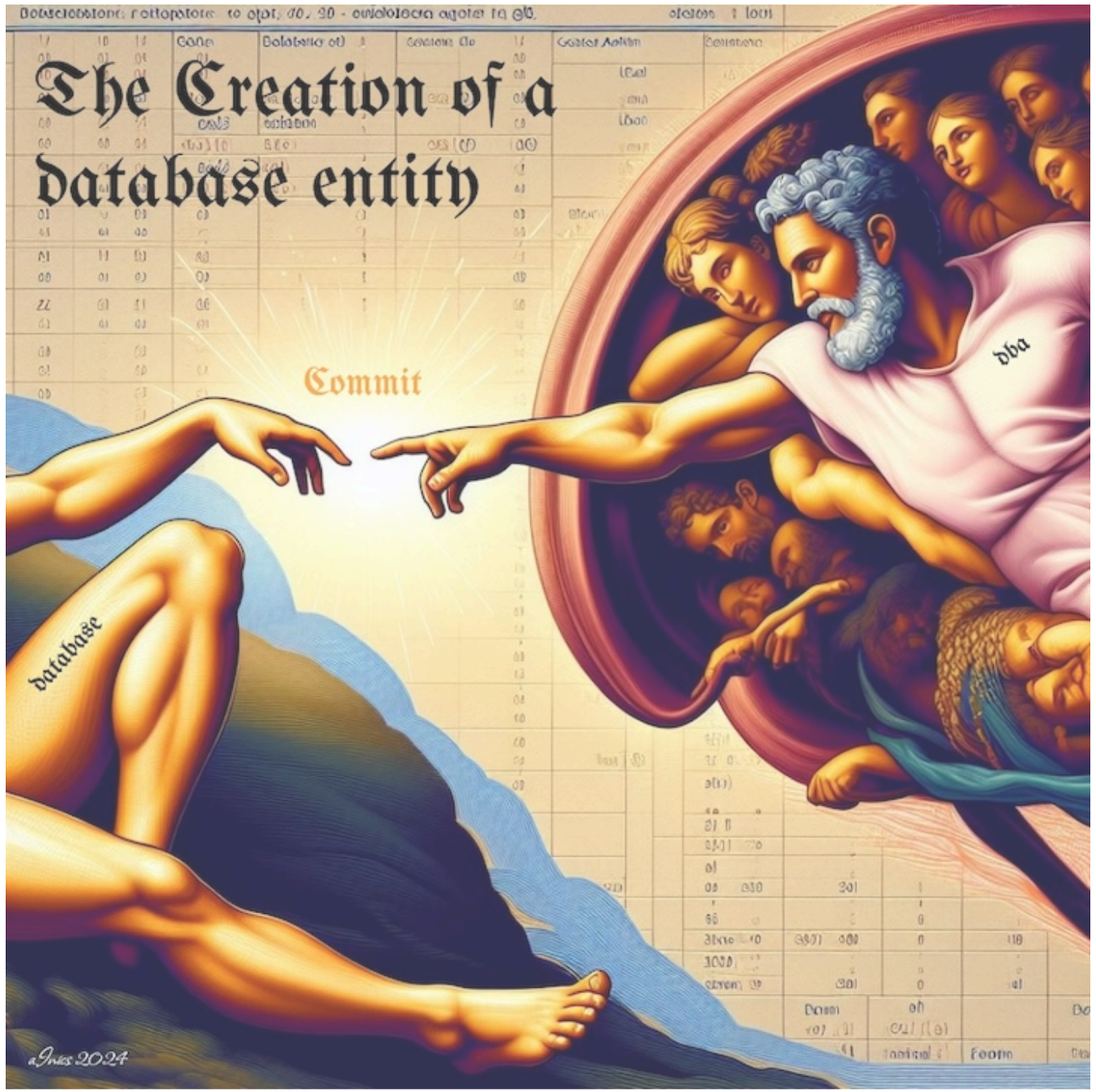
Structured data in Data Science

- Goal 1: Connect steps in the structured data processing pipeline to a real-world case.
- Goal 2: Preprocess structured data and prepare features/labels for modeling using pandas.
- Goal 3: Understand how Principal Component Analysis can help explore data.
- Goal 4: Understand how cross-validation works for time-series data.
- Goal 5: Have a general understanding of Decision Tree and Random Forest.
- Goal 6: Understand the concept of permutation feature importance.
- Goal 7: Experiment with different feature sets and reflect on the choice of features.



https://multix.io/structured-data-module/docs/overview-structured-data.html

SQL


The Creation of a database entity
Commit
dba
database
aJones 2024

Spark SQL

- One use of Spark SQL is to execute SQL queries.
- Spark SQL can also be used to read data from an existing Hive installation.
- When running SQL from within another programming language the results will be returned as a Dataset/DataFrame.
- You can also interact with the SQL interface using the command-line or over JDBC/ODBC.

Dataset

- A Dataset is a distributed collection of data.
- Dataset is a new interface added in Spark 1.6 that provides the benefits of RDDs (strong typing, ability to use powerful lambda functions) with the benefits of Spark SQL's optimized execution engine.
- A Dataset can be constructed from JVM objects and then manipulated using functional transformations (map, flatMap, filter, etc.).
- The Dataset API is available in Scala and Java. Python does not have the support for the Dataset API. But due to Python's dynamic nature, many of the benefits of the Dataset API are already available (i.e. you can access the field of a row by name naturally row.columnName). The case for R is similar

Data Frame

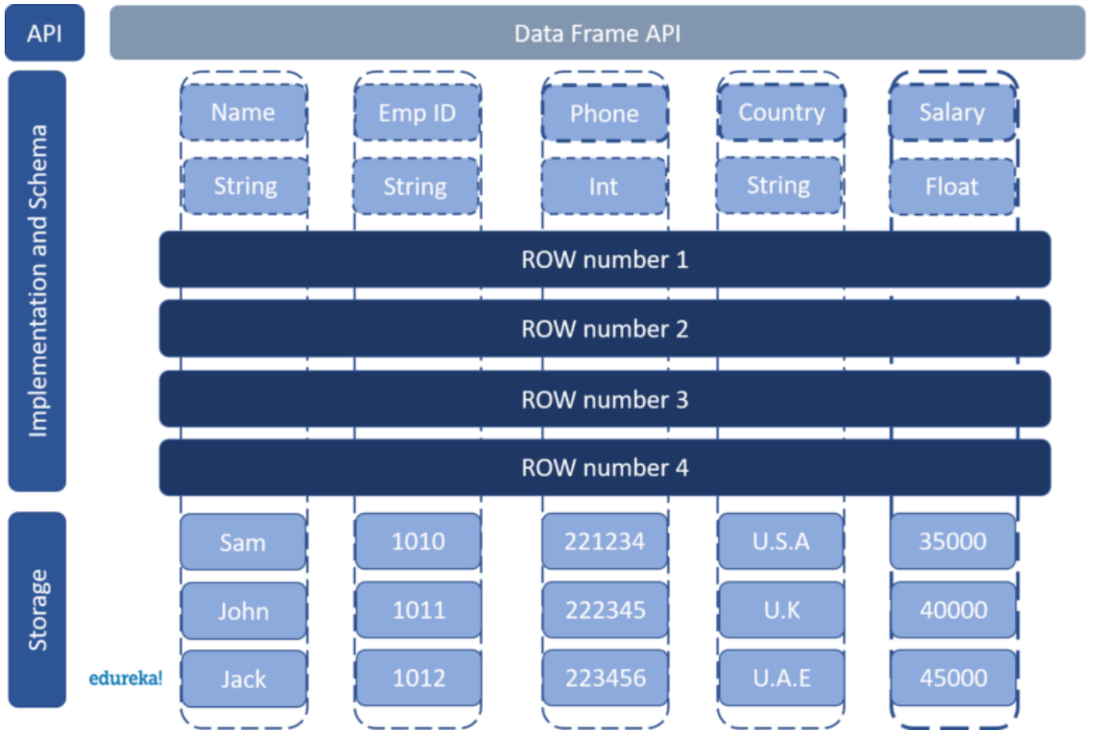
- A DataFrame is a Dataset organized into named columns.
- It is conceptually equivalent to a table in a relational database or a data frame in R/Python, but with richer optimizations under the hood.
- DataFrames can be constructed from a wide array of sources such as: structured data files, tables in Hive, external databases, or existing RDDs.
- The DataFrame API is available in Scala, Java, Python, and R. In Scala and Java, a DataFrame is represented by a Dataset of Rows.

Yet another definitions of DataFrames

https://www.databricks.com/glossary/what-are-dataframes

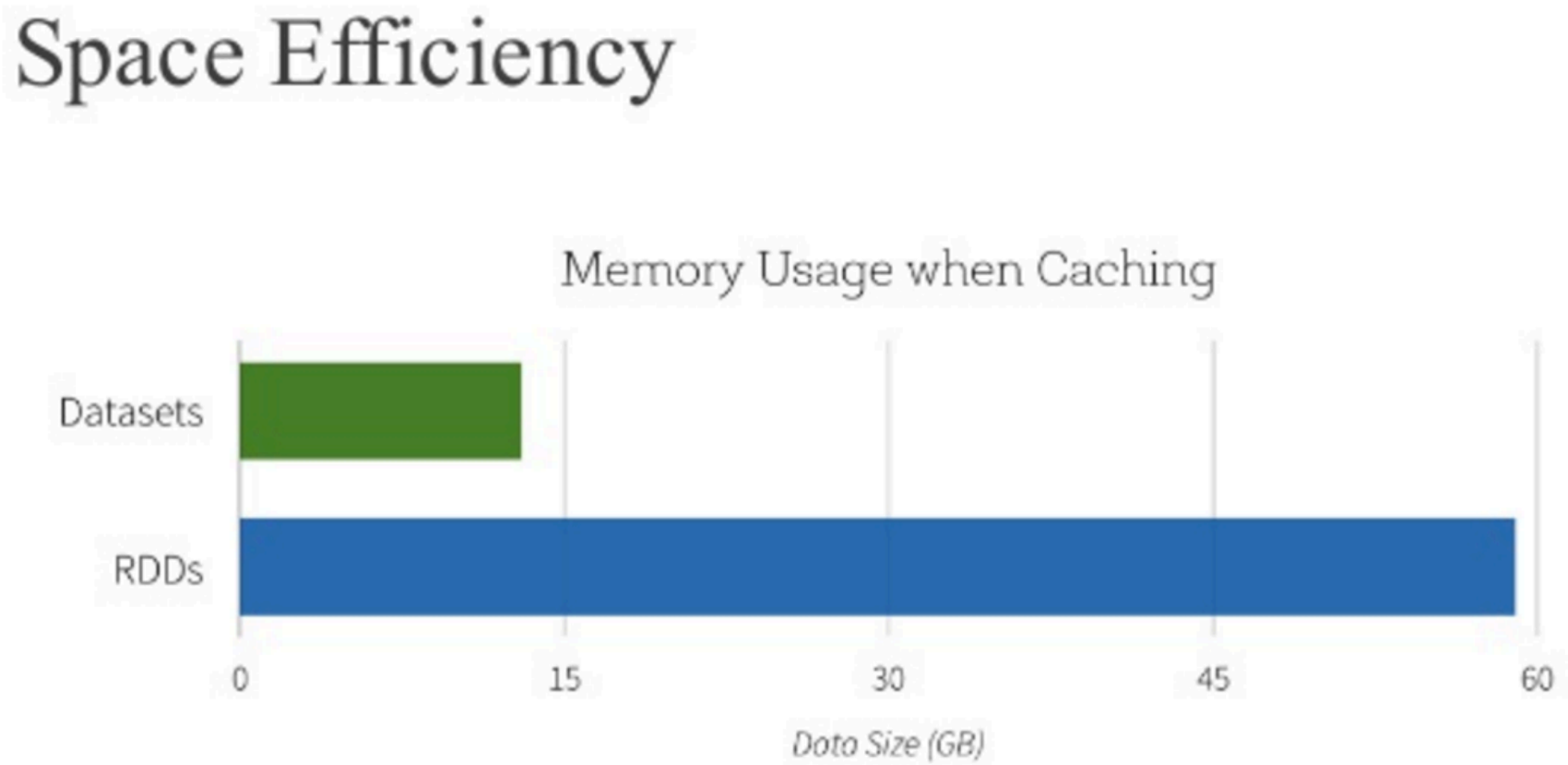
The concept of a DataFrame is common across many different languages and frameworks. DataFrames are the main data type used in pandas, the popular Python data analysis library, and DataFrames are also used in R, Scala, and other languages.

- Every DataFrame contains a blueprint, known as a schema, that defines the name and data type of each column.
- Spark DataFrames can contain universal data types like StringType and IntegerType, as well as data types that are specific to Spark, such as StructType.
- Missing or incomplete values are stored as null values in the DataFrame.

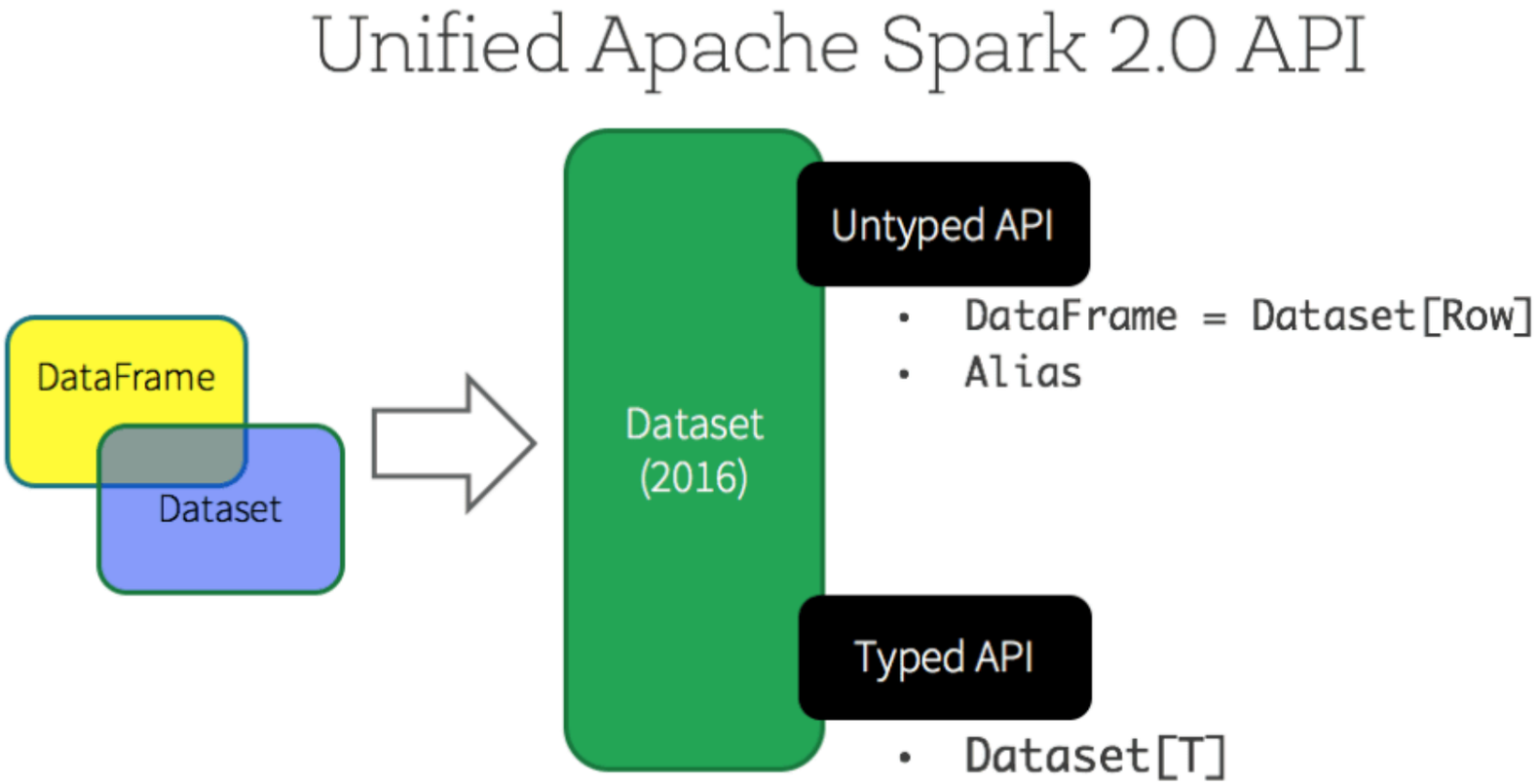


https://www.edureka.co/blog/dataframes-in-spark/

Dataframe efficency



Evolution


Unified Apache Spark 2.0 API
DataFrame
Dataset
Dataset
(2016)
Untyped API
• DataFrame = Dataset[Row]
• Alias
Typed API
• Dataset[T]

Data Frame Examples

EDA

Flights