Adnan Boota
Irene López

24 May, 2018

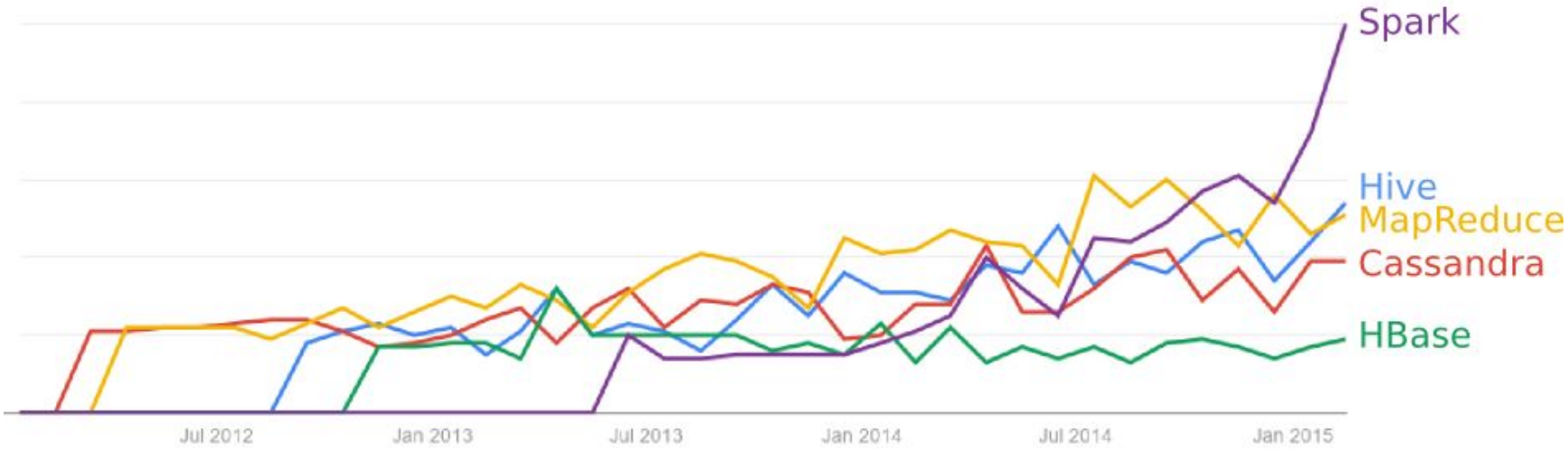
What is Spark?

- Cluster computing platform
- Fast and general purpose
- Objective: Become a unified engine for distributed data processing, making it easy and accessible
- Provides high-level APIs in Java, Scala, Python and R
- Programming model similar to MapReduce but extended with a data-sharing abstraction

What is Spark?

- 2009 - Developed at the AMPLab of UC, Berkeley
- 2013 - Taken on by the Apache Software Foundation
- Corporate backers: Databricks, IBM, Yahoo!, Intel and Huawei, among others.
- Most active platform on Big Data
- Can run as standalone or as part of a cluster.
- Official support for several popular cluster managers.

Popularity and adoption

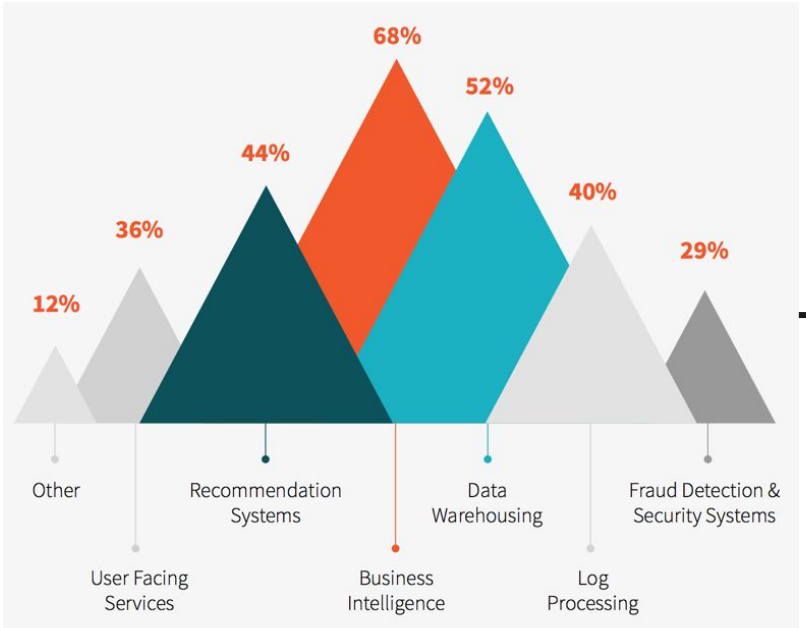


Selected Big Data activity on Google Trends

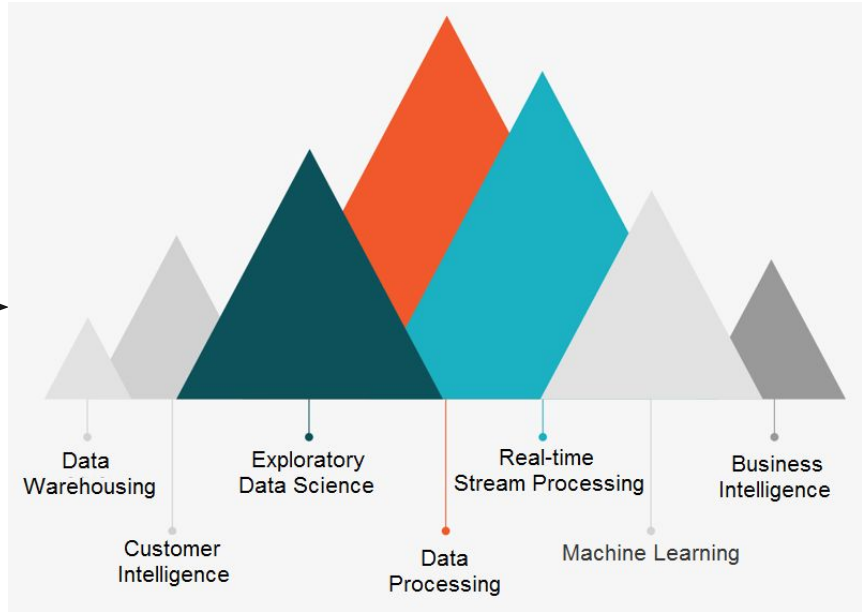
Advantages

- Improved performance for most tasks
- Lower entry barrier due to language support, abstractions and modularity.
- Efficient and flexible enough to combine different types of processing tasks.
- Promotes the addition of new features to existing applications due to the use of a unified API.

Popular Use Cases

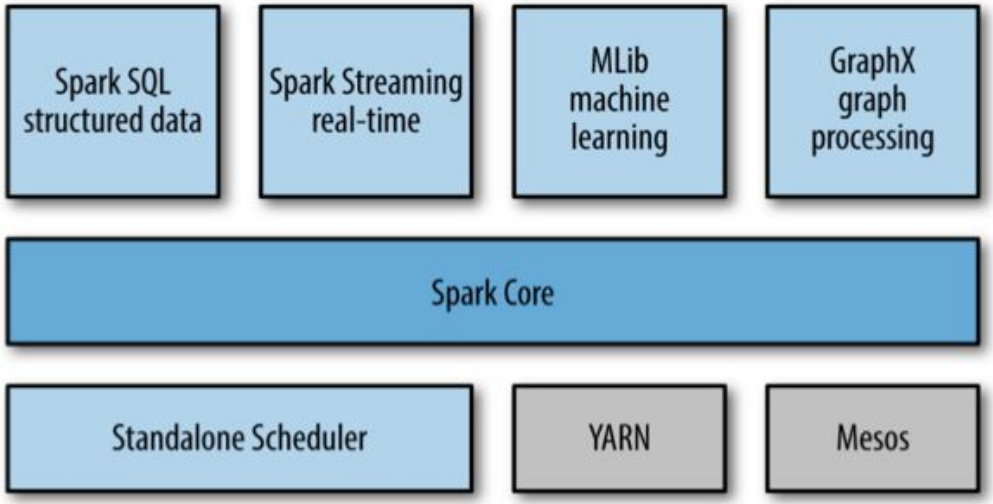


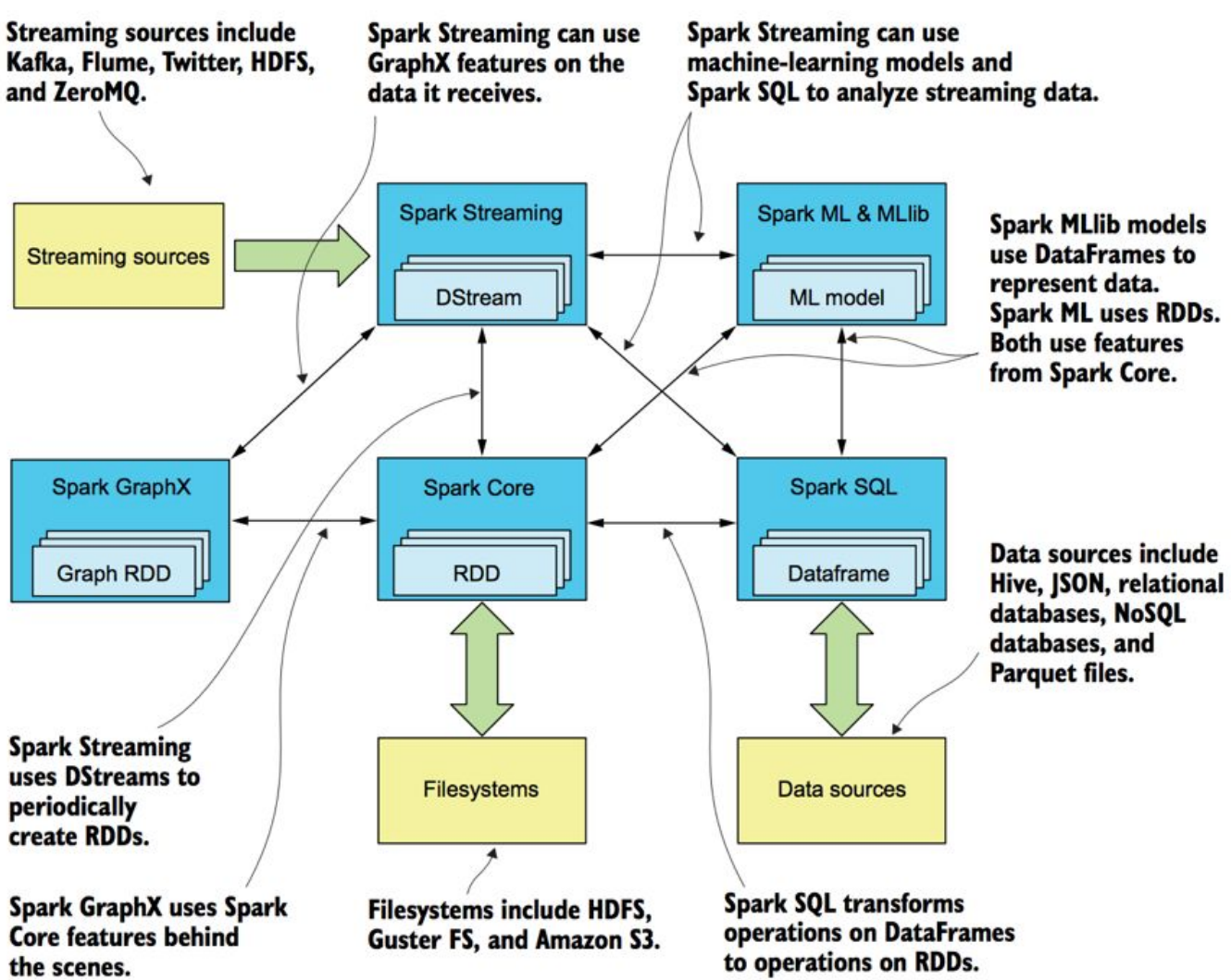
2015



2016

Spark Stack


Spark SQL
structured data
Spark Streaming
real-time
MLib
machine
learning
GraphX
graph
processing
Spark Core
Standalone Scheduler
YARN
Mesos


Streaming sources include Kafka, Flume, Twitter, HDFS, and ZeroMQ.
Spark Streaming can use GraphX features on the data it receives.
Spark Streaming can use machine-learning models and Spark SQL to analyze streaming data.
Streaming sources
Spark Streaming
DStream
Spark ML & MLlib
ML model
Spark MLlib models use DataFrames to represent data. Spark ML uses RDDs. Both use features from Spark Core.
Spark GraphX
Graph RDD
Spark Core
RDD
Spark SQL
Dataframe
Data sources include Hive, JSON, relational databases, NoSQL databases, and Parquet files.
Spark Streaming uses DStreams to periodically create RDDs.
Filesystems
Data sources
Spark GraphX uses Spark Core features behind the scenes.
Filesystems include HDFS, Guster FS, and Amazon S3.
Spark SQL transforms operations on DataFrames to operations on RDDs.

Spark Core

Basic functionalities: Task scheduling, fault recovery, memory management, storage systems' interaction, etc.

RDDs - Resilient Distributed Datasets

- Programming abstractions: Fault-tolerant collections of items distributed across many computer nodes that can be manipulated in parallel.
- Use of DAGs - Directed Acyclic Graph

Spark SQL

Allows working with structured data using either SQL or the DataFrame API

DataFrames

- High-level abstractions for basic data transformations
- Immutable distributed collections of data like RDDs, but which also organize data into named columns

Spark MLlib

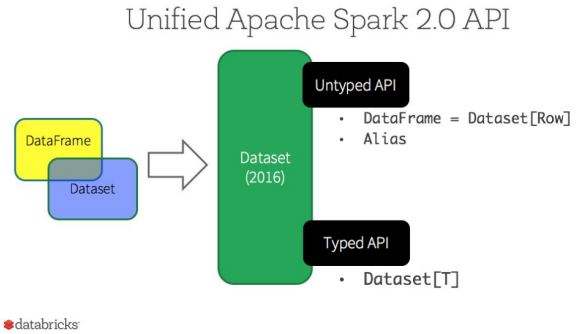
MLlib is Apache Spark's scalable machine learning library.

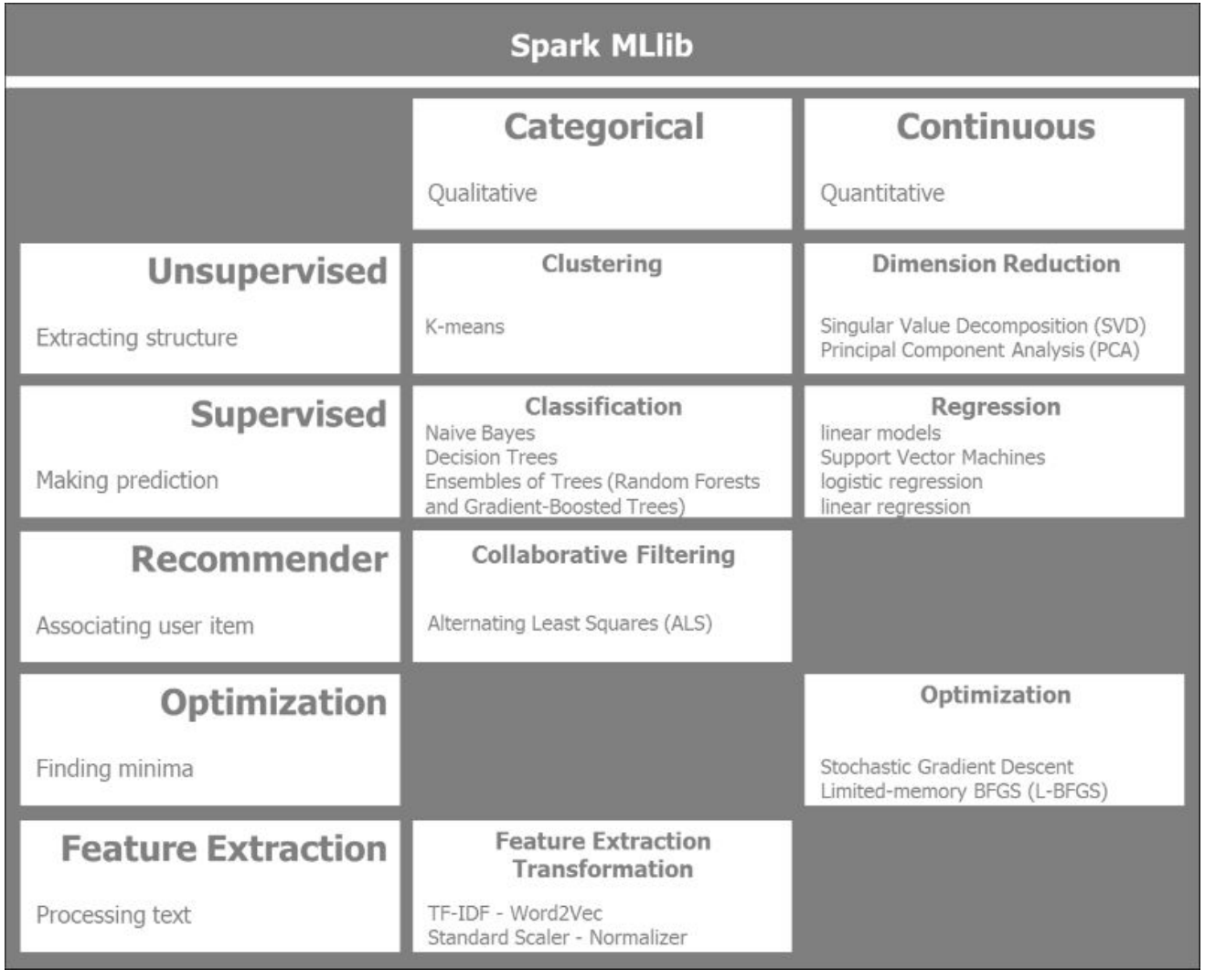
MLlib is

- Distributed machine learning on spark core
- Designed for simplicity, scalability, and easy integration
- Focuses on data problems and models

DataSet

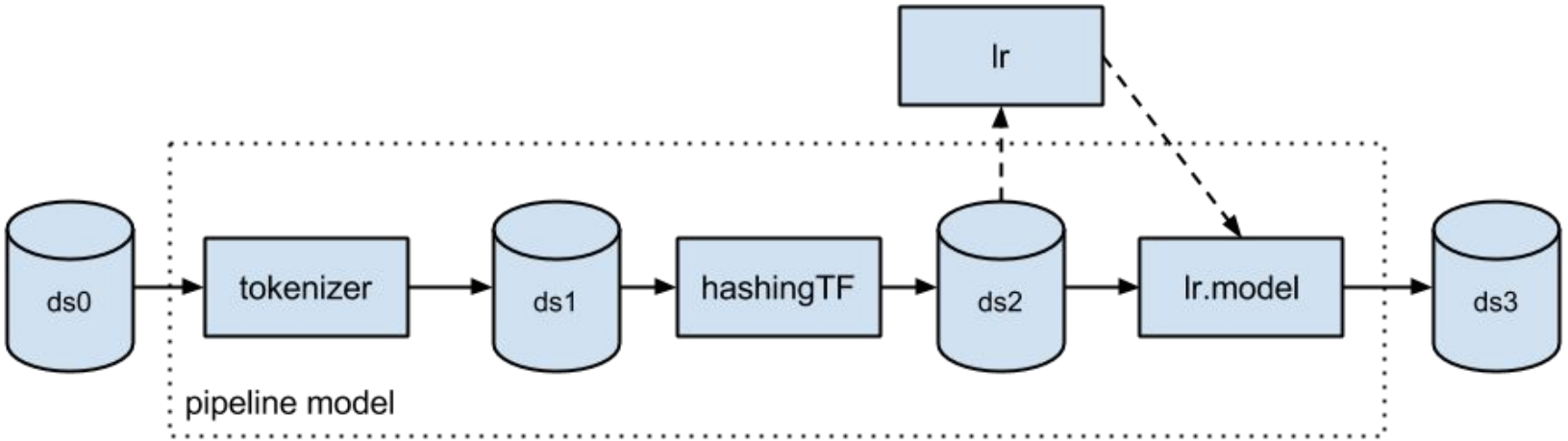
- A Dataset is a distributed collection of data.
- Provides the benefits of RDDs with the Spark SQL's optimized execution engine



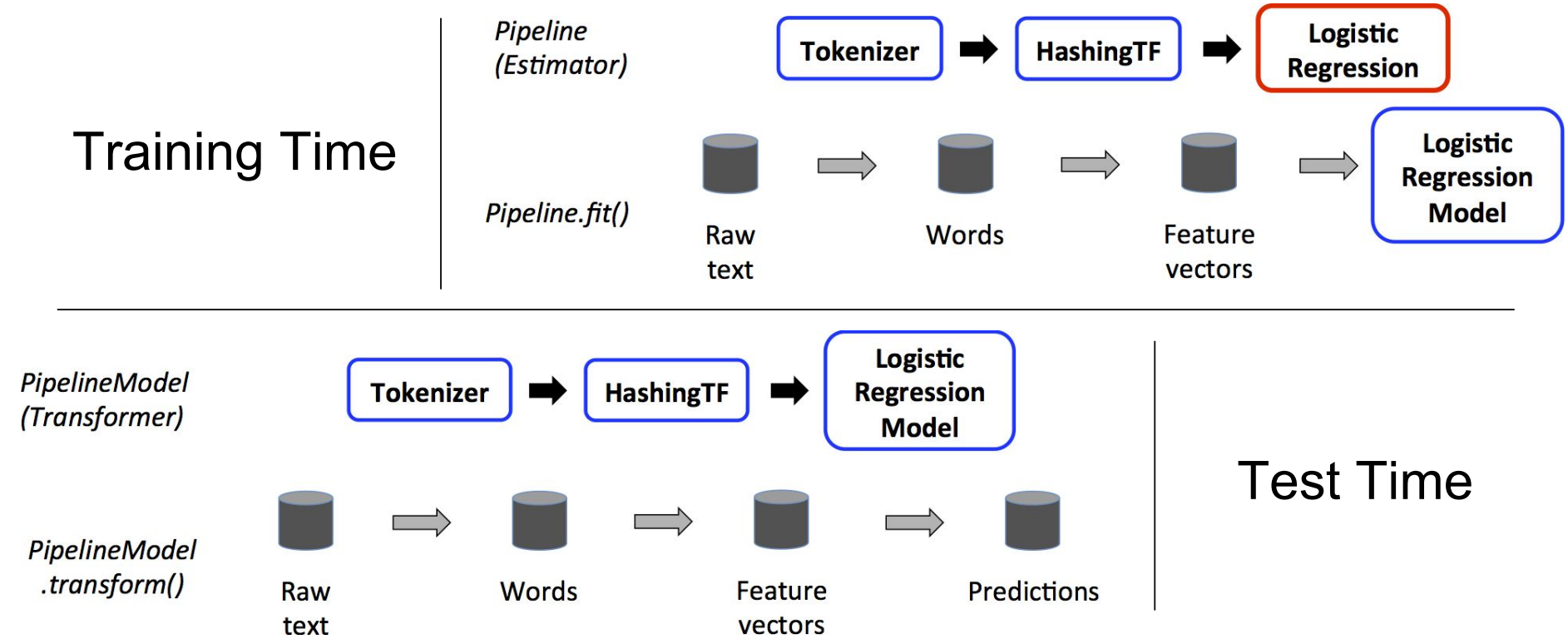

Spark MLlib
Categorical
Qualitative
Continuous
Quantitative
Unsupervised
Extracting structure
Clustering
K-means
Dimension Reduction
Singular Value Decomposition (SVD)
Principal Component Analysis (PCA)
Supervised
Making prediction
Classification
Naive Bayes
Decision Trees
Ensembles of Trees (Random Forests and Gradient-Boosted Trees)
Regression
linear models
Support Vector Machines
logistic regression
linear regression
Recommender
Associating user item
Collaborative Filtering
Alternating Least Squares (ALS)
Optimization
Finding minima
Optimization
Stochastic Gradient Descent
Limited-memory BFGS (L-BFGS)
Feature Extraction
Processing text
Feature Extraction
Transformation
TF-IDF - Word2Vec
Standard Scaler - Normalizer

MLlib Pipelines

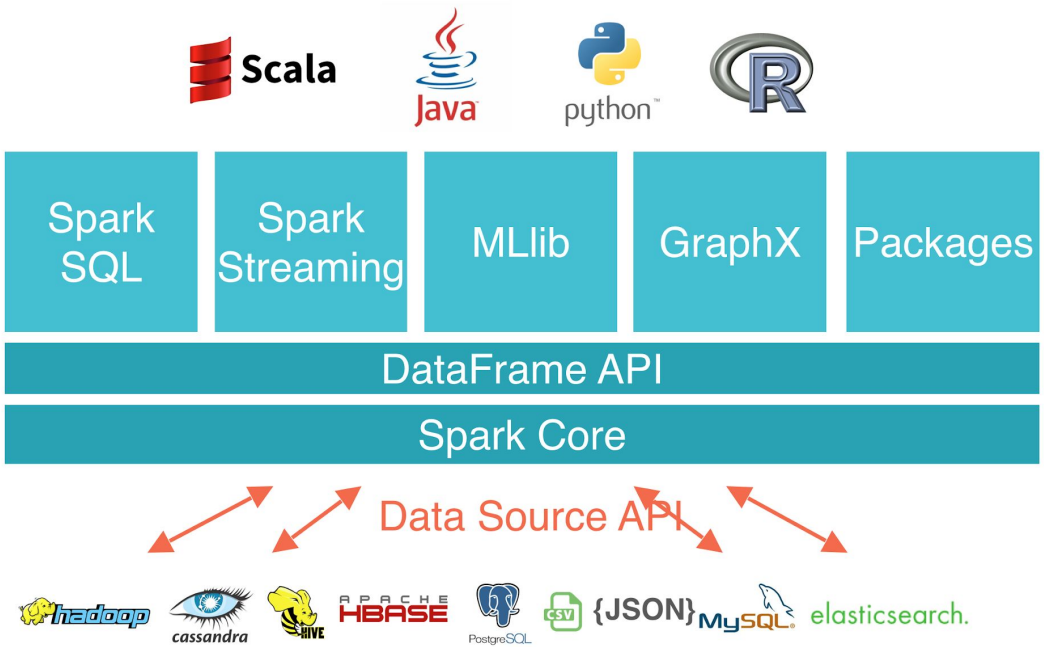
- Split each document's text into words
- Convert each document's words into a numerical feature vector
- Learn a prediction model using the feature vectors and labels



MLlib Pipelines - Working



Spark Programming Languages



Spark Programming Languages

Python

```
text_file = sc.textFile("hdfs://...")
counts = text_file.flatMap(lambda line:
          line.split(" ")) \
         .map(lambda word: (word,1)) \
         .reduceByKey(lambda a, b: a + b)
 counts.saveAsTextFile("hdfs://...")
```

Scala

```
val textFile = sc.textFile("hdfs://...")
val counts = textFile.flatMap(line =>
          line.split(" "))
          .map(word => (word, 1))
          .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
```

Java

```
JavaRDD<String> textFile =
 sc.textFile("hdfs://...");
JavaPairRDD<String, Integer> counts =
 textFile
    .flatMap(s -> Arrays.asList(s.split("
                                 ")).iterator())
    .mapToPair(word -> new Tuple2<>(word, 1))
    .reduceByKey((a, b) -> a + b);
counts.saveAsTextFile("hdfs://...");
```

Word Count Example

Spark Packages

Spark Packages features integrations

- Various data sources, management tools
- Higher level domain-specific libraries
- Machine learning algorithms
- Code samples, and other Spark content

spark-packages.org is a community package index

Spark in Cloud

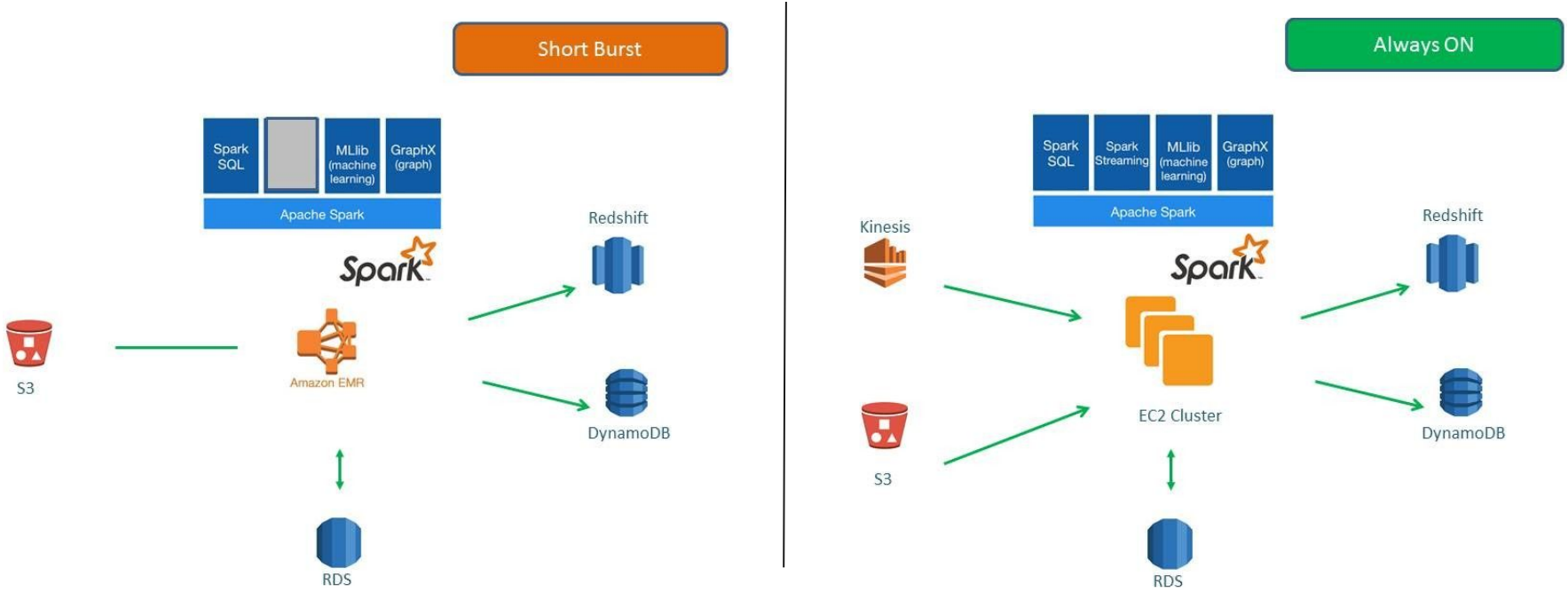
Spark can be deployed in a traditional on-premises data center as well as in the cloud. The cloud allows organizations to deploy Spark without the need to acquire hardware or specific setup expertise

Vendors who currently have an offer for the cloud include

- AWS
- Microsoft Azure
- Google Cloud
- Oracle Cloud
- IBM Bluemix
- Databricks

Spark in Cloud - AWS

With AWS, there are two primary methods of building big data cluster for spark.



Spark in Cloud - Google Cloud