Apache Spark

Lecture 25

APACHE Spark™

Apache Spark™

Apache Spark

- Apache Spark is an open-source, distributed processing system used for big data workloads. (**In-memory computing framework**)
  - It utilizes **in-memory caching** and optimized query execution for fast queries against data of any size.
  - Spark is a **fast and general engine for large-scale data processing**.

- The **fast** part means that it's faster than previous approaches to work with Big Data like classical MapReduce. The secret for being faster is that Spark runs on memory (RAM), and that makes the processing much faster than on disk drives.

- The **general** part means that it can be used for multiple things like running distributed SQL, creating data pipelines, ingesting data into a database, running Machine Learning algorithms, working with graphs or data streams, and much more.

What is Apache Spark?

Apache Spark is an open-source data processing engine to store and process data in real-time across various clusters of computers using simple programming constructs

Support various programming languages

Developers and data scientists incorporate Spark into their applications to rapidly query, analyze, and transform data at scale

What is Apache Spark - Benefits of Apache Spark

Speed

Engineered from the bottom-up for performance, Spark can be **100x faster than Hadoop for large scale data processing** by exploiting in memory computing and other optimizations. Spark is also fast when data is stored on disk, and currently holds the world record for large-scale on-disk sorting.

Ease of Use

Spark has easy-to-use APIs for operating on large datasets. This includes a collection of over 100 operators for transforming data and familiar data frame APIs for manipulating semi-structured data.

A Unified Engine

Spark comes packaged with higher-level libraries, including support for SQL queries, streaming data, machine learning and graph processing. These standard libraries increase developer productivity and can be seamlessly combined to create complex workflows.

Shortcomings of MapReduce

Learning objectives

- List the main bottlenecks of MapReduce
- Explain how Apache Spark solves them

Shortcomings of MapReduce

Force your pipeline into Map and Reduce steps

Other workflows? i.e. join, filter, map-reduce-map

Shortcomings of MapReduce

Read from disk for each MapReduce job

Iterative algorithms? i.e. machine learning

Shortcomings of MapReduce

Only native JAVA programming interface

Other languages?
Interactivity?

Solution?

- New framework: same features of MapReduce and more
- Capable of reusing Hadoop ecosystem, e.g. HDFS, YARN…
- Born at UC Berkeley

Solutions by Spark

Other workflows? i.e. join, filter, map-reduce-map

~20 highly efficient distributed operations, any combination of them

Solutions by Spark

Iterative algorithms? i.e. machine learning

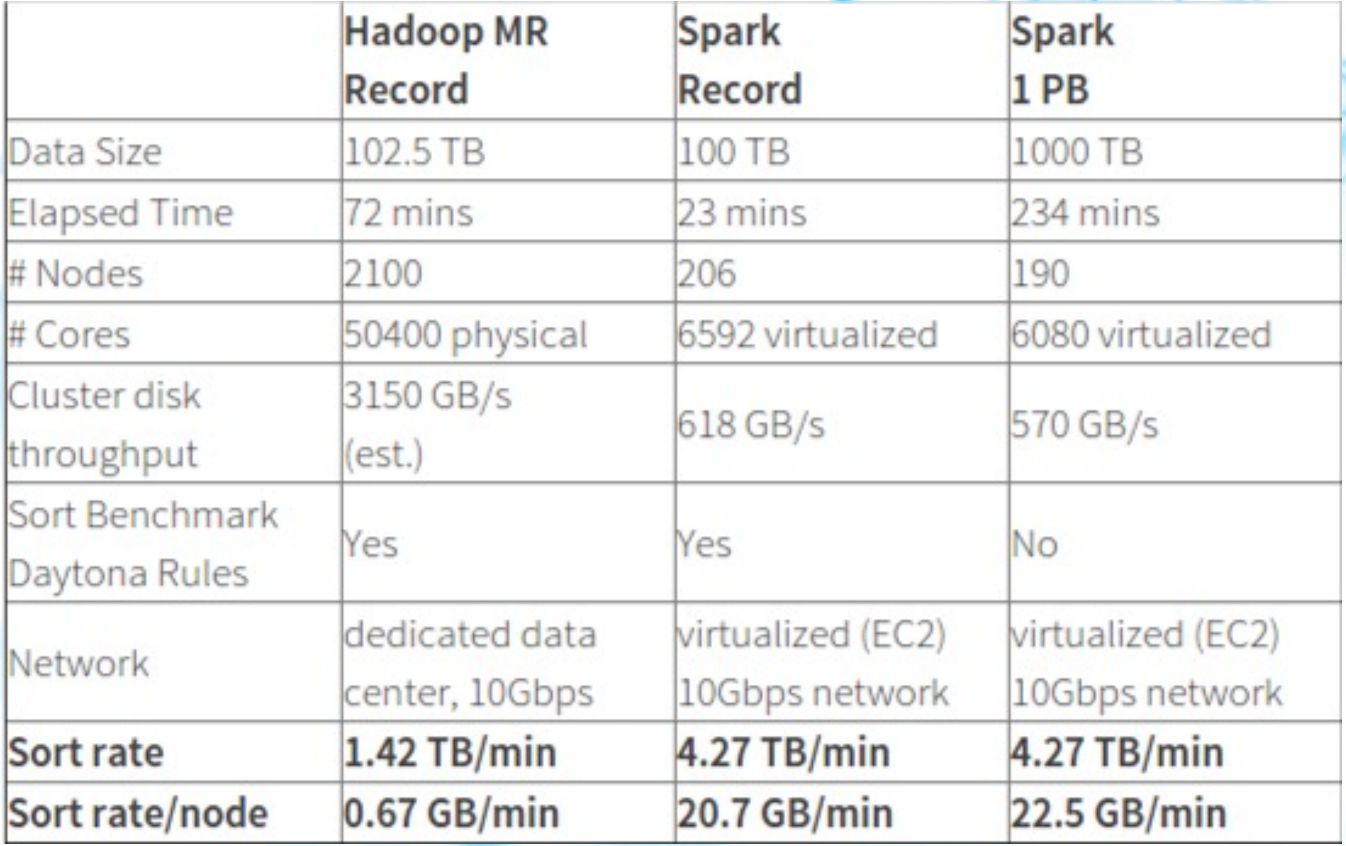
in-memory caching of data, specified by the user

Solutions by Spark

Interactivity? Other languages?

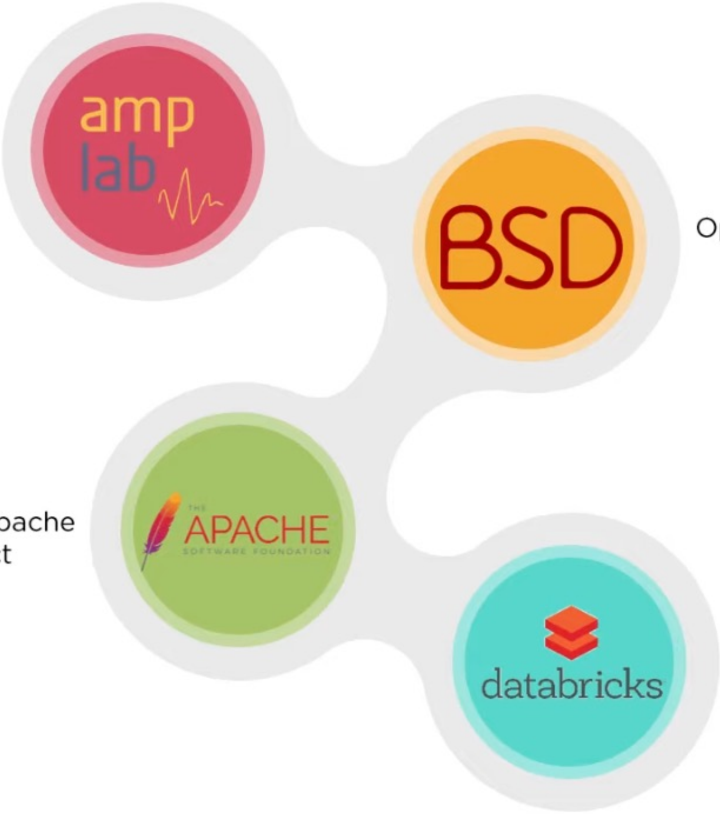
Native Python, Scala (, R) interface. Interactive shells.

100TB Sorting competition



	Hadoop MR Record	**Spark Record**	**Spark 1 PB**
Data Size	102.5 TB	100 TB	1000 TB
Elapsed Time	72 mins	23 mins	234 mins
# Nodes	2100	206	190
# Cores	50400 physical	6592 virtualized	6080 virtualized
Cluster disk throughput	3150 GB/s (est.)	618 GB/s	570 GB/s
Sort Benchmark Daytona Rules	Yes	Yes	No
Network	dedicated data center, 10Gbps	virtualized (EC2) 10Gbps network	virtualized (EC2) 10Gbps network
Sort rate	**1.42 TB/min**	**4.27 TB/min**	**4.27 TB/min**
Sort rate/node	**0.67 GB/min**	**20.7 GB/min**	**22.5 GB/min**

History of Apache Spark



2009

Started as a project at
UC Berkley AMPLab

2010

Open sourced under a
BSD license

2013

Spark became an Apache
top level project

2014

Used by Databricks to sort
large-scale datasets and set
a new world record

Apache Spark

Apache Spark is a lightning-fast **unified analytics engine** for big data and machine learning. It was originally developed at UC Berkeley in 2009.

The largest open source project in data processing.

Since its release, **Apache Spark**, the unified analytics engine, has seen rapid adoption by enterprises across a wide range of industries. Internet powerhouses such as Netflix, Yahoo, and eBay have deployed Spark at massive scale, collectively processing multiple petabytes of data on clusters of over 8,000 nodes. It has quickly become the largest open source community in big data, with over 1000 contributors from 250+ organizations.

Spark Features

Fast processing

Spark contains Resilient Distributed Datasets (RDD) which saves time taken in reading, and writing operations and hence, it runs almost ten to hundred times faster than Hadoop

Spark Features

Fast processing

Hive-edited.pptx

In-memory computing

In Spark, data is stored in the RAM, so it can access the data quickly and accelerate the speed of analytics

Spark Features

Fast processing

In-memory computing

Flexible

Spark supports multiple languages and allows the developers to write applications in Java, Scala, R, or Python

Spark Features

Fast processing

In-memory computing

Flexible

Fault tolerance

Spark contains Resilient Distributed Datasets (RDD) that are designed to handle the failure of any worker node in the cluster. Thus, it ensures that the loss of data reduces to zero

Spark Features

Fast processing

In-memory computing

Flexible

Fault tolerance

Better analytics

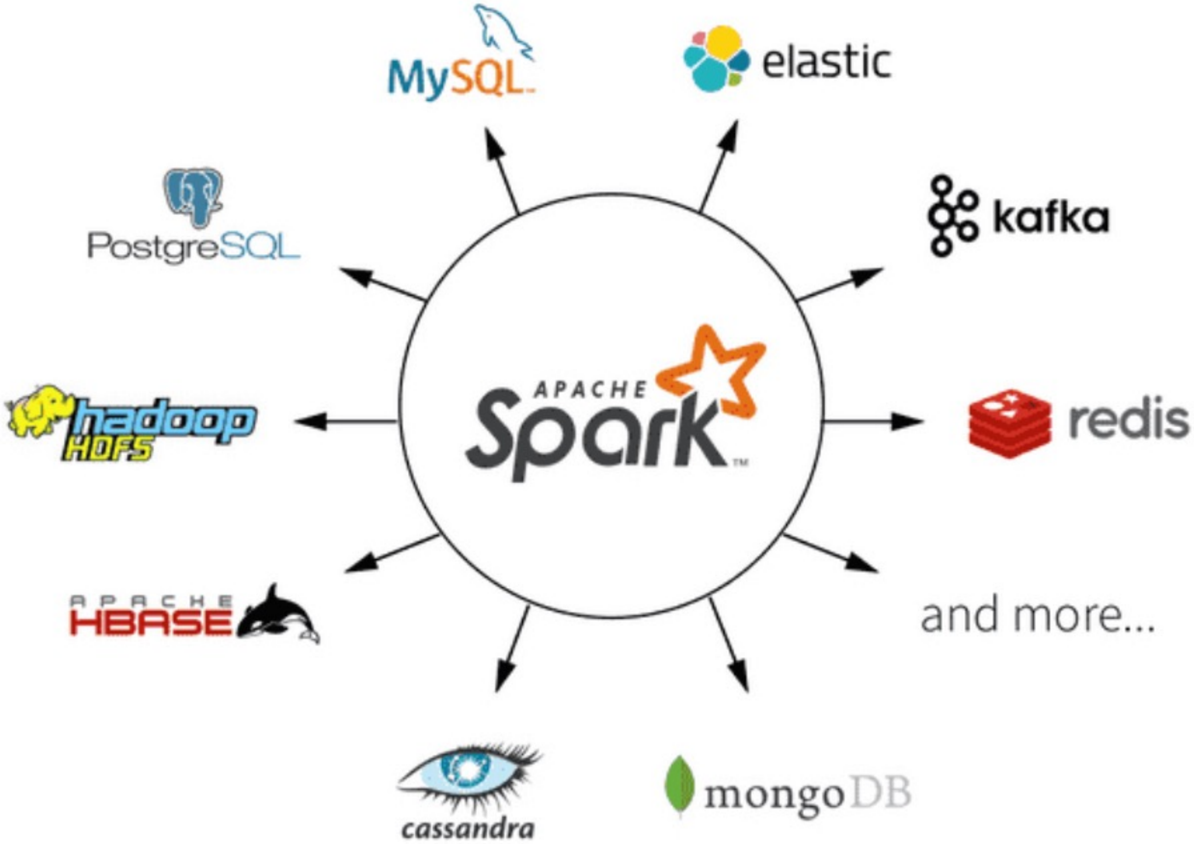
Spark has a rich set of SQL queries, machine learning algorithms, complex analytics, etc. With all these functionalities, analytics can be performed better

Features

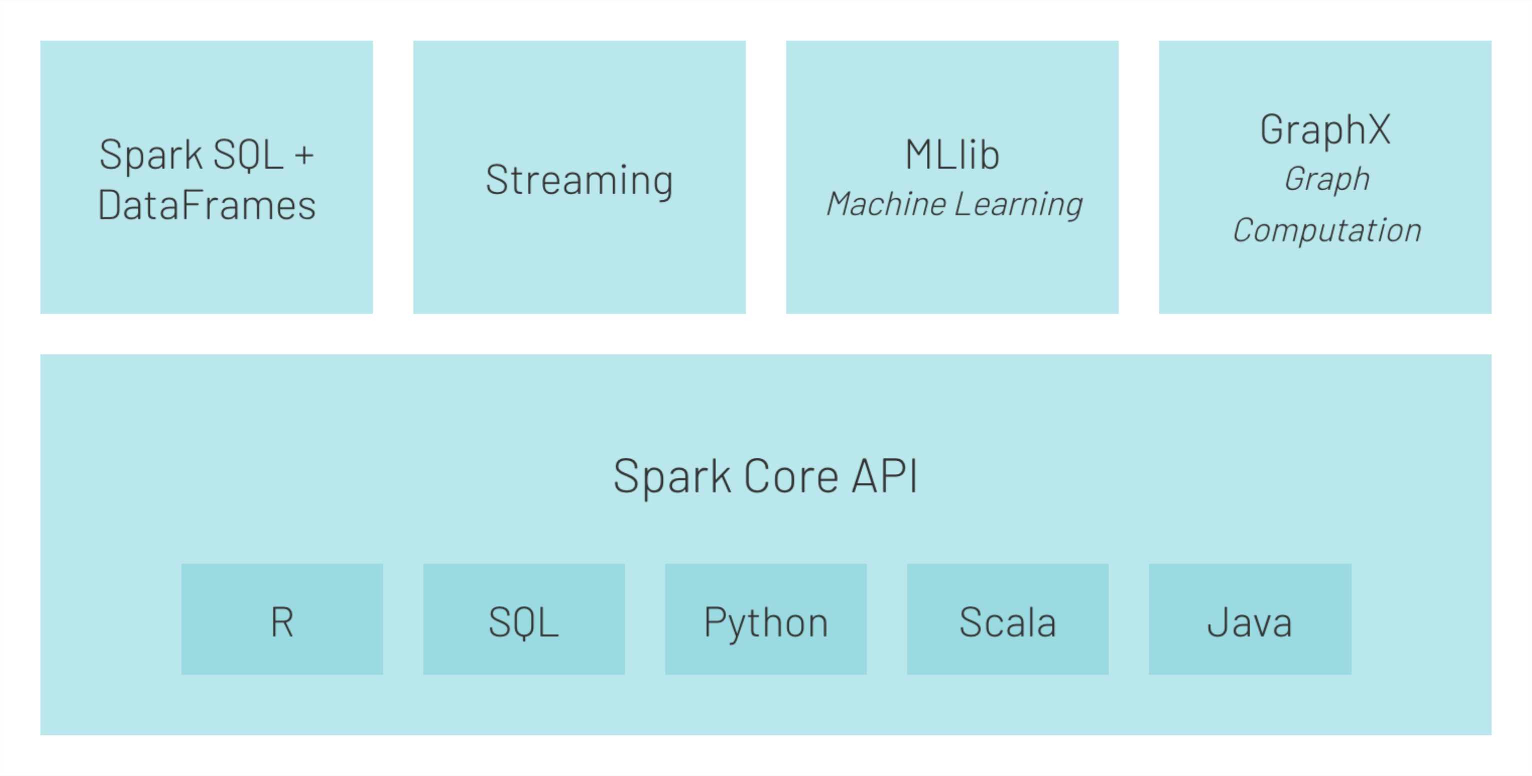
- **Fast Processing**
  - The most important feature of Apache Spark that has made the big data world choose this technology over others is its speed. Big data is characterized by volume, variety, velocity, and veracity which needs to be processed at a higher speed. Spark contains **Resilient Distributed Dataset (RDD)** which saves time in reading and writing operations, allowing it to run almost **ten to one hundred times faster than Hadoop**.
- **Flexibility**
  - Apache Spark supports multiple languages and allows the developers to write applications in Java, Scala, R, or Python.
- **In-memory Computing**
  - Spark stores the data in the RAM of servers which allows quick access and in turn accelerates the speed of analytics.

Features

- **Real-time Processing**
  - Spark is able to process **real-time streaming data**. Unlike MapReduce which processes only stored data, Spark is able to process real-time data and is, therefore, able to produce instant outcomes.
- **Better Analytics**
  - In contrast to MapReduce that includes Map and Reduce functions, Spark includes much more than that. Apache Spark consists of a rich set of **SQL queries, machine learning algorithms, complex analytics**, etc. With all these functionalities, analytics can be performed in a better fashion with the help of Spark.


APACHE Spark
MySQL
elastic
kafka
redis
and more...
mongoDB
cassandra
APACHE HBASE
hadoop HDFS
PostgreSQL

Apache Spark Ecosystem



Spark SQL + DataFrames

Streaming

MLlib
Machine Learning

GraphX
Graph Computation

Spark Core API

R

SQL

Python

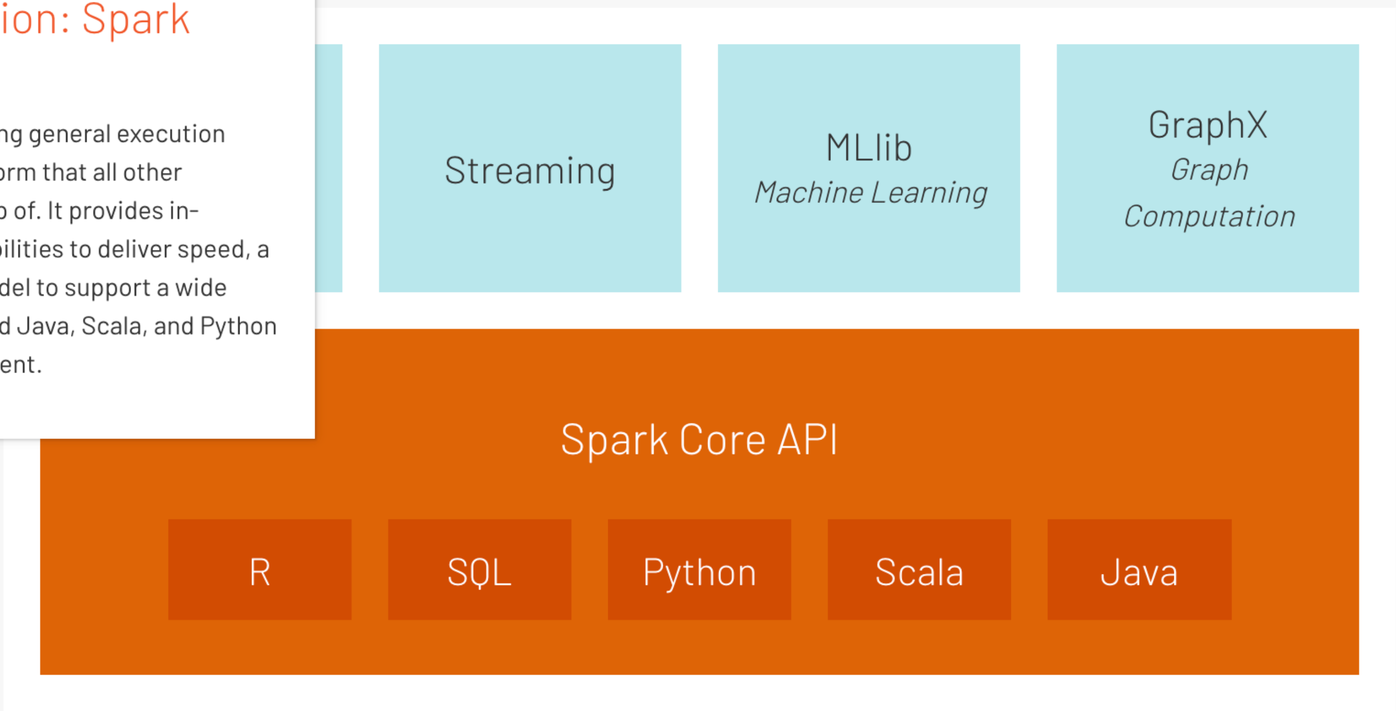
Scala

Java

Apache Spark Ecosystem

General Execution: Spark Core

Spark Core is the underlying general execution engine for the Spark platform that all other functionality is built on top of. It provides in-memory computing capabilities to deliver speed, a generalized execution model to support a wide variety of applications, and Java, Scala, and Python APIs for ease of development.



Streaming

MLlib
Machine Learning

GraphX
Graph Computation

Spark Core API

R

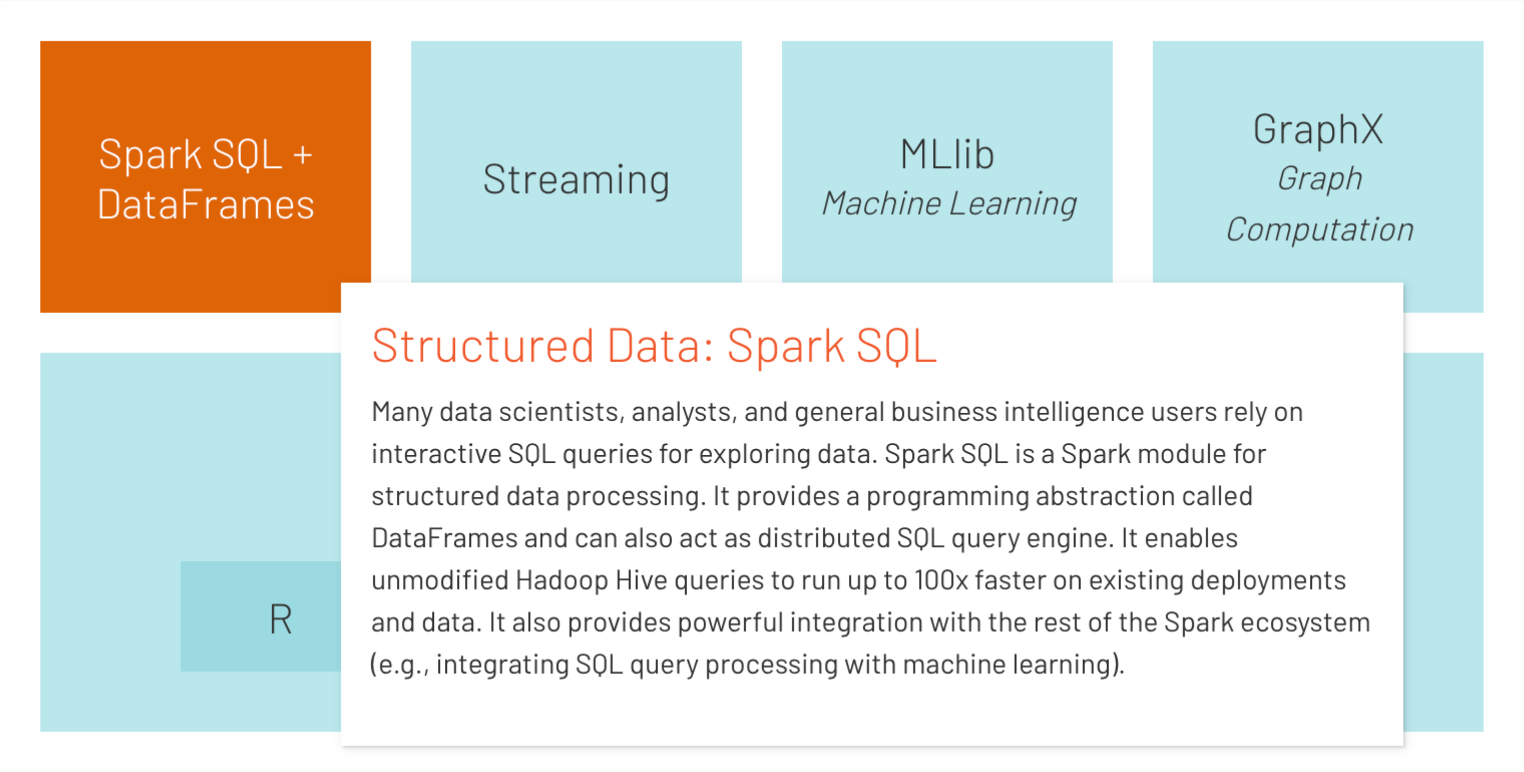
SQL

Python

Scala

Java

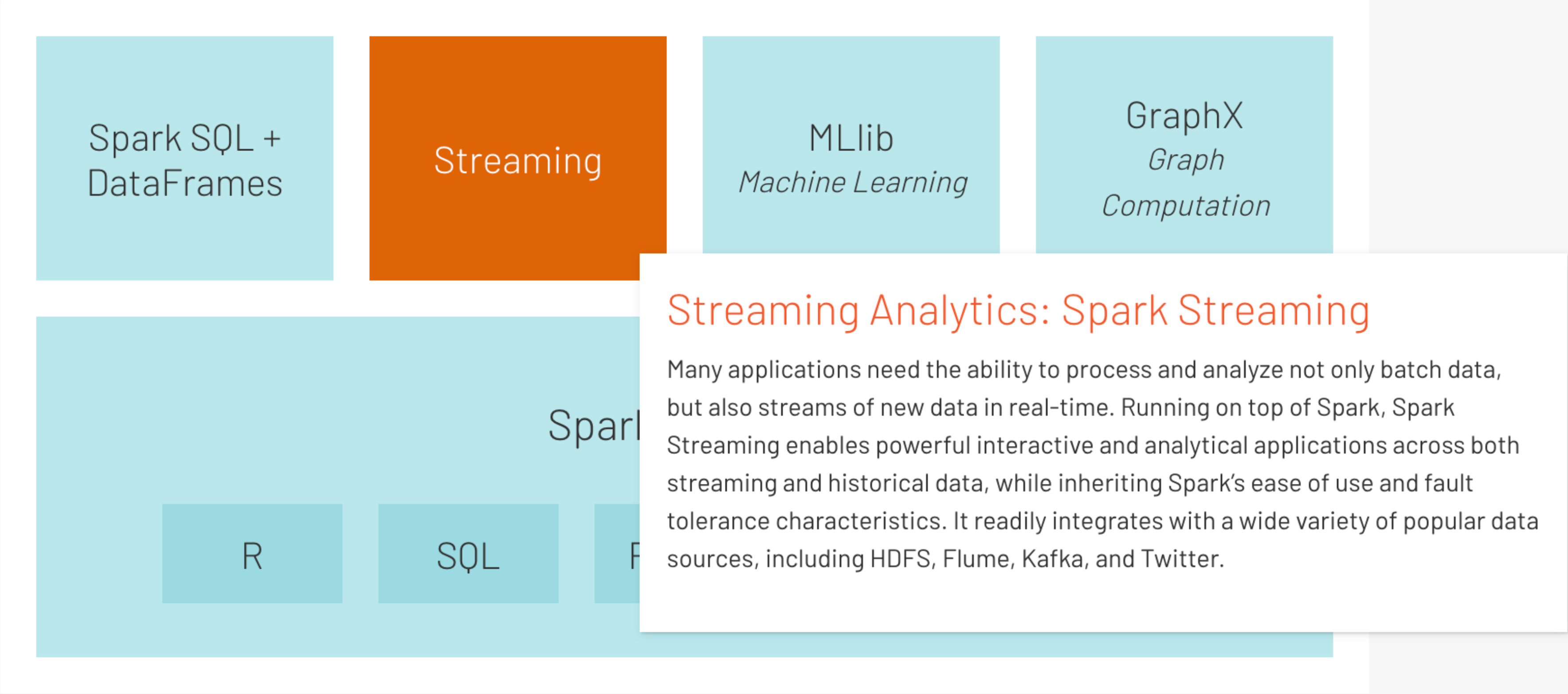
Apache Spark Ecosystem



Structured Data: Spark SQL

Many data scientists, analysts, and general business intelligence users rely on interactive SQL queries for exploring data. Spark SQL is a Spark module for structured data processing. It provides a programming abstraction called DataFrames and can also act as distributed SQL query engine. It enables unmodified Hadoop Hive queries to run up to 100x faster on existing deployments and data. It also provides powerful integration with the rest of the Spark ecosystem (e.g., integrating SQL query processing with machine learning).

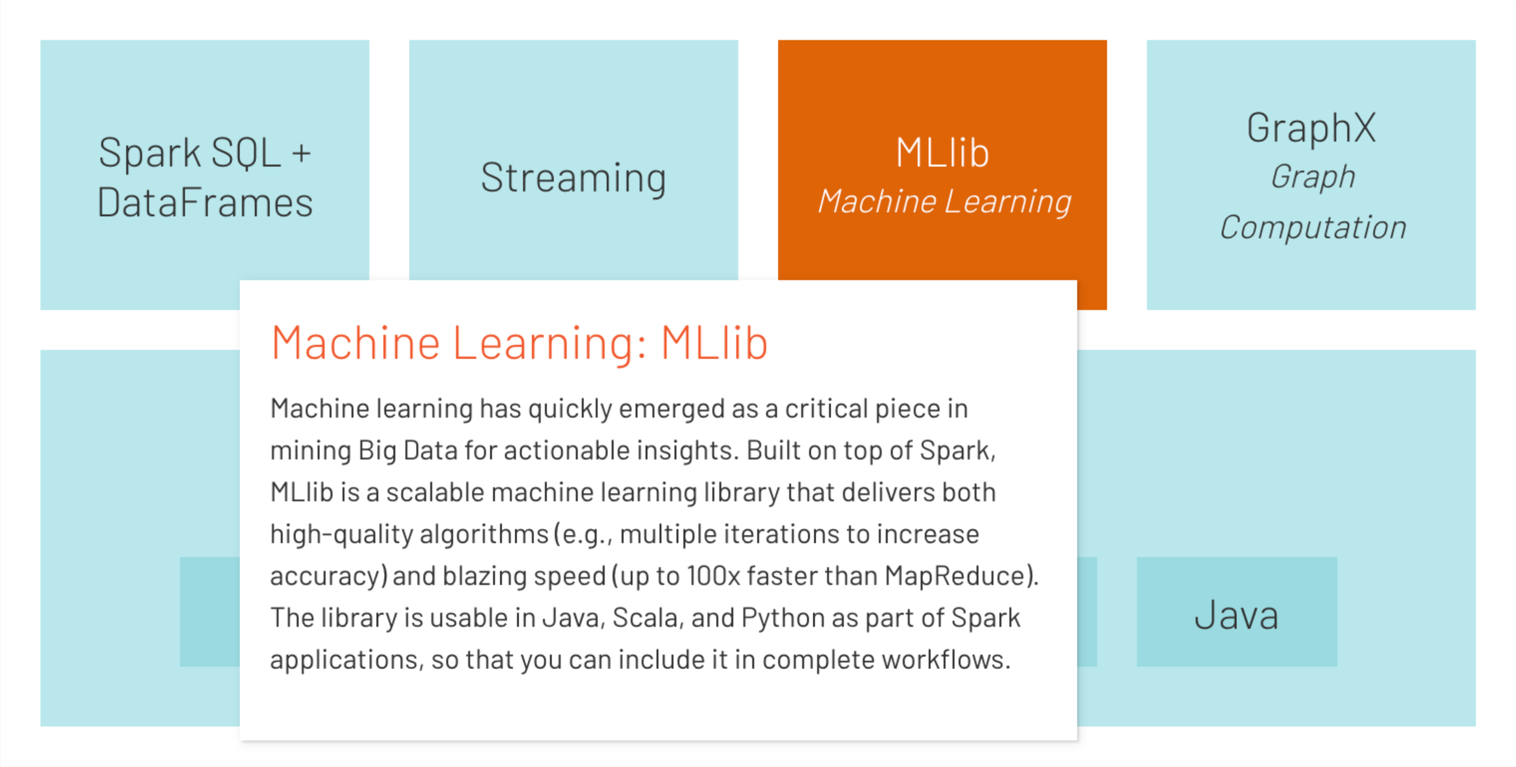
Apache Spark Ecosystem



Streaming Analytics: Spark Streaming

Many applications need the ability to process and analyze not only batch data, but also streams of new data in real-time. Running on top of Spark, Spark Streaming enables powerful interactive and analytical applications across both streaming and historical data, while inheriting Spark's ease of use and fault tolerance characteristics. It readily integrates with a wide variety of popular data sources, including HDFS, Flume, Kafka, and Twitter.

Apache Spark Ecosystem



Spark SQL + DataFrames

Streaming

MLlib
Machine Learning

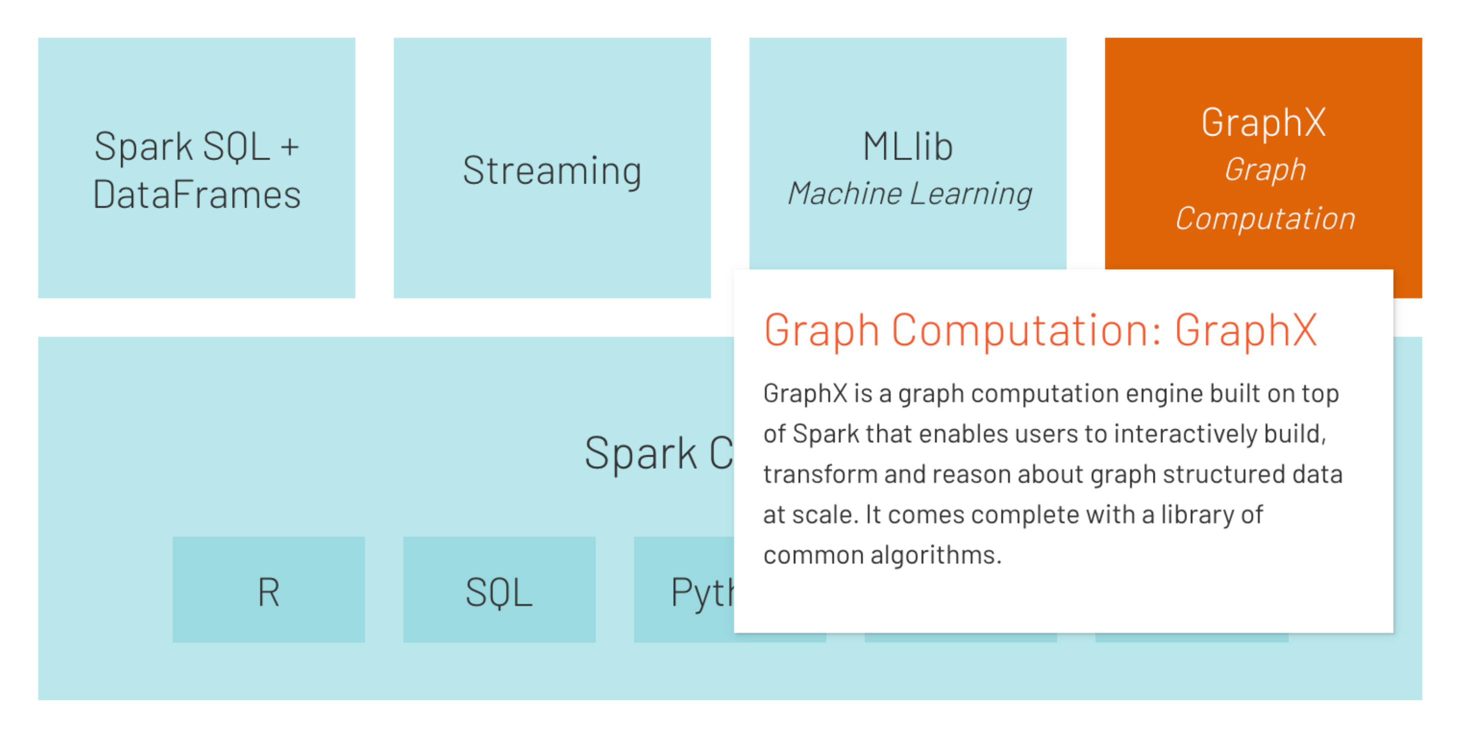
GraphX
Graph Computation

Java

Machine Learning: MLlib

Machine learning has quickly emerged as a critical piece in mining Big Data for actionable insights. Built on top of Spark, MLlib is a scalable machine learning library that delivers both high-quality algorithms (e.g., multiple iterations to increase accuracy) and blazing speed (up to 100x faster than MapReduce). The library is usable in Java, Scala, and Python as part of Spark applications, so that you can include it in complete workflows.

Apache Spark Ecosystem

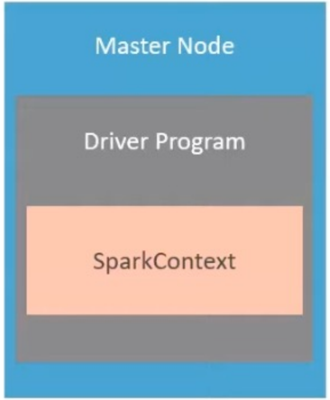


Graph Computation: GraphX

GraphX is a graph computation engine built on top of Spark that enables users to interactively build, transform and reason about graph structured data at scale. It comes complete with a library of common algorithms.

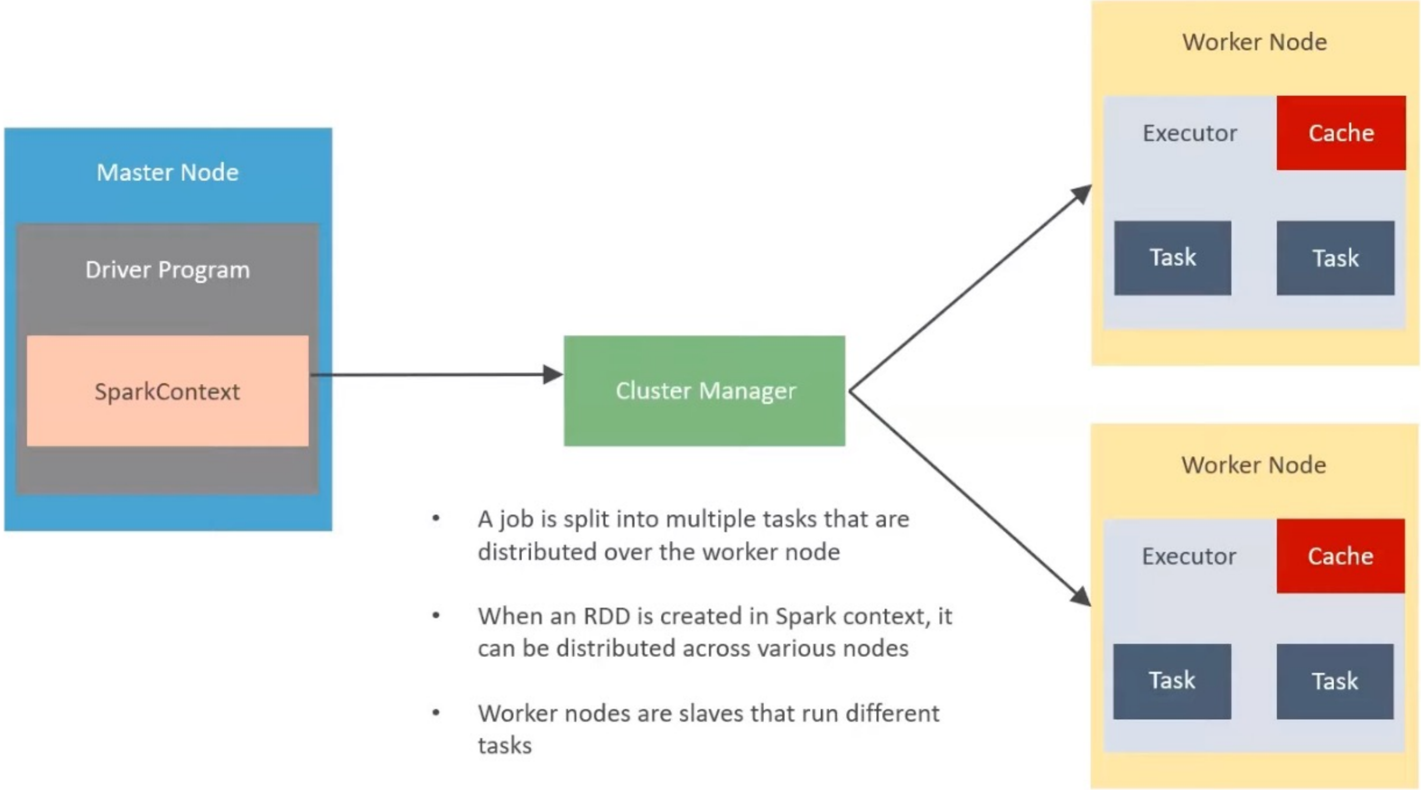
Spark Architecture

Apache Spark uses a master-slave architecture that consists of a driver, that runs on a master node, and multiple executors which run across the worker nodes in the cluster



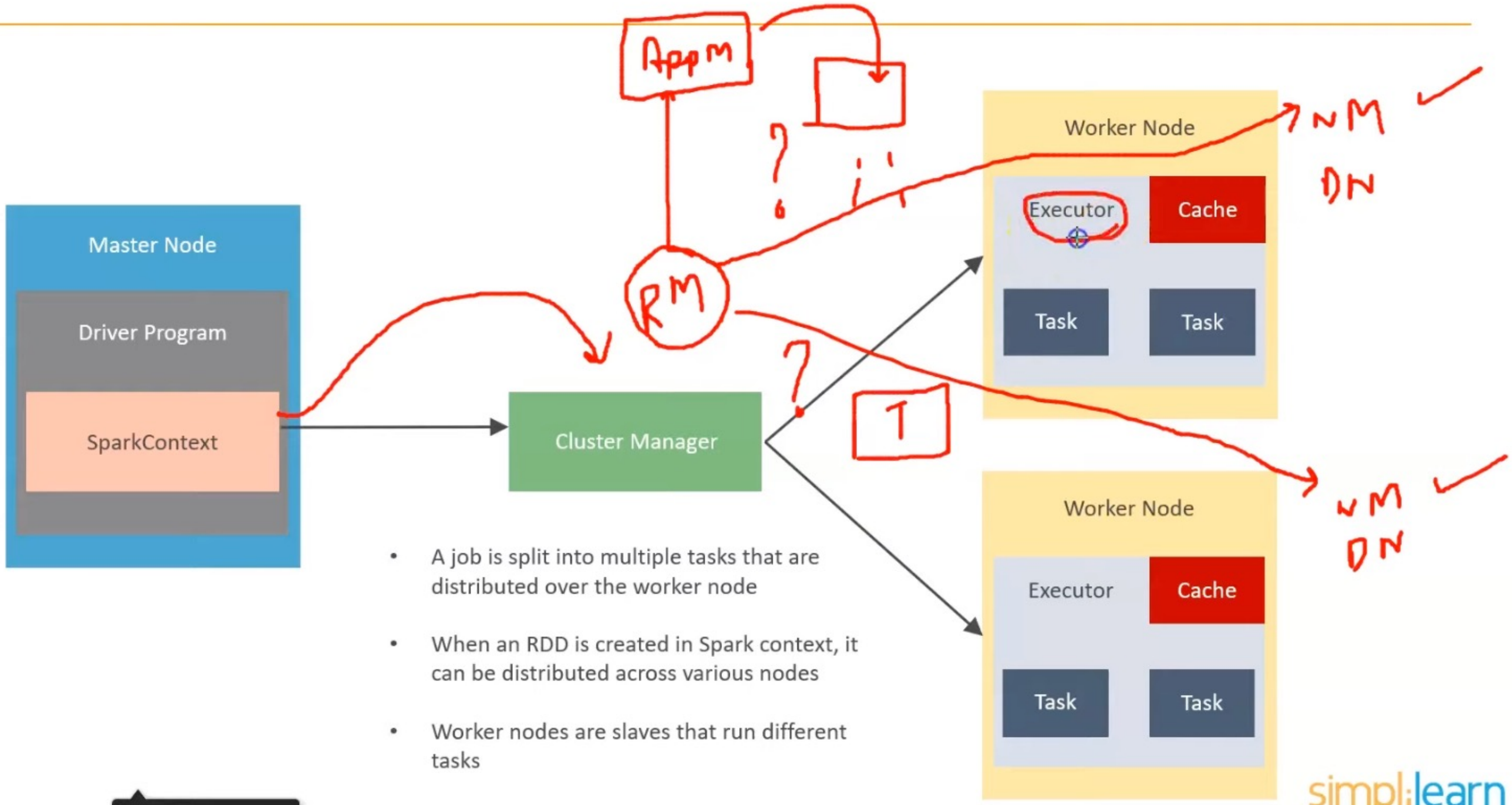
- Master Node has a Driver Program
- The Spark code behaves as a driver program and creates a SparkContext, which is a gateway to all the Spark functionalities

Spark Architecture



- A job is split into multiple tasks that are distributed over the worker node
- When an RDD is created in Spark context, it can be distributed across various nodes
- Worker nodes are slaves that run different tasks

Spark Architecture



- A job is split into multiple tasks that are distributed over the worker node
- When an RDD is created in Spark context, it can be distributed across various nodes
- Worker nodes are slaves that run different tasks

Spark Cluster Managers

Standalone mode

1

By default, applications submitted to the standalone mode cluster will run in FIFO order, and each application will try to use all available nodes

2

Apache Mesos is an open-source project to manage computer clusters, and can also run Hadoop applications

3

Apache YARN is the cluster resource manager of Hadoop 2. Spark can be run on YARN

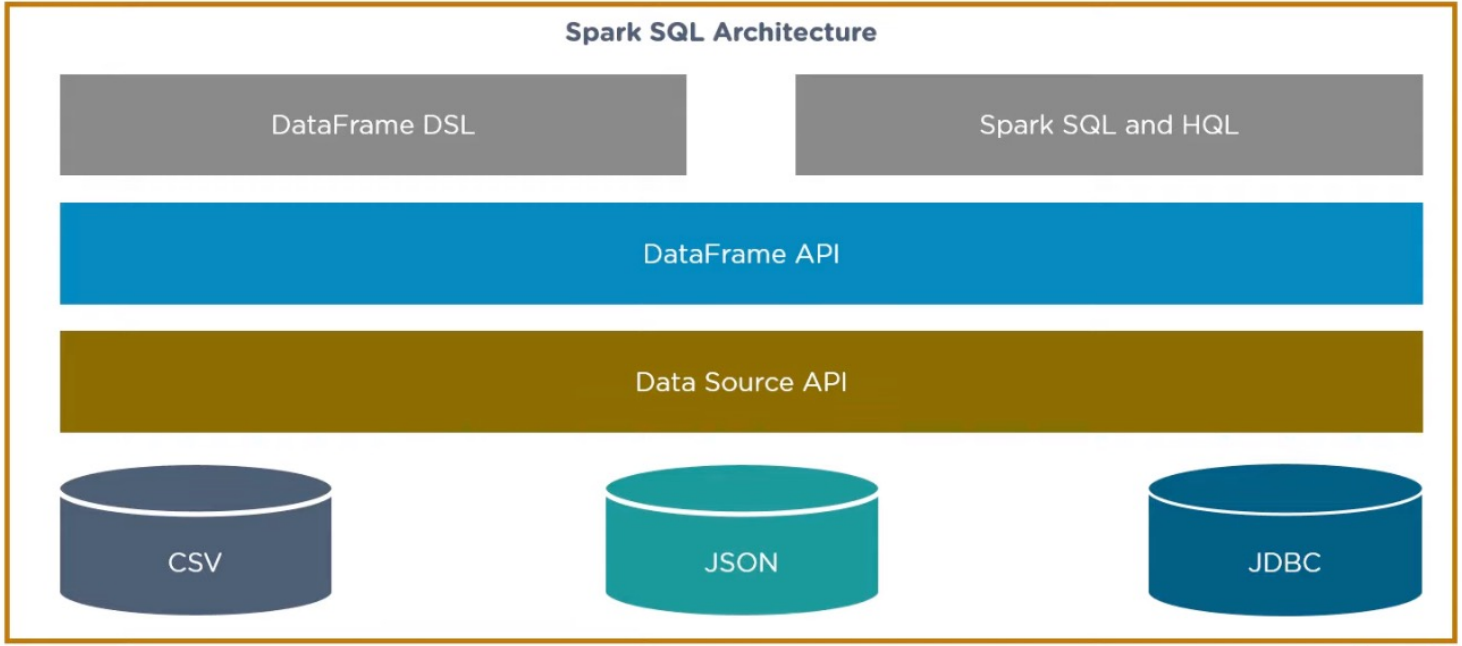
4

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications

Spark SQL

Spark SQL framework component is used for structured and semi-structured data processing

Spark SQL



Spark SQL Architecture

DataFrame DSL

Spark SQL and HQL

DataFrame API

Data Source API

CSV

JSON

JDBC

Spark MLlib

MLlib is a low-level machine learning library that is simple to use, is scalable, and compatible with various programming languages

MLlib eases the deployment and development of scalable machine learning algorithms

MLlib

It contains machine learning libraries that have an implementation of various machine learning algorithms

Clustering

Classification

Collaborative Filtering

GraphX

GraphX is Spark's own Graph Computation Engine and data store

GraphX

Provides a uniform tool for ETL

Exploratory data analysis

Interactive graph computations