Introduction to Apache Spark

Tushar B. Kute,
http://tusharkute.com

Introduction

- Industries are using Hadoop extensively to analyze their data sets.
- The reason is that Hadoop framework is based on a simple programming model (MapReduce) and it enables a computing solution that is scalable, flexible, fault-tolerant and cost effective.
- Here, the main concern is to maintain speed in processing large datasets in terms of waiting time between queries and waiting time to run the program.

Apache Spark

- Spark was introduced by Apache Software Foundation for speeding up the Hadoop computational computing software process.
- As against a common belief, Spark is not a modified version of Hadoop and is not, really, dependent on Hadoop because it has its own cluster management. Hadoop is just one of the ways to implement Spark.
- Spark uses Hadoop in two ways – one is storage and second is processing. Since Spark has its own cluster management computation, it uses Hadoop for storage purpose only.

Apache Spark

- Apache Spark is a lightning-fast cluster computing technology, designed for fast computation.
- It is based on Hadoop MapReduce and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing.
- The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application.

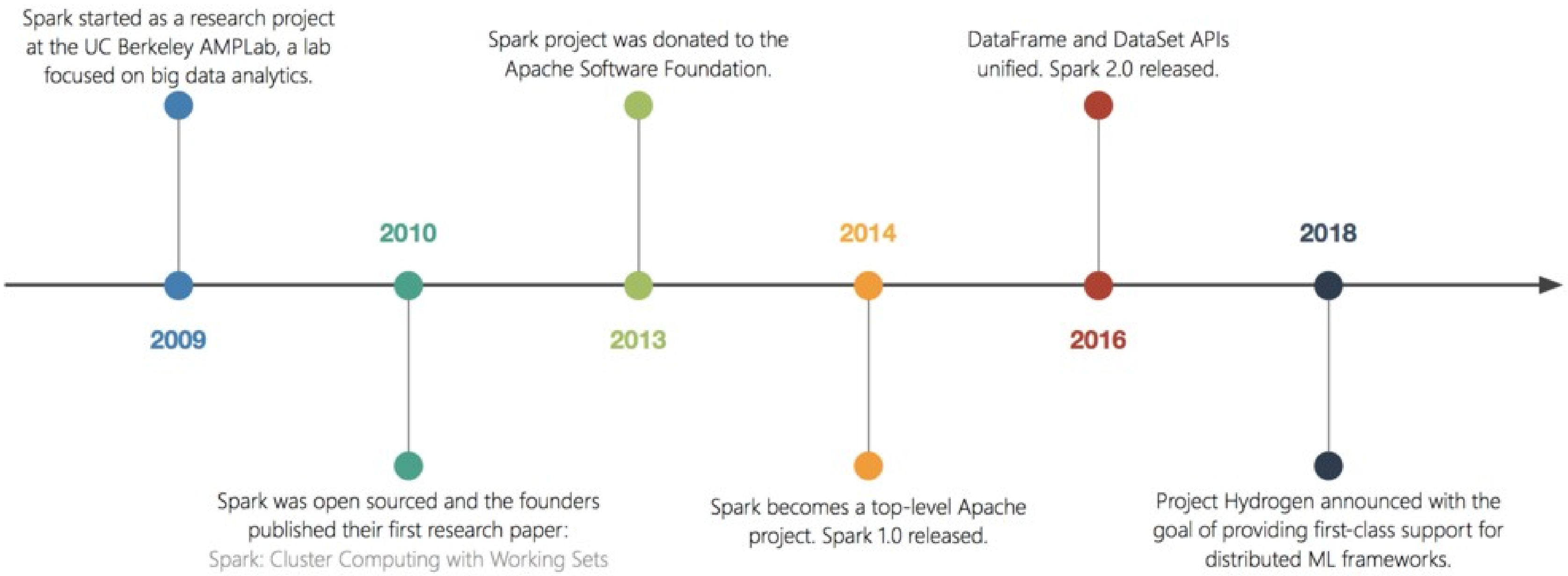
Apache Spark

- Spark is designed to cover a wide range of workloads such as batch applications, iterative algorithms, interactive queries and streaming.
- Apart from supporting all these workload in a respective system, it reduces the management burden of maintaining separate tools.

Apache Spark

- Basically, Apache Spark offers high-level APIs to users, such as Java, Scala, Python, and R.
- Although, Spark is written in Scala still offers rich APIs in Scala, Java, Python, as well as R. We can say, it is a tool for running spark applications.
- Most importantly, by comparing Spark with Hadoop, it is 100 times faster than Hadoop In-Memory mode and 10 times faster than Hadoop On-Disk mode.

Apache Spark: Timeline



2009

Spark started as a research project at the UC Berkeley AMPLab, a lab focused on big data analytics.

2010

Spark was open sourced and the founders published their first research paper: Spark: Cluster Computing with Working Sets

2013

Spark project was donated to the Apache Software Foundation.

2014

Spark becomes a top-level Apache project. Spark 1.0 released.

2016

DataFrame and DataSet APIs unified. Spark 2.0 released.

2018

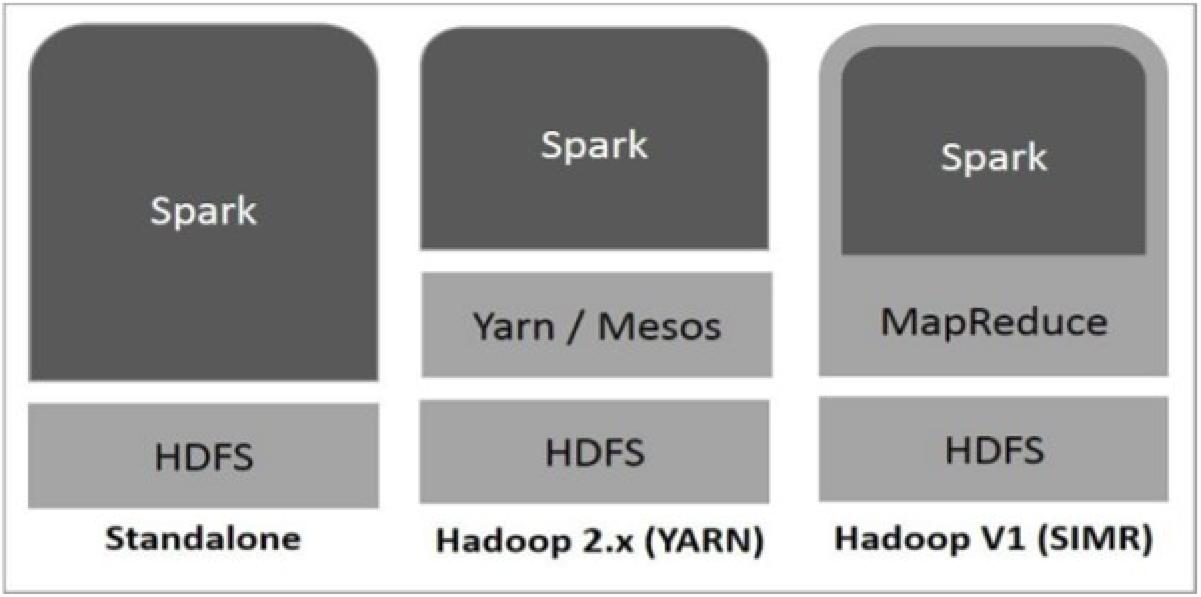
Project Hydrogen announced with the goal of providing first-class support for distributed ML frameworks.

Apache Spark: Features

- Speed – Spark helps to run an application in Hadoop cluster, up to 100 times faster in memory, and 10 times faster when running on disk. This is possible by reducing number of read/write operations to disk. It stores the intermediate processing data in memory.
- Supports multiple languages – Spark provides built-in APIs in Java, Scala, or Python. Therefore, you can write applications in different languages. Spark comes up with 80 high-level operators for interactive querying.
- Advanced Analytics – Spark not only supports ‘Map’ and ‘reduce’. It also supports SQL queries, Streaming data, Machine learning (ML), and Graph algorithms.

Apache Spark on Hadoop

- The following diagram shows three ways of how Spark can be built with Hadoop components.



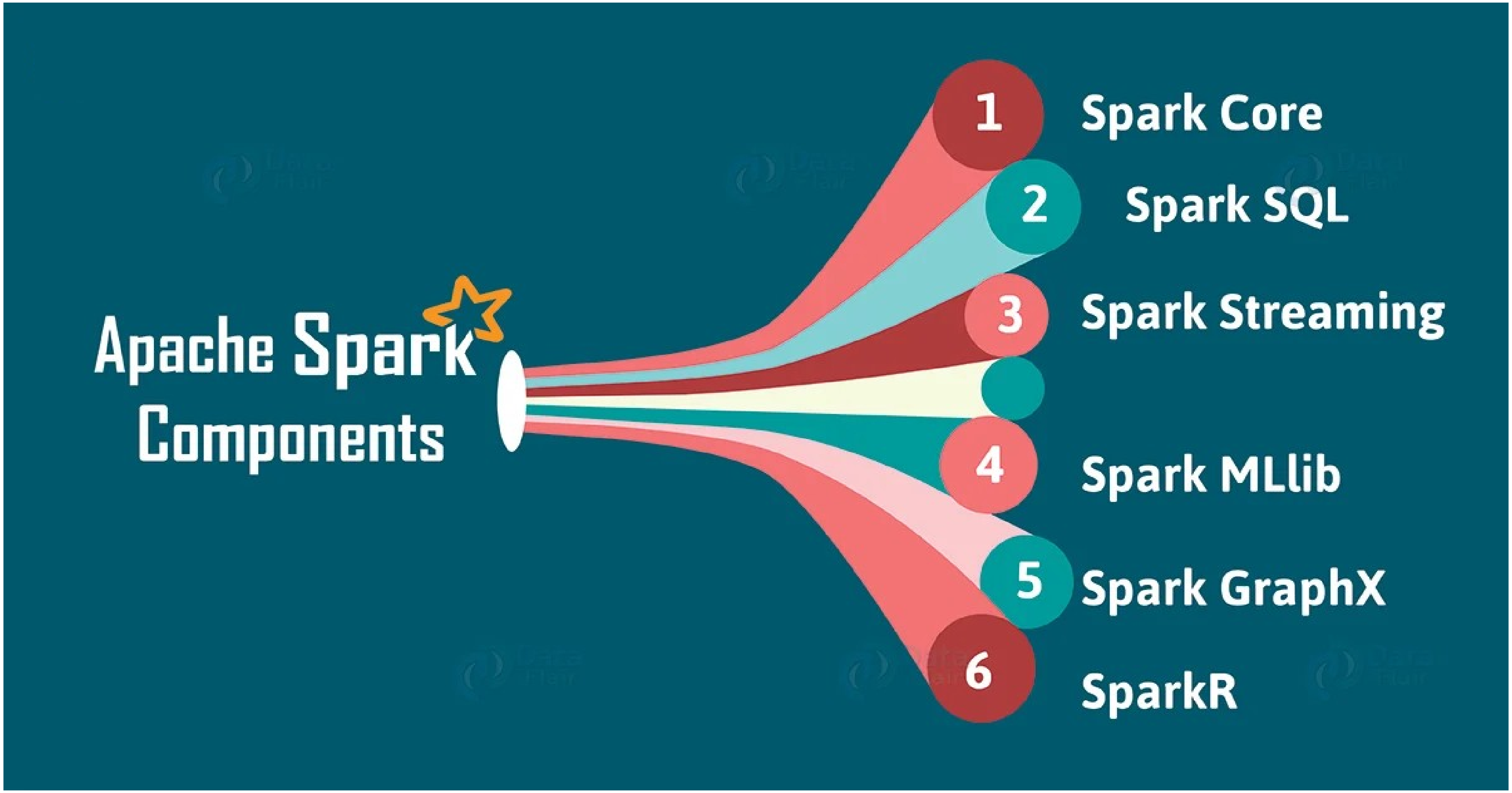
Apache Spark on Hadoop

- There are three ways of Spark deployment as explained below.
  - Standalone – Spark Standalone deployment means Spark occupies the place on top of HDFS(Hadoop Distributed File System) and space is allocated for HDFS, explicitly.
  - Here, Spark and MapReduce will run side by side to cover all spark jobs on cluster.

Apache Spark on Hadoop

- Hadoop Yarn – Hadoop Yarn deployment means, simply, spark runs on Yarn without any pre-installation or root access required. It helps to integrate Spark into Hadoop ecosystem or Hadoop stack. It allows other components to run on top of stack.
- Spark in MapReduce (SIMR) – Spark in MapReduce is used to launch spark job in addition to standalone deployment. With SIMR, user can start Spark and uses its shell without any administrative access.

Apache Spark Components



Spark Core

- Spark Core is a central point of Spark. Basically, it provides an execution platform for all the Spark applications.
- Moreover, to support a wide array of applications, Spark Provides a generalized platform.

Spark SQL

- On the top of Spark, Spark SQL enables users to run SQL/HQL queries.
- We can process structured as well as semi-structured data, by using Spark SQL.
- Moreover, it offers to run unmodified queries up to 100 times faster on existing deployments.

Spark Streaming

- Basically, across live streaming, Spark Streaming enables a powerful interactive and data analytics application.
- Moreover, the live streams are converted into micro-batches those are executed on top of spark core

Spark MLlib

- MLlib is a distributed machine learning framework above Spark because of the distributed memory-based Spark architecture.
- It is, according to benchmarks, done by the MLlib developers against the Alternating Least Squares (ALS) implementations.
- Spark MLlib is nine times as fast as the Hadoop disk-based version of Apache Mahout (before Mahout gained a Spark interface).

Spark GraphX

- GraphX is a distributed graph-processing framework on top of Spark.
- It provides an API for expressing graph computation that can model the user-defined graphs by using Pregel abstraction API.
- It also provides an optimized runtime for this abstraction.

SparkR

- Basically, to use Apache Spark from R. It is R package that gives light-weight frontend.
- Moreover, it allows data scientists to analyze large datasets. Also allows running jobs interactively on them from the R shell.
- Although, the main idea behind SparkR was to explore different techniques to integrate the usability of R with the scalability of Spark.

Uses of Spark

- Data integration: The data generated by systems are not consistent enough to combine for analysis. To fetch consistent data from systems we can use processes like Extract, transform, and load (ETL). Spark is used to reduce the cost and time required for this ETL process.
- Stream processing: It is always difficult to handle the real-time generated data such as log files. Spark is capable enough to operate streams of data and refuses potentially fraudulent operations.

Uses of Spark

- Machine learning: Machine learning approaches become more feasible and increasingly accurate due to enhancement in the volume of data. As spark is capable of storing data in memory and can run repeated queries quickly, it makes it easy to work on machine learning algorithms.
- Interactive analytics: Spark is able to generate the respond rapidly. So, instead of running pre-defined queries, we can handle the data interactively.

Thank you

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com