Kavya Gundala

22-12-2023

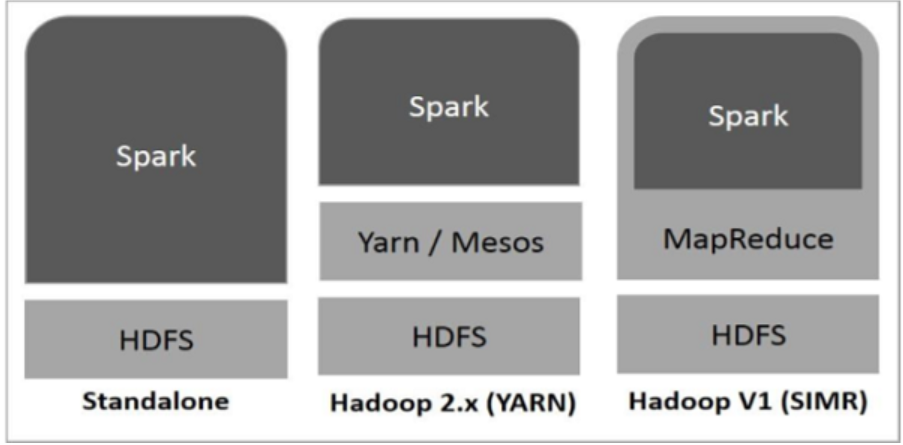
**Apache Spark:**

* 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**.
* Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. The main feature of Spark is its **in-memory cluster computing** that increases the processing speed of an application.

**Features:**

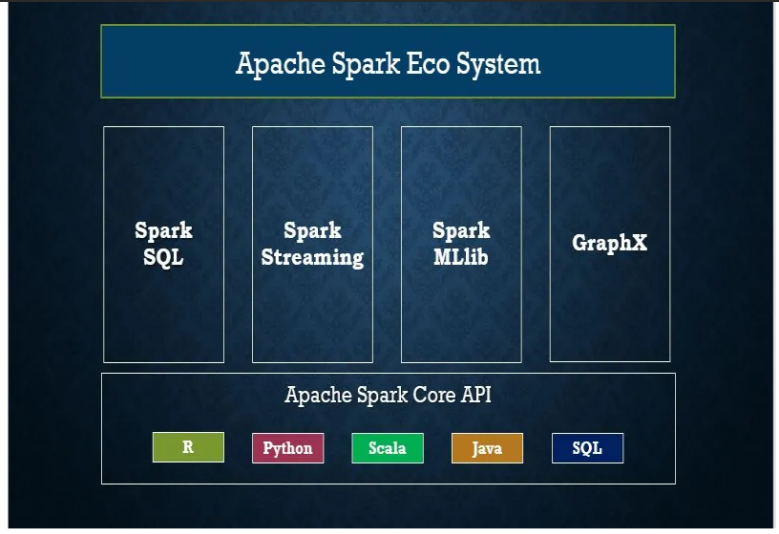
* **Speed, Supports multiple languages, Advanced analytics**

**Spark Built 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.
* **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.

****

**Components of Spark:**

* **Apache Spark Core:** Spark Core is the underlying general execution engine for spark platform that all other functionality is built upon. It provides In-Memory computing and referencing datasets in external storage systems.
* **Spark SQL:** Spark SQL is a component on top of Spark Core that introduces a new data abstraction called SchemaRDD, which provides support for structured and semi-structured data.
* **Spark Streaming:** Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD (Resilient Distributed Datasets) transformations on those mini-batches of data.
* **MLlib (Machine Learning Library):** 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).
* **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.

**Apache Spark Resilient Distributed Datasets:**

* Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects.
* There are two ways to create RDDs − **parallelizing** an existing collection in your driver program, or **referencing a dataset** in an external storage system, such as a shared file system, HDFS, HBase, or any data source offering a Hadoop Input Format.

**PySpark:**

* PySpark is very well used in the Data Science and Machine Learning community as there are many widely used data science libraries written in Python including NumPy, and TensorFlow.
* It supports processing large sets of datsets.

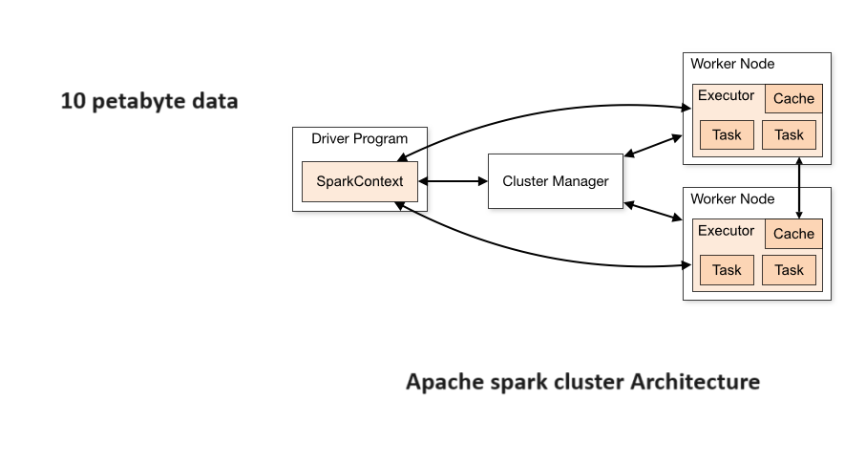
**Features of PySpark:**

* In-memory computation
* Distributed processing using parallelize
* Can be used with many cluster managers (Spark, Yarn, Mesos e.t.c)
* Fault-tolerant
* Immutable
* Lazy evaluation
* Cache & persistence
* Inbuild-optimization when using DataFrames
* Supports ANSI SQL

**Advantages:**

* PySpark is a general-purpose, in-memory, distributed processing engine that allows you to process data efficiently in a distributed fashion.
* Applications running on PySpark are 100x faster than traditional systems.
* You will get great benefits from using PySpark for data ingestion pipelines.
* Using PySpark we can process data from Hadoop HDFS, AWS S3, and many file systems.
* PySpark also is used to process real-time data using Streaming and Kafka.
* Using PySpark streaming you can also stream files from the file system and also stream from the socket.
* PySpark natively has machine learning and graph libraries.

**PySpark Architecture:**

****

**Cluster Manager Types:**

* Standalone – a simple cluster manager included with Spark that makes it easy to set up a cluster.
* Apache Mesos – Mesons is a Cluster manager that can also run Hadoop MapReduce and PySpark applications.
* Hadoop YARN – the resource manager in Hadoop 2. This is mostly used as a cluster manager.
* Kubernetes – an open-source system for automating deployment, scaling, and management of containerized applications.

**PySpark Modules & Packages:**

* PySpark RDD (pyspark.RDD)
* PySpark DataFrame and SQL (pyspark.sql)
* PySpark Streaming (pyspark.streaming)
* PySpark MLib (pyspark.ml, pyspark.mllib)
* PySpark GraphFrames (GraphFrames)
* PySpark Resource (pyspark.resource) It’s new in PySpark 3.0