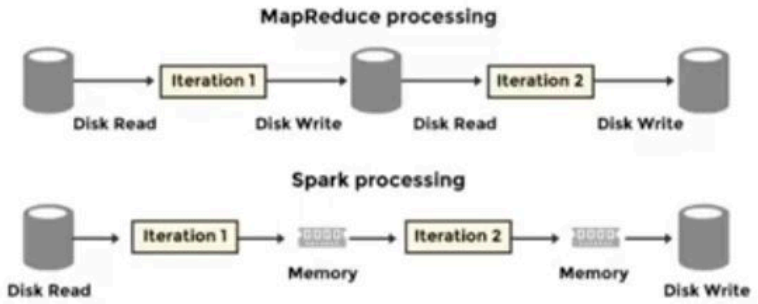
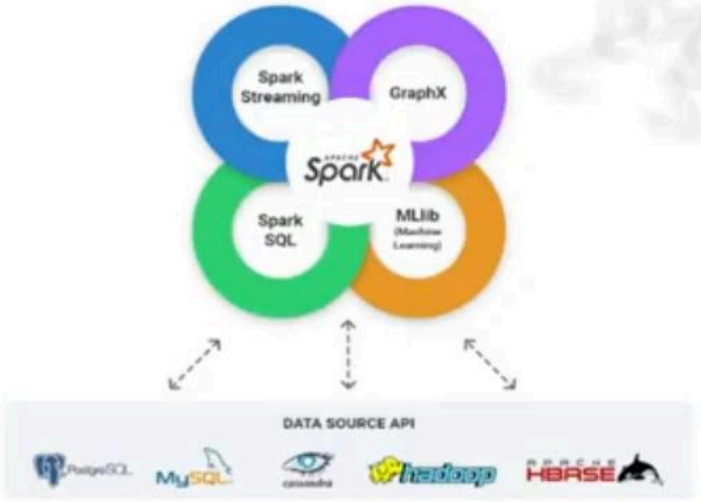
Introduction

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





Installing spark

Step 1: Verifying Java Installation

- Java installation is one of the mandatory things in installing Spark.
- $java -version

Step 2: Verifying Scala Installation

- You should Scala language to implement Spark.
- $scala -version

Step 3: Downloading Apache Spark

- Download the latest version of Spark.
- Find the Spark tar file in the download folder.

Step 4: Installing Spark

- Extracting Spark tar - $ tar xvf spark-1.3.1-bin-hadoop2.6.tgz
- Setting up the environment for Spark.

Step 4: Verifying the Spark Installation

- $spark-shell

Spark applications

Machine Learning:

- Apache Spark is equipped with a scalable Machine Learning Library called MLlib that can perform advanced analytics such as clustering

Fog Computing:

- Fog computing, also called fog networking or fogging, describes a decentralized computing structure located between the cloud and devices that produce data.

Processing Streaming Data:

- Stream processing is the processing of data in motion, or in other words, computing on data directly as it is produced or received. The majority of data are born as continuous streams: sensor events, user activity on a website, financial trades, and so on

most popular companies that are utilizing various applications of Apache Spark:

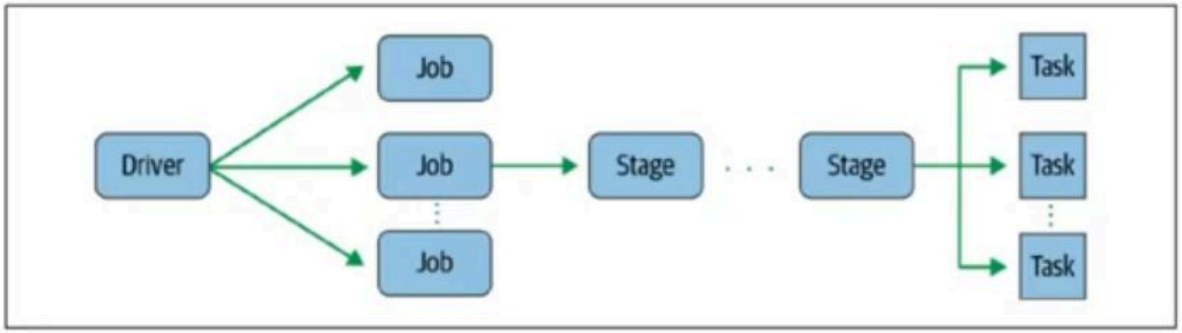
- **Uber:** Uber uses Kafka, Spark Streaming, and HDFS for building a continuous ETL pipeline.
- **Pinterest:** One of the successful web and mobile application companies, Pinterest uses Spark Streaming in order to gain deep insight into customer engagement details.

Jobs, stages and tasks

Job - A parallel computation consisting of multiple tasks that get spawned in response to a Spark action (e.g., save(), collect()).

Stage - Each job gets divided into smaller sets of tasks called stages that depend on each other. As part of the DAG nodes, stages are created based on what operations can be performed serially or in parallel. Not all Spark operations can happen in a single stage, so they may be divided into multiple stages.

Task - A single unit of work or execution that will be sent to a Spark executor.



Resilient Distributed Database(RDD)

- The fundamental abstraction in Spark is the RDD, short for Resilient Distributed Dataset.
- It is a read-only (immutable) collection of objects or records, partitioned across the cluster that can be operated on in parallel.
- A partition can be reconstructed if the hosting node experiences failure.
- RDDs are a lower-level API; the other two Spark data abstractions namely DataFrames and Datasets compile to an RDD.
- The constituent records or objects within an RDD are Java, Python, or Scala objects. Anything can be stored in any format in these objects.

Resilient: means an RDD is fault-tolerant and able to recompute missing or damaged partitions due to node failures.

Distributed: means data making up an RDD is spread across a cluster of machines.

Datasets: refer to representations of the data records we work with. External Data can be loaded using a variety of sources such as JSON file, CSV file, text file or database via JDBC.

Anatomy of a Spark job run

Spark application contains several components, all of which exists whether you are running Spark on a single machine or across a cluster of hundreds or thousands of nodes.
The components of the spark application are Driver, the Master, the Cluster Manager and the Executors.
All of the spark components including the driver, master, executor processes run in java virtual machines(JVMs). A JVM is a cross-platform runtime engine that executes the instructions compiled into java bytecode. Scala, which spark is written in, compiles into bytecode and runs on JVMs.

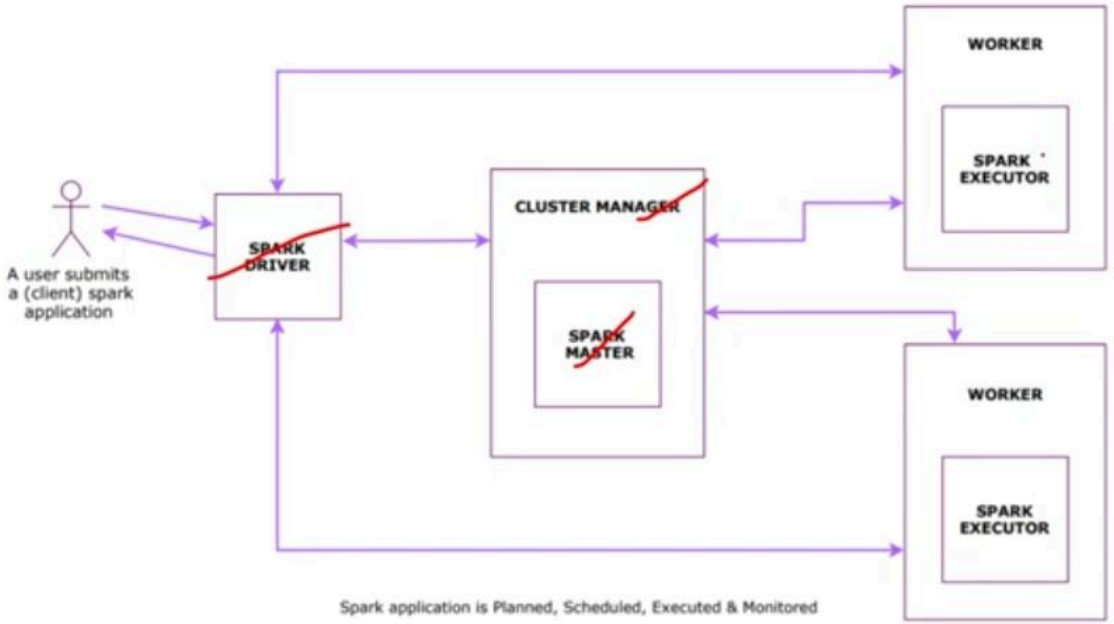
Spark Driver:

- The life of Spark programs starts and ends with the Spark Driver. The Spark driver is the process which the clients used to submit the spark program.

Spark Executors:

- Spark executors are the host processes on which the tasks from Spark DAG runs.
- Executors reserve CPU and Memory resources on slave nodes or workers in slave node
- Executors are dedicated to specific spark application and terminated when the application completes. Spark executors can run hundreds of tasks within a spark program.

Anatomy of a Spark job run



Anatomy of a Spark job run

Spark on YARN

- Apache Spark is an in-memory distributed data processing engine and YARN is a cluster management technology.
- **Spark provides two modes:** YARN client mode and YARN cluster mode.
- YARN client mode, the driver runs in the client.
- Client mode is also useful when building Spark programs, since any debugging output is immediately visible.
- In YARN cluster mode, the driver runs on the cluster in the YARN application master.
- YARN cluster mode is appropriate for production jobs.
- In YARN cluster mode, the entire application runs on the cluster.
- In YARN cluster mode, YARN will also retry the application if the application master fails.