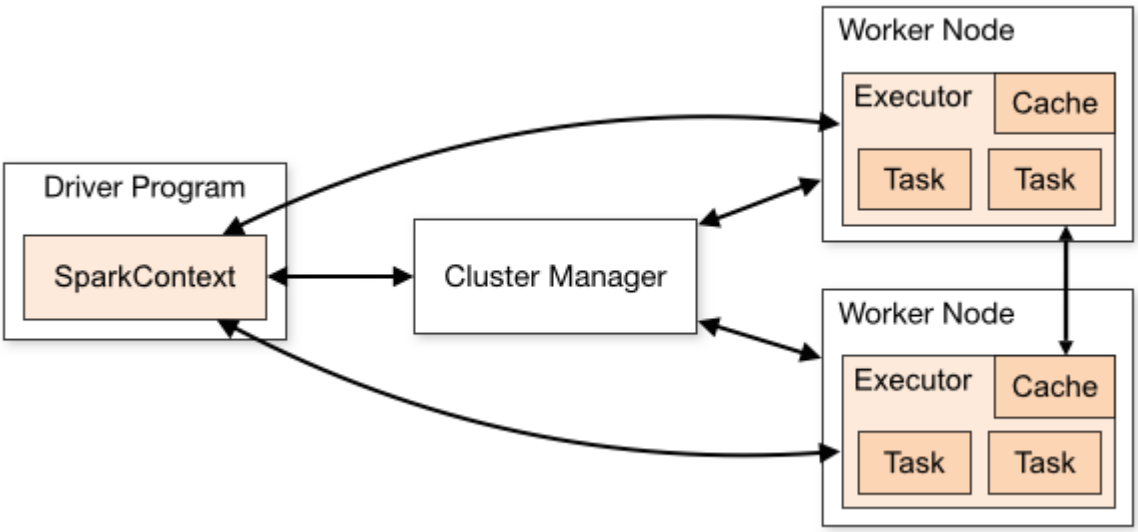
SPARK ARCHITECTURE

Apache Spark is a unified analytics engine for large-scale data processing. It is an **in-memory computation processing engine** where the data is kept in random access memory (RAM) instead of some slow disk drives and is processed in **parallel**.

Spark Architecture



Spark Execution Process Overview

Spark follows **Master-Slave architecture** where we have one master central coordinator called **Spark Driver** which communicates with multiple distributed **Worker nodes** (slaves). Each Worker node consists of one or more **Executors** which are responsible for running the task. The Driver has all the information about the Executors at all the time as the Executors register themselves with the Driver. This working combination of Driver and Workers is known

as **Spark Application** which can be launched with the help a **Cluster Manager**.

Components

Driver :

Driver is a Java process. The main() method of our program runs in the Driver process. It executes the user code and creates a SparkSession or SparkContext. The SparkSession is responsible to create DataFrame, DataSet, RDD, execute SQL, perform Transformation & Action, etc. It determines the total number of Tasks to be run by the Executors in the Worker node by checking the Lineage. Once the Physical Plan is generated, the driver with the help of Cluster Manager schedules the execution of the tasks (transformation and action) to the Executors in the Worker nodes. It keeps track of the data (in the form of metadata) which was cached in Executor's (worker's) memory.

Executor :

Executors resides in the Worker node and they are launched at the start of a Spark Application in coordination with the Cluster Manager. They are dynamically launched and removed by the Driver as per the requirements. Each Executor runs individual task and return result to the driver. During this time, it can cache the data within the Worker node.

Cluster Manager :

Spark is dependent on the Cluster Manager to launch the Executors and also the Driver (in Cluster mode). Spark can be run with any of the Cluster Manager like Spark Standalone Mode, YARN, Apache Mesos and Kubernetes. Spark provides a script named "**spark-submit**" which helps us to connect with a different kind of Cluster Manager and it controls the number of resources the application is going to get i.e. it decides the number of Executors to be launched, how much CPU and memory should be allocated for each Executor, etc.

STEPS OFQUERY EXECUTION OFSPARK

1. User submits a job using "spark-submit".
2. "spark-submit" will in-turn launch the Driver which will execute the main() method of our code.
3. Driver contacts the cluster manager and requests for resources to launch the Executors.
4. The cluster manager launches the Executors on behalf of the Driver.
5. Once the Executors are launched, they establish a direct connection with the Driver.
6. The driver determines the total number of Tasks by checking the Lineage.
7. The driver creates the Logical and Physical Plan.

8. Once the Physical Plan is generated, Spark allocates the Tasks to the Executors.
9. Task runs on Executor and each Task upon completion returns the result to the Driver.
10. Finally, when all Task is completed, the main() method running in the Driver exits, i.e. main() method invokes sparkContext.stop().
11. Finally, Spark releases all the resources from the Cluster Manager.