Apache Spark

- Apache spark is general purpose in memory compute engine.
- Hadoop provides storage,computation ,resource management in terms of HDFS,mapreduce & yarn.
- Spark is a plug & play compute engine
- Spark computes the data in memory whereas mapreduce in disk
- Spark latency is less due to low disk read and write operations

Storage system – HDFS,Amazon S3,local file system

Any resource manager like Yarn,mesos,kubernetes

Spark is an alternative to mapreduce

10x faster than mapreduce

Spark is an open source distributed computing engine. We use it for processing and analyzing a large amount of data. Likewise, hadoop mapreduce, it also works to distribute data across the cluster. It helps to process data in parallel.

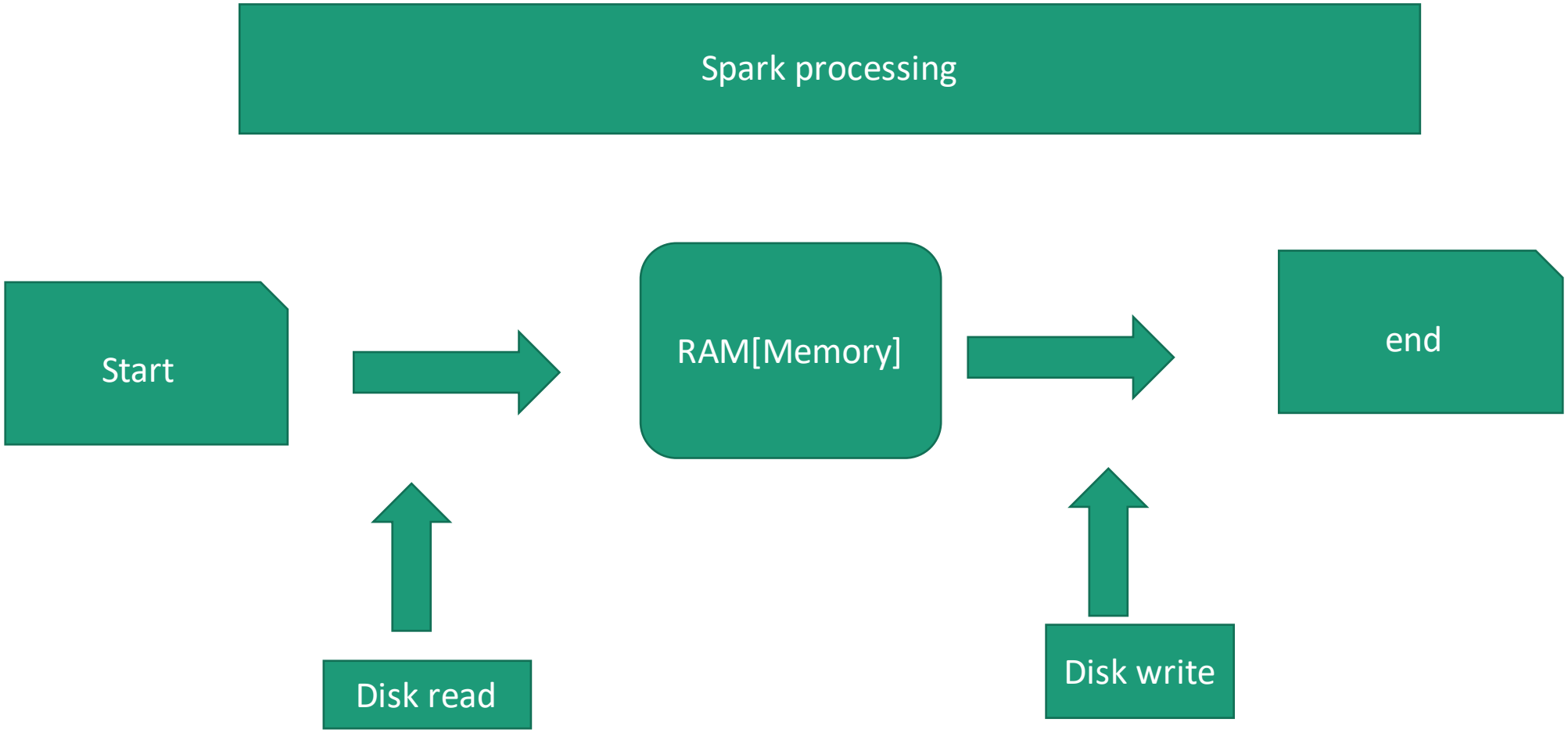

Spark processing
Start
RAM[Memory]
end
Disk read
Disk write

Fig: Features of Spark

Spark

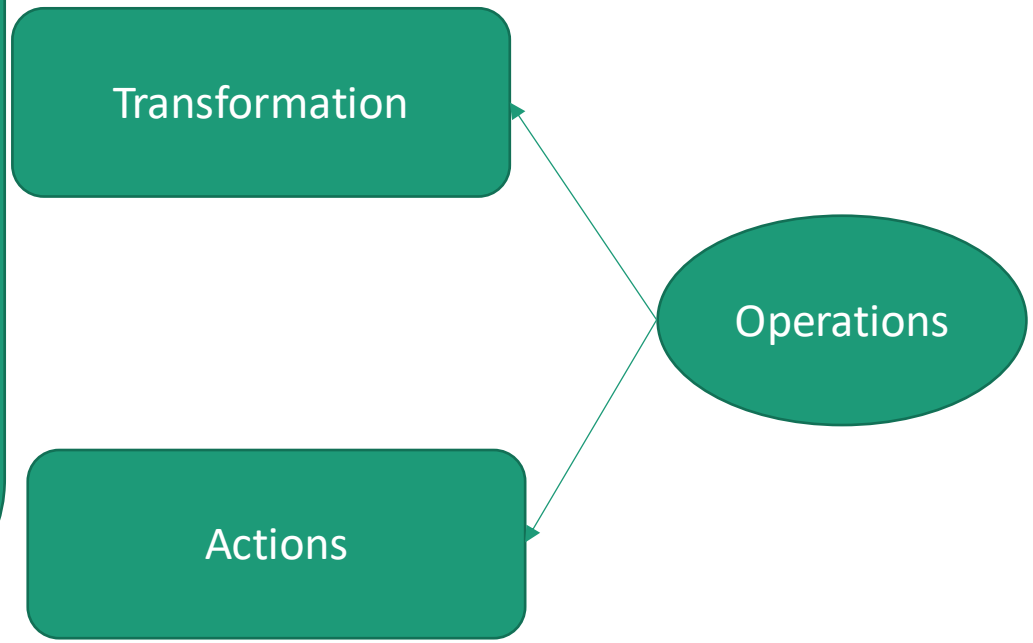
- Machine learning
- Data cleaning
- Streaming
- Hive support

RDD – Resilent distributed dataset

Dataframe

Dataset

Val RDD1=sc.textFile("abc.txt") ----------→Transformation 1
Val RDD2=RDD1.map() -------→Transformation 2
Val RDD3=RDD2.filter() -→Transformation 3
RDD3.count() →Action



Note : Transformations are lazy until an action is being called to execute. DAG – Generate when we compute a spark statement

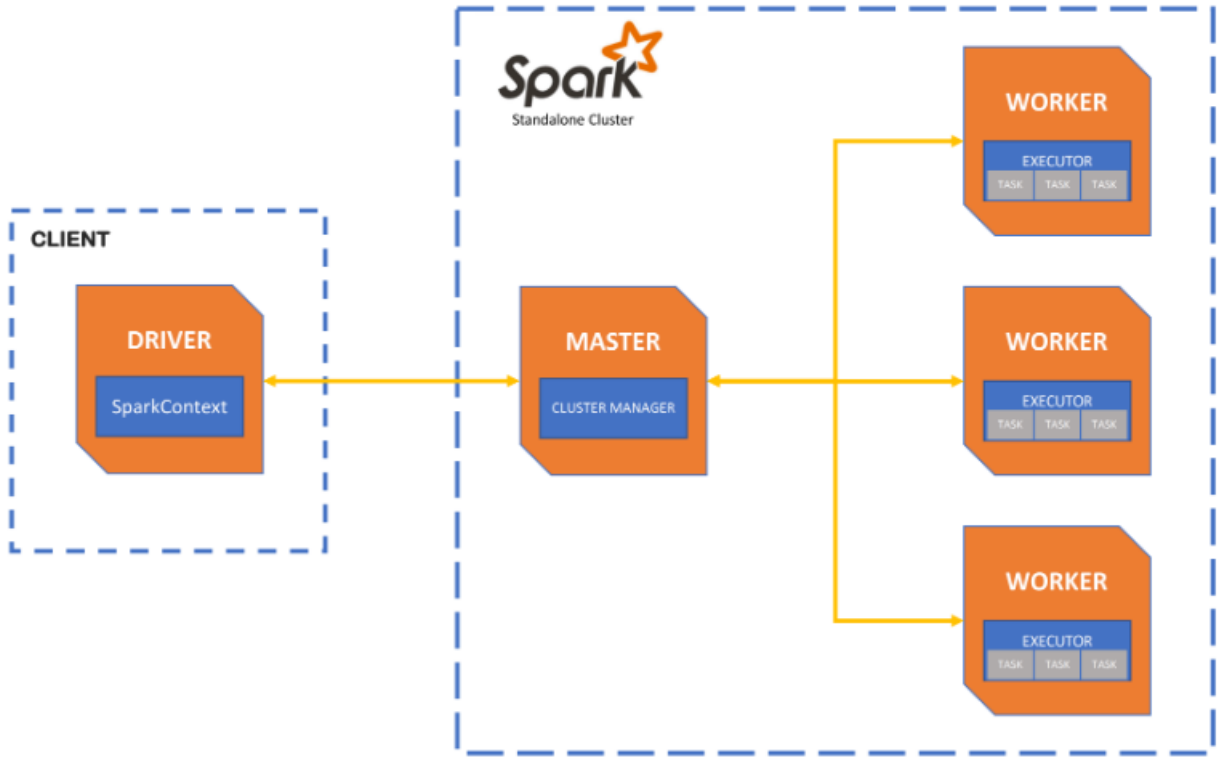
text file →map→filter

The Resilient Distributed Datasets are the group of data items that can be stored in-memory on worker nodes. Here,

•Resilient: Restore the data on failure.

•Distributed: Data is distributed among different nodes.

•Dataset: Group of data.



Apache Spark standalone cluster architecture in client mode

The Driver Program is a process that runs the main() function of the application and creates the **SparkContext** object. The purpose of **SparkContext** is to coordinate the spark applications, running as independent sets of processes on a cluster.

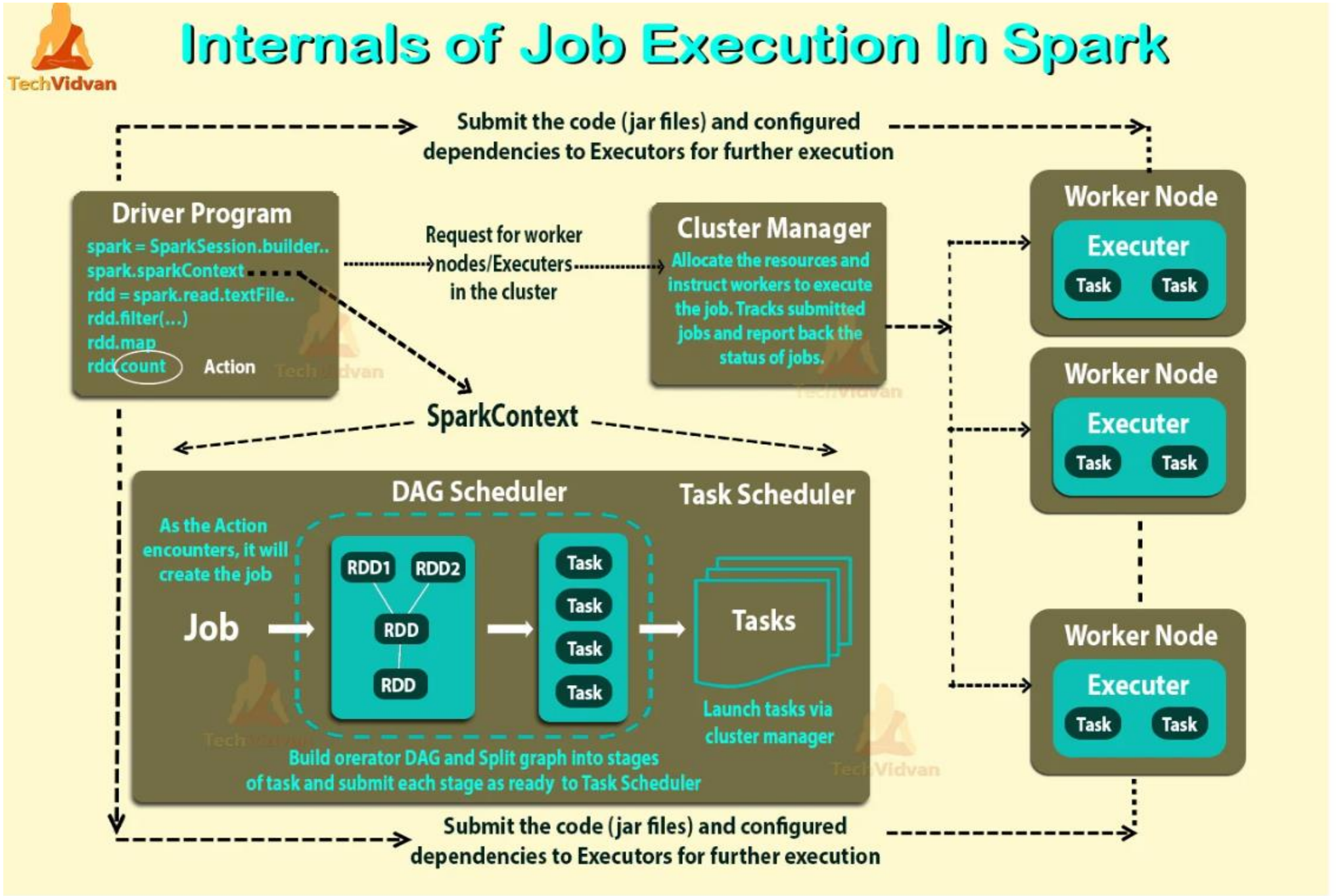
To run on a cluster, the **SparkContext** connects to a different type of cluster managers and then perform the following tasks: -

•It acquires executors on nodes in the cluster.

•Then, it sends your application code to the executors. Here, the application code can be defined by JAR or Python files passed to the SparkContext.

•At last, the SparkContext sends tasks to the executors to run.

Internals of Job Execution In Spark



Worker Node

- The worker node is a slave node
- Its role is to run the application code in the cluster.

Executor

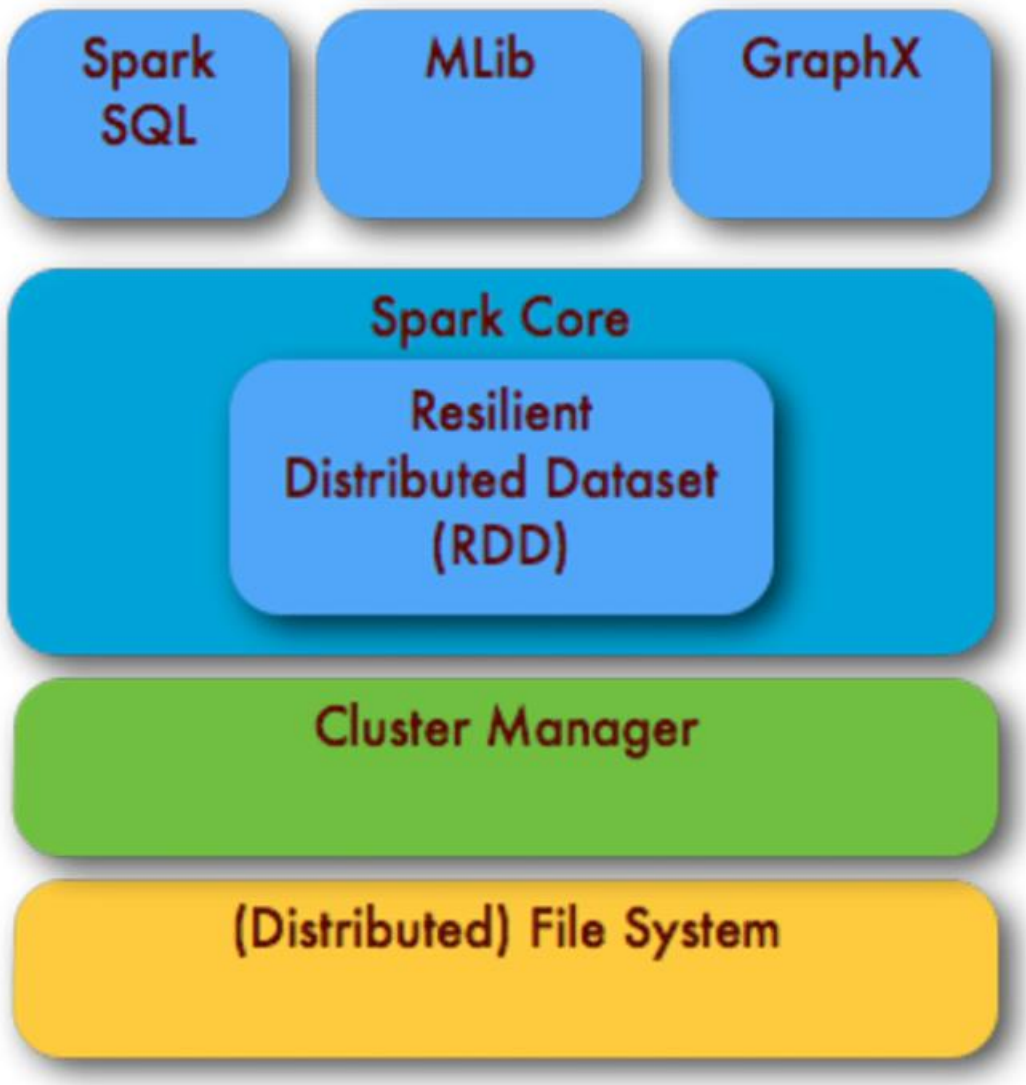
- An executor is a process launched for an application on a worker node.
- It runs tasks and keeps data in memory or disk storage across them.
- It read and write data to the external sources.
- Every application contains its executor.

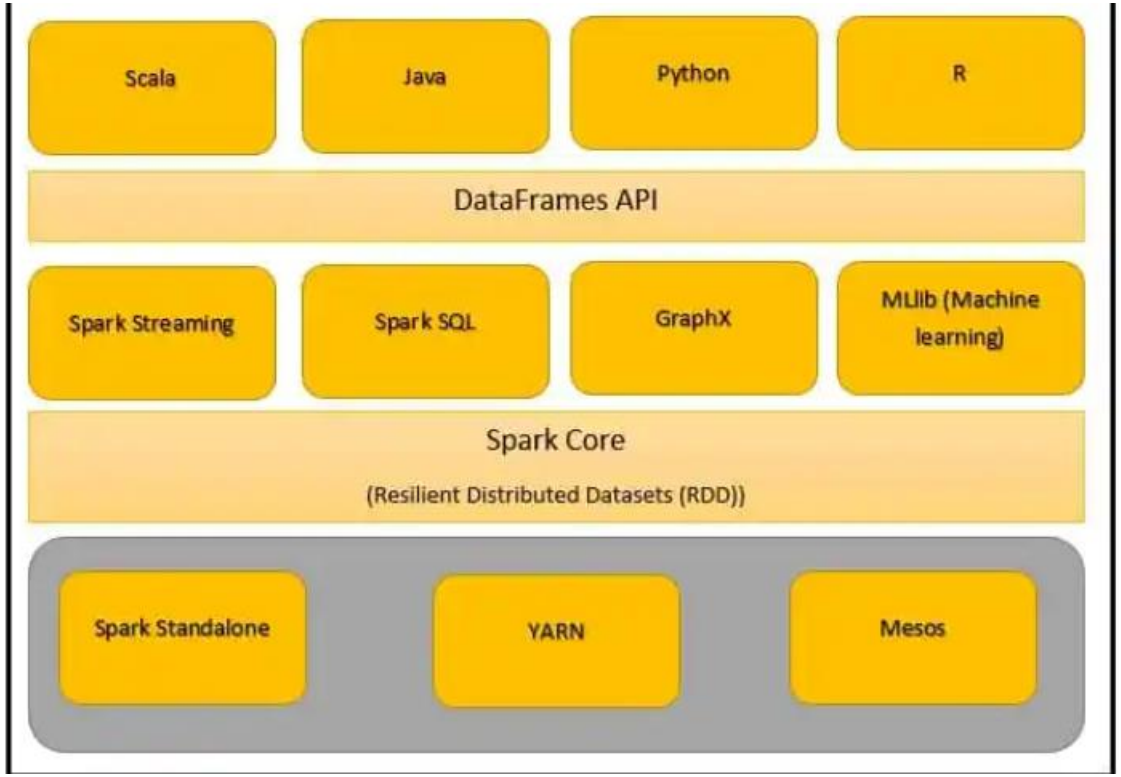
SparkContext is the main entry point to spark core. It allows us to access further functionalities of spark. This helps to establish a connection to spark execution environment. It provides access to spark cluster even with a resource manager. Sparkcontext act as master of spark application.

- Directed- Graph which is directly connected from one node to another. This creates a sequence.
- Acyclic – It defines that there is no cycle or loop available.
- Graph – It is a combination of *vertices and edges*, with all the connections in a sequence

We can call it a sequence of computations, performed on data. In this graph, edge refers to transformation on top of data. while vertices refer to an RDD partition.

This helps to eliminate the Hadoop mapreduce multistage execution model. It also provides efficient performance over Hadoop.


Spark SQL
MLib
GraphX
Spark Core
Resilient Distributed Dataset (RDD)
Cluster Manager
(Distributed) File System


Scala
Java
Python
R
DataFrames API
Spark Streaming
Spark SQL
GraphX
MLlib (Machine learning)
Spark Core
(Resilient Distributed Datasets (RDD))
Spark Standalone
YARN
Mesos

Thank you !!