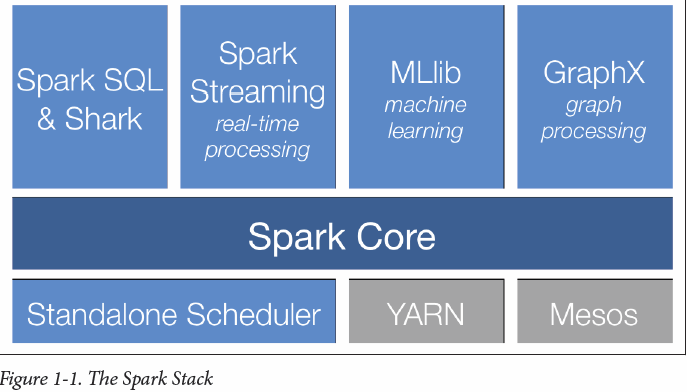
# Spark Framework stack

The Spark project contains multiple closely-integrated components. At its core, Spark is a “computational engine” that is responsible for scheduling, distributing, and monitoring applications consisting of many computational tasks across many worker machines, or a computing cluster



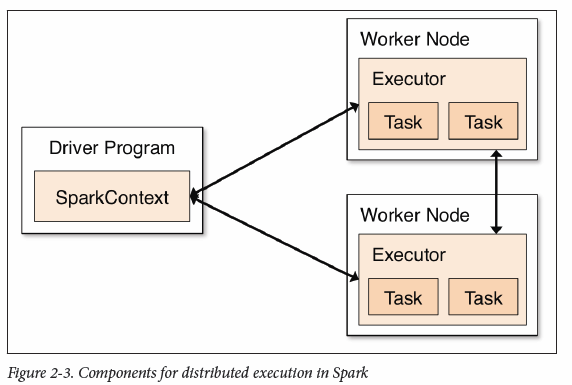
Spark can create distributed datasets from any file stored in the Hadoop distributed file system (HDFS) or other storage systems supported by Hadoop (including your local file system, Amazon S3, Cassandra, Hive, HBase, etc). Spark supports text files, SequenceFiles, Avro, Parquet, and any other Hadoop InputFormat.

# Spark Concepts

## Driver Program

At a high level, every Spark application consists of a driver program that launches various parallel operations on a cluster. The driver program contains your application’s main function and defines distributed datasets on the cluster, then applies operations to them.

Driver programs access Spark through a SparkContext object, which represents a connection to a computing cluster



## RDD

An RDD in Spark is simply an immutable distributed collection of objects. Each RDD is split into multiple partitions, which may be computed on different nodes of the cluster. RDDs offer two types of operations: transformations and actions. Although you can define new RDDs any time, Spark only computes them in a lazy fashion, the first time they are used in an action

Machine generated alternative text: Spark Driver
Cluster Worker
Cluster
Worker
Executor
Cluster Master
Mesos, YARN, or
Standalone
Executor
Cluster Worker
Executor