PROFILE ASSISTANT FOR DATA LAKE (PADL)

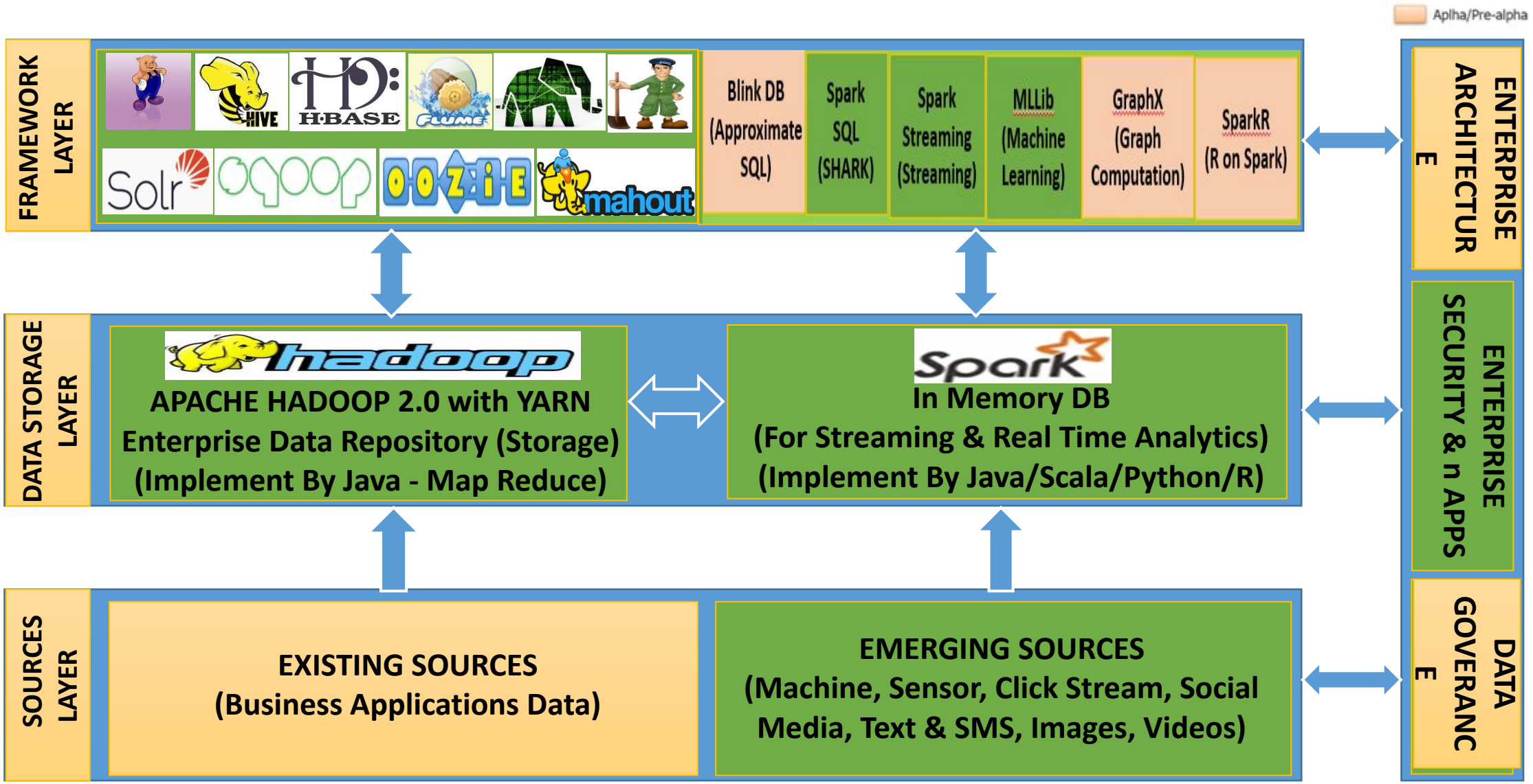
For Real Time Analytics

Presented By
Asit Piri

Presentation on Apache Spark

FAST DATA ARCHITECTURE

(For In Memory DB & Real Time Analytics By using Apache Hadoop & Spark)



What is Spark?

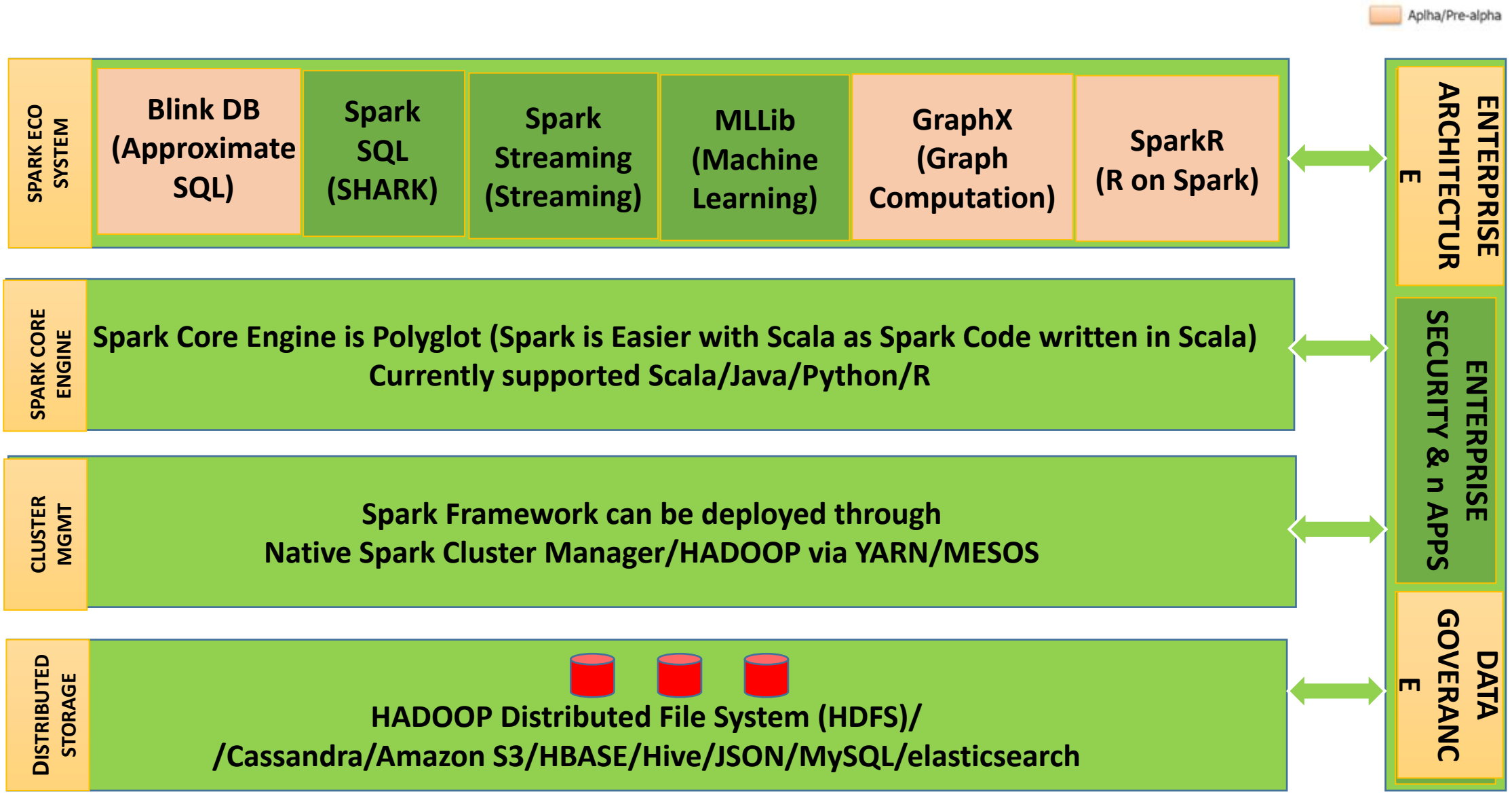
Apache Spark is a powerful open source parallel processing engine for big data built around speed, ease of use and sophisticated analytics. It was originally developed in 2009 in UC Berkeley's AMP Lab and open sourced in 2010.

Apache Spark can process data from variety of data sources including HDFS, no-sql databases such Hbase and relational data stores such as hive. It can process data is in memory as well as data on disk and take advantage of data locality.

It has interfaces to Scala, Java as well as Python.

Spark which is an improvement over map reduce paradigm. Spark extended the map reduce paradigm by adding more functionality than map and reduce and also made many performance improvements to make it ten to hundred times faster than map reduce. It is considered as the future map reduce and is may supposed to eventually replace map reduce.

SPARK FRAMEWORK / ARCHITECTURE



Spark Architecture

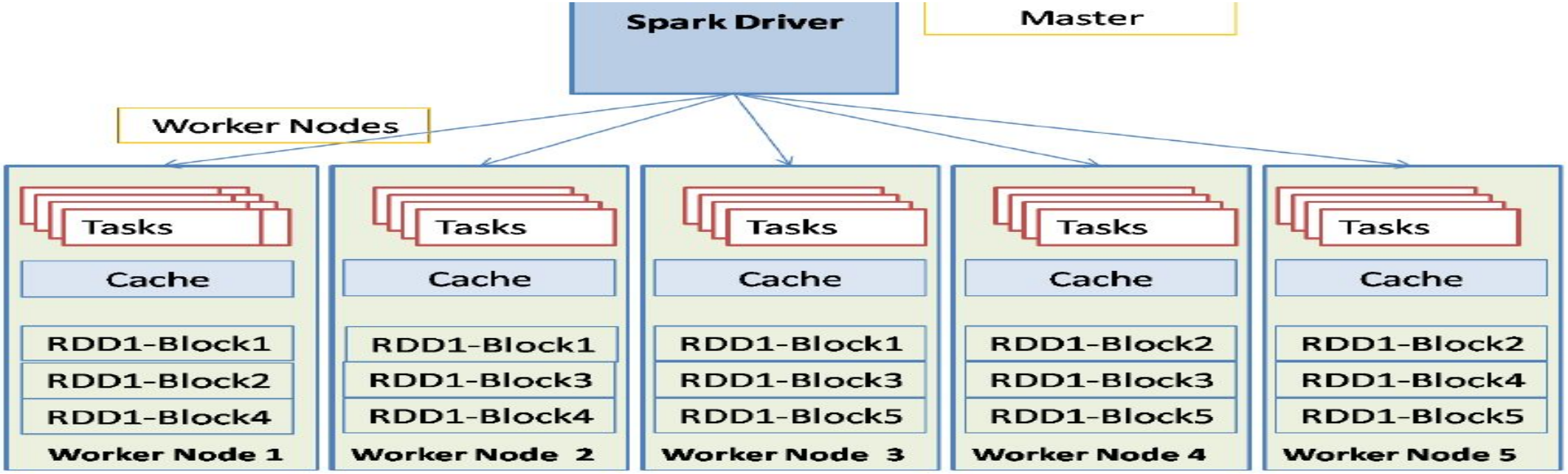
Spark works on the concept of RDDs (Resilient Distributed Datasets). RDDs are data that are distributed over a cluster of machines and are resilient. Resilient means that they can be rebuilt on failure based on the information stored.

Spark works by applying a series of transformations to the RDDs. RDDs can be initially created by loading data from local storage or from HDFS. Then a series of transformations are applied to RDDs to get the resultant RDD.

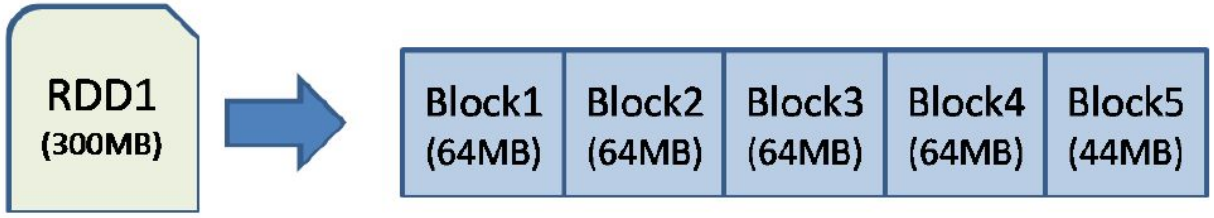
Spark architecture is similar to Hadoop in that it works with cluster of machines in a master-worker architecture.

RDDs form the backbone of Spark. These are immutable partitioned collection of objects on which a series of transformations can be applied to derive at the results. They are distributed across a cluster. You can use the existing HDFS cluster to store the RDDs. RDDs are stored in cache by default and are stored to disk only if persistence is required.

Spark Architecture Components



In the above diagram, spark is storing RDD called RDD1 which is 300 MB in size. The RDD is stored using HDFS with the default replication factor of 3 and block size of 64 MB.



What is RDD?

Resilient Distributed Data Sets (RDDs) are the fundamental unit of data and core concept of Spark Framework. RDD is fault tolerant and immutable.

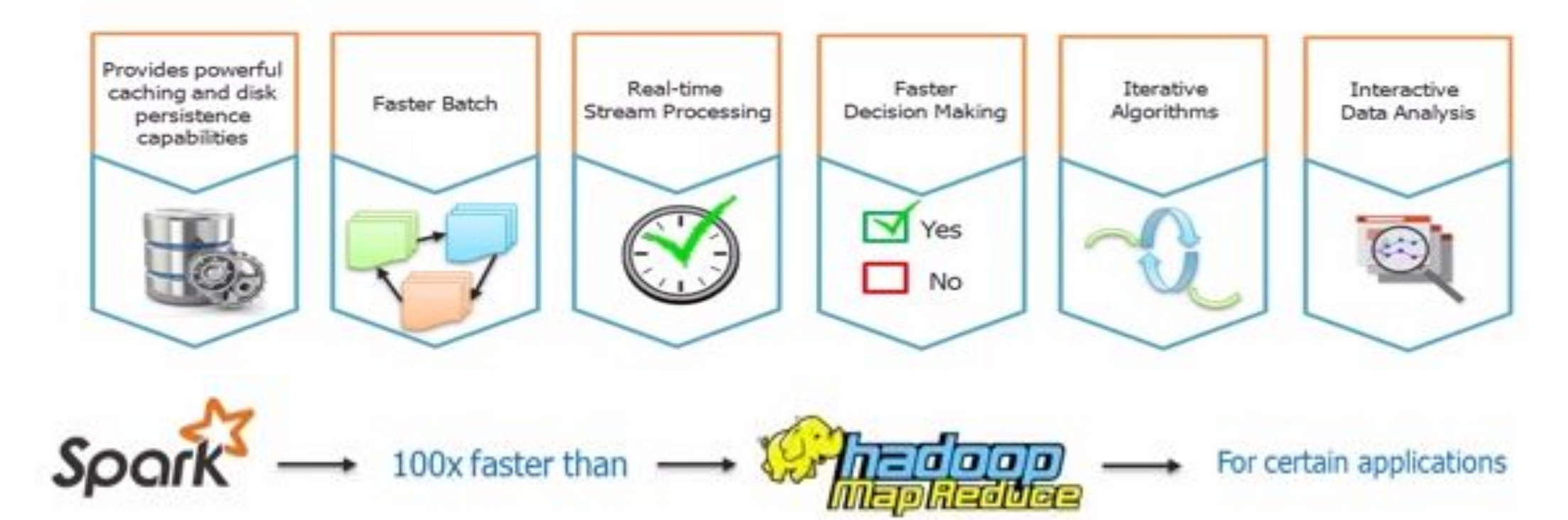
- Resilient: If data in memory is lost, can be recreated by using directed acyclic graph.
- Distributed: Stored in memory across the cluster.
- Dataset: Initial data can be come from a file or created programmatically.

RDDs can be store any types of data Primitive Types (Integer, Character, Boolean etc.) as well as Files (Text files, Sequence Files etc.).

RDD supports two types of operations Transformation (return a new RDD) & Action (evaluate and return a new value).

- Transformation Functions: Map, Filter, pipe, coalesce, flatMap, groupByKey, reduceByKey, agreegateByKey.
- Action Operation: Reduce, Collect, count, First, Take, countByKey and foreach

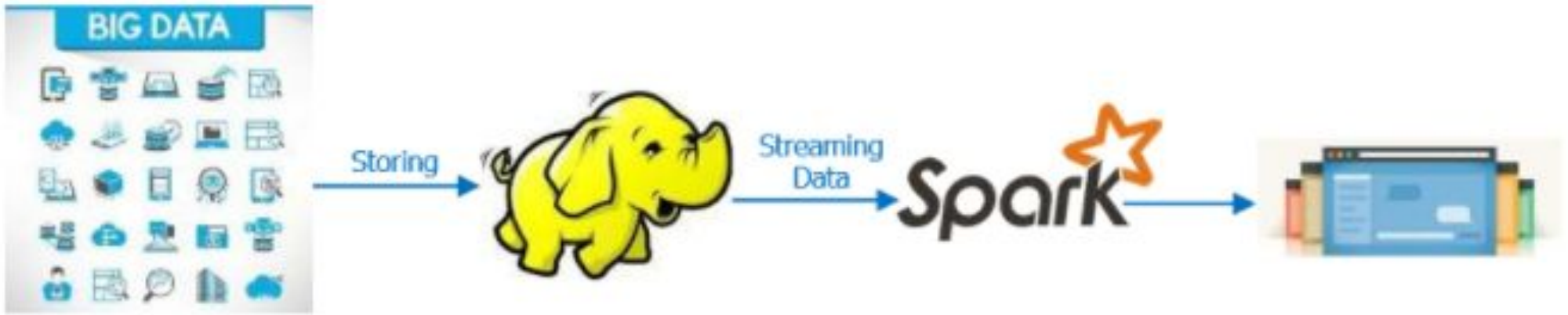
Apache Spark Features



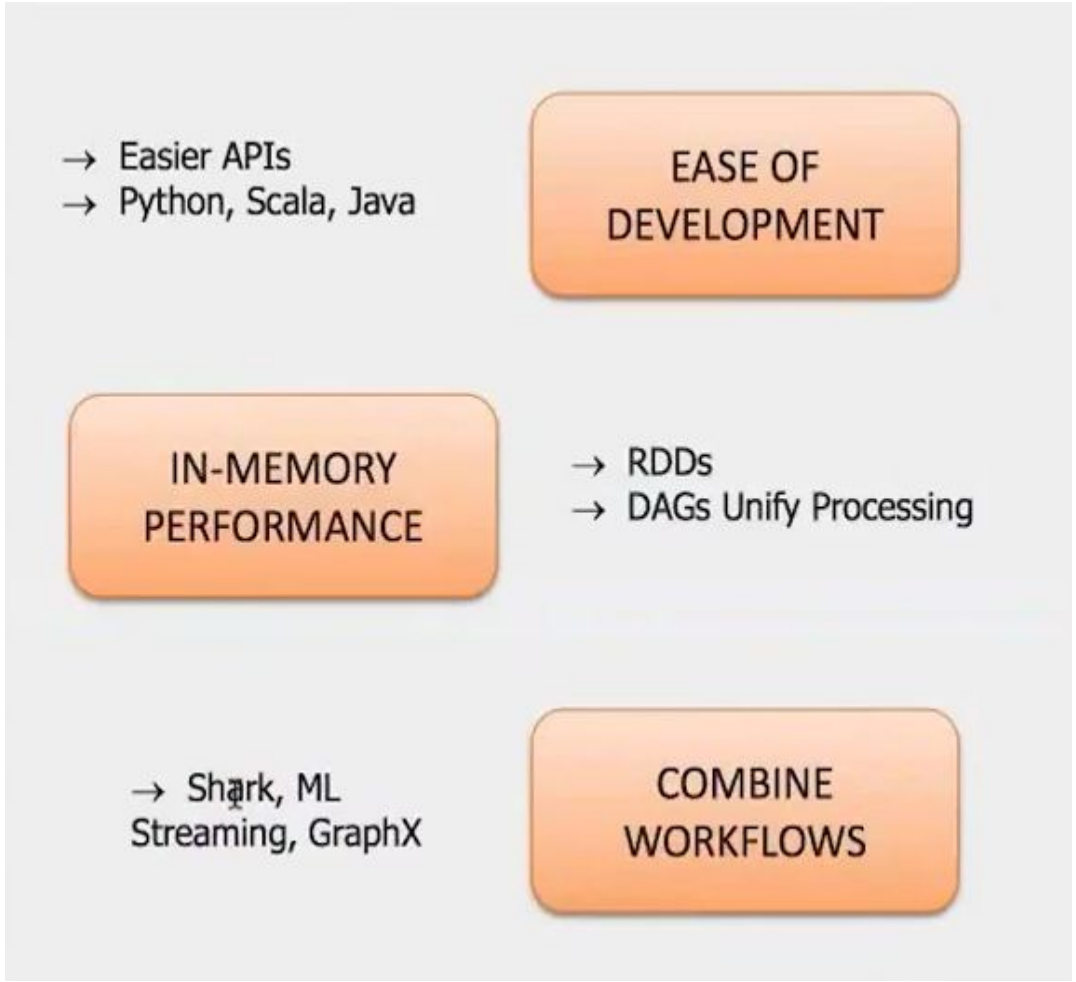
Hadoop Map Reduce –> Parallel Process but Read or Write to Disk.

Spark –> Parallel Process but Read or Write to Memory (cache) not to disk and spill out the data to disk as and when memory is reach to threshold. Suitable for window operation.

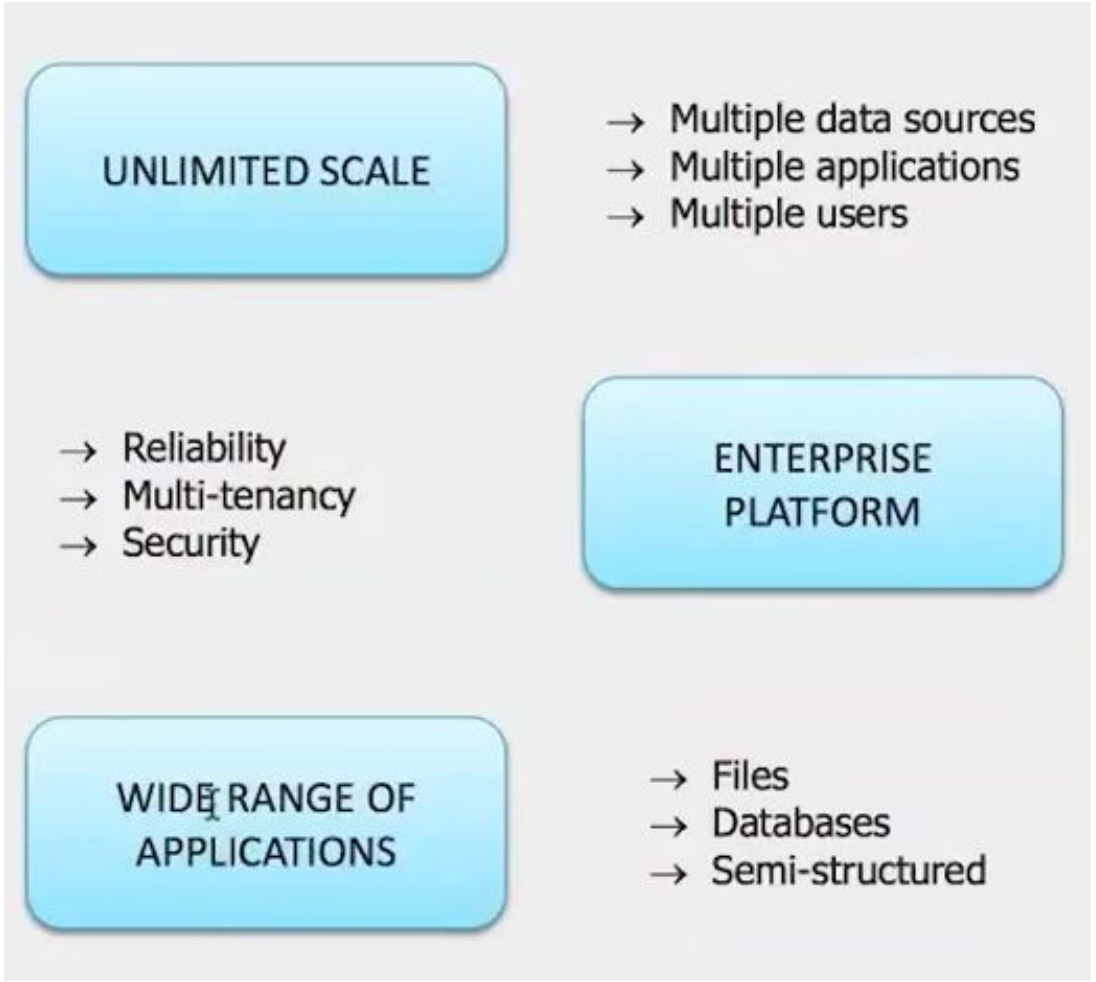
Near Real Time Analytics – Accepted Way



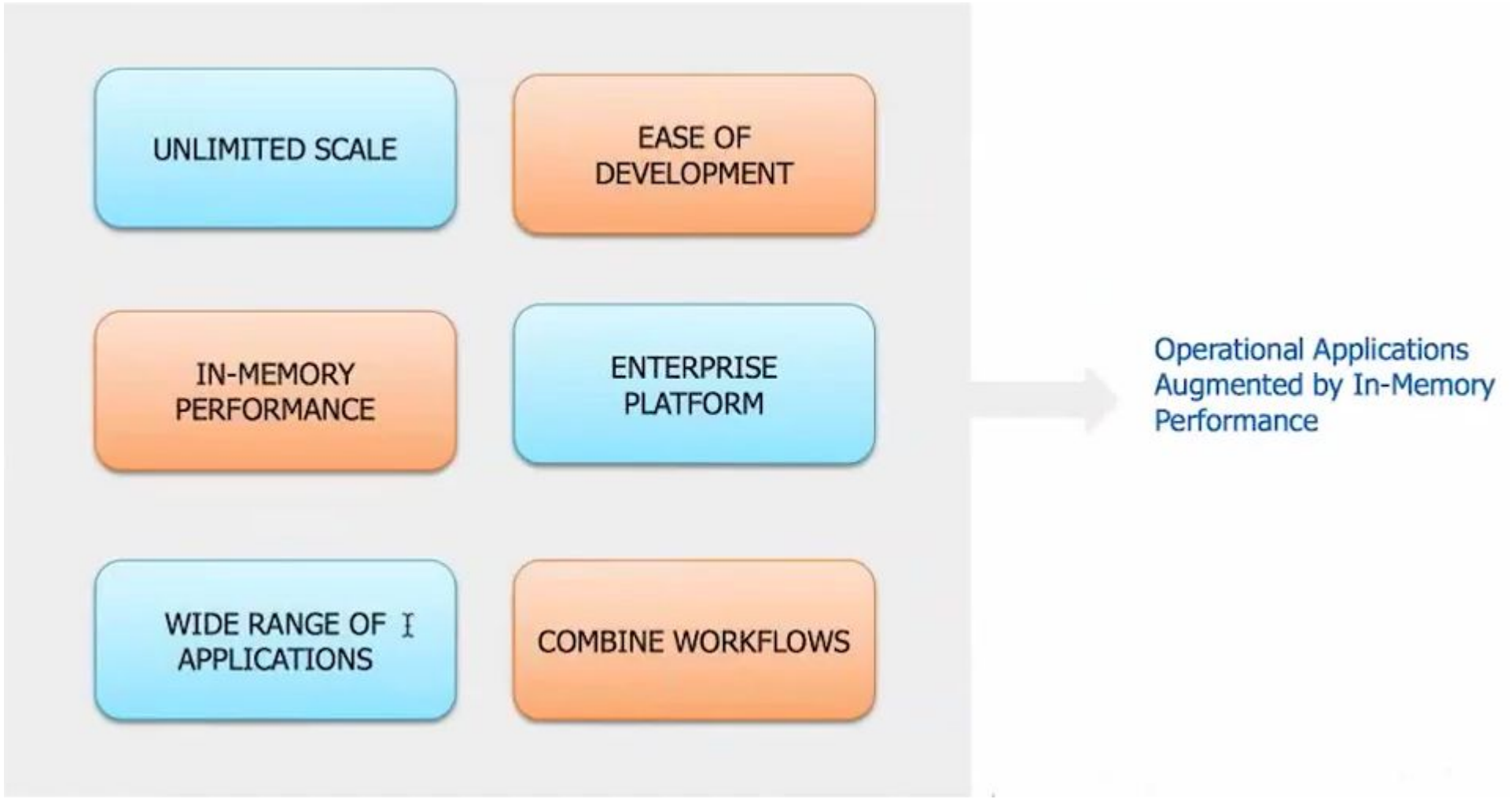
Spark: Advantages



Hadoop: Advantages



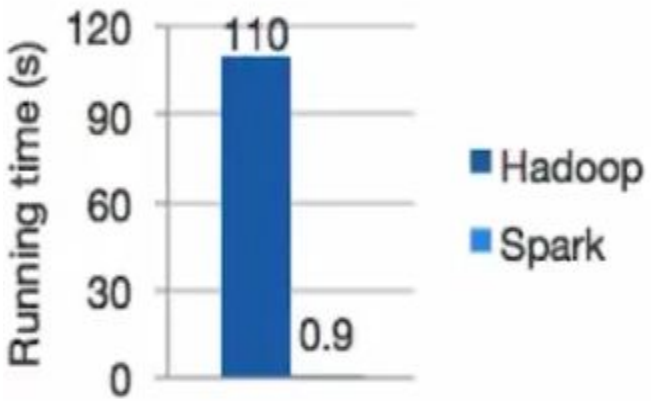
Spark + Hadoop Collaborative Advantages



Why Spark?

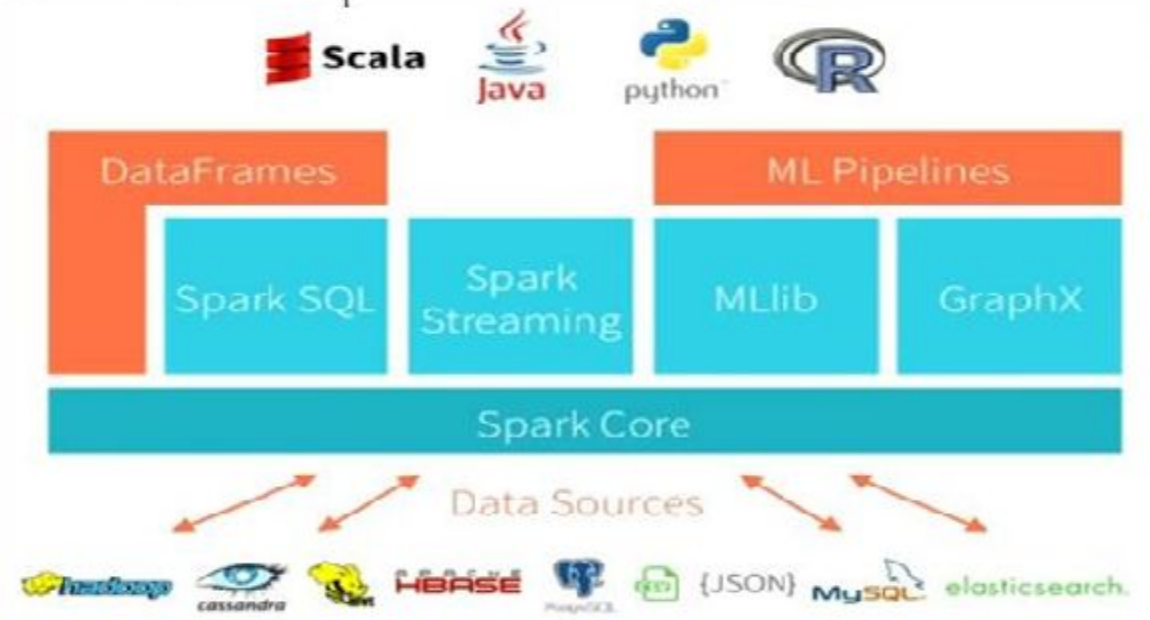
Speed

Run programs up to 100x faster than Hadoop Map Reduce in memory, or 10x faster on disk.



Ease of Use

Supports different languages for developing applications using Spark



Generality

Combine SQL, streaming, and complex analytics into one platform

Runs Everywhere

Spark runs on Hadoop, Mesos, standalone, or in the cloud.

Limitation of Spark

- Apache Spark executes the jobs in micro batches that are very short say approximately 5 seconds or less than that. Apache Spark has over the time been successful in providing more stability when compared to the real time stream oriented Hadoop Framework.
- In case if the data size is greater than memory then under such circumstances Apache Spark will not be able to leverage its cache and there is much probability that it will be far slower than the batch processing of Map Reduce.
- High Cost, Memory (RAM) are always going to be more expensive than HDDs. Maximum RAM on a system is much smaller than Maximum HDD. So a Spark Cluster will require more no. of nodes to derive the low latency advantage of Spark.

Apache Spark VS Apache Storm

Spark: Spark is an in-memory data-processing platform that is compatible with Hadoop data sources but runs much faster than Hadoop Map Reduce. It's well suited for machine learning jobs, as well as interactive data queries, and is easier for many developers because it includes APIs in Scala, Python and Java.

Storm: Apache Storm is an open source distributed real-time computation system. Storm makes it easy to process streams of data, doing for real-time processing what Hadoop did for batch processing.

One key difference between these two technologies is that Spark performs Data-Parallel computations while Storm performs Task-Parallel computations. Apache Storm may be considered as a special use case of Apache Spark.