SPARK

Outline

- Introduction to Spark
- MR vs Spark
- Spark Components
- RDD Overview
- Spark Architecture

What is Spark?

Apache :

Spark™ is a fast and general engine for large-scale data processing.

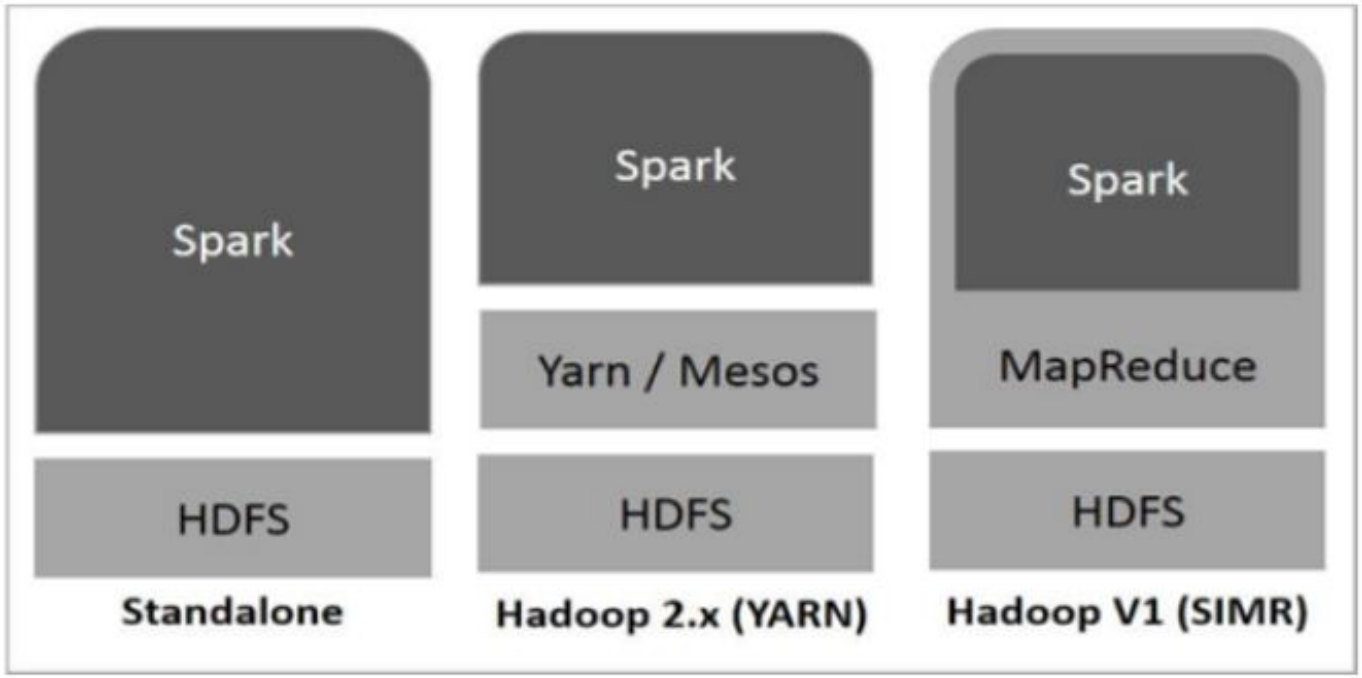
Datairicks:

Spark™ is a powerful open source processing engine built around speed, ease of use, and sophisticated analytics.

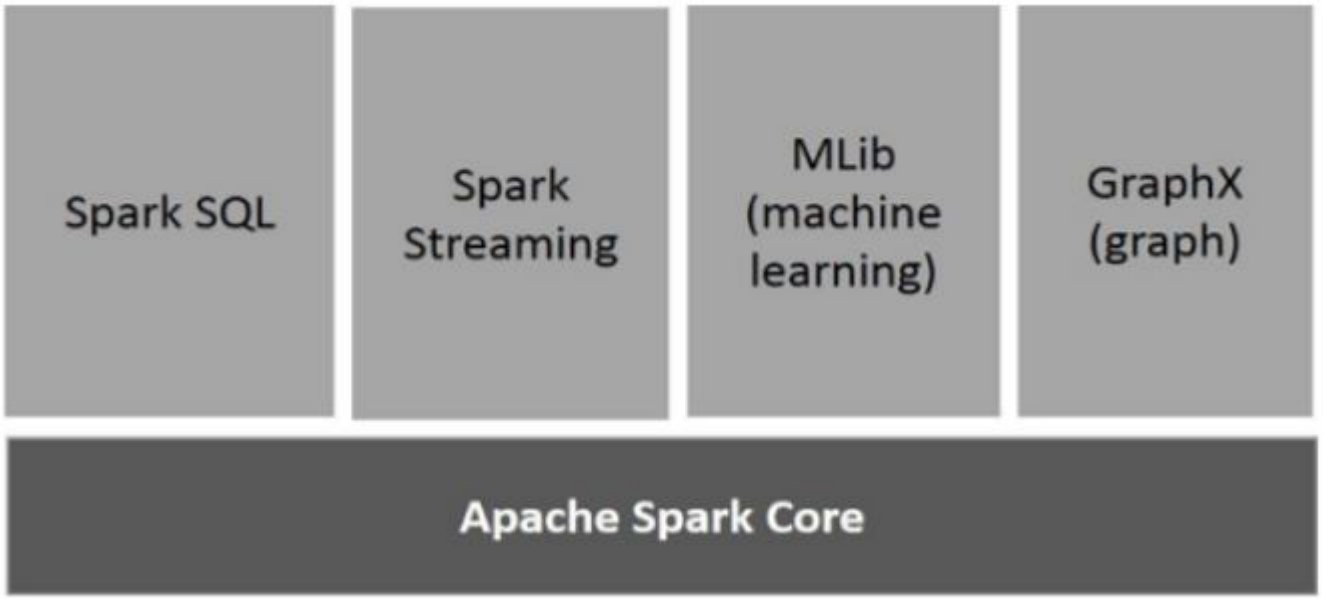
Spark is open source distributed computing engine for data processing and data analytics.

- **It was originally developed at UC Berkeley in 2009**

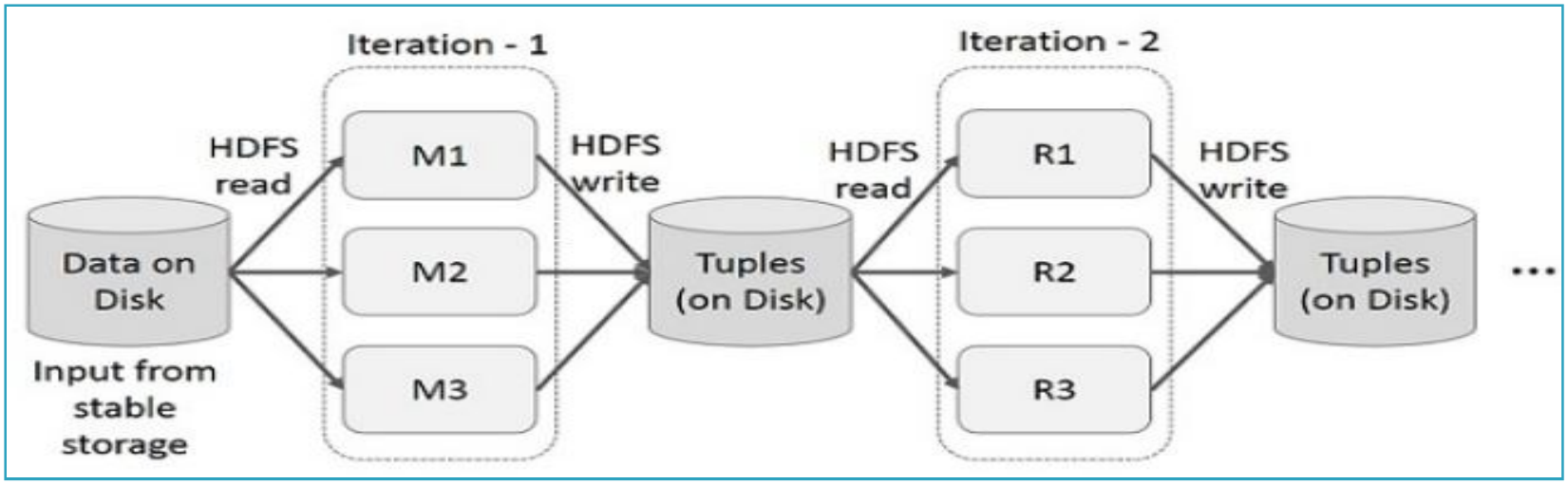
Spark Built on Hadoop

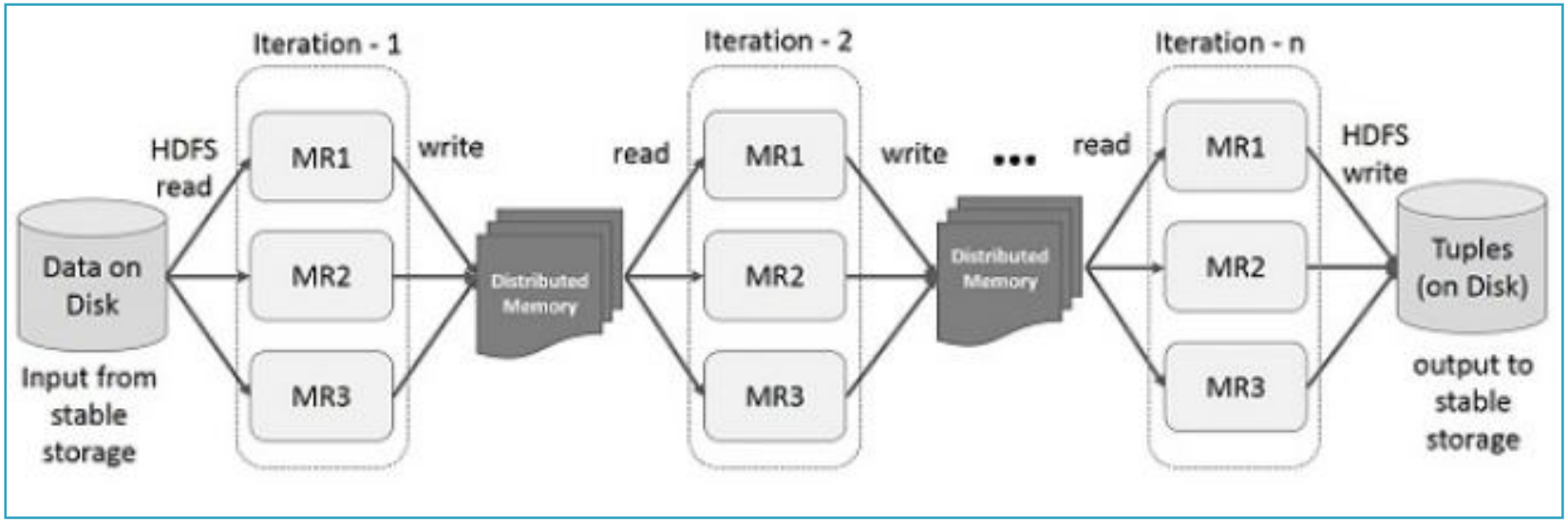


Spark Components

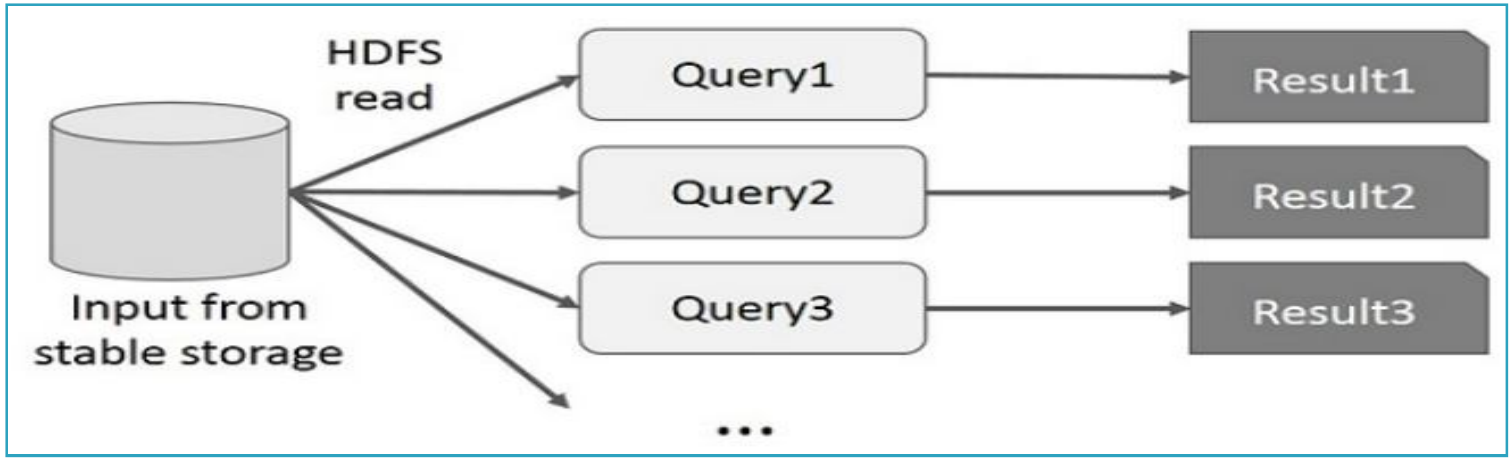


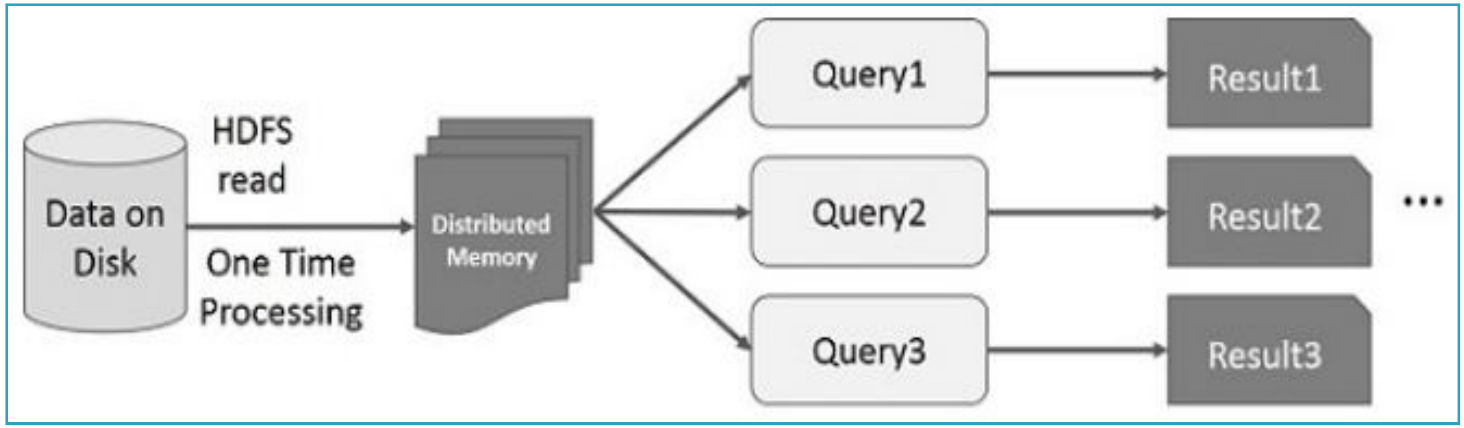
Spark vs MR – Iterative operation use case



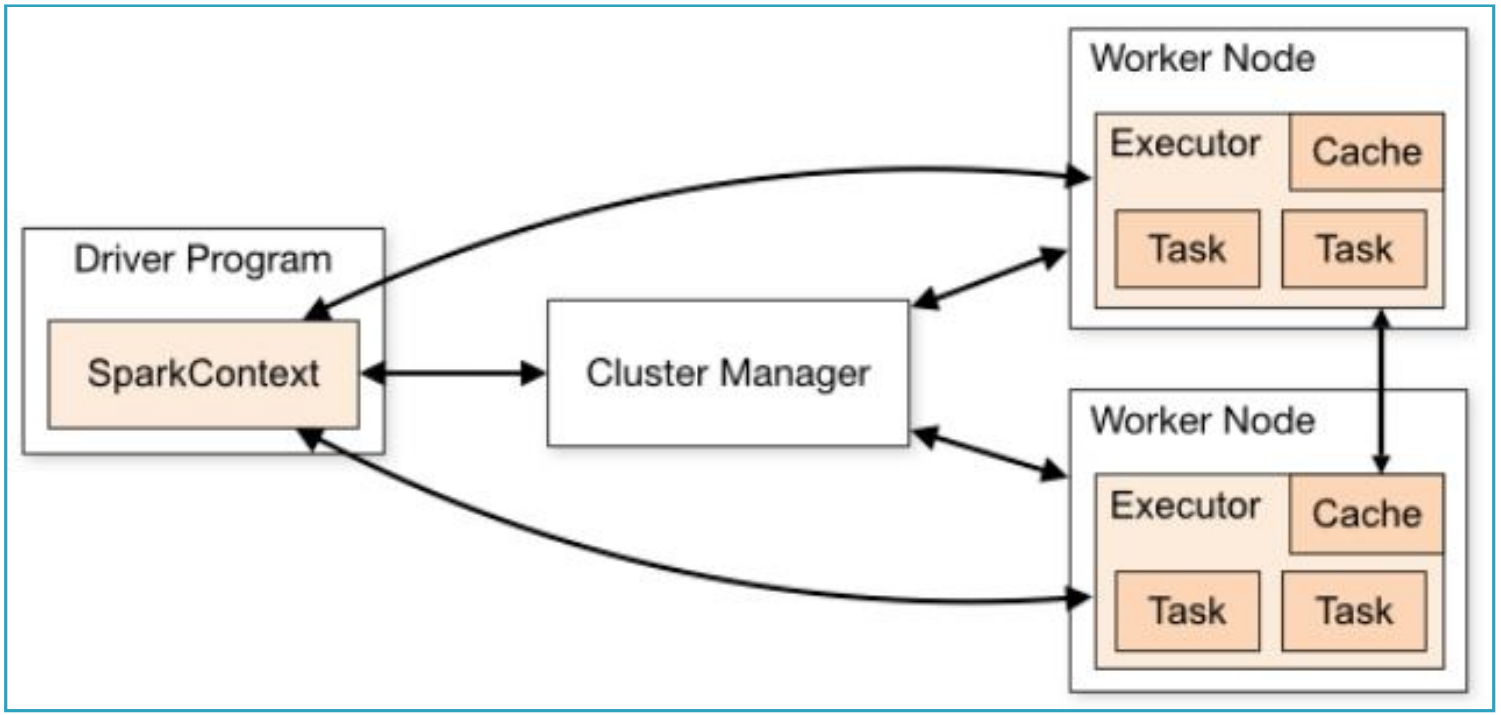


Spark vs MR – Interactive operation use case

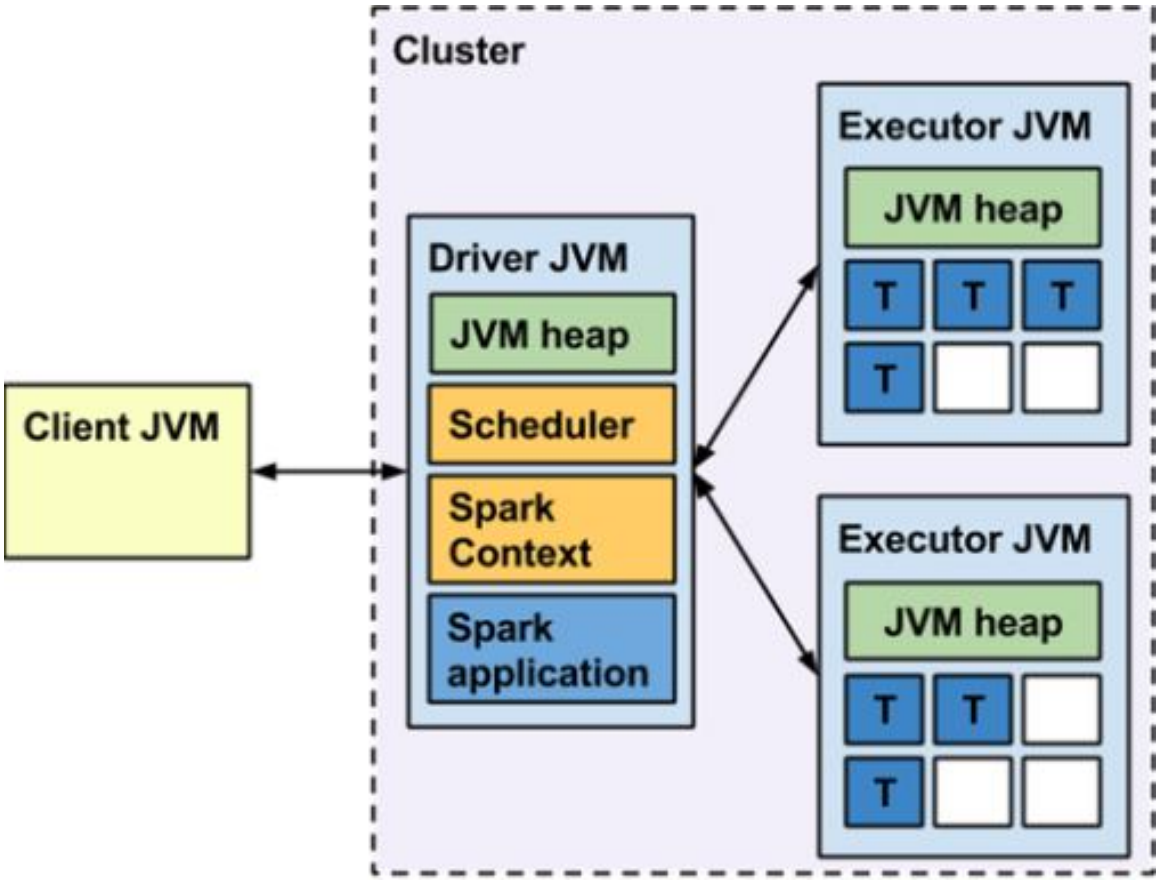




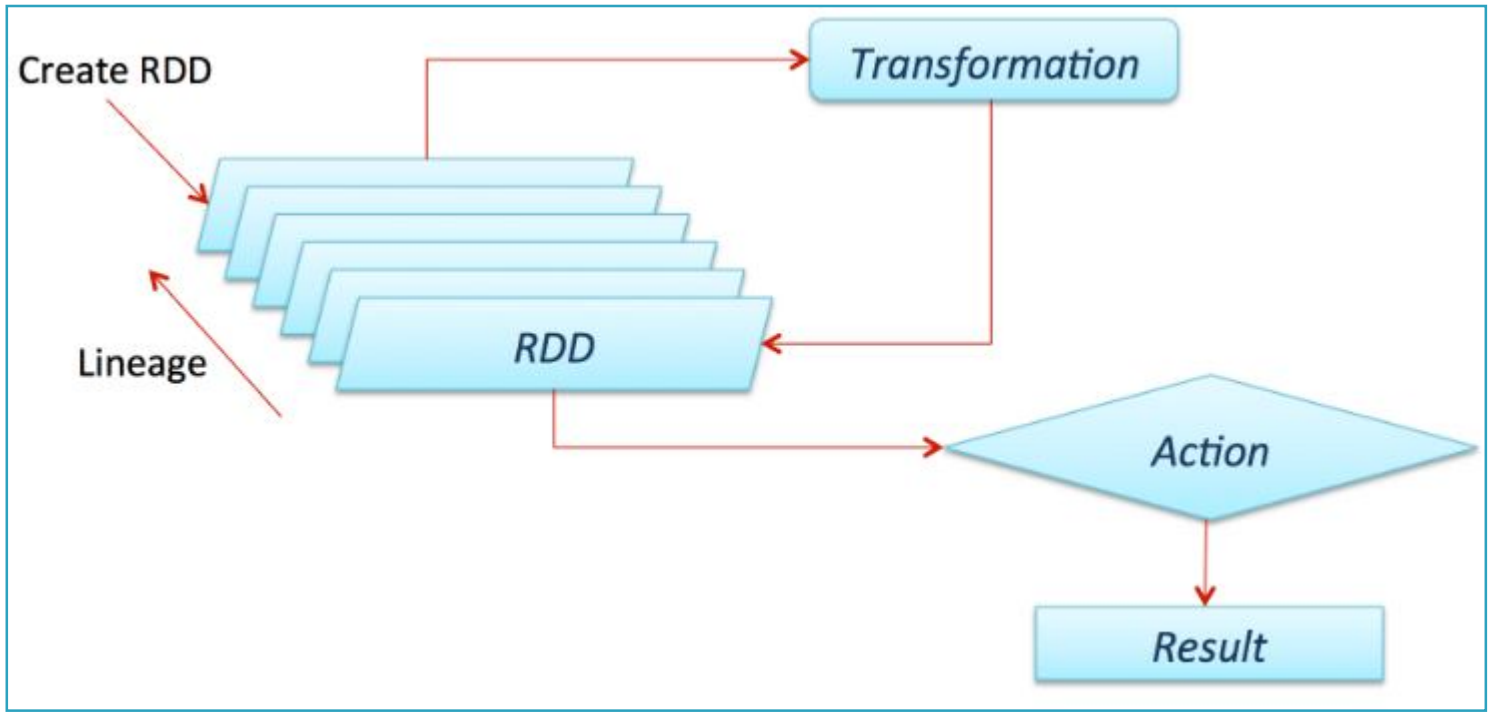
Spark Cluster Architecture



Spark Cluster Architecture (contd.)



Spark RDD



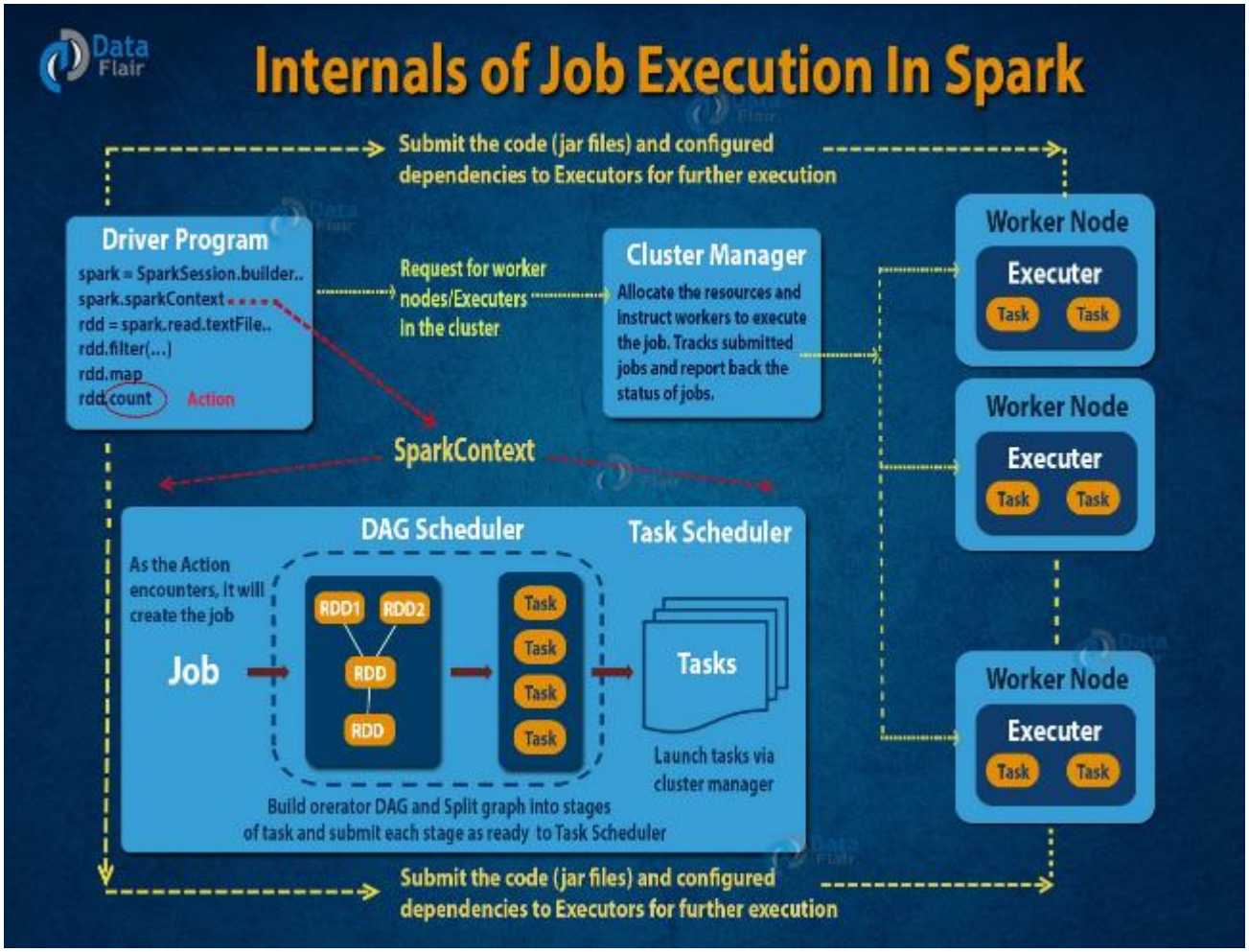
SPARK RDD (Resilient Distributed Datasets)

- RDD is a fundamental data structure of Spark.
- It is an immutable distributed collection of objects that can be stored in memory or disk across a cluster.
- Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.
- Parallel functional transformations (map, filter, …).
- Automatically rebuilt on failure.
- RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.

SPARK RDD (Resilient Distributed Datasets)

- Formally, an RDD is a read-only, partitioned collection of records.
- RDDs can be created through deterministic operations on either data on stable storage or other RDDs.
- RDD is a fault-tolerant collection of elements that can be operated on in parallel.
- There are two ways to create RDDs:
  - parallelizing an existing collection in your driver program.
  - referencing a dataset in an external storage system, such as a shared file system, HDFS, HBase, or any data source offering a Hadoop Input Format.
- Spark makes use of the concept of RDD to achieve faster and efficient MapReduce operations. Let us first discuss how MapReduce operations take place and why they are not so efficient.

Internals of Job Execution



Thank You!