Hadoop and Spark for Data Scientists

Lecture 4 : Apache Spark (Spark Core and Spark SQL)

(Data Science and Analysis at Scale)

Hassan Ahmadkhani

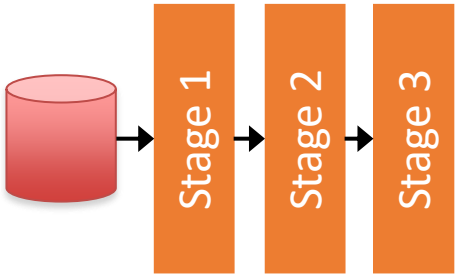
dayche.com . گروه دایکه

دای‌کِه DAYCHE

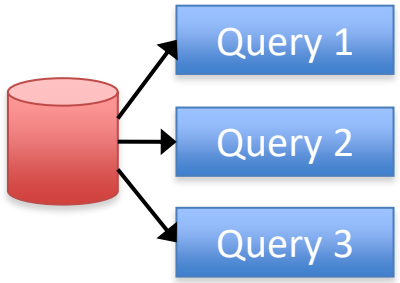
- **Spark** (Berkeley): general in-memory computing
- Map Reduce simplified “big data” analysis on large, unreliable clusters
- Map Reduce Users wanted more:
  - More *complex*, multi-stage applications
  - More *interactive* queries
  - More *low-latency* online processing

Spark Motivation

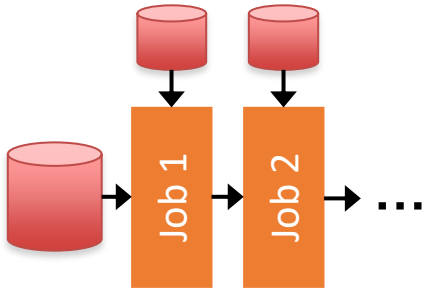
- Complex jobs, interactive queries and online processing all need one thing that MR lacks: Efficient primitives for **data sharing**



Iterative job



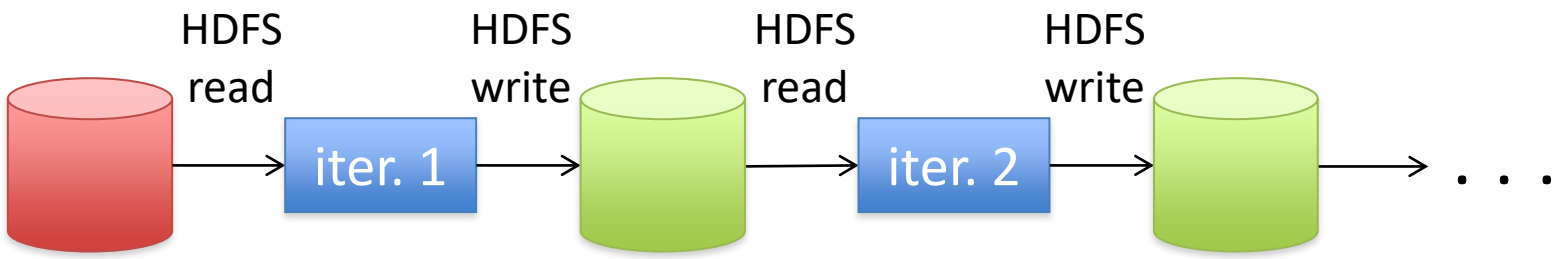
Interactive mining

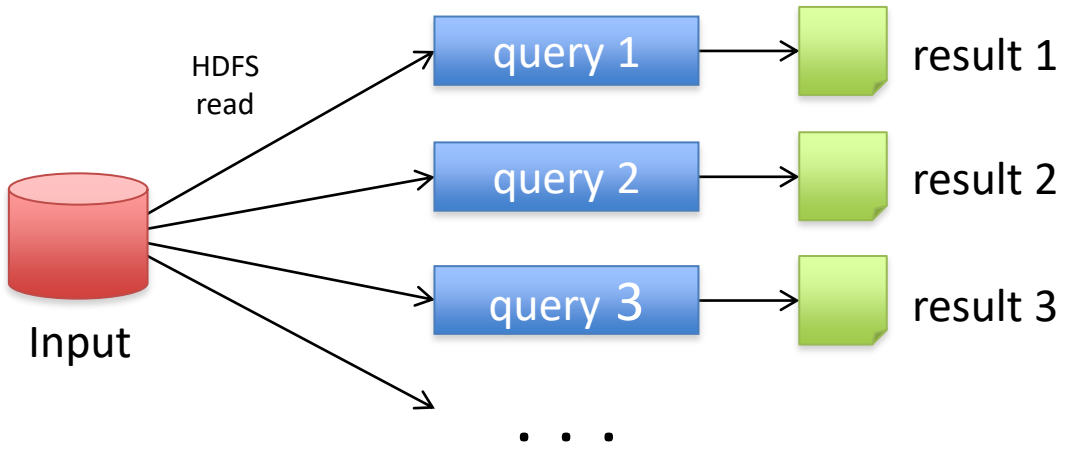


Stream processing

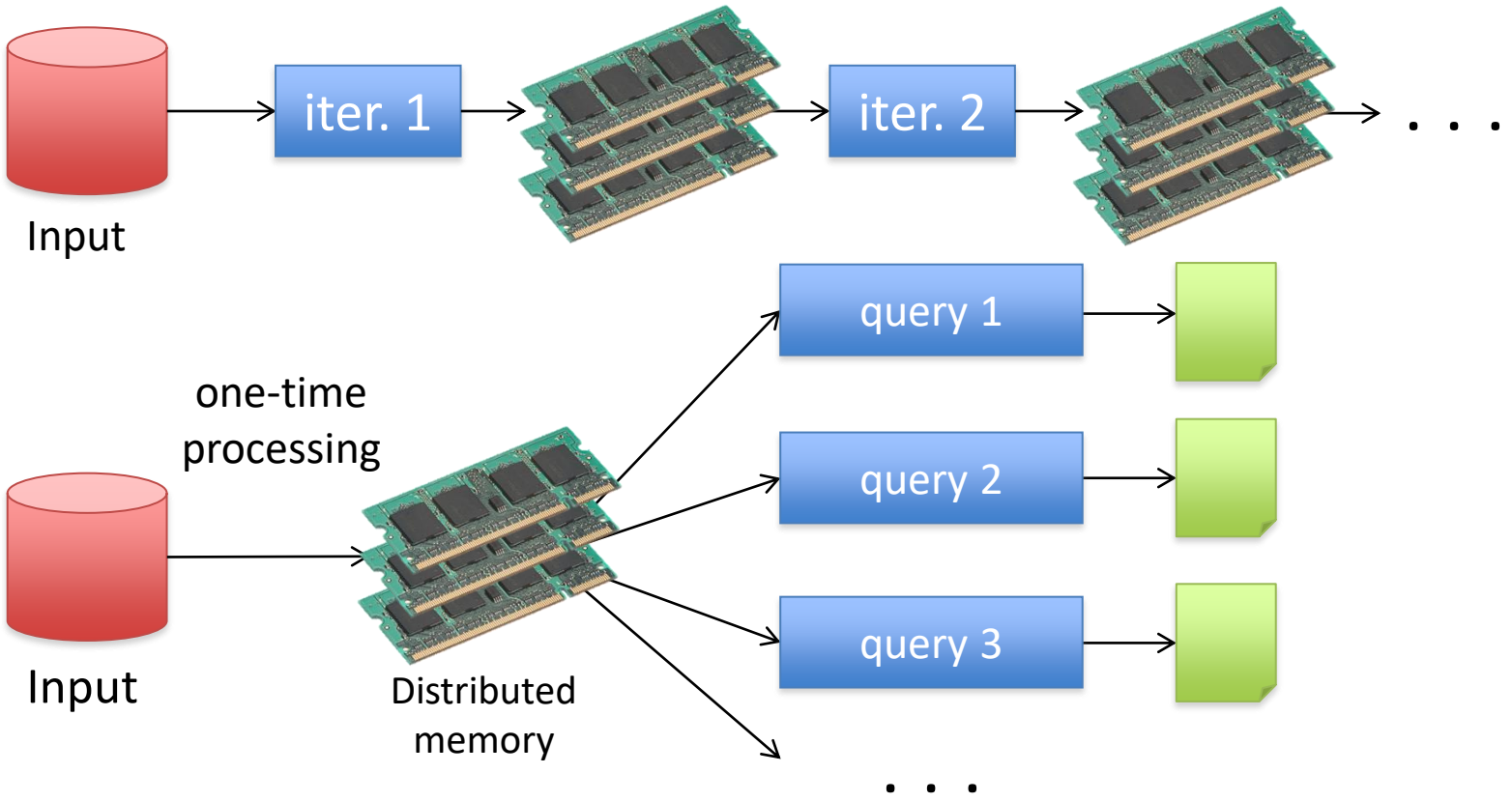
Spark Motivation

Problem: in MR, only way to share data across jobs is stable storage (e.g. file system) -> **slow!**





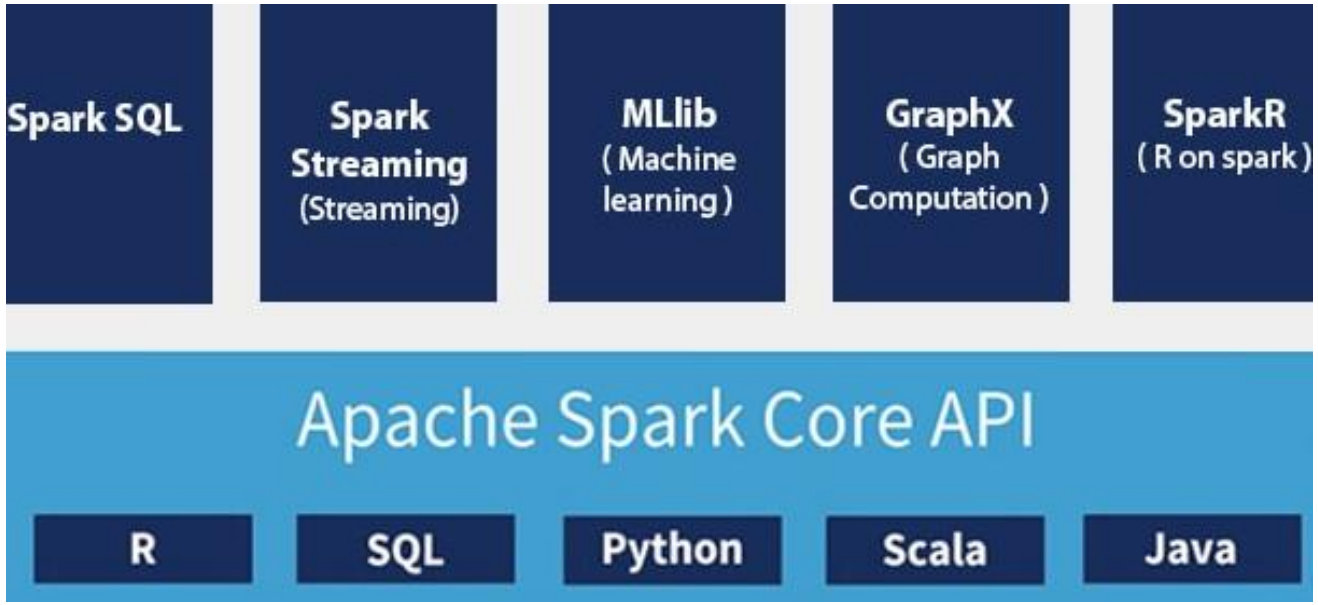
- **Spark Goal: In-Memory Data Sharing** (Cache , Persist, Unpersist)



10-100× faster than network and disk

Lecture 4 : Apache Spark (Spark Core and Spark SQL)

Spark Components



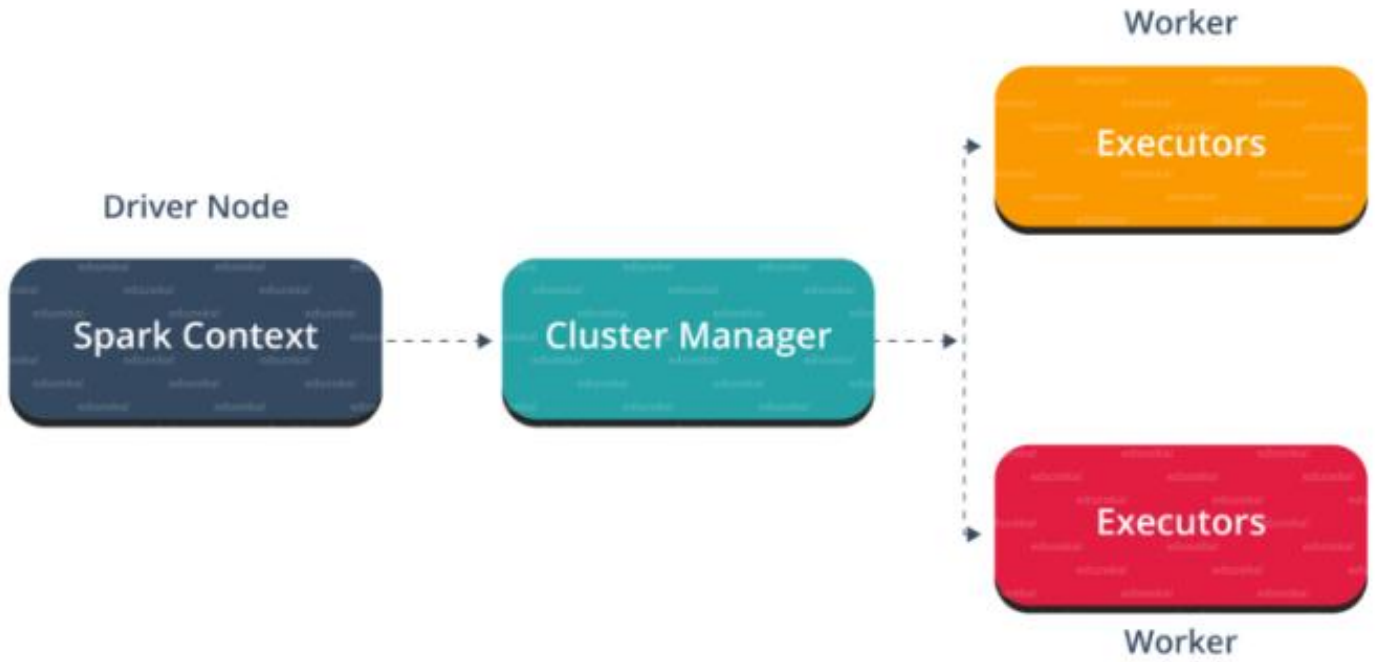
Spark Core

- Optimized Real Time, batch processing and ETL in big data

You can combine these libraries seamlessly in the same application

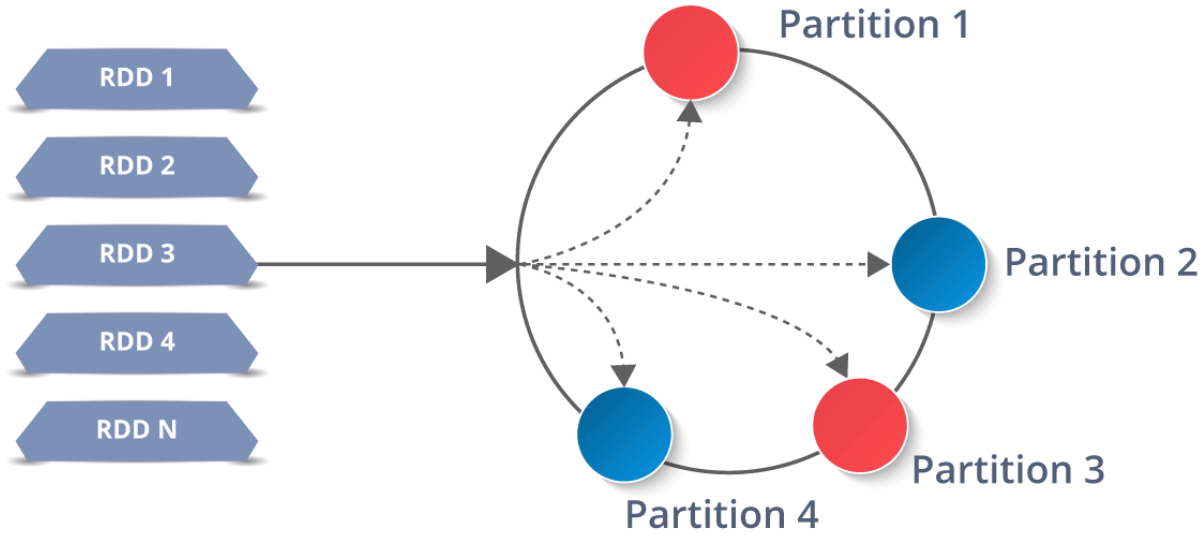
more general: map/reduce is just one set of supported constructs

Spark Processing Cluster Architecture



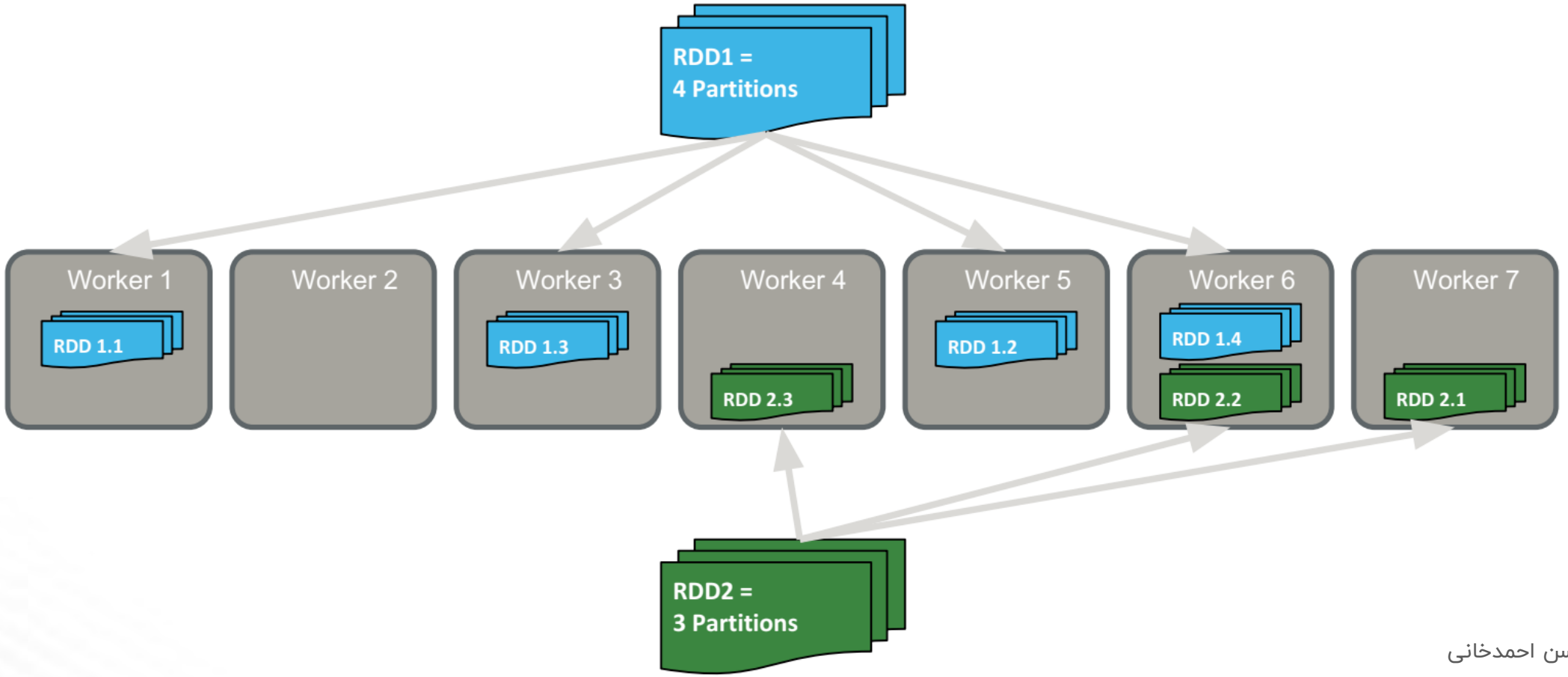
- Spark cluster components vs Map Reduce cluster components
  - Driver – Worker
  - Job tracker – Task tracker

- **Interface to access the operating system's service**
  - Spark shell (scala)
  - Pyspark
  - …
- Spark **transformations** and **actions**
  - filter(), partitions(), cache(), count(), collect
- Transformations and actions on Data



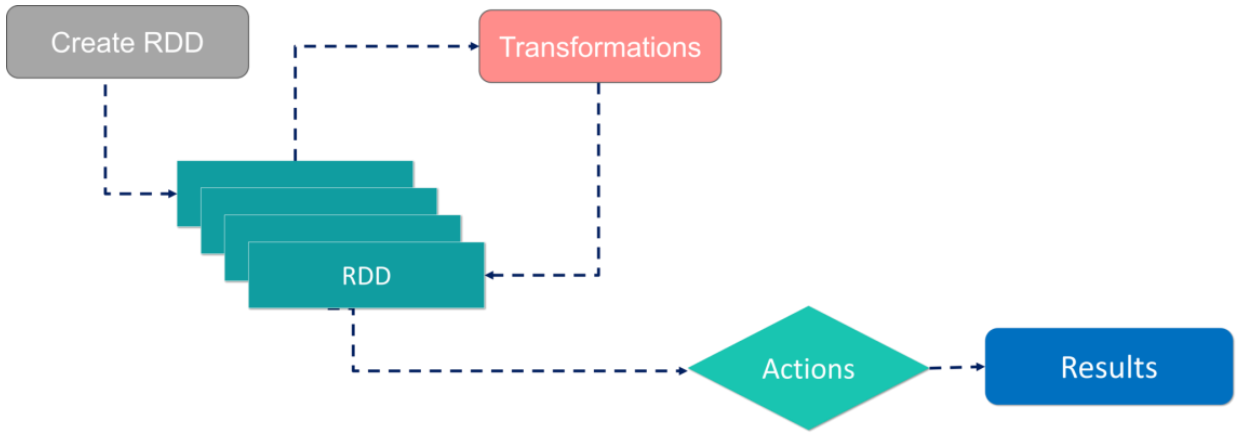
- **RDD (Resilient Distributed Dataset)**
  - the fundamental data structure of spark which are collection of objects which computes on the different node of the cluster.
  - every dataset in **Spark RDD** is logically partitioned across many servers

Lecture 4 : Apache Spark (Spark Core and Spark SQL)



تولید محتوا: حسن احمدخانی
daychegroup
daychegroup
dayche.com | گروه دایکه

- **RDD (Resilient Distributed Dataset)**
  - Once you create an RDD it becomes ***immutable***.
  - By immutable I mean, an object whose state cannot be modified after it is created, but they can surely be transformed



- **Transformations:** They are the operations that are applied to create a new RDD.
- **Actions:** They are applied on an RDD to instruct Apache Spark to apply computation and pass the result back to the driver.
- A ***partition*** is a *logical chunk* of a *large distributed data set*

- **Spark Shell**
- **Spark Applications**
- **Spark is Still Based on MapReduce Principles**

تولید محتوا: حسن احمدخانی
daychegroup
daychegroup
dayche.com | گروه دایکه

Essential Core & Intermediate Spark Operations

General

- map
- filter
- flatMap
- mapPartitions
- mapPartitionsWithIndex
- groupBy
- sortBy

Math / Statistical

- sample
- randomSplit

Set Theory / Relational

- union
- intersection
- subtract
- distinct
- cartesian
- zip

Data Structure / I/O

- keyBy
- zipWithIndex
- zipWithUniqueID
- zipPartitions
- coalesce
- repartition
- repartitionAndSortWithinPartitions
- pipe

Spark Operations = TRANSFORMATIONS + ACTIONS

- reduce
- collect
- aggregate
- fold
- first
- take
- forEach
- top
- treeAggregate
- treeReduce
- forEachPartition
- collectAsMap

- count
- takeSample
- max
- min
- sum
- histogram
- mean
- variance
- stdev
- sampleVariance
- countApprox
- countApproxDistinct

- takeOrdered

- saveAsTextFile
- saveAsSequenceFile
- saveAsObjectFile
- saveAsHadoopDataset
- saveAsHadoopFile
- saveAsNewAPIHadoopDataset
- saveAsNewAPIHadoopFile

تولید محتوا: حسن احمدخانی
daychegroup
daychegroup
dayche.com | گروه دایکه

Essential Core & Intermediate PairRDD Operati

General

- flatMapValues
- groupByKey
- reduceByKey
- reduceByKeyLocally
- foldByKey
- aggregateByKey
- sortByKey
- combineByKey

Math / Statistical

- sampleByKey

Set Theory / Relational

- cogroup (=groupWith)
- join
- subtractByKey
- fullOuterJoin
- leftOuterJoin
- rightOuterJoin

Data Structure

- partitionBy

Spark Operations = TRANSFORMATIONS + ACTIONS

- keys
- values

- countByKey
- countByValue
- countByValueApprox
- countApproxDistinctByKey
- countApproxDistinctByKey
- countByKeyApprox
- sampleByKeyExact

تولید محتوا: حسن احمدخانی
daychegroup
daychegroup
dayche.com | گروه دایکه