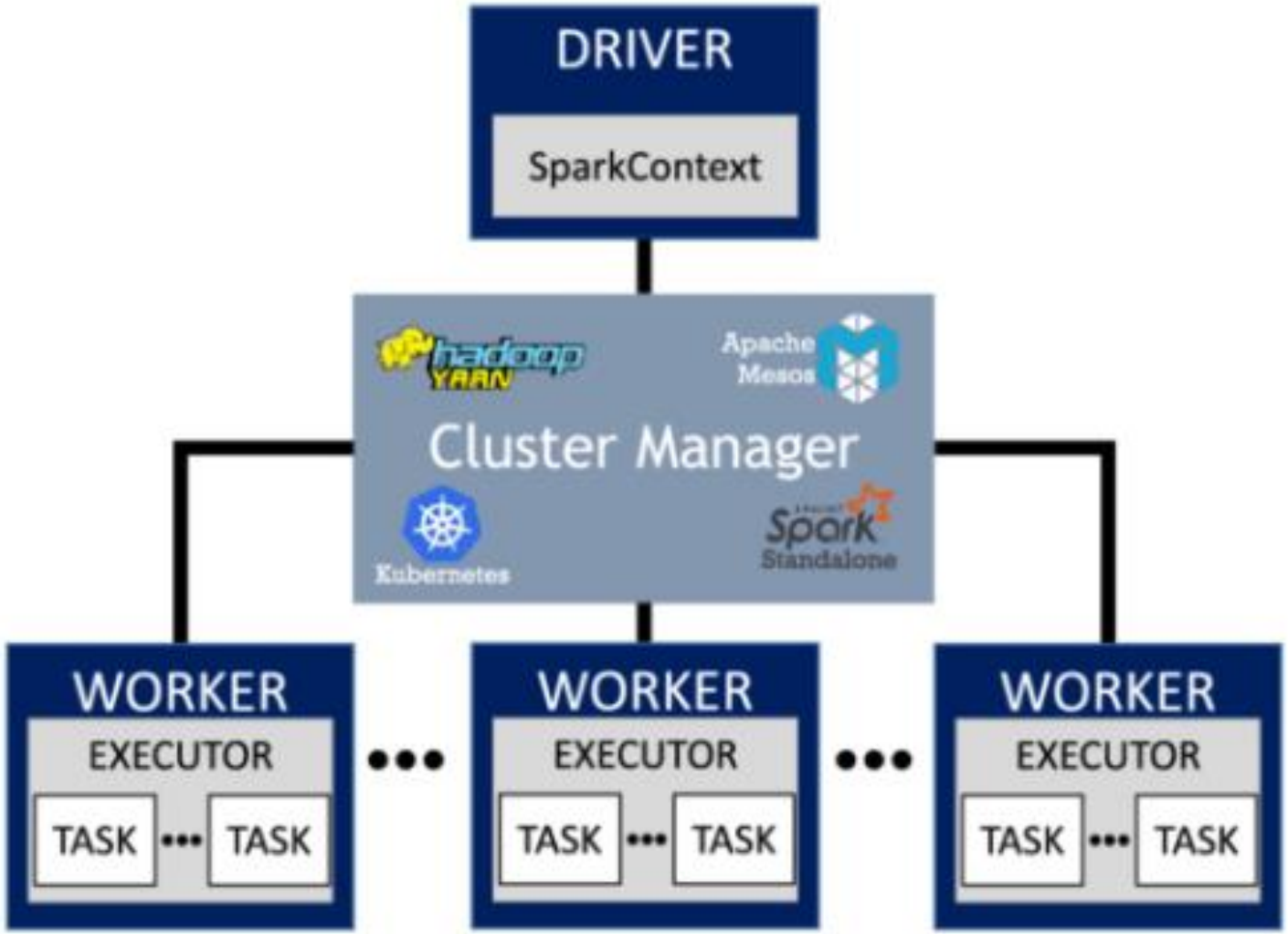
Systems, paradigms and algorithms for Big Data
TD 1

Why Spark ?

- Ease creation of complicated data analysis task
- Support iterative algorithms, like gradient descent
- Ease data exploration
- Make it easier to explore data interactively
  - Developer friendly, no need to create tons of classes and jobs…
  - Repl mode
  - Scala/Python API similar to functional programming

Main Components



How to run it

- Repl : Spark-shell, notebooks ➔ exploratory mode
- Spark-submit : runs a script ➔ scheduled job
- Api for Java, Scala, Python.
- Deploy Mode : client vs cluster

Dependencies

- Depends on Hadoop Libraries (HDFS and Yarn)
- Requires a Java Runtime Environment (need Java in your system path)

RDD

Resilient Distributed Dataset

- The base building block of your application
- Lazy : doesn’t compute anything before it really needs to do so.
- Immutable : a transformation doesn’t change the data set, it returns a new RDD.
- Fault Tolerant : partition can be recomputed in case of failure

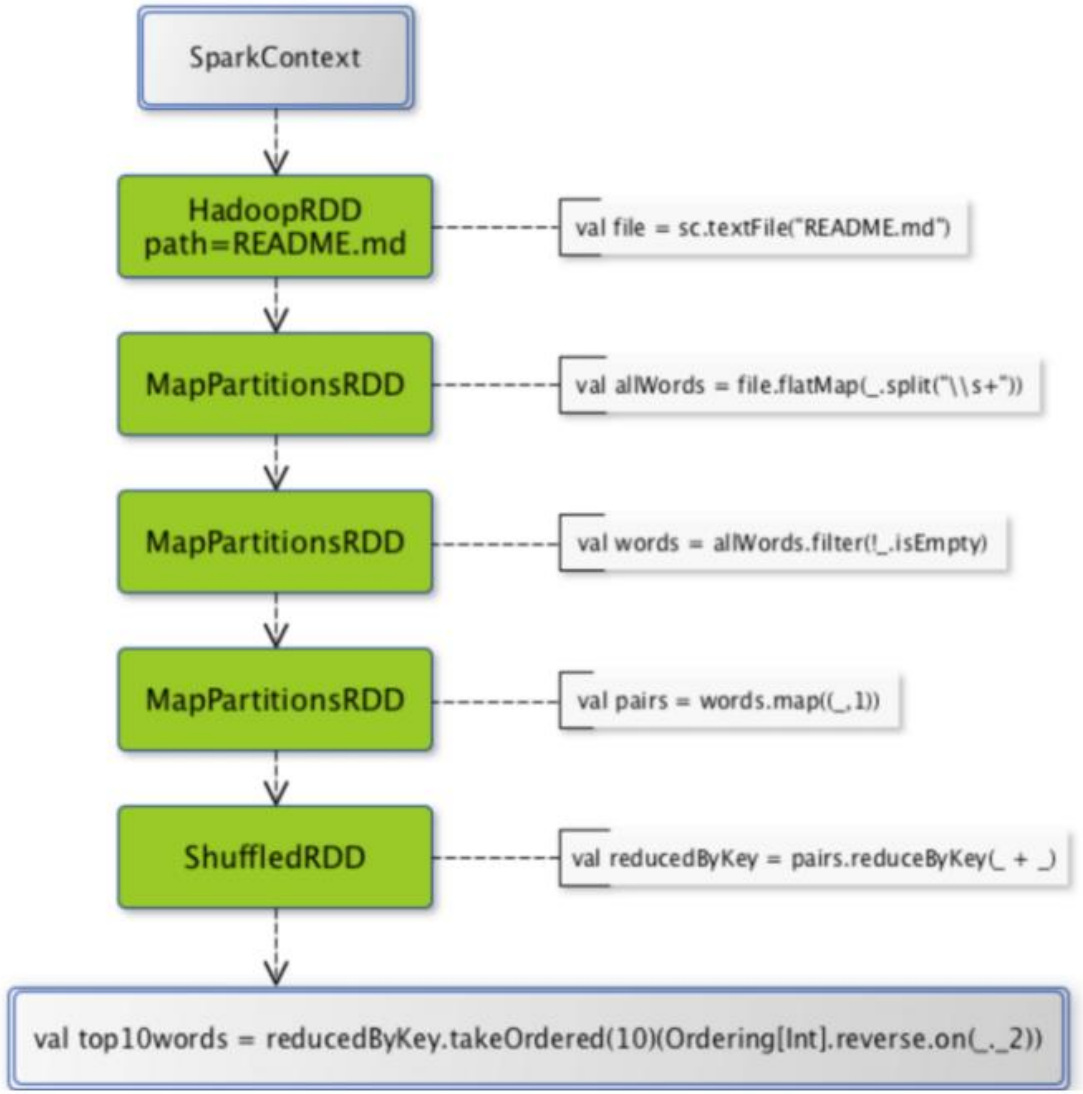
Graph

- Conceptually, an RDD is a node of a graph of functions.
- API is pretty much similar with **Functional Programming. Forget the for-loop.**

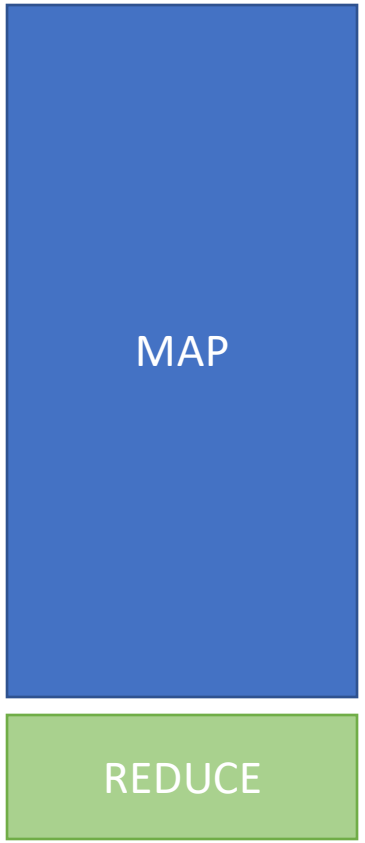
Inside the RDD:

- Knows the partitions he is working-on, how data is partitionned
- Knows how to iterate over each partition to yield records
- Know RDD’s it depends-on

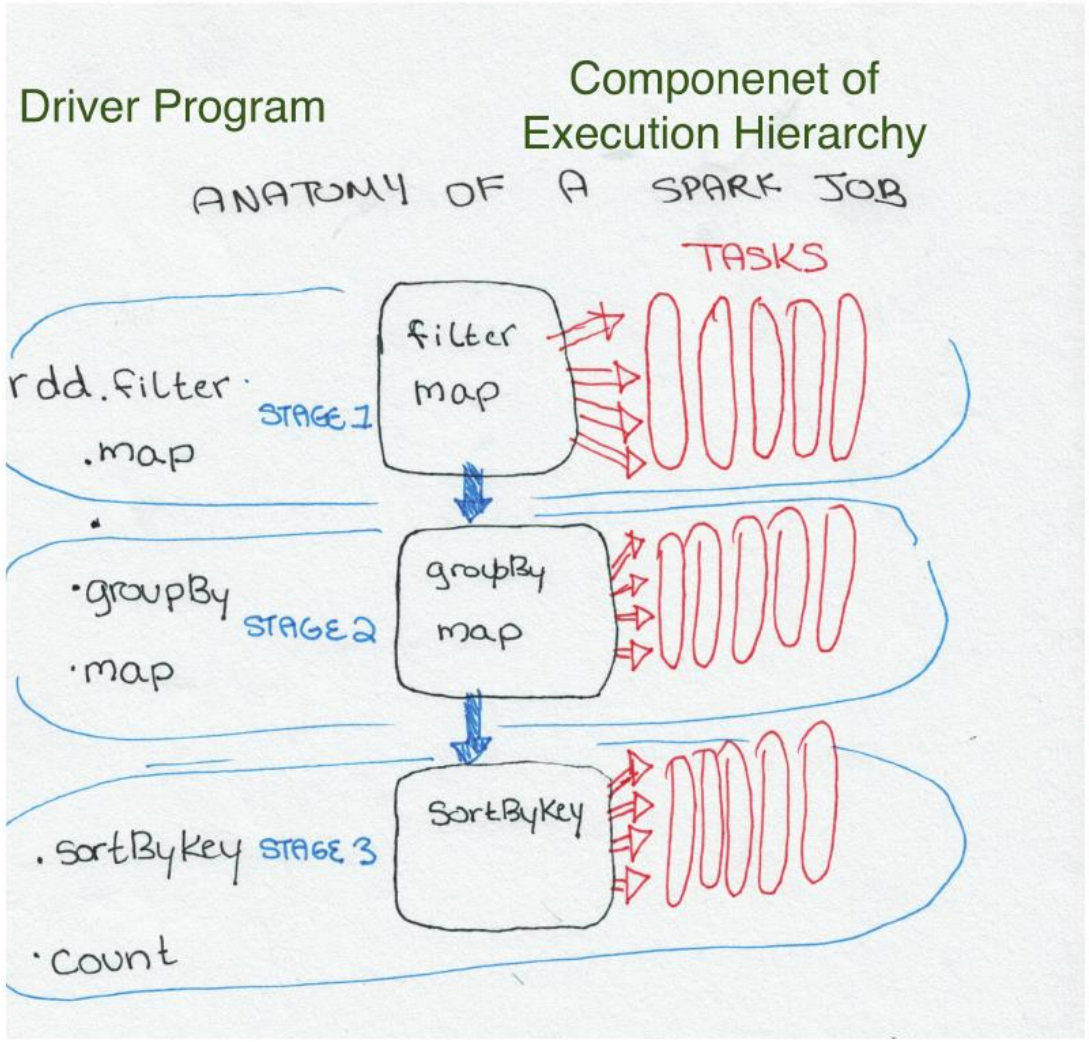
RDD as a Directed Acyclic Graph



New RDD is created after every transformation.(DAG graph)

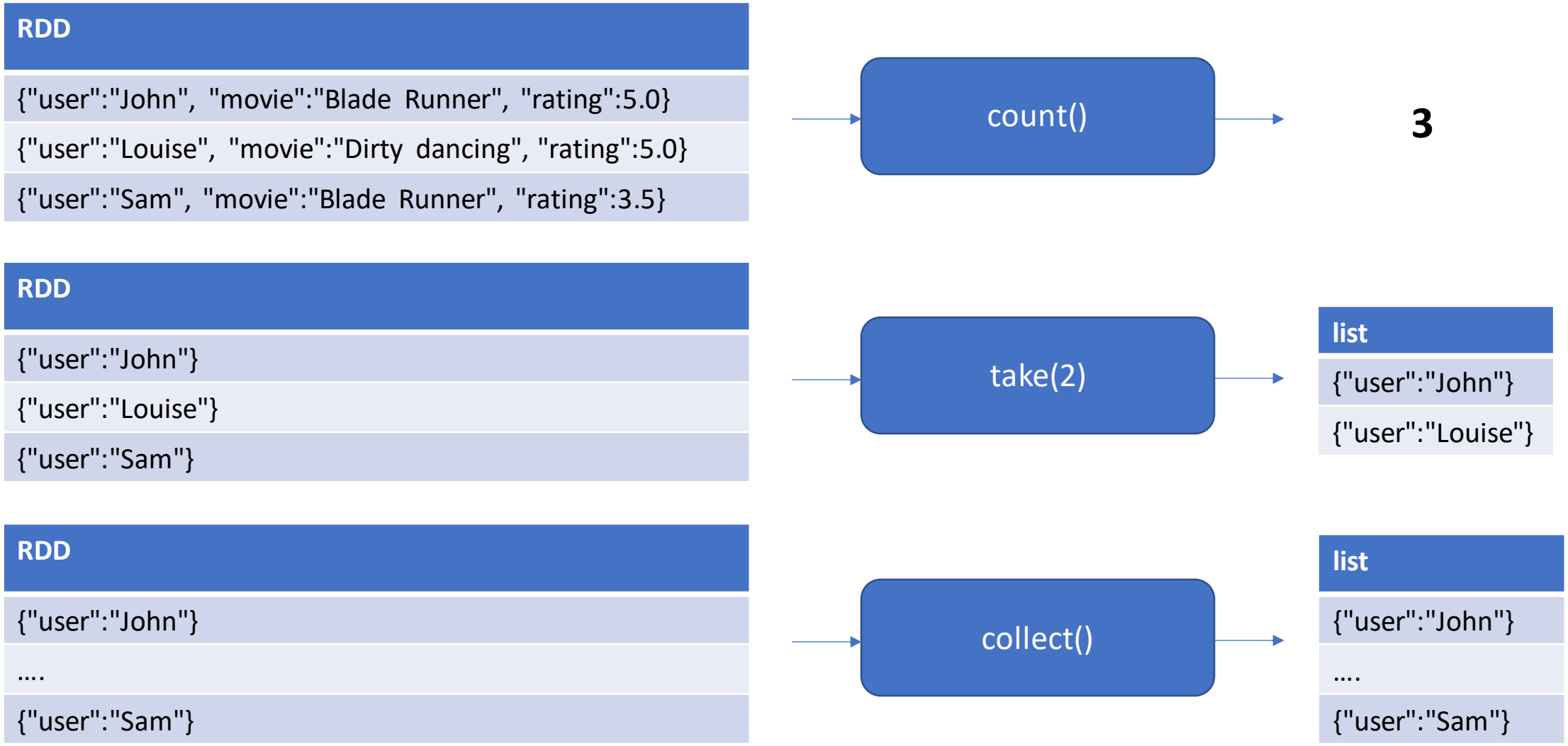


Jobs, Stages, Tasks

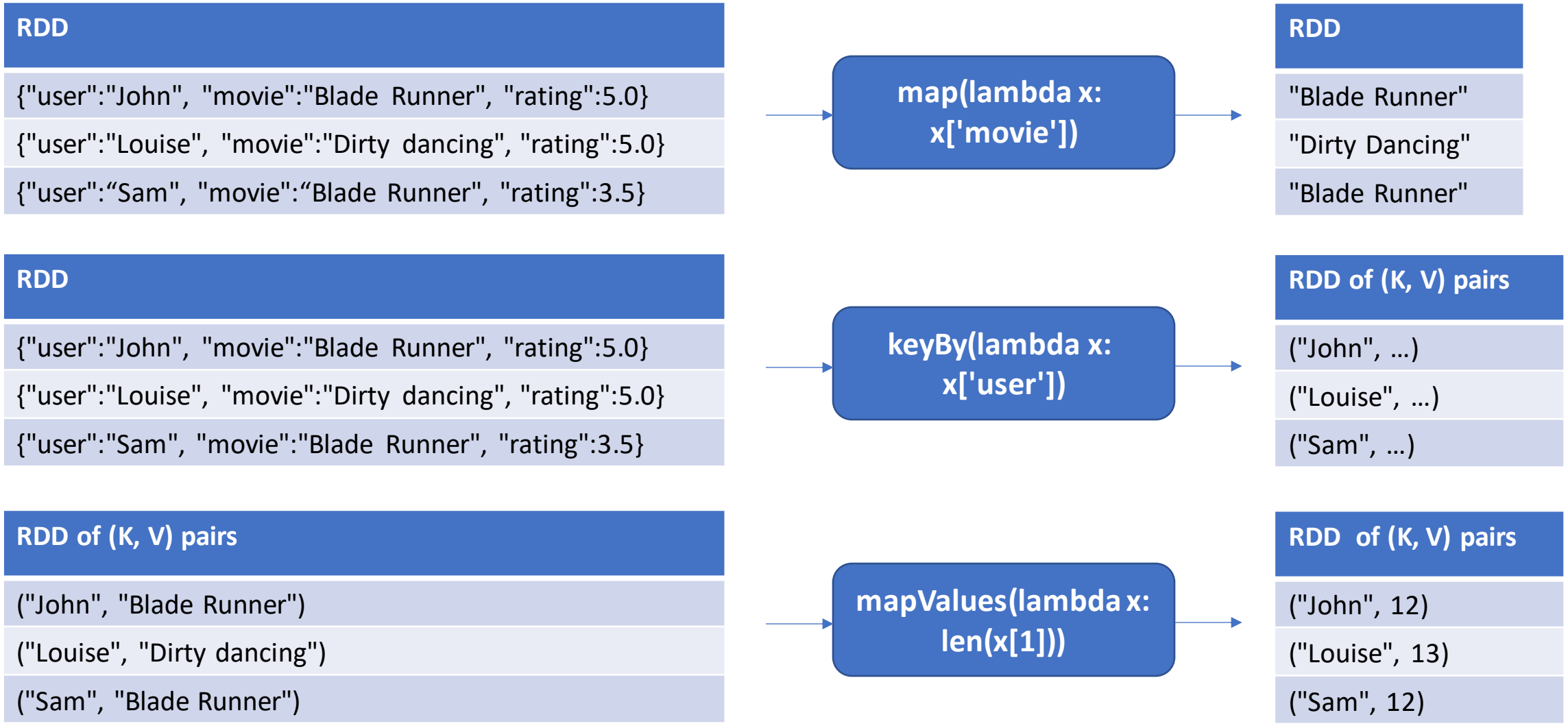


Source : high performance Spark

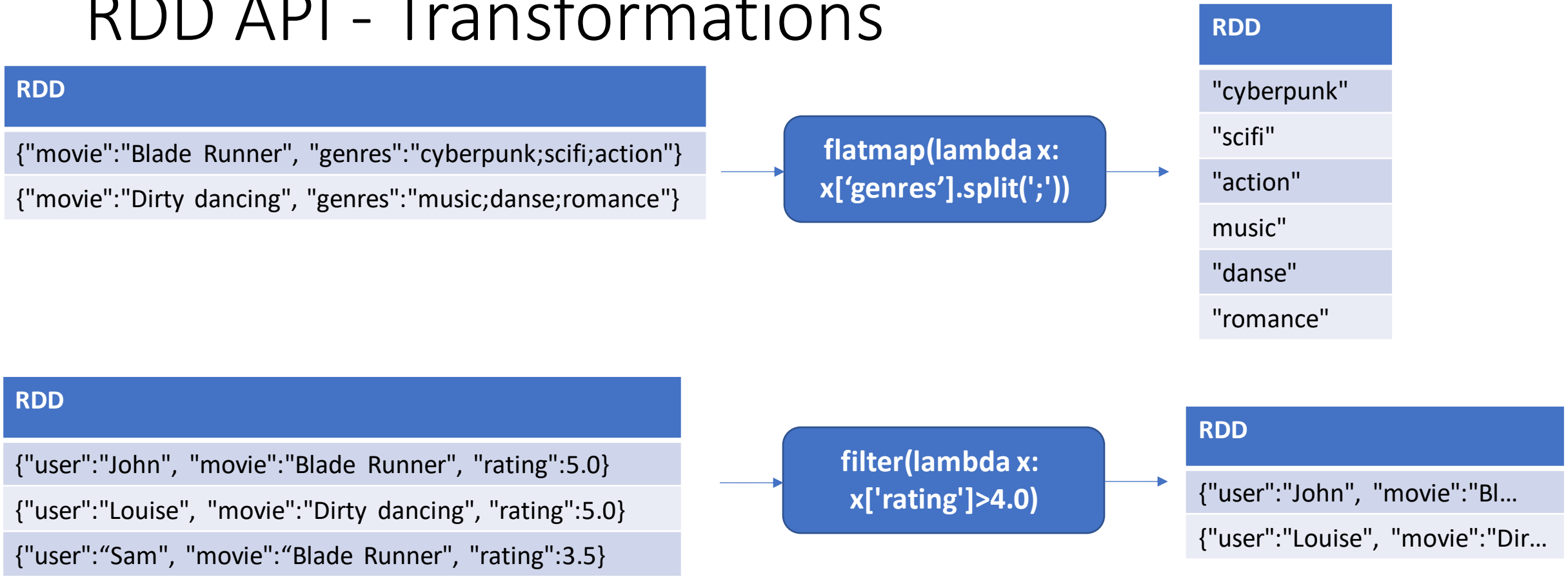
RDD API - Actions



RDD API - Transformations

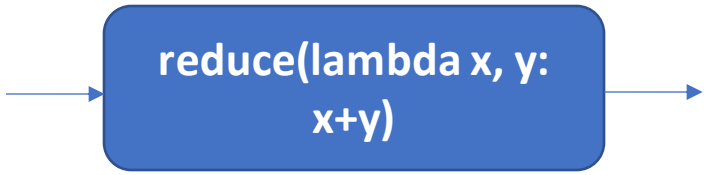


RDD API - Transformations



RDD API – Actions

RDD
5.0
5.0
3.5



13.5

RDD of (K, V) pairs
("Blade Runner", 5.0)
("Dirty dancing", 5.0)
("Blade Runner", 3.5)

RDD of (K, V) pairs
("Blade Runner", 8.5)
("Dirty dancing", 5.0)

Other useful functions

- **Join (shuffle ? Lazy ?)**
- **Sample**
- **mappartitions**
- **zippartitions**

Links

- https://spark.apache.org/docs/latest/api/scala/org/apache/spark/api/java/JavaPairRDD.html
- https://0x0fff.com/hadoop-mapreduce-comprehensive-description/