

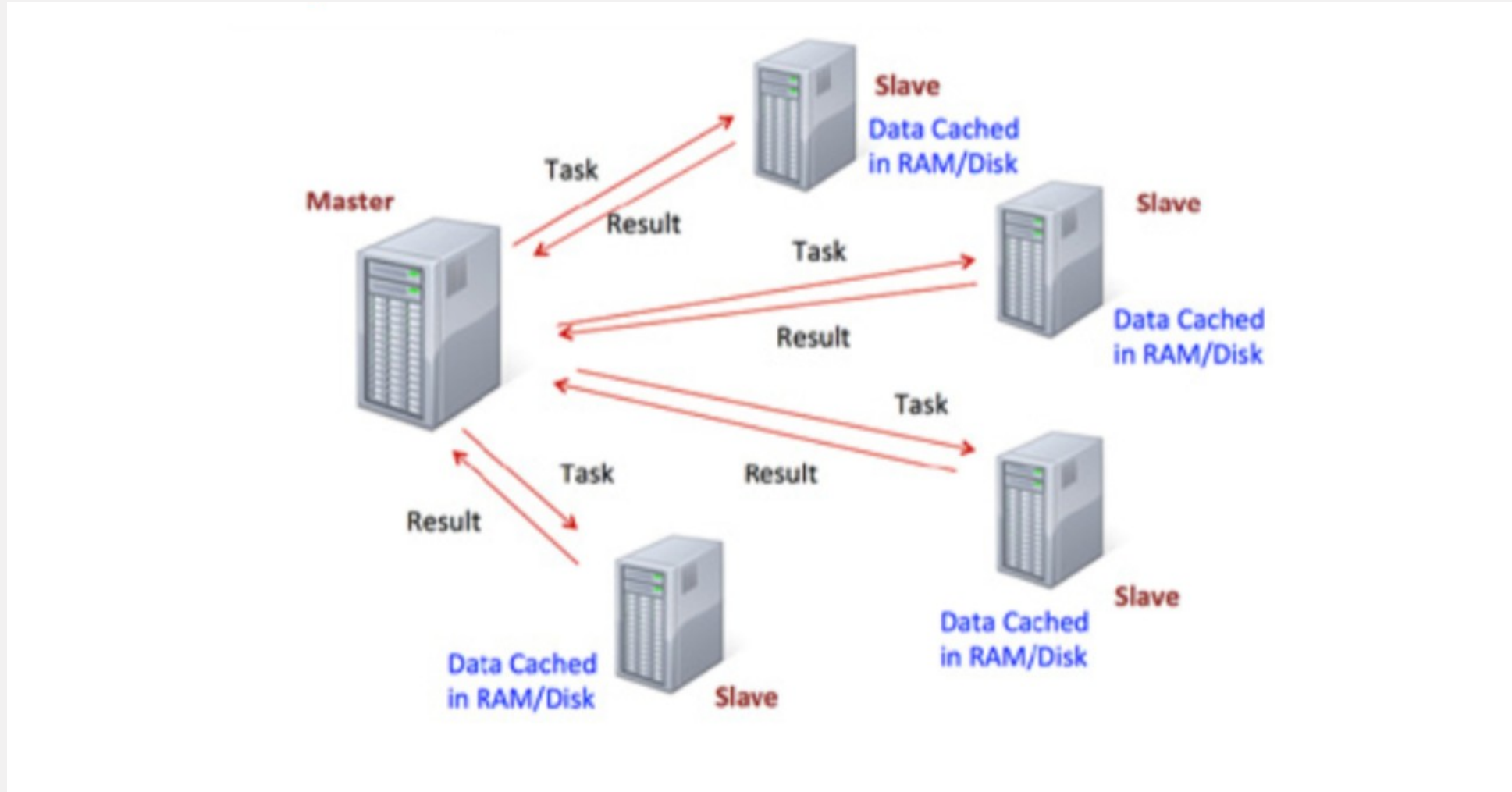
4.4 Spark Architecture

DSC 232R, Class 4 : Spark - 2

April 13th



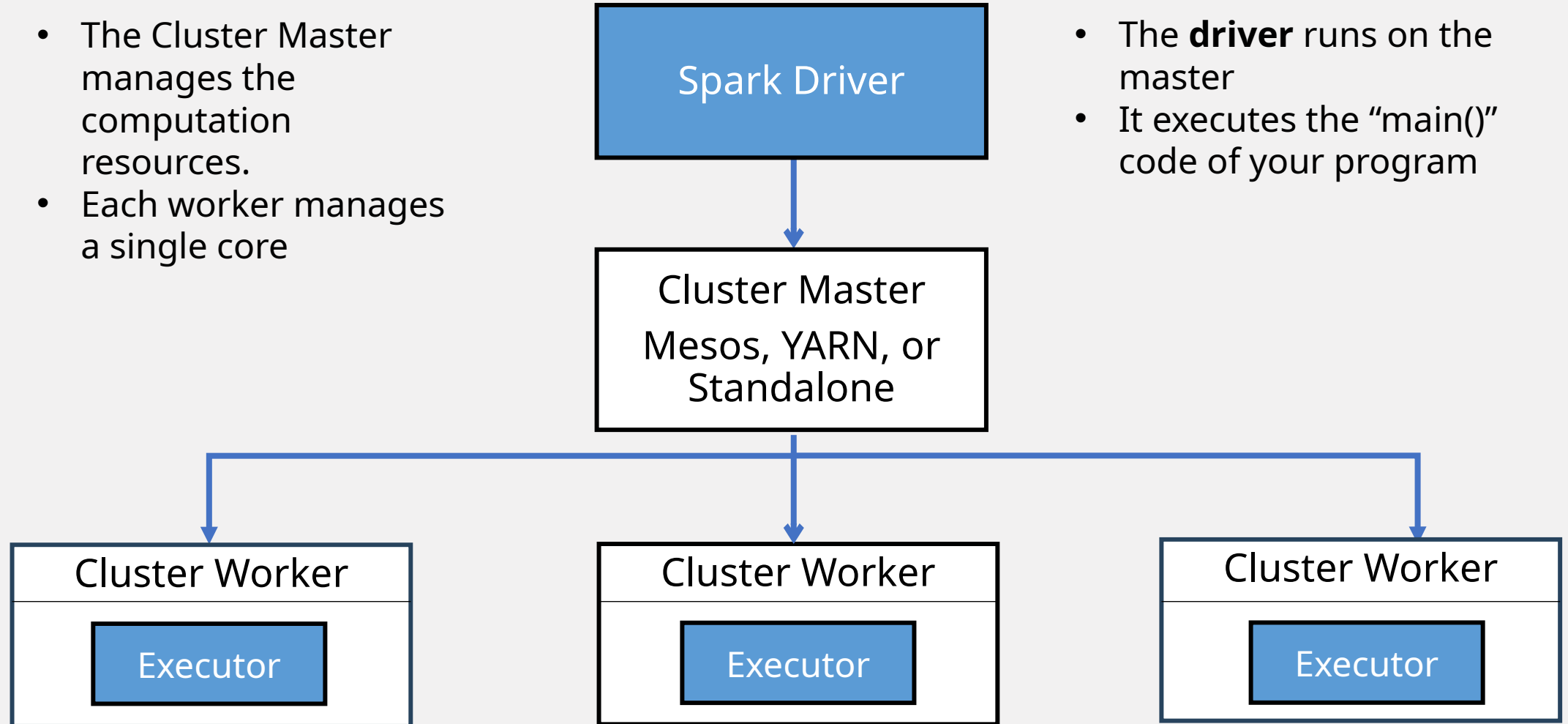
Hardware Organization



In local installation, cores serve as master & slaves

Spatial Software Organization

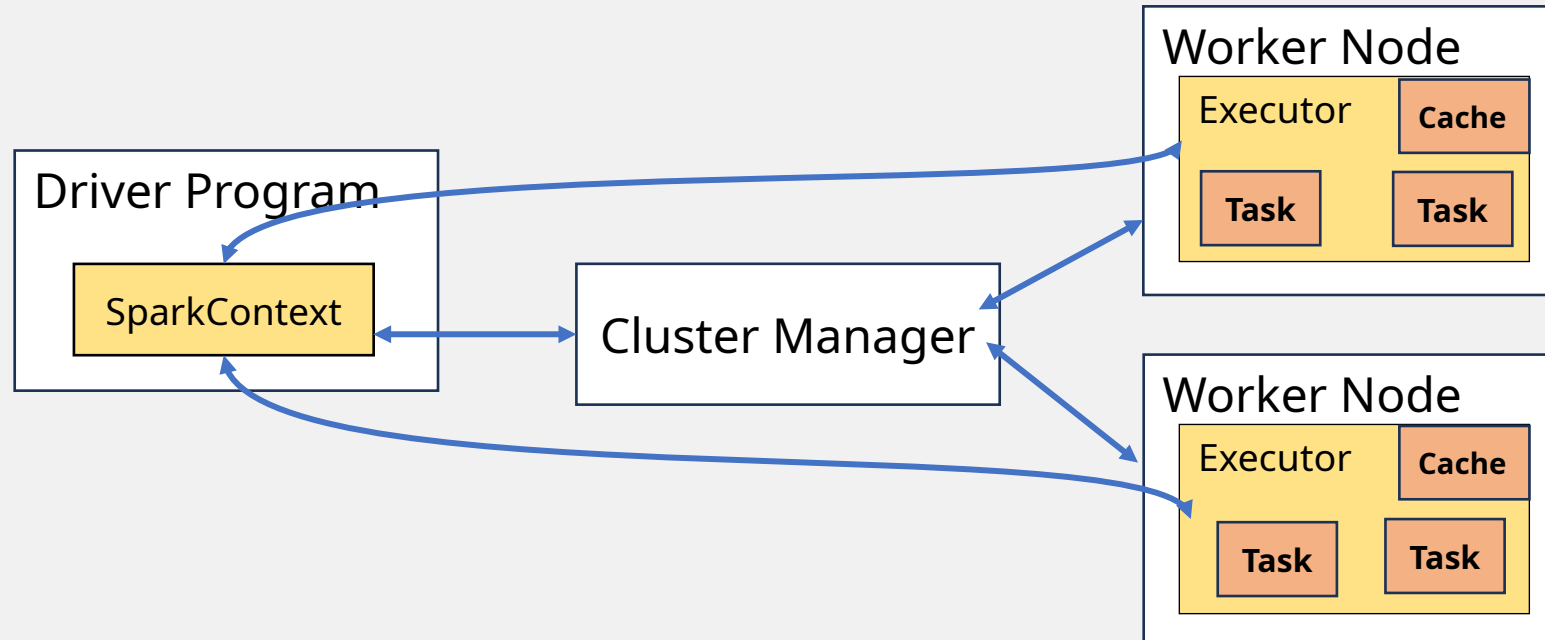
- The Cluster Master manages the computation resources.
- Each worker manages a single core



- The **driver** runs on the master
- It executes the "main()" code of your program

- Each RDD is partitioned among the workers
- Workers manage partitions and Executors
- Executors execute task on their partition, are myopic

Spatial Organization (more detail)



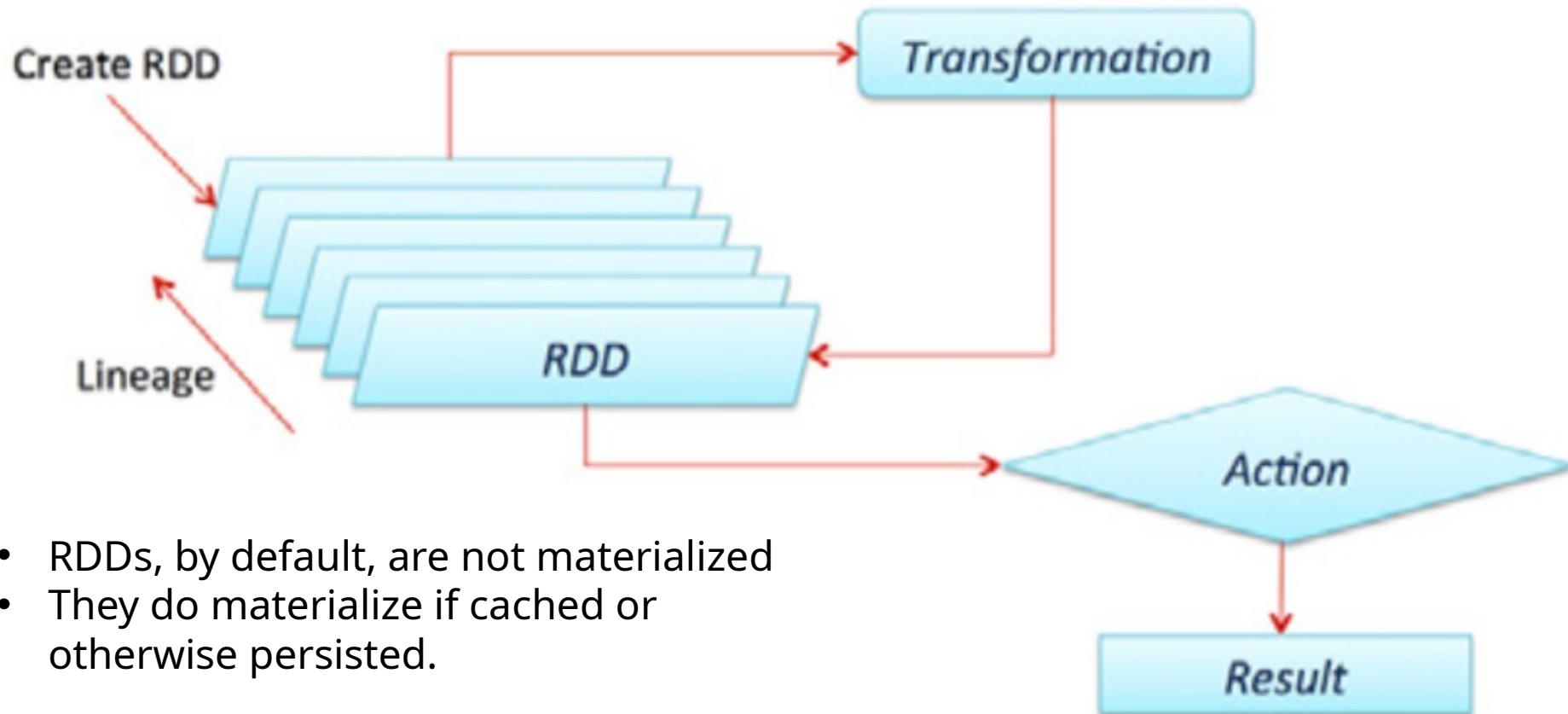
- SparkContext (sc) is the abstraction that encapsulates the cluster for the driver node (and the programmer)
- Worker nodes manage resources in a single slave machine
- Worker nodes communicate with the cluster manager
- Executors are the processes that can perform **tasks**.
- Cache refers to the local memory on the slave machine

Materialization

- ▯ Consider RDD1
- ▯ `-> Map (x: x*x) -> RDD2`
- ▯ `->Reduce (x,y:x+y)-> float (in head node)`

- ▯ RDD1 -> RDD2 is a lineage
- ▯ RDD2 can be consumed as it is being generated.
- ▯ Does not have to be **materialized** = stored in memory

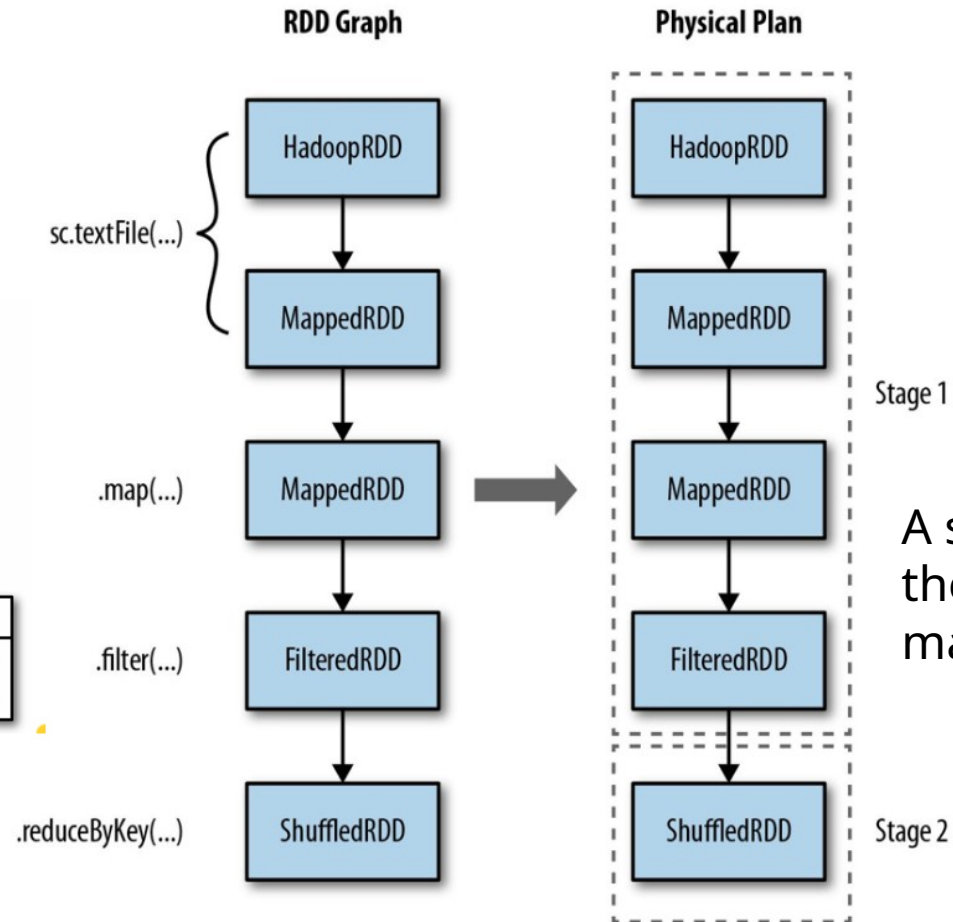
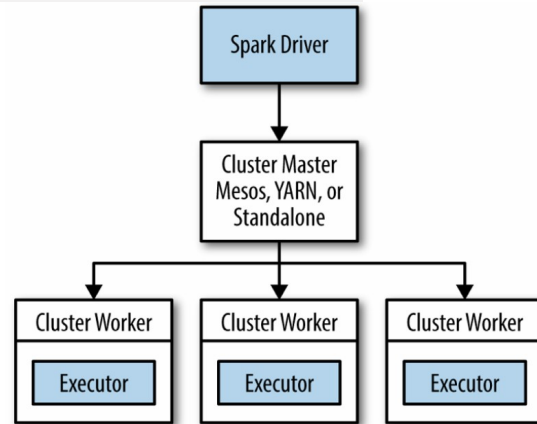
RDD Processing



Temporal Organization

RDD Graph and Physical plan

Recall Spatial Organization



A stage ends when the RDD needs to be materialized

Figure 8-1. RDD transformations pipelined into physical stages

Terms and Concepts of Execution

- ▮ RDDs are **partitioned** across workers.
- ▮ RDD graph defines the **Lineage** of the RDDs.
- ▮ SparkContext divides the RDD graph into **stages** which define the execution plan (or physical plan)
- ▮ A **task** corresponds to the to *one stage*, restricted to *one partition*.
- ▮ An **executor** is a process that performs tasks.

Summary

- ▮ Spark computation is broken into tasks
- ▮ Spatial organization: different data
- ▮ partitions on different machine
- ▮ Temporal organization: Computation
- ▮ is broken into stages. a sequence of stages.
- ▮ Next: persistence and checkpointing