

## Master Node

Master

Driver

```
def main() {  
  val sc = new SparkContext()  
  val rdd = sc.makeRDD()  
  val finalRDD = rdd.transformation()  
  val result = finalRDD.action()  
}
```

mapOutputTrackerMaster

mapStatuses:  
HashMap<stageId,  
Array[MapStatus]>

mapOutputTrackerMaster  
Actor

## Worker Node 1

Worker

ExecutorRunner

CoarseGrainedExecutorBackend

BasicBlockFetcherIterator

blockManager.getMultiple(blockByAddress)

mapOutputTrackerWorker

mapOutputTrackerMaster  
ActorRef

getServerStatuses()

blockStoreShuffleFetcher  
statuses => blocksByAddress

fetch(shuffleId, reduceId)

Executor

rdd.iterator()

GetMapOutputStatuses(shuffleId)  
statuses: Array[(BlockManagerId, FS)]  
for given reducer