Big data platform (Spark) performance acceleration

*Mentors: Tony Tan, Ning Wu, Yong Wang and **Theo Gkountouvas***

By:

Grishma Atul Thakkar

Virat Goradia

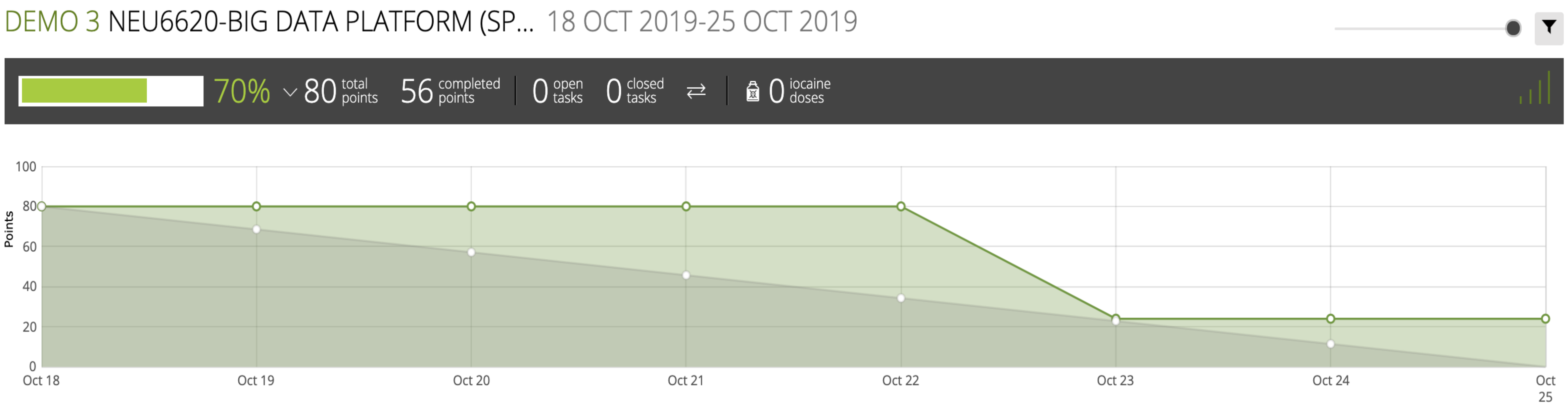
Nipun Midha

Baoshu Brady Qi

Sprint Goals

- Further understand existing spark internal code
- Design strategies to implement N-Way merge algorithm
- Start implementing the N-Way merge algorithm.

Burndown Chart

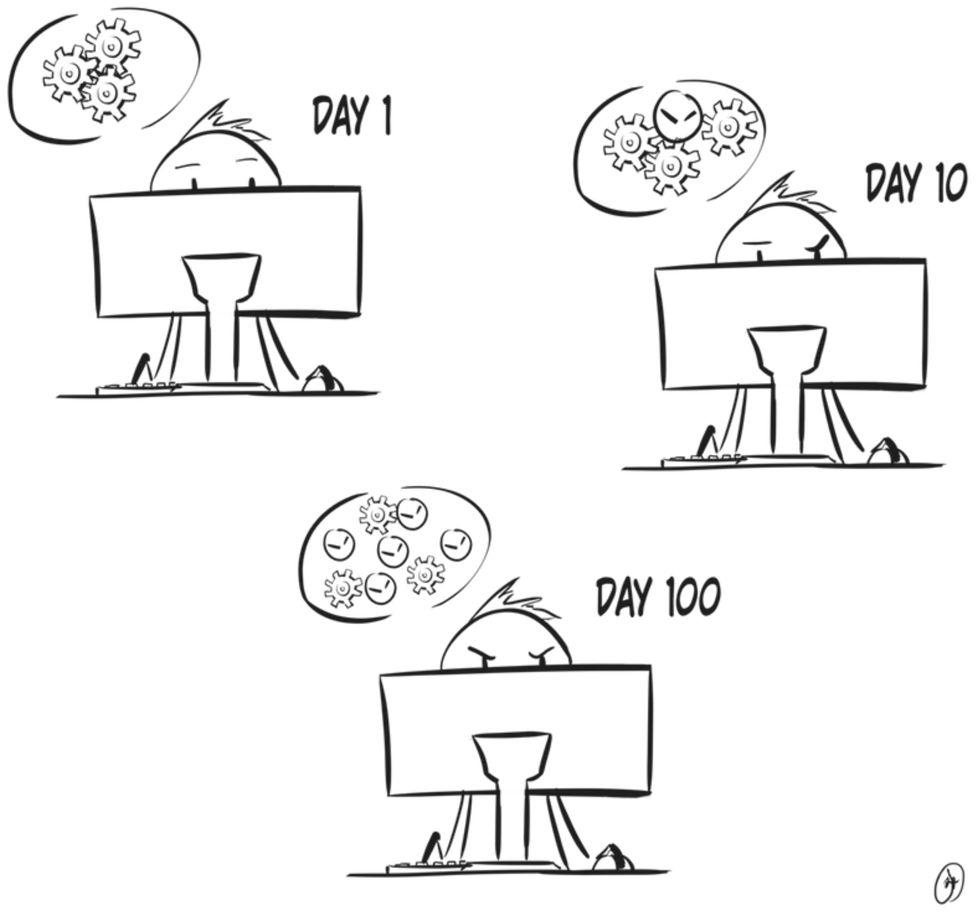


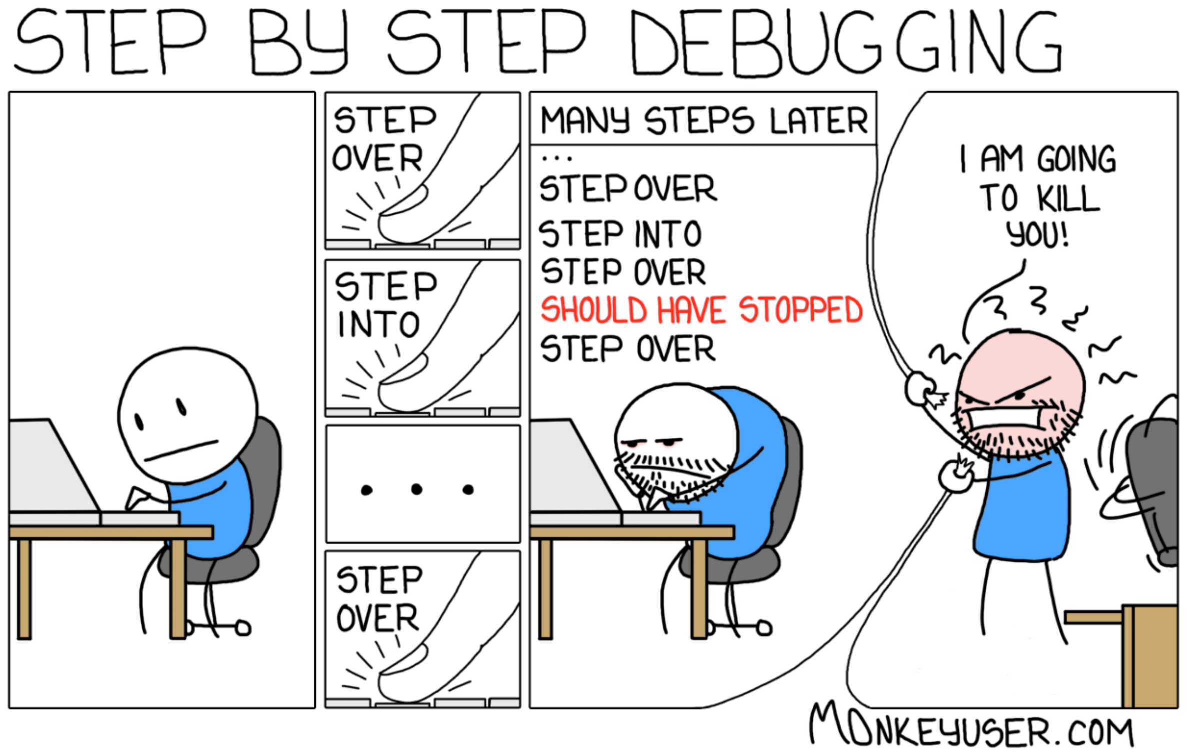
DEMO 3 NEU6620-BIG DATA PLATFORM (SP... 18 OCT 2019-25 OCT 2019

70% 80 total points 56 completed points 0 open tasks 0 closed tasks 0 iocaine doses

Spark Codebase

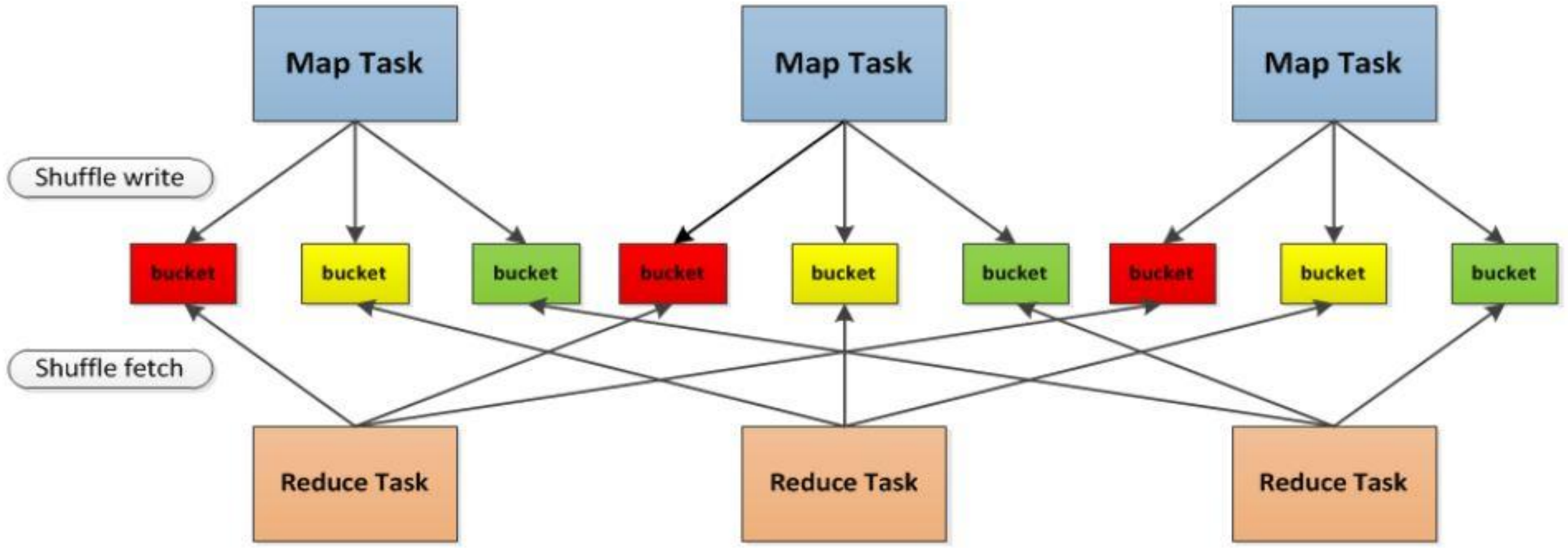
839,842 **lines of code**



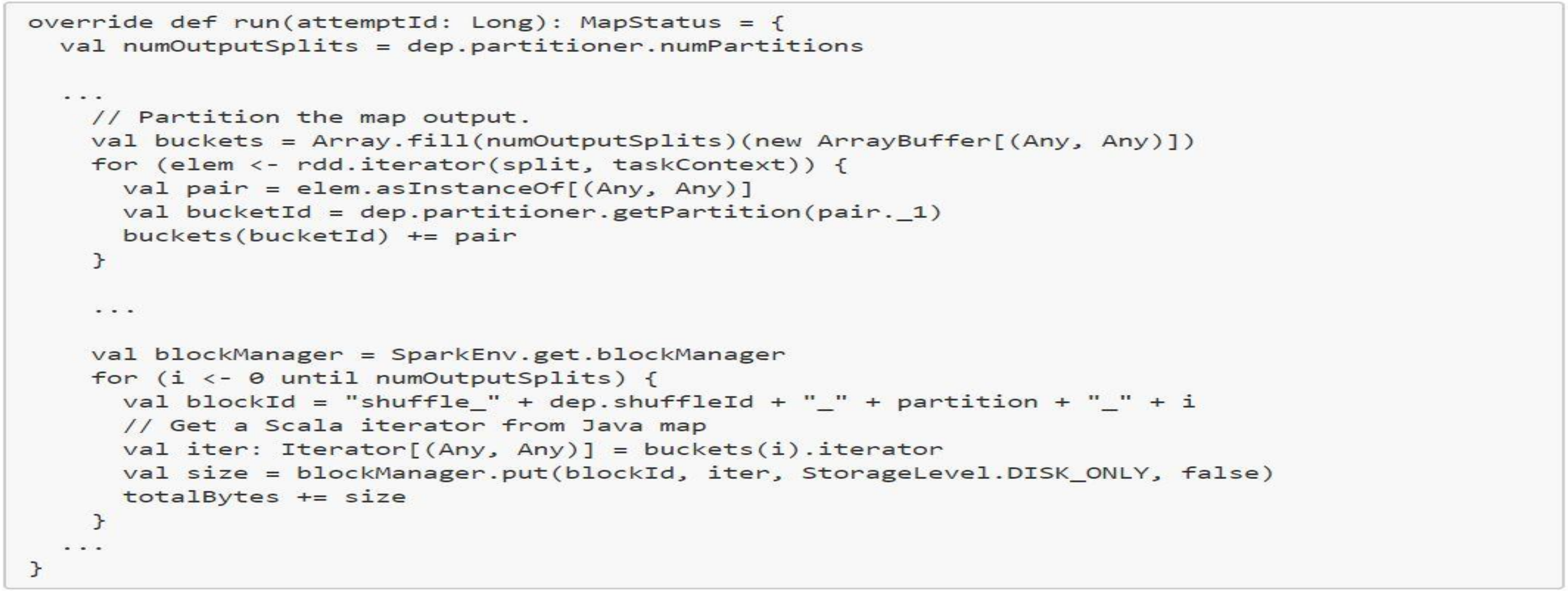


Challenges

Spark's Shuffle Phase



ShuffleWrite - Older Version



```scala
override def run(attemptId: Long): MapStatus = {
  val numOutputSplits = dep.partitioner.numPartitions

  ...
    // Partition the map output.
    val buckets = Array.fill(numOutputSplits)(new ArrayBuffer[(Any, Any)])
    for (elem <- rdd.iterator(split, taskContext)) {
      val pair = elem.asInstanceOf[(Any, Any)]
      val bucketId = dep.partitioner.getPartition(pair._1)
      buckets(bucketId) += pair
    }

    ...

    val blockManager = SparkEnv.get.blockManager
    for (i <- 0 until numOutputSplits) {
      val blockId = "shuffle_" + dep.shuffleId + "_" + partition + "_" + i
      // Get a Scala iterator from Java map
      val iter: Iterator[(Any, Any)] = buckets(i).iterator
      val size = blockManager.put(blockId, iter, StorageLevel.DISK_ONLY, false)
      totalBytes += size
    }
  ...
}
```

Problems with it

1. Spark will first save all the data in memory and then dump it into disk

2. Each mapper will generate a small file for each reducer. I/O will be significantly slowed when transfer large amount of small pieces of files.

ShuffleBlockManager

Spark then introduces ShuffleBlockManager, which assigns a DiskObjectWriter to each bucket, to manage the shuffling files .

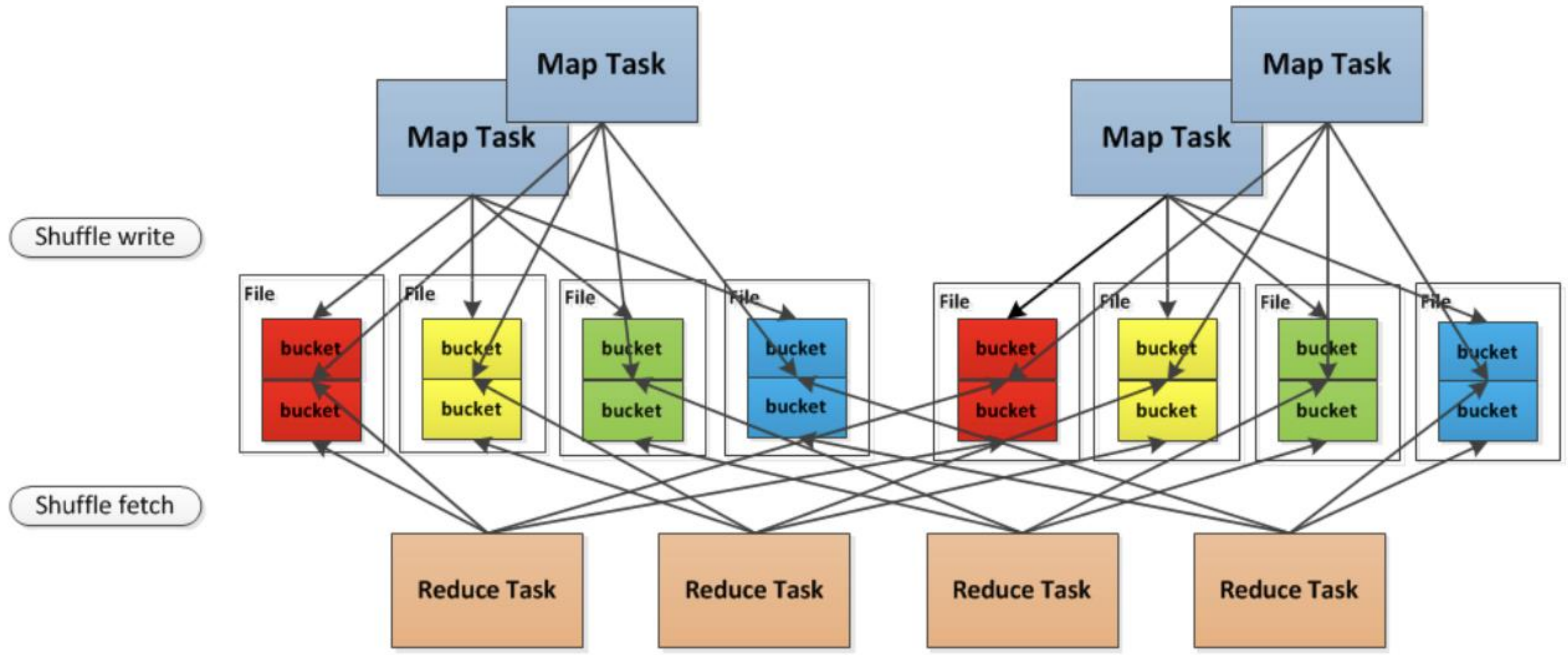
```
// Write the map output to its associated buckets.
for (elem <- rdd.iterator(split, taskContext)) {
  val pair = elem.asInstanceOf[Product2[Any, Any]]
  val bucketId = dep.partitioner.getPartition(pair._1)
  buckets.writers(bucketId).write(pair)
}
```

Problems still exist

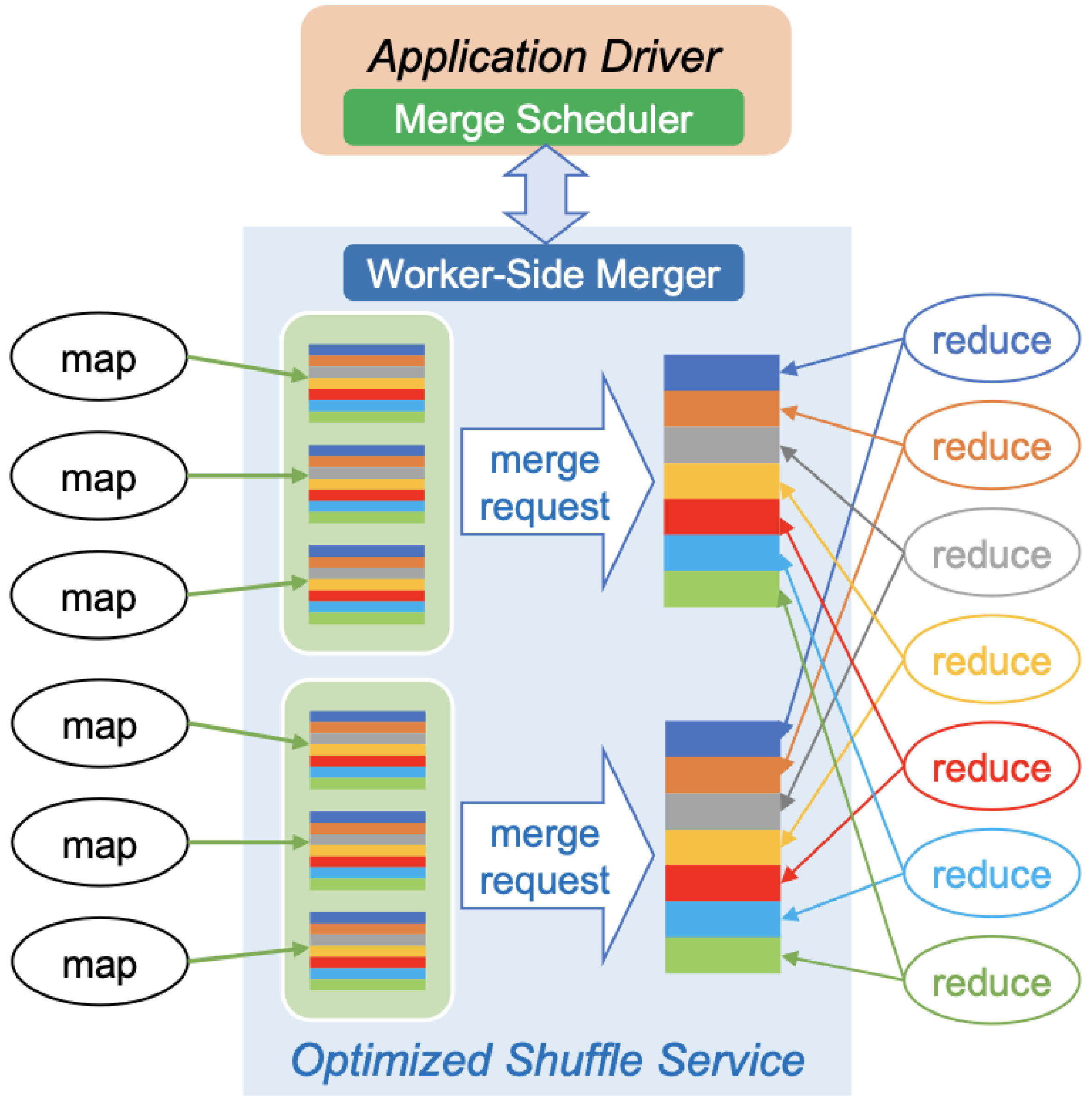
Problem with large amount of small files still exists.

Number of fix-sized writer buffer depends on the number of reducers. For a 8 core machine, 1k reducer requires 800 MB memory.

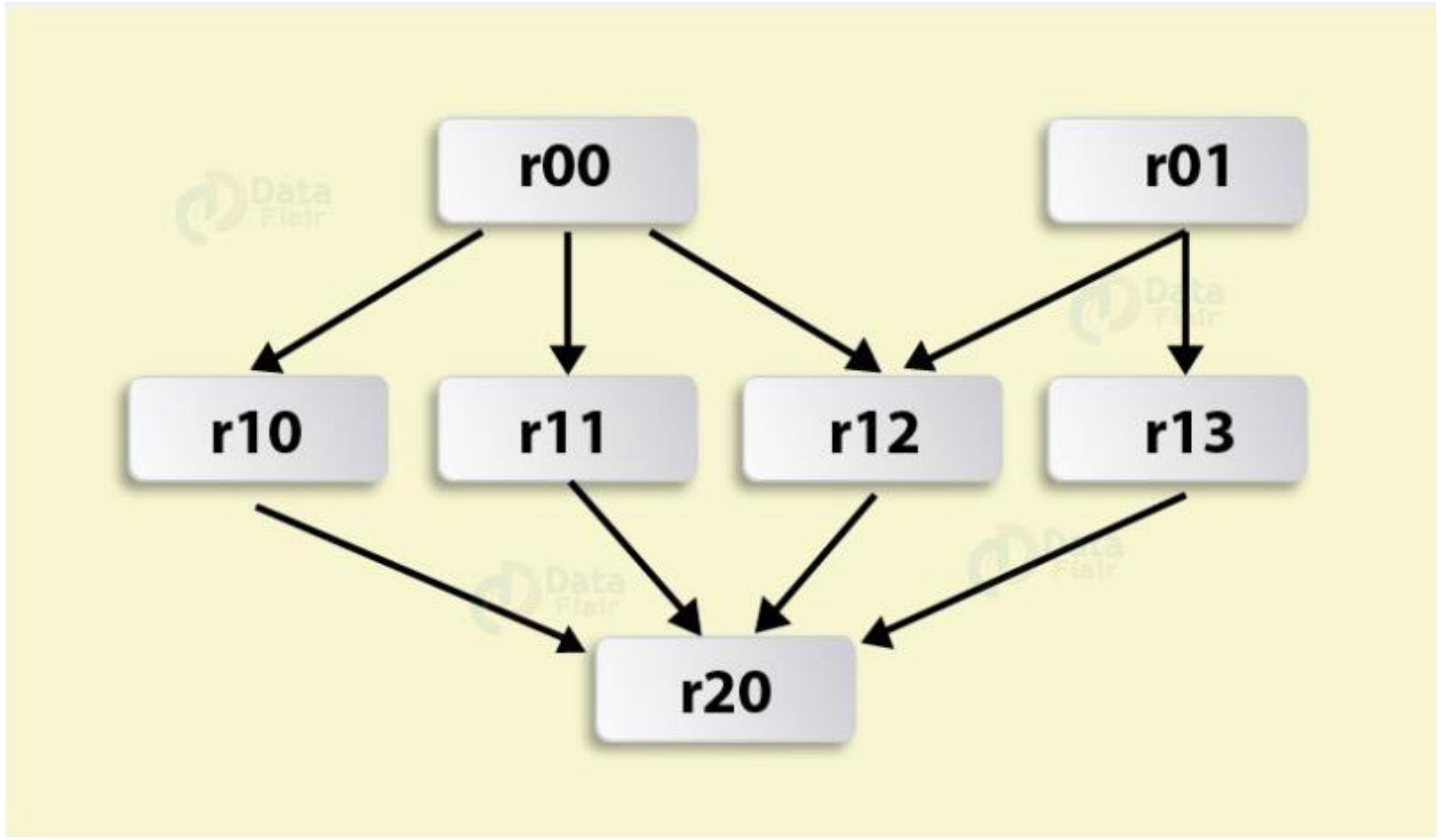
Shuffle consolidation



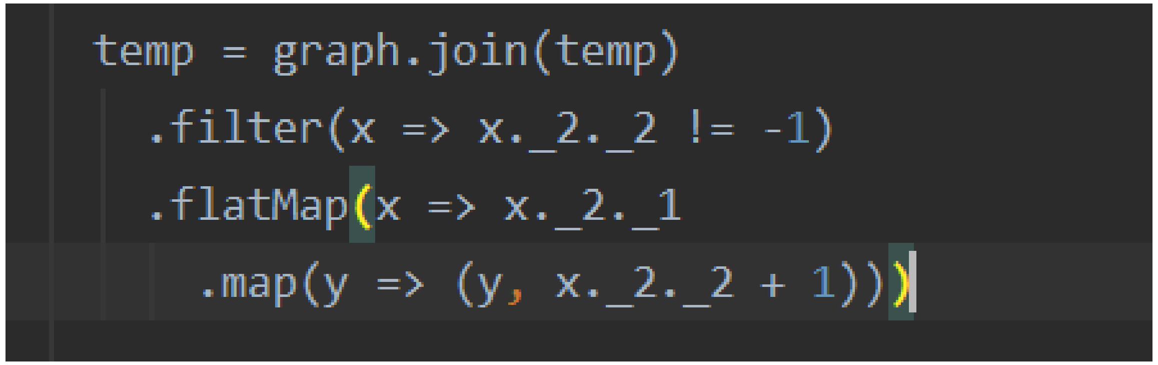
Go back to Riffle paper



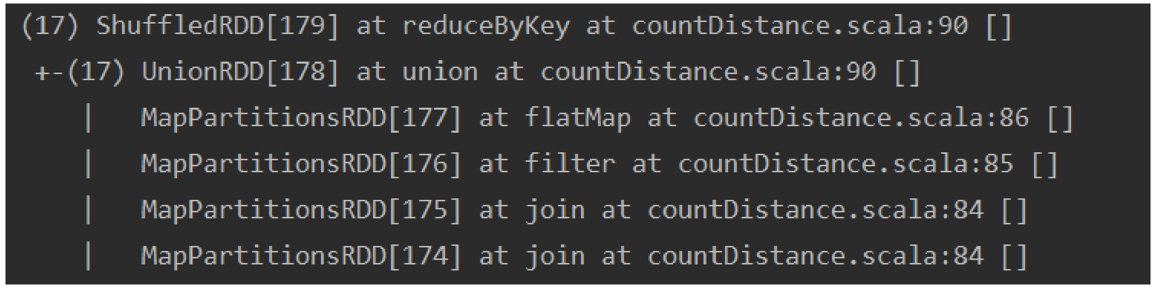
RDD Lineage



toDebugString()

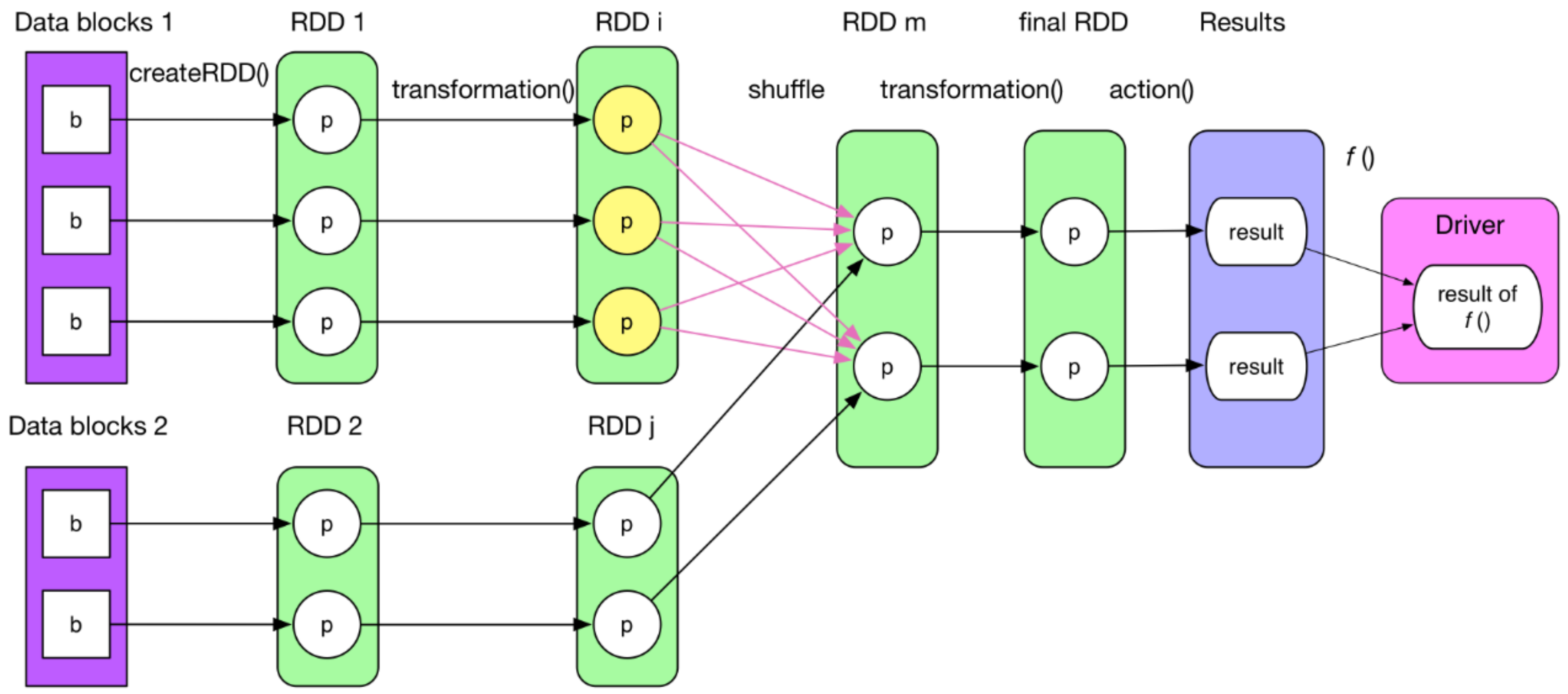


```
temp = graph.join(temp)
  .filter(x => x._2._2 != -1)
  .flatMap(x => x._2._1
    .map(y => (y, x._2._2 + 1)))
```

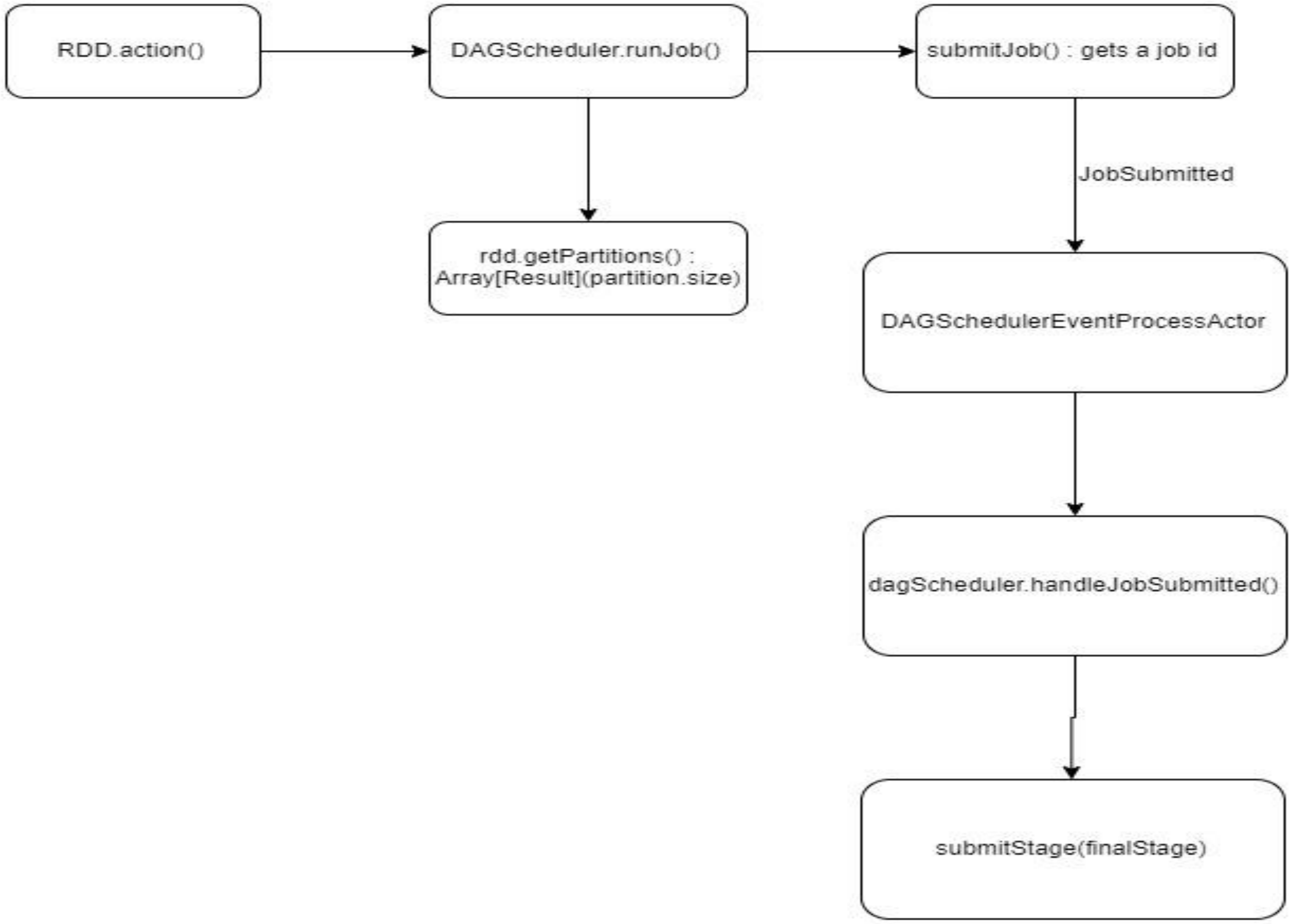


```
(17) ShuffledRDD[179] at reduceByKey at countDistance.scala:90 []
 +-(17) UnionRDD[178] at union at countDistance.scala:90 []
    |   MapPartitionsRDD[177] at flatMap at countDistance.scala:86 []
    |   MapPartitionsRDD[176] at filter at countDistance.scala:85 []
    |   MapPartitionsRDD[175] at join at countDistance.scala:84 []
    |   MapPartitionsRDD[174] at join at countDistance.scala:84 []
```

Transformation and action



Demo


RDD.action()
DAGScheduler.runJob()
submitJob() : gets a job id
rdd.getPartitions() : Array[Result](partition.size)
JobSubmitted
DAGSchedulerEventProcessActor
dagScheduler.handleJobSubmitted()
submitStage(finalStage)

Next Sprint Goals

- Design strategies to implement N-Way merge algorithm
- Start implementing the N-Way merge algorithm.

Any Questions?

Thank You!