data = sc.textFile("/data/t1.txt",3).cache() #这里加上cache，m1的操作就是PROCESS\_LOCAL数据本地性，Stages界面可以看到，就不会有/tmp问题，不cache()就会有/tmp问题。

data.glom().collect()

m1 = data.groupBy(lambda x: x.split(" ")[0],2).cache()

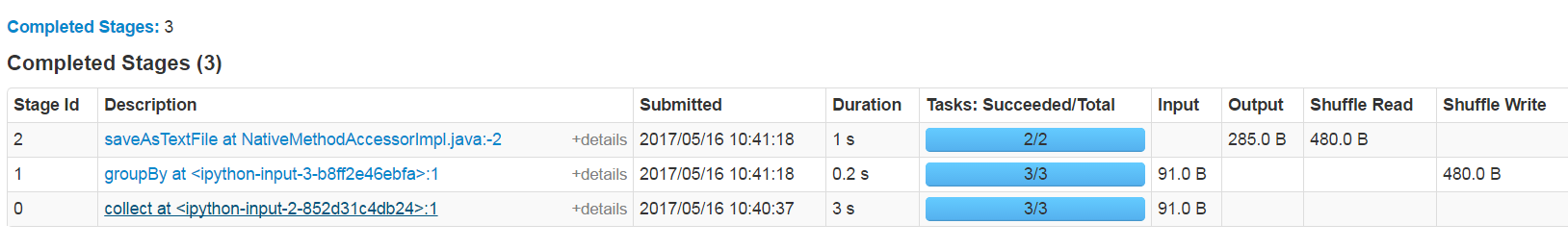
m1.saveAsTextFile("/data/sort004")

当前stage的task数是指调用了shuffle操作的那个RDD的分区。比如stage1的task数是data的分区数，而非groupBy()中的分区数，groupBy()是在具体是在下个stage指定分区执行的，m1的分区数是groupBy()中指定的分区数。

当前Stage的Description无非是Action和shuffle算子，shuffle算子有返回RDD，这个RDD在shuffle算子中可以指定分区，shuffle算子的操作具体在第二个Stage运行，首先看shuffle指定了几个分区，就启动几个task，去进行shuffle read，读取数据后进行，groupBy等等shuffle操作。

Action：

first(),take(),count(),takeSample(),takeOrdered(n, [ordering]),collect(),reduce(),countByKey(),foreach(func),saveAsTextFile(),saveAsNewAPIHadoopDataset().



适当的cache()会避免/tmp问题！

GC时，Executor、task会停止，因此调整spark.shuffle.io.maxRetries和spark.shuffle.io.retryWait是很有必要的！

解决/tmp问题：

1. 尽量避免groupBy()等操作，拆分操作。
2. 合理进行cache()
3. 设置Executors,在一个节点上。
4. 调整三个参数：（参数不起作用？）

.set("spark.shuffle.file.buffer","32k")

.set("spark.shuffle.io.maxRetries","10")

.set("spark.shuffle.io.retryWait","4s")

GC会影响/tmp问题：<http://blog.csdn.net/duan_zhihua/article/details/51901207>

Cdh5不支持sparkR：https://www.cloudera.com/documentation/enterprise/latest/topics/spark.html

