cloudera
Ask Bigger Questions

Why your Spark job is failing

Sandy Ryza

Me

- Data science at Cloudera
- Recently lead Apache Spark development at Cloudera
- Before that, committing on Apache YARN and MapReduce
- Hadoop project management committee

```
com.esotericsoftware.kryo.
KryoException: Unable to find class:
$iwC$$iwC$$iwC$$iwC$$iwC$$iwC$$iwC$$iwC
$$iwC$$iwC$$anonfun$4$$anonfun$apply$3
```

Lightning-Fast Cluster Computing

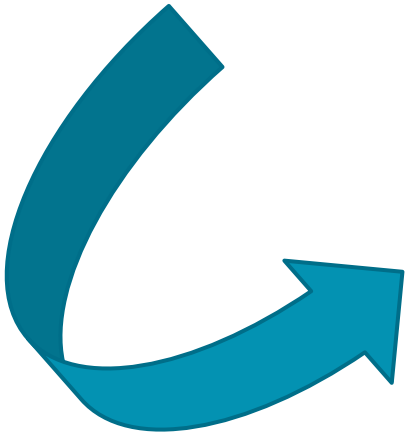

BREAKING NEWS
7:32 39°
Manhole Explosions
abc 7
UPPER WEST SIDE
EYEWITNESS NEWS

```
org.apache.spark.SparkException: Job aborted due to stage failure: Task
0.0:0 failed 4 times, most recent failure: Exception failure in TID 6 on
host bottou02-10g.pa.cloudera.com: java.lang.ArithmeticException: / by
zero
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply$mcII$sp(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        scala.collection.Iterator$$anon$11.next(Iterator.scala:328)
        org.apache.spark.util.Utils$.getIteratorSize(Utils.scala:1016)
        [...]
 Driver stacktrace:
    at org.apache.spark.scheduler.DAGScheduler.
org$apache$spark$scheduler$DAGScheduler$$failJobAndIndependentStages
(DAGScheduler.scala:1033)
    at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.
apply(DAGScheduler.scala:1017)
    at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.
apply(DAGScheduler.scala:1015)
    [...]
```

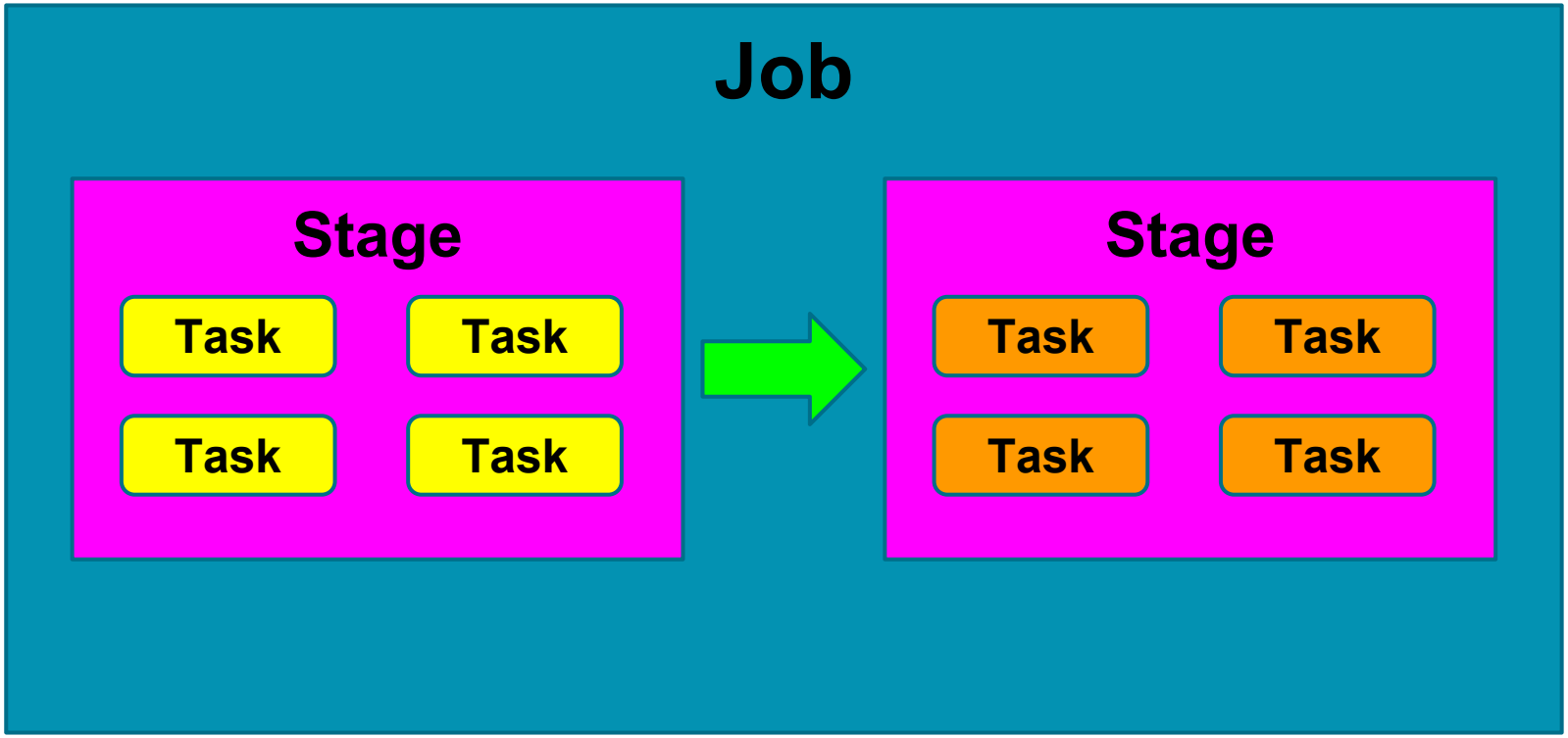
```
org.apache.spark.SparkException: Job aborted due to stage failure: Task
0.0:0 failed 4 times, most recent failure: Exception failure in TID 6 on
host bottou02-10g.pa.cloudera.com: java.lang.ArithmeticException: / by
zero
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply$mcII$sp(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        scala.collection.Iterator$$anon$11.next(Iterator.scala:328)
        org.apache.spark.util.Utils$.getIteratorSize(Utils.scala:1016)
        [...]
```

```
org.apache.spark.SparkException: Job aborted due to stage failure: Task
0.0:0 failed 4 times, most recent failure: Exception failure in TID 6 on
host bottou02-10g.pa.cloudera.com: java.lang.ArithmeticException: / by
zero
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply$mcII$sp(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        scala.collection.Iterator$$anon$11.next(Iterator.scala:328)
        org.apache.spark.util.Utils$.getIteratorSize(Utils.scala:1016)
        [...]
```

```
val file = sc.textFile("hdfs://...")
file.filter(_.startsWith(“banana”))
    .count()
```



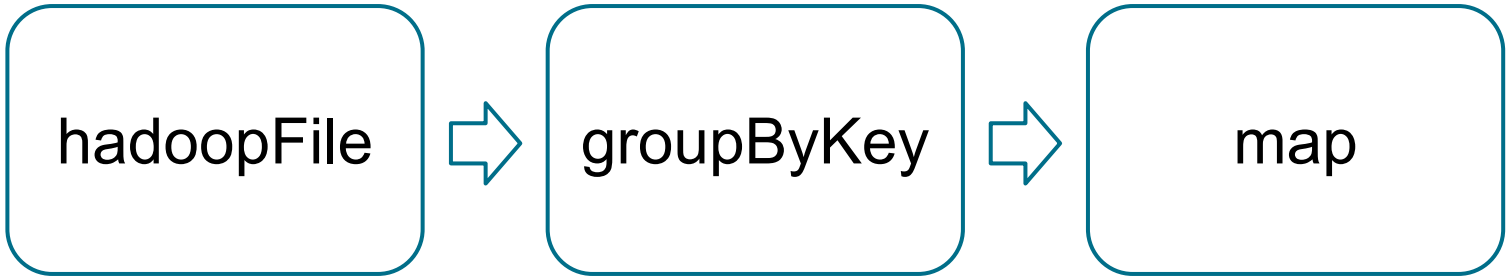
Anatomy of a job



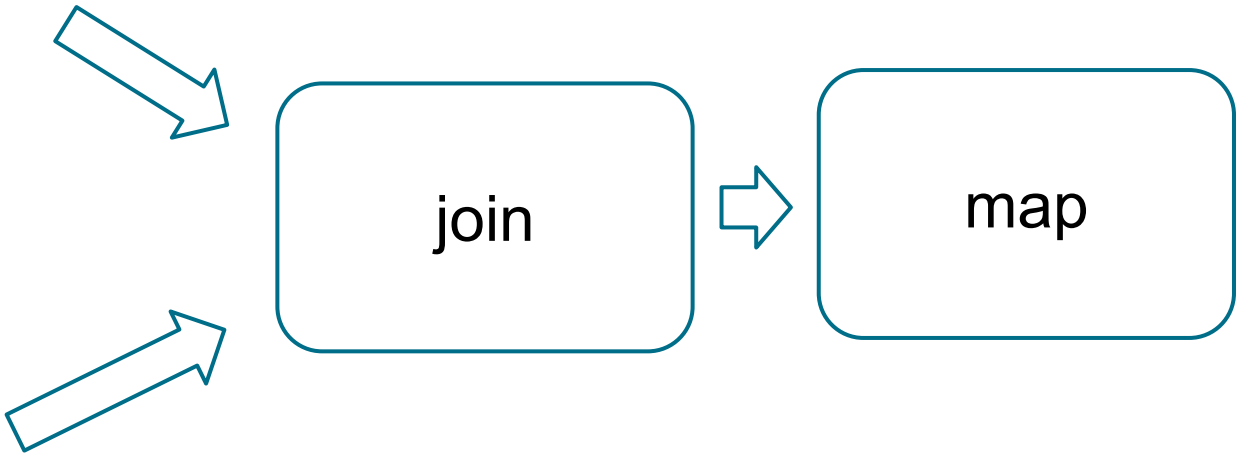
What the heck is a stage?

```
val rdd1 = sc.textFile("hdfs://...")
             .map(someFunc)
             .filter(filterFunc)
```

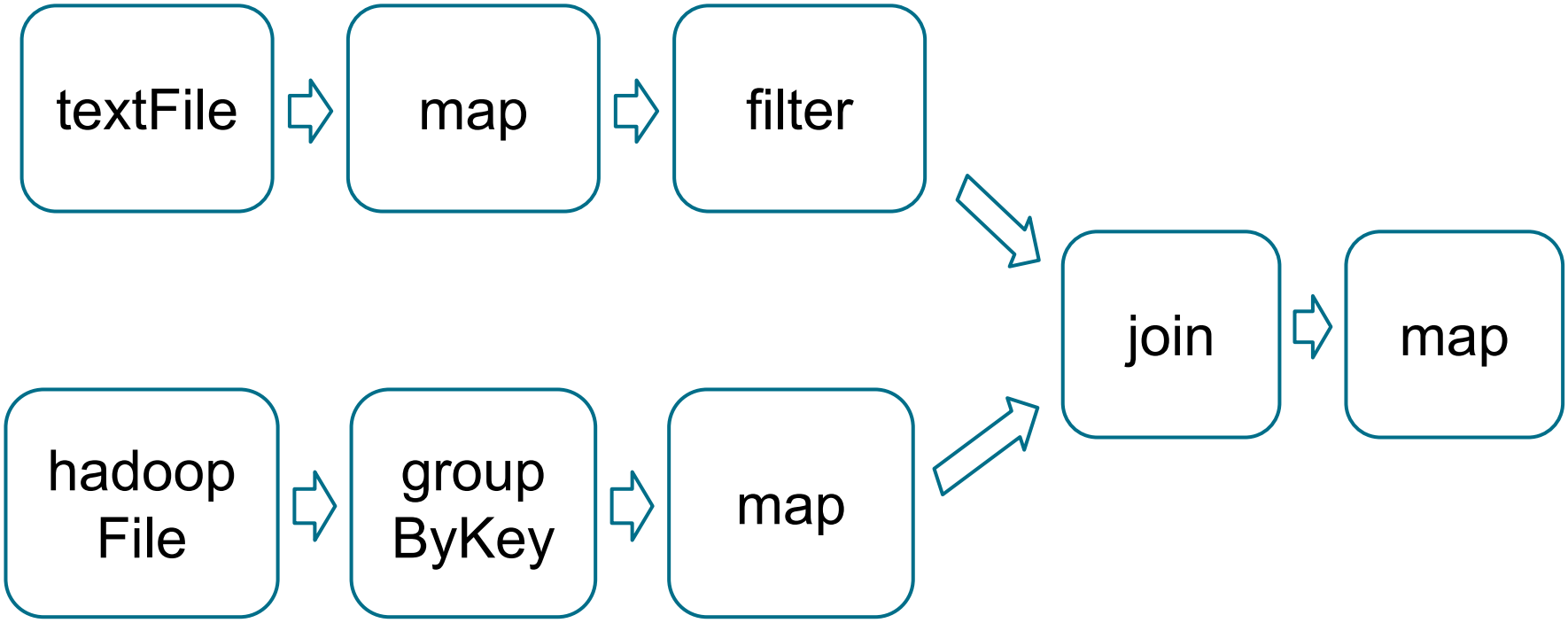
```
val rdd2 = sc.hadoopFile("hdfs:
//...")
                .groupByKey()
                .map(someOtherFunc)
```

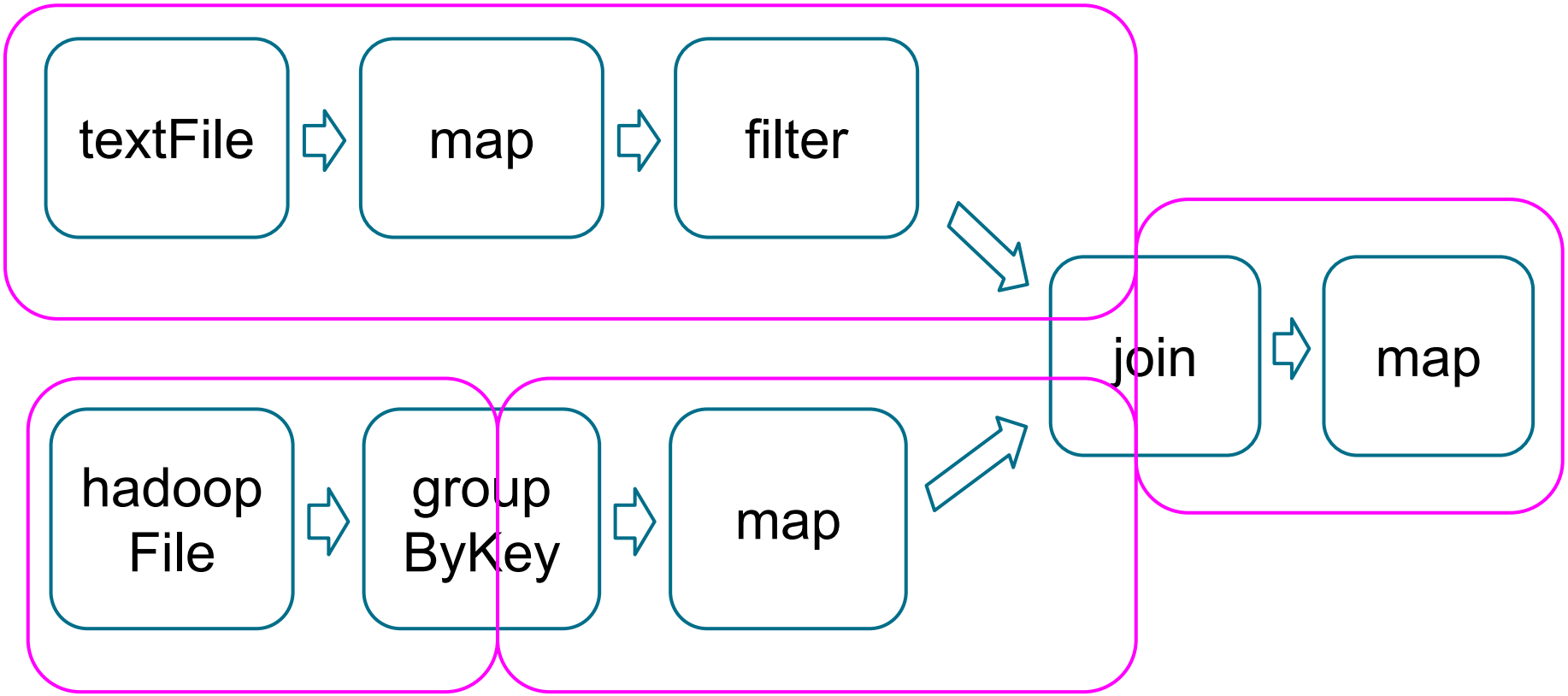


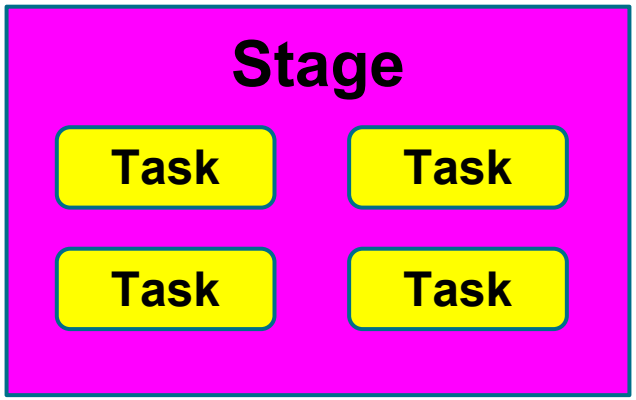
```
val rdd3 = rdd1.join(rdd2)
               .map(someFunc)
```

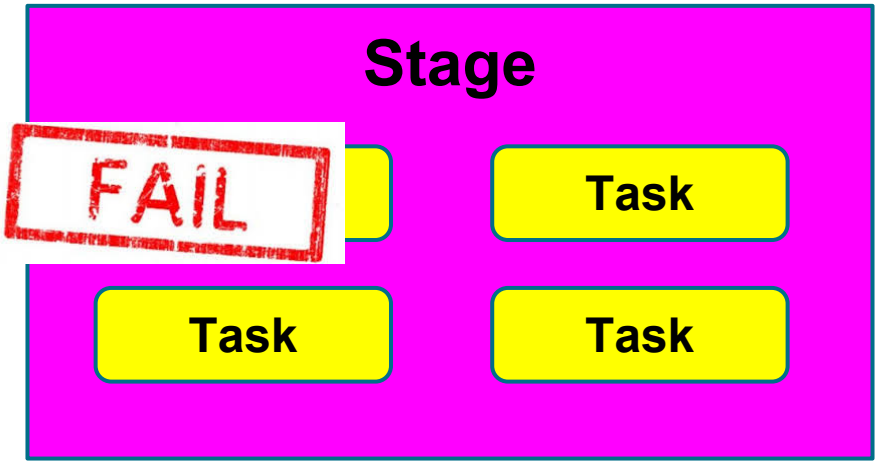


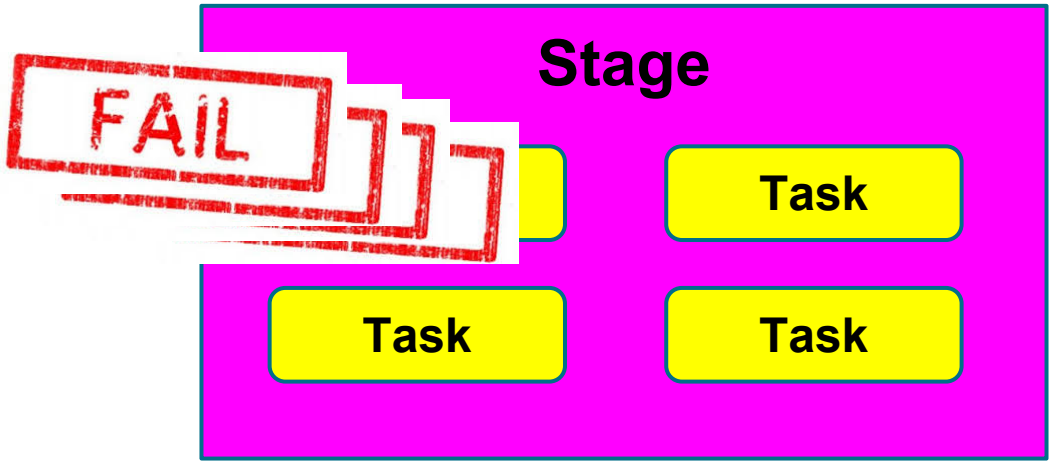
```
rdd3.collect()
```


textFile
map
filter
hadoop File
group ByKey
map
join
map


textFile
map
filter
hadoop File
group ByKey
map
join
map


Stage
Task
Task
Task
Task

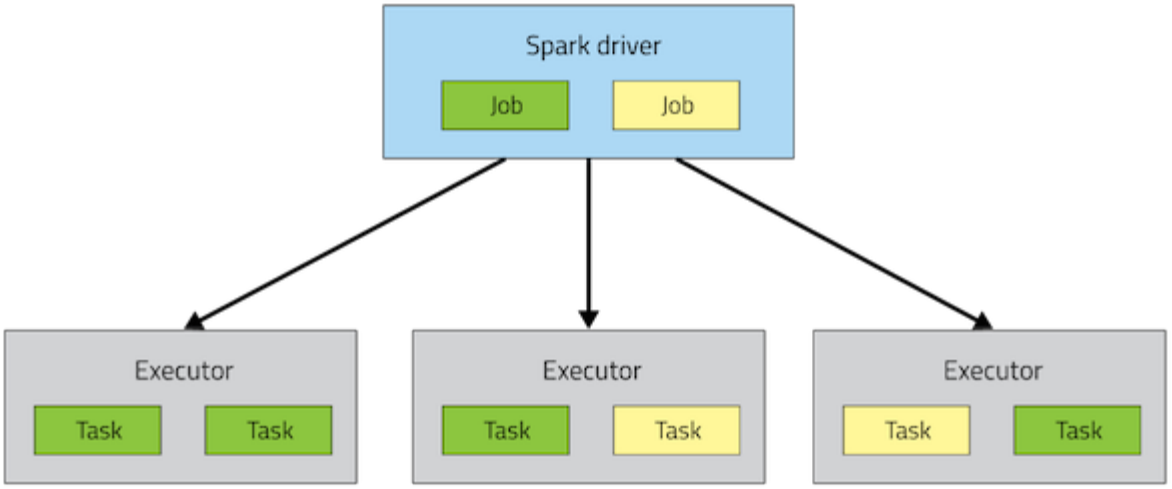

Stage
FAIL
Task
Task
Task


Stage
FAIL
Task
Task
Task

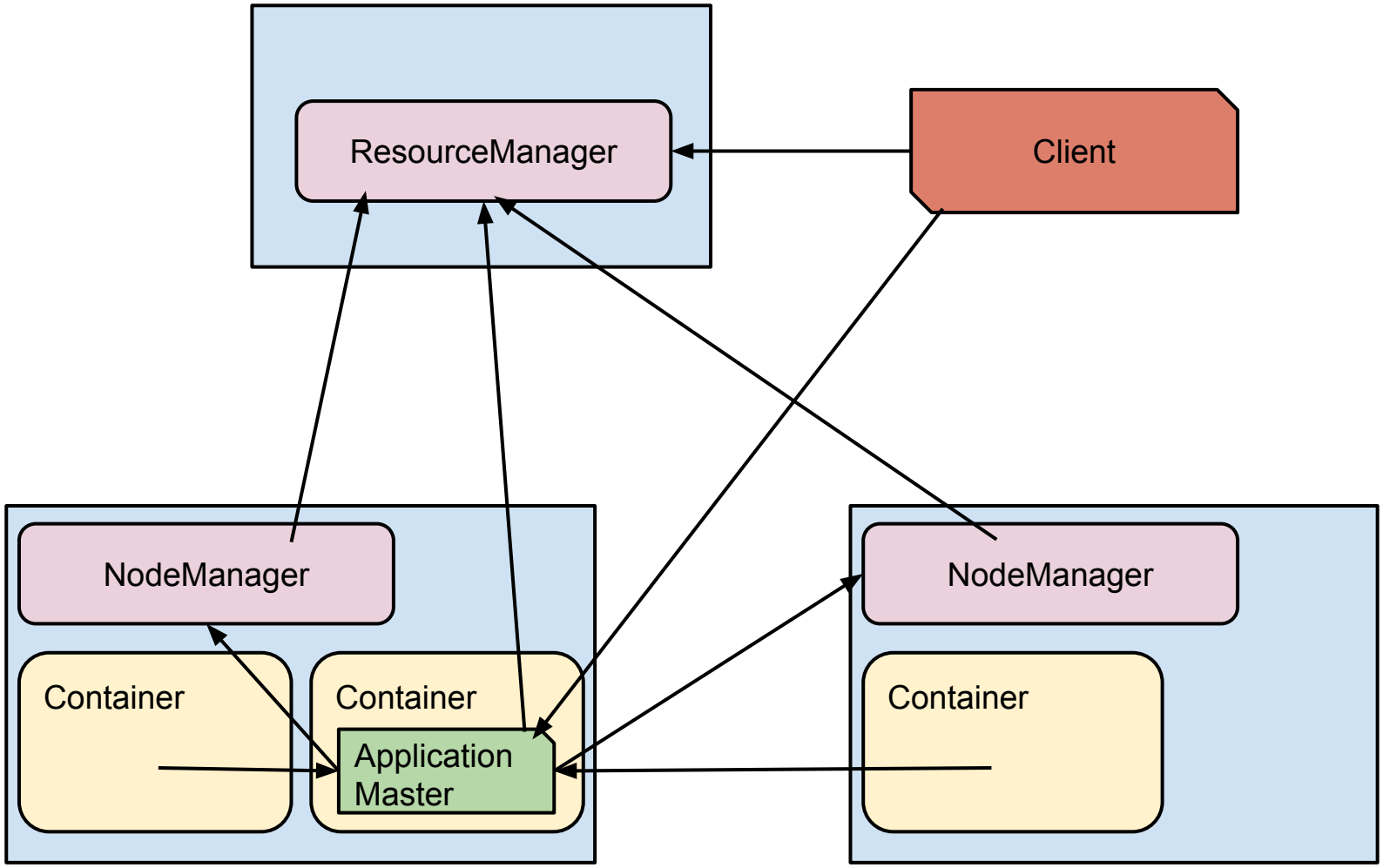
```
org.apache.spark.SparkException: Job aborted due to stage failure: Task
0.0:0 failed 4 times, most recent failure: Exception failure in TID 6 on
host bottou02-10g.pa.cloudera.com: java.lang.ArithmeticException: / by
zero
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply$mcII$sp(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        $iwC$$iwC$$iwC$$iwC$$anonfun$1.apply(<console>:13)
        scala.collection.Iterator$$anon$11.next(Iterator.scala:328)
        org.apache.spark.util.Utils$.getIteratorSize(Utils.scala:1016)
        [...]
```

```
14/04/22 11:59:58 ERROR executor.Executor: Exception in task ID 2866
java.io.IOException: Filesystem closed
        at org.apache.hadoop.hdfs.DFSClient.checkOpen(DFSClient.java:565)
        at org.apache.hadoop.hdfs.DFSInputStream.readWithStrategy(DFSInputStream.java:648)
        at org.apache.hadoop.hdfs.DFSInputStream.read(DFSInputStream.java:706)
        at java.io.DataInputStream.read(DataInputStream.java:100)
        at org.apache.hadoop.util.LineReader.readDefaultLine(LineReader.java:209)
        at org.apache.hadoop.util.LineReader.readLine(LineReader.java:173)
        at org.apache.hadoop.mapred.LineRecordReader.next(LineRecordReader.java:206)
        at org.apache.hadoop.mapred.LineRecordReader.next(LineRecordReader.java:45)
        at org.apache.spark.rdd.HadoopRDD$$anon$1.getNext(HadoopRDD.scala:164)
        at org.apache.spark.rdd.HadoopRDD$$anon$1.getNext(HadoopRDD.scala:149)
        at org.apache.spark.util.NextIterator.hasNext(NextIterator.scala:71)
        at org.apache.spark.InterruptibleIterator.hasNext(InterruptibleIterator.scala:27)
        at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:327)
        at scala.collection.Iterator$$anon$14.hasNext(Iterator.scala:388)
        at scala.collection.Iterator$$anon$14.hasNext(Iterator.scala:388)
        at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:327)
        at scala.collection.Iterator$class.foreach(Iterator.scala:727)
        at scala.collection.AbstractIterator.foreach(Iterator.scala:1157)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:161)
        at org.apache.spark.scheduler.ShuffleMapTask.runTask(ShuffleMapTask.scala:102)
        at org.apache.spark.scheduler.Task.run(Task.scala:53)
        at org.apache.spark.executor.Executor$TaskRunner$$anonfun$run$1.apply$mcV$sp(Executor.scala:211)
        at org.apache.spark.deploy.SparkHadoopUtil$$anon$1.run(SparkHadoopUtil.scala:42)
        at org.apache.spark.deploy.SparkHadoopUtil$$anon$1.run(SparkHadoopUtil.scala:41)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:415)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1408)
        at org.apache.spark.deploy.SparkHadoopUtil.runAsUser(SparkHadoopUtil.scala:41)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:176)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
        at java.lang.Thread.run(Thread.java:724)
```

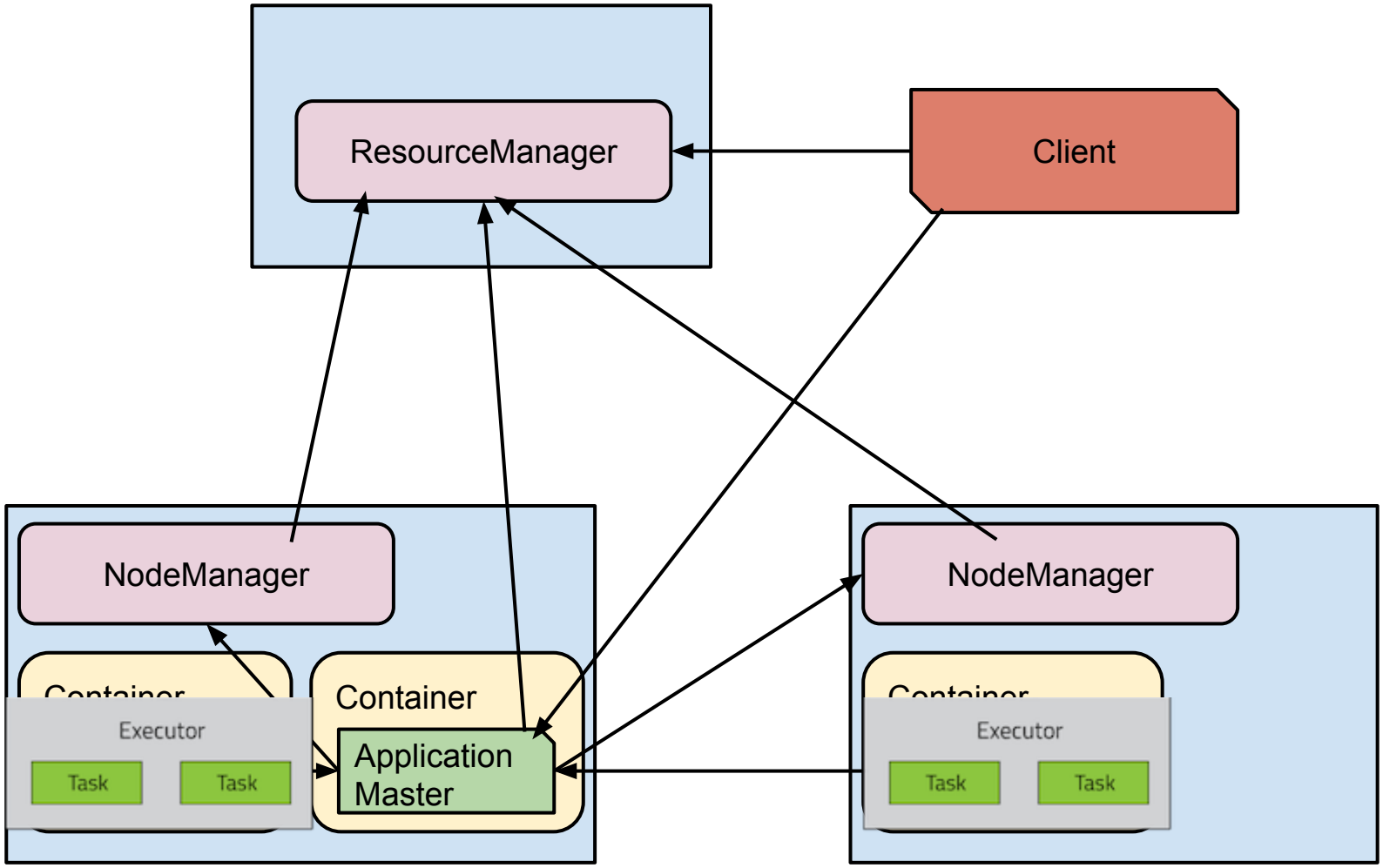
Spark architecture



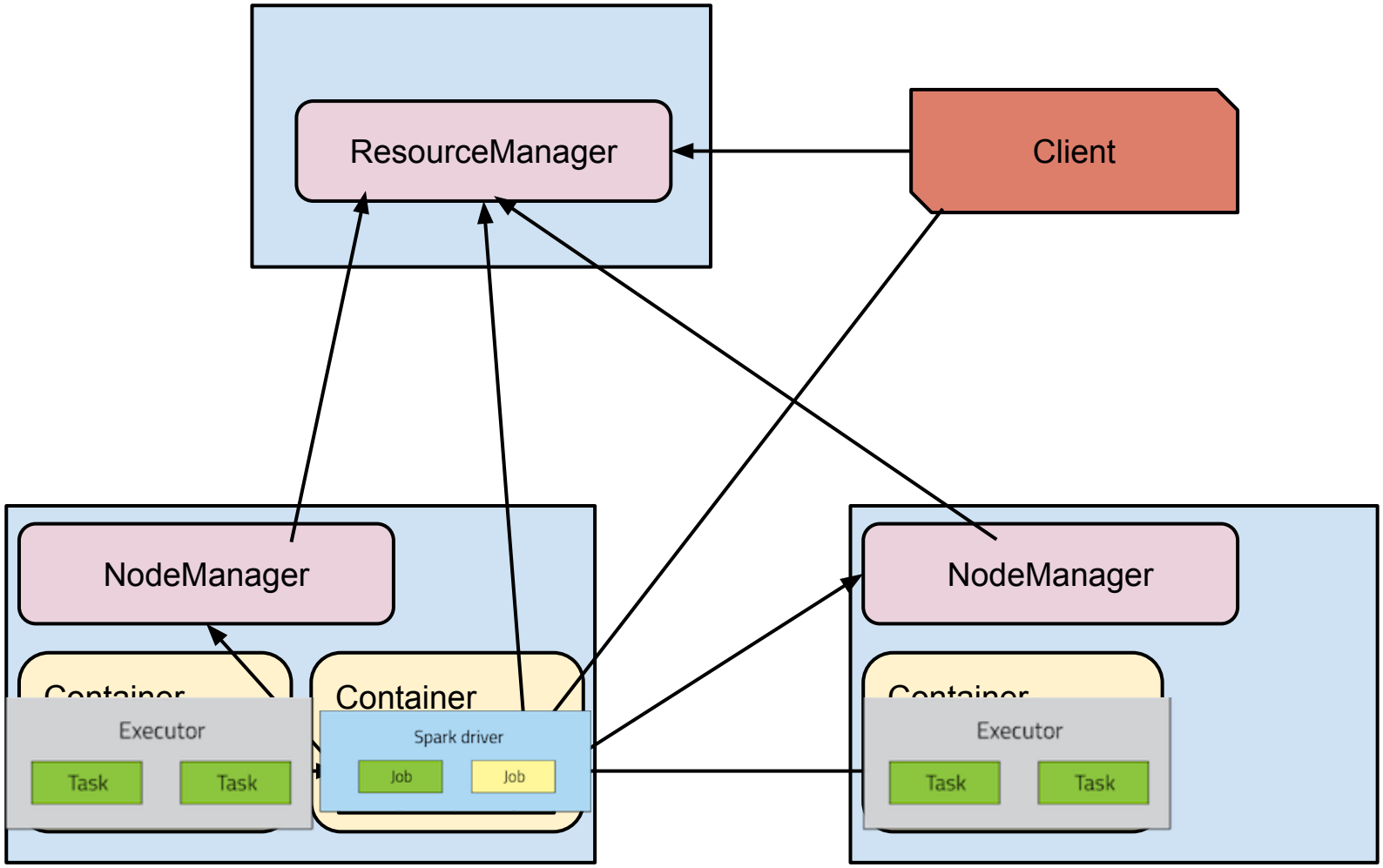
YARN architecture



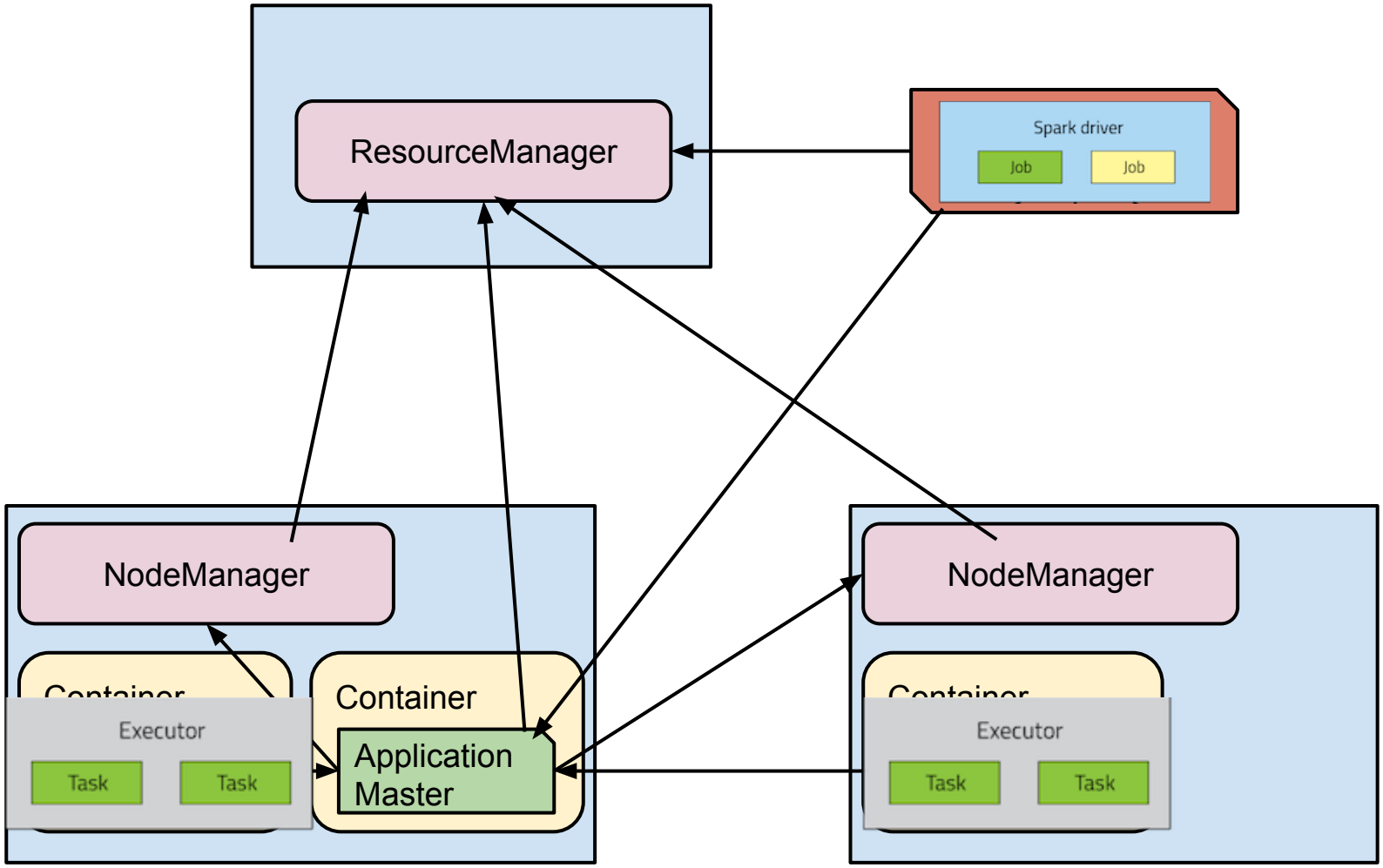
Spark on YARN architecture



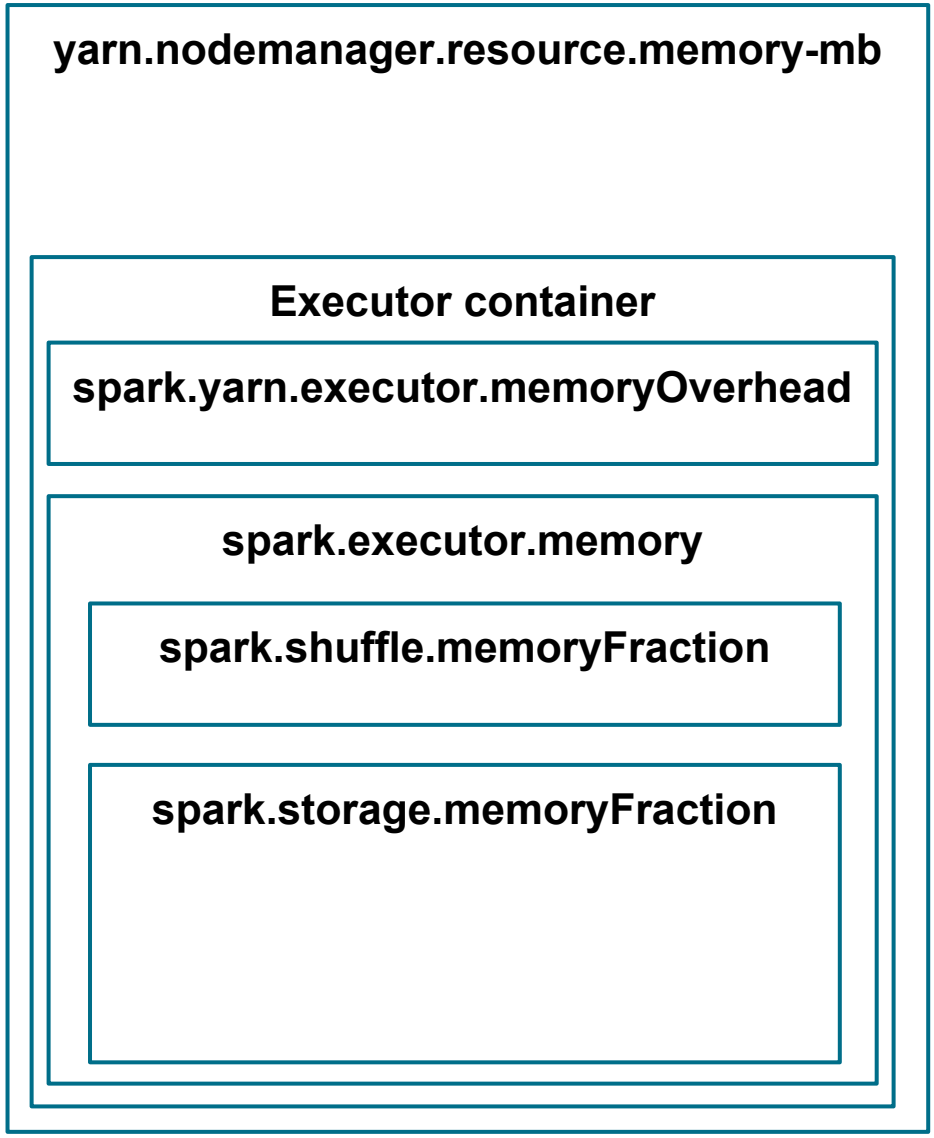
Spark on YARN architecture



Spark on YARN architecture



```
Container [pid=63375,
containerID=container_1388158490598_0001_01_00
0003] is running beyond physical memory
limits. Current usage: 2.1 GB of 2 GB physical
memory used; 2.8 GB of 4.2 GB virtual memory
used. Killing container.
```


yarn.nodemanager.resource.memory-mb
Executor container
spark.yarn.executor.memoryOverhead
spark.executor.memory
spark.shuffle.memoryFraction
spark.storage.memoryFraction

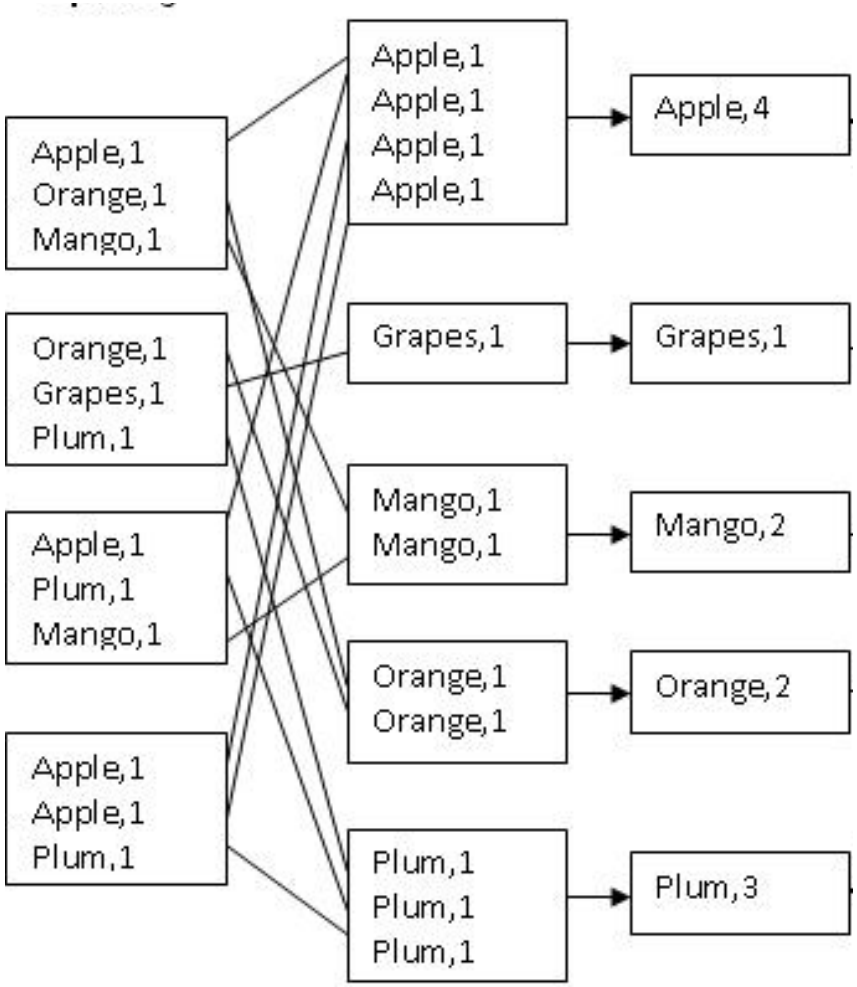

Dr. E. Godot, DDS

GC Stalls

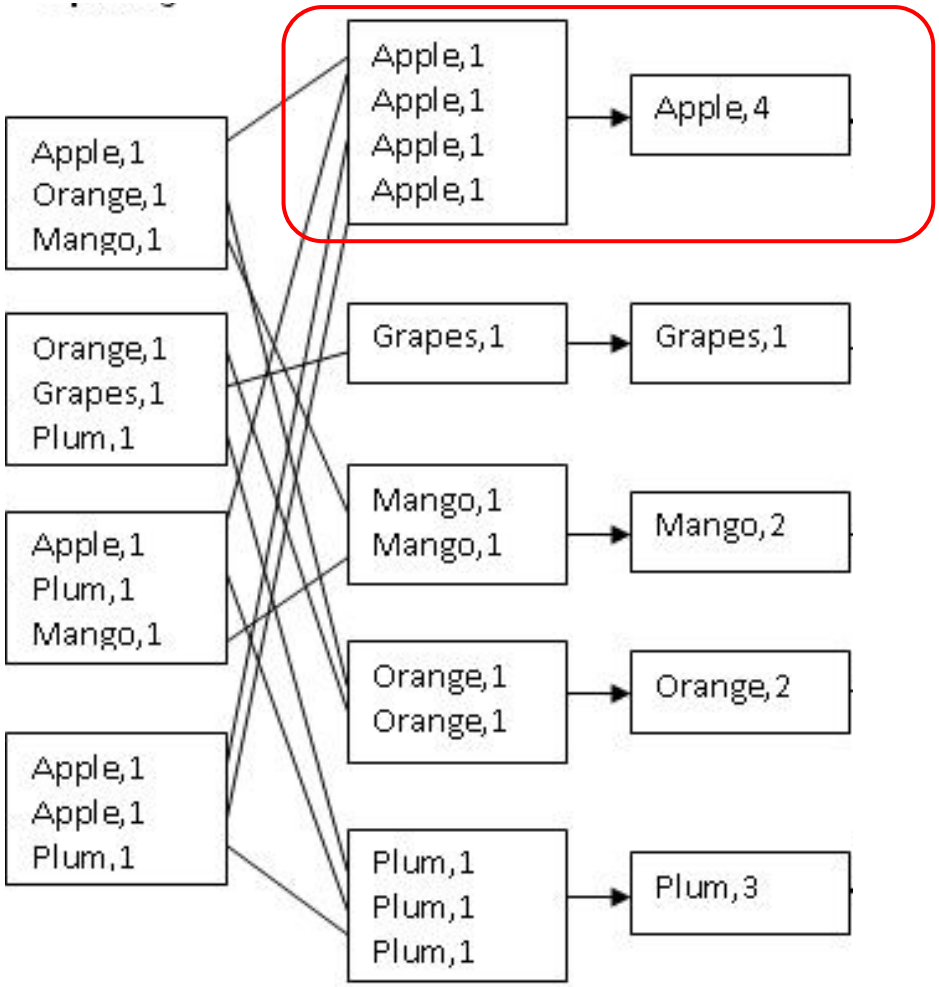
Task Index	Task ID	Status	Locality Level	Executor	Launch Time	Duration	GC Time
1	0	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.82 h	9.59 h
2	1	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.00 h	8.97 h
3	2	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.39 h	9.16 h
0	3	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.09 h	8.88 h
6	4	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	11.65 h	8.54 h
4	5	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	11.68 h	8.62 h
7	6	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.19 h	9.12 h
12	7	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	11.62 h	8.50 h
8	8	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.57 h	9.40 h
9	9	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.02 h	8.98 h
5	10	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.24 h	9.04 h
11	11	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	11.11 h	8.15 h
10	12	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	11.84 h	8.68 h
13	13	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	11.85 h	8.74 h
18	14	SUCCESS	NODE_LOCAL		2014/06/13 13:14:16	12.26 h	9.17 h

Too much spilling

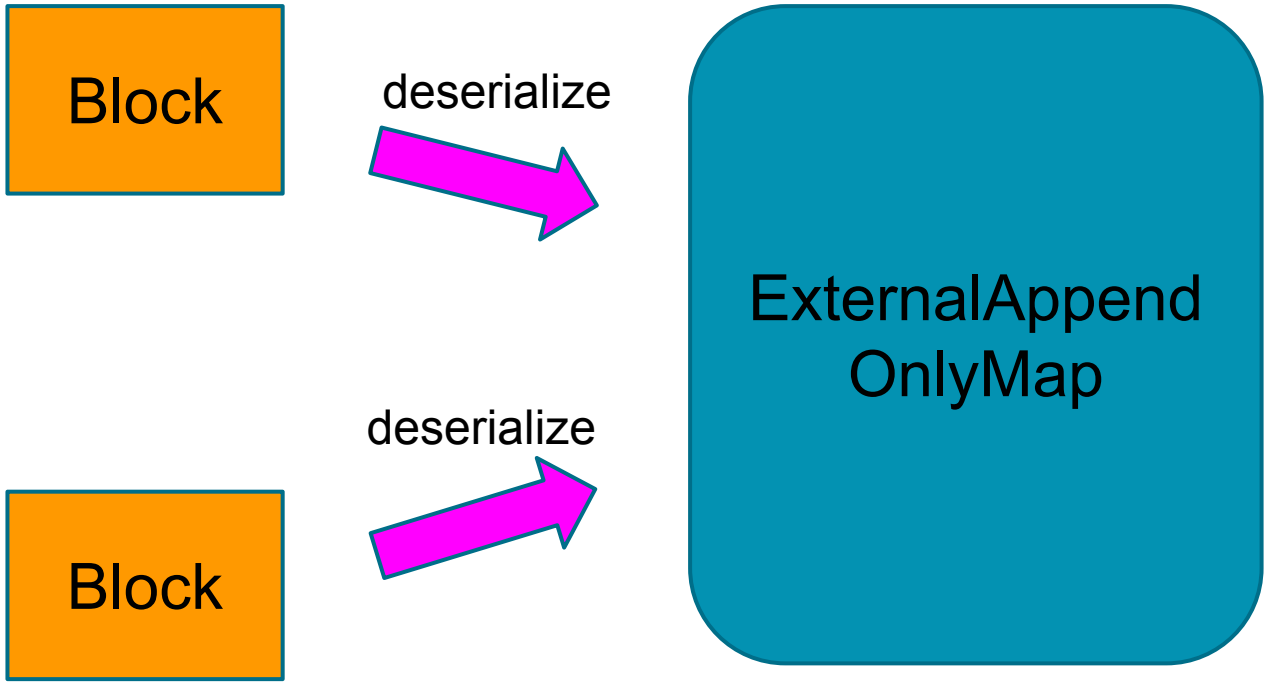
Most perf issues are in shuffles!



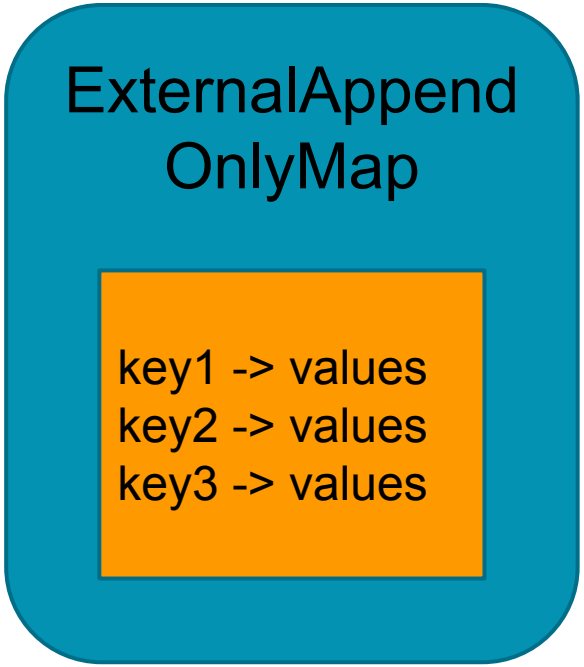
Most perf issues are in shuffles!



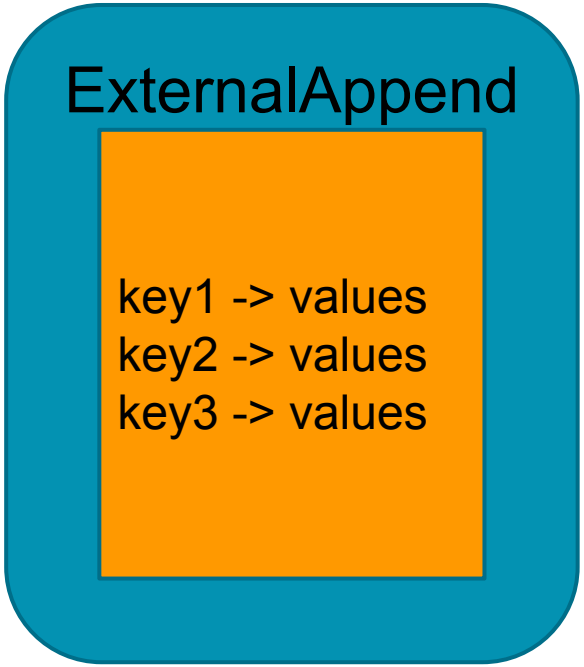
Inside a Task: Fetch & Aggregation



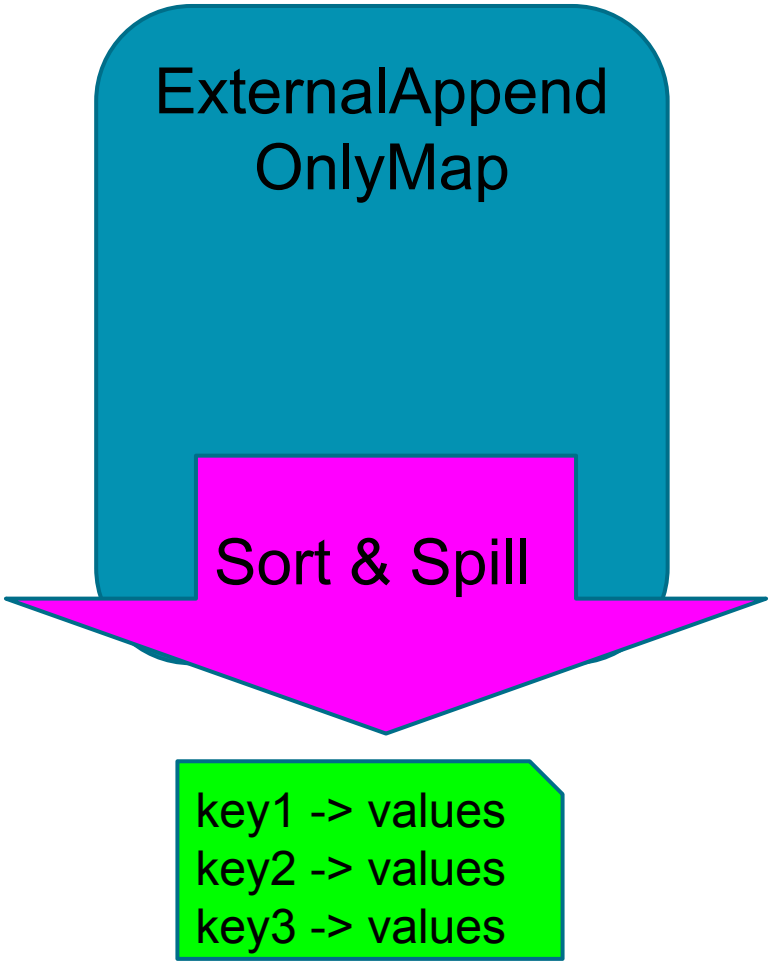
Inside a Task: Fetch & Aggregation



Inside a Task: Fetch & Aggregation



Inside a Task: Fetch & Aggregation

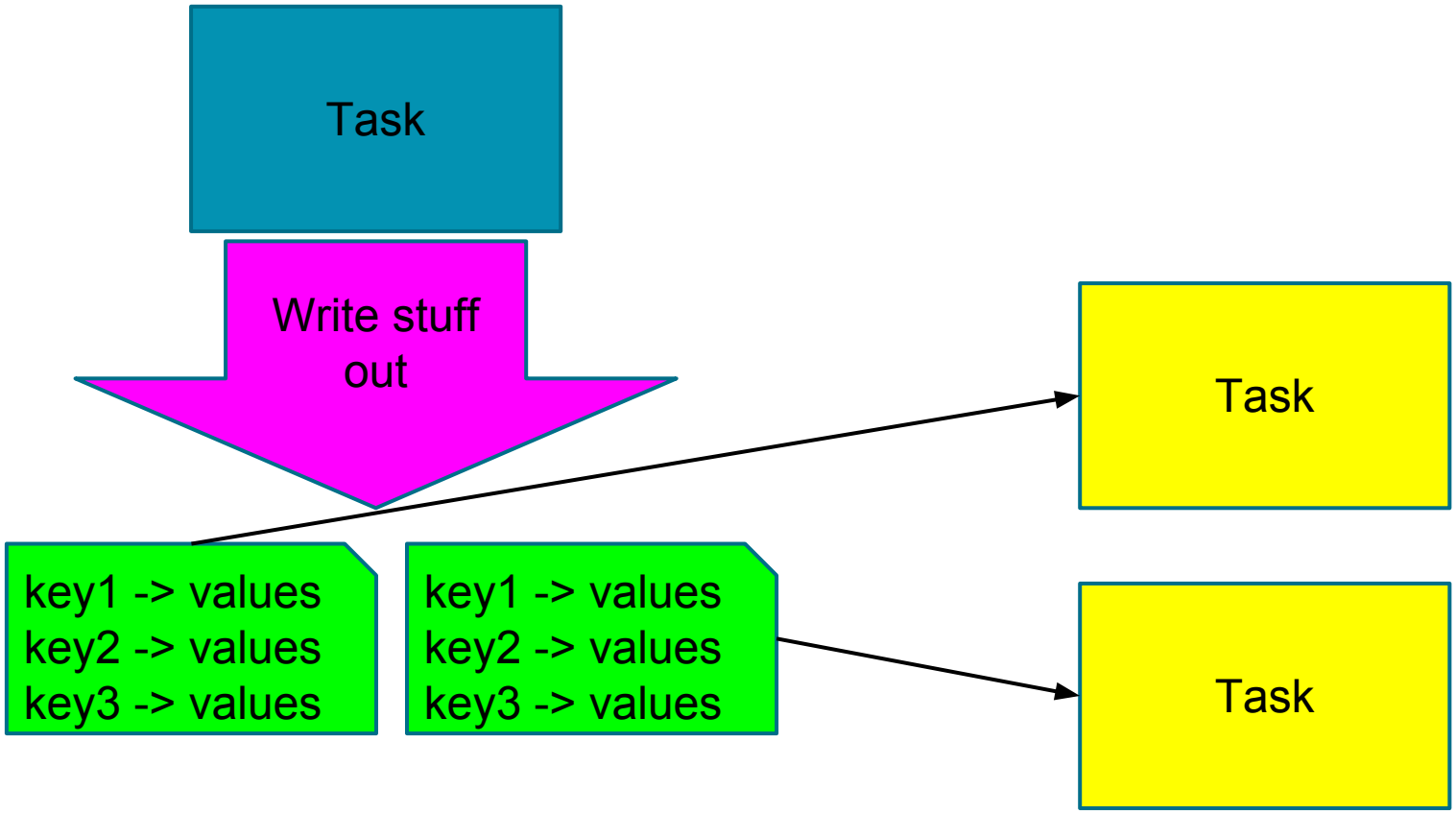


```
rdd.reduceByKey(reduceFunc,
                numPartitions=1000)
```

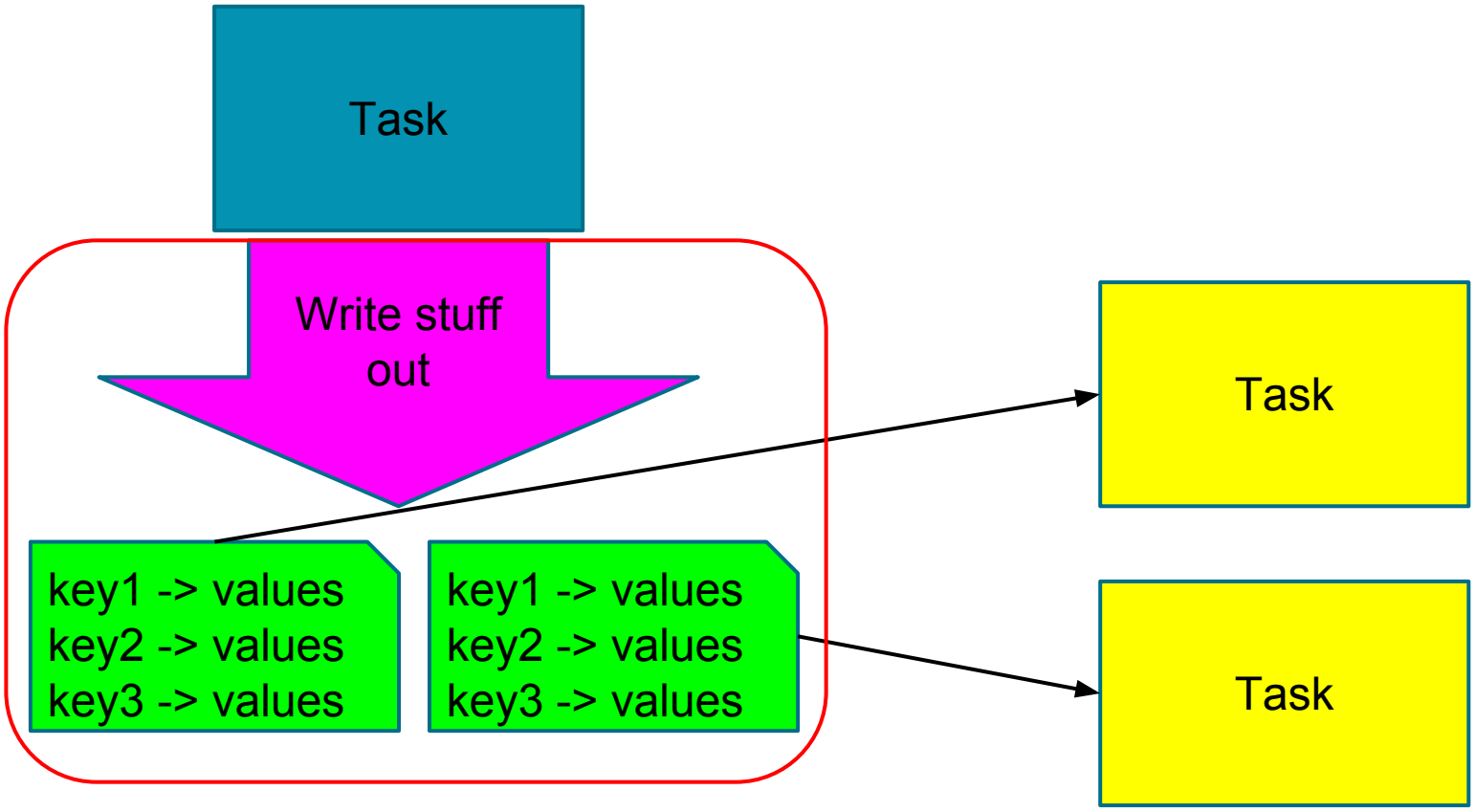
Too many open files!

```
java.io.FileNotFoundException:
/dn6/spark/local/spark-local-
20140610134115-
2cee/30/merged_shuffle_0_368_14 (Too many
open files)
```

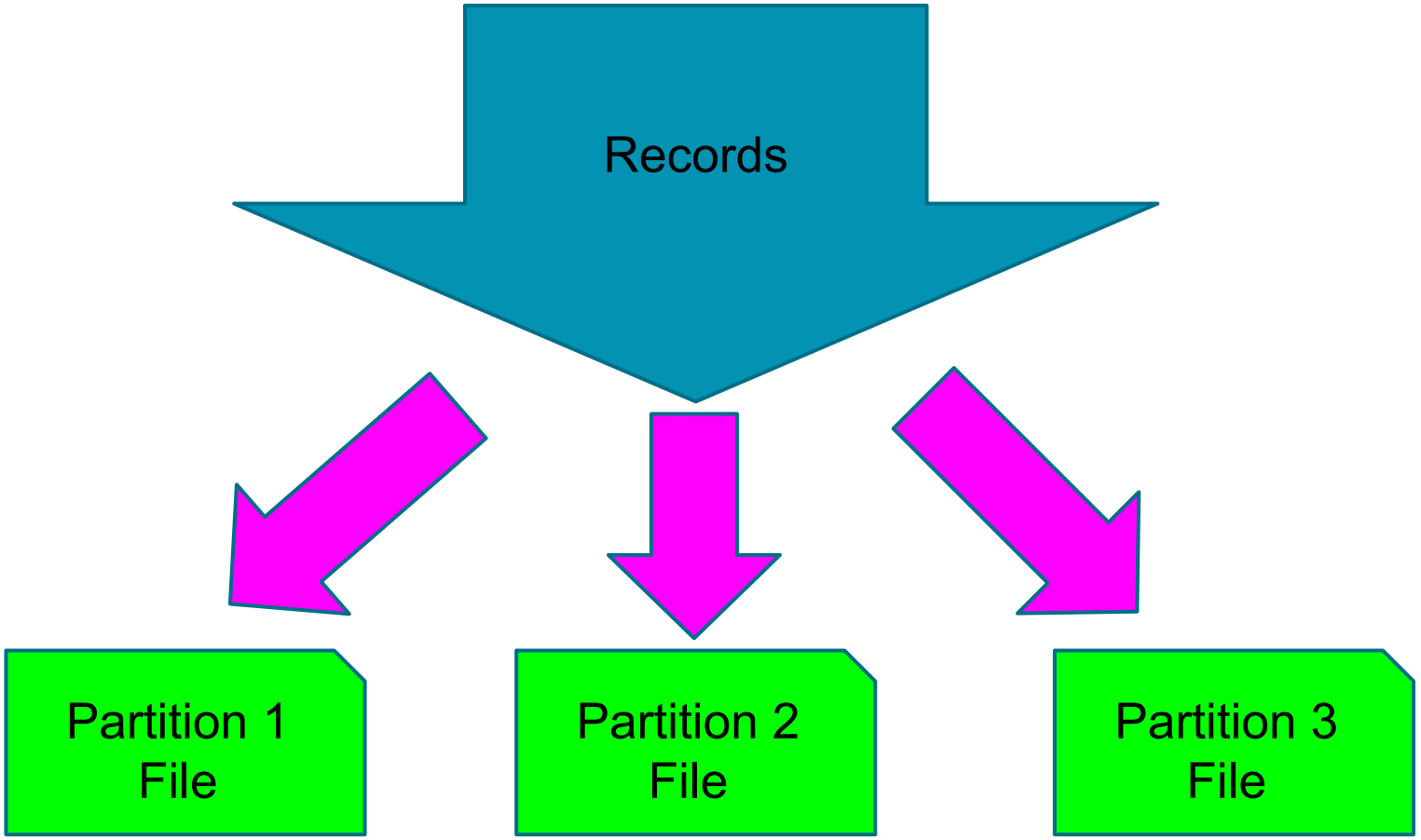
Shuffle: Sending ("map") Side



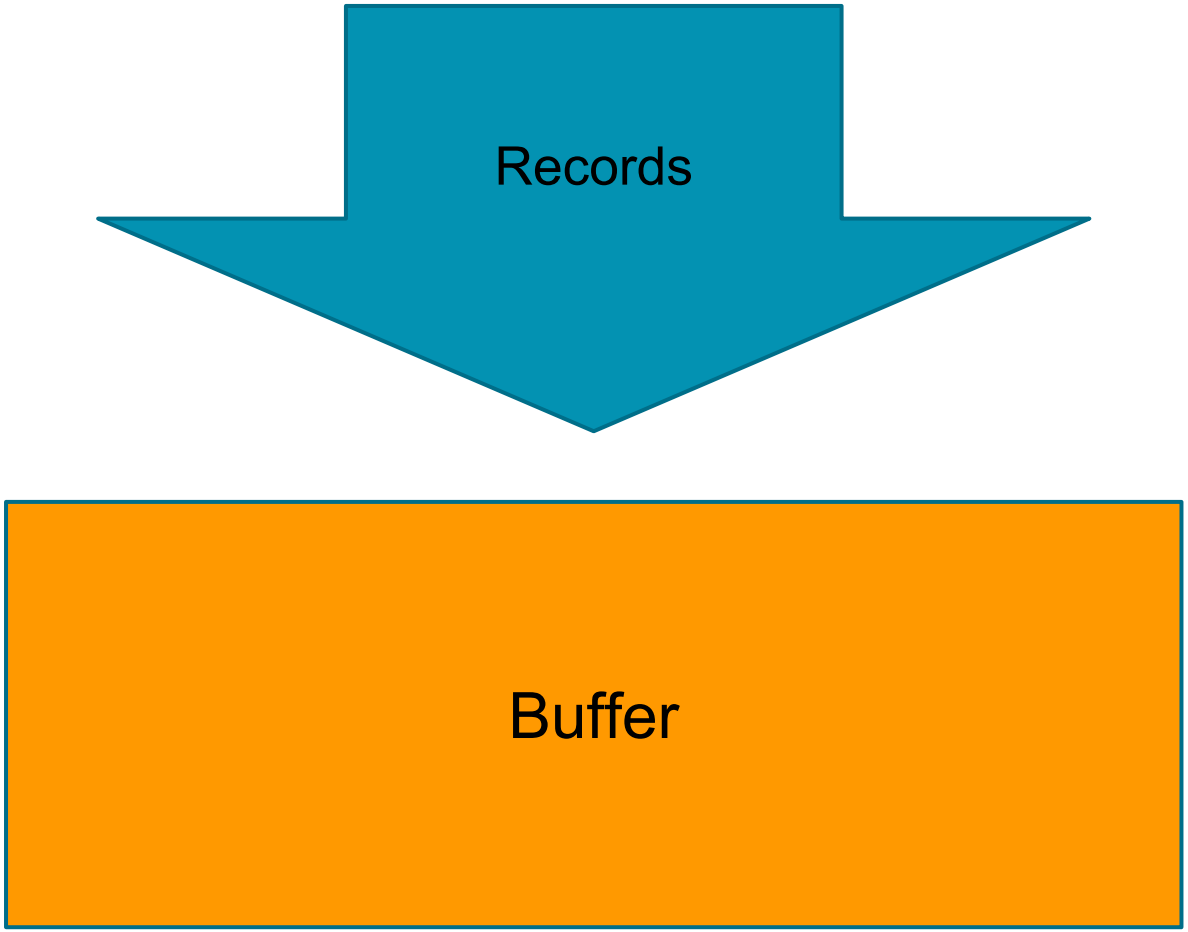
Shuffle: Sending ("map") Side



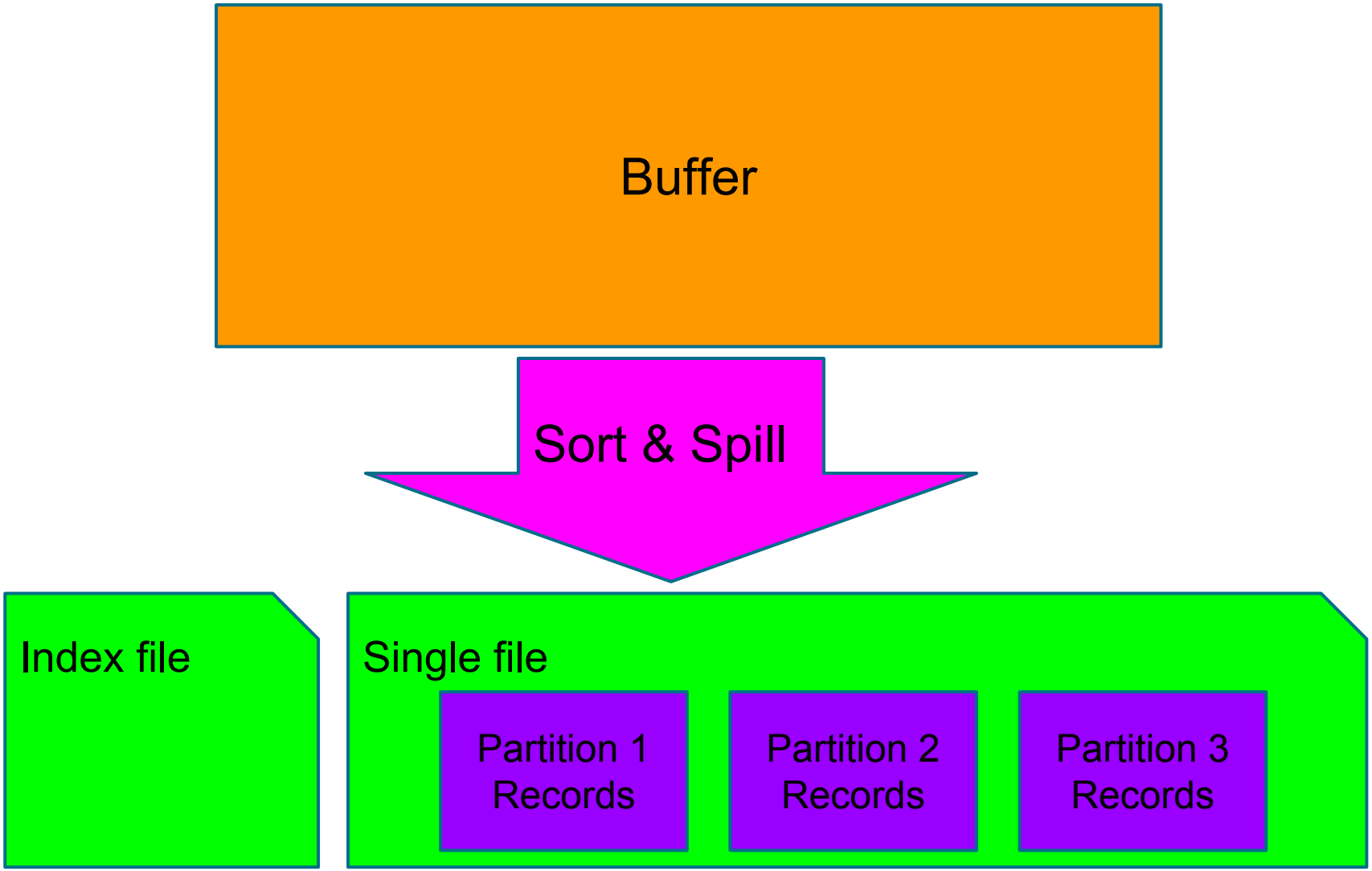
Hash-based shuffle



Sort-based shuffle



Spill


Buffer
Sort & Spill
Index file
Single file
Partition 1 Records
Partition 2 Records
Partition 3 Records

```
conf.set("spark.shuffle.manager",
         SORT)
```

Is Spark bad?

- No
- Distributed systems are complicated

Thanks!