TP2: Apache Spark

Eya Zaoui, Khalil Bouzir, Farouk Labidi – RT5

Repo: https://github.com/eya-cfu/tp-big-data

OBJECTIVES

Get started with Apache Spark, a large-scale parallel data processing engine working on top of Hadoop.

SETUP

We're launching Spark on top of Hadoop YARN, where we have one master node and 2 data nodes and managed by YARN.

```
                                        hadoop-slave2
806dbbb32f70   liliasfaxi/spark-hadoop:hv-2.7.2   "sh -c 'service ssh …"   12 days a
go    Up 12 days              0.0.0.0:8040->8042/tcp, :::8040->8042/tcp

                                        hadoop-slave1
6956190b19b5   liliasfaxi/spark-hadoop:hv-2.7.2   "sh -c 'service ssh …"   12 days a
go    Up 12 days              0.0.0.0:7077->7077/tcp, :::7077->7077/tcp, 0.0.0.0:808
8->8088/tcp, :::8088->8088/tcp, 0.0.0.0:16010->16010/tcp, :::16010->16010/tcp, 0.0.0
.0:50070->50070/tcp, :::50070->50070/tcp   hadoop-master
eyaz@main:~$
```

```
root@hadoop-master:~# jps
352 SecondaryNameNode
509 ResourceManager
1358 Jps
159 NameNode
```

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.2.0
      /_/

Using Scala version 2.11.8 (OpenJDK 64-Bit Server VM, Java 1.8.0_191)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

TESTING THE SPARK SHELL AND SCALA

1)

```
root@hadoop-master:~/file1.count# ls
_SUCCESS  part-00000  part-00001
root@hadoop-master:~/file1.count# cat part-00000
(Hello,2)
(Wordcount!,1)
root@hadoop-master:~/file1.count# cat part-00001
(Spark,1)
(:),1)
(Also,1)
(Hadoop,1)
root@hadoop-master:~/file1.count#
```

2)

```
scala> val docs = sc.textFile("/docs")
docs: org.apache.spark.rdd.RDD[String] = /docs MapPartitionsRDD[1] at textFile at <c
onsole>:24

scala> val lower = docs.map(line => line.toLowerCase)
lower: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at map at <console>:26

scala> val words = lower.flatMap(line => line.split("\\s+"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[3] at flatMap at <console
>:28

scala> val counts = words.map(word => (word,1))
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[4] at map at <con
sole>:30

scala> val freq = counts.reduceByKey(_ + _)
org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: hdfs://ha
doop-master:9000/docs
  at org.apache.hadoop.mapred.FileInputFormat.singleThreadedListStatus(FileInputForm
at.java:287)
  at org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:229)
  at org.apache.hadoop.mapred.FileInputFormat.getSplits(FileInputFormat.java:315)
  at org.apache.spark.rdd.HadoopRDD.getPartitions(HadoopRDD.scala:194)
  at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:252)
  at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:250)
  at scala.Option.getOrElse(Option.scala:121)
```

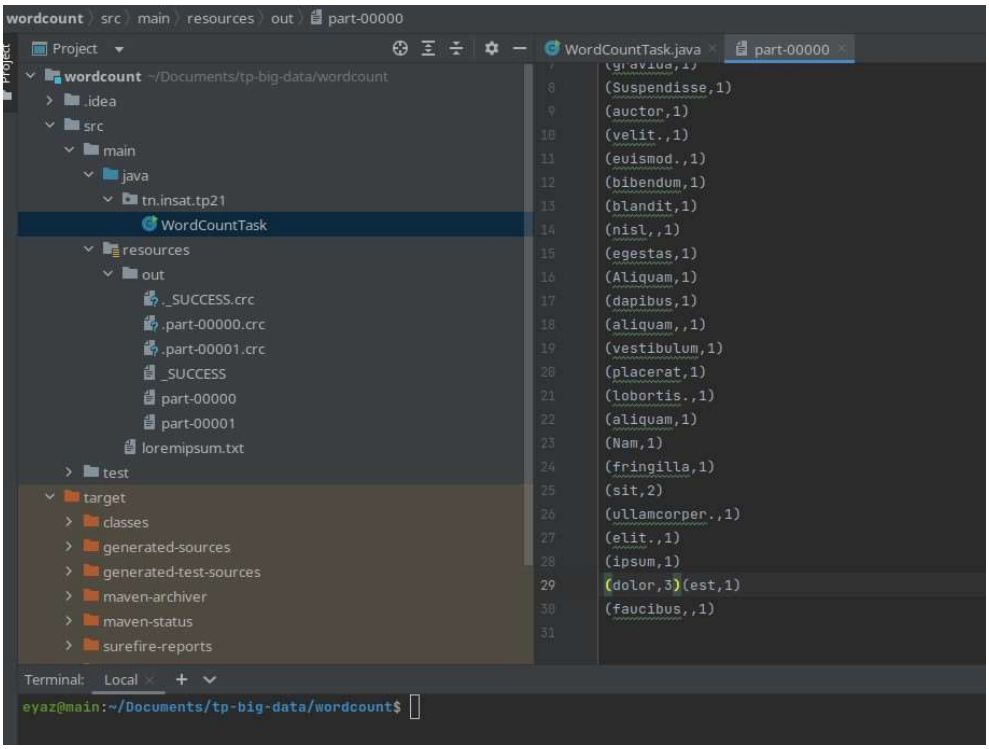
I forgot that my input file is named “doc”, not “docs”. It’s interesting to see the *lazy evaluation* of Spark *transformations* as the error was only raised in the reduce step (doing an *action*).

After fixing the problem, here is the result:

```
scala> val docs = sc.textFile("doc")
docs: org.apache.spark.rdd.RDD[String] = doc MapPartitionsRDD[6] at textFile at <cor
sole>:24

scala>   val lower = docs.map(line => line.toLowerCase)
lower: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[7] at map at <console>:26

scala>   val words = lower.flatMap(line => line.split("\\s+"))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[8] at flatMap at <console
>:28

scala>   val counts = words.map(word => (word,1))
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at map at <cor
sole>:30

scala>   val freq = counts.reduceByKey(_ + _)
freq: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[10] at reduceByKey at <c
onsole>:32

scala> freq.map(_.swap)
res0: org.apache.spark.rdd.RDD[(Int, String)] = MapPartitionsRDD[11] at map at <cons
ole>:35

scala> val top = freq.map(_swap).top(6)
<console>:34: error: not found: value _swap
       val top = freq.map(_swap).top(6)
                          ^

scala> val top = freq.map(_.swap).top(6)
top: Array[(Int, String)] = Array((2,hello), (1,world), (1,new), (1,line), (1,a))

scala>
```

SPARK BATCH IN JAVA

1) Local execution:



2) Executing on the Hadoop cluster:

```
22/12/02 19:07:07 INFO executor.Executor: Finished task 0.0 in stage 1.0 (TID 2). 11
81 bytes result sent to driver
22/12/02 19:07:07 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 1.0 (TID
 3, localhost, executor driver, partition 1, ANY, 4621 bytes)
22/12/02 19:07:07 INFO executor.Executor: Running task 1.0 in stage 1.0 (TID 3)
22/12/02 19:07:07 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID
 2) in 634 ms on localhost (executor driver) (1/2)
22/12/02 19:07:07 INFO storage.ShuffleBlockFetcherIterator: Getting 2 non-empty bloc
ks out of 2 blocks
22/12/02 19:07:07 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches
 in 0 ms
22/12/02 19:07:08 INFO output.FileOutputCommitter: File Output Committer Algorithm v
ersion is 1
22/12/02 19:07:08 INFO output.FileOutputCommitter: Saved output of task 'attempt_202
21202190656_0001_m_000001_3' to hdfs://hadoop-master:9000/user/root/output/_temporar
y/0/task_20221202190656_0001_m_000001
22/12/02 19:07:08 INFO mapred.SparkHadoopMapRedUtil: attempt_20221202190656_0001_m_0
00001_3: Committed
22/12/02 19:07:08 INFO executor.Executor: Finished task 1.0 in stage 1.0 (TID 3). 12
24 bytes result sent to driver
22/12/02 19:07:08 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 1.0 (TID
 3) in 310 ms on localhost (executor driver) (2/2)
22/12/02 19:07:08 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks
 have all completed, from pool
22/12/02 19:07:08 INFO scheduler.DAGScheduler: ResultStage 1 (saveAsTextFile at Word
CountTask.java:33) finished in 0.941 s
22/12/02 19:07:08 INFO scheduler.DAGScheduler: Job 0 finished: saveAsTextFile at Wor
dCountTask.java:33, took 11.276438 s
22/12/02 19:07:08 INFO spark.SparkContext: Invoking stop() from shutdown hook
22/12/02 19:07:08 INFO server.AbstractConnector: Stopped Spark@68596877{HTTP/1.1,[ht
tp/1.1]}{0.0.0.0:4041}
22/12/02 19:07:08 INFO ui.SparkUI: Stopped Spark web UI at http://172.18.0.2:4041
22/12/02 19:07:08 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterE
ndpoint stopped!
22/12/02 19:07:08 INFO memory.MemoryStore: MemoryStore cleared
22/12/02 19:07:08 INFO storage.BlockManager: BlockManager stopped
22/12/02 19:07:08 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
22/12/02 19:07:08 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndp
oint: OutputCommitCoordinator stopped!
22/12/02 19:07:08 INFO spark.SparkContext: Successfully stopped SparkContext
22/12/02 19:07:08 INFO util.ShutdownHookManager: Shutdown hook called
22/12/02 19:07:08 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-7f493
96a-099f-4b40-bfe9-5b95f23715e3
root@hadoop-master:~#
```

```
root@hadoop-master:~# head output/part-00000
(166.92,82)
(116.84,86)
(379.92,78)
(411.79,95)
(487.01,72)
(4.24,75)
(326.52,88)
(384.14,74)
(244.04,85)
(144.96,89)
```

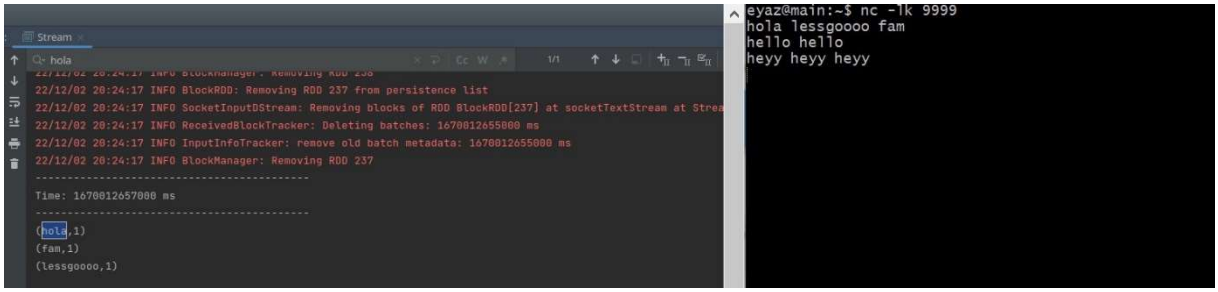
3) Testing on the cluster with YARN:

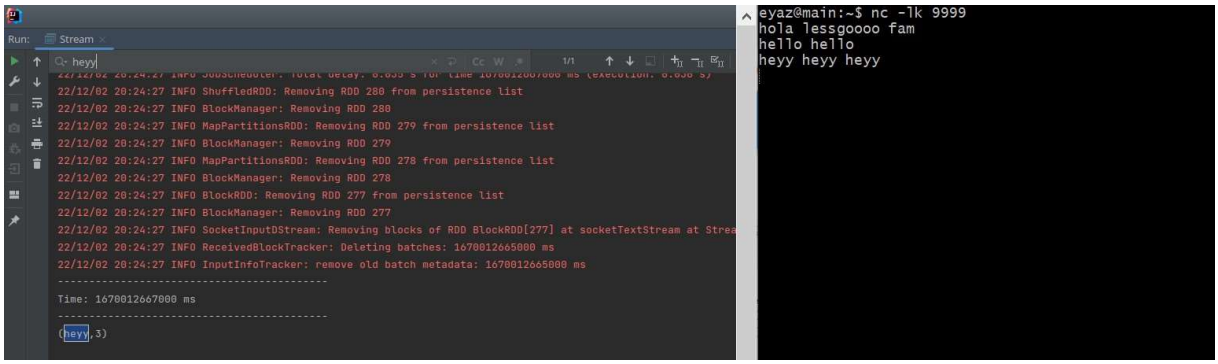
```
22/12/03 12:42:26 INFO client.RMProxy: Connecting to ResourceManager at hadoop-maste
r/172.18.0.2:8032
22/12/03 12:42:26 INFO yarn.Client: Requesting a new application from cluster with 2
 NodeManagers
22/12/03 12:42:26 INFO yarn.Client: Verifying our application has not requested more
 than the maximum memory capability of the cluster (8192 MB per container)
22/12/03 12:42:26 INFO yarn.Client: Will allocate AM container, with 4505 MB memory
including 409 MB overhead
22/12/03 12:42:26 INFO yarn.Client: Setting up container launch context for our AM
22/12/03 12:42:26 INFO yarn.Client: Setting up the launch environment for our AM con
tainer
22/12/03 12:42:26 INFO yarn.Client: Preparing resources for our AM container
22/12/03 12:42:28 WARN yarn.Client: Neither spark.yarn.jars nor spark.yarn.archive i
s set, falling back to uploading libraries under SPARK_HOME.
22/12/03 12:42:30 INFO yarn.Client: Uploading resource file:/tmp/spark-88860f42-9f72
-4756-9e05-009f7a2a4711/__spark_libs__6758761974844561678.zip -> hdfs://hadoop-maste
r:9000/user/root/.sparkStaging/application_1670067556499_0004/__spark_libs__67587619
74844561678.zip
22/12/03 12:42:34 INFO yarn.Client: Uploading resource file:/root/wordcount-1.jar ->
 hdfs://hadoop-master:9000/user/root/.sparkStaging/application_1670067556499_0004/wo
rdcount-1.jar
22/12/03 12:42:34 INFO yarn.Client: Uploading resource file:/tmp/spark-88860f42-9f72
-4756-9e05-009f7a2a4711/__spark_conf__7219029802393812475.zip -> hdfs://hadoop-maste
r:9000/user/root/.sparkStaging/application_1670067556499_0004/__spark_conf__.zip
22/12/03 12:42:34 INFO spark.SecurityManager: Changing view acls to: root
```

```
root@hadoop-master:~# hadoop fs -ls output2/
Found 3 items
-rw-r--r--   2 root supergroup          0 2022-12-03 12:33 output2/_SUCCESS
-rw-r--r--   2 root supergroup     300531 2022-12-03 12:33 output2/part-00000
-rw-r--r--   2 root supergroup     300511 2022-12-03 12:33 output2/part-00001
root@hadoop-master:~# hadoop fs -tail output2/part-00000
.2,87)
(375.38,73)
(291.13,65)
(446.42,77)
(452.49,81)
```

SPARK STREAMING

1) Local test:





2) Testing on the cluster: