

MSDS610 Week 6 Spark Assignment - Nathan Worsham

I installed the latest Spark and chose to download the package that was prebuilt for Hadoop 2.6 and later. Then as usual, unpacked it, and created a sym link.

```
spark-1.6.0-bin-hadoop2.6/examples/src/main/resources/people.txt
spark-1.6.0-bin-hadoop2.6/examples/src/main/resources/full_user.avsc
spark-1.6.0-bin-hadoop2.6/examples/src/main/resources/users.parquet
spark-1.6.0-bin-hadoop2.6/examples/src/main/resources/users.avro
spark-1.6.0-bin-hadoop2.6/examples/src/main/resources/people.json
spark-1.6.0-bin-hadoop2.6/examples/src/main/resources/user.avsc
spark-1.6.0-bin-hadoop2.6/NOTICE
spark-1.6.0-bin-hadoop2.6/RELEASE
[hadoop@week1 ~]$ ls
apache-hive-1.2.1-bin  hadoopdata  hive  ml-100k.zip  pig_1455010061604.log  pigtutorial.tar.gz
derby.log             hbase       metastore_db  pig  pig_test  pig_test  scripts
hadoop               hbase-1.1.2  ml-100k      pig-0.15.0  pigtap    pigtap    shakespeareoutput.txt
[hadoop@week1 ~]$ ln -s /home/hadoop/spark-1.6.0-bin-hadoop2.6 /home/hadoop/spark
[hadoop@week1 ~]$ ls -l
total 382288
drwxr-xr-x.  8 hadoop hadoop    4096 Feb  1 21:53 apache-hive-1.2.1-bin
-rw-rw-rw-.  1 hadoop hadoop   21050 Feb  3 22:31 derby.log
drwxr-xr-x. 10 hadoop hadoop    4096 Feb  9 02:14 hadoop
drwxr-xr-x.  3 hadoop hadoop    17 Jan 12 17:44 hadoopdata
drwxr-xr-x.  8 hadoop hadoop    4096 Jan 25 19:36 hbase
drwxr-xr-x.  8 hadoop hadoop    4096 Jan 25 19:12 hbase-1.1.2
drwxrwxr-x.  1 root  root      34 Feb  1 21:54 hive -> /home/hadoop/apache-hive-1.2.1-bin
drwxr-xr-x.  5 hadoop hadoop    4096 Feb  3 22:31 metastore_db
drwxr-xr-x.  2 hadoop hadoop    4096 Jan 29 13:26 ml-100k
-rw-rw-rw-.  1 hadoop hadoop  492402 Jan 29 13:28 ml-100k.zip
drwxrwxr-x.  1 hadoop hadoop    23 Feb  8 21:48 pig -> /home/hadoop/pig-0.15.0
drwxr-xr-x. 16 hadoop hadoop    4096 Jun  1 2015 pig-0.15.0
-rw-rw-rw-.  1 hadoop hadoop    5230 Feb  9 02:37 pig_1455010061604.log
drwxr-xr-x. 15 hadoop hadoop    4096 Feb  9 07:01 pig_test
drwxr-xr-x.  4 hadoop hadoop    4096 Feb 10 09:56 pigtap
-rw-rw-rw-.  1 hadoop hadoop  10467367 Feb  9 07:16 pigtutorial.tar.gz
drwxrwxr-x.  2 hadoop hadoop    30 Feb  2 21:41 scripts
-rw-rw-rw-.  1 hadoop hadoop   356409 Jan 12 20:04 shakespeareoutput.txt
-rw-rw-rw-.  1 hadoop hadoop  4538523 Aug 16 2011 shakespeare.txt
drwxrwxr-x.  1 hadoop hadoop    30 Feb 15 12:03 spark -> /home/hadoop/spark-1.6.0-bin-hadoop2.6
drwxr-xr-x. 12 hadoop hadoop    4096 Dec 21 19:22 spark-1.6.0-bin-hadoop2.6
-rw-rw-rw-.  1 hadoop hadoop 209160904 Dec 20 16:21 spark-1.6.0-bin-hadoop2.6.tgz
drwxrwxr-x.  4 hadoop hadoop    51 Jan 25 22:38 zookeeper
[hadoop@week1 ~]$
```

As far as configuration went, it seemed just a couple of environmental variables needed to be set. HADOOP_CONF_DIR was already set from last week, so I just set SPARK_DIST_CLASSPATH to \$HADOOP_HOME and SPARK_LOCAL_IP to 10.0.2.15. By the end of the assignment I realized I needed/wanted one last environmental variable because I was once again getting a similar warning to week 1–

Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

in week 1 I ended up setting the JAVA_LIBRARY_PATH, which was still set but not fixing the problem. I ended up setting the following to fix it:

```
export LD_LIBRARY_PATH=$HADOOP_HOME/lib/native:$LD_LIBRARY_PATH
```

I then ran the example Scala command it gave of ./bin/run-example SparkPi 10. Even after scanning the output I wasn't sure what I was supposed to look for or what the script even did. After looking through the actual script, I realized that somewhere in the mass of text that it was supposed to output "Pi is roughly..." and that really is all the example script does is calculate Pi, though I ran the script several times and each time the output is slightly different. It seems that number at the end of the command tells Spark how many processes or threads to split the job into.

```
16/02/15 16:05:51 INFO executor.Executor: Finished task 9.0 in stage 0.0 (TID 9) 1001 bytes result sent to driver
16/02/15 16:05:51 INFO scheduler.TaskSetManager: Finished task 9.0 in stage 0.0 (TID 9) in 50 ms on localhost (10/10)
16/02/15 16:05:51 INFO scheduler.DAGScheduler: ResultStage 0 (reduce at SparkPi.scala:36) finished in 2.152 s
16/02/15 16:05:51 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed. Free pool
16/02/15 16:05:51 INFO scheduler.DAGScheduler: Job 0 finished: reduce at SparkPi.scala:36, took 2.758919 s
Pi is roughly 3.142512
16/02/15 16:05:51 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler(/metrics/json,null)
16/02/15 16:05:51 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler(/stages/stage/kill,null)
16/02/15 16:05:51 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler(/api,null)
16/02/15 16:05:51 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler(/,null)
```

I went ahead and looked through the scala examples directory but wasn't really sure how to use all of these examples. I did go ahead and try running the SparkALS example in the same manner as the SparkPi example. This did seem to work, it outputted an RMSE (root-mean-square-error) value from 5 iterations. Though I'm not sure what the values indicate.

```
[hadoop@week1 spark]$ ./bin/run-example SparkALS 10
WARN: This is a naive implementation of ALS and is given as an example!
Please use the ALS method found in org.apache.spark.mllib.recommendation
for more conventional use.

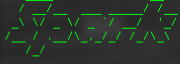
Running with M=10, U=500, F=10, iters=5
16/02/16 07:42:36 INFO spark.SparkContext: Running Spark version 1.6.0
16/02/16 07:42:37 WARN util.NativeCodeLoader: Unable to load native-hadoop
16/02/16 07:42:37 WARN util.Utils: Your hostname, week1 resolves to a lo
16/02/16 07:42:37 WARN util.Utils: Set SPARK_LOCAL_IP if you need to b
16/02/16 07:42:37 INFO spark.SecurityManager: Changing view acls to: had
16/02/16 07:42:37 INFO spark.SecurityManager: Changing modify acls to: h
16/02/16 07:42:37 INFO spark.SecurityManager: SecurityManager: authentic
16/02/16 07:42:38 INFO util.Utils: Successfully started service 'sparkDr
16/02/16 07:42:39 INFO slf4j.Slf4jLogger: Slf4jLogger started
16/02/16 07:42:39 INFO Remoting: Starting remoting
16/02/16 07:42:39 INFO util.Utils: Successfully started service 'sparkDr
16/02/16 07:42:39 INFO Remoting: Remoting started; listening on address
16/02/16 07:42:39 INFO spark.SparkEnv: Registering MapOutputTracker
16/02/16 07:42:39 INFO spark.SparkEnv: Registering BlockManagerMaster

16/02/16 07:42:47 INFO scheduler.DAGScheduler: ResultStage 9 (
16/02/16 07:42:47 INFO scheduler.DAGScheduler: Job 9 finished:
16/02/16 07:42:47 INFO storage.MemoryStore: Block broadcast_22
16/02/16 07:42:47 INFO storage.MemoryStore: Block broadcast_22
16/02/16 07:42:47 INFO storage.BlockManagerInfo: Added broadcast
16/02/16 07:42:47 INFO spark.SparkContext: Created broadcast 22
RMSE = 0.83306617446760545

16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
16/02/16 07:42:47 INFO handler.ContextHandler: stopped o.s.j.s
```

As another test I tried running the Python IDLE in spark, which worked as expected.

```
16/02/15 21:58:34 INFO storage.BlockManagerMaster: Trying to register BlockMa
16/02/15 21:58:34 INFO storage.BlockManagerMasterEndpoint: Registering block
16/02/15 21:58:34 INFO storage.BlockManagerMaster: Registered BlockManager
Welcome to

 version 1.6.0

Using Python version 2.7.5 (default, Nov 28 2015 02:00:19)
SparkContext available as sc, HiveContext available as sqlContext.
(>>> help())

Welcome to Python 2.7! This is the online help utility.

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://docs.python.org/2.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules. To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules",
"keywords", or "topics". Each module also comes with a one-line summary
of what it does; to list the modules whose summaries contain a given word
such as "spam", type "modules spam".

(help> modules

Please wait a moment while I gather a list of all available modules...

BaseHTTPServer      bisect              io                  setuptools
Bastion              bsddb               itertools          sets
CDROM                bz2                 javapackages       sgmlib
CGIHTTPServer        cPickle             json               sha
ConfigParser         cProfile            keyword             shelve
Cookie               cStringIO           lib2to3             shlex
DLFCN                calendar            liblzma             shutils
...

```

Scala Shell

I connected to the Scala shell as directed and it worked as expected. I am not familiar with scala, but looking it up on the internet seems it is a programming language whose name means "scalable language" (scala-lang.org, 2016). Continuing on, it appears that `val` must set a variable or object and I successfully set the `readme` file to the `textFile` object. However when I tried the next line to count the words in the file I received an error about the input path not existing in the HDFS.

```
16/02/15 22:24:46 INFO repl.SparkILoop: Created sql context (with Hive support)...
SQL context available as sqlContext.

scala> val textFile = sc.textFile("README.md")
16/02/15 22:25:10 INFO storage.MemoryStore: Block broadcast_0 stored as values in memory (estimated size 61.8 KB, free 61.8 KB)
16/02/15 22:25:10 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size 19.4 KB, free 81.2 KB)
16/02/15 22:25:10 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on localhost:47557 (size: 19.4 KB, free: 517.4 MB)
16/02/15 22:25:10 INFO spark.SparkContext: Created broadcast 0 from textFile at <console>:27
textFile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:27

scala> textFile.count()
org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: hdfs://localhost:9000/user/hadoop/README.md
at org.apache.hadoop.mapred.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:385)
```

I went ahead and using `copyFromLocal`, copied the file to the HDFS but again received the same error. I came across a stackoverflow.com (2014) thread that had a very similar problem, the solution was to use "file:///home/hadoop/spark/README.md" instead of just "README.md". I went ahead and tried this but the first time I did not put enough forward slashes and received an error message of

Wrong FS: file:///home/hadoop/spark/README.md, expected: file:///

[illegible]

```
scala> textFile(file.first())
16/02/16 01:59:03 INFO aspred.F:
16/02/16 01:59:03 INFO spark.Spi
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO storage.L
16/02/16 01:59:03 INFO storage.L
16/02/16 01:59:03 INFO storage.L
16/02/16 01:59:03 INFO spark.Spi
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO executor
16/02/16 01:59:03 INFO rdd.Hadoop
16/02/16 01:59:03 INFO executor
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO schedule
16/02/16 01:59:03 INFO schedule
res4: String = # Apache Spark
```

```

local= val line=spark => textFile(filter(line => line.contains("Spark")))
linesWithSpark) op:org.apache.spark.rdd.RDD[String] => MapPartitionsRDD[10] at filter at <console>:29

scala> textFile.filter(line => line.contains("Spark")).count()
16/02/16 02:02:25 INFO spark.SparkContext: Starting job: count at <console>:30
16/02/16 02:02:25 INFO scheduler.DAGScheduler: Got job 4 (count at <console>:30) with 1 output parts
16/02/16 02:02:25 INFO scheduler.DAGScheduler: Final stage: ResultStage 4 (count at <console>:30)
16/02/16 02:02:25 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/02/16 02:02:25 INFO scheduler.DAGScheduler: Missing parents: List()
16/02/16 02:02:25 INFO scheduler.DAGScheduler: Submitting ResultStage 4 (MapPartitionsRDD[10] at fil
16/02/16 02:02:25 INFO storage.MemoryStore: Block broadcast_9 stored as values in memory (estimated
16/02/16 02:02:25 INFO storage.MemoryStore: Block broadcast_9_piece0 stored as bytes in memory (esti
16/02/16 02:02:25 INFO storage.BlockManagerInfo: Added broadcast_9_piece0 in memory on localhost:808
16/02/16 02:02:25 INFO spark.SparkContext: Created broadcast 9 from broadcast at DAGScheduler: local
16/02/16 02:02:25 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 4 (MapPar
16/02/16 02:02:25 INFO scheduler.TaskSchedulerImpl: Added task set 4.0 with 1 tasks
16/02/16 02:02:25 INFO scheduler.TaskScheduler: Starting task 0.0 in stage 4.0 (TID 4, localhost, 9
16/02/16 02:02:25 INFO executor.Executor: Running task 0.0 in stage 4.0 (TID 4)
16/02/16 02:02:25 INFO rdd.HadoopRDD: Input split: file:/home/hadoop/spark/README.md#3359
16/02/16 02:02:25 INFO executor.Executor: Finished task 0.0 in stage 4.0 (TID 4). 2082 bytes result
16/02/16 02:02:25 INFO scheduler.TaskScheduler: Finished task 0.0 in stage 4.0 (TID 4) in 37 ms on lo
16/02/16 02:02:25 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all comple
16/02/16 02:02:25 INFO scheduler.TaskScheduler: ResultStage 4 (count at <console>:30) finished in 0.0
16/02/16 02:02:25 INFO scheduler.DAGScheduler: Job 4 finished: count at <console>:30, took 0.851519
res5: Long = 17

```

For the "More on RDD Operations" section, I received the answer of line 14 being the longest, however I found it fairly tough trying to understand the command they gave. I even took a look at the Python version—since I am more familiar with Python—but that seemed even more cryptic. The exercise goes on to say to make the command "easier" to understand it uses the `Math.max()` function but again I'm not familiar with it, so it doesn't clear up much for me but does confirm the same output of 14. That being said, looking at the Python version of the example, it shows it using a custom function instead and the function is much easier to read.

```
scala> textFile.map(line => line.split(" ").size).reduce(a, b) => if (a > b) a else b
16/02/16 02:18:59 INFO spark.SparkContext: Starting job: reduce at <console>:30
16/02/16 02:18:59 INFO scheduler.DAGScheduler: Got job 7 (reduce at <console>:30) with 1 output partitions
16/02/16 02:18:59 INFO scheduler.DAGScheduler: Final stage: ResultStage 7 (reduce at <console>:30)
16/02/16 02:18:59 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/02/16 02:18:59 INFO scheduler.DAGScheduler: Missing parents: List()
16/02/16 02:18:59 INFO scheduler.DAGScheduler: Submitting ResultStage 7 (MapPartitionsRDD[15] at reduce at <console>:30)
16/02/16 02:18:59 INFO storage.MemoryStore: Block broadcast_12_piece0 stored as values in memory (estimated size 2.3 KB, free 1011.4 KB)
16/02/16 02:18:59 INFO storage.MemoryStore: Block broadcast_12_piece0 stored as bytes in memory (estimated size 2.3 KB, free 1011.4 KB)
16/02/16 02:18:59 INFO storage.BlockManagerInfo: Added broadcast_12_piece0 in memory on localhost:60615 (size: 2.3 KB, free: 1011.4 KB)
16/02/16 02:18:59 INFO spark.SparkContext: Created broadcast 12 from broadcast at DAGScheduler.scala:1086
16/02/16 02:18:59 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 7 (MapPartitionsRDD[15] at reduce at <console>:30)
16/02/16 02:18:59 INFO scheduler.TaskSchedulerImpl: Adding task set 7.0 with 1 tasks
16/02/16 02:18:59 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 7.0 (TID 7)
16/02/16 02:18:59 INFO rdd.HadoopRDD: Input split: file:/home/hadoop/spark/README.ad:0+3359
16/02/16 02:18:59 INFO executor.Executor: Running task 0.0 in stage 7.0 (TID 7)
16/02/16 02:18:59 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 7.0 (TID 7)
16/02/16 02:18:59 INFO scheduler.DAGScheduler: ResultStage 7 (reduce at <console>:30) finished in 0.201 s
16/02/16 02:18:59 INFO scheduler.DAGScheduler: Job 7 finished: reduce at <console>:30, took 0.517923 s
res1: Int = 14

scala> import java.lang.Math
import java.lang.Math
scala> textFile.map(line => line.split(" ").size).reduce(a, b) => Math.max(a, b)
16/02/16 02:21:01 INFO spark.SparkContext: Starting job: reduce at <console>:31
16/02/16 02:21:01 INFO scheduler.DAGScheduler: Got job 8 (reduce at <console>:31) with 1 output partitions
16/02/16 02:21:01 INFO scheduler.DAGScheduler: Final stage: ResultStage 8 (reduce at <console>:31)
16/02/16 02:21:01 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/02/16 02:21:01 INFO scheduler.DAGScheduler: Missing parents: List()
16/02/16 02:21:01 INFO scheduler.DAGScheduler: Submitting ResultStage 8 (MapPartitionsRDD[16] at reduce at <console>:31)
16/02/16 02:21:01 INFO storage.MemoryStore: Block broadcast_13_piece0 stored as values in memory (estimated size 2.3 KB, free 1011.4 KB)
16/02/16 02:21:01 INFO storage.MemoryStore: Block broadcast_13_piece0 stored as bytes in memory (estimated size 2.3 KB, free 1011.4 KB)
16/02/16 02:21:01 INFO storage.BlockManagerInfo: Added broadcast_13_piece0 in memory on localhost:60615 (size: 2.3 KB, free: 1011.4 KB)
16/02/16 02:21:01 INFO spark.SparkContext: Created broadcast 13 from broadcast at DAGScheduler.scala:1086
16/02/16 02:21:01 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 8 (MapPartitionsRDD[16] at reduce at <console>:31)
16/02/16 02:21:01 INFO scheduler.TaskSchedulerImpl: Adding task set 8.0 with 1 tasks
16/02/16 02:21:01 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 8.0 (TID 8)
16/02/16 02:21:01 INFO executor.Executor: Running task 0.0 in stage 8.0 (TID 8)
16/02/16 02:21:01 INFO rdd.HadoopRDD: Input split: file:/home/hadoop/spark/README.ad:0+3359
16/02/16 02:21:01 INFO executor.Executor: Finished task 0.0 in stage 8.0 (TID 8)
16/02/16 02:21:01 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 8.0 (TID 8)
16/02/16 02:21:01 INFO scheduler.DAGScheduler: ResultStage 8 (reduce at <console>:31) finished in 0.196 s
16/02/16 02:21:01 INFO scheduler.DAGScheduler: Job 8 finished: reduce at <console>:31, took 0.517923 s
res1: Int = 14
```

Finally I did the per-word counts as directed. I found it strange that the output was cut off.

```
scala> val wordCounts = textFile.flatMap(line => line.split(" ").map(word => (word, 1))).reduceByKey((a, b) => a + b)
wordCounts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[16] at reduceByKey at <console>:30

scala> wordCounts.collect()
16/02/16 02:39:36 INFO spark.SparkContext: Starting job: collect at <console>:33
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Registering RDD 15 (map at <console>:30)
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Got job 9 (collect at <console>:33) with 1 output partitions
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Final stage: ResultStage 10 (collect at <console>:33)
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 9)
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 9)
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 9 (MapPartitionsRDD[15] at map at <console>:30)
16/02/16 02:39:36 INFO storage.MemoryStore: Block broadcast_14_piece0 stored as values in memory (estimated size 4.1 KB, free 1011.4 KB)
16/02/16 02:39:36 INFO storage.MemoryStore: Block broadcast_14_piece0 stored as bytes in memory (estimated size 2.3 KB, free 1011.4 KB)
16/02/16 02:39:36 INFO storage.BlockManagerInfo: Added broadcast_14_piece0 in memory on localhost:60615 (size: 2.3 KB, free: 1011.4 KB)
16/02/16 02:39:36 INFO spark.SparkContext: Created broadcast 14 from broadcast at DAGScheduler.scala:1086
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ShuffleMapStage 9 (MapPartitionsRDD[15] at map at <console>:30)
16/02/16 02:39:36 INFO scheduler.TaskSchedulerImpl: Adding task set 9.0 with 1 tasks
16/02/16 02:39:36 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 9.0 (TID 9), localhost, partition 0, PROCESS_LOCAL, 0x0
16/02/16 02:39:36 INFO executor.Executor: Running task 0.0 in stage 9.0 (TID 9)
16/02/16 02:39:36 INFO rdd.HadoopRDD: Input split: file:/home/hadoop/spark/README.ad:0+3359
16/02/16 02:39:36 INFO executor.Executor: Finished task 0.0 in stage 9.0 (TID 9). 2253 bytes result sent to driver
16/02/16 02:39:36 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 9.0 (TID 9) in 277 ms on localhost (1/1)
16/02/16 02:39:36 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed, from pool
16/02/16 02:39:36 INFO scheduler.DAGScheduler: ShuffleMapStage 9 (map at <console>:30) finished in 0.201 s
16/02/16 02:39:36 INFO scheduler.DAGScheduler: looking for newly runnable stages
16/02/16 02:39:36 INFO scheduler.DAGScheduler: running: Set()
16/02/16 02:39:36 INFO scheduler.DAGScheduler: waiting: Set(ResultStage 10)
16/02/16 02:39:36 INFO scheduler.DAGScheduler: failed: Set()
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Submitting ResultStage 10 (ShuffledRDD[16] at reduceByKey at <console>:30)
16/02/16 02:39:36 INFO storage.MemoryStore: Block broadcast_15 stored as values in memory (estimated size 2.6 KB, free 1016.3 KB)
16/02/16 02:39:36 INFO storage.MemoryStore: Block broadcast_15_piece0 stored as bytes in memory (estimated size 1509.0 KB, free 1016.3 KB)
16/02/16 02:39:36 INFO storage.BlockManagerInfo: Added broadcast_15_piece0 in memory on localhost:60615 (size: 1509.0 KB, free 1016.3 KB)
16/02/16 02:39:36 INFO spark.SparkContext: Created broadcast 15 from broadcast at DAGScheduler.scala:1086
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 10 (ShuffledRDD[16] at reduceByKey at <console>:30)
16/02/16 02:39:36 INFO scheduler.TaskSchedulerImpl: Adding task set 10.0 with 1 tasks
16/02/16 02:39:36 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 10.0 (TID 10), localhost, partition 0, NODE_LOCAL, 0x0
16/02/16 02:39:36 INFO executor.Executor: Running task 0.0 in stage 10.0 (TID 10)
16/02/16 02:39:36 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/02/16 02:39:36 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 4 ms
16/02/16 02:39:36 INFO executor.Executor: Finished task 0.0 in stage 10.0 (TID 10). 7240 bytes result sent to driver
16/02/16 02:39:36 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 10.0 (TID 10) in 195 ms on localhost (1/1)
16/02/16 02:39:36 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 10.0, whose tasks have all completed, from pool
16/02/16 02:39:36 INFO scheduler.DAGScheduler: ResultStage 10 (collect at <console>:33) finished in 0.196 s
16/02/16 02:39:36 INFO scheduler.DAGScheduler: Job 9 finished: collect at <console>:33, took 0.517923 s
res13: Array[(String, Int)] = Array((package,1), (For,2), (Programs,1), (processing,1), (Because,1), (The,1), (cluster,1), (rough,1), (several,1), (This,2), (graph,1), (Have,2), (storage,1), (Specific,1), (To,2), (page,1)(http://spark.apache.org/docs/0.9.0/), (engine,1), (version,1), (file,1), (documentation,1), (processing,1), (the,21), (are,1), (systems,1), (params,1), (not,1), (4), (build,3), (when,1), (be,2), (Tests,1), (Apache,1), (/bin/run-example,2), (programs,1), (including,3), (Spark,1), (B,1), (programming,1), (...))
```

I found I could list all of the output using `println(wordCountsResults.deep.mkString("\n"))`.

```
scala> println(wordCountsResults.deep.mkString("\n"))
(package,1)
(For,2)
(Programs,1)
(processing,,1)
(Because,1)
(The,1)
(cluster,,1)
(its,1)
(run,1)
(APIs,1)
(have,1)
(Try,1)
(computation,1)
(through,1)
(several,1)
(This,2)
(graph,1)
(Hive,2)
(storage,1)
(["Specifying,1)
(To,2)
(page)(http://spark.apache.org/documentation.html),1)
(Once,1)
("yarn",1)
(prefer,1)
(SparkPi,2)
(engine,1)
(version,1)
(file,1)
(documentation,,1)
(processing,,1)
(the,21)
(are,1)
(systems,,1)
(params,1)
(not,1)
(different,1)
(refer,2)
(Interactive,2)
(R,,1)
(given,,1)
(If,4)
(build,3)
(when,1)
```

References

scala-lang.org, 2016. Retrieved from <http://www.scala-lang.org/what-is-scala.html>

stackoverflow.com, 2014. Retrieved from <http://stackoverflow.com/questions/27299923/how-to-load-local-file-in-sc-textfile-instead-of-hdfs>

stackoverflow.com, 2010. Retrieved from <http://stackoverflow.com/questions/3328085/scala-printing-arrays>

spark.apache.org, n.d. Retrieved from <http://spark.apache.org/docs/latest>