Big Data Application Homework #3 - Summer 2017

4. Use the REPL to explore Spark RDDs. (Use the Answer Sheet document to record the answers to the blue prompts below.)

a. In a terminal window, start the Scala Spark Shell: $ spark-shell

b. Spark creates a SparkContext object for you called sc. Make sure the object exists: scala> sc

c. Using command completion, you can see all the available SparkContext methods: type: sc.[TAB]

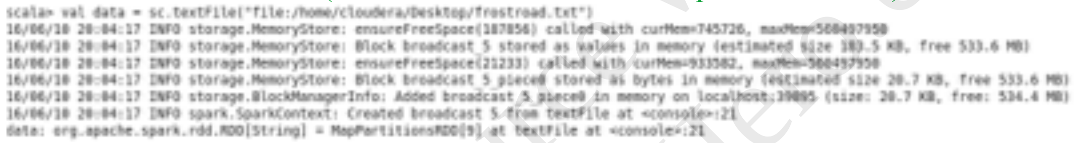
scala schema

d. Copy the input file, frostroad.txt, into the local filesystem of your VM (not in HDFS).

e. Define an RDD to be created from frostroad.txt that exists in your local file system (not HDFS). (Reference the file as: “file:**/path/to/your/file/**frostroad.txt")

1) Provide the command you used to create your RDD.

```
val data = sc.textFile("file:/home/cloudera/Desktop/frostroad.txt")
```

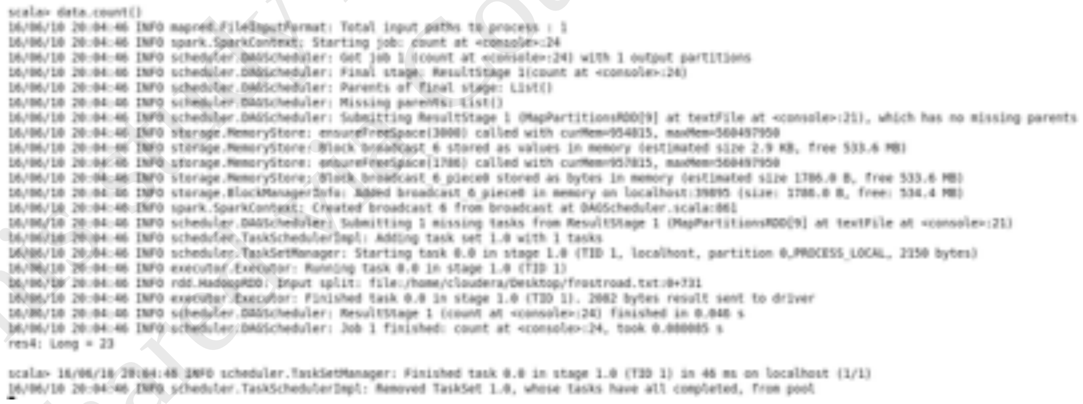


```
scala> val data = sc.textFile("file:/home/cloudera/Desktop/frostroad.txt")
16/06/10 20:04:17 INFO storage.MemoryStore: ensureFreeSpace(187856) called with curMem=745726, maxMem=560497950
16/06/10 20:04:17 INFO storage.MemoryStore: Block broadcast_5 stored as values in memory (estimated size 183.5 KB, free 533.6 MB)
16/06/10 20:04:17 INFO storage.MemoryStore: ensureFreeSpace(21233) called with curMem=933582, maxMem=560497950
16/06/10 20:04:17 INFO storage.MemoryStore: Block broadcast_5_piece0 stored as bytes in memory (estimated size 20.7 KB, free 533.6 MB)
16/06/10 20:04:17 INFO storage.BlockManagerInfo: Added broadcast_5_piece0 in memory on localhost:39895 (size: 20.7 KB, free: 534.4 MB)
16/06/10 20:04:17 INFO spark.SparkContext: Created broadcast 5 from textFile at <console>:21
data: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[9] at textFile at <console>:21
```

f. Once the above command is issued, remember that Spark has not yet read the file. It will not do so until you perform an action on the RDD. Count the number of lines in the dataset.

2) Provide the command you used to count the elements (lines) in your RDD.

```
data.count()
```



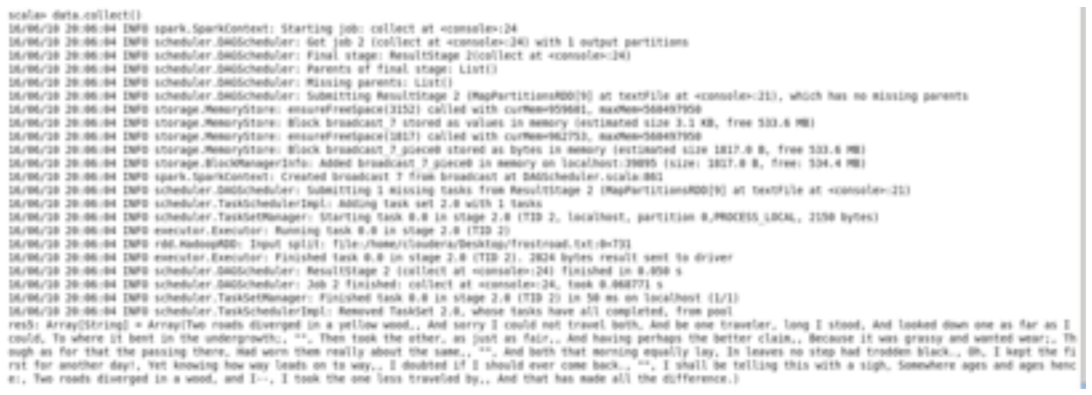
```
scala> data.count()
16/06/10 20:04:46 INFO mapred.FileInputFormat: Total input paths to process : 1
16/06/10 20:04:46 INFO spark.SparkContext: Starting job: count at <console>:24
16/06/10 20:04:46 INFO scheduler.DAGScheduler: Got job 1 (count at <console>:24) with 1 output partitions
16/06/10 20:04:46 INFO scheduler.DAGScheduler: Final stage: ResultStage 1(count at <console>:24)
16/06/10 20:04:46 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/06/10 20:04:46 INFO scheduler.DAGScheduler: Missing parents: List()
16/06/10 20:04:46 INFO scheduler.DAGScheduler: Submitting ResultStage 1 (MapPartitionsRDD[9] at textFile at <console>:21), which has no missing parents
16/06/10 20:04:46 INFO storage.MemoryStore: ensureFreeSpace(3000) called with curMem=954815, maxMem=560497950
16/06/10 20:04:46 INFO storage.MemoryStore: Block broadcast_6 stored as values in memory (estimated size 2.9 KB, free 533.6 MB)
16/06/10 20:04:46 INFO storage.MemoryStore: ensureFreeSpace(1786) called with curMem=957815, maxMem=560497950
16/06/10 20:04:46 INFO storage.MemoryStore: Block broadcast_6_piece0 stored as bytes in memory (estimated size 1786.0 B, free 533.6 MB)
16/06/10 20:04:46 INFO storage.BlockManagerInfo: Added broadcast_6_piece0 in memory on localhost:39895 (size: 1786.0 B, free: 534.4 MB)
16/06/10 20:04:46 INFO spark.SparkContext: Created broadcast 6 from broadcast at DAGScheduler.scala:861
16/06/10 20:04:46 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (MapPartitionsRDD[9] at textFile at <console>:21)
16/06/10 20:04:46 INFO scheduler.TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
16/06/10 20:04:46 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 1.0 (TID 1, localhost, partition 0,PROCESS_LOCAL, 2150 bytes)
16/06/10 20:04:46 INFO executor.Executor: Running task 0.0 in stage 1.0 (TID 1)
16/06/10 20:04:46 INFO rdd.HadoopRDD: Input split: file:/home/cloudera/Desktop/frostroad.txt:0+731
16/06/10 20:04:46 INFO executor.Executor: Finished task 0.0 in stage 1.0 (TID 1). 2082 bytes result sent to driver
16/06/10 20:04:46 INFO scheduler.DAGScheduler: ResultStage 1 (count at <console>:24) finished in 0.048 s
16/06/10 20:04:46 INFO scheduler.DAGScheduler: Job 1 finished: count at <console>:24, took 0.080085 s
res4: Long = 23

scala> 16/06/10 20:04:46 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 46 ms on localhost (1/1)
16/06/10 20:04:46 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
```

3) Provide the number of elements.

23

g. Use collect to display the data in the RDD. This is convenient for very small RDDs like this one, but be careful using collect for more typical large datasets.

4) Provide the collect command you used.



```
scala> data.collect()
16/06/10 20:06:04 INFO spark.SparkContext: Starting job: collect at <console>:24
16/06/10 20:06:04 INFO scheduler.DAGScheduler: Got job 2 (collect at <console>:24) with 1 output partitions
16/06/10 20:06:04 INFO scheduler.DAGScheduler: Final stage: ResultStage 2(collect at <console>:24)
16/06/10 20:06:04 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/06/10 20:06:04 INFO scheduler.DAGScheduler: Missing parents: List()
16/06/10 20:06:04 INFO scheduler.DAGScheduler: Submitting ResultStage 2 (MapPartitionsRDD[9] at textFile at <console>:21), which has no missing parents
16/06/10 20:06:04 INFO storage.MemoryStore: ensureFreeSpace(3152) called with curMem=959681, maxMem=560497950
16/06/10 20:06:04 INFO storage.MemoryStore: Block broadcast_7 stored as values in memory (estimated size 3.1 KB, free 533.6 MB)
16/06/10 20:06:04 INFO storage.MemoryStore: ensureFreeSpace(1817) called with curMem=962753, maxMem=560497950
16/06/10 20:06:04 INFO storage.MemoryStore: Block broadcast_7_piece0 stored as bytes in memory (estimated size 1817.0 B, free 533.6 MB)
16/06/10 20:06:04 INFO storage.BlockManagerInfo: Added broadcast_7_piece0 in memory on localhost:39895 (size: 1817.0 B, free: 534.4 MB)
16/06/10 20:06:04 INFO spark.SparkContext: Created broadcast 7 from broadcast at DAGScheduler.scala:861
16/06/10 20:06:04 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 2 (MapPartitionsRDD[9] at textFile at <console>:21)
16/06/10 20:06:04 INFO scheduler.TaskSchedulerImpl: Adding task set 2.0 with 1 tasks
16/06/10 20:06:04 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 2.0 (TID 2, localhost, partition 0,PROCESS_LOCAL, 2150 bytes)
16/06/10 20:06:04 INFO executor.Executor: Running task 0.0 in stage 2.0 (TID 2)
16/06/10 20:06:04 INFO rdd.HadoopRDD: Input split: file:/home/cloudera/Desktop/frostroad.txt:0+731
16/06/10 20:06:04 INFO executor.Executor: Finished task 0.0 in stage 2.0 (TID 2). 2824 bytes result sent to driver
16/06/10 20:06:04 INFO scheduler.DAGScheduler: ResultStage 2 (collect at <console>:24) finished in 0.050 s
16/06/10 20:06:04 INFO scheduler.DAGScheduler: Job 2 finished: collect at <console>:24, took 0.068771 s
16/06/10 20:06:04 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2.0 (TID 2) in 50 ms on localhost (1/1)
16/06/10 20:06:04 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
res5: Array[String] = Array(Two roads diverged in a yellow wood,, And sorry I could not travel both, And be one traveler, long I stood, And looked down one as far as I
could, To where it bent in the undergrowth;, "", Then took the other, as just as fair,, And having perhaps the better claim,, Because it was grassy and wanted wear;, Th
ough as for that the passing there, Had worn them really about the same,, "", And both that morning equally lay, In leaves no step had trodden black., Oh, I kept the fi
rst for another day!, Yet knowing how way leads on to way,, I doubted if I should ever come back., "", I shall be telling this with a sigh, Somewhere ages and ages henc
e:, Two roads diverged in a wood, and I--, I took the one less traveled by,, And that has made all the difference.)
```

data.collect()

h. Using command completion, view the available transformations and actions you can perform on an RDD. Type: mydata.[TAB]

5. Continued: Transform a small dataset using RDDs. (Use the Answer Sheet document to record the answers to the blue prompts below.)

a. Copy the weblog file, 2014-03-15.log, into the VM. Create a directory in HDFS called loudacre/weblog and put the file into the weblog directory.

5) Provide the command you used to create the HDFS directory.

```
hdfs dfs -mkdir loudacre
hdfs dfs -mkdir loudacre/weblog
```

6) Provide the command you used to put the file into HDFS.

```
hdfs dfs -put Desktop/2014-03-15.log loudacre/weblog/
```

b. View the HDFS version of the file.

7) Provide the command you used to view the file.

The format of the file is:

IP Address: ***116.180.70.237***

-

User ID : ***128*** *[15/Sep/2013:23:59:53 +0100]* *Request:* ***"GET /KBDOC-00031.html HTTP/1.0"***

...

```
hdfs dfs -cat loudacre/weblog/2014-03-15.log
```

c. Store the full file path to a variable named logfile, then process the lines in logfile as follows:

Provide the commands you used for all of the following steps:

8) Initialize **`logfile`**.

```
val logfile = "loudacre/weblog/2014-03-15.log"
```

9) Create an RDD from the file.

```
val data  = sc.textFile(logfile)
```

10) View the first 10 lines of the data.

```
data.take(10).foreach(println)
```

11) Create an RDD containing only lines that are requests for **`jpg`** files.

```
val OnlyJpgs = data.filter(_.contains("jpg"))
```

12) View the first 10 lines of the data.

```
OnlyJpgs.take(10).foreach(println)
```

13) Chain the previous commands into a single command that counts the number of JPG requests.

```
val NumJpgRequests = sc.textFile(logfile).filter(_.contains("jpg")).count()
```

14) Create an RDD using the **`map`** function to return the length of each line of the log file.

```
val lengths = data.map(line => line.length())
```

15) Create an RDD using the **`map`** and **`split`** functions to map an array of words for each line.

```
val words = data.map(line => line.split(" "))
```

16) Create an RDD containing only the IP addresses from each line.

```
val ips = data.map(line => line.split(" ")(0))
```

17) Use **`foreach(println)`** to output IP addresses.

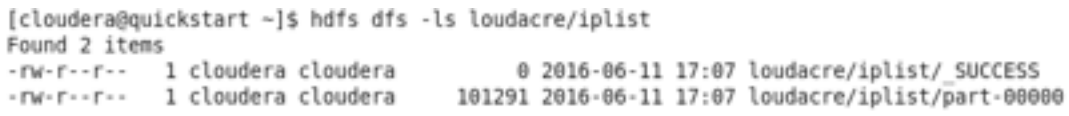
```
ips.foreach(println)
```

18) Save the list of IP addresses to an HDFS directory named **`loudacre/iplist`** using **`saveAsTextFile`**.

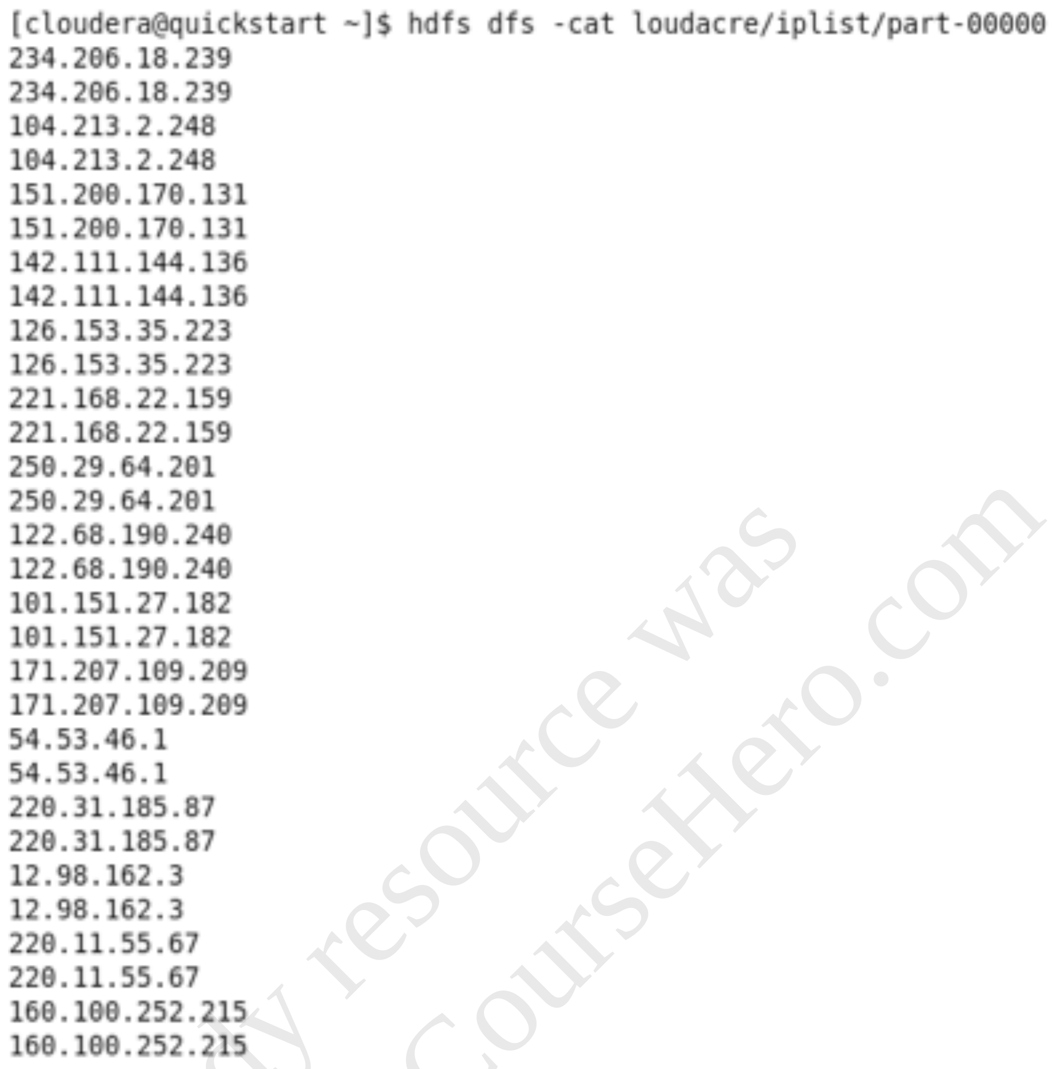
```
ips.saveAsTextFile("loudacre/iplist")
```

d. Store the full file path to a variable named logfile, then process the lines in logfile as follows:

19) Provide a screenshot of the contents of the **`loudacre/iplist`** folder.



```
[cloudera@quickstart ~]$ hdfs dfs -ls loudacre/iplist
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2016-06-11 17:07 loudacre/iplist/_SUCCESS
-rw-r--r--   1 cloudera cloudera     101291 2016-06-11 17:07 loudacre/iplist/part-00000
```



```
[cloudera@quickstart ~]$ hdfs dfs -cat loudacre/iplist/part-00000
234.206.18.239
234.206.18.239
104.213.2.248
104.213.2.248
151.200.170.131
151.200.170.131
142.111.144.136
142.111.144.136
126.153.35.223
126.153.35.223
221.168.22.159
221.168.22.159
250.29.64.201
250.29.64.201
122.68.190.240
122.68.190.240
101.151.27.182
101.151.27.182
171.207.109.209
171.207.109.209
54.53.46.1
54.53.46.1
220.31.185.87
220.31.185.87
12.98.162.3
12.98.162.3
220.11.55.67
220.11.55.67
160.100.252.215
160.100.252.215
```

6. Transform a large dataset using RDDs. (Use the Answer Sheet document to record the answers to the blue prompts below.)

e. Copy the weblogs.zip file to the VM, unzip it, and store it to the loudacre directory.

Provide the commands you used for all of the following steps:

20) Initialize `logfile`.

```
var logfile="loudacre/weblogs/FlumeData.*"
```

21) Create an RDD from the file.

```
var logs = sc.textFile(logfile)
```

22) View the first 10 lines of the data.

```
logs.take(10).foreach(println)
```

23) Create an RDD containing only lines that are requests for `jpg` files.

```
var jpglogs = logs.filter(line => line.contains(".jpg"))
```

24) View the first 10 lines of the data.

```
jpglogs.take(10).foreach(println)
```

25) Chain the previous commands into a single command that counts the number of JPG requests.

```
sc.textFile(logfile).filter(line => line.contains(".jpg")).count()
```

26) Create an RDD using the **`map`** function to return the length of each line of the log file.

```
val lengths = logs.map(line => line.length())
```

27) Create an RDD using the **`map`** and **`split`** functions to map an array of words for each line.

```
val words = logs.map(line => line.split(" "))
```

28) Create an RDD containing only the IP addresses from each line.

```
val ips = logs.map(line => line.split(" ")(0))
```

29) Use **`foreach(println)`** to output IP addresses.

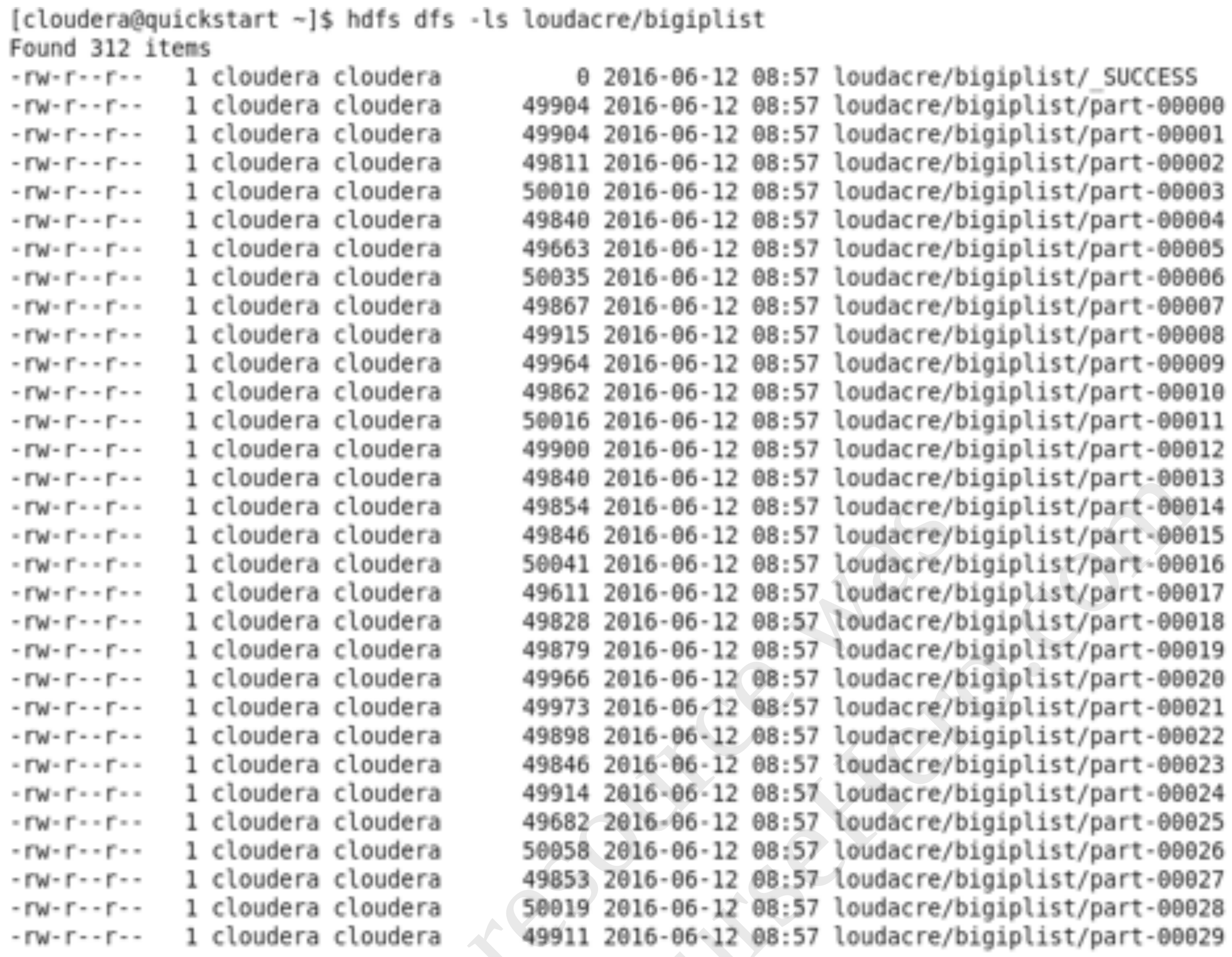
```
ips.foreach(println)
```

30) Save the list of IP addresses to a file in an HDFS directory named **`loudacre/bigiplist`** - use **`saveAsTextFile`**.

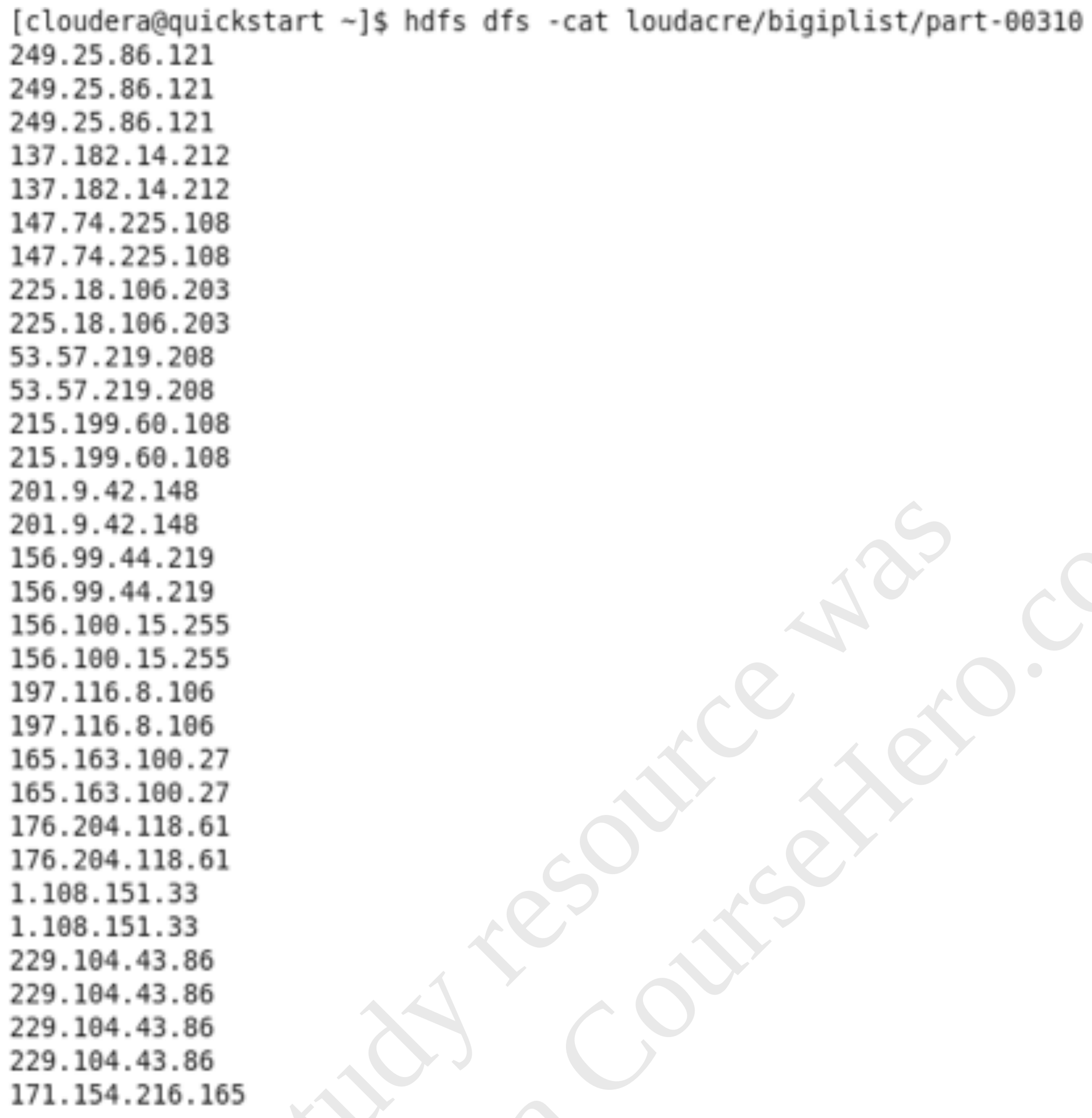
```
ips.saveAsTextFile("loudacre/bigiplist")
```

31) Provide a screenshot of the contents of the **`loudacre/bigiplist`** folder.

__Note__: You may see multiple files, including several part-xxxxx files, which are the files containing the output data. "Part" (partition) files are numbered because there may be results from multiple tasks running in the cluster (the tasks are part of your Spark job). Review the contents of one of the files to confirm that they were created correctly.



```
[cloudera@quickstart ~]$ hdfs dfs -ls loudacre/bigiplist
Found 312 items
-rw-r--r--   1 cloudera cloudera          0 2016-06-12 08:57 loudacre/bigiplist/_SUCCESS
-rw-r--r--   1 cloudera cloudera      49904 2016-06-12 08:57 loudacre/bigiplist/part-00000
-rw-r--r--   1 cloudera cloudera      49904 2016-06-12 08:57 loudacre/bigiplist/part-00001
-rw-r--r--   1 cloudera cloudera      49811 2016-06-12 08:57 loudacre/bigiplist/part-00002
-rw-r--r--   1 cloudera cloudera      50010 2016-06-12 08:57 loudacre/bigiplist/part-00003
-rw-r--r--   1 cloudera cloudera      49840 2016-06-12 08:57 loudacre/bigiplist/part-00004
-rw-r--r--   1 cloudera cloudera      49663 2016-06-12 08:57 loudacre/bigiplist/part-00005
-rw-r--r--   1 cloudera cloudera      50035 2016-06-12 08:57 loudacre/bigiplist/part-00006
-rw-r--r--   1 cloudera cloudera      49867 2016-06-12 08:57 loudacre/bigiplist/part-00007
-rw-r--r--   1 cloudera cloudera      49915 2016-06-12 08:57 loudacre/bigiplist/part-00008
-rw-r--r--   1 cloudera cloudera      49964 2016-06-12 08:57 loudacre/bigiplist/part-00009
-rw-r--r--   1 cloudera cloudera      49862 2016-06-12 08:57 loudacre/bigiplist/part-00010
-rw-r--r--   1 cloudera cloudera      50016 2016-06-12 08:57 loudacre/bigiplist/part-00011
-rw-r--r--   1 cloudera cloudera      49900 2016-06-12 08:57 loudacre/bigiplist/part-00012
-rw-r--r--   1 cloudera cloudera      49840 2016-06-12 08:57 loudacre/bigiplist/part-00013
-rw-r--r--   1 cloudera cloudera      49854 2016-06-12 08:57 loudacre/bigiplist/part-00014
-rw-r--r--   1 cloudera cloudera      49846 2016-06-12 08:57 loudacre/bigiplist/part-00015
-rw-r--r--   1 cloudera cloudera      50041 2016-06-12 08:57 loudacre/bigiplist/part-00016
-rw-r--r--   1 cloudera cloudera      49611 2016-06-12 08:57 loudacre/bigiplist/part-00017
-rw-r--r--   1 cloudera cloudera      49828 2016-06-12 08:57 loudacre/bigiplist/part-00018
-rw-r--r--   1 cloudera cloudera      49879 2016-06-12 08:57 loudacre/bigiplist/part-00019
-rw-r--r--   1 cloudera cloudera      49966 2016-06-12 08:57 loudacre/bigiplist/part-00020
-rw-r--r--   1 cloudera cloudera      49973 2016-06-12 08:57 loudacre/bigiplist/part-00021
-rw-r--r--   1 cloudera cloudera      49898 2016-06-12 08:57 loudacre/bigiplist/part-00022
-rw-r--r--   1 cloudera cloudera      49846 2016-06-12 08:57 loudacre/bigiplist/part-00023
-rw-r--r--   1 cloudera cloudera      49914 2016-06-12 08:57 loudacre/bigiplist/part-00024
-rw-r--r--   1 cloudera cloudera      49682 2016-06-12 08:57 loudacre/bigiplist/part-00025
-rw-r--r--   1 cloudera cloudera      50058 2016-06-12 08:57 loudacre/bigiplist/part-00026
-rw-r--r--   1 cloudera cloudera      49853 2016-06-12 08:57 loudacre/bigiplist/part-00027
-rw-r--r--   1 cloudera cloudera      50019 2016-06-12 08:57 loudacre/bigiplist/part-00028
-rw-r--r--   1 cloudera cloudera      49911 2016-06-12 08:57 loudacre/bigiplist/part-00029
```



```
[cloudera@quickstart ~]$ hdfs dfs -cat loudacre/bigiplist/part-00310
249.25.86.121
249.25.86.121
249.25.86.121
137.182.14.212
137.182.14.212
147.74.225.108
147.74.225.108
225.18.106.203
225.18.106.203
53.57.219.208
53.57.219.208
215.199.60.108
215.199.60.108
201.9.42.148
201.9.42.148
156.99.44.219
156.99.44.219
156.100.15.255
156.100.15.255
197.116.8.106
197.116.8.106
165.163.100.27
165.163.100.27
176.204.118.61
176.204.118.61
1.108.151.33
1.108.151.33
229.104.43.86
229.104.43.86
229.104.43.86
229.104.43.86
171.154.216.165
```