**Big Data Application Development - Summer 2017**

**Homework 3, Part 2 Answer Sheet**

4. Use the REPL to explore Spark RDDs.

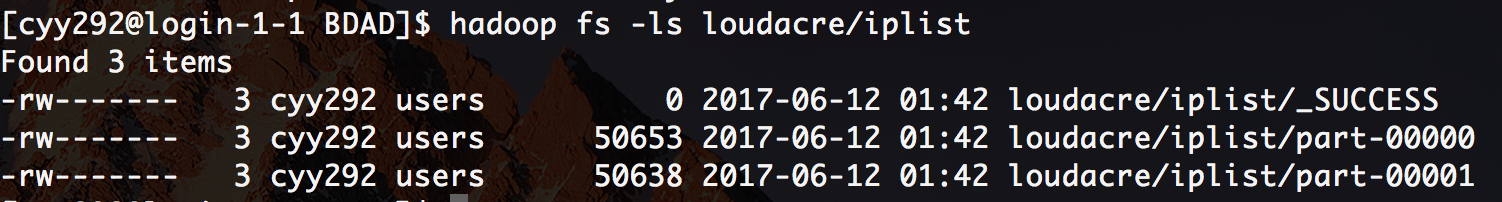
|  |  |
| --- | --- |
| 1) Provide the command you used to create your RDD. | val mydata = sc.textFile("File:///home/cyy292/BDAD/frostroad.txt") |
| 2) Provide the command you used to count the elements (lines) in your RDD. | mydata.count |
| 3) Provide the number of elements. | res1: Long = 23 |
| 4) Provide the collect command you used. | mydata.collect |
| 5) Provide the command you used to create the HDFS directory. | hadoop fs -mkdir loudacre  hadoop fs -mkdir loudacre/weblog |
| 6) Provide the command you used to put the file into HDFS. | hadoop fs -put 2014-03-15.log loudacre/weblog |
| 7) Provide the command you used to view the file. | hadoop fs -cat loudacre/weblog/2014-03-15.log |

5. Transform a small dataset using RDDs.

|  |  |
| --- | --- |
| 8) Initialize **logfile**. | val logfile = "/user/cyy292/loudacre/weblog/2014-03-15.log" |
| 9) Create an RDD from the file. | val logRDD = sc.textFile(logfile) |
| 10) View the first 10 lines of the data. | logRDD.take(10).foreach(println) |
| 11) Create an RDD containing only lines that are requests for **jpg** files. | val logRDD\_jpg = logRDD.filter(\_.contains("jpg")) |
| 12) View the first 10 lines of the data. | logRDD\_jpg.take(10).foreach(println) |
| 13) Chain the previous commands into a single command that counts the number of JPG requests. | scala> logRDD.filter(\_.contains("jpg")).count  res24: Long = 423 |
| 14) Create an RDD using the **map** function to return the length of each line of the log file. | val line\_len = logRDD.map(\_.length) |
| 15) Create an RDD using the **map** and **split** functions to map an array of words for each line. | val line\_split = logRDD.map(\_.split(' ')) |
| 16) Create an RDD containing only the IP addresses from each line. | val line\_IP = line\_split.map(\_(0)) |
| 17) Use **foreach(println)** to output IP addresses. | line\_IP.foreach(println) |
| 18) Save the list of IP addresses to an HDFS directory named **loudacre/iplist** using **saveAsTextFile**. | line\_IP.saveAsTextFile("loudacre/iplist") |

5. Transform a small dataset using RDDs. (continued)

19) Provide a screenshot of the contents of the **loudacre/iplist** folder. (Paste it below.)



6. Transform a large dataset using RDDs.

|  |  |
| --- | --- |
| 20) Initialize **logfile**. | val logfile = "loudacre/weblogs/FlumeData.1424275921" |
| 21) Create an RDD from the file. | var logsRDD: org.apache.spark.rdd.RDD[String] = sc.emptyRDD  scala> for (a <- 226 to 536)       | logsRDD = logsRDD.union(sc.textFile(logfile + a.toString)) |
| 22) View the first 10 lines of the data. | logsRDD.take(10).foreach(println) |
| 23) Create an RDD containing only lines that are requests for **jpg** files. | val logsRDD\_jpg = logsRDD.filter(\_.contains("jpg")) |
| 24) View the first 10 lines of the data. | logsRDD\_jpg.take(10).foreach(println) |
| 25) Chain the previous commands into a single command that counts the number of JPG requests. | scala> logsRDD.filter(\_.contains("jpg")).count  res53: Long = 64978 |
| 26) Create an RDD using the **map** function to return the length of each line of the log file | val logsRDD\_length = logsRDD.map(\_.length) |
| 27) Create an RDD using the **map** and **split** functions to map an array of words for each line. | val logsRDD\_split = logsRDD.map(\_.split(' ')) |
| 28) Create an RDD containing only the IP addresses from each line. | val logsRDD\_IP = logsRDD\_split.map(\_(0)) |
| 29) Use **foreach(println)** to output IP addresses. | logsRDD\_IP.foreach(println) |
| 30) Save the list of IP addresses to a file in an HDFS directory named **loudacre/bigiplist** - use **saveAsTextFile**. | logsRDD\_IP.saveAsTextFile("loudacre/bigiplist") |

6. Transform a large dataset using RDDs. (continued)

31) Provide a screenshot of the contents of the **loudacre/bigiplist** folder. (Paste it below.)

