**Big Data Application Development - Spring 2018**

**Homework 3, Part 2 Answer Sheet**

4. Use the REPL to explore Spark RDDs.

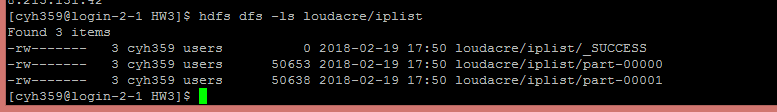
|  |  |
| --- | --- |
| 1) Provide the command you used to create your RDD. | val mydata = sc.textFile("File:///home/cyh359/BDAD/HW3/frostroad.txt") |
| 2) Provide the command you used to count the elements (lines) in your RDD. | mydata.count |
| 3) Provide the number of elements. | res0: Long = 23 |
| 4) Provide the collect command you used. | mydata.collect |
| 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 2014-03-15.log loudacre/weblog |
| 7) Provide the command you used to view the file. | hdfs dfs -cat loudacre/weblog/2014-03-15.log |

5. Transform a small dataset using RDDs.

|  |  |
| --- | --- |
| 8) Initialize **logfile**. | val logfile = "/user/cyh359/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 log\_jpg = logRDD.filter(\_.contains("jpg")) |
| 12) View the first 10 lines of the data. | log\_jpg.take(10).foreach(println) |
| 13) Chain the previous commands into a single command that counts the number of JPG requests. | logRDD.filter(\_.contains("jpg")).count  res3: Long = 423 |
| 14) Create an RDD using the **map** function to return the length of each line of the log file. | val lineLength = logRDD.map(line => line.length()) |
| 15) Create an RDD using the **map** and **split** functions to map an array of words for each line. | val eachLine = logRDD.map(line => line.split(" ")) |
| 16) Create an RDD containing only the IP addresses from each line. | val ipAddress = logRDD.map(line => line.split(" ")(0)) |
| 17) Use **foreach(println)** to output IP addresses. | ipAddress.foreach(println) |
| 18) Save the list of IP addresses to an HDFS directory named **loudacre/iplist** using **saveAsTextFile**. | ipAddress.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**. | var logfile="loudacre/weblogs/FlumeData.\*" |
| 21) Create an RDD from the file. | var logBatch = sc.textFile(logfile) |
| 22) View the first 10 lines of the data. | logBatch.take(10).foreach(println) |
| 23) Create an RDD containing only lines that are requests for **jpg** files. | var onlyJPG = logBatch.filter(line => line.contains(".jpg")) |
| 24) View the first 10 lines of the data. | onlyJPG.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()  res3: Long = 64978 |
| 26) Create an RDD using the **map** function to return the length of each line of the log file | val line\_length = logBatch.map(line => line.length()) |
| 27) Create an RDD using the **map** and **split** functions to map an array of words for each line. | val eachLine = logBatch.map(line => line.split(" ")) |
| 28) Create an RDD containing only the IP addresses from each line. | val ipAddress = logBatch.map(line => line.split(" ")(0)) |
| 29) Use **foreach(println)** to output IP addresses. | ipAddress.foreach(println) |
| 30) Save the list of IP addresses to a file in an HDFS directory named **loudacre/bigiplist** - use **saveAsTextFile**. | ipAddress.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.)

