Vijay's Assignment – Spark Streaming2

There are two parts this case study

- First Part - You have to create a Spark Application which streams data from a file on local directory on your machine and does the word count on the fly. The word should be done by the spark application in such a way that as soon as you drop the file in your local directory, your spark application should immediately do the word count for you.

What did I do?

1) Created a folder C:\Users\VIJAYLAKSHMANAN\spark\input
2) Spark streaming code given below monitors this folder for any "new" files copied to it.
3) Once a file is received, it runs a flatmap to remove all the hierarchical lines and then map to suffix 1 to each word. Using reduceByKey function each word (key) occurance is counted
4) The file uploaded has data separed by ","

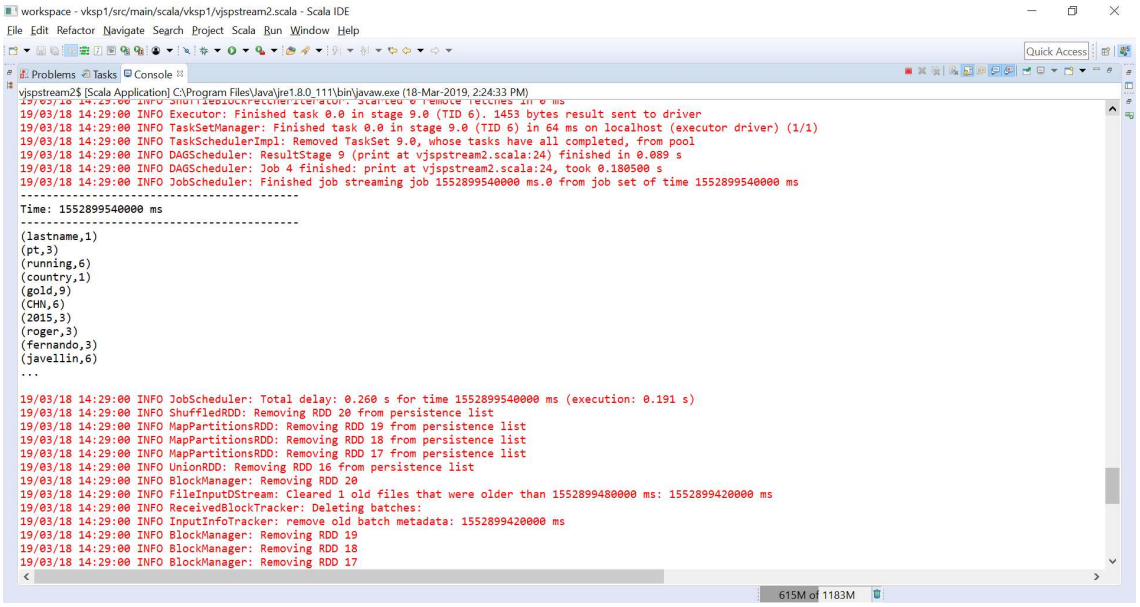
sports1.txt

5)

Code:

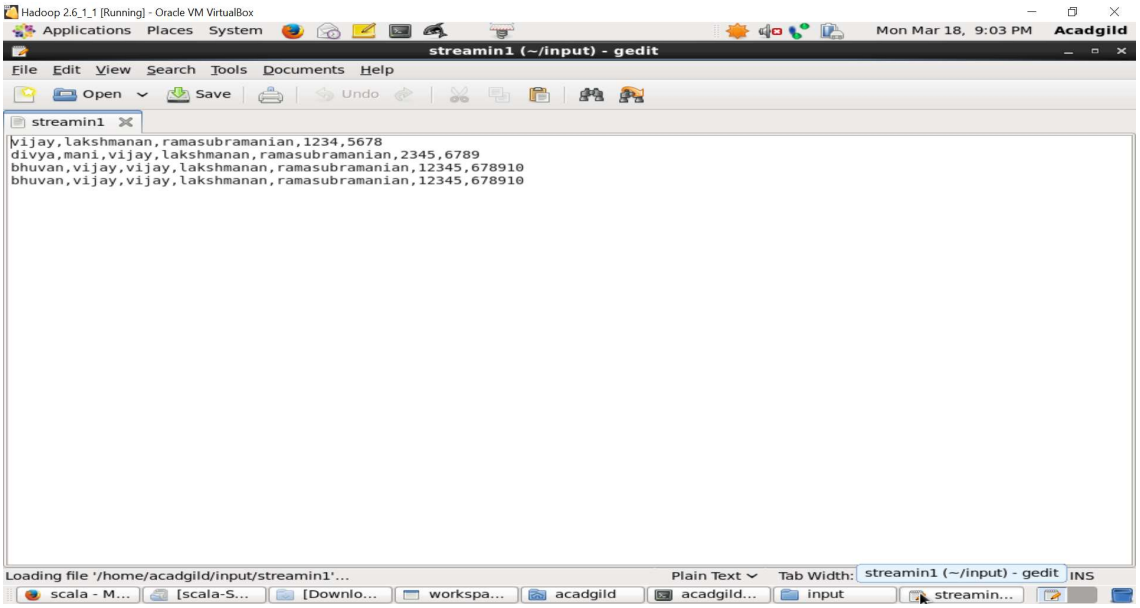
```
package vksp1
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.rdd.RDD.rddToPairRDDFunctions
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext

object vjspstream2 {
  def main(args: Array[String]) = {
    val conf = new SparkConf()
      .setAppName("vjspstream2")
      .setMaster("local")

    val ssc= new StreamingContext(conf, Seconds(60))

    val lines1 = ssc.textFileStream("C:\\Users\\VIJAYLAKSHMANAN\\spark\\input")

    val words = lines1.flatMap(line=>line.split(","))

    val pairs = words.map(x=>(x,1))

    val wcnt = pairs.reduceByKey(_+_)

    wcnt.print()

    ssc.start()
    ssc.awaitTermination()

   }
}
```

Output



▪ Second Part - In this part, you will have to create a Spark Applicationwhich should do the following

Text File used:



1. Pick up a file from the local directory and do the word count

Code:

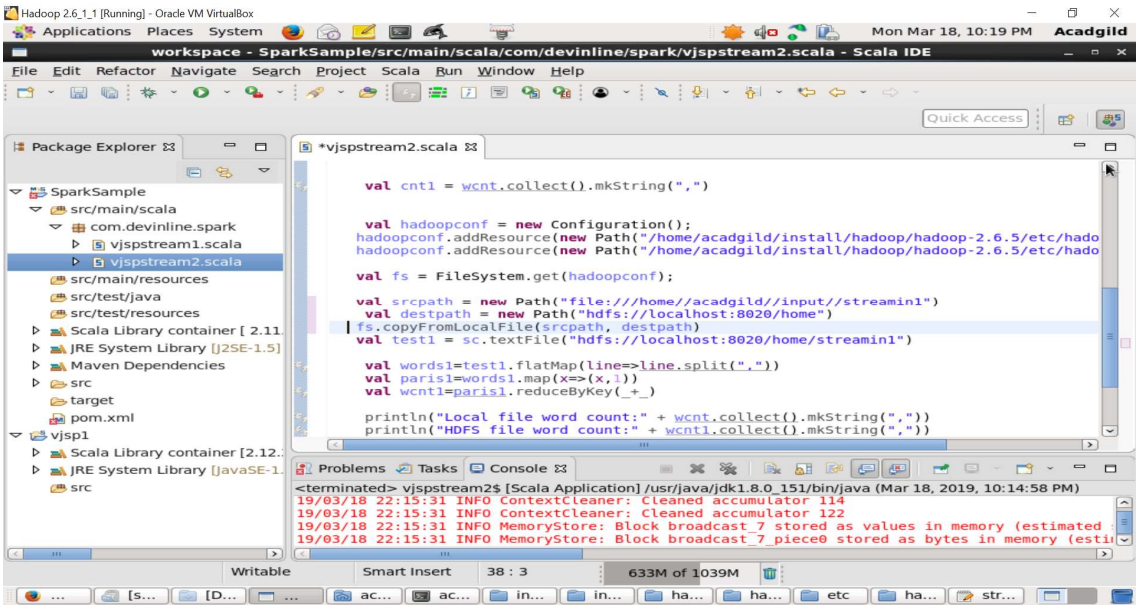
vjspstream2.scala

```
package com.devinline.spark
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext

import org.apache.hadoop.conf.Configuration
import org.apache.hadoop.fs.FileSystem
import org.apache.hadoop.fs.Path

import org.apache.commons.io.IOUtils


object vjspstream2 {
  def main(args: Array[String]){
    val conf = new SparkConf().setAppName("vjspstream2").setMaster("local")
    val sc = new SparkContext(conf)

    val test = sc.textFile("file:///home//acadgild//input//streamin1")

    val words=test.flatMap(line=>line.split(","))
    val paris=words.map(x=>(x,1))
    val wcnt=paris.reduceByKey(_+_)

    println(wcnt.collect().mkString(","))
```

Output:

```
19/03/18 20:57:44 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 18 ms
19/03/18 20:57:44 INFO Executor: Finished task 0.0 in stage 1.0 (TID 1). 1580 bytes result sent to driver
19/03/18 20:57:44 INFO TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 330 ms on localhost (executor drive
19/03/18 20:57:44 INFO DAGScheduler: ResultStage 1 (collect at vjspstream2.scala:25) finished in 0.451 s
19/03/18 20:57:44 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
19/03/18 20:57:44 INFO DAGScheduler: Job 0 finished: collect at vjspstream2.scala:25, took 3.194586 s
(678910,2),(divya,1),(ramasubramanian,4),(12345,2),(lakshmanan,4),(mani,1),(6789,1),(,1),(2345,1),(1234,1),(5678,1),
```

2. Then in the same Spark Application, write the code to put the same file on HDFS.



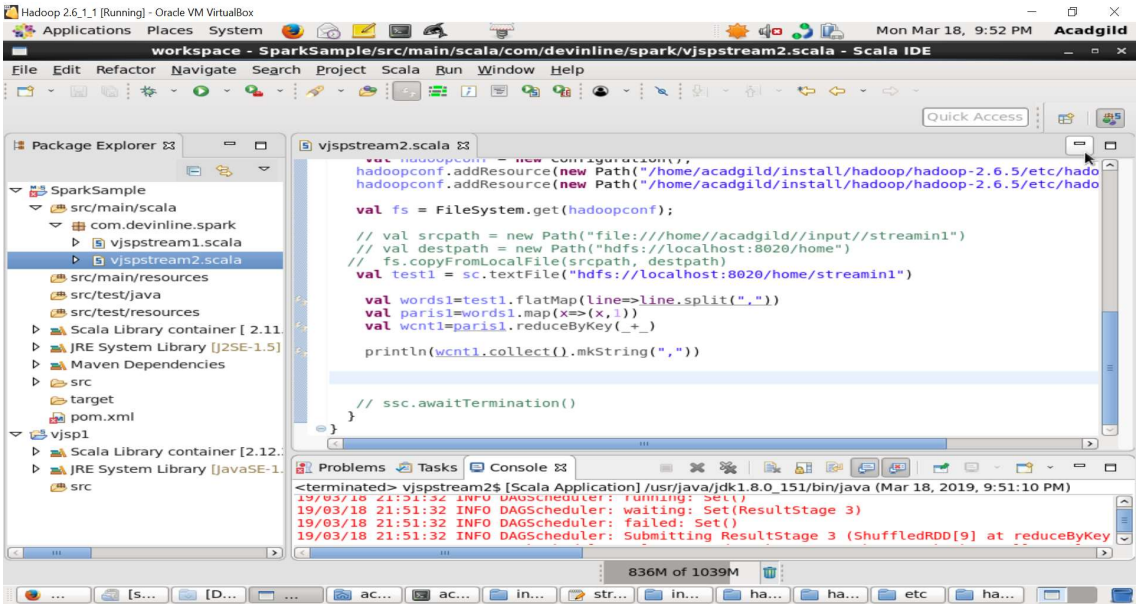
Output

```
[acadgild@localhost ~]$ hadoop fs -ls /home/
19/03/18 21:46:10 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 2 items
drwxr-xr-x   - acadgild supergroup          0 2019-01-14 19:08 /home/acadgild
-rw-r--r--   1 acadgild supergroup        216 2019-03-18 21:45 /home/streamin1
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -ls /home/streamin1
19/03/18 21:46:35 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
-rw-r--r--   1 acadgild supergroup        216 2019-03-18 21:45 /home/streamin1
[acadgild@localhost ~]$ hadoop fs -ls /home/streamin1/
19/03/18 21:46:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
-rw-r--r--   1 acadgild supergroup        216 2019-03-18 21:45 /home/streamin1
[acadgild@localhost ~]$ hadoop fs -ls /home/streamin1
19/03/18 21:46:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
-rw-r--r--   1 acadgild supergroup        216 2019-03-18 21:45 /home/streamin1
[acadgild@localhost ~]$ hadoop fs -cat /home/streamin1
19/03/18 21:47:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
vijay,lakshmanan,ramasubramanian,1234,5678
divya,mani,vijay,lakshmanan,ramasubramanian,2345,6789
bhuvan,vijay,vijay,lakshmanan,ramasubramanian,12345,678910
bhuvan,vijay,vijay,lakshmanan,ramasubramanian,12345,678910
```

3. Then in same Spark Application, do the word count of the file copied on HDFS in step 2

Code:

Copy to HDFS code has been commented out as the file has been copied already as shown above

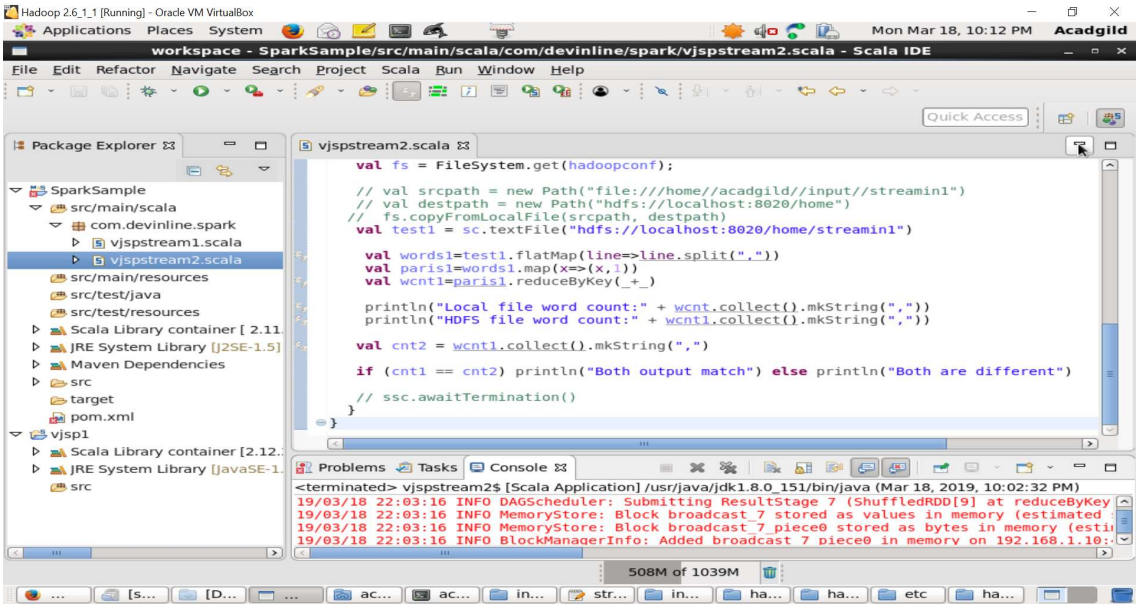


Output:

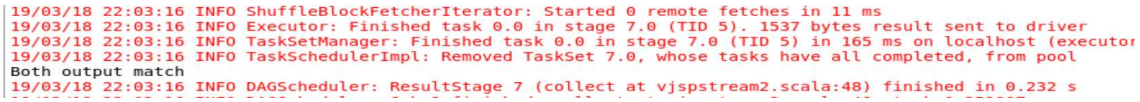
```
19/03/18 21:51:33 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 3) in 149 ms on localhost (executor driver
19/03/18 21:51:33 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
19/03/18 21:51:33 INFO DAGScheduler: ResultStage 3 (collect at vjspstream2.scala:42) finished in 0.195 s
19/03/18 21:51:33 INFO DAGScheduler: Job 1 finished: collect at vjspstream2.scala:42, took 1.041608 s
(678910,2),(divya,1),(ramasubramanian,4),(12345,2),(lakshmanan,4),(mani,1),(6789,1),(,1),(2345,1),(1234,1),(5678,1),(
19/03/18 21:51:33 INFO SparkContext: Invoking stop() from shutdown hook
```

4. Lastly, compare the word count of step 1 and 2. Both should match, other throw an error

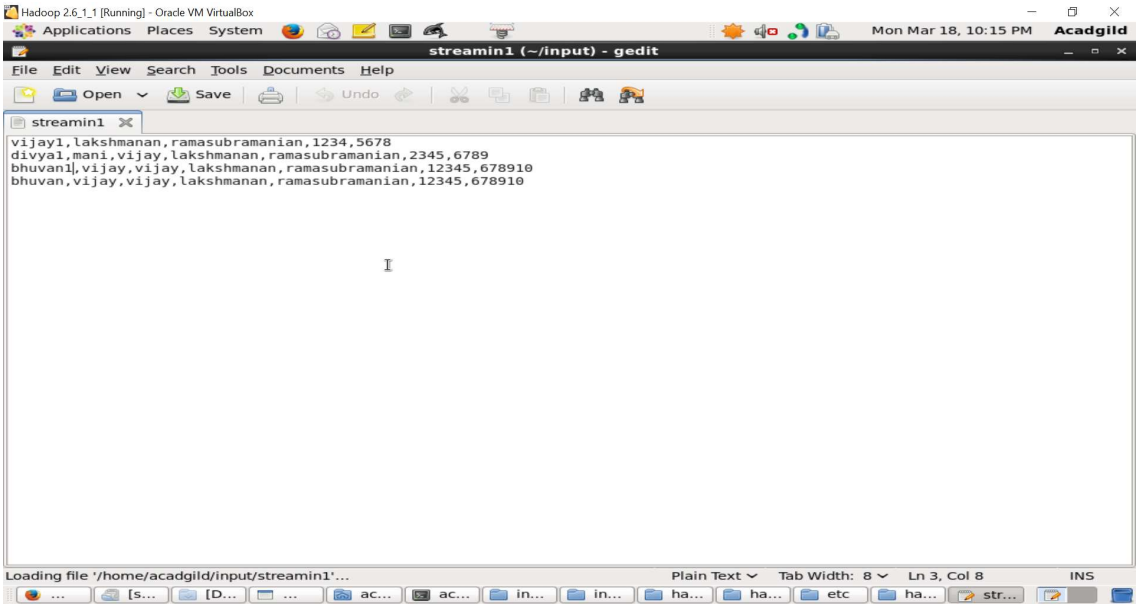
Code:



Output:



After modifying the local file as given below



Output:

```
19/03/18 22:15:31 INFO TaskSetManager: Finished task 0.0 in stage 7.0 (TID 5) in 126 ms on localhost (execut
19/03/18 22:15:31 INFO DAGScheduler: ResultStage 7 (collect at vjspstream2.scala:48) finished in 0.184 s
Both are different
19/03/18 22:15:31 INFO DAGScheduler: Job 3 finished: collect at vjspstream2.scala:48, took 0.199089 s
19/03/18 22:15:31 INFO TaskSchedulerImpl: Removed TaskSet 7.0, whose tasks have all completed, from pool
19/03/18 22:15:31 INFO SparkContext: Invoking stop() from shutdown hook
```