BIG DATA HADOOP AND SPARK DEVLOPMENT

ASSIGNMENT 26

Table of Contents:

BIG DATA HADOOPAND SPARK DEVELOPMENT

1. Introduction

In this assignment, the given tasks are performed and Output of the tasks are recorded in the form of Screenshots.

2. Objective

This Assignment consolidates the deeper understanding of the Session – 26 Spark streaming

3. Problem Statement

- ## Task 1

  o Read a stream of Strings, fetch the words which can be converted to numbers. Filter out the rows, where the sum of numbers in that line is odd. Provide the sum of all the remaining numbers in that batch.

- ## Task 2

  o Read two streams
    1. List of strings input by user
    2. Real-time set of offensive words
    Find the word count of the offensive words inputted by the user as per the real-time set of offensive words

4. Expected Output

- ## Task 1
  - o Read a stream of Strings, fetch the words which can be converted to numbers. Filter out the rows, where the sum of numbers in that line is odd. Provide the sum of all the remaining numbers in that batch.

- Here, we have a list of words and associate integer to it in a variable "wordtoNumbers"
- Broadcast this variable to a variable called "wordtoBroadcast"
- To match the words with numbers, we write a function "linewordNumbers", which takes input as argument and then we split the input with space and broadcast the words to numbers.
- We create a streaming context, which takes the input every 15 seconds.
- We take the input from the netcat with host as localhost and port number 9999
- We find the count of evenlines by taking the sum of the numbers associated with each word and then find if the number is even or not. If it is even, we print the number of evenlines, if not we just print the inputted data.
- This is program is executed using spark shell

```scala
scala> import org.apache.spark.streaming._
import org.apache.spark.streaming._

scala> import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.{SparkConf, SparkContext}

scala>

scala> val evenlines = sc.accumulator(0)
warning: there were two deprecation warnings; re-run with -deprecation for details
evenlines: org.apache.spark.Accumulator[Int] = 0

scala>  val wordtoNumbers = Map("Hi" -> 1, "this" -> 2, "is" -> 3, "BDH" -> 4, "Session" -> 5, "it" -> 6, "is" -> 7, "a" -> 8, "W
ondeful" -> 9, "session" -> 10)
wordtoNumbers: scala.collection.immutable.Map[String,Int] = Map(this -> 2, is -> 7, it -> 6, a -> 8, Session -> 5, BDH -> 4, sess
ion -> 10, Hi -> 1, Wondeful -> 9)

scala> val wordtoBroadcast = sc.broadcast(wordtoNumbers)
wordtoBroadcast: org.apache.spark.broadcast.Broadcast[scala.collection.immutable.Map[String,Int]] = Broadcast(1)

scala>

scala> def linewordNumbers(line:String):Int ={
     |    var sum:Int=0
     |    val words = line.split(" ")
     |    for(word <- words)sum +=wordtoBroadcast.value.get(word).getOrElse(0)
     |    sum
     |    }
linewordNumbers: (line: String)Int

scala>    val ssc = new StreamingContext(sc, Seconds(15))
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@4d3a7025

scala>    val stream = ssc.socketTextStream("localhost", 9999)
stream: org.apache.spark.streaming.dstream.ReceiverInputDStream[String] = org.apache.spark.streaming.dstream.SocketInputDStream@b
8f6500
```

Output is marked in blue and red boxes as shown below:

```
scala>

scala> stream.foreachRDD(line =>{val linestr = line.collect().toList.mkString(" ")
     |    if(linestr != ""){var numTotal = linewordNumbers(linestr);if(numTotal %2 ==1) println(linestr)
     | else{evenlines +=    numTotal
     |    println("Sum of lines with even words so far: " +evenlines.value.toInt)}}})

scala>    ssc.start()

scala>    ssc.awaitTermination()
18/06/09 17:36:32 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:36:32 WARN storage.BlockManager: Block input-0-1528545992600 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:36:35 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:36:35 WARN storage.BlockManager: Block input-0-1528545995000 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:36:43 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:36:43 WARN storage.BlockManager: Block input-0-1528546003600 replicated to only 0 peer(s) instead of 1 peers
hi this is BDH session
18/06/09 17:36:49 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:36:49 WARN storage.BlockManager: Block input-0-1528546009200 replicated to only 0 peer(s) instead of 1 peers
it is wonderful session
18/06/09 17:37:55 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:37:55 WARN storage.BlockManager: Block input-0-1528546074800 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:37:57 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:37:57 WARN storage.BlockManager: Block input-0-1528546077000 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:37:58 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:37:58 WARN storage.BlockManager: Block input-0-1528546078000 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even words so far: 10
18/06/09 17:38:01 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:38:01 WARN storage.BlockManager: Block input-0-1528546081000 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even words so far: 24
18/06/09 17:38:45 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:38:45 WARN storage.BlockManager: Block input-0-1528546125200 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:38:47 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:38:47 WARN storage.BlockManager: Block input-0-1528546127400 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:38:49 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:38:49 WARN storage.BlockManager: Block input-0-1528546129000 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:38:55 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:38:55 WARN storage.BlockManager: Block input-0-1528546135000 replicated to only 0 peer(s) instead of 1 peers
```

```
18/06/09 17:38:55 WARN storage.BlockManager: Block input-0-1528546135000 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:38:59 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:38:59 WARN storage.BlockManager: Block input-0-1528546139200 replicated to only 0 peer(s) instead of 1 peers
it is wonderful session Hi this is BDH session
18/06/09 17:39:03 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:39:03 WARN storage.BlockManager: Block input-0-1528546143600 replicated to only 0 peer(s) instead of 1 peers
it is wonderful session
18/06/09 17:40:34 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:40:34 WARN storage.BlockManager: Block input-0-1528546234200 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:40:35 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:40:35 WARN storage.BlockManager: Block input-0-1528546235600 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:40:37 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:40:37 WARN storage.BlockManager: Block input-0-1528546236800 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:40:39 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:40:39 WARN storage.BlockManager: Block input-0-1528546238800 replicated to only 0 peer(s) instead of 1 peers
18/06/09 17:40:40 WARN storage.RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
18/06/09 17:40:40 WARN storage.BlockManager: Block input-0-1528546240400 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even words so far: 48
```

This is the input given by user in netcat:

```
[acadgild@localhost ~]$ nc -lk 9999
hi
this is
BDH session
it is wonderful session
Hi
this
is
BDH session
it is
wonderful
session
Hi this
is BDH session
it is wonderful session
Hi
this
is
BDH
session
```

Task 2:

- Read two streams
  1. List of strings input by user

  2. Real-time set of offensive words

Find the word count of the offensive words inputted by the user as per the real-time set of offensive words

- ➢ Here we compare two list of strings, one list contains the list of offensive words which is saved in an array and other list of strings are the real time data which are inputted by user using netcat.
- ➢ After comparison, we find the number of times the user has inputted that word in the real time data set by comparing with the list of words given in the array.
- ➢ Below program is written and executed in IntelliJ

```
package SparkstreamingAssignments

//packages required for the program

import org.apache.spark.SparkConf
import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.{Seconds, StreamingContext}
import scala.collection.mutable.ArrayBuffer

object offensive {
    //ArrayBuffer to store list of offensive words in memory
    val wordList: ArrayBuffer[String] = ArrayBuffer.empty[String]



// main function which takes two arguments, one argument takes the hostname and other
//port
    def main(args: Array[String]) {
      if (args.length < 2) {
        System.err.println("Usage: OffensiveWordCount <hostname> <port>")
        System.exit(1)
      }

      // Create the spark context and spark configuration with a 60 second batch size
        val sparkConf = new
SparkConf().setAppName("OffensiveWordCount").setMaster("local[*]")
        val ssc = new StreamingContext(sparkConf, Seconds(60))
// created a variable of array with offensive words
        val words = Array("shit","damn","idiot","stupid","dash")
// creating connection with netcat and store the data in the memory and disk
        val lines = ssc.socketTextStream(args(0), args(1).toInt,
StorageLevel.MEMORY_AND_DISK_SER)
//now we split the data inputted by user with a space and convert the inputted the //data
to lowercase and filter them with words matching with array and count them
        val wordcounts = lines.flatMap(line => line.split(" ").filter(w=>
words.contains(w.toLowerCase)).map(word =>(word,1))).reduceByKey(_ + _)
// print the contents
      wordcounts.print()

      ssc.start()
      ssc.awaitTermination()
    }
}
```
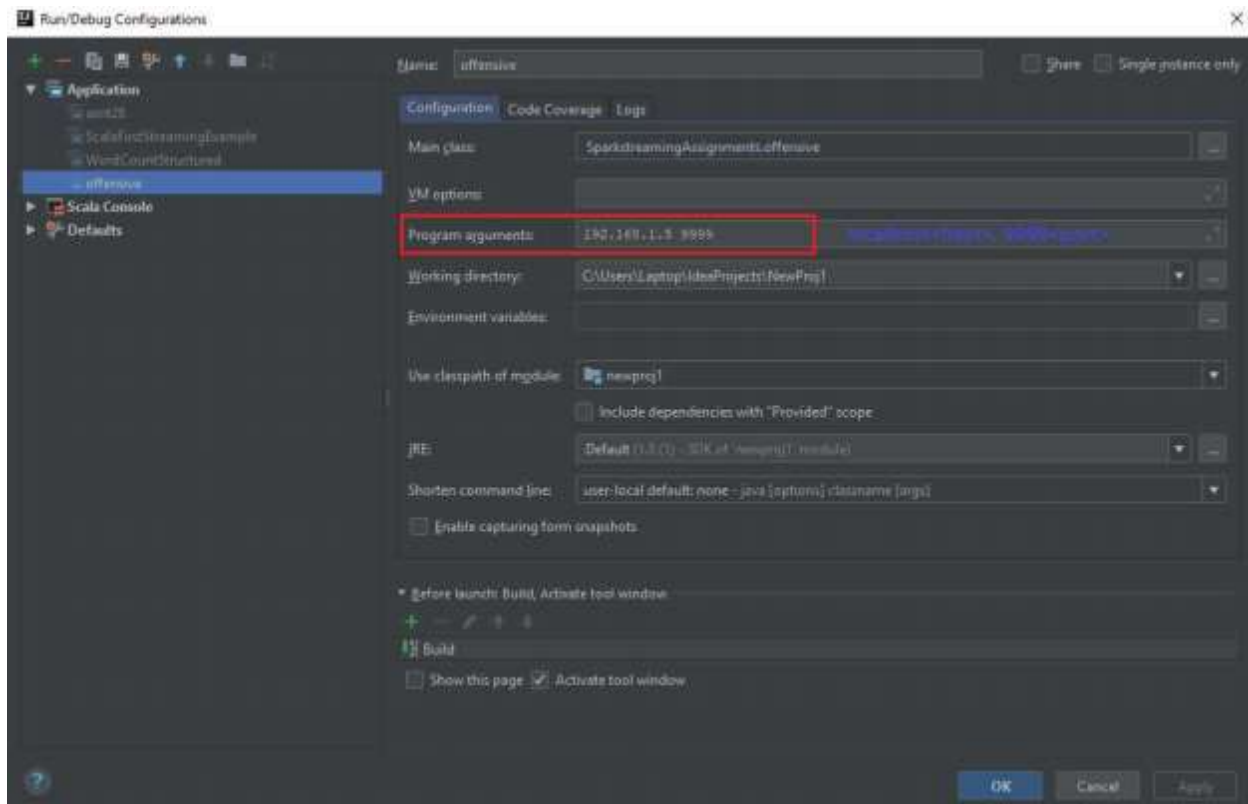
We provide the arguments in edit configurations, we provide ipaddress of VM and connecting port i.e. 9999

We open netcat and provide the input:

Output after we run the program, and we can see the counts of the offensive words entered.

```
18/06/14 11:47:00 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 11 ms
18/06/14 11:47:00 INFO Executor: Finished task 0.0 in stage 4.0 (TID 9). 1639 bytes result sent to driver
18/06/14 11:47:00 INFO TaskSetManager: Finished task 0.0 in stage 4.0 (TID 9) in 57 ms on localhost (executor driver) (1/3)
18/06/14 11:47:00 INFO Executor: Finished task 1.0 in stage 4.0 (TID 10). 1800 bytes result sent to driver
18/06/14 11:47:00 INFO Executor: Finished task 2.0 in stage 4.0 (TID 11). 1893 bytes result sent to driver
18/06/14 11:47:00 INFO TaskSetManager: Finished task 1.0 in stage 4.0 (TID 10) in 74 ms on localhost (executor driver) (2/3)
18/06/14 11:47:00 INFO TaskSetManager: Finished task 2.0 in stage 4.0 (TID 11) in 74 ms on localhost (executor driver) (3/3)
18/06/14 11:47:00 INFO TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all completed, from pool
18/06/14 11:47:00 INFO DAGScheduler: ResultStage 4 (print at offensive.scala:24) finished in 0.084 s
18/06/14 11:47:00 INFO DAGScheduler: Job 2 finished: print at offensive.scala:24, took 0.113255 s

-------------------------------------------
Time: 1528957020000 ms
-------------------------------------------
(dash,2)
(damn,1)
(idiot,3)
(stupid,2)


18/06/14 11:47:00 INFO JobScheduler: Finished job streaming job 1528957020000 ms.0 from job set of time 1528957020000 ms
18/06/14 11:47:00 INFO JobScheduler: Total delay: 0.620 s for time 1528957020000 ms (execution: 0.542 s)
18/06/14 11:47:00 INFO ReceivedBlockTracker: Deleting batches:
18/06/14 11:47:00 INFO InputInfoTracker: remove old batch metadata:
```