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.

3. Output

Code and screenshots

Cd workspace-eclipse

Cd wordOdd

Sbt package

Sbt eclipse

***import org.apache.spark.\_***

***import org.apache.spark.streaming.\_***

***import scala.collection.mutable.ListBuffer***

***object WordOdd {***

***def main(args:Array[String]) {***

***val SparkConf = new SparkConf().setAppName("WordOdd").setMaster("local[2]")***

***// Create a local StreamingContext with batch interval of 10 second***

***val ssc = new StreamingContext(SparkConf, Seconds(10))***

***/\* Create a DStream that will connect to hostname and port, like localhost 9999. As stated earlier, DStream will get created from StreamContext, which in return is created from SparkContext. \*/***

***val lines = ssc.socketTextStream("localhost",9999)***

***// Using this DStream (lines) we will perform transformation or output operation.***

***def toInt(s: String): Option[Int] = {***

***try {***

***Some(Integer.parseInt(s.trim))***

***} catch {***

***// catch Exception to catch null 's'***

***case e: Exception => None***

***}***

***}***

***val mainstring = lines***

***val sentence = mainstring.split("\\W+")***

***val testing = sentence.flatMap(toInt);***

***var oddLines= new ListBuffer[String]()***

***var sum = testing.sum***

***if(sum %2 == 1){***

***oddLines+= mainstring***

***}***

***oddLines.foreach(println)***

***ssc.start() // Start the computation***

***ssc.awaitTermination() // Wait for the computation to terminate***

***}***

***}***