ASSIGNMENT 24 Spark Streaming

To solve the both the tasks, we create the spark application, which has two Scala files with object Even_Number_Line.scala and Offensive_Words_Count.scala.

Task 1

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.

Below screen shot shows the spark application to filter the lines containing even numbers

```
import org.apache.spark.{SparkConf,SparkContext}

import org.apache.spark.streaming.{Seconds, StreamingContext}

object Even_Number_Line{

  def main(args: Array[String]):Unit = {


    // create a function which
  def Get_Lines_Sum(input : String) : Double ={

    val line = input.split(" ")
     var number : Double = 0.0
    for (x <- line)
      {
      try{
        val value = x.toDouble
      number = number + value
      }
      catch
        {
        case ex : Exception => {}
      }
    }
     return number
    }

    println("This is the task1 of assignment session 26")
```

```
  println("This is the task1 of assignment session 26")

val conf = new SparkConf().setMaster("local[2]").setAppName("EvenLines")
val sc = new SparkContext(conf)

sc.setLogLevel("WARN")
println("Spark Context Created")

// Create a local StreamingContext with working thread and batch interval of 20 seconds
val ssc = new StreamingContext(sc, Seconds(30))

println("Spark Streaming Context Created")

// Create a DStream that will connect to hostname:port,localhost:9999
val lines = ssc.socketTextStream( hostname="localhost", port= 9999)

//filter the even string from input line by using Get_Lines function
val lines_filter = lines.filter(x => Get_Lines_Sum(x)%2 == 0)

//add all the numbers the even string from input line by using Get_Lines function
val lines_sum = lines_filter.map(x => Get_Lines_Sum(x))
```

```
44
45  //filter the even string from input line by using Get_Lines function
46  val lines_filter = lines.filter(x => Get_Lines_Sum(x)%2 == 0)
47
48  //add all the numbers the even string from input line by using Get_Lines function
49  val lines_sum = lines_filter.map(x => Get_Lines_Sum(x))
50
51    println("Lines with even sum:")
52    lines_filter.print()
53
54    println("Sum of the numbers in even lines:")
55    lines_sum.reduce(_+_).print()
56
57    // Start the computation
58  ssc.start()
59
60  // Wait for the computation to terminate
61  ssc.awaitTermination()
62  }
63
64 }
```

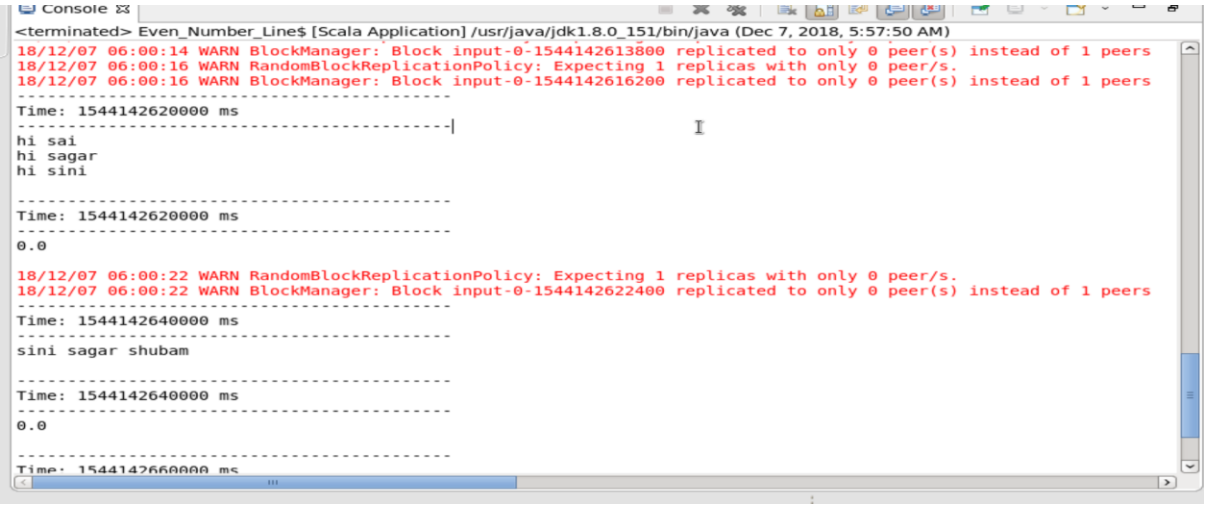
Writable | Smart Insert | 51 : 36

eclipse-workspace - sp...

Net cat command to send the stream data

```
[acadgild@localhost Desktop]$ nc -lk 9999
hi sai
hi sagar
hi sini
sini sagar shubam
^C
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Desktop]$
```

Output of console



Task 2

Read two streams

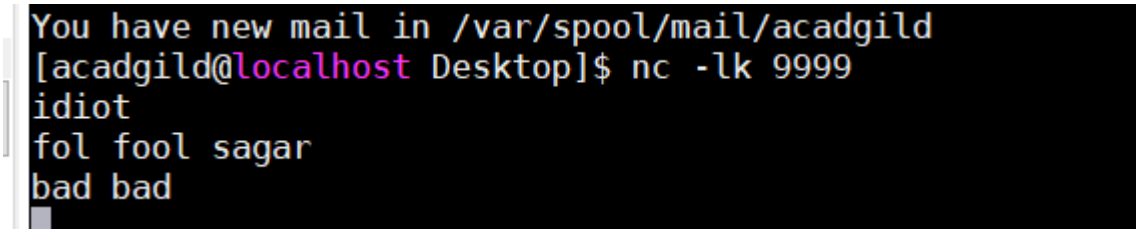
1. List of strings input by user 2. Real-time set of offensive words Find the word count of the offensive words inputted by the user as per the real-time set of offensive words

Below screen shot shows the spark application to filter the offensive words form input string entered by use

```
object Offensive_Word_Count {
  def main(args: Array[String]):Unit = {
    println("This is the task2 of assignment session 26")

  val conf = new SparkConf().setMaster("local[2]").setAppName("SparkStreamingExample")
  val sc = new SparkContext(conf)

  sc.setLogLevel("WARN")
  println("Spark Context Created")

  //create a set of offensive words which we use to compare and filter these words from input string
    val offensive_word_list: Set[String] = Set("idiot", "fool", "bad", "nonsense")
  //print the list of these offensive words
  println(s"$offensive_word_list")
  // Create a local StreamingContext with working thread and batch interval of 20 seconds
  val ssc = new StreamingContext(sc, Seconds(20))

  println("Spark Streaming Context Created !")
  // Create a DStream that will connect to hostname:port,localhost:9999
  val lines = ssc.socketTextStream( hostname="localhost", port= 9999)
    // Split each line into words
    val words = lines.flatMap(_.split(" ")).map (x => x)
    // filter the offensive words from input string by using set and count words
  val Offensive_Word_Count = words.filter(x => offensive_word_list.contains(x)).map(x => (x, 1)).reduceByKey
  Offensive_Word_Count.print()
    // Start the computation
  ssc.start()
  // Wait for the computation to terminate
```

Netcat command to send live stream data

In the above spark application, we have a set of words that we considered as offensive words. “idiot”,”fool”,”bad” ,“nonsense”



Output Console