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

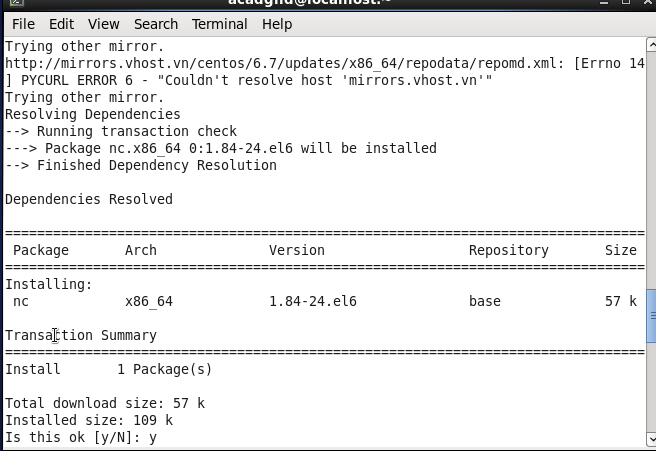
line is odd.

Provide the sum of all the remaining numbers in that batch

Sudo yum install nc

Nc -lk 9999

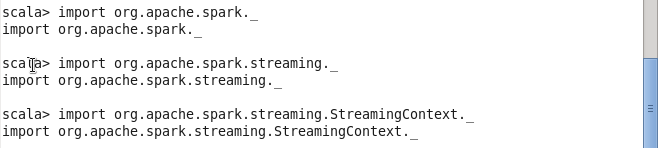
/home/acadgild/spark-2.2.1-bin-hadoop2.7/bin/spark-shell –master local[4]

****

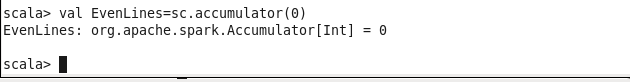
Import org.apache.spark.\_

Import org.apache.spark.streaming.\_

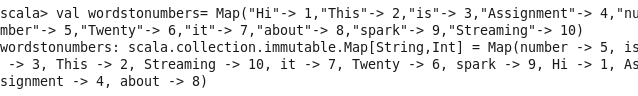
Import org.apache.spark.streaming.StreamingContext.\_

****

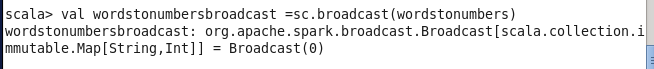
Val EvalLines=sc.accumulator(0)

****

Val wordstonumbers = map(“Hi” -> 1,”This” -> 2,”is” ->3.”Assignment”- >4,”number”->5,”Twenty” ->6,”it”->7,”about”->8,”spark”->9,”Streaming”->10)



Val wordstonumbersbroadcast=sc.broadcast(wordstonumbers)



Def lineWordNumberSum(line:String):Int= {

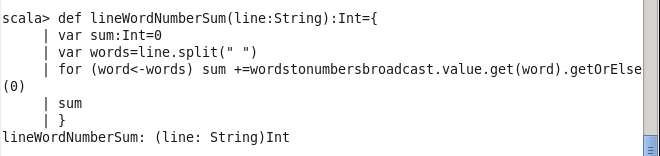
Var sum:Int =0

Var words=line.split(“ “)

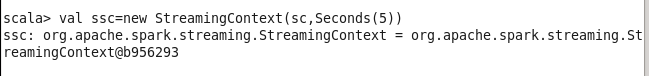
For (word->words)sum += wordstonumbersbroadcast.value.get(word).getOrElse(0)

sum

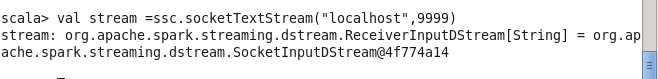
}



Val ssc= new StreamingContext(sc,Seconds(5))



Val stream = ssc.socketTextSteam(“localhost”,9999)



Stream.foreachRDD(line=>{val lineStr=line.collect().toList.mkString(“”)

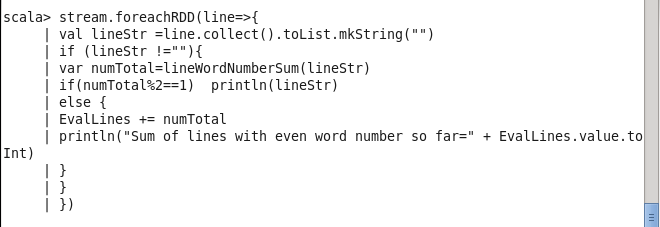
If(line !=””) {var numTotal =lineWordNumberSum(lineStr) if (numTotal % 2) println(linestr)

Else

{Evenlines +=numTotal

Println(“Sum of lines with even word number so far =” +EvenLines.value.toInt)}}

})



Ssc.start()

Ssc.awaitTermination()

**Screen Shot as below :**

**\**