# Assignment 20.1

Problem Statement:-

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.

Solution:-

Step1: Start Spark Shell with 4 threads

Use the command below to start spark-shell with 4 threads

spark-shell --master local[4]

Screenshot is as below:

```
[acadgild@localhost assignment_20.2]$
[acadgild@localhost assignment_20.2]$
[acadgild@localhost assignment_20.2]$
[acadgild@localhost assignment_20.2]$ spark-shell --master local[4]
log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Using Spark's repl log4j profile: org/apache/spark/log4j-defaults-repl.properties
To adjust logging level use sc.setLogLevel("INFO")
Welcome to


      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_65)
Type in expressions to have them evaluated.
Type :help for more information.
17/12/24 18:24:18 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 192.168.0
.4 instead (on interface eth6)
17/12/24 18:24:18 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Spark context available as sc.
17/12/24 18:24:27 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/24 18:24:28 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/24 18:24:34 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not ena
bled so recording the schema version 1.2.0
17/12/24 18:24:34 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
17/12/24 18:24:36 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
17/12/24 18:24:37 WARN Connection: BoneCP specified but not present in CLASSPATH (or one of dependencies)
SQL context available as sqlContext.

scala>

scala>
```

- Step2: Declare all the packages

Declare spark and streaming packages as below:

import org.apache.spark._

import org.apache.spark.streaming._

import org.apache.spark.streaming.StreamingContext._

Step3: Declare a accumulator

Declare accumulator totalEvenLinesWordNumber which will keep track of sum of number of word numbers in lines so far

val totalEvenLinesWordNumber = sc.accumulator(0)

## Step4: Define a wordNumberMap  map for converting word to number

Define a map for converting word to number. If word is not there in map then 0 will be returned. Broadcast the map. Code is as below

```
val wordNumberMap = Map("Hi" -> 1, "my" -> 2, "name" -> 3, "is" -> 4, "Hello" ->5, "Monimoy" -> 6, "John" -> 7, "Bob"->8,    "Vibhu" ->9)
val wordNumberMapBroadcast = sc.broadcast(wordNumberMap)
```

## Step4: Define a function to return sum of word converted to number in a line

Define a function lineWordNumberTotal which will split a line based on blank space to get all the words in a next. Next in the lookup wordNumberMapBroadcast , based on word, corresponding number is retrieved and sum all these numbers together.

Code is as below:

```
def lineWordNumberTotal(line:String):Int = {

   var sum:Int = 0

   var words = line.split(" ")

   for (word <- words) sum += wordNumberMapBroadcast.value.get(word).getOrElse(0)

   sum

}
```

Screenshot of steps 2 to 4 is as below:

```
scala>

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

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

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

scala>

scala> val totalEvenLinesWordNumber = sc.accumulator(0)
totalEvenLinesWordNumber: org.apache.spark.Accumulator[Int] = 0

scala> val wordNumberMap = Map("Hi" -> 1, "my" -> 2, "name" -> 3, "is" -> 4, "Hello" ->5, "Monimoy" -> 6, "John" -> 7, "Bob"-
>8,      "Vibhu" ->9)
wordNumberMap: scala.collection.immutable.Map[String,Int] = Map(name -> 3, John -> 7, Vibhu -> 9, is -> 4, Bob -> 8, my -> 2,
 Hello -> 5, Hi -> 1, Monimoy -> 6)

scala> val wordNumberMapBroadcast = sc.broadcast(wordNumberMap)
wordNumberMapBroadcast: org.apache.spark.broadcast.Broadcast[scala.collection.immutable.Map[String,Int]] = Broadcast(0)

scala>

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

## Step5: Start Text Streaming

- Start text streaming on localhost with port number 9999 and interval 15 seconds and return the stream. Code is as below:

val ssc = new StreamingContext(sc, Seconds(15))

val stream = ssc.socketTextStream("localhost", 9999)

## Step6: Process each RDD in stream

Process each RDD in stream. First convert the RDD to string. If it is not blank calculate word number for each word and sum them  using function

lineWordNumberTotal and put to variable numTotal. If numTotal is odd, print the corresponding line. Also, add numTotal to accumulator accu totalEvenLinesWordNumber and print the sum

```scala
stream.foreachRDD(line => {

    val lineStr = line.collect().toList.mkString("")

    if (lineStr != "") {

       var numTotal = lineWordNumberTotal(lineStr)

       if (numTotal % 2 == 1) println(lineStr)

       else {

          totalEvenLinesWordNumber += numTotal

          println("Sum of lines with even word number so far =" +
totalEvenLinesWordNumber.value.toInt)

       }

    }

})
```

Step7: Start the streams

Start the streams and wait till its termination. Code is as below:

```scala
ssc.start()

ssc.awaitTermination()
```

Screenshot for steps 5 to 7 is as below:

```
scala>

scala> val ssc = new StreamingContext(sc, Seconds(15))
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@6b42829

scala> val stream = ssc.socketTextStream("localhost", 9999)
stream: org.apache.spark.streaming.dstream.ReceiverInputDStream[String] = org.apache.spark.streaming.dstream.SocketInputDStre
am@705ba09a

scala> stream.foreachRDD(line => {
     |     val lineStr = line.collect().toList.mkString("")
     |     if (lineStr != "") {
     |         var numTotal = lineWordNumberTotal(lineStr)
     |         if (numTotal % 2 == 1) println(lineStr)
     |         else {
     |             totalEvenLinesWordNumber += numTotal
     |             println("Sum of lines with even word number so far =" + totalEvenLinesWordNumber.value.toInt)
     |         }
     |     }
     | } )

scala> ssc.start()
```

Step8: Start netcat from a terminal

From a terminal start netcat on port 9999 using nc ommand below and start typing lines

nc -lk 9999

The screenshot is as below:

```
[acadgild@localhost ~]$
[acadgild@localhost ~]$ nc -lk 9999
Hi
Hi Monimoy
Hi Rob
Hi John
Hi my name is Vibhu
Hello All
Hi my name is Bob
Hello Monimoy
Hello how are you doing
Hi Hi
Hi my name is Rob
```

Step9: Display the output

The lines with odd numbered word number sum will be displayed. For lines with even numbered word number, the summation done so far will be displayed. The screenshot is as below:

For example

"Hi Monimoy" has total word number value 7 which is odd, so the line will be displayed

"Hi John" has total word number value 8 which is even number so summation will be displayed

```
        |       val lineStr = line.collect().toList.mkString("")
        |       if (lineStr != "") {
        |           var numTotal = lineWordNumberTotal(lineStr)
        |           if (numTotal % 2 == 1) println(lineStr)
    I   |           else {
        |               totalEvenLinesWordNumber += numTotal
        |               println("Sum of lines with even word number so far =" + totalEvenLinesWordNumber.value.toInt)
        |           }
        |       }
        | } )

scala> ssc.start()

scala> ssc.awaitTermination()
17/12/26 22:31:11 WARN BlockManager: Block input-0-1514307671400 replicated to only 0 peer(s) instead of 1 peers
Hi
17/12/26 22:31:22 WARN BlockManager: Block input-0-1514307682600 replicated to only 0 peer(s) instead of 1 peers
Hi Monimoy
17/12/26 22:31:38 WARN BlockManager: Block input-0-1514307698000 replicated to only 0 peer(s) instead of 1 peers
Hi Rob
17/12/26 22:32:14 WARN BlockManager: Block input-0-1514307734600 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even word number so far =8
17/12/26 22:32:58 WARN BlockManager: Block input-0-1514307777800 replicated to only 0 peer(s) instead of 1 peers
Hi my name is Vibhu
17/12/26 22:33:14 WARN BlockManager: Block input-0-1514307794000 replicated to only 0 peer(s) instead of 1 peers
Hello All
17/12/26 22:33:46 WARN BlockManager: Block input-0-1514307825800 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even word number so far =26
17/12/26 22:34:11 WARN BlockManager: Block input-0-1514307850800 replicated to only 0 peer(s) instead of 1 peers
Hello Monimoy
17/12/26 22:35:00 WARN BlockManager: Block input-0-1514307900400 replicated to only 0 peer(s) instead of 1 peers
Hello how are you doing
17/12/26 22:35:33 WARN BlockManager: Block input-0-1514307933600 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even word number so far =28
17/12/26 22:36:00 WARN BlockManager: Block input-0-1514307960000 replicated to only 0 peer(s) instead of 1 peers
Sum of lines with even word number so far =38
```