

ASSIGNMENT 20.2

Problem Statement:

Read two streams:

1. List of strings input by user
2. Real-time set of offensive words

Find the word count of offensive words input by a user as per the real-time set of offensive words.

Solution:

Here is the Spark code in Scala to create two streams and find the set of offensive words every minute over socket connection:

OffensiveWordCount.scala:

```
package org.spark

// import required Spark packages

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

/*
 * Counts words in UTF-8 encoded, '\n' delimited text received from the network every second.
 *
 * Usage: OffensiveWordCount <hostname> <port>
 * <hostname> and <port> describes TCP server that Spark Streaming would connect to receive data.
 * To run this on your local machine, you need to first run a Netcat server
 * ` $ nc -lk 9999 `
 */
```

```

object OffensiveWordCount {
  //ArrayBuffer to store a list of offensive words in memory
  val wordList: ArrayBuffer[String] = ArrayBuffer.empty[String]
  def main(args: Array[String]) {

    // check whether user has passed sufficient command line arguments for execution of the program
    if (args.length < 2) {
      System.err.println("Usage: OffensiveWordCount <hostname> <port>")
      System.exit(1)
    }

    // create a streaming context with 60 seconds batch size
    val sparkConf = new SparkConf().setAppName ("OffensiveWordCount")
    val ssc = new StreamingContext(sparkConf, Seconds(60))

    // create a text file stream to monitor an HDFS directory for any text file in it having a set of
    // offensive words
    val offensiveLines = ssc.textFileStream ("hdfs://localhost:9000/offensiveWords/")

    // create a socket stream on target ip:port and count the words in input stream of \n delimited text
    // (eg. generated by 'nc'). Note that no duplication in storage level only for running locally.
    // Replication necessary in distributed scenario for fault tolerance.
    val lines = ssc.socketTextStream(args(0), args(1).toInt, StorageLevel.MEMORY_AND_DISK_SER)

    // getting offensive words from file
    val offensiveWordCount = offensiveLines.flatMap (line => line.split(" "))
      .map (word => (word, 1))
      .reduceByKey (_ + _)

    //storing offensive words in an ArrayBuffer
    offensiveWordCount.foreachRDD (x => { x.foreach (words => { wordList += words._1 }) })
  }
}

```

```

// get count of all words entered by user
val wordCount = lines.flatMap (line => line.split (" "))
                        .map (word => (word, 1))
                        .reduceByKey (_ + _)

// get word count of only the offensive words matched from the other stream
val offensiveWordsRDD = wordCount.filter {x => (matchWord (x._1)%2) ==1 }
offensiveWordsRDD.print()

// start the streaming operation and keep it running until the program gets terminated
ssc.start()
ssc.awaitTermination()
}

/*
 * Filter Method for offensive words
 */

def matchWord(ln : String): Double={
    val lineWords = ln.trim.toLowerCase()
    var num: Double = 0

    // compare the word passed by user with each of those from words list
    // return 1 in case they match
    for(y<-wordList) {
        if(y.toLowerCase() == lineWords) {
            num = 1
            num
        }
    }
    num // return 1 for words that match with the list of words stored
}
}

```

```
1 package org.spark
2
3 import org.apache.spark.SparkConf
4 import org.apache.spark.storage.StorageLevel
5 import org.apache.spark.streaming.{Seconds, StreamingContext}
6 import scala.collection.mutable.ArrayBuffer
7 import org.apache.spark.streaming.dstream.DStream
8 import org.apache.spark.SparkContext
9
10 /**
11  * Counts words in UTF8 encoded, '\n' delimited text received from the network every second.
12  *
13  * Usage: OffensiveWordCount <hostname> <port>
14  * <hostname> and <port> describe the TCP server that Spark Streaming would connect to receive data.
15  *
16  * To run this on your local machine, you need to first run a Netcat server
17  * $ nc -lk 9999
18  */
19 object OffensiveWordCount {
20   //ArrayBuffer to store list of offensive words in memory
21   val wordList: ArrayBuffer[String] = ArrayBuffer.empty[String]
22
23   def main(args: Array[String]) {
24     if (args.length < 2) {
25       System.err.println("Usage: OffensiveWordCount <hostname> <port>")
26       System.exit(1)
27     }
28
29     // Create the context with a 60 second batch size
30     val sparkConf = new SparkConf().setAppName("OffensiveWordCount")
31     val ssc = new StreamingContext(sparkConf, Seconds(60))
32
33     //Creating text file stream to store offensive words.
34     //It will monitor HDFS directory /offensiveWords
35     val offensiveLines = ssc.textFileStream("hdfs://localhost:9000/offensiveWords/")
36     val lines = ssc.socketTextStream(args(0), args(1).toInt, StorageLevel.MEMORY_AND_DISK_SER)
37
38     // Create a socket stream on target ip:port and count the
39     // words in input stream of \n delimited text (eg. generated by 'nc')
40     // Note that no duplication in storage level only for running locally.
41     // Replication necessary in distributed scenario for fault tolerance.
42
43     //getting offensive words from file
44     val offensiveWordCount = offensiveLines.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
45     //storing offensive words in ArrayBuffer
46     offensiveWordCount.foreachRDD(a => { a.foreach(f => {wordList += f._1})})
47
48     //Getting all word count of all words entered by user
49     val wordCount = lines.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
50
51     //Getting word count of offensive words only
52     val offensiveWordsRDD = wordCount.filter {x => matchWord(x._1)%2==1 }
53     offensiveWordsRDD.print()
54
55     ssc.start()
56     ssc.awaitTermination()
57   }
58 }
```

```

62  /**
63   * Filter Method for offensive words
64   */
65  def matchWord(ln : String): Double={
66    val lineWords = ln.trim.toLowerCase()
67    var num: Double = 0
68
69    for(y<-wordList)
70    {
71      if(y.toLowerCase() == lineWords)
72      {
73        num = 1
74        return num
75      }
76    }
77  }
78
79  return num
80 }
81
82 }

```

Output:

Open a terminal and run netcat command to input some words which will be captured by the streaming application:

```
$ nc -lk 9999
```

```

[root@localhost acadgild]# nc -lk 9999
he is a dumb guy
she is an idiot

```

On another terminal run the following command to submit the jar file for execution of above code:

```
$ spark-submit --master local[2] --class org.spark.OffensiveWordCount Assignment20_1.jar localhost 9999
```

```

[acadgild@localhost scala-2.10]$ pwd
/home/acadgild/IdeaProjects/sampleProject/target/scala-2.10
[acadgild@localhost scala-2.10]$ ls
classes nohup.out resolution-cache sampleproject_2.10-0.1.jar
[acadgild@localhost scala-2.10]$ ls -ltr
total 132
drwxrwxr-x. 5 acadgild acadgild 4096 Mar  5 00:29 resolution-cache
drwxrwxr-x. 3 acadgild acadgild 4096 Mar  5 00:48 classes
-rw-r-----. 1 acadgild acadgild 91578 Mar  5 01:20 nohup.out
-rw-rw-r--. 1 acadgild acadgild 32597 Mar  5 01:38 sampleproject_2.10-0.1.jar
[acadgild@localhost scala-2.10]$
[acadgild@localhost scala-2.10]$ nohup spark-submit --master local[2] --class org.spark.OffensiveWordCount sampleproject_2.10-0.1.jar localhost 9999 &
[1] 10536
[acadgild@localhost scala-2.10]$ nohup: ignoring input and appending output to 'nohup.out'

```

As we enter some words on the first terminal we can see whether it is captured by our streaming application:

```
Applications Places System Mon Mar 5, 1:48 AM Acadgild
acadgild@localhost:~/IdeaProjects/sampleProject/target/scala-2.10
File Edit View Search Terminal Help
18/03/05 01:47:01 INFO MemoryStore: Block broadcast_10 stored as values in memory (estimated size 3.0 KB, free 103.7 KB)
18/03/05 01:47:01 INFO MemoryStore: Block broadcast_10_piece0 stored as bytes in memory (estimated size 1830.0 B, free 105.5 KB)
18/03/05 01:47:01 INFO BlockManagerInfo: Added broadcast_10_piece0 in memory on localhost:54952 (size: 1830.0 B, free: 517.4 MB)
18/03/05 01:47:01 INFO SparkContext: Created broadcast 10 from broadcast at DAGScheduler.scala:1006
18/03/05 01:47:01 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 18 (MapPartitionsRDD[30] at filter at OffensiveWordCount.scala:55)
18/03/05 01:47:01 INFO TaskSchedulerImpl: Adding task set 18.0 with 1 tasks
18/03/05 01:47:01 INFO TaskSetManager: Starting task 0.0 in stage 18.0 (TID 14, localhost, partition 1,PROCESS_LOCAL, 1962 bytes)
18/03/05 01:47:01 INFO Executor: Running task 0.0 in stage 18.0 (TID 14)
18/03/05 01:47:01 INFO ShuffleBlockFetcherIterator: Getting 0 non-empty blocks out of 0 blocks
18/03/05 01:47:01 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
18/03/05 01:47:01 INFO Executor: Finished task 0.0 in stage 18.0 (TID 14). 1161 bytes result sent to driver
18/03/05 01:47:01 INFO DAGScheduler: ResultStage 18 (print at OffensiveWordCount.scala:56) finished in 0.030 s
18/03/05 01:47:01 INFO DAGScheduler: Job 9 finished: print at OffensiveWordCount.scala:56, took 0.133729 s
-----
Time: 1520194620000 ms
-----
18/03/05 01:47:01 INFO JobScheduler: Finished job streaming job 1520194620000 ms.1 from job set of time 1520194620000 ms
18/03/05 01:47:01 INFO JobScheduler: Total delay: 1.125 s for time 1520194620000 ms (execution: 0.600 s)
18/03/05 01:47:01 INFO ShuffledRDD: Removing RDD 15 from persistence list
18/03/05 01:47:01 INFO TaskSetManager: Finished task 0.0 in stage 18.0 (TID 14) in 38 ms on localhost (1/1)
18/03/05 01:47:01 INFO TaskSchedulerImpl: Removed TaskSet 18.0, whose tasks have all completed, from pool
18/03/05 01:47:01 INFO MapPartitionsRDD: Removing RDD 14 from persistence list
18/03/05 01:47:01 INFO MapPartitionsRDD: Removing RDD 13 from persistence list
18/03/05 01:47:01 INFO MapPartitionsRDD: Removing RDD 12 from persistence list

18/03/05 01:48:00 INFO TaskSetManager: Starting task 0.0 in stage 24.0 (TID 18, localhost, partition 1,PROCESS_LOCAL, 1962 bytes)
18/03/05 01:48:00 INFO Executor: Running task 0.0 in stage 24.0 (TID 18)
18/03/05 01:48:00 INFO ShuffleBlockFetcherIterator: Getting 0 non-empty blocks out of 0 blocks
18/03/05 01:48:00 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
18/03/05 01:48:00 INFO Executor: Finished task 0.0 in stage 24.0 (TID 18). 1161 bytes result sent to driver
18/03/05 01:48:00 INFO TaskSetManager: Finished task 0.0 in stage 24.0 (TID 18) in 12 ms on localhost (1/1)
18/03/05 01:48:00 INFO TaskSchedulerImpl: Removed TaskSet 24.0, whose tasks have all completed, from pool
18/03/05 01:48:00 INFO DAGScheduler: ResultStage 24 (print at OffensiveWordCount.scala:56) finished in 0.005 s
18/03/05 01:48:00 INFO DAGScheduler: Job 12 finished: print at OffensiveWordCount.scala:56, took 0.043208 s
-----
Time: 1520194680000 ms
-----
18/03/05 01:48:00 INFO JobScheduler: Finished job streaming job 1520194680000 ms.1 from job set of time 1520194680000 ms
18/03/05 01:48:00 INFO JobScheduler: Total delay: 0.365 s for time 1520194680000 ms (execution: 0.203 s)
18/03/05 01:48:00 INFO ShuffledRDD: Removing RDD 25 from persistence list
18/03/05 01:48:00 INFO BlockManager: Removing RDD 25
18/03/05 01:48:00 INFO MapPartitionsRDD: Removing RDD 24 from persistence list
18/03/05 01:48:00 INFO BlockManager: Removing RDD 24
18/03/05 01:48:00 INFO MapPartitionsRDD: Removing RDD 23 from persistence list
18/03/05 01:48:00 INFO BlockManager: Removing RDD 23
```

We can see that there is nothing captured on our application run since we haven't loaded any text file containing a set of abusive words to an HDFS directory specified in our application code. Let's do this and see the difference.

```
[acadgild@localhost ~]$ cat OffensiveWords.txt
idiot
dumb
stupid
mudblood
[acadgild@localhost ~]$ hadoop fs -put OffensiveWords.txt /offensiveWords/
18/03/05 02:36:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
[acadgild@localhost ~]$ hadoop fs -ls /offensiveWords/
18/03/05 02:36:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 1 items
-rw-r--r-- 1 acadgild supergroup 27 2018-03-05 02:36 /offensiveWords/OffensiveWords.txt
[acadgild@localhost ~]$
```

```
acadgild@localhost:~/IdeaProjects/sampleProject/target/scala-2.10
File Edit View Search Terminal Help
18/03/05 03:00:02 INFO Executor: Running task 0.0 in stage 12.0 (T
18/03/05 03:00:02 INFO ShuffleBlockFetcherIterator: Getting 1 non-e
18/03/05 03:00:02 INFO ShuffleBlockFetcherIterator: Started 0 remot
18/03/05 03:00:02 INFO Executor: Finished task 0.0 in stage 12.0 (T
18/03/05 03:00:02 INFO TaskSetManager: Finished task 0.0 in stage 1
18/03/05 03:00:02 INFO TaskSchedulerImpl: Removed TaskSet 12.0, whc
18/03/05 03:00:02 INFO DAGScheduler: ResultStage 12 (print at Offer
18/03/05 03:00:02 INFO DAGScheduler: Job 6 finished: print at Offer
-----
Time: 1520199000000 ms
(idiot,1)
18/03/05 03:00:02 INFO JobScheduler: Finished job streaming job 152
18/03/05 03:00:02 INFO JobScheduler: Total delay: 2.712 s for time
18/03/05 03:00:02 INFO ShuffledRDD: Removing RDD 5 from persistence
18/03/05 03:00:02 INFO MapPartitionsRDD: Removing RDD 4 from persis
18/03/05 03:00:02 INFO MapPartitionsRDD: Removing RDD 3 from persis
18/03/05 03:00:02 INFO BlockManager: Removing RDD 5
18/03/05 03:00:02 INFO BlockManager: Removing RDD 4
18/03/05 03:00:02 INFO MapPartitionsRDD: Removing RDD 2 from persis
18/03/05 03:00:02 INFO UnionRDD: Removing RDD 1 from persistence li
18/03/05 03:00:02 INFO FileInputDStream: Cleared 0 old files that w
18/03/05 03:00:02 INFO MapPartitionsRDD: Removing RDD 10 from persi
18/03/05 03:00:02 INFO BlockManager: Removing RDD 2
18/03/05 03:00:02 INFO BlockManager: Removing RDD 3
18/03/05 03:00:02 INFO ShuffledRDD: Removing RDD 9 from persistence
18/03/05 03:00:02 INFO BlockManager: Removing RDD 1
18/03/05 03:00:02 INFO MapPartitionsRDD: Removing RDD 8 from persis
18/03/05 03:00:02 INFO BlockManager: Removing RDD 10
18/03/05 03:00:02 INFO MapPartitionsRDD: Removing RDD 7 from persis
18/03/05 03:00:02 INFO BlockManager: Removing RDD 9
18/03/05 03:00:02 INFO BlockRDD: Removing RDD 6 from persistence li
18/03/05 03:00:02 INFO BlockManager: Removing RDD 8
18/03/05 03:00:02 INFO BlockManager: Removing RDD 7
18/03/05 03:00:02 INFO BlockManager: Removing RDD 6
18/03/05 03:00:02 INFO SocketInputDStream: Removing blocks of RDD BlockRDD[6] at socketTextStream at OffensiveWordCount.scala
:38 of time 1520199000000 ms
```

```
root@localhost:/home/acadgild
File Edit View Search Terminal Help
Assignment-13-2-code.txt~ Documents Mu
Assignment-14-1-code Downloads NC
Assignment-14-1-code.txt eclipse Pi
Assignment-14-1-code.txt~ EvenLines.scala pr
[root@localhost acadgild]# gedit assingment-20-2-scala-code.t
[root@localhost acadgild]#
[root@localhost acadgild]#
[root@localhost acadgild]# nc -lk 9999
he is a dumb guy
she is an idiot
he is a stupid guy
he is an idiot
she is dumb
he is also stupid idiot
^C
[root@localhost acadgild]# nc -lk 9999
he is a dumb guy
she is dumb
he is also stupid idiot
he is a stupid guy
mudblood
^C
[root@localhost acadgild]# nc -lk 9999
he is a dumb guy
she is a stupid girl
he is an idiot
she is a dumb idiot
she don't know anything, she is dumb
he is a stupid guy
mudblood
```

```
acadgild@localhost:~/IdeaProjects/sampleProject/target/scala-2.10
File Edit View Search Terminal Help
18/03/05 03:01:00 INFO DAGScheduler: Submitting 1 missing tasks fro
iveWordCount.scala:55)
18/03/05 03:01:00 INFO TaskSchedulerImpl: Adding task set 18.0 with
18/03/05 03:01:00 INFO TaskSetManager: Starting task 0.0 in stage 1
)
18/03/05 03:01:00 INFO Executor: Running task 0.0 in stage 18.0 (T
18/03/05 03:01:00 INFO ShuffleBlockFetcherIterator: Getting 2 non-e
18/03/05 03:01:00 INFO ShuffleBlockFetcherIterator: Started 0 remot
18/03/05 03:01:00 INFO Executor: Finished task 0.0 in stage 18.0 (T
18/03/05 03:01:00 INFO TaskSetManager: Finished task 0.0 in stage 1
18/03/05 03:01:00 INFO TaskSchedulerImpl: Removed TaskSet 18.0, whc
18/03/05 03:01:00 INFO DAGScheduler: ResultStage 18 (print at Offer
18/03/05 03:01:00 INFO DAGScheduler: Job 9 finished: print at Offer
-----
Time: 1520199060000 ms
(dumb,2)
(idiot,1)
18/03/05 03:01:00 INFO JobScheduler: Finished job streaming job 152
18/03/05 03:01:00 INFO JobScheduler: Total delay: 0.632 s for time
18/03/05 03:01:00 INFO ShuffledRDD: Removing RDD 16 from persistence
18/03/05 03:01:00 INFO BlockManager: Removing RDD 16
18/03/05 03:01:00 INFO MapPartitionsRDD: Removing RDD 15 from persi
18/03/05 03:01:00 INFO BlockManager: Removing RDD 15
18/03/05 03:01:00 INFO MapPartitionsRDD: Removing RDD 14 from persi
18/03/05 03:01:00 INFO BlockManager: Removing RDD 14
18/03/05 03:01:00 INFO MapPartitionsRDD: Removing RDD 13 from persi
18/03/05 03:01:00 INFO BlockManager: Removing RDD 13
18/03/05 03:01:00 INFO UnionRDD: Removing RDD 12 from persistence
18/03/05 03:01:00 INFO FileInputDStream: Cleared 1 old files that w
18/03/05 03:01:00 INFO MapPartitionsRDD: Removing RDD 21 from persi
18/03/05 03:01:00 INFO BlockManager: Removing RDD 12
18/03/05 03:01:00 INFO BlockManager: Removing RDD 21
18/03/05 03:01:00 INFO ShuffledRDD: Removing RDD 20 from persistence list
18/03/05 03:01:00 INFO BlockManager: Removing RDD 20
18/03/05 03:01:00 INFO MapPartitionsRDD: Removing RDD 19 from persistence list
18/03/05 03:01:00 INFO BlockManager: Removing RDD 19
```

```
root@localhost:/home/acadgild
File Edit View Search Terminal Help
Assignment-13-2-code.txt~ Documents Mu
Assignment-14-1-code Downloads NC
Assignment-14-1-code.txt eclipse Pi
Assignment-14-1-code.txt~ EvenLines.scala pr
[root@localhost acadgild]# gedit assingment-20-2-scala-code.t
[root@localhost acadgild]#
[root@localhost acadgild]#
[root@localhost acadgild]# nc -lk 9999
he is a dumb guy
she is an idiot
he is a stupid guy
he is an idiot
she is dumb
he is also stupid idiot
^C
[root@localhost acadgild]# nc -lk 9999
he is a dumb guy
she is dumb
he is also stupid idiot
he is a stupid guy
mudblood
^C
[root@localhost acadgild]# nc -lk 9999
he is a dumb guy
she is a stupid girl
he is an idiot
she is a dumb idiot
she don't know anything, she is dumb
he is a stupid guy
mudblood
```

The screenshot shows a dual-terminal window. The left terminal, titled 'acadgild@localhost:~/IdeaProjects/sampleProject/target/scala-2.10', displays Spark logs. The right terminal, titled 'root@localhost:/home/acadgild', shows a user input stream. In the left terminal, a yellow box highlights the text '(mudblood,1)' and '(stupid,1)'. In the right terminal, a yellow box highlights the text 'she don't know anything, she is dumb' and 'mudblood'.

```
18/03/05 03:02:00 INFO BlockManagerInfo: Added broadcast_18_piece0
18/03/05 03:02:00 INFO SparkContext: Created broadcast 18 from broad
18/03/05 03:02:00 INFO DAGScheduler: Submitting 1 missing tasks fro
iveWordCount.scala:55)
18/03/05 03:02:00 INFO TaskSchedulerImpl: Adding task set 24.0 with
18/03/05 03:02:00 INFO TaskSetManager: Starting task 0.0 in stage 2
)
18/03/05 03:02:00 INFO Executor: Running task 0.0 in stage 24.0 (TI
18/03/05 03:02:00 INFO ShuffleBlockFetcherIterator: Getting 1 non-e
18/03/05 03:02:00 INFO ShuffleBlockFetcherIterator: Started 0 remot
18/03/05 03:02:00 INFO Executor: Finished task 0.0 in stage 24.0 (T
18/03/05 03:02:00 INFO TaskSetManager: Finished task 0.0 in stage 2
18/03/05 03:02:00 INFO TaskSchedulerImpl: Removed TaskSet 24.0, whc
18/03/05 03:02:00 INFO DAGScheduler: ResultStage 24 (print at Offe
18/03/05 03:02:00 INFO DAGScheduler: Job 12 finished: print at Offe
-----
Time: 1520199120000 ms
-----
(mudblood,1)
(stupid,1)
18/03/05 03:02:00 INFO JobScheduler: Finished job streaming job 152
18/03/05 03:02:00 INFO JobScheduler: Total delay: 0.562 s for time
18/03/05 03:02:00 INFO ShuffledRDD: Removing RDD 26 from persisten
18/03/05 03:02:00 INFO BlockManager: Removing RDD 26
18/03/05 03:02:00 INFO MapPartitionsRDD: Removing RDD 25 from persi
18/03/05 03:02:00 INFO BlockManager: Removing RDD 25
18/03/05 03:02:00 INFO MapPartitionsRDD: Removing RDD 24 from persi
18/03/05 03:02:00 INFO BlockManager: Removing RDD 24
18/03/05 03:02:00 INFO MapPartitionsRDD: Removing RDD 23 from pers
18/03/05 03:02:00 INFO BlockManager: Removing RDD 23
18/03/05 03:02:00 INFO UnionRDD: Removing RDD 22 from persistence l
18/03/05 03:02:00 INFO BlockManager: Removing RDD 22
18/03/05 03:02:00 INFO FileInputDStream: Cleared 1 old files that were older than 1520199060000 ms: 1520199000000 ms
18/03/05 03:02:00 INFO MapPartitionsRDD: Removing RDD 31 from persistence list
18/03/05 03:02:00 INFO BlockManager: Removing RDD 31
18/03/05 03:02:00 INFO ShuffledRDD: Removing RDD 30 from persistence list
18/03/05 03:02:00 INFO BlockManager: Removing RDD 30
```

Now we can see **the count of each abusive word** entered by a user on the side terminal for a batch of lines every 60 seconds. This is how we can create two different streams and process the data in parallel using **Spark Streaming APIs**.