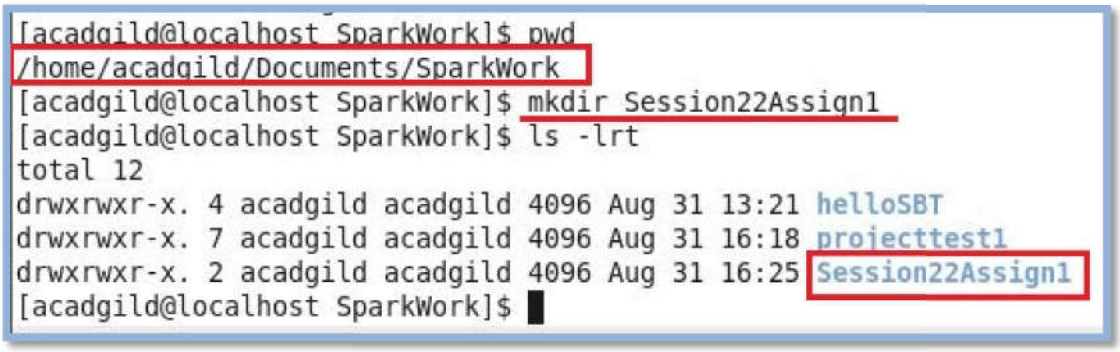
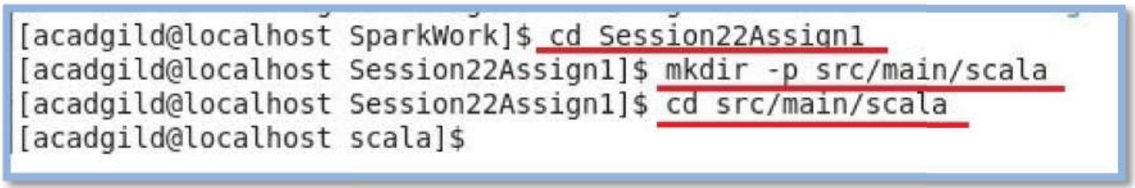
Spark Streaming using TCP Socket

Step 1: Create Project Directory "Session22Assign1" inside /home/acadgild/Documents/SparkWork/

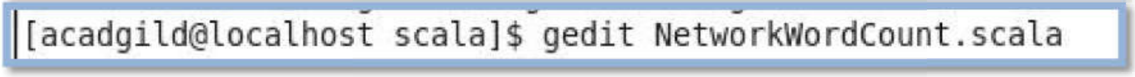


```
[acadgild@localhost SparkWork]$ pwd
/home/acadgild/Documents/SparkWork
[acadgild@localhost SparkWork]$ mkdir Session22Assign1
[acadgild@localhost SparkWork]$ ls -lrt
total 12
drwxrwxr-x. 4 acadgild acadgild 4096 Aug 31 13:21 helloSBT
drwxrwxr-x. 7 acadgild acadgild 4096 Aug 31 16:18 projecttest1
drwxrwxr-x. 2 acadgild acadgild 4096 Aug 31 16:25 Session22Assign1
[acadgild@localhost SparkWork]$ 
```

Step 2: Change to project directory "Session22Assign1" and create "src/main/scala" directory structure inside it, after that, change to "scala" directory and create "NetworkWordCount.scala" file inside it which contains code to be run:

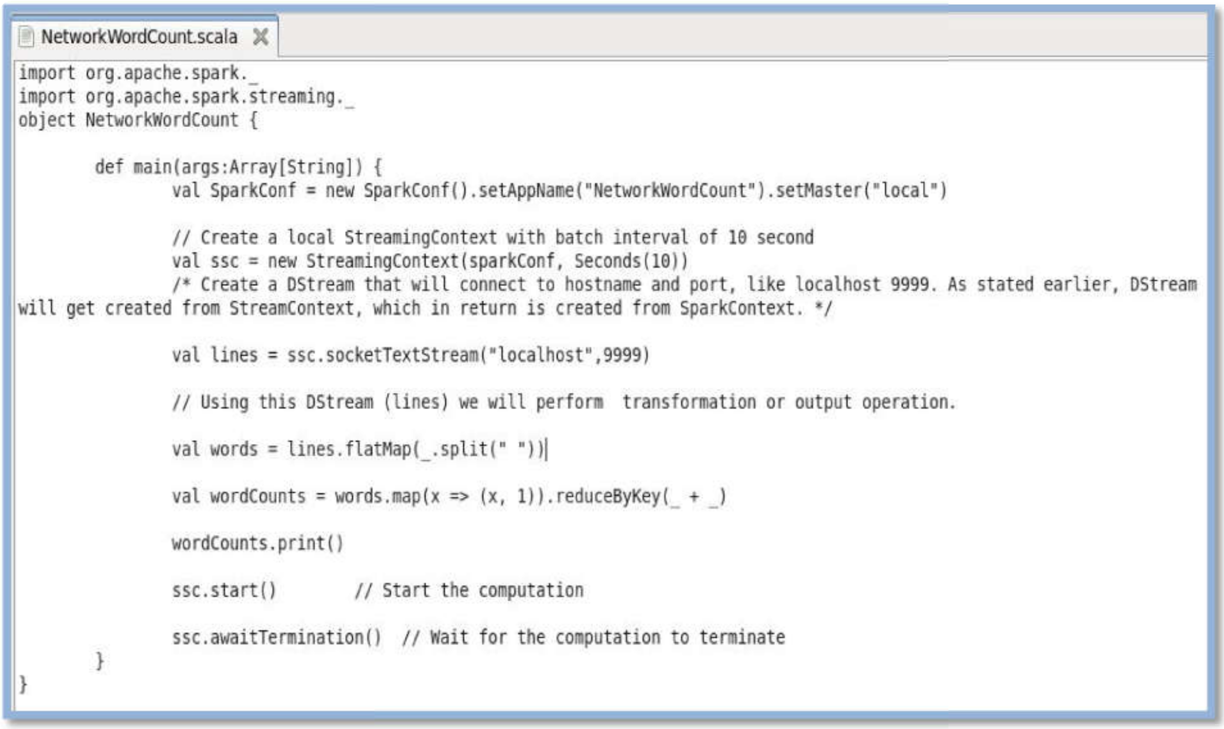


```
[acadgild@localhost SparkWork]$ cd Session22Assign1
[acadgild@localhost Session22Assign1]$ mkdir -p src/main/scala
[acadgild@localhost Session22Assign1]$ cd src/main/scala
[acadgild@localhost scala]$
```



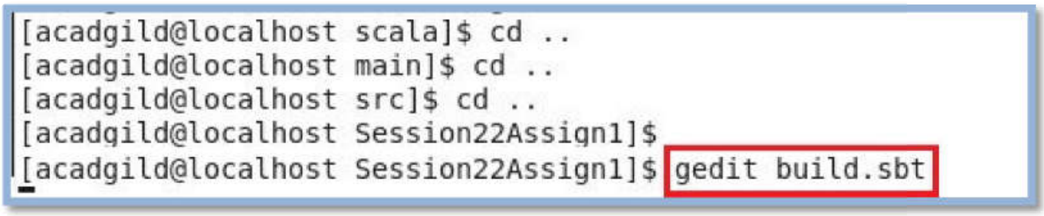
```
[acadgild@localhost scala]$ gedit NetworkWordCount.scala
```

Step 3: Write below contents inside "NetworkWordCount.scala" file



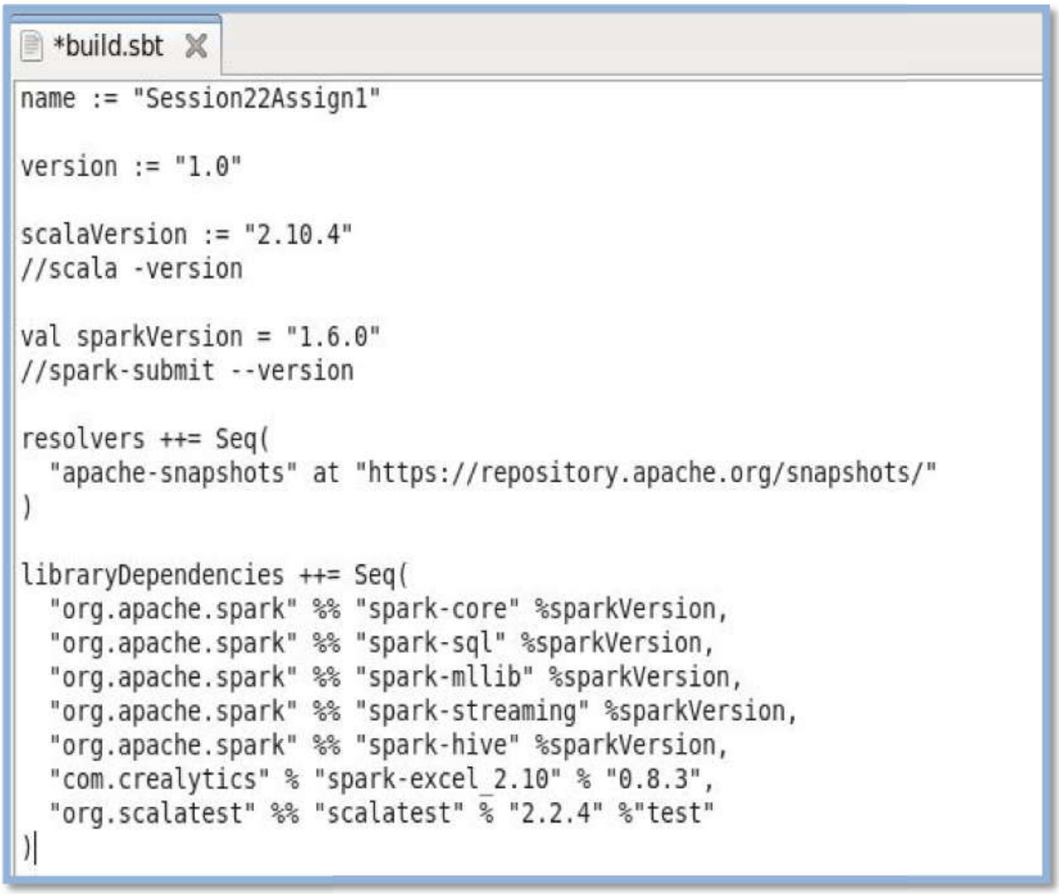
```
import org.apache.spark._
import org.apache.spark.streaming._
object NetworkWordCount {

        def main(args:Array[String]) {
                val SparkConf = new SparkConf().setAppName("NetworkWordCount").setMaster("local")

                // 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.

                val words = lines.flatMap(_.split(" "))

                val wordCounts = words.map(x => (x, 1)).reduceByKey(_ + _)

                wordCounts.print()

                ssc.start()        // Start the computation

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

Step 5: Now change to main project directory "Session22Assign1" to create build.sbt file inside it



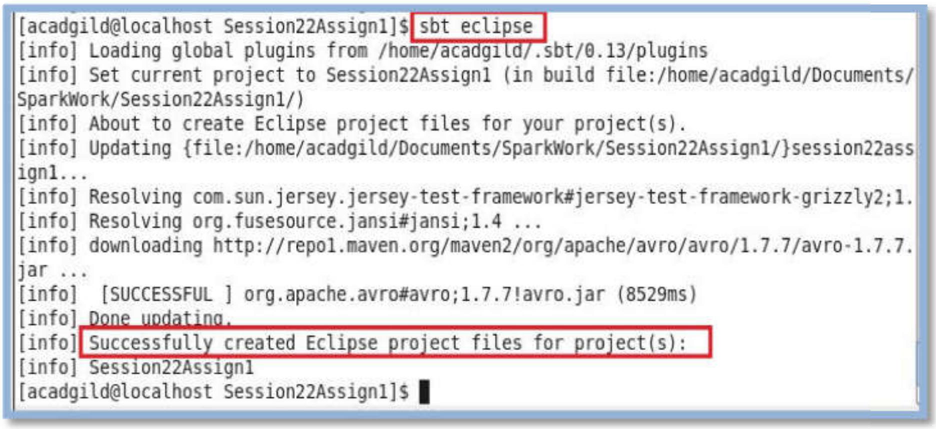
```
[acadgild@localhost scala]$ cd ..
[acadgild@localhost main]$ cd ..
[acadgild@localhost src]$ cd ..
[acadgild@localhost Session22Assign1]$
[acadgild@localhost Session22Assign1]$ gedit build.sbt
```

Write following contents inside build.sbt



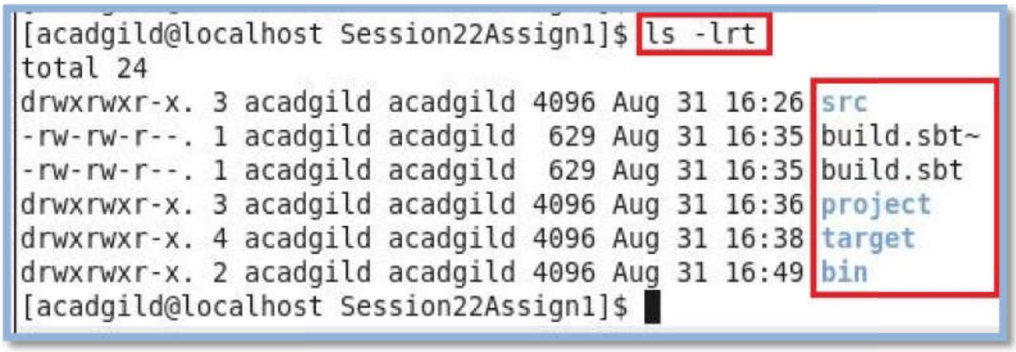
```
name := "Session22Assign1"

version := "1.0"

scalaVersion := "2.10.4"
//scala -version

val sparkVersion = "1.6.0"
//spark-submit --version

resolvers ++= Seq(
  "apache-snapshots" at "https://repository.apache.org/snapshots/"
)

libraryDependencies ++= Seq(
  "org.apache.spark" %% "spark-core" %sparkVersion,
  "org.apache.spark" %% "spark-sql" %sparkVersion,
  "org.apache.spark" %% "spark-mllib" %sparkVersion,
  "org.apache.spark" %% "spark-streaming" %sparkVersion,
  "org.apache.spark" %% "spark-hive" %sparkVersion,
  "com.crealytics" % "spark-excel_2.10" % "0.8.3",
  "org.scalatest" %% "scalatest" % "2.2.4" %"test"
)
```

Step 6: Run "sbt eclipse" command inside "Session22Assign1" project directory



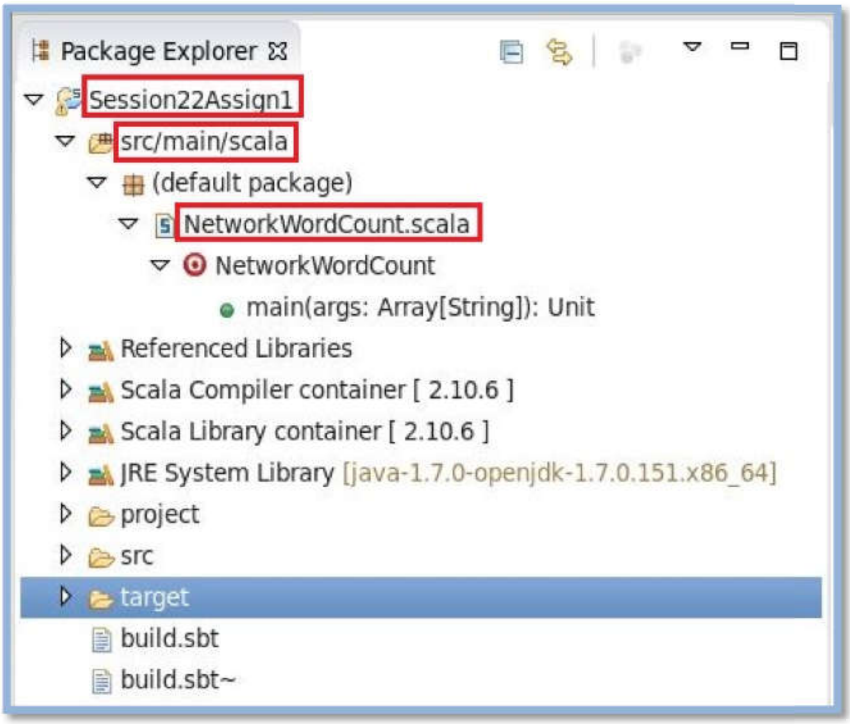
```
[acadgild@localhost Session22Assign1]$ sbt eclipse
[info] Loading global plugins from /home/acadgild/.sbt/0.13/plugins
[info] Set current project to Session22Assign1 (in build file:/home/acadgild/Documents/
SparkWork/Session22Assign1/)
[info] About to create Eclipse project files for your project(s).
[info] Updating {file:/home/acadgild/Documents/SparkWork/Session22Assign1/}session22ass
ign1...
[info] Resolving com.sun.jersey.jersey-test-framework#jersey-test-framework-grizzly2;1.
[info] Resolving org.fusesource.jansi#jansi;1.4 ...
[info] downloading http://repo1.maven.org/maven2/org/apache/avro/avro/1.7.7/avro-1.7.7.
jar ...
[info]  [SUCCESSFUL ] org.apache.avro#avro;1.7.7!avro.jar (8529ms)
[info] Done updating.
[info] Successfully created Eclipse project files for project(s):
[info] Session22Assign1
[acadgild@localhost Session22Assign1]$
```

Step 7: We can see the following directory structure inside "Session22Assign1" project directory after firing "sbt eclipse" command



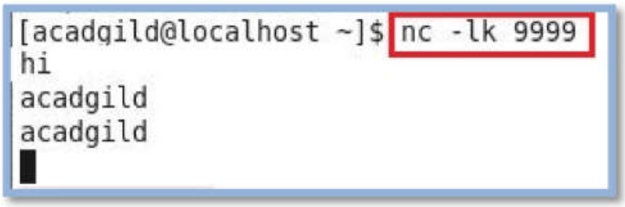
```
[acadgild@localhost Session22Assign1]$ ls -lrt
total 24
drwxrwxr-x. 3 acadgild acadgild 4096 Aug 31 16:26 src
-rw-rw-r--. 1 acadgild acadgild  629 Aug 31 16:35 build.sbt~
-rw-rw-r--. 1 acadgild acadgild  629 Aug 31 16:35 build.sbt
drwxrwxr-x. 3 acadgild acadgild 4096 Aug 31 16:36 project
drwxrwxr-x. 4 acadgild acadgild 4096 Aug 31 16:38 target
drwxrwxr-x. 2 acadgild acadgild 4096 Aug 31 16:49 bin
[acadgild@localhost Session22Assign1]$
```

Step 8: Now import the project inside eclipse, after import, following directory structure will be shown in eclipse



```
Package Explorer
Session22Assign1
  src/main/scala
    (default package)
      NetworkWordCount.scala
        NetworkWordCount
          main(args: Array[String]): Unit
  Referenced Libraries
  Scala Compiler container [ 2.10.6 ]
  Scala Library container [ 2.10.6 ]
  JRE System Library [java-1.7.0-openjdk-1.7.0.151.x86_64]
  project
  src
  target
  build.sbt
  build.sbt~
```

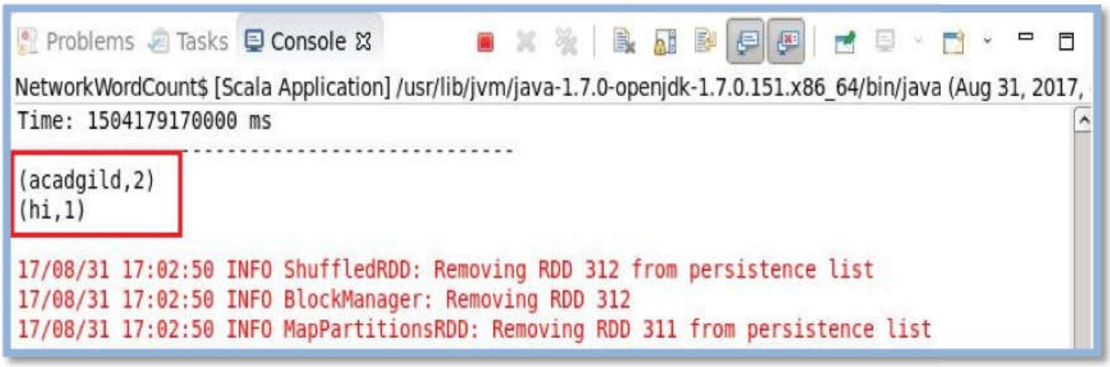
Step 9: Open new terminal and type "nc –lk 9999" command to run "netcat" as a data server, after that, type few words



```
[acadgild@localhost ~]$ nc -lk 9999
hi
acadgild
acadgild
```

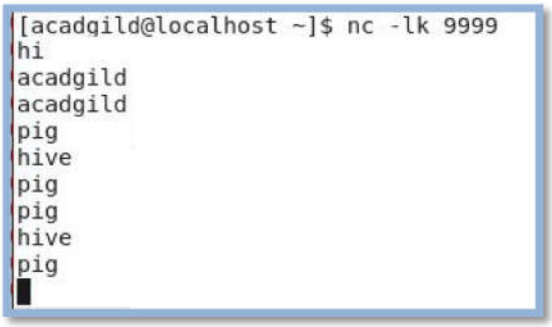
This terminal acts as a server where words are fed continuously, and our Spark Streaming code counts the number of occurrences (in a batch interval of 10 sec).

Step 10: In eclipse, where project is imported, we can check the output in console,



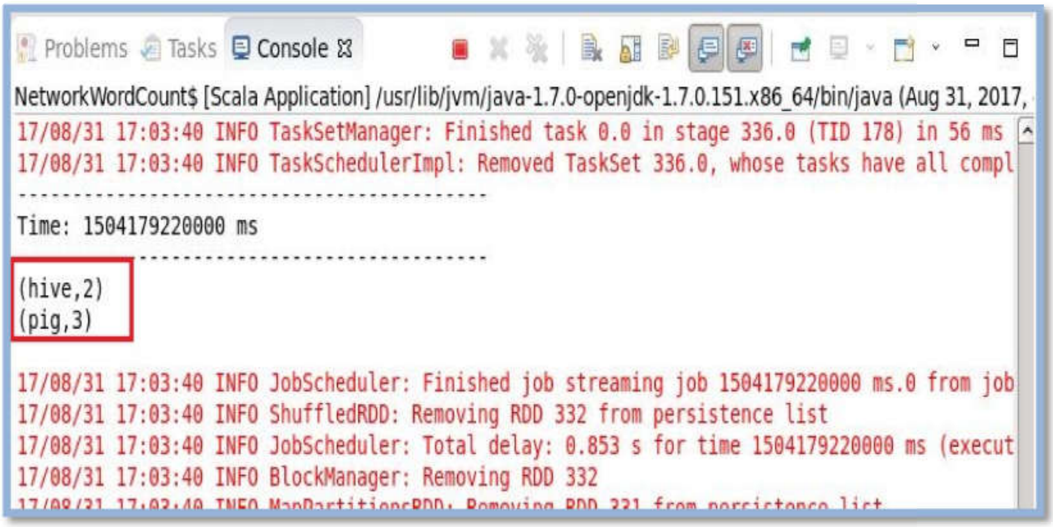
```
NetworkWordCount$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.151.x86_64/bin/java (Aug 31, 2017,
Time: 1504179170000 ms
-------------------------------------------
(acadgild,2)
(hi,1)

17/08/31 17:02:50 INFO ShuffledRDD: Removing RDD 312 from persistence list
17/08/31 17:02:50 INFO BlockManager: Removing RDD 312
17/08/31 17:02:50 INFO MapPartitionsRDD: Removing RDD 311 from persistence list
```

Step 11: Again, type some words in data server terminal



```
[acadgild@localhost ~]$ nc -lk 9999
hi
acadgild
acadgild
pig
hive
pig
pig
hive
pig
```

Step 12: In eclipse, check the output in console,



```
NetworkWordCount$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.151.x86_64/bin/java (Aug 31, 2017,
17/08/31 17:03:40 INFO TaskSetManager: Finished task 0.0 in stage 336.0 (TID 178) in 56 ms
17/08/31 17:03:40 INFO TaskSchedulerImpl: Removed TaskSet 336.0, whose tasks have all compl
-------------------------------------------
Time: 1504179220000 ms
-------------------------------------------
(hive,2)
(pig,3)

17/08/31 17:03:40 INFO JobScheduler: Finished job streaming job 1504179220000 ms.0 from job
17/08/31 17:03:40 INFO ShuffledRDD: Removing RDD 332 from persistence list
17/08/31 17:03:40 INFO JobScheduler: Total delay: 0.853 s for time 1504179220000 ms (execut
17/08/31 17:03:40 INFO BlockManager: Removing RDD 332
17/08/31 17:03:40 INFO MapPartitionsRDD: Removing RDD 331 from persistence list
```

As the interval has been set at 10 sec, that's why output is captured like above.