BigData Assignment 8.1

Spark Streaming using TCP Socket

A demo of Spark Streaming from a TCP socket. In this, we will perform the task of counting words in text data received from a data server listening on a TCP socket.

Solution -

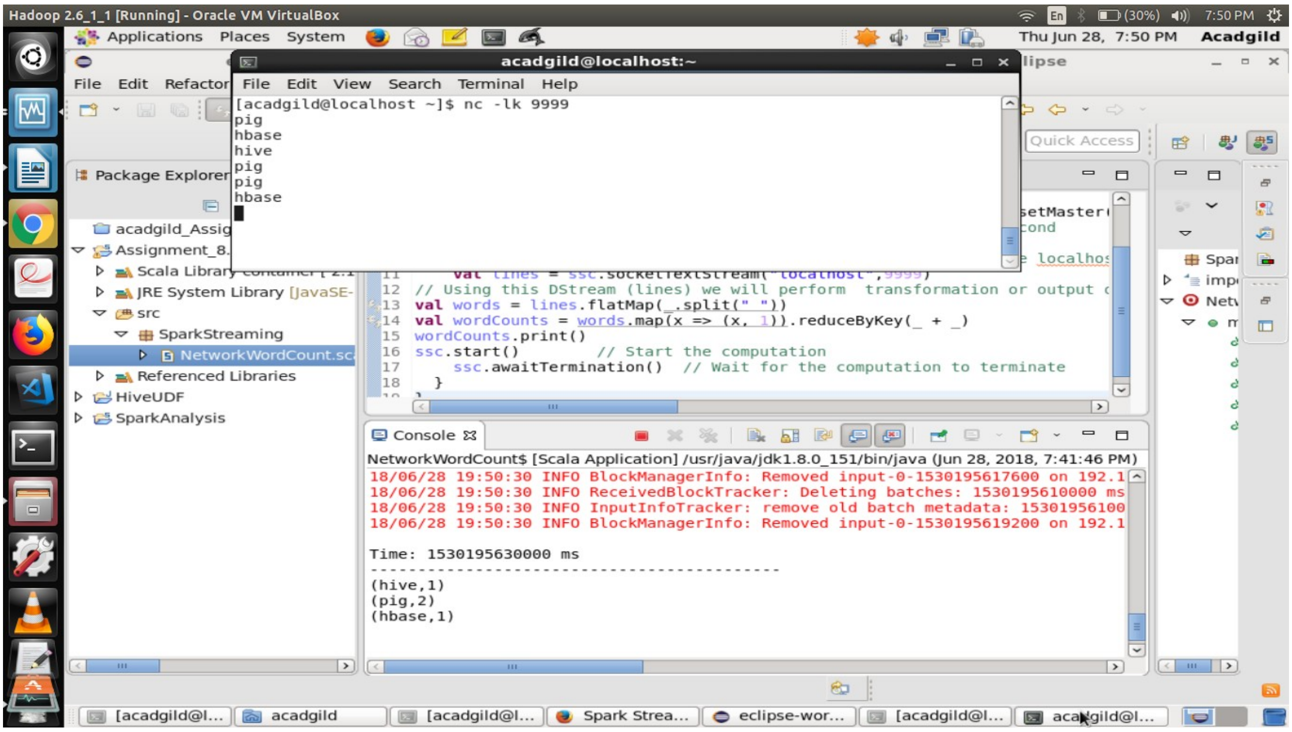
<u>NetworkWordCount.scala</u>

```
package SparkStreaming

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

object NetworkWordCount {
        def main(args:Array[String]) {
                val SparkConf = new
SparkConf().setAppName("NetworkWordCount").setMaster("local[2
]")
                // 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
```

```
        }
}
```

- Parallely in another terminal type “nc –lk 9999” command to run “netcat” as a data server, after that, typed few words

  nc -lk 9999
  pig
  hbase
  hive
  pig
  pig
  pig
  hbase



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

```
    val lines = ssc.socketTextStream("localhost",9999)
// Using this DStream (lines) we will perform  transformation or output c
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
  }
```

```
NetworkWordCount$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 28, 2018, 7:41:46 PM)
18/06/28 19:50:30 INFO BlockManagerInfo: Removed input-0-1530195617600 on 192.1
18/06/28 19:50:30 INFO ReceivedBlockTracker: Deleting batches: 1530195610000 ms
18/06/28 19:50:30 INFO InputInfoTracker: remove old batch metadata: 15301956100
18/06/28 19:50:30 INFO BlockManagerInfo: Removed input-0-1530195619200 on 192.1

Time: 1530195630000 ms
-------------------------------------------
(hive,1)
(pig,2)
(hbase,1)
```

The code was runed in eclipse and in the console it can be seen that it shows the wordcount after every 10s.

Thats why hbase count is 1 , pig count is 2 and hive count is 1. In 10s , it captured these much words.