CASE STUDY 5

Spark Streaming

Assignment 27: Case Study Spark Streaming

Problem Statement

- There are two parts this case study :

- **- First Part :**

  You have to create a Spark Application which streams data from a file on local directory on your machine and does the word count on the fly. The word should be done by the spark application in such a way that as soon as you drop the file in your local directory, your spark application should immediately do the word count for you.

- **- Second Part** :

  In this part, you will have to create a Spark Application which should do the following :

1. Pick up a file from the local directory and do the word count

2. Then in the same Spark Application, write the code to put the same file on HDFS.

3. Then in same Spark Application, do the word count of the file copied on HDFS in step 2

4. Lastly, compare the word count of step 1 and 2. Both should match, other throw an error

Solution:

Initial Execution:

```
[acadgild@localhost ~]$ jps
2964 Jps
[acadgild@localhost ~]$ sudo service sshd start
[sudo] password for acadgild:
[acadgild@localhost ~]$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
18/09/07 21:05:23 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
```

```
Starting namenodes on [localhost]
localhost: starting namenode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-namenode-localhost.localdomain.
out
localhost: starting datanode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-datanode-localhost.localdomain.o
ut
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/hadoop-acadgild-secondarynamenode-localhost.loc
aldomain.out
18/09/07 21:05:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-resourcemanager-localhost.localdom
ain.out
localhost: starting nodemanager, logging to
/home/acadgild/install/hadoop/hadoop-2.6.5/logs/yarn-acadgild-nodemanager-localhost.localdomain.
out
[acadgild@localhost ~]$ jps
3680 Jps
3545 ResourceManager
3386 SecondaryNameNode
3115 NameNode
3212 DataNode
3646 NodeManager
[acadgild@localhost ~]$
```

- **- First Part :**

  You have to create a Spark Application which streams data from a file on local directory on your machine and does the word count on the fly. The word should be done by the spark application in such a way that as soon as you drop the file in your local directory, your spark application should immediately do the word count for you.

Initial Terminal Execution:

```
[acadgild@localhost Local_Streaming_Test]$ vi test1.txt


Hi i love hadoop bigdata
Hadoop is fun
spark is fast
spark is fun
Compare Now
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test1.txt" 5L, 78C                                              5,1         All
:wq
```

```
[acadgild@localhost Local_Streaming_Test]$ vi test2.txt

it is second file
hadoop spark streaming is fun
I love hadoop
Bigdata is fun
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test2.txt" 6L, 79C
:wq
```

Program Solution :

```
package com.acadgild.casestudy;

import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.log4j.{Level,Logger}


object SparkFileStreamingWordCount {

  def main(args: Array[String]): Unit = {
    println("hey Spark Streaming")

    val conf = new
SparkConf().setMaster("local[2]").setAppName("SparkSteamingExample")
    conf.set("spark.testing.memory", "2147480000")
    val sc = new SparkContext(conf)
val rootLogger =Logger.getRootLogger()
   rootLogger.setLevel(Level.ERROR)
    val ssc = new StreamingContext(sc, Seconds(15))
    val lines =
ssc.textFileStream("file:///home/acadgild/Desktop/Local_Streaming_Test")
    val words = lines.flatMap(_.split(" "))
    val wordCounts = words.map(x => (x, 1)).reduceByKey(_ + _)
    wordCounts.print()
    ssc.start()
    ssc.awaitTermination()


  }

}
```

Console Output :

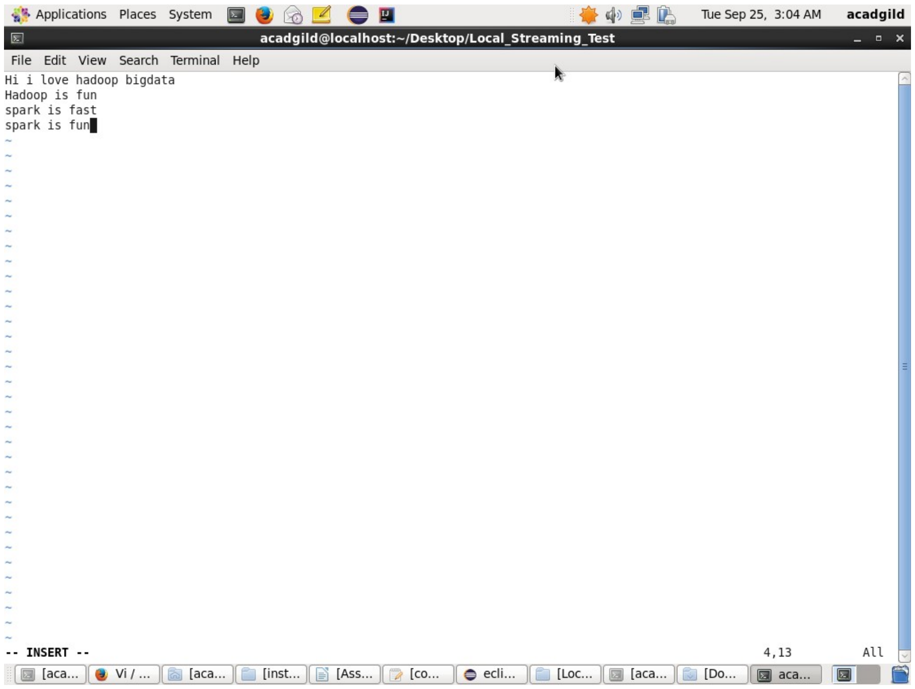
```
hey Spark Streaming
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/09/25 03:03:48 INFO SparkContext: Running Spark version 2.1.0
18/09/25 03:03:55 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
18/09/25 03:04:00 WARN Utils: Your hostname, localhost.localdomain resolves to a
loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface eth1)
18/09/25 03:04:00 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
18/09/25 03:04:01 INFO SecurityManager: Changing view acls to: acadgild
18/09/25 03:04:01 INFO SecurityManager: Changing modify acls to: acadgild
18/09/25 03:04:01 INFO SecurityManager: Changing view acls groups to:
18/09/25 03:04:01 INFO SecurityManager: Changing modify acls groups to:
18/09/25 03:04:01 INFO SecurityManager: SecurityManager: authentication disabled;
ui acls disabled; users  with view permissions: Set(acadgild); groups with view
permissions: Set(); users  with modify permissions: Set(acadgild); groups with
```

```
modify permissions: Set()
18/09/25 03:04:14 INFO Utils: Successfully started service 'sparkDriver' on port
33033.
18/09/25 03:04:15 INFO SparkEnv: Registering MapOutputTracker
18/09/25 03:04:16 INFO SparkEnv: Registering BlockManagerMaster
18/09/25 03:04:17 INFO BlockManagerMasterEndpoint: Using
org.apache.spark.storage.DefaultTopologyMapper for getting topology information
18/09/25 03:04:17 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
18/09/25 03:04:18 INFO DiskBlockManager: Created local directory at
/tmp/blockmgr-e9656f66-7ecd-48c3-a00c-5207e2c27a6a
18/09/25 03:04:19 INFO MemoryStore: MemoryStore started with capacity 1048.8 MB
18/09/25 03:04:21 INFO SparkEnv: Registering OutputCommitCoordinator
18/09/25 03:04:31 WARN Utils: Service 'SparkUI' could not bind on port 4040.
Attempting port 4041.
18/09/25 03:04:32 INFO Utils: Successfully started service 'SparkUI' on port
4041.
18/09/25 03:04:32 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at
http://10.0.2.15:4041
18/09/25 03:04:36 INFO Executor: Starting executor ID driver on host localhost
18/09/25 03:04:37 INFO Utils: Successfully started service
'org.apache.spark.network.netty.NettyBlockTransferService' on port 44047.
18/09/25 03:04:37 INFO NettyBlockTransferService: Server created on
10.0.2.15:44047
18/09/25 03:04:37 INFO BlockManager: Using
org.apache.spark.storage.RandomBlockReplicationPolicy for block replication
policy
18/09/25 03:04:37 INFO BlockManagerMaster: Registering BlockManager
BlockManagerId(driver, 10.0.2.15, 44047, None)
18/09/25 03:04:37 INFO BlockManagerMasterEndpoint: Registering block manager
10.0.2.15:44047 with 1048.8 MB RAM, BlockManagerId(driver, 10.0.2.15, 44047,
None)
18/09/25 03:04:37 INFO BlockManagerMaster: Registered BlockManager
BlockManagerId(driver, 10.0.2.15, 44047, None)
18/09/25 03:04:37 INFO BlockManager: Initialized BlockManager:
BlockManagerId(driver, 10.0.2.15, 44047, None)
-------------------------------------------
Time: 1537824900000 ms
-------------------------------------------

-------------------------------------------
Time: 1537824915000 ms
-------------------------------------------

-------------------------------------------
Time: 1537824930000 ms
-------------------------------------------

-------------------------------------------
Time: 1537824945000 ms
-------------------------------------------
(is,3)
(fast,1)
(love,1)
(Now,1)
(bigdata,1)
(spark,2)
(hadoop,1)
(i,1)
```

```
(Compare,1)
(fun,2)
...

-------------------------------------------
Time: 1537824960000 ms
-------------------------------------------

-------------------------------------------
Time: 1537824975000 ms
-------------------------------------------

-------------------------------------------
Time: 1537824990000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825005000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825020000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825035000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825050000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825065000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825080000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825095000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825110000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825125000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825140000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825155000 ms
```

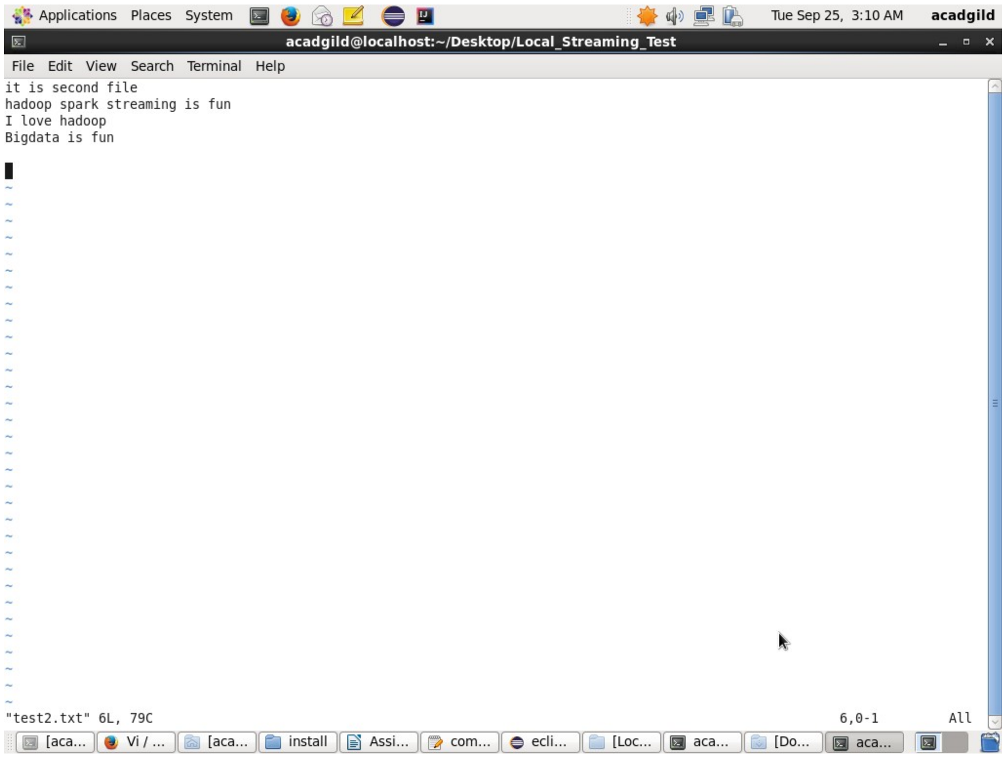
```
-------------------------------------------

-------------------------------------------
Time: 1537825170000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825185000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825200000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825215000 ms
-------------------------------------------
(is,3)
(second,1)
(love,1)
(,2)
(streaming,1)
(Bigdata,1)
(file,1)
(it,1)
(spark,1)
(hadoop,2)
...

-------------------------------------------
Time: 1537825230000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825245000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825260000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825275000 ms
-------------------------------------------

-------------------------------------------
Time: 1537825290000 ms
-------------------------------------------
```

ScreenShots OUTPUTS



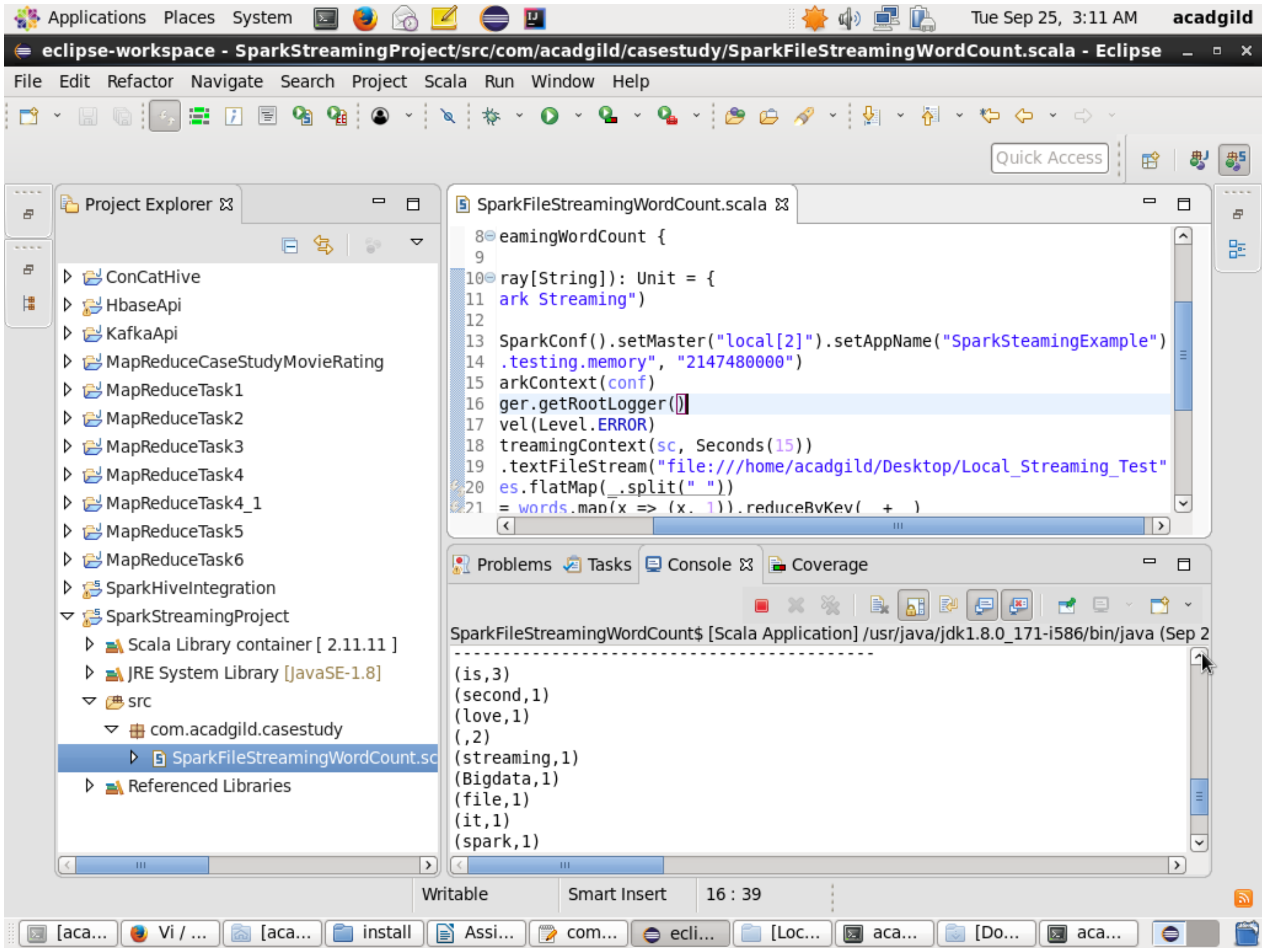
Applications Places System
Tue Sep 25, 3:04 AM acadgild
acadgild@localhost:~/Desktop/Local_Streaming_Test
File Edit View Search Terminal Help

```
Hi i love hadoop bigdata
Hadoop is fun
spark is fast
spark is fun
~
-- INSERT --                                              4,13          All
```



Applications Places System
Tue Sep 25, 3:10 AM acadgild
acadgild@localhost:~/Desktop/Local_Streaming_Test
File Edit View Search Terminal Help

```
it is second file
hadoop spark streaming is fun
I love hadoop
Bigdata is fun

~
"test2.txt" 6L, 79C                                        6,0-1         All
```


Applications Places System
Tue Sep 25, 3:11 AM acadgild
eclipse-workspace - SparkStreamingProject/src/com/acadgild/casestudy/SparkFileStreamingWordCount.scala - Eclipse
File Edit Refactor Navigate Search Project Scala Run Window Help
Quick Access
Project Explorer
ConCatHive
HbaseApi
KafkaApi
MapReduceCaseStudyMovieRating
MapReduceTask1
MapReduceTask2
MapReduceTask3
MapReduceTask4
MapReduceTask4_1
MapReduceTask5
MapReduceTask6
SparkHiveIntegration
SparkStreamingProject
Scala Library container [2.11.11]
JRE System Library [JavaSE-1.8]
src
com.acadgild.casestudy
SparkFileStreamingWordCount.sc
Referenced Libraries
SparkFileStreamingWordCount.scala
Problems Tasks Console Coverage
Writable Smart Insert 16 : 39

```
eamingWordCount {

ray[String]): Unit = {
ark Streaming")

SparkConf().setMaster("local[2]").setAppName("SparkSteamingExample")
.testing.memory", "2147480000")
arkContext(conf)
ger.getRootLogger()
vel(Level.ERROR)
treamingContext(sc, Seconds(15))
.textFileStream("file:///home/acadgild/Desktop/Local_Streaming_Test"
es.flatMap(_.split(" "))
= words.map(x => (x, 1)).reduceByKey(_ + _)
```

```
SparkFileStreamingWordCount$ [Scala Application] /usr/java/jdk1.8.0_171-i586/bin/java (Sep 2
-------------------------------------------
(is,3)
(second,1)
(love,1)
(,2)
(streaming,1)
(Bigdata,1)
(file,1)
(it,1)
(spark,1)
```

- Second Part :

In this part, you will have to create a Spark Application which should do the following :

1. Pick up a file from the local directory and do the word count

2. Then in the same Spark Application, write the code to put the same file on HDFS.

3. Then in same Spark Application, do the word count of the file copied on HDFS in step 2

4. Lastly, compare the word count of step 1 and 2. Both should match, other throw an error

Initial Terminal Execution:

```
[acadgild@localhost ~]$ jps

3744 NodeManager

5410 SparkSubmit

3635 ResourceManager

3259 DataNode

3163 NameNode

7022 Jps

5022

[acadgild@localhost ~]$ hdfs dfs -ls /

18/09/25 03:36:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable

Found 8 items

drwxr-xr-x   - acadgild supergroup          0 2018-07-04 22:39 /SQOOPOUT

drwxr-xr-x   - acadgild supergroup          0 2018-07-04 23:13 /SQOOPOUT1

drwxr-xr-x   - acadgild supergroup          0 2018-09-07 21:06 /hadoopdata

drwxr-xr-x   - acadgild supergroup          0 2018-09-24 03:24 /hbase

drwxr-xr-x   - acadgild supergroup          0 2018-07-16 07:56 /home
```

```
drwxr-xr-x   - acadgild supergroup          0 2018-07-04 09:16 /sqoopout

drwx-wx-wx   - acadgild supergroup          0 2018-07-11 00:19 /tmp

drwxr-xr-x   - acadgild supergroup          0 2018-07-15 22:56 /user

[acadgild@localhost ~]$ hdfs dfs -ls /user

18/09/25 03:36:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable

Found 2 items

drwxr-xr-x   - acadgild supergroup          0 2018-08-02 05:30 /user/acadgild

drwxr-xr-x   - acadgild supergroup          0 2018-07-15 22:56 /user/hive

[acadgild@localhost ~]$ hdfs dfs -mkdir /user/streaming

18/09/25 03:37:32 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable

[acadgild@localhost ~]$ hdfs dfs -ls /user

18/09/25 03:37:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable

Found 3 items

drwxr-xr-x   - acadgild supergroup          0 2018-08-02 05:30 /user/acadgild

drwxr-xr-x   - acadgild supergroup          0 2018-07-15 22:56 /user/hive

drwxr-xr-x   - acadgild supergroup          0 2018-09-25 03:37 /user/streaming

[acadgild@localhost ~]$
```

```
[acadgild@localhost HDFS_Streaming_Test]$ vi test1.txt

i i love hadoop bigdata
Hadoop is fun
spark is fast
spark is fun
We are working on Spark Streaming
We will test the word count program on local and hdfs file
We will learn hadoop and spark
Compare Now
~
~
~
~
~
```

```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                    1,1           All
```

:wq

Program Solution :

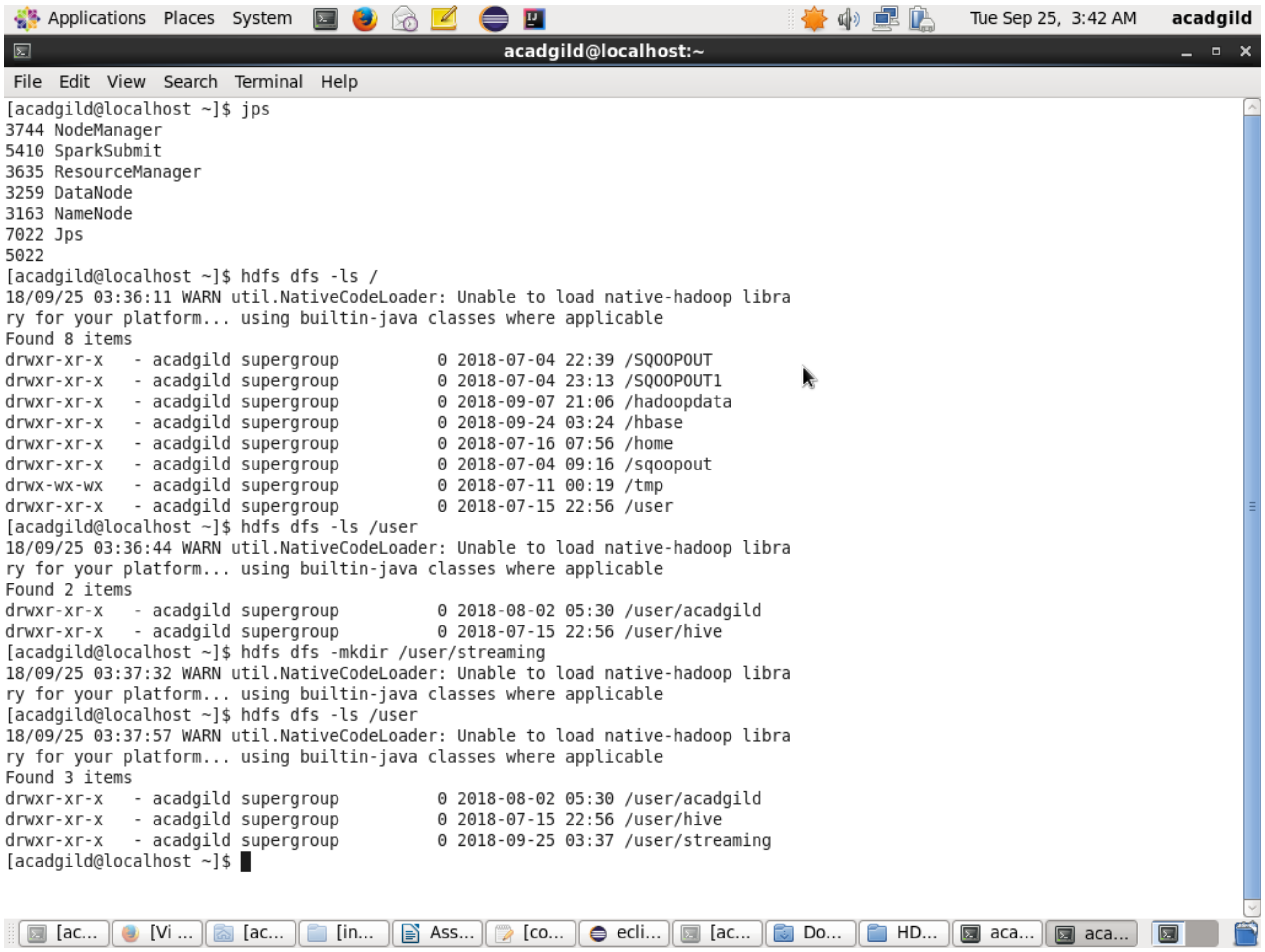
```
package com.acadgild.casestudy;

import java.io.File
import org.apache.spark.{SparkConf, SparkContext}
import scala.io.Source._
import org.apache.log4j.{Level,Logger}



object SparkHDFSWordCountComparison {

  private var localFilePath: File = new
File("/home/acadgild/Desktop/HDFS_Streaming_Test/test1.txt")
  private var dfsDirPath: String = "hdfs://localhost:8020/user/streaming"
  private val NPARAMS = 2
  def main(args: Array[String]): Unit = {

    println("SparkHDFSWordCountComparison : Main Called Successfully")

    println("Performing local word count")
    val fileContents = readFile(localFilePath.toString())

    println("Performing local word count - File Content ->>"+fileContents)
    val localWordCount = runLocalWordCount(fileContents)

    println("SparkHDFSWordCountComparison : Main Called Successfully -> Local
Word Count is ->>"+localWordCount)

    println("Performing local word count Completed !!")

    println("Creating Spark Context")

    val conf = new
SparkConf().setMaster("local[2]").setAppName("SparkHDFSWordCountComparisonApp")
    conf.set("spark.testing.memory", "2147480000")
    val sc = new SparkContext(conf)
val rootLogger =Logger.getRootLogger()
   rootLogger.setLevel(Level.ERROR)


    println("Spark Context Created")

    println("Writing local file to DFS")
    val dfsFilename = dfsDirPath + "/dfs_read_write_test"
    val fileRDD = sc.parallelize(fileContents)
    fileRDD.saveAsTextFile(dfsFilename)
    println("Writing local file to DFS Completed")

    println("Reading file from DFS and running Word Count")
    val readFileRDD = sc.textFile(dfsFilename)

    val dfsWordCount = readFileRDD
      .flatMap(_.split(" "))
      .flatMap(_.split("\t"))
      .filter(_.nonEmpty)
      .map(w => (w, 1))
      .countByKey()
      .values
      .sum

    sc.stop()

    if (localWordCount == dfsWordCount) {
      println(s"Success! Local Word Count ($localWordCount) " +
        s"and DFS Word Count ($dfsWordCount) agree.")
    } else {
      println(s"Failure! Local Word Count ($localWordCount) " +
        s"and DFS Word Count ($dfsWordCount) disagree.")
    }



  }
  private def printUsage(): Unit = {
    val usage: String = "DFS Read-Write Test\n" +
      "\n" +
      "Usage: localFile dfsDir\n" +
      "\n" +
      "localFile - (string) local file to use in test\n" +
      "dfsDir - (string) DFS directory for read/write tests\n"

    println(usage)
  }

  private def readFile(filename: String): List[String] = {
    val lineIter: Iterator[String] = fromFile(filename).getLines()
    val lineList: List[String] = lineIter.toList
    lineList
  }

  def runLocalWordCount(fileContents: List[String]): Int = {
    fileContents.flatMap(_.split(" "))
      .flatMap(_.split("\t"))
      .filter(_.nonEmpty)
      .groupBy(w => w)
      .mapValues(_.size)
      .values
      .sum
  }


}
```

Console Output :

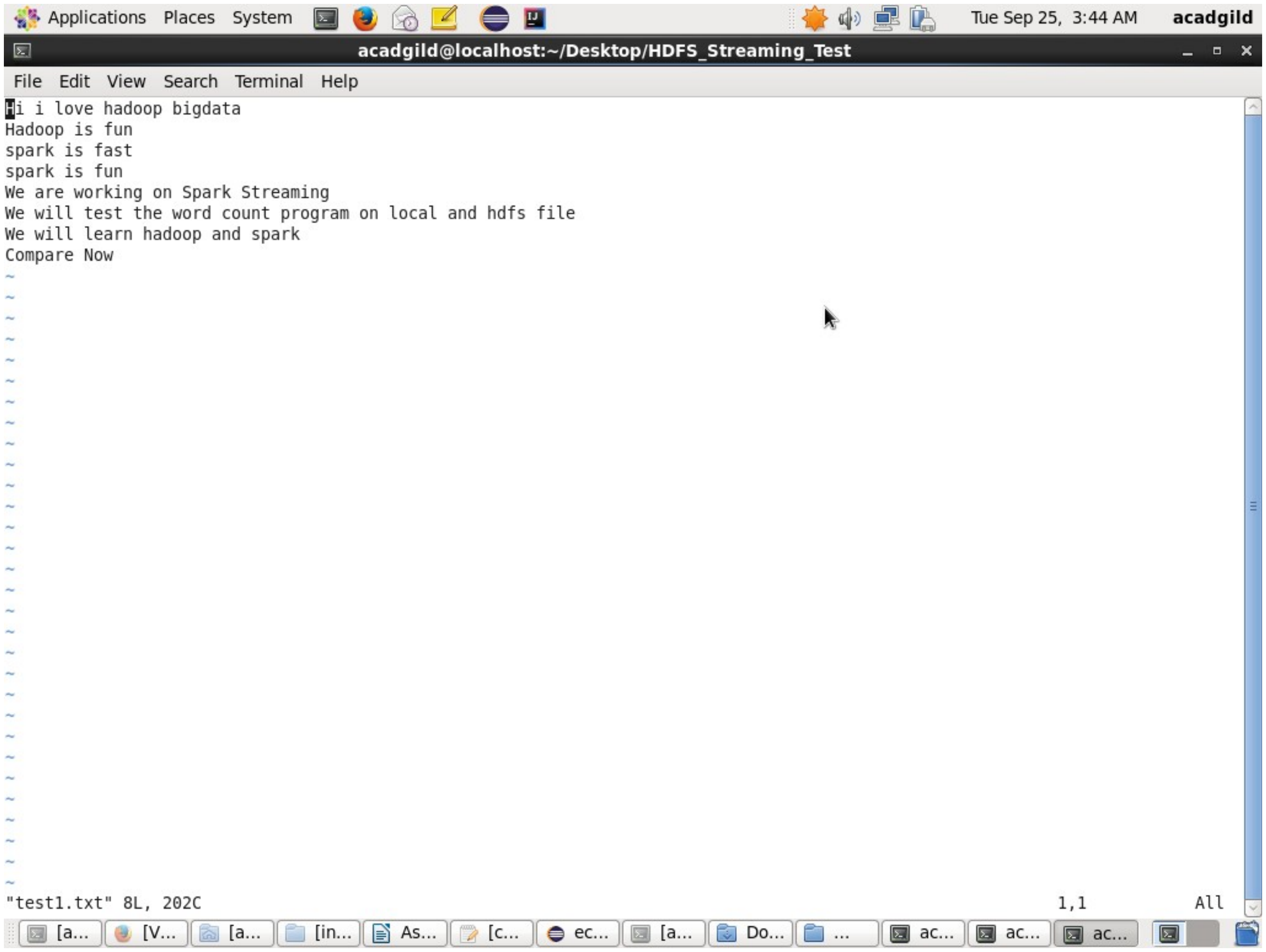
```
SparkHDFSWordCountComparison : Main Called Successfully
Performing local word count
Performing local word count - File Content ->>List(Hi i love hadoop bigdata,
Hadoop is fun, spark is fast, spark is fun, We are working on Spark Streaming, We
will test the word count program on local and hdfs file, We will learn hadoop and
spark, Compare Now)
SparkHDFSWordCountComparison : Main Called Successfully -> Local Word Count is
->>40
Performing local word count Completed !!
Creating Spark Context
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
18/09/25 03:56:45 INFO SparkContext: Running Spark version 2.1.0
18/09/25 03:56:47 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
18/09/25 03:56:49 WARN Utils: Your hostname, localhost.localdomain resolves to a
loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface eth1)
18/09/25 03:56:49 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
18/09/25 03:56:50 INFO SecurityManager: Changing view acls to: acadgild
18/09/25 03:56:50 INFO SecurityManager: Changing modify acls to: acadgild
18/09/25 03:56:50 INFO SecurityManager: Changing view acls groups to:
```

```
18/09/25 03:56:50 INFO SecurityManager: Changing modify acls groups to: 
18/09/25 03:56:50 INFO SecurityManager: SecurityManager: authentication disabled;
ui acls disabled; users  with view permissions: Set(acadgild); groups with view
permissions: Set(); users  with modify permissions: Set(acadgild); groups with
modify permissions: Set()
18/09/25 03:56:53 INFO Utils: Successfully started service 'sparkDriver' on port
35175.
18/09/25 03:56:53 INFO SparkEnv: Registering MapOutputTracker
18/09/25 03:56:53 INFO SparkEnv: Registering BlockManagerMaster
18/09/25 03:56:53 INFO BlockManagerMasterEndpoint: Using
org.apache.spark.storage.DefaultTopologyMapper for getting topology information
18/09/25 03:56:53 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
18/09/25 03:56:54 INFO DiskBlockManager: Created local directory at
/tmp/blockmgr-9bd45519-4f71-4e99-a286-d8fcf7b28654
18/09/25 03:56:54 INFO MemoryStore: MemoryStore started with capacity 1048.8 MB
18/09/25 03:56:54 INFO SparkEnv: Registering OutputCommitCoordinator
18/09/25 03:56:58 WARN Utils: Service 'SparkUI' could not bind on port 4040.
Attempting port 4041.
18/09/25 03:56:58 INFO Utils: Successfully started service 'SparkUI' on port
4041.
18/09/25 03:56:58 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at
http://10.0.2.15:4041
18/09/25 03:57:00 INFO Executor: Starting executor ID driver on host localhost
18/09/25 03:57:00 INFO Utils: Successfully started service
'org.apache.spark.network.netty.NettyBlockTransferService' on port 44998.
18/09/25 03:57:00 INFO NettyBlockTransferService: Server created on
10.0.2.15:44998
18/09/25 03:57:01 INFO BlockManager: Using
org.apache.spark.storage.RandomBlockReplicationPolicy for block replication
policy
18/09/25 03:57:01 INFO BlockManagerMaster: Registering BlockManager
BlockManagerId(driver, 10.0.2.15, 44998, None)
18/09/25 03:57:01 INFO BlockManagerMasterEndpoint: Registering block manager
10.0.2.15:44998 with 1048.8 MB RAM, BlockManagerId(driver, 10.0.2.15, 44998,
None)
18/09/25 03:57:01 INFO BlockManagerMaster: Registered BlockManager
BlockManagerId(driver, 10.0.2.15, 44998, None)
18/09/25 03:57:01 INFO BlockManager: Initialized BlockManager:
BlockManagerId(driver, 10.0.2.15, 44998, None)
Spark Context Created
Writing local file to DFS
Writing local file to DFS Completed
Reading file from DFS and running Word Count
Success! Local Word Count (40) and DFS Word Count (40) agree.
```

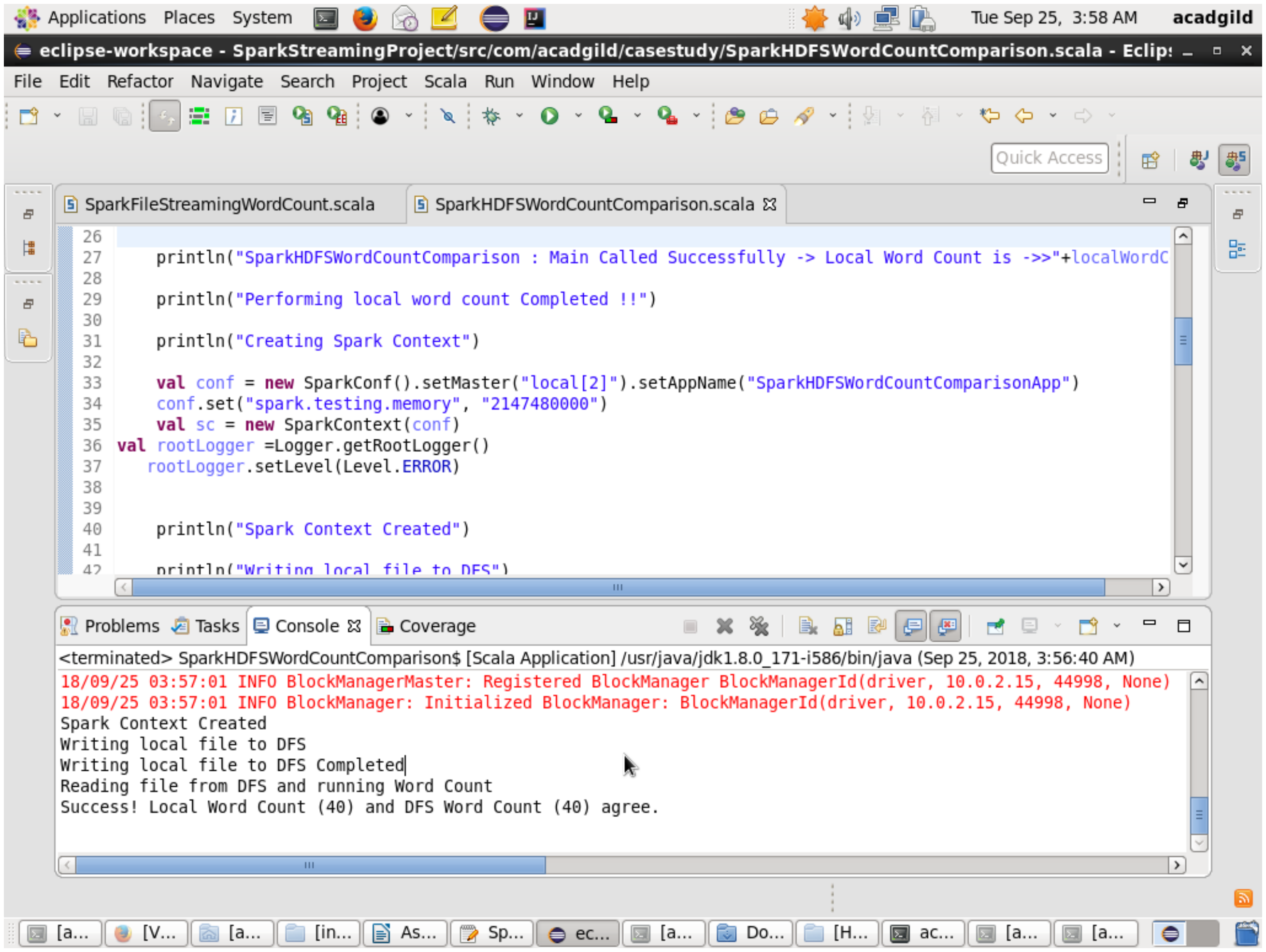
ScreenShots OUTPUTS


Applications Places System
Tue Sep 25, 3:42 AM acadgild
acadgild@localhost:~
File Edit View Search Terminal Help

```
[acadgild@localhost ~]$ jps
3744 NodeManager
5410 SparkSubmit
3635 ResourceManager
3259 DataNode
3163 NameNode
7022 Jps
5022
[acadgild@localhost ~]$ hdfs dfs -ls /
18/09/25 03:36:11 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 8 items
drwxr-xr-x   - acadgild supergroup          0 2018-07-04 22:39 /SQOOPOUT
drwxr-xr-x   - acadgild supergroup          0 2018-07-04 23:13 /SQOOPOUT1
drwxr-xr-x   - acadgild supergroup          0 2018-09-07 21:06 /hadoopdata
drwxr-xr-x   - acadgild supergroup          0 2018-09-24 03:24 /hbase
drwxr-xr-x   - acadgild supergroup          0 2018-07-16 07:56 /home
drwxr-xr-x   - acadgild supergroup          0 2018-07-04 09:16 /sqoopout
drwx-wx-wx   - acadgild supergroup          0 2018-07-11 00:19 /tmp
drwxr-xr-x   - acadgild supergroup          0 2018-07-15 22:56 /user
[acadgild@localhost ~]$ hdfs dfs -ls /user
18/09/25 03:36:44 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x   - acadgild supergroup          0 2018-08-02 05:30 /user/acadgild
drwxr-xr-x   - acadgild supergroup          0 2018-07-15 22:56 /user/hive
[acadgild@localhost ~]$ hdfs dfs -mkdir /user/streaming
18/09/25 03:37:32 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hdfs dfs -ls /user
18/09/25 03:37:57 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x   - acadgild supergroup          0 2018-08-02 05:30 /user/acadgild
drwxr-xr-x   - acadgild supergroup          0 2018-07-15 22:56 /user/hive
drwxr-xr-x   - acadgild supergroup          0 2018-09-25 03:37 /user/streaming
[acadgild@localhost ~]$
```


Applications Places System
Tue Sep 25, 3:44 AM acadgild
acadgild@localhost:~/Desktop/HDFS_Streaming_Test
File Edit View Search Terminal Help

```
Hi i love hadoop bigdata
Hadoop is fun
spark is fast
spark is fun
We are working on Spark Streaming
We will test the word count program on local and hdfs file
We will learn hadoop and spark
Compare Now
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test1.txt" 8L, 202C                                                          1,1           All
```


Applications Places System
Tue Sep 25, 3:58 AM acadgild
eclipse-workspace - SparkStreamingProject/src/com/acadgild/casestudy/SparkHDFSWordCountComparison.scala - Eclip
File Edit Refactor Navigate Search Project Scala Run Window Help
Quick Access
SparkFileStreamingWordCount.scala
SparkHDFSWordCountComparison.scala
Problems Tasks Console Coverage

```

    println("SparkHDFSWordCountComparison : Main Called Successfully -> Local Word Count is ->>"+localWordC

    println("Performing local word count Completed !!")

    println("Creating Spark Context")

    val conf = new SparkConf().setMaster("local[2]").setAppName("SparkHDFSWordCountComparisonApp")
    conf.set("spark.testing.memory", "2147480000")
    val sc = new SparkContext(conf)
val rootLogger =Logger.getRootLogger()
   rootLogger.setLevel(Level.ERROR)


    println("Spark Context Created")

    println("Writing local file to DFS")
```

```
<terminated> SparkHDFSWordCountComparison$ [Scala Application] /usr/java/jdk1.8.0_171-i586/bin/java (Sep 25, 2018, 3:56:40 AM)
18/09/25 03:57:01 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, 10.0.2.15, 44998, None)
18/09/25 03:57:01 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 10.0.2.15, 44998, None)
Spark Context Created
Writing local file to DFS
Writing local file to DFS Completed
Reading file from DFS and running Word Count
Success! Local Word Count (40) and DFS Word Count (40) agree.
```