Case Study V

Spark Streaming

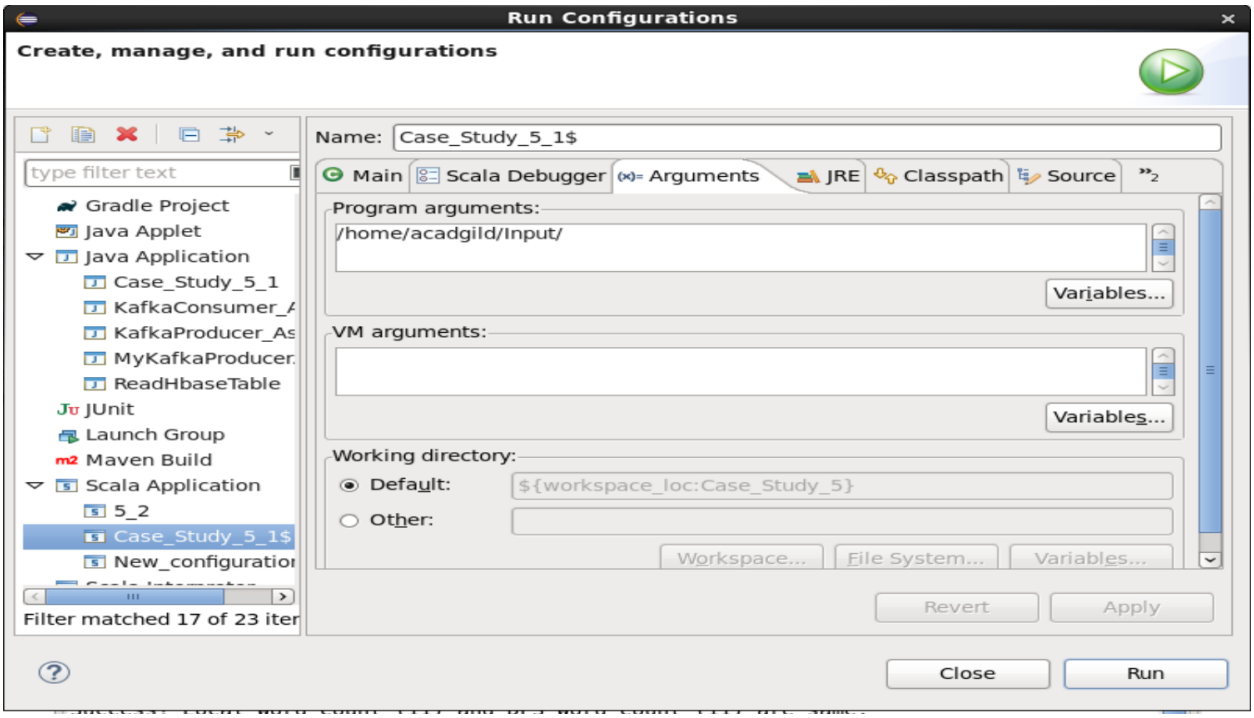
- Prachi Mohite

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.

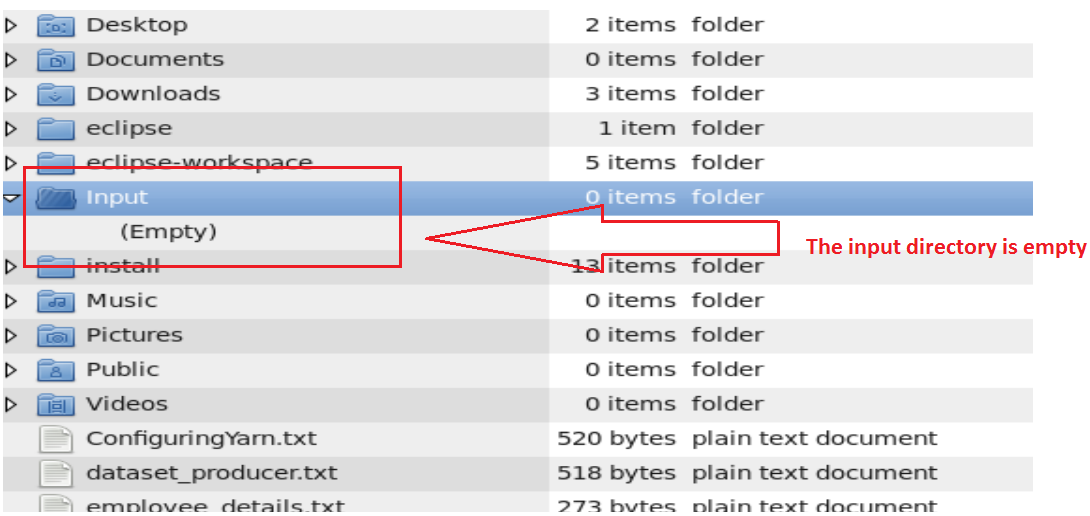
Complete Code

```
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark.{SparkConf, SparkContext}

object Case_Study_5_1 {
  def main(args: Array[String]): Unit = {
    println("hey Scala")
    if(args.length!=1)
    {
      System.err.println("Enter the name of file");
      System.exit(1)
    }
    val count = args.length
    println(count)

    //Created spark context
    val conf = new
SparkConf().setMaster("local[*]")setAppName("Case_Study_5");
    val sc = new SparkContext(conf);
    println("Spark conext created")
    //Set log level
    sc.setLogLevel("WARN")

    //creating the streaming context
    val ssc = new StreamingContext(sc,Seconds(15))

    //Read the file (which is given as input from command line
arguments)
    val filePath = args(0).toString()
    val lines = ssc.textFileStream(filePath)

    //println(lines.count())
    val wordCount = lines.flatMap(_.split(" 
")).map(x=>(x,1)).reduceByKey(_+_)
    wordCount.print()
    ssc.start()
    ssc.awaitTermination()
```

```
    }
}
```



Part 1.1 Output when Directory is empty



Case_Study_5_2.scala | Problems | Tasks | Case_Study_5_1.scala

```
8     println("hey Scala")
9     if (args.length != 2) {
10      System.err.println("Usage Case_Study_5_2<LocalDirectory> <HDFSDirector
11      System.exit(1)
12    }
13
14    val localFilePath = args(0).toString()
15    val dfsDirPath = args(1).toString()
```

Console

```
Case_Study_5_1$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 22, 2018, 11:17:45 AM)
Spark conext created
/home/acadgild/Input/
-------------------------------------------
Time: 1529646495000 ms
-------------------------------------------

-------------------------------------------
Time: 1529646510000 ms
-------------------------------------------

-------------------------------------------
Time: 1529646525000 ms
-------------------------------------------
```

Output Part 1.2 When added a file to directory

Desktop	2 items	folde
Documents	0 items	folde
Downloads	3 items	folde
eclipse	1 item	folde
eclipse-workspace	5 items	folde
Input	1 item	folde
test5.txt	51 bytes	plain
install	13 items	folde
Music	0 items	folde
Pictures	0 items	folde
Public	0 items	folde
Videos	0 items	folde
ConfiguringYarn.txt	520 bytes	plain
dataset_producer.txt	518 bytes	plain
employee_details.txt	273 bytes	plain

```
Console
Case_Study_5_1$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 22, 2018, 11:17:45 AM)
-------------------------------------------
Time: 1529646780000 ms
-------------------------------------------
(is,1)
(another,1)
(through,1)
(data,1)
(This,2)
(testing,1)
(for,2)
(assignment,1)
(streaming,1)
(file,2)
```

```
[acadgild@localhost Input]$ cat test5.txt
This is another file for testing data streaming through file for This assignment
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Input]$
```

Added some more files to test streaming.

```
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ cd Input
[acadgild@localhost Input]$ nano test1.txt
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Input]$ nano test2.txt
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Input]$ cat test1.txt
New file for word count need to create new as well
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Input]$ cat test2.txt
Second file , for demo of second time
I am liking this Second demo for second time

[acadgild@localhost Input]$
```

First file Streamed

Second file streamed

```
-------------------------------------------
Time: 1528977840000 ms
-------------------------------------------
(this,1)
(second,2)
(am,1)
(I,1)
(demo,2)
(,1)
(of,1)
(Second,2)
(for,2)
(time,2)
...

-------------------------------------------
Time: 1528977855000 ms
-------------------------------------------
```

```
-------------------------------------------
Time: 1528977750000 ms
-------------------------------------------
(New,1)
(need,1)
(to,1)
(as,1)
(word,1)
(new,1)
(for,1)
(count,1)
(create,1)
(file,1)
...

-------------------------------------------
Time: 1528977765000 ms
-------------------------------------------
```

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

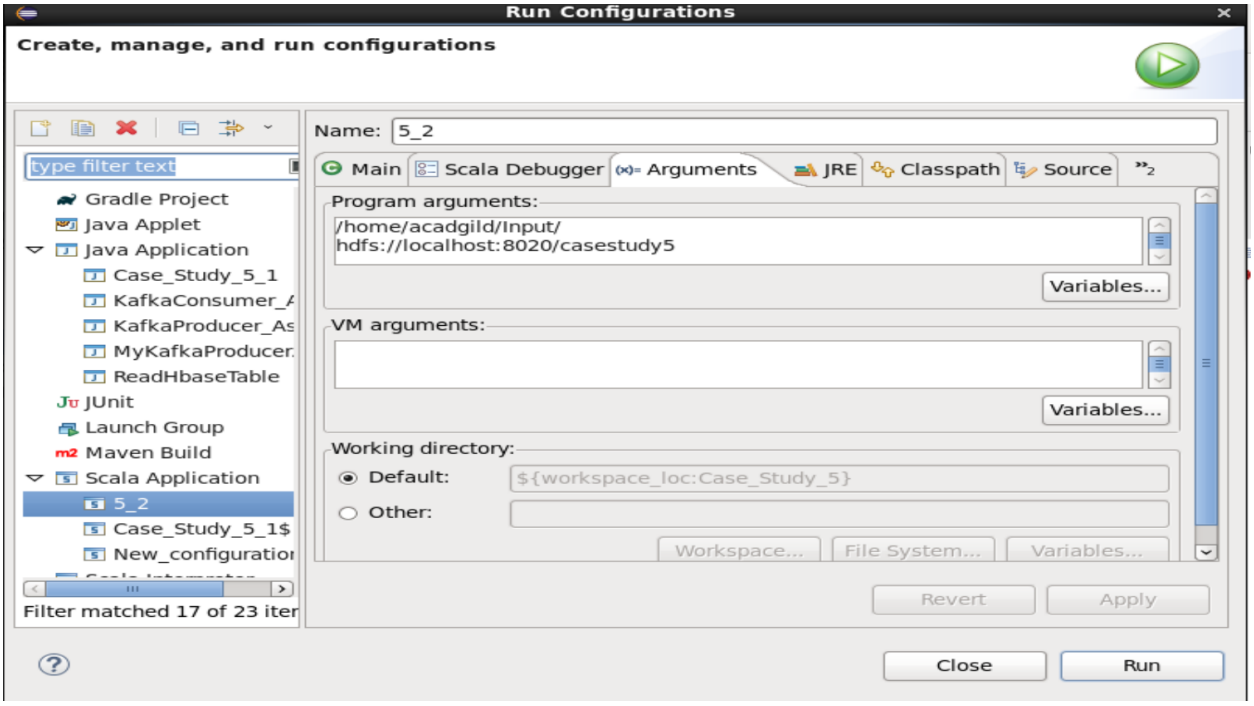
Complete Code

```
import org.apache.spark.{SparkConf, SparkContext}

import org.apache.log4j.{Level, Logger}
import scala.io.Source._

object Case_Study_5_2 {
  def main(args: Array[String]): Unit = {
    println("hey Scala")
```

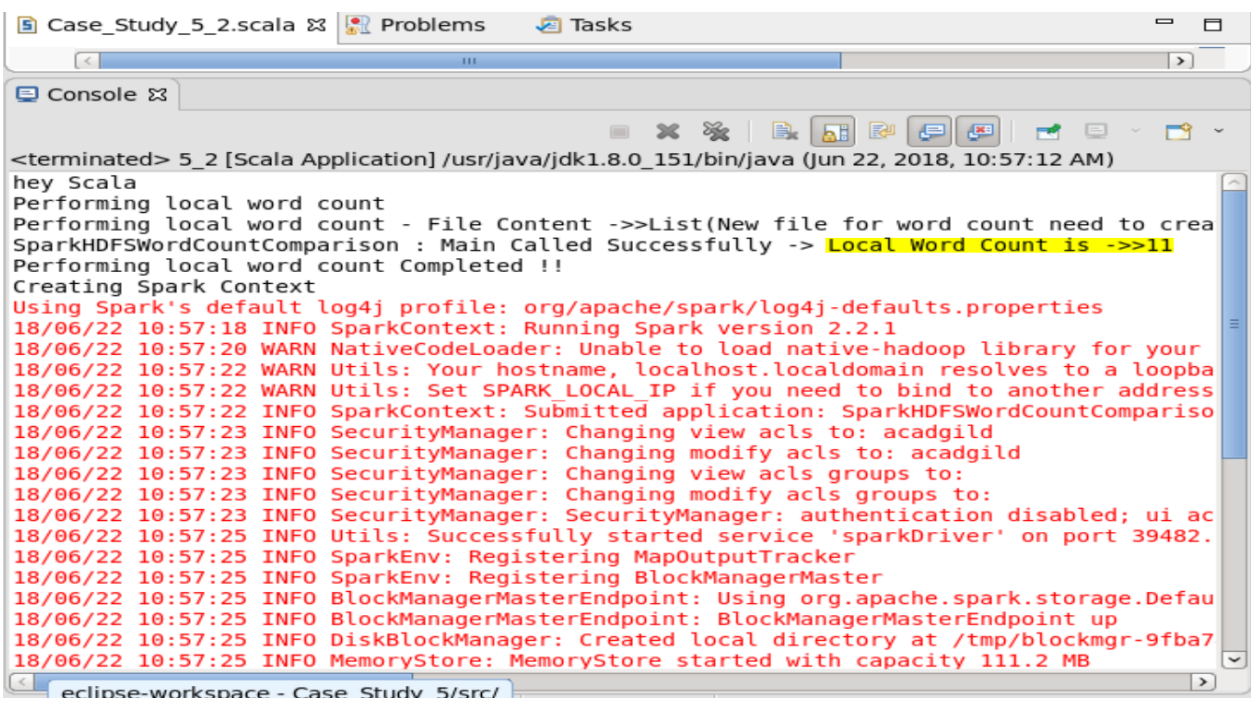
```
    if (args.length != 2) {
      System.err.println("Usage Case_Study_5_2<LocalDirectory>
<HDFSDirectory>");
      System.exit(1)
    }

    val localFilePath = args(0).toString()
    val dfsDirPath = args(0).toString()

    //println("HDFSWordCountComparison : Main Called Successfully")

    println("Performing local word count")
    val fileContents = readFile(localFilePath.toString()+"test1.txt")

    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("SparkHDFSWordCountCompar
isonApp")
    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 + "/local_to_hdfs"
    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) are same.")
    } else {
      println(s"Failure! Local Word Count ($localWordCount) " +
        s"and DFS Word Count ($dfsWordCount) are not same.")
    }
  }

  private def RunLocalWordCount(fileContent:List[String]):Int={
    var wordCount = fileContent.flatMap(_.split(" 
")).filter(_.nonEmpty).groupBy(w=>w).mapValues(_.size).values.sum;
    return wordCount;
  }

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

}
```

Running Configurations



Word count from Local file placed at /home/acadgild/test1.txt



File written to HDFS and word count performed on HDFS.

```
Case_Study_5_2.scala   Problems   Tasks
Console
<terminated> 5_2 [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 22, 2018, 10:57:12 AM)
18/06/22 10:57:25 INFO Utils: Successfully started service 'sparkDriver' on port 39482.
18/06/22 10:57:25 INFO SparkEnv: Registering MapOutputTracker
18/06/22 10:57:25 INFO SparkEnv: Registering BlockManagerMaster
18/06/22 10:57:25 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.Defau
18/06/22 10:57:25 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
18/06/22 10:57:25 INFO DiskBlockManager: Created local directory at /tmp/blockmgr-9fba7
18/06/22 10:57:25 INFO MemoryStore: MemoryStore started with capacity 111.2 MB
18/06/22 10:57:26 INFO SparkEnv: Registering OutputCommitCoordinator
18/06/22 10:57:27 INFO Utils: Successfully started service 'SparkUI' on port 4040.
18/06/22 10:57:28 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://192.168
18/06/22 10:57:29 INFO Executor: Starting executor ID driver on host localhost
18/06/22 10:57:29 INFO Utils: Successfully started service 'org.apache.spark.network.ne
18/06/22 10:57:29 INFO NettyBlockTransferService: Server created on 192.168.0.6:43967
18/06/22 10:57:29 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicat
18/06/22 10:57:30 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driv
18/06/22 10:57:30 INFO BlockManagerMasterEndpoint: Registering block manager 192.168.0.
18/06/22 10:57:30 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(drive
18/06/22 10:57:30 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, 1
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 (11) and DFS Word Count (11) are same.
```

Word count after reading the file from HDFS is same as from local

Verifying the file has been written to HDFS.

```
[acadgild@localhost ~]$ hadoop fs -ls /casestudy5
18/06/22 11:14:58 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 2018-06-14 18:36 /casestudy5/local_toHDFS
drwxr-xr-x   - acadgild supergroup          0 2018-06-22 11:12 /casestudy5/local_to_hdfs
[acadgild@localhost ~]$ hadoop fs -ls /casestudy5/local_to_hdfs
18/06/22 11:15:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 3 items
-rw-r--r--   3 acadgild supergroup          0 2018-06-22 11:12 /casestudy5/local_to_hdfs/_SUCCESS
-rw-r--r--   3 acadgild supergroup          0 2018-06-22 11:12 /casestudy5/local_to_hdfs/part-00000
-rw-r--r--   3 acadgild supergroup         53 2018-06-22 11:12 /casestudy5/local_to_hdfs/part-00001
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
```

Folder created under casestudy5 and file is also written