

Case_Study_-_IV Spark_Streaming

PROBLEM STATEMENT –

1. 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.
2. In this part, you will have to create a Spark Application which should do the following:
 - Pick up a file from the local directory and do the word count.
 - Then in the same Spark Application, write the code to put the same file on HDFS.
 - Then in same Spark Application, do the word count of the file copied on HDFS in step 2.
 - Lastly, compare the word count of step 1 and 2. Both should match, other throw an error.

SOLUTION –

PROBLEM STATEMENT 1:

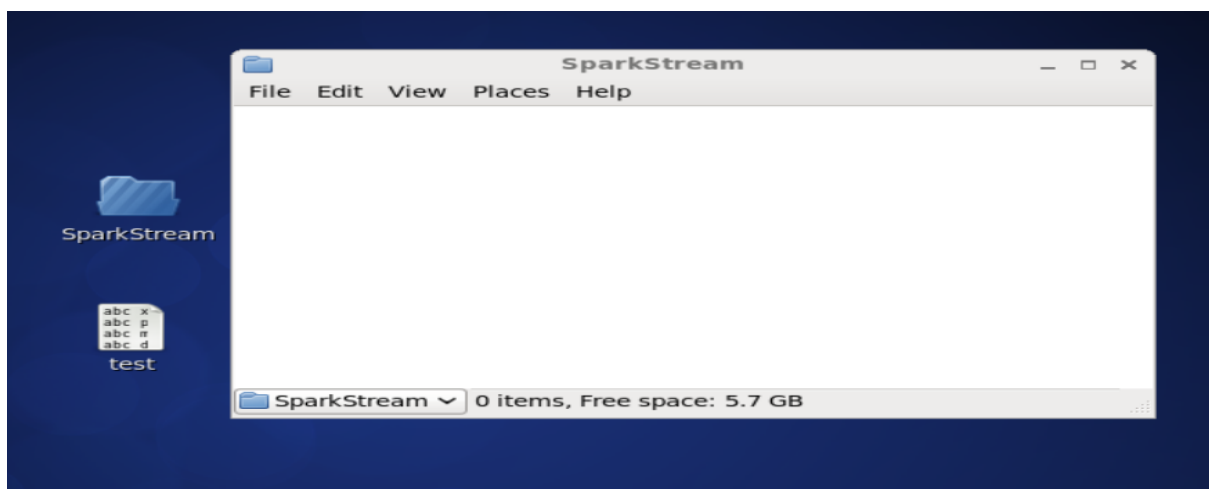
CODE -

```
1 import org.apache.spark.{SparkConf, SparkContext}
2
3 object SparkFileStreamingWordCount {
4
5     def main(args: Array[String]): Unit = {
6         println("hey Spark Streaming")
7         val conf = new SparkConf().setMaster("local[2]").setAppName("SparkSteamingExample")
8         val sc = new SparkContext(conf)
9         val rootLogger = Logger.getRootLogger()
10        rootLogger.setLevel(Level.ERROR)
11        val ssc = new StreamingContext(sc, Seconds(15))
12        val lines = ssc.textFileStream("file:///home/acadgild/Desktop/SparkStream/")
13        val words = lines.flatMap(_.split(" "))
14        val wordCounts = words.map(x => (x, 1)).reduceByKey(_ + _)
15        wordCounts.print()
16        ssc.start()
17        ssc.awaitTermination()
18    }
19 }
20 }
```

CODE EXPLANATION –

1. Line 11: Setting up spark context.
2. Line 15: Setting up Spark streaming context which will stream for every 15 seconds.
3. Line 16: The **textFileStream** will check the directory path for streaming.

“/home/acadgild/Desktop/SparkStream/”

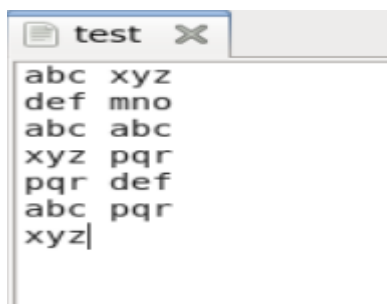


OUTPUT –

Below is the screenshot after running the program, the streaming has started and the directory **/SparkStream** do not have any files.

```
18/11/25 19:46:01 INFO BlockManager: Initialized BlockManager:
-----
Time: 1543155375000 ms
-----
Time: 1543155390000 ms
-----
Time: 1543155405000 ms
-----
Time: 1543155420000 ms
-----
Time: 1543155435000 ms
-----
Time: 1543155450000 ms
-----
```

Now let's copy the file **test.txt** into the directory **/SparkStream**.



test

```
abc xyz
def mno
abc abc
xyz pqr
pqr def
abc pqr
xyz|
```

Below is the output screenshot after copying the file **test.txt** to the directory **/SparkStream**.

```
-----
Time: 1543155435000 ms
-----
Time: 1543155450000 ms
-----
Time: 1543155465000 ms
-----
(mno,1)
(abc,4)
(pqr,3)
(def,2)
(xyz,3)
```

The word count has been executed on the fly.

PROBLEM STATEMENT 2:

We have to create a directory `"/user/streaming"` in Hadoop environment.

```
[acadgild@localhost ~]$ hadoop fs -mkdir /user/streaming
18/11/25 20:11:03 WARN util.NativeCodeLoader: Unable to load native-hadoop library
asses where applicable
[acadgild@localhost ~]$ hadoop fs -ls /user
18/11/25 20:11:13 WARN util.NativeCodeLoader: Unable to load native-hadoop library
asses where applicable
Found 2 items
drwxr-xr-x - acadgild supergroup          0 2018-02-09 14:50 /user/hive
drwxr-xr-x - acadgild supergroup          0 2018-11-25 20:11 /user/streaming
You have new mail in /var/spool/mail/acadgild
```

We will be using the `test.txt` file to demonstrate this example.

CODE -

```
6 object SparkHDFSWordCountComparison {
7
8   private var localFilePath: File = new File("/home/acadgild/Desktop/SparkStream/test.txt")
9   private var dfsDirPath: String = "hdfs://localhost:8020/user/streaming"
10  private val NPARAMS = 2
11
12
13  def main(args: Array[String]): Unit = {
14    //parseArgs(args)
15    println("SparkHDFSWordCountComparison : Main Called Successfully")
16    println("Performing local word count")
17    val fileContents = readFile(localFilePath.toString())
18    println("Performing local word count - File Content ->>" + fileContents)
19    val localWordCount = runLocalWordCount(fileContents)
20    println("SparkHDFSWordCountComparison : Main Called Successfully -> Local Word Count is ->>" + localWordCount)
21    println("Performing local word count Completed !!")
22    println("Creating Spark Context")
23    val conf = new SparkConf().setMaster("local[2]").setAppName("SparkHDFSWordCountComparisonApp")
24    val sc = new SparkContext(conf)
25    val rootLogger = Logger.getLogger()
26    rootLogger.setLevel(Level.ERROR)
27    println("Spark Context Created")
28    println("Writing local file to DFS")
29    val dfsFilename = dfsDirPath + "/dfs_read_write_test"
30    val fileRDD = sc.parallelize(fileContents)
31    fileRDD.saveAsTextFile(dfsFilename)
32    println("Writing local file to DFS Completed")
33    println("Reading file from DFS and running Word Count")
34    val readFileRDD = sc.textFile(dfsFilename)
```

```
35    val dfsWordCount = readFileRDD
36      .flatMap(_.split(" "))
37      .flatMap(_.split("\t"))
38      .filter(_.nonEmpty)
39      .map(w => (w, 1))
40      .countByKey()
41      .values
42      .sum
43    sc.stop()
44    if (localWordCount == dfsWordCount) {
45      println(s"Success! Local Word Count ($localWordCount) " +
46        s"and DFS Word Count ($dfsWordCount) agree.")
47    } else {
48      println(s"Failure! Local Word Count ($localWordCount) " +
49        s"and DFS Word Count ($dfsWordCount) disagree.")
50    }
51  }
52  private def readFile(filename: String): List[String] = {
53    val lineIter: Iterator[String] = fromFile(filename).getLines()
54    val lineList: List[String] = lineIter.toList
55    lineList
56  }
57  def runLocalWordCount(fileContents: List[String]): Int = {
58    fileContents.flatMap(_.split(" ")).flatMap(_.split("\t"))
59      .filter(_.nonEmpty).groupBy(w => w)
60      .mapValues(_.size).values.sum
61  }
62 }
```

CODE EXPLANATION –

1. Line 8: contains the local directory which contains the test.txt file on which word count will be performed and Line 9: contains the Hadoop directory to which the contents of word count will be copied to.

2. Line 19: Will perform local word count on **test.txt** file

3. Line 31: Copy the contents from **test.txt** to **“/user/streaming/dfs_read_write_test”**

4. Line 35: Performs word count from the RDD **readFileRDD**

OUTPUT –

Below is the screenshot of local word count on test.txt.

```
SparkHDFSWordCountComparison$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Nov 25, 2018, 8:11:50 PM)
SparkHDFSWordCountComparison : Main Called Successfully
Performing local word count
Performing local word count - File Content ->List(abc xyz, def mno, abc abc, xyz pqr, pqr def, abc pqr, xyz)
SparkHDFSWordCountComparison : Main Called Successfully -> Local Word Count is ->13
Performing local word count Completed !!
Creating Spark Context
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```

The below screenshot shows the file being written to Hadoop and displays successful message.

```
18/11/25 20:12:03 INFO BlockManager: Initialized BlockManager: BlockManagerId(
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 (13) and DFS Word Count (13) agree.
```

Now let's check the data written to Hadoop, refer the below screenshot.

```
[acadgild@localhost ~]$ hadoop fs -ls /user/streaming
18/11/25 20:13:04 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using
asses where applicable
Found 1 items
drwxr-xr-x  - acadgild supergroup          0 2018-11-25 20:12 /user/streaming/dfs_read_write_test
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -ls /user/streaming/dfs_read_write_test
18/11/25 20:13:35 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using
asses where applicable
Found 3 items
-rw-r--r--  3 acadgild supergroup          0 2018-11-25 20:12 /user/streaming/dfs_read_write_test/_SUCCESS
-rw-r--r--  3 acadgild supergroup        24 2018-11-25 20:12 /user/streaming/dfs_read_write_test/part-00000
-rw-r--r--  3 acadgild supergroup        28 2018-11-25 20:12 /user/streaming/dfs_read_write_test/part-00001
[acadgild@localhost ~]$ hadoop fs -cat /user/streaming/dfs_read_write_test/part-00000
18/11/25 20:13:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using
asses where applicable
abc xyz
def mno
abc abc
[acadgild@localhost ~]$ hadoop fs -cat /user/streaming/dfs_read_write_test/part-00001
18/11/25 20:14:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using
asses where applicable
xyz pqr
pqr def
abc pqr
xyz
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ █
```