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

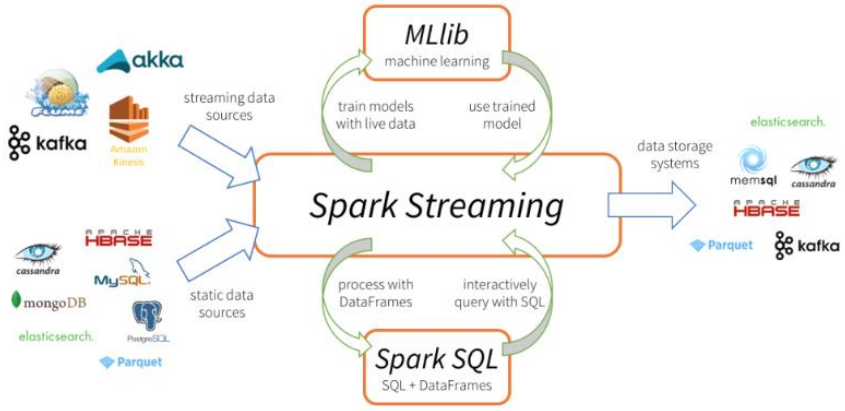
2. Then in the same Spark Application, write the code to put the same file on HDFS.

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

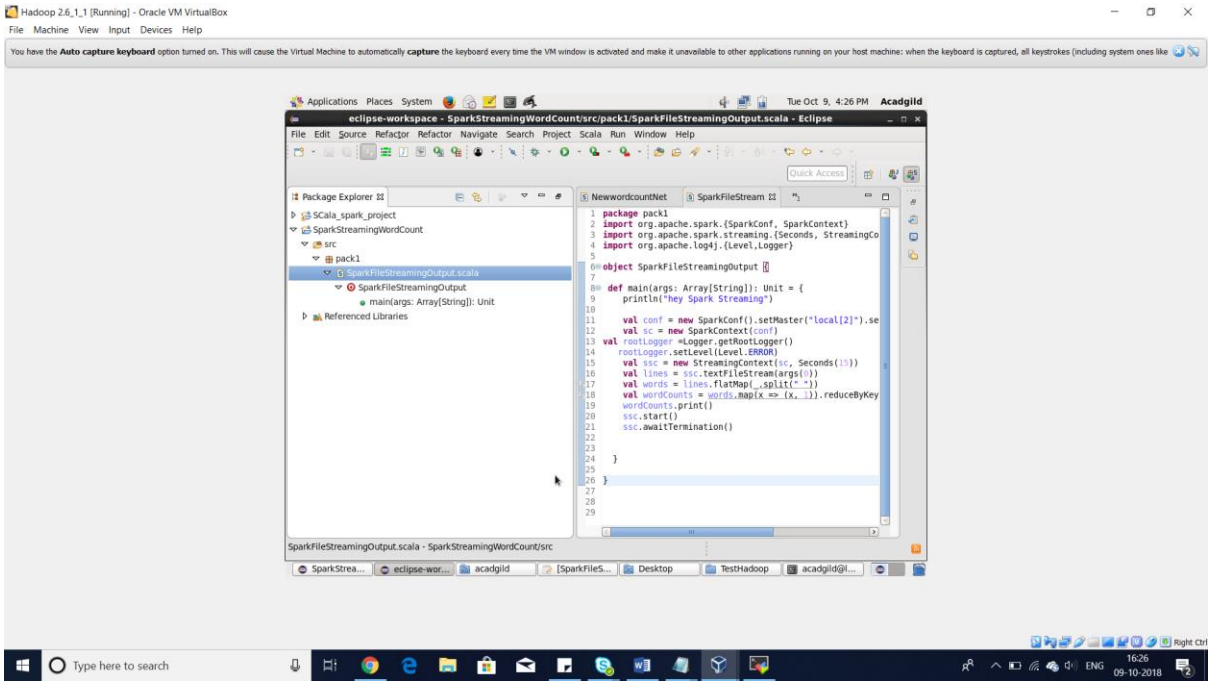
4. Lastly, compare the word count of step 1 and 2. Both should match, other throw an error

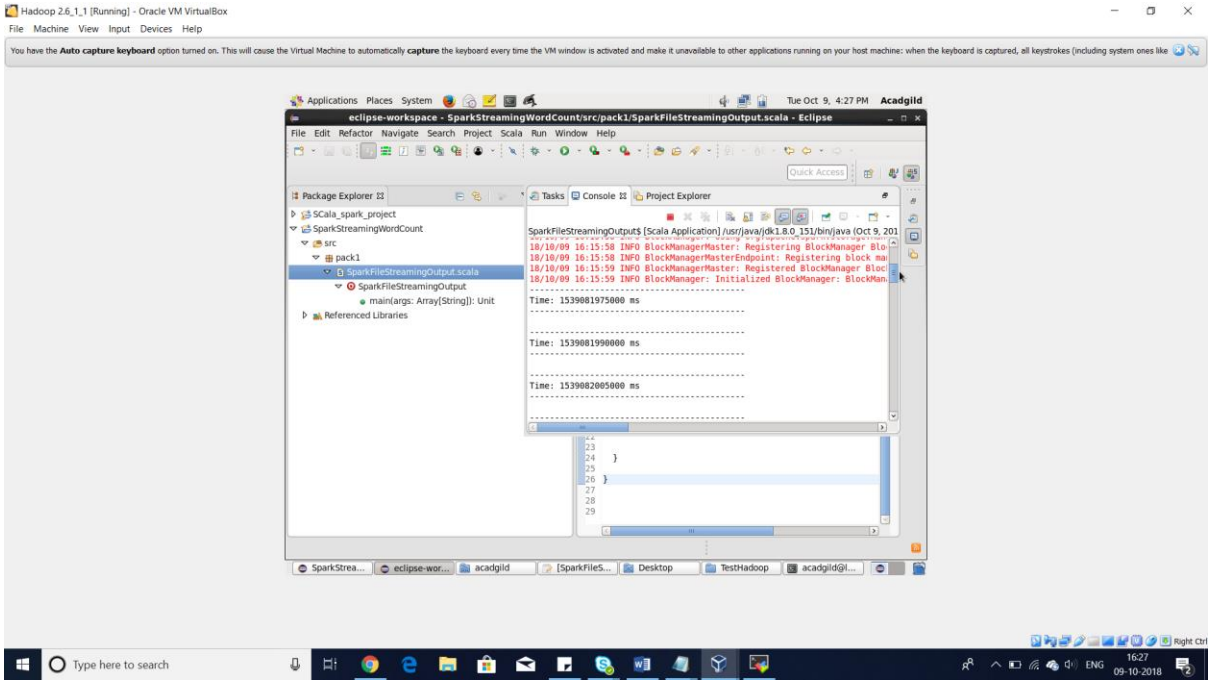
Solution:

Apache Spark Streaming is a scalable fault-tolerant streaming processing system that natively supports both batch and streaming workloads. Spark Streaming is an extension of the core Spark API that allows data engineers and data scientists to process real-time data from various sources including (but not limited to) Kafka, Flume, and Amazon Kinesis. This processed data can be pushed out to file systems, databases, and live dashboards. Its key abstraction is a Discretized Stream or, in short, a DStream, which represents a stream of data divided into small batches. DStreams are built on RDDs, Spark's core data abstraction. This allows Spark Streaming to seamlessly integrate with any other Spark components like MLlib and Spark SQL.

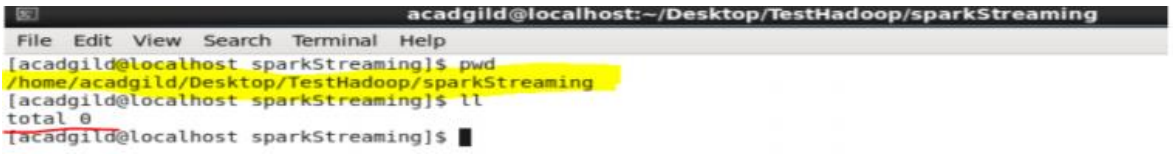


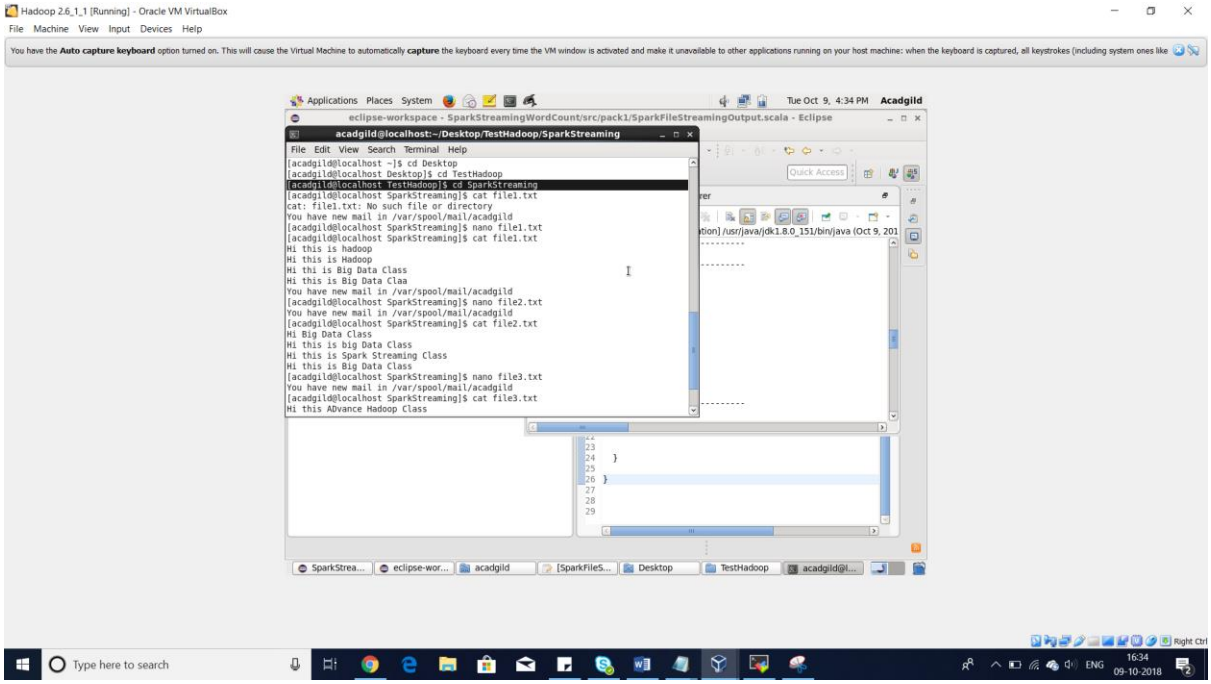
Below is the screen shot of the code run in eclipse for spark streaming. Separte files are attached in GitHub for the code also.

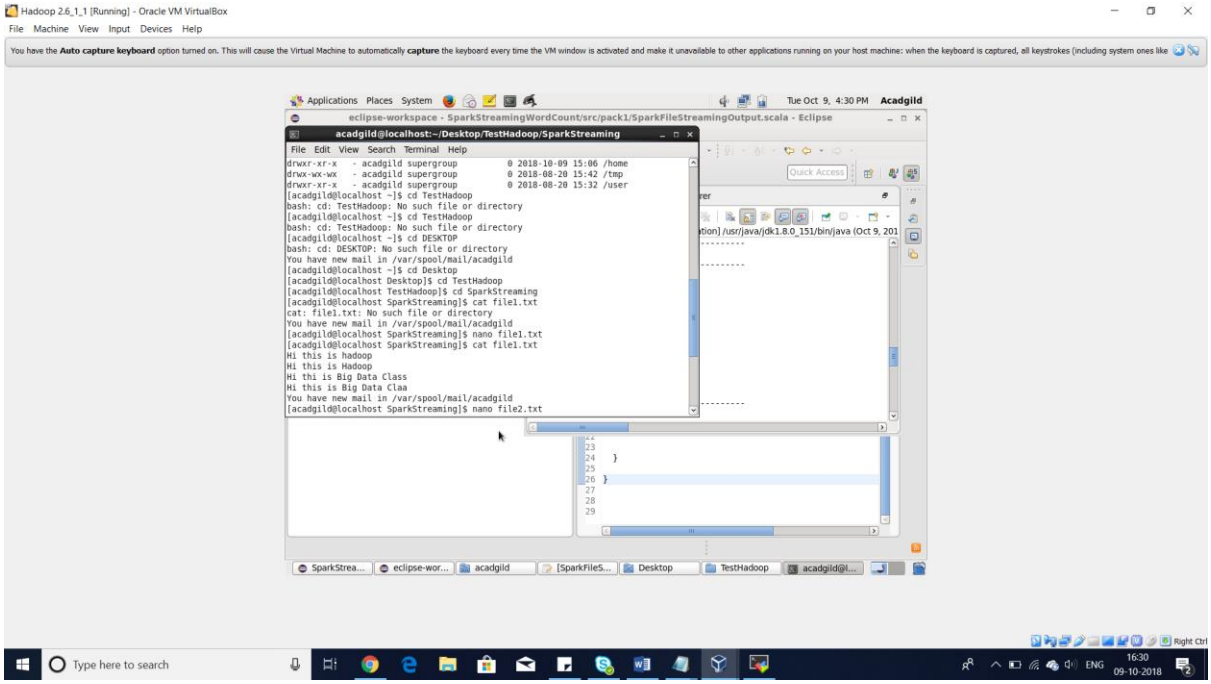


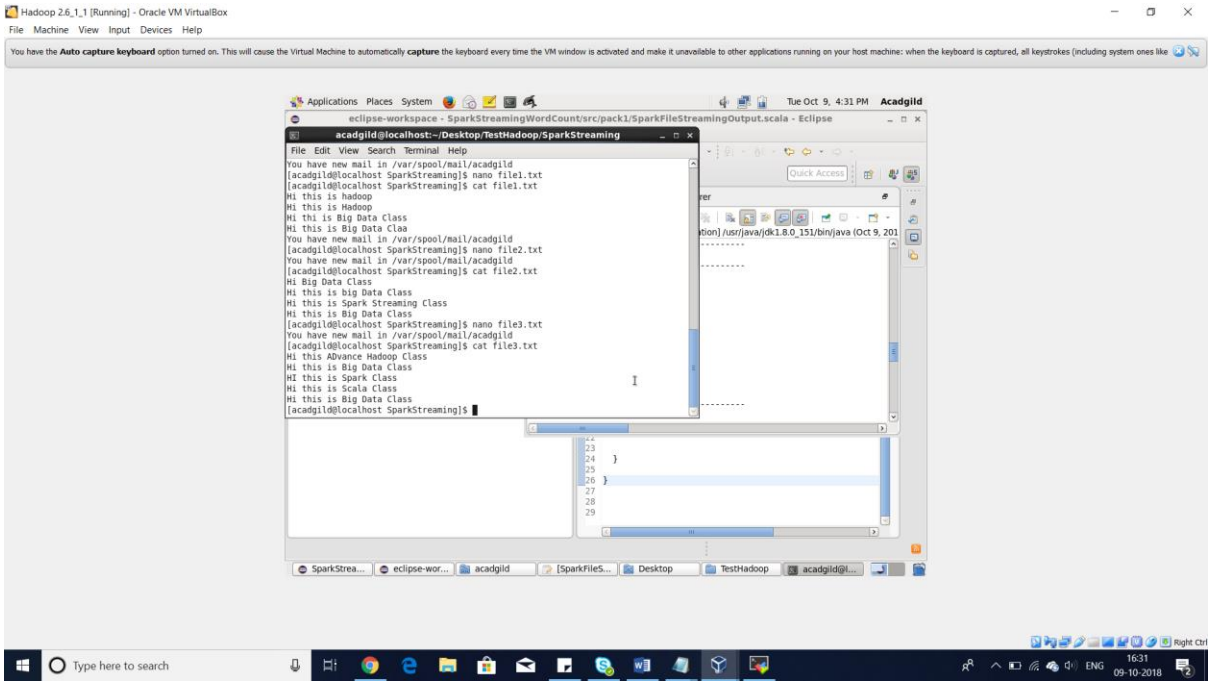


The Directory created for streaming file is: /home/acadgild/Desktop/TestHadoop/SparkStreaming

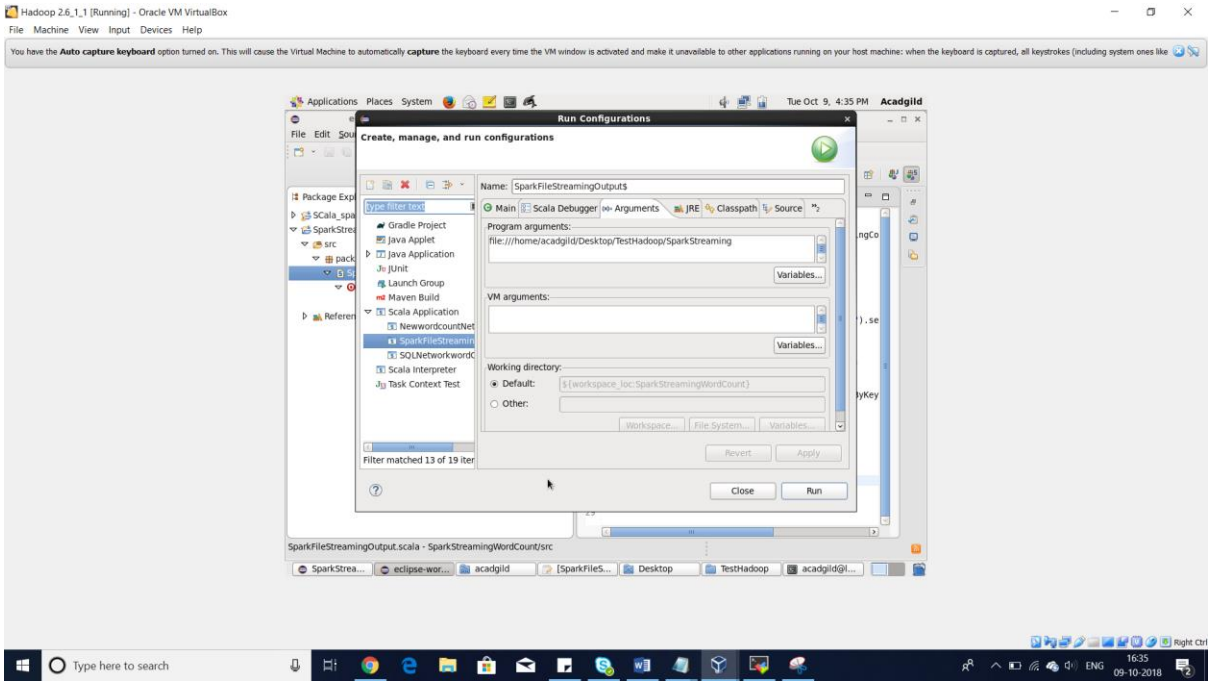




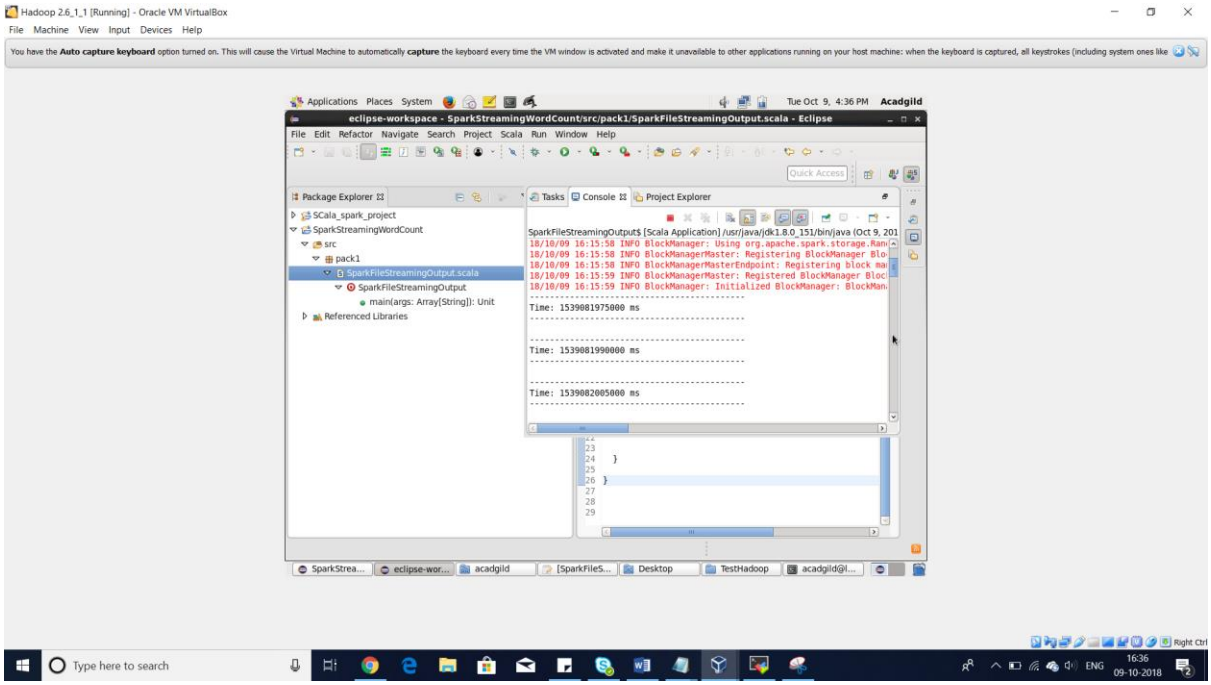

Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is captured, all keystrokes (including system ones like
Applications Places System
Tue Oct 9, 4:30 PM Acadgild
eclipse-workspace - SparkStreamingWordCount/src/pack1/SparkFileStreamingOutput.scala - Eclipse
acadgild@localhost:~/Desktop/TestHadoop/SparkStreaming
File Edit View Search Terminal Help
drwxr-xr-x - acadgild supergroup 0 2018-10-09 15:06 /home
drwx-wx-wx - acadgild supergroup 0 2018-08-20 15:42 /tmp
drwxr-xr-x - acadgild supergroup 0 2018-08-20 15:32 /user
[acadgild@localhost ~]$ cd TestHadoop
bash: cd: TestHadoop: No such file or directory
[acadgild@localhost ~]$ cd TestHadoop
bash: cd: TestHadoop: No such file or directory
[acadgild@localhost ~]$ cd DESKTOP
bash: cd: DESKTOP: No such file or directory
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ cd Desktop
[acadgild@localhost Desktop]$ cd TestHadoop
[acadgild@localhost TestHadoop]$ cd SparkStreaming
[acadgild@localhost SparkStreaming]$ cat file1.txt
cat: file1.txt: No such file or directory
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost SparkStreaming]$ nano file1.txt
[acadgild@localhost SparkStreaming]$ cat file1.txt
Hi this is hadoop
Hi this is Hadoop
Hi thi is Big Data Class
Hi this is Big Data Claa
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost SparkStreaming]$ nano file2.txt
Quick Access
ation] /usr/java/jdk1.8.0_151/bin/java (Oct 9, 201
SparkStrea... eclipse-wor... acadgild [SparkFileS... Desktop TestHadoop acadgild@l...
Type here to search
ENG 16:30 09-10-2018


Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is captured, all keystrokes (including system ones like
Applications Places System
Tue Oct 9, 4:31 PM Acadgild
eclipse-workspace - SparkStreamingWordCount/src/pack1/SparkFileStreamingOutput.scala - Eclipse
acadgild@localhost:~/Desktop/TestHadoop/SparkStreaming
File Edit View Search Terminal Help
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost SparkStreaming]$ nano file1.txt
[acadgild@localhost SparkStreaming]$ cat file1.txt
Hi this is hadoop
Hi this is Hadoop
Hi thi is Big Data Class
Hi this is Big Data Claa
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost SparkStreaming]$ nano file2.txt
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost SparkStreaming]$ cat file2.txt
Hi Big Data Class
Hi this is big Data Class
Hi this is Spark Streaming Class
Hi this is Big Data Class
[acadgild@localhost SparkStreaming]$ nano file3.txt
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost SparkStreaming]$ cat file3.txt
Hi this ADvance Hadoop Class
Hi this is Big Data Class
HI this is Spark Class
Hi this is Scala Class
Hi this is Big Data Class
[acadgild@localhost SparkStreaming]$
Quick Access
ation] /usr/java/jdk1.8.0_151/bin/java (Oct 9, 201
SparkStrea... eclipse-wor... acadgild [SparkFileS... Desktop TestHadoop acadgild@l...
Type here to search
ENG 16:31 09-10-2018

Specifying the above created directory as input to program arguments

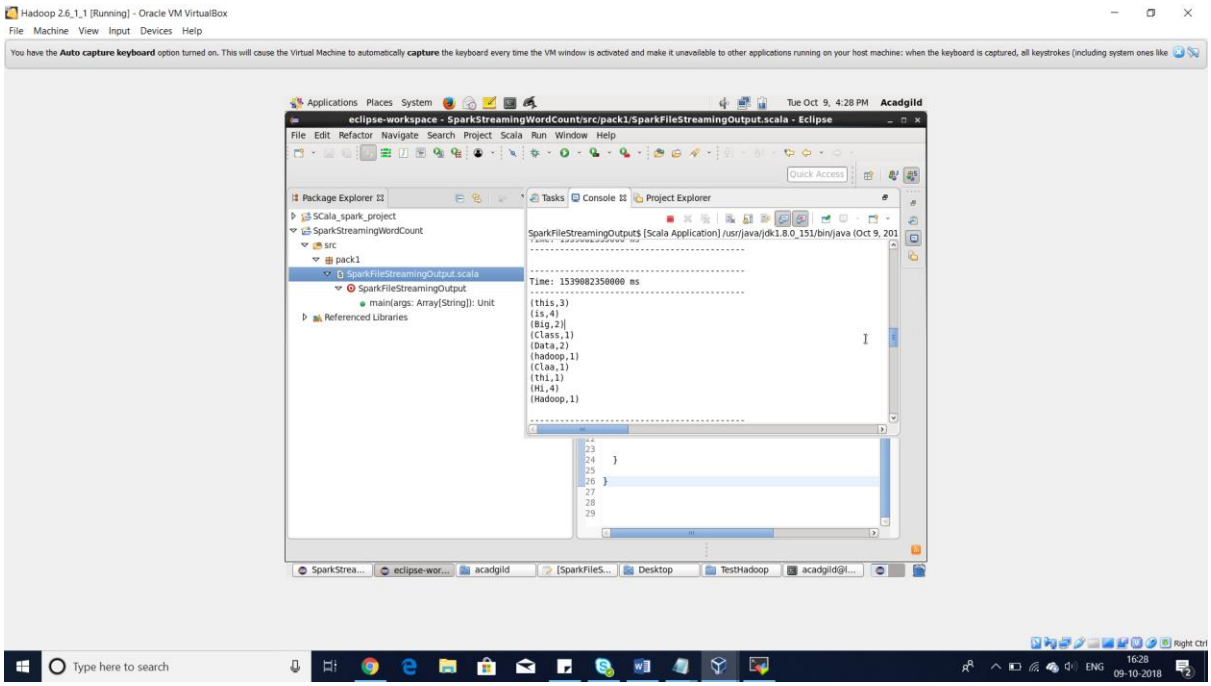


arg(0) specified within the program reads entire files that are getting added to the above directory & performs the word count operation on the fly. Run the application as Scala Application.

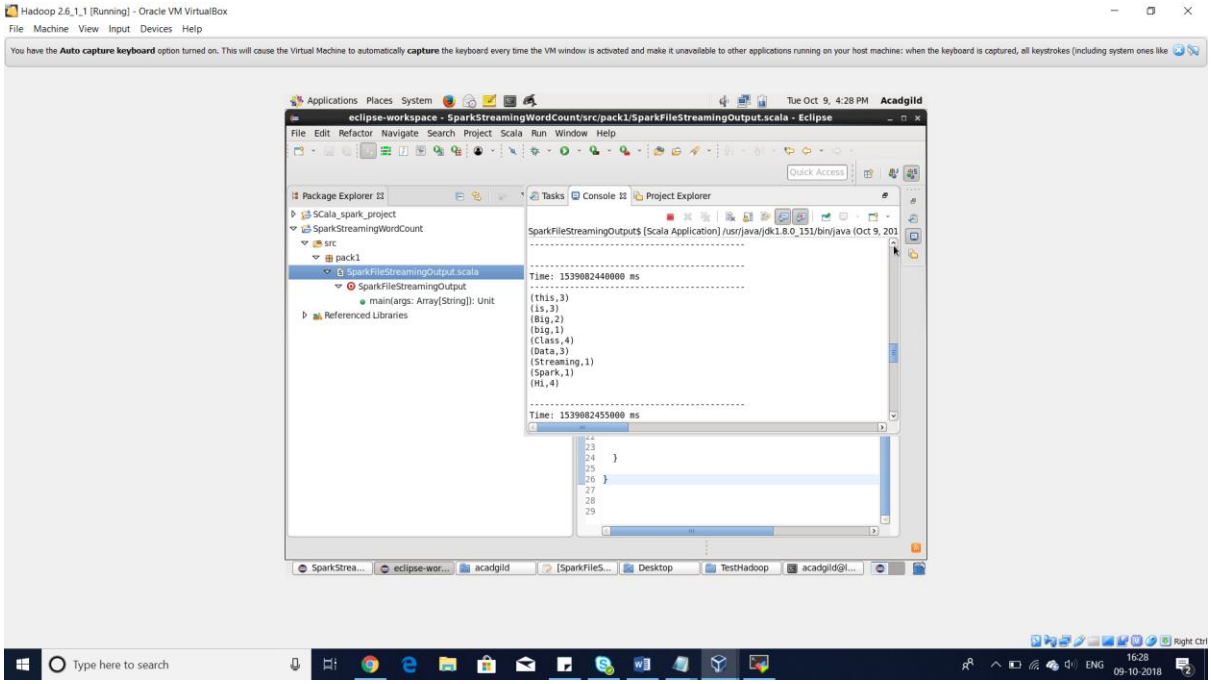


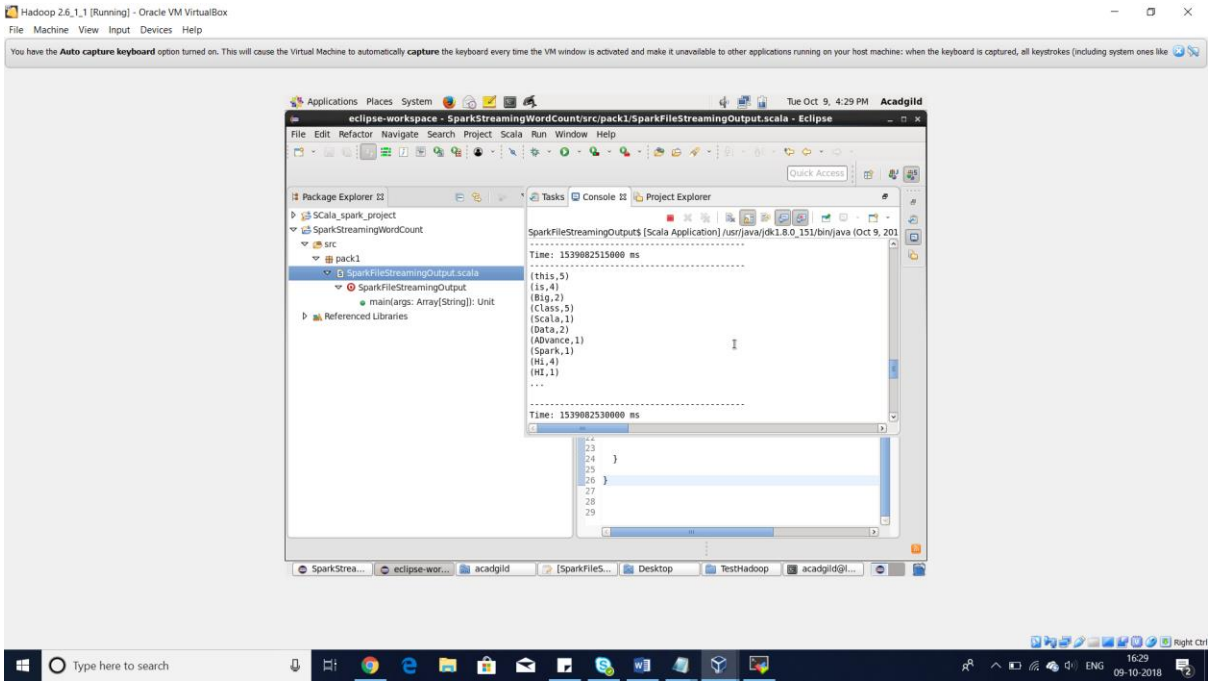
In our program we have created three text file file1, file2 and fly3 in the directory on the fly for spark streaming.

The contents of file1.txt getting read by spark streaming application and computing word count on the fly.



Similarly, creating two other files file2.txt & file3.txt whose words would be counted by spark streaming application on the fly





Objective 2: 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

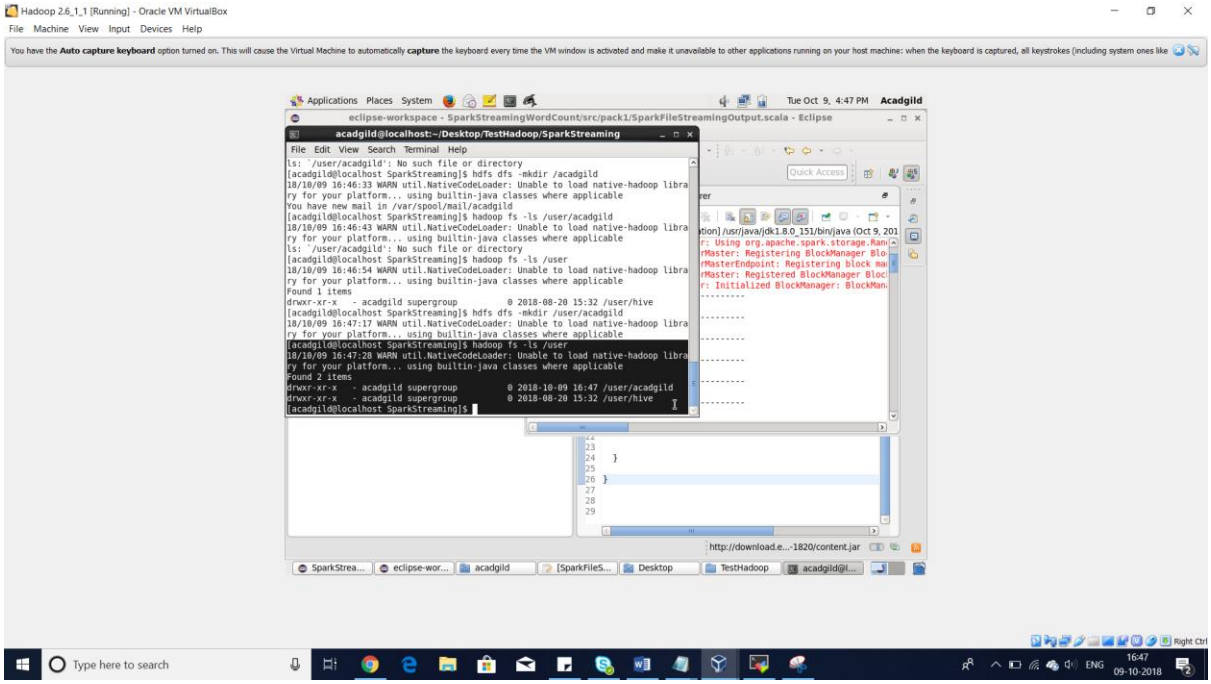
2. Then in the same Spark Application, write the code to put the same file on HDFS.

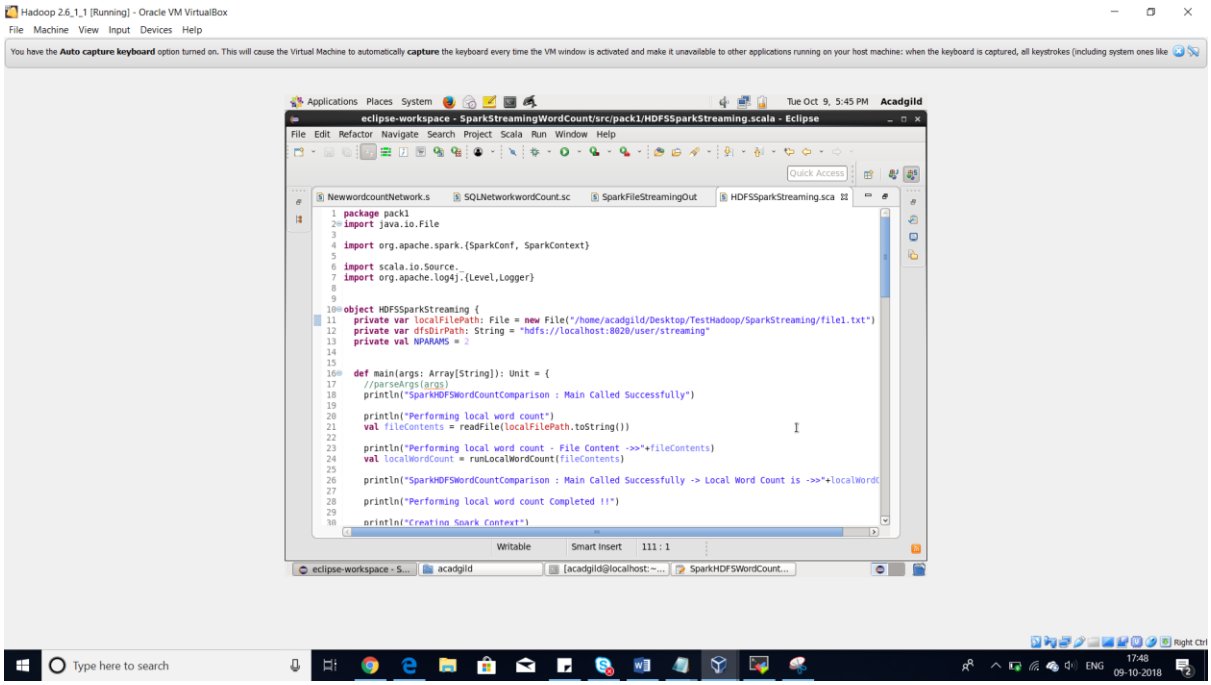
3. Then in same Spark Application, do the word count of the file copied on HDFS in step 2

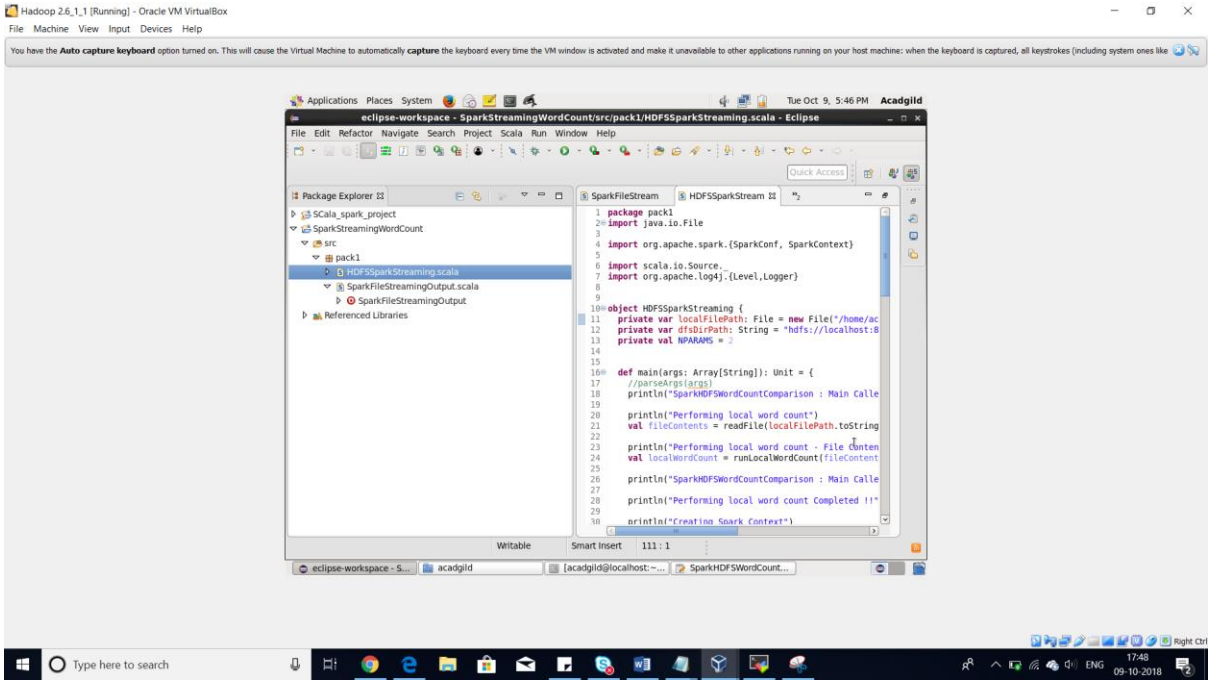
4. Lastly, compare the word count of step 1 and 2. Both should match, other throw an error.

Solution:

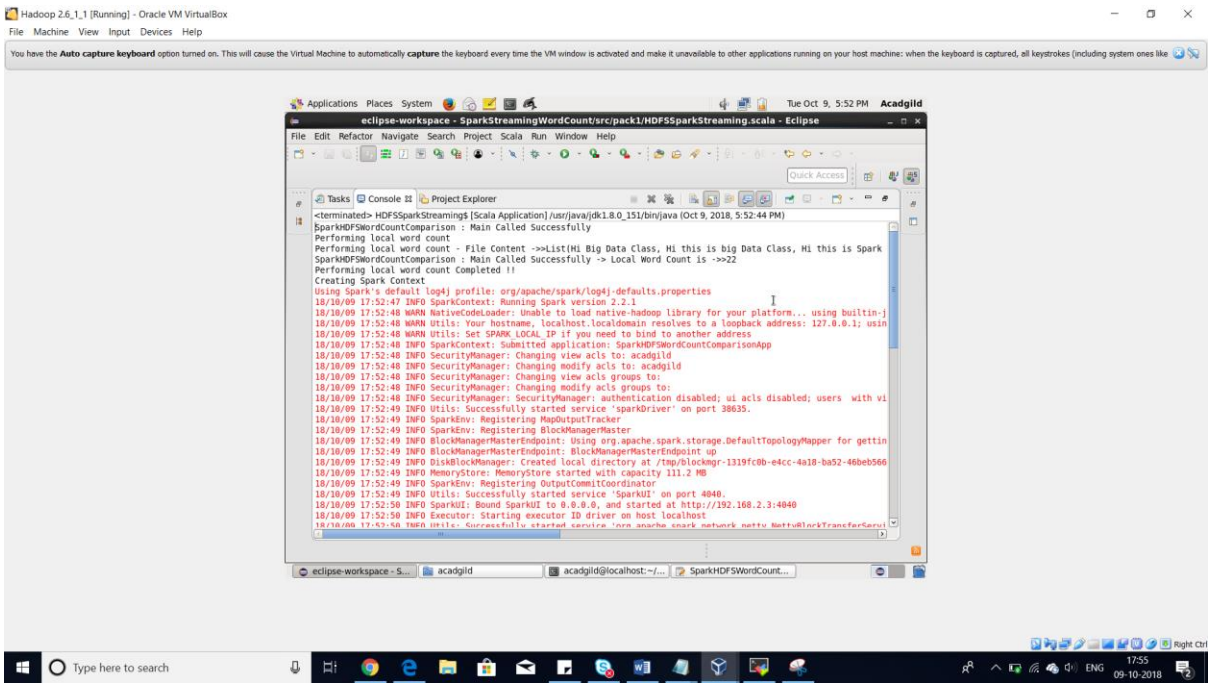
HDFS does not contain streaming directory before the application is run


Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is captured, all keystrokes (including system ones like
Applications Places System
Tue Oct 9, 4:47 PM Acadgild
eclipse-workspace - SparkStreamingWordCount/src/pack1/SparkFileStreamingOutput.scala - Eclipse
acadgild@localhost:~/Desktop/TestHadoop/SparkStreaming
File Edit View Search Terminal Help
ls: `/user/acadgild': No such file or directory
[acadgild@localhost SparkStreaming]$ hdfs dfs -mkdir /acadgild
18/10/09 16:46:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost SparkStreaming]$ hadoop fs -ls /user/acadgild
18/10/09 16:46:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
ls: `/user/acadgild': No such file or directory
[acadgild@localhost SparkStreaming]$ hadoop fs -ls /user
18/10/09 16:46:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x - acadgild supergroup 0 2018-08-20 15:32 /user/hive
[acadgild@localhost SparkStreaming]$ hdfs dfs -mkdir /user/acadgild
18/10/09 16:47:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[acadgild@localhost SparkStreaming]$ hadoop fs -ls /user
18/10/09 16:47:28 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 2018-10-09 16:47 /user/acadgild
drwxr-xr-x - acadgild supergroup 0 2018-08-20 15:32 /user/hive
[acadgild@localhost SparkStreaming]$
Type here to search
ENG
16:47
09-10-2018

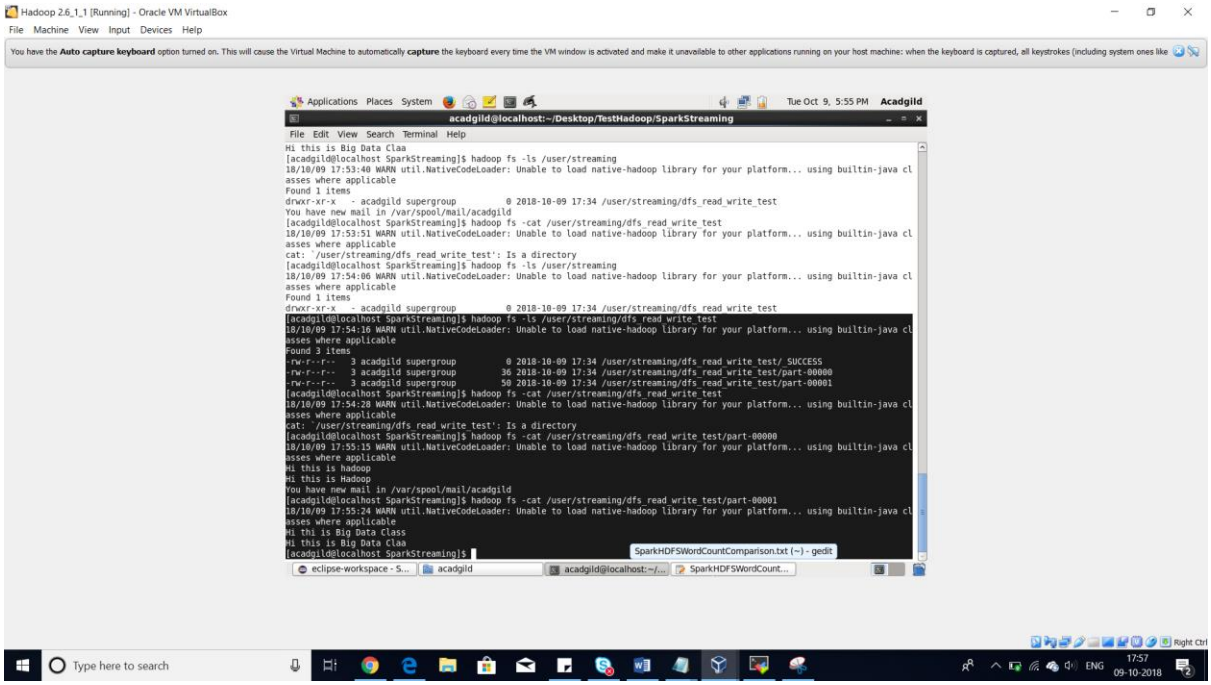

Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically capture the keyboard every time the VM window is activated and make it unavailable to other applications running on your host machine: when the keyboard is captured, all keystrokes (including system ones like
Applications Places System
Tue Oct 9, 5:45 PM Acadgild
eclipse-workspace - SparkStreamingWordCount/src/pack1/HDFSSparkStreaming.scala - Eclipse
File Edit Refactor Navigate Search Project Scala Run Window Help
NewwordcountNetwork.s
SQLNetworkwordCount.sc
SparkFileStreamingOut
HDFSSparkStreaming.sca
package pack1
import java.io.File
import org.apache.spark.{SparkConf, SparkContext}
import scala.io.Source._
import org.apache.log4j.{Level,Logger}
object HDFSSparkStreaming {
private var localFilePath: File = new File("/home/acadgild/Desktop/TestHadoop/SparkStreaming/file1.txt")
private var dfsDirPath: String = "hdfs://localhost:8020/user/streaming"
private val NPARAMS = 2
def main(args: Array[String]): Unit = {
//parseArgs(args)
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 ->>"+localWordC
println("Performing local word count Completed !!")
println("Creating Spark Context")
Writable
Smart Insert
111 : 1
Type here to search
ENG
17:48
09-10-2018



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, otherwise throw an error.

Below screen shot shows the word count copied in HDFS and the word count in local directory match.