**Steps for Word Count in Cloudera: (Without Combiner)**

1. Open virtual box and then start cloudera quickstart.



1. Open Eclipse present on the cloudera desktop.



3) Create a new Java project clicking: File -> New -> Project -> Java Project -> Next (“WordCount” is the  project name).



4)Adding the Hadoop libraries to the project Click on Libraries -> Add External JARs Click on  File System -> usr -> lib -> hadoop Select all the libraries (JAR Files) -> click OK Click on  Add External jars, -> client -> select all jar files -> ok -> Finish



















5) Right Click on the name of Project “WordCount” -> New -> class Don’t write anything for package Write Name Textbox write “WordCount” -> Finish Then WordCount.java window will pop up









6) Right Click on the project name WordCount -> Export -> Java -> JAR File -> Next -> for select the export destination for JAR file: browse -> Name : WordCount.jar -> save in folder -> cloudera -> Finish -> OK











7) Verify jar file from terminal by using Open terminal & type “ ls ” There it will show WordCount.jar

Check current working directory ->pwd





8) We need to create an input file in local file system

Creating an input file named as “abc”.



Here listing all the directory present in hdfs using hdfs dfs -ls / command



9) Now we have to move this input file to hdfs. For this we create a direcory on hdfs using

command hdfs dfs -mkdir /inputnew.



Then we can verify whether this directory is created or not using ls command hdfs dfs -ls /



Move the input file to this directory created in hdfs by using either put command or copyFromLocal command.



Now checking whether the “abc” present in /inputdir directory of hdfs or not using hdfs dfs -ls /inputdir command



As we can see “abc” file is present in /inputdir directory of hdfs. Now we will see the content of this file using hdfs dfs –cat /inputdir/abc command



10) Running Mapreduce Program on Hadoop, syntax is hadoop jar jarFileName.jar ClassName /InputFileAddress /outputdir

**i.e. hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /outputdir**



**Map-Reduce Framework**



**As we can see in the above output,**

**Combine input records=0**

**Combine output records=0**

**We are getting this because we have commented the Combiner line in main function.**

**And Reduce shuffle bytes coming as,**

**Reduce shuffle bytes=1876**

**So when we are not using combiner 1876 bytes acting as an input for the reducer.**

11) Then we can verify the content of outputdir directory and in that part-r file has the actual

output by using the command Hdfs dfs -cat /outputdir/part-r-00000 This will give us final output.

The same file can also be accessed using a browser. For every execution of this program we need

to delete the output directory or give a new name to the output directory every time.

1st we are checking whether the outputdir directory is created in hdfs or not using command

**hdfs dfs -ls /**



Now let’s check what we have inside this **outputdir** directory using command as **hdfs dfs -ls**

**/outputdir**



Now we want to read the content of the **part-r-00000 file** which present inside the **outputdir**

using command **hdfs dfs -cat /outputdir/part-r-00000**





**It will give the count of number of times each word has occurred as output.**

**12) The same file can also be accessed using a browser.**

Browse the Directory by

**Hadoop->HDFS Namenode->Ultilities ->Browse the file system**





Now downloading the **part-r-00000** file.



Inside the **part-r-00000** file it will have the same output as we are getting after executing using

command **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**





**For every execution of this program we need to delete the output directory or give a new**

**name to the output directory every time.**

**Implementation of WordCount problem using Hadoop MapReduce (With Combiner) in Eclipse:**

1) We will perform the same steps as we have done above for WordCount (without using

combiner) in that we just uncommenting the combiner line in main function**.**



2)And will delete the WordCount.jar file in which all jar files are present from **/home/cloudera.**



**We have successfully deleted the WordCount.jar file.**



1. Now exporting the jar files Right Click on the project name WordCount -> Export -> Java ->JAR File -> Next -> for select the export destination for JAR file: browse -> Name : WordCount.jar -> save in folder -> cloudera -> Finish -> OK











4)Now checking the WordCount.jar file is created or not using –ls command



5) Running Mapreduce Program on Hadoop, syntax is hadoop jar jarFileName.jar ClassName

/InputFileAddress /outputdir

**i.e. hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**

here I am using the same input file ‘abc’ which I have created earlier for WordCount

example (Without Combiner**). For every execution of this program we need to delete the**

**output directory or give a new name to the output directory every time.** So here I am

giving the new name to the output directory as **‘op1’.**





* As we can see from above image the the combiner input and output records coming out as,

**Combine input records=177**

**Combine output records=84**

* Earlier it was coming out as “zero” while executing WordCount (without combiner).

**Combine input records=0**

**Combine output records=0**

* And also here we are getting the Reduce Shuffle bytes as,

**Reduce shuffle bytes=942**

Earlier while executing WordCount (without combiner) it is coming out as,

**Reduce shuffle bytes=1876**

* So Combiner is used to save the Network Bandwidth. So for saving the Network

bandwidth we make use of combiner. So instead of sending every word over the

network what we do is we incorporate the logic of the reducer at the combiner side so

that the less amount of information can be transmitted over the network.

* So when we are not using combiner 1876 bytes acting as an input for the reducer. And

when we are making use of the combiner so 942 bytes acting as input for the reducer.

6) The same file can also be accessed using a browser.

Browse the Directory by

**Hadoop->HDFS Namenode->Ultilities ->Browse the file system**





Now downloading the **part-r-00000** file.



Inside the **part-r-00000** file it will have the same output as we are getting after executing using

command **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**



