

**CSE488: Big Data Analytics**

**[SPRING 2023]**

**Assignment**

**MapReduce used for Word Count**

**Submitted by:**

Student ID: 2019-3-60-046

Student Name: Mohsenul Kabir Mithun

**Assignment Objective:**

Familiarize with the Hadoop Big Data Processing Platform and MapReduce Programming Model.

**Assignment Outcome:**

After completing this assignment successfully able to learn:

1. Install and use a virtual machine executing Hadoop Framework.

2. Write basic MapReduce programs and execute in Hadoop Framework.

**Required Applications/Tools**

* Cloudera Distribution of Hadoop
  + Download Cloudera from this URL <https://www.cloudera.com/downloads.html>
  + Since Access Restricted now you can use this link to download Cloudera QuickStart VM <https://mega.nz/folder/CGYxiJwY#B_9BNqbh8U-GSGdMdrQGsA>
* VirtualBox
  + Download VirtualBox from this URL https://www.virtualbox.org/wiki/Downloads

**Setting up a Virtual Machine**

* Download and install VirtualBox on your machine.
* Download the Cloudera QuickStart VM
* Uncompress the VM archive.
* Start VirtualBox and click Import Appliance in the File dropdown menu. Click the folder icon beside the location field. Browse to the uncompressed archive folder, select the .ovf file, and click the Open button. Click the Continue button. Click the Import button.
* Your virtual machine should now appear in the left column. Select it and click on Start to launch it.

**Step Screenshot:**

**A screenshot of a computer

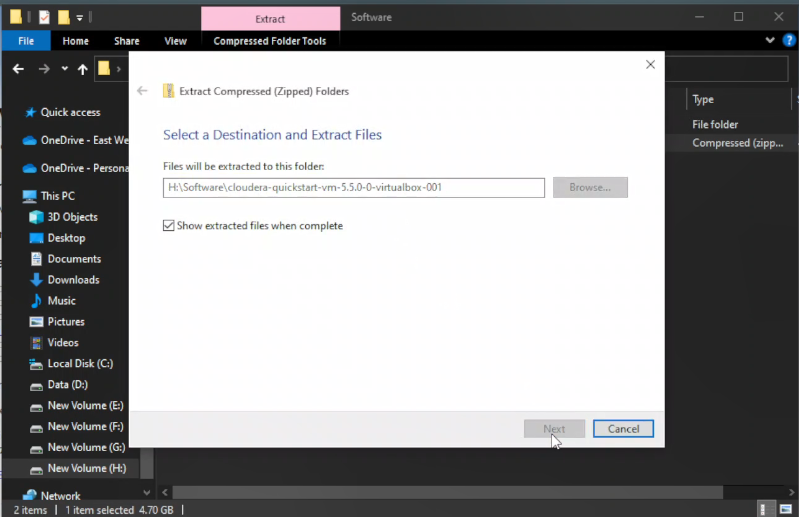
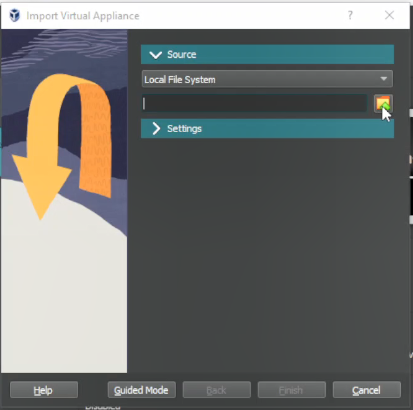
Description automatically generated with medium confidence**

Fig 1: Uncompress the VM archive Fig 2: Start VirtualBox and import appliance.

 A screenshot of a computer

Description automatically generated with medium confidence

Fig 3: Select the .ovf file

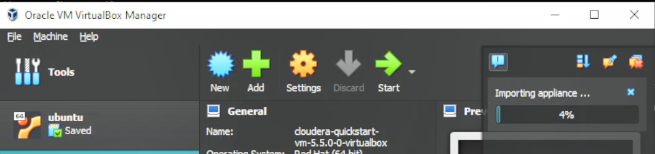


Fig 4: Click Start

Text

Description automatically generated

Fig 5: Finally, you see this screen.

**Creating a Hadoop project in Eclipse: (MapReduce used for WordCount)**

**🡺 MapReduce Program for Word Count:**

1. Open Eclipse IDE

2. Copy 'training' project and paste it on same place by name it "WordCount"

3. Expand WordCount 🡪 src 🡪 (default package)

4. Rename 4 of 3 java files are "WordCount.java", "MapForWordCount.java", "ReducerForWordCount.java" and replace the code that I provide bellow,

=====================================================================

WordCount.java

=====================================================================

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

public class WordCount {

public static void main(String[] args) throws Exception {

if (args.length != 2) {

System.out.printf("Usage: WordCount <input dir> <output dir>\n");

System.exit(-1);

}

Configuration config = new Configuration();

Path input = new Path(args[0]);

Path output = new Path(args[1]);

@SuppressWarnings("deprecation")

Job job = new Job();

job.setJarByClass(WordCount.class);

job.setJarByClass(WordCount.class);

job.setMapperClass(MapForWordCount.class);

job.setReducerClass(ReducerForWordCount.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, input);

FileOutputFormat.setOutputPath(job,output);

boolean success = job.waitForCompletion(true);

System.exit(success ? 0 : 1);

}

}

=====================================================================

MapForWordCount.java

=====================================================================

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class MapForWordCount extends Mapper<LongWritable, Text, Text, IntWritable> {

@Override

public void map(LongWritable key, Text value, Context context)

throws IOException, InterruptedException {

String line = value.toString();

String[] words = line.split(",");

for (String word: words){

Text outputKey = new Text(word.toUpperCase().trim());

IntWritable outputValue = new IntWritable(1);

context.write(outputKey, outputValue);

}

}

}

=====================================================================

ReducerForWordCount.java

=====================================================================

import java.io.IOException;

import org.apache.hadoop.io.DoubleWritable;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class ReducerForWordCount extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override

public void reduce(Text key, Iterable<IntWritable> values, Context context)

throws IOException, InterruptedException {

int sum = 0;

for(IntWritable value: values){

sum += value.get();

}

context.write(key, new IntWritable(sum));

}

}

=====================================================================

**Step Screenshot:**

Application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Fig 6: Open Eclipse IDE Fig 7: Copy 'training' project

Graphical user interface, application

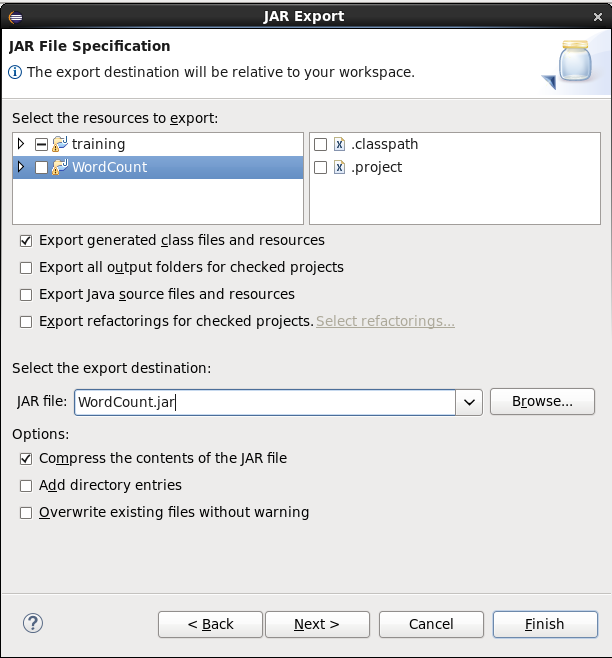
Description automatically generatedGraphical user interface, text, application

Description automatically generated

Fig 8: Paste and rename it “WordCount” Fig 9: Rename sll java file as step 4

5. Right click on 'WordCount' project 🡪 export 🡪 java 🡪 select jar 🡪 Next 🡪 give a jar file name as "WordCount" 🡪 Next 🡪 Next 🡪 Finish

6. Avoid if any error message occurs.



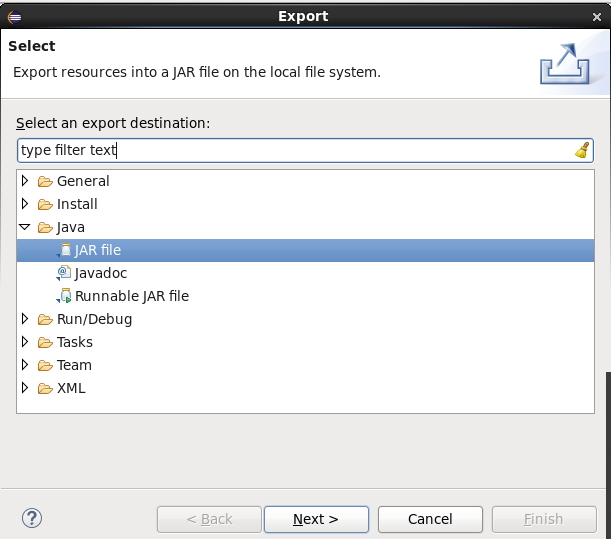


Fig 10: Export Jar file

=====================================================================

Open Web Browser

=====================================================================

7. Go to http://quickstart.cloudera:8888/accounts/login/?next=/

8. Login by using Username: cloudera & Password: cloudera

9. Go to file browser that you see upper right side of the browser

10. Create new directory by click on New 🡪 Directory 🡪 provide name as ‘wordcount’ 🡪 go that directory create new file as file1.txt

11. Go to that file1.txt 🡪 click on 'edit file' option and add some text as coma separated. Then save it.

Graphical user interface, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

Fig 11: Creating wordcount directory

Graphical user interface, text, application, email

Description automatically generated

Fig 12: Creating txt file with some text

=====================================================================

CMD

=====================================================================

12. open CMD and write bellow command

hadoop jar WordCount.jar WordCount /user/cloudera/wordcount/file1.txt wordcountout

More generic way to explain:

🡪 hadoop jar <jar file> <main class> <input path> <output path>

explain ==>

(i) 1st <jar file>, that is in my case is "WordCount.jar" if you get any error then provide complete location of that jar file. In that case the cmd will be "jar /home/cloudera/workspace/WordCount WordCount /user/cloudera/wordcount/ output"

‘’’

* Open computer from desktop home screen 🡪 filesystem 🡪 home 🡪 cloudera 🡪 workspace (here you will find the created jar file in default case if you do not select any location).
* If you move or copy the jar file in to one step back folder which is cloudera then you don't need to provide complete location. Then you can write simple step 12 cmd.
* You can also choose a location of jar file while creating a new jar file...😅

‘’’

(ii) 2nd "WordCount" is Project name

(iii) "/user/cloudera/wordcount/file1.txt" that is input location. if you have multiple txt file then provide just folder name like that "/user/cloudera/wordcount/"

(iv) "wordcountout" output location folder name. You don’t need to create this folder. It will create automatic.

13. To see output go to ‘Web Browser’ where you create a folder and add txt file. Here you also see another folder name as "wordcountout" In this folder you will find the output.

Or

Open CMD and run that command.

---------------------------------------------------------------------------------------------------------------------

hdfs dfs -cat /user/cloudera/wordcountout/part-r-00000

Explain:

"/user/cloudera/wordcountout/part-r-00000" location of my output.

To see all file list cmd:

hdfs dfs -ls /user/cloudera

Graphical user interface

Description automatically generated with low confidence

Fig 13: Run CMD command.

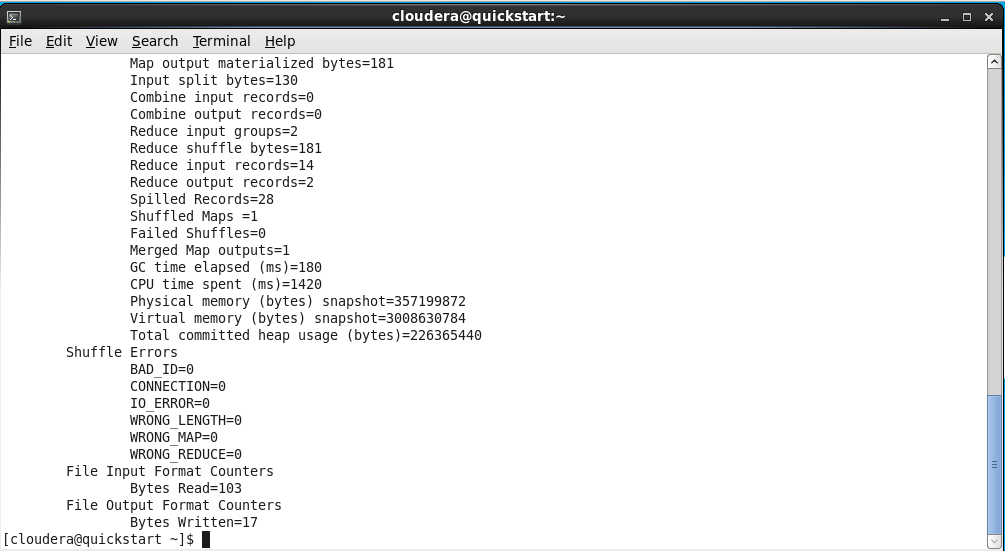


Fig 14: Final output of CMD

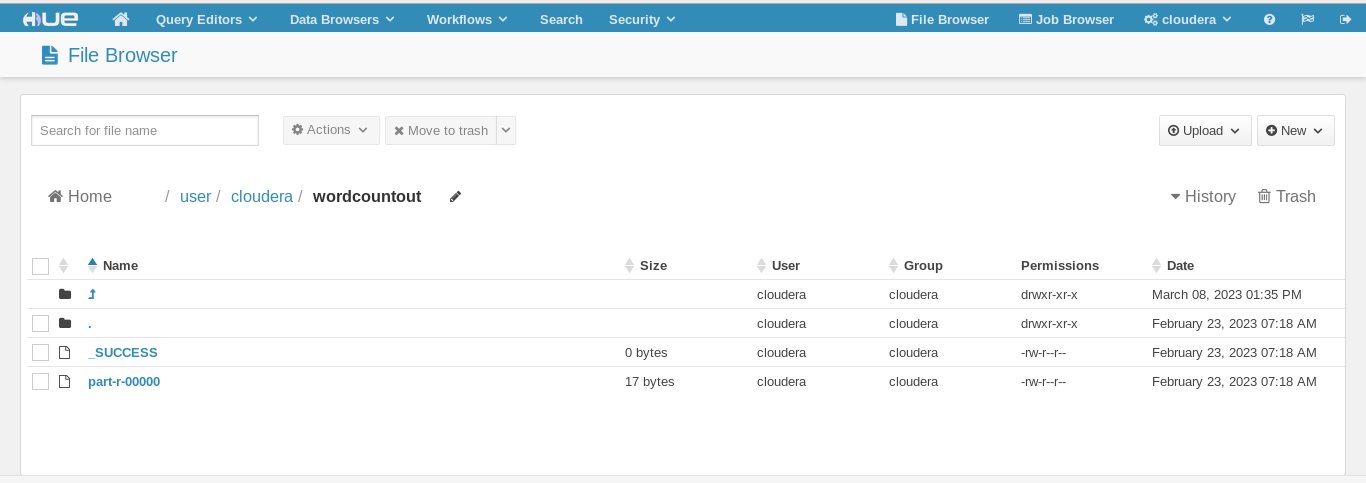


Fig 15: Open wordcountout to see that

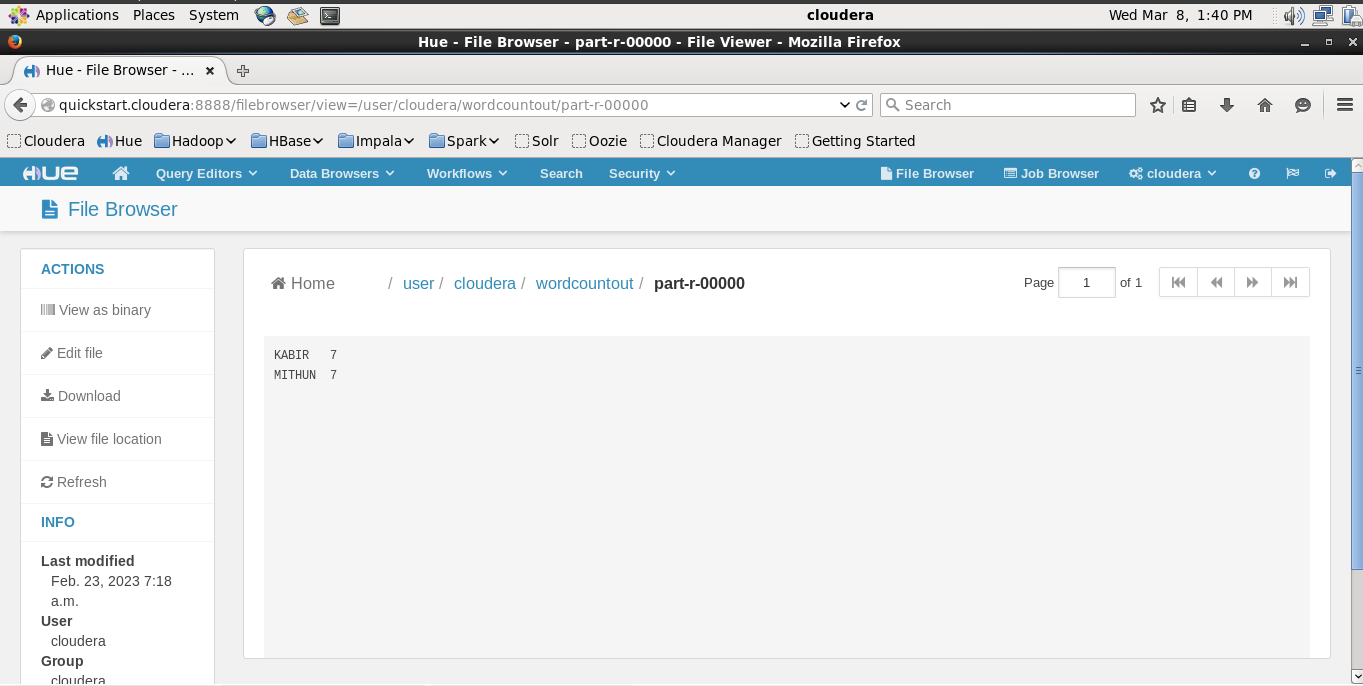


Fig 16: Final OutPut

**Alternative by cmd:**

Graphical user interface, text, application, email

Description automatically generated

Thank You 🙂