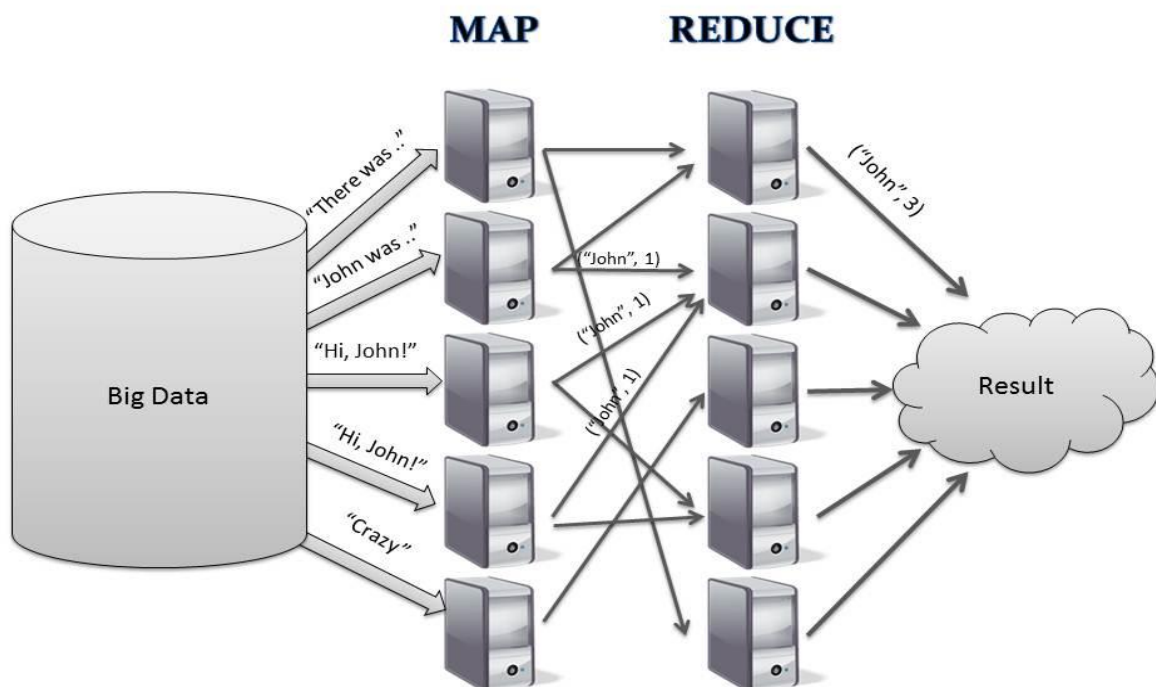


Documentation of Map Reduce

-Prepared by Vignesh.R (15CSE107)

Abstract:

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

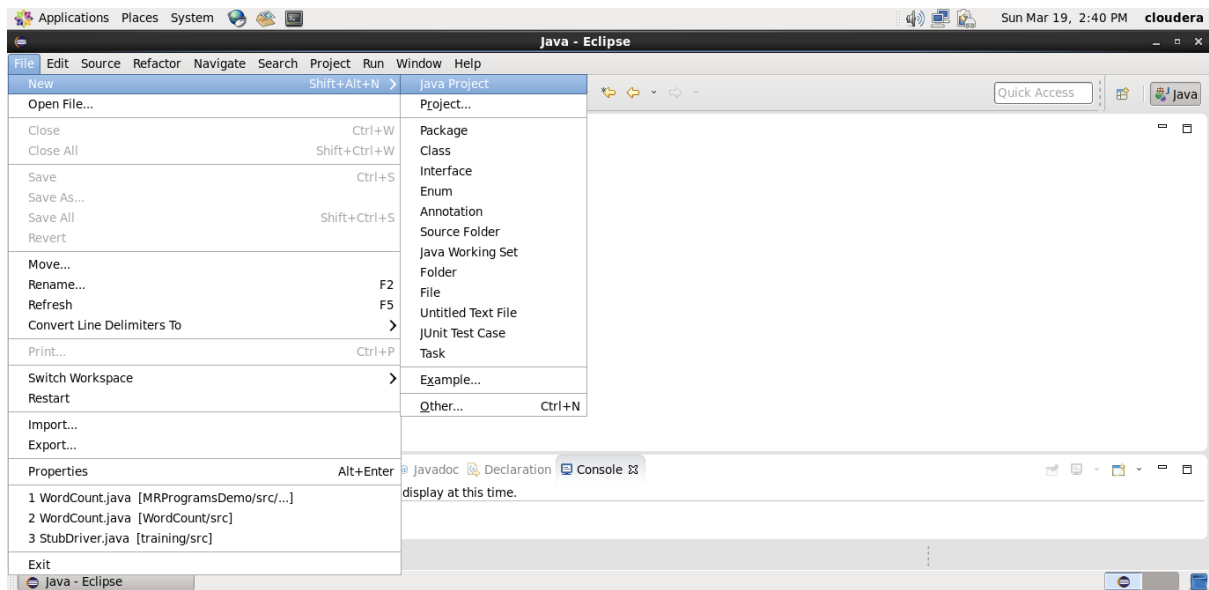


- **Map stage** : The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
- **Reduce stage** : This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

Creating WordCount.jar and exporting it:-

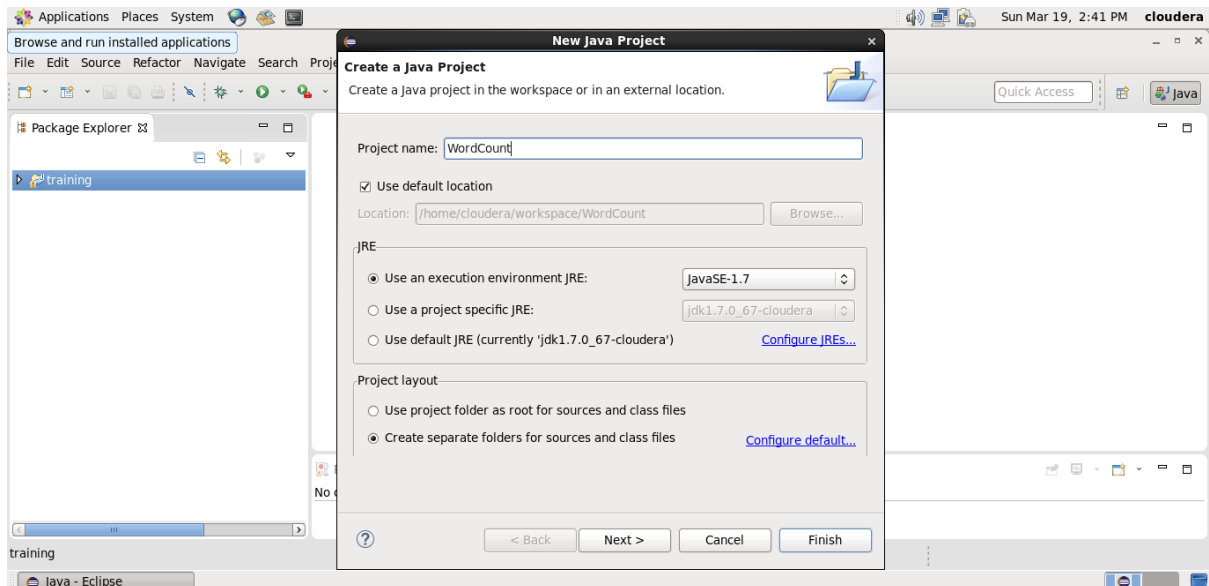
Step 1:

Open Eclipse and Click on File > New > Java Project .



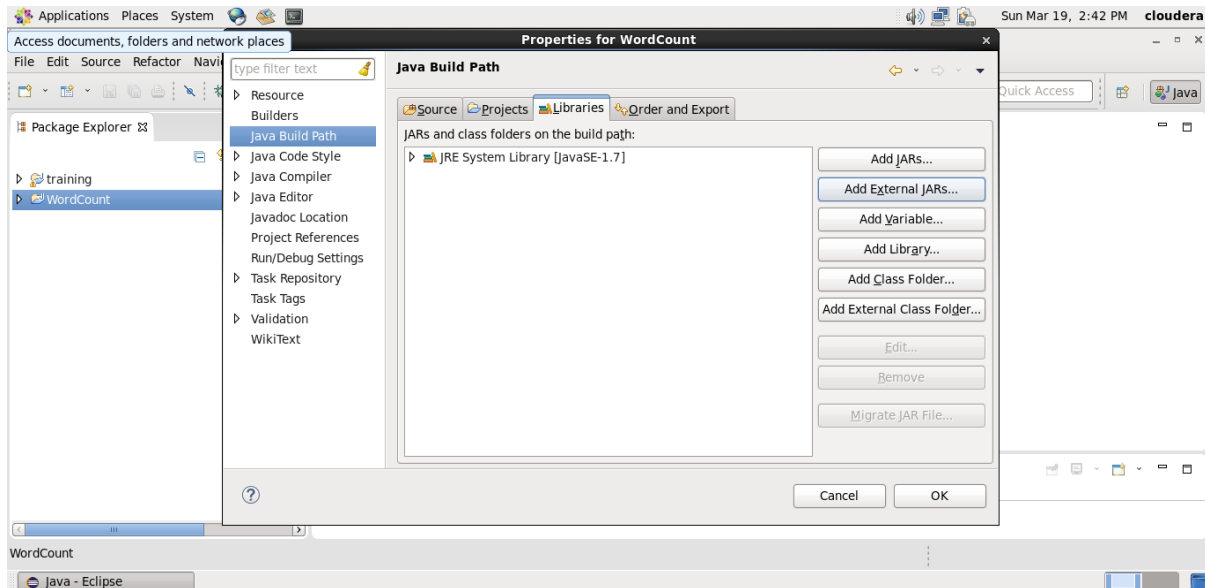
Step 2 :

Give the name 'WordCount' as your project name and click 'Finish'.



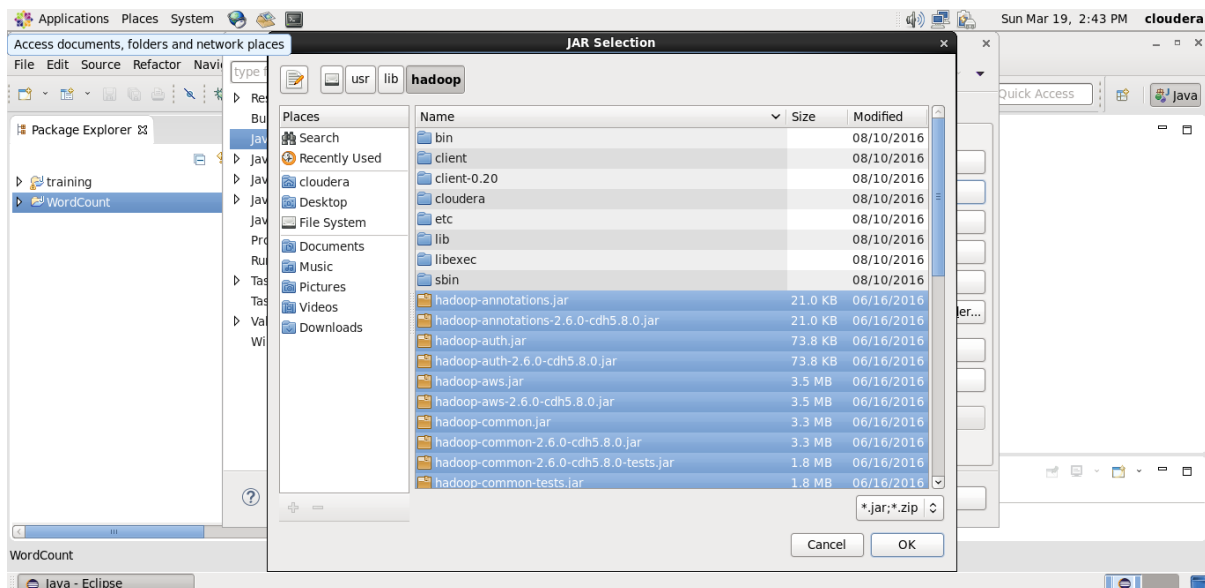
Step 3:

Right click on WordCount project and select 'Properties'. Click 'Java Build Path' and switch to Libraries tab and click on 'Add external JARs'.



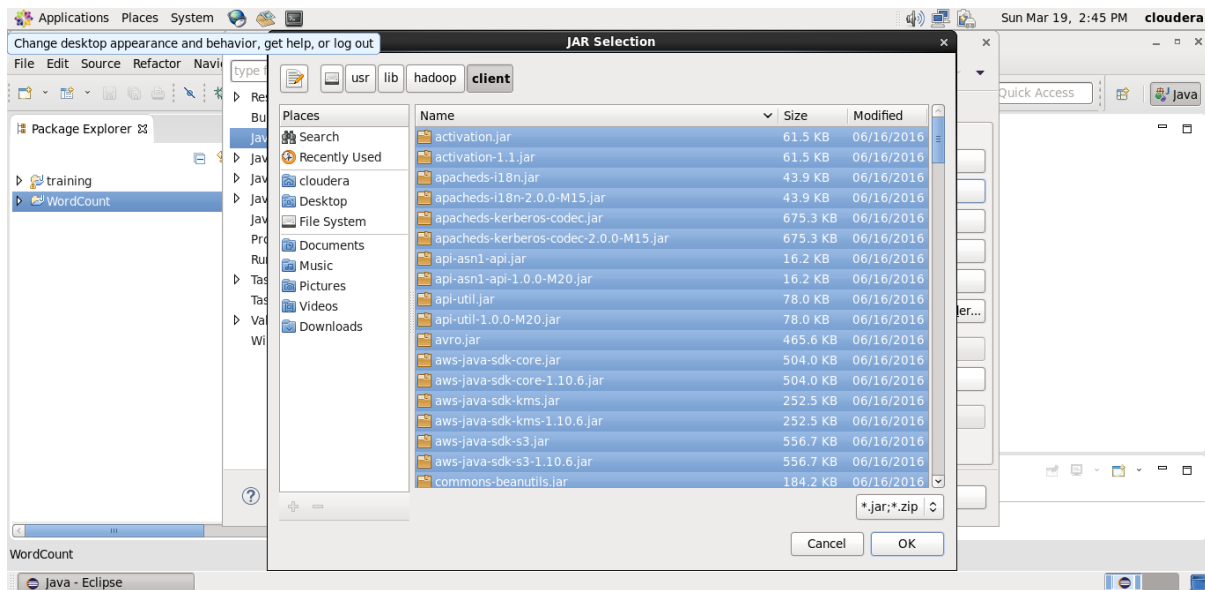
Step 4:

Select all the JAR files in usr >> lib >> hadoop directory to add them.



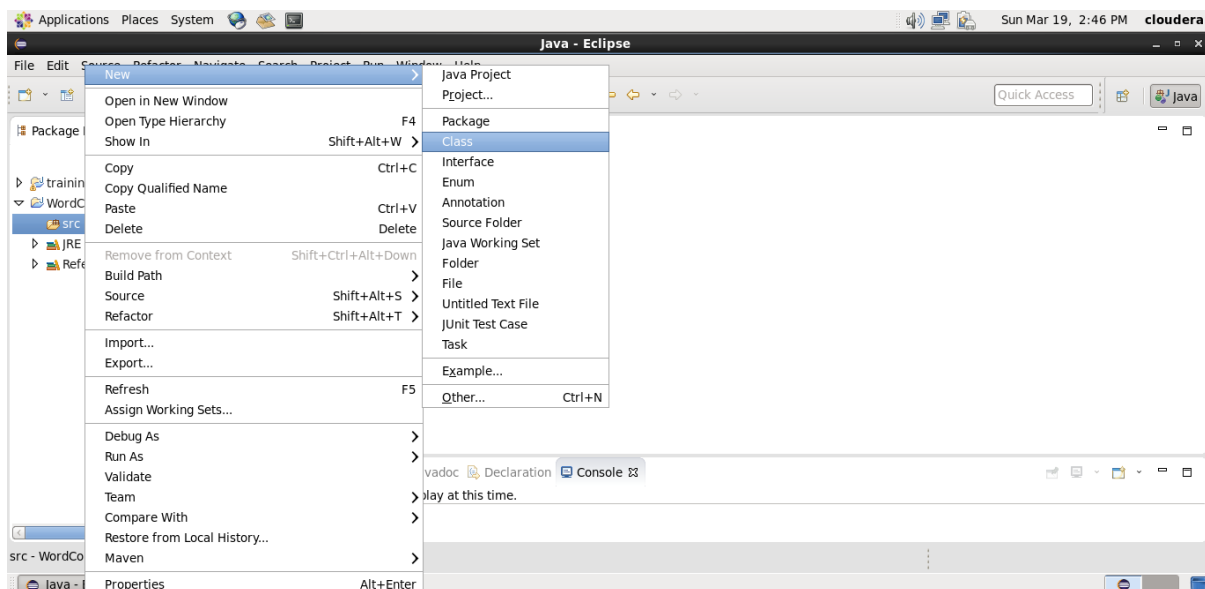
Step 5:

Again add all jar files in usr >> lib >> hadoop >> client directory and press OK.



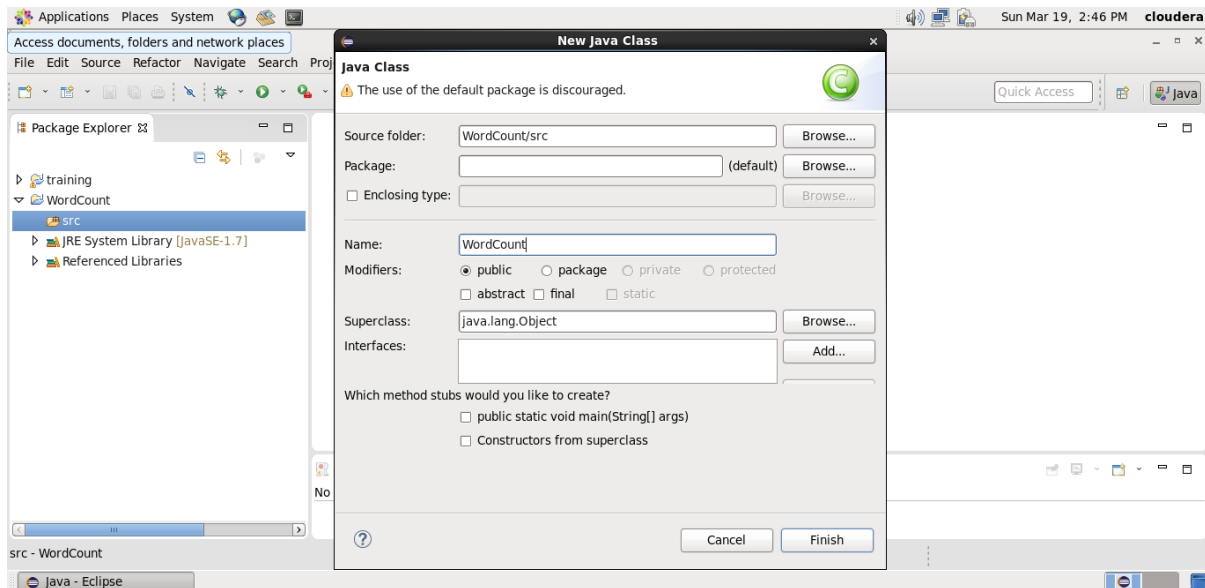
Step 6:

Right click on src, New >> Class.



Step 7:

Enter the project name as 'WordCount' and click 'Finish'.

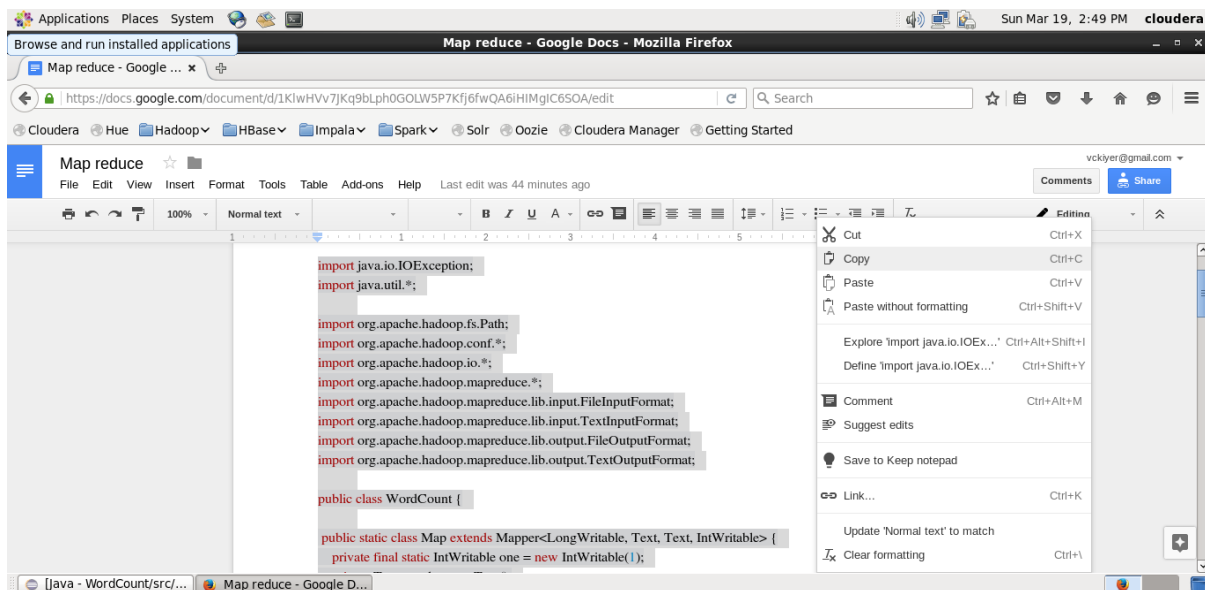


Step 8:

Open browser and copy and paste the Java Source code of map reduce program from the link given.

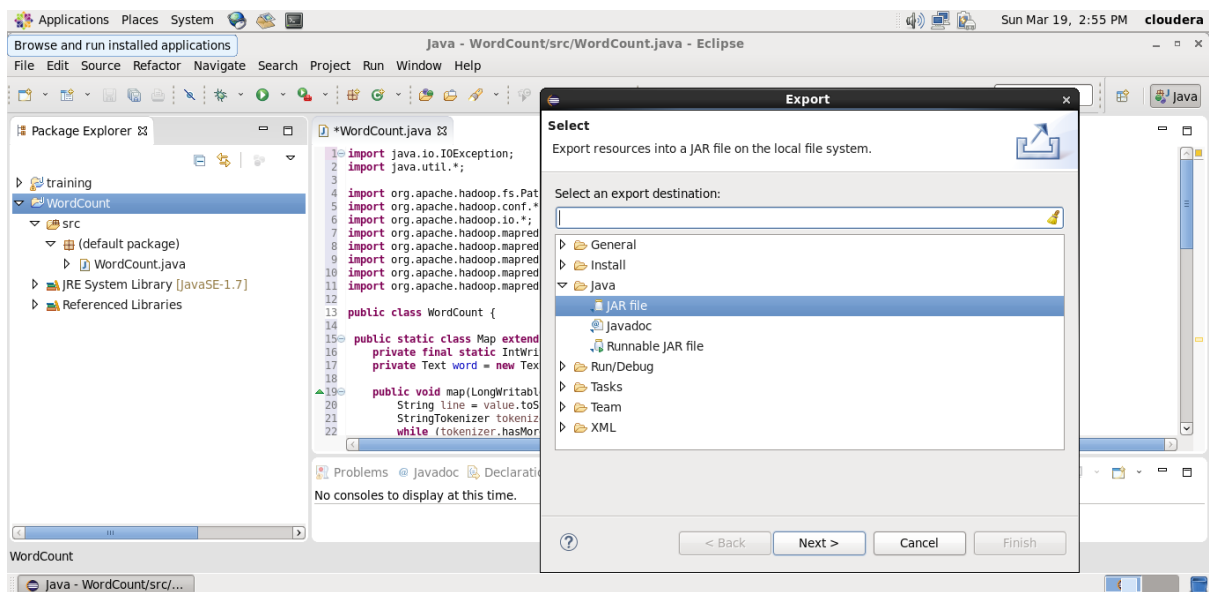
URL:

<https://docs.google.com/document/d/1KlwHVv7JKq9bLph0GOLW5P7Kfj6fwQA6iHIMgIC6SOA/edit?usp=sharing>



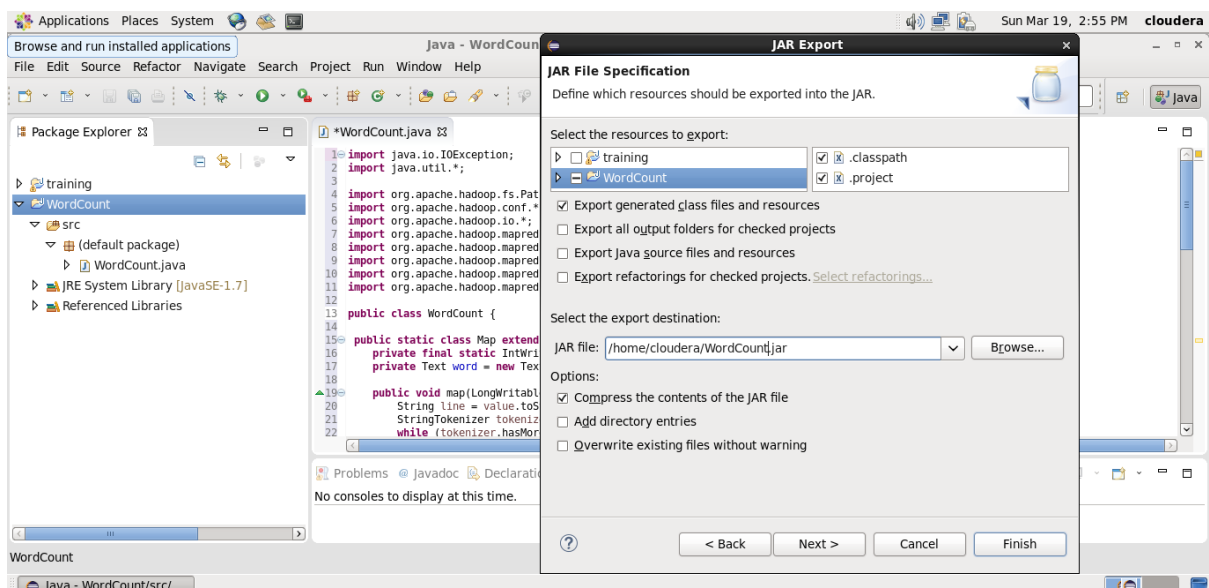
Step 9:

Right click on the WordCount project and select Export >> Java >> JAR file. Then click on 'Next'.



Step 10:

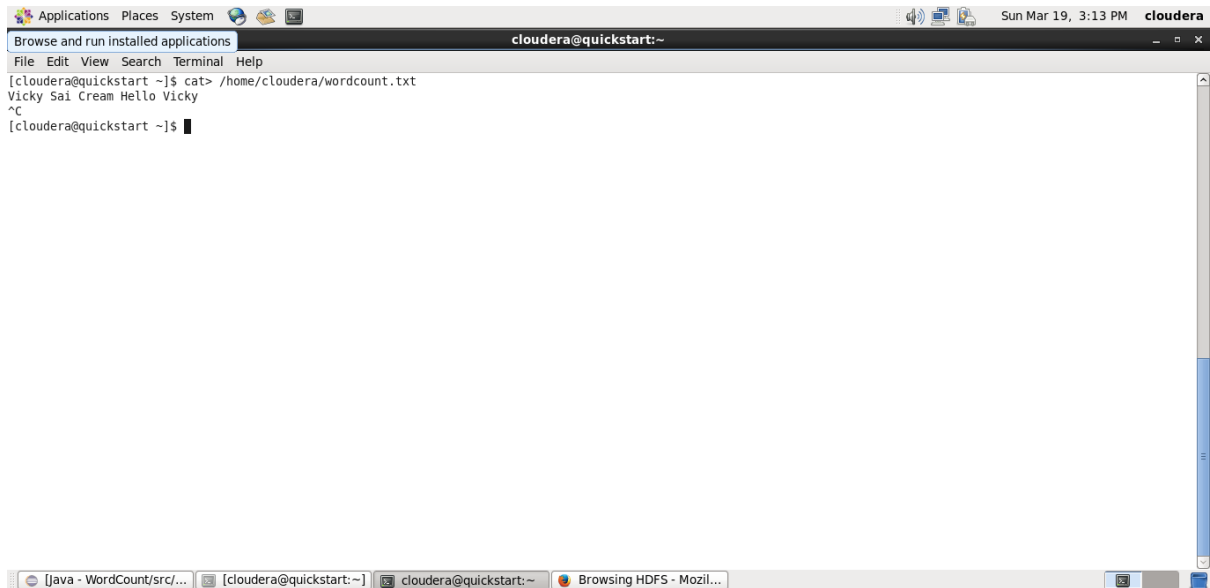
Name the JAR file and click 'Finish'.



Creating a text file for Mapreduce job to work on:

Step 11: Open a new terminal and create a normal text file .

Command: `cat> /home/cloudera/wordcount.txt`



A screenshot of a terminal window titled "cloudera@quickstart:~". The window shows the command `cat> /home/cloudera/wordcount.txt` being executed. The output of the command is displayed as follows:

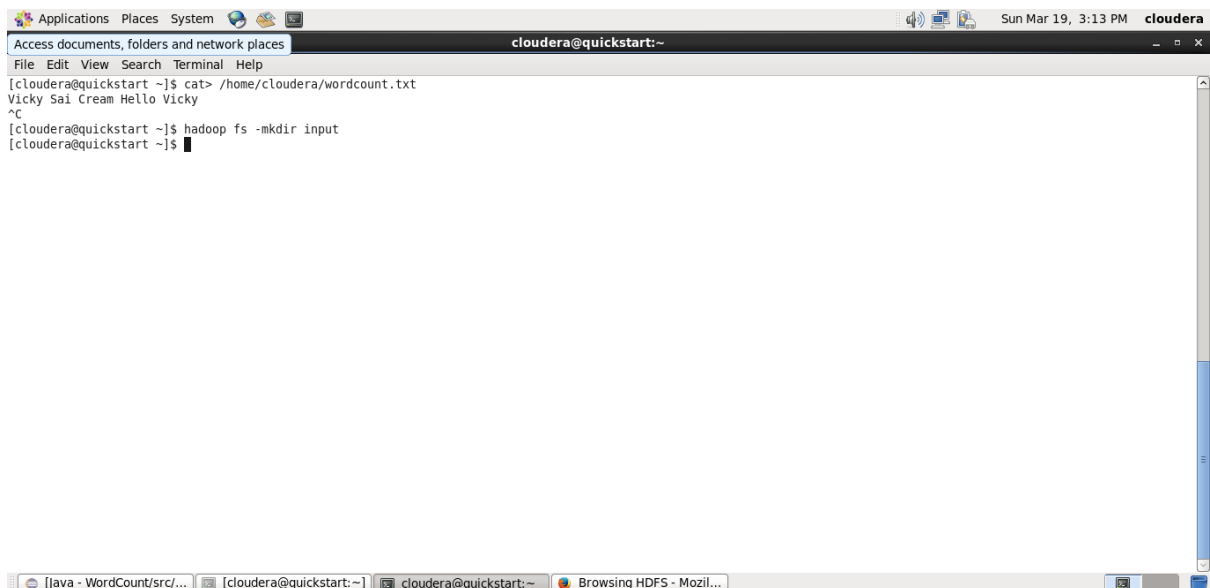
```
[cloudera@quickstart ~]$ cat> /home/cloudera/wordcount.txt
Vicky Sai Cream Hello Vicky
^C
[cloudera@quickstart ~]$
```

The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows the date and time as "Sun Mar 19, 3:13 PM" and the username "cloudera".

Step 12:

Make a new Directory using the following command.

Command: `hadoop fs -mkdir input`



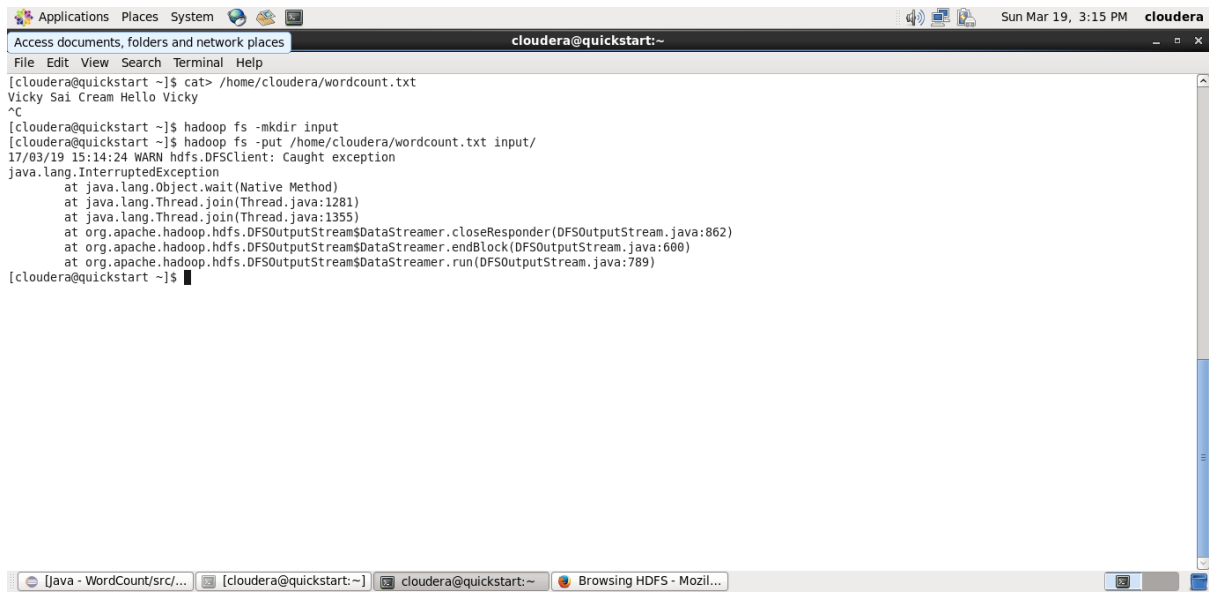
A screenshot of a terminal window titled "cloudera@quickstart:~". The window shows the command `hadoop fs -mkdir input` being executed. The output of the command is displayed as follows:

```
[cloudera@quickstart ~]$ cat> /home/cloudera/wordcount.txt
Vicky Sai Cream Hello Vicky
^C
[cloudera@quickstart ~]$ hadoop fs -mkdir input
[cloudera@quickstart ~]$
```

The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows the date and time as "Sun Mar 19, 3:13 PM" and the username "cloudera".

Step 13: Copy the created text file to the new directory created in HDFS.

Command: `hadoop fs -put /home/cloudera/wordcount.txt input/`



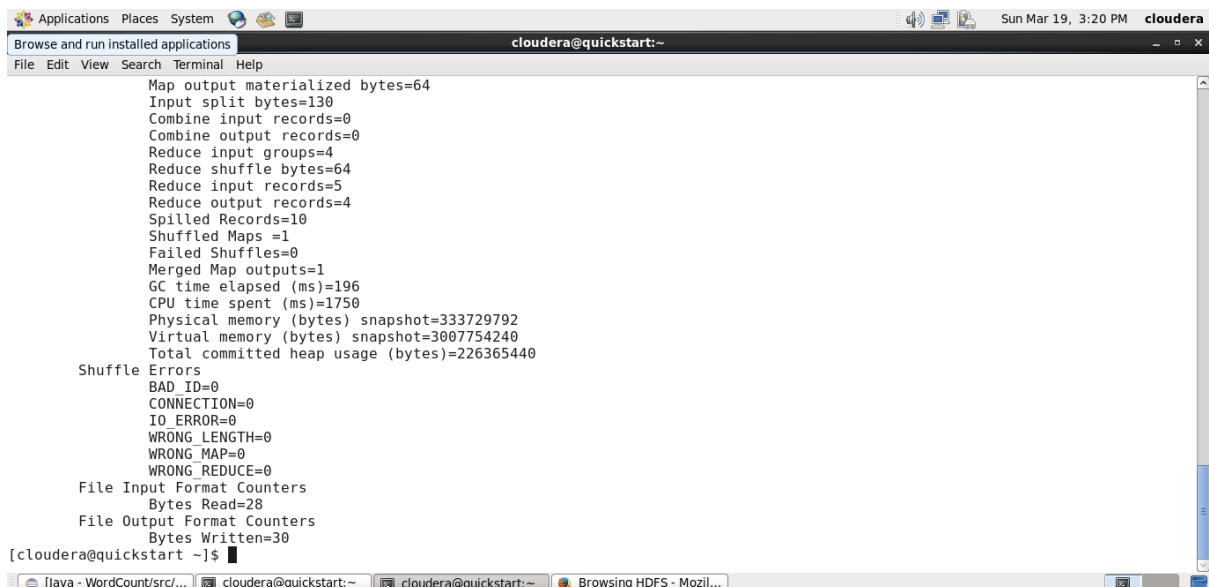
```
cloudera@quickstart:~$ cat /home/cloudera/wordcount.txt
Vicky Sai Cream Hello Vicky
^C
cloudera@quickstart:~$ hadoop fs -mkdir input
cloudera@quickstart:~$ hadoop fs -put /home/cloudera/wordcount.txt input/
17/03/19 15:14:24 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:862)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:600)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:789)
cloudera@quickstart:~$
```

Initializing mapreduce job:

Step 14:

Initialize the mapreduce job by giving the following command and wait for sometime.

Command: `hadoop jar /home/cloudera/WordCount.jar WordCount input/wordcount.txt output`



```
cloudera@quickstart:~$ hadoop jar /home/cloudera/WordCount.jar WordCount input/wordcount.txt output
Map output materialized bytes=64
Input split bytes=130
Combine input records=0
Combine output records=0
Reduce input groups=4
Reduce shuffle bytes=64
Reduce input records=5
Reduce output records=4
Spilled Records=10
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=196
CPU time spent (ms)=1750
Physical memory (bytes) snapshot=333729792
Virtual memory (bytes) snapshot=3007754240
Total committed heap usage (bytes)=226365440

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=28
File Output Format Counters
Bytes Written=30
cloudera@quickstart:~$
```

Now wait for about 50-70 seconds while the mapreduce job is being performed for the data created earlier.

Output mapreduce job:

Step 15:

The output directory of the mapreduce program is listed using the following command.

Command: `hadoop fs -ls output`

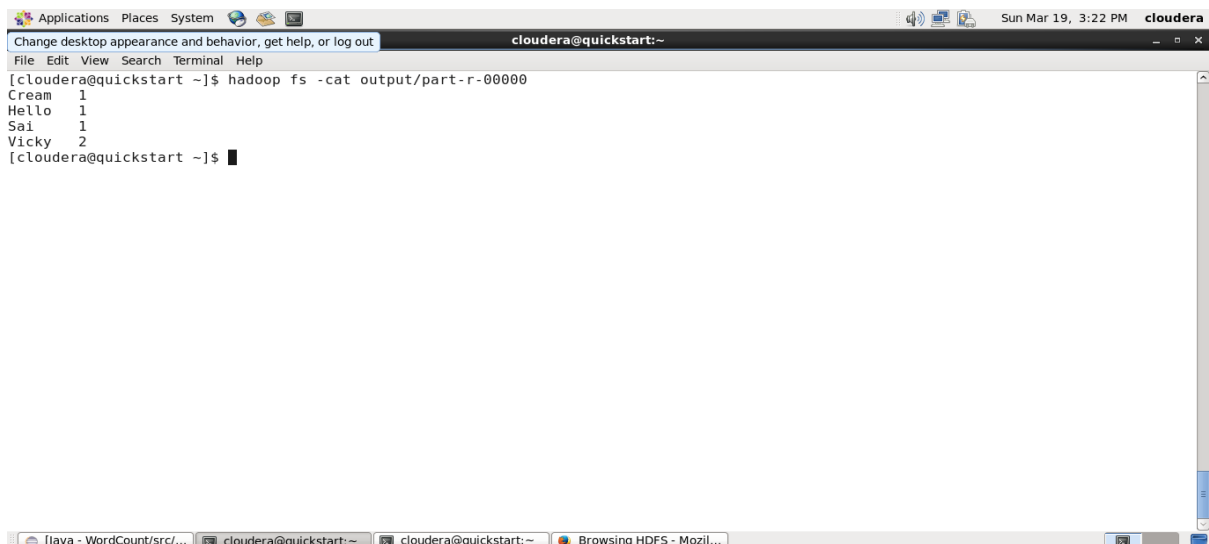


```
Applications Places System cloudera@quickstart:~
Browse and run installed applications
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hadoop fs -ls output
Found 2 items
-rw-r--r-- 1 cloudera cloudera 0 2017-03-19 15:16 output/ SUCCESS
-rw-r--r-- 1 cloudera cloudera 30 2017-03-19 15:16 output/part-r-00000
[cloudera@quickstart ~]$
```

Step 16:

The final output of the mapreduce program is found using the following command.

Command: `hadoop fs -cat output/part-r-00000`



```
Applications Places System cloudera@quickstart:~
Change desktop appearance and behavior, get help, or log out
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hadoop fs -cat output/part-r-00000
Cream 1
Hello 1
Sai 1
Vicky 2
[cloudera@quickstart ~]$
```

Conclusion:

Thus the map reduce job was successfully applied for a particular data and the number of times a word is repeated is identified.