PRACTICAL 5

MapReduce Matrix Multiplication

Matrix-vector and matrix-matrix calculations fit nicely into the MapReduce style of computing.

Suppose we have a pxq matrix M, whose element in row *i* and column *j* will be denoted m_{ij} and a qxr matrix N whose element in row j and column k is donated by n_{jk} then the product P = MN will be pxr matrix P whose element in row i and column k will be donated by p_{ik}, where $P(i,k) = m_{ij} * n_{jk}$.

Matrix Data Model for MapReduce

We represent matrix M as a relation M(I,J,V), with tuples (i,j,m_{ij}), and matrix N as a relation N(J,K,W), with tuples (j,k,n_{jk}). Most matrices are sparse so large amount of cells have value zero. When we represent matrices in this form, we do not need to keep entries for the cells that have values of zero to save large amount of disk space.

MapReduce

We will write Map and Reduce functions to process input files. Map function will produce $key, value$ pairs from the input data as it is described in Algorithm 1. Reduce function uses the output of the Map function and performs the calculations and produces $key, value$ pairs as described in Algorithm 2. All outputs are written to HDFS.

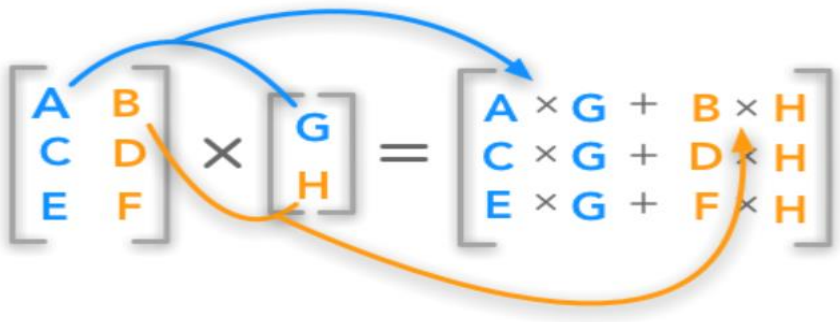
The Map Task

For matrix M, map task will produce key,value

The Reduce Task

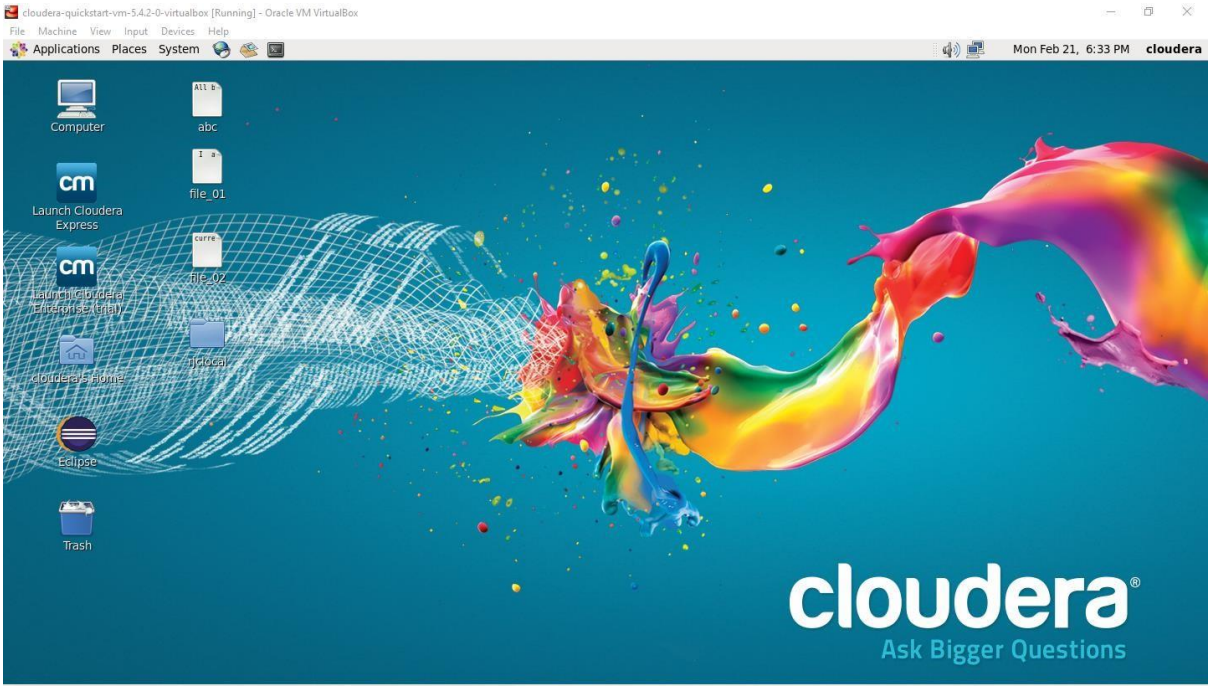
Reduce task takes the $key,value$ pairs as the input and process one key at a time. For each key it divides the

values in two separate lists for M and N.

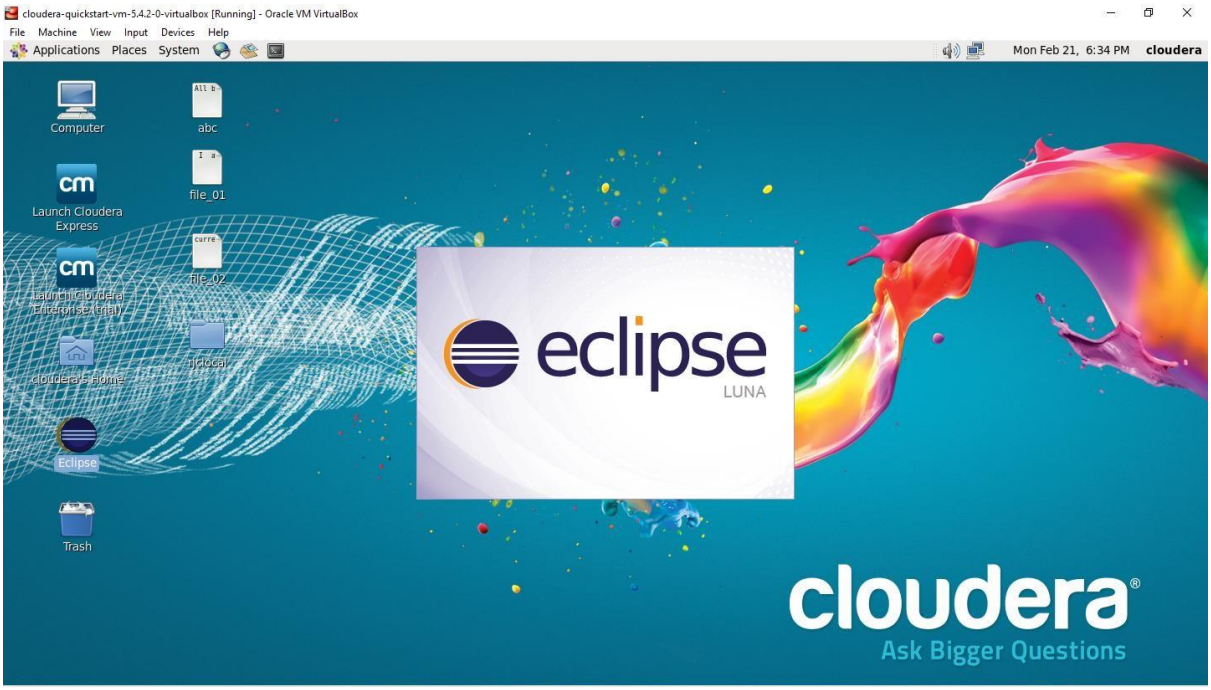


- Steps to determine maximum temperature using Hadoop MapReduce in Cloudera:

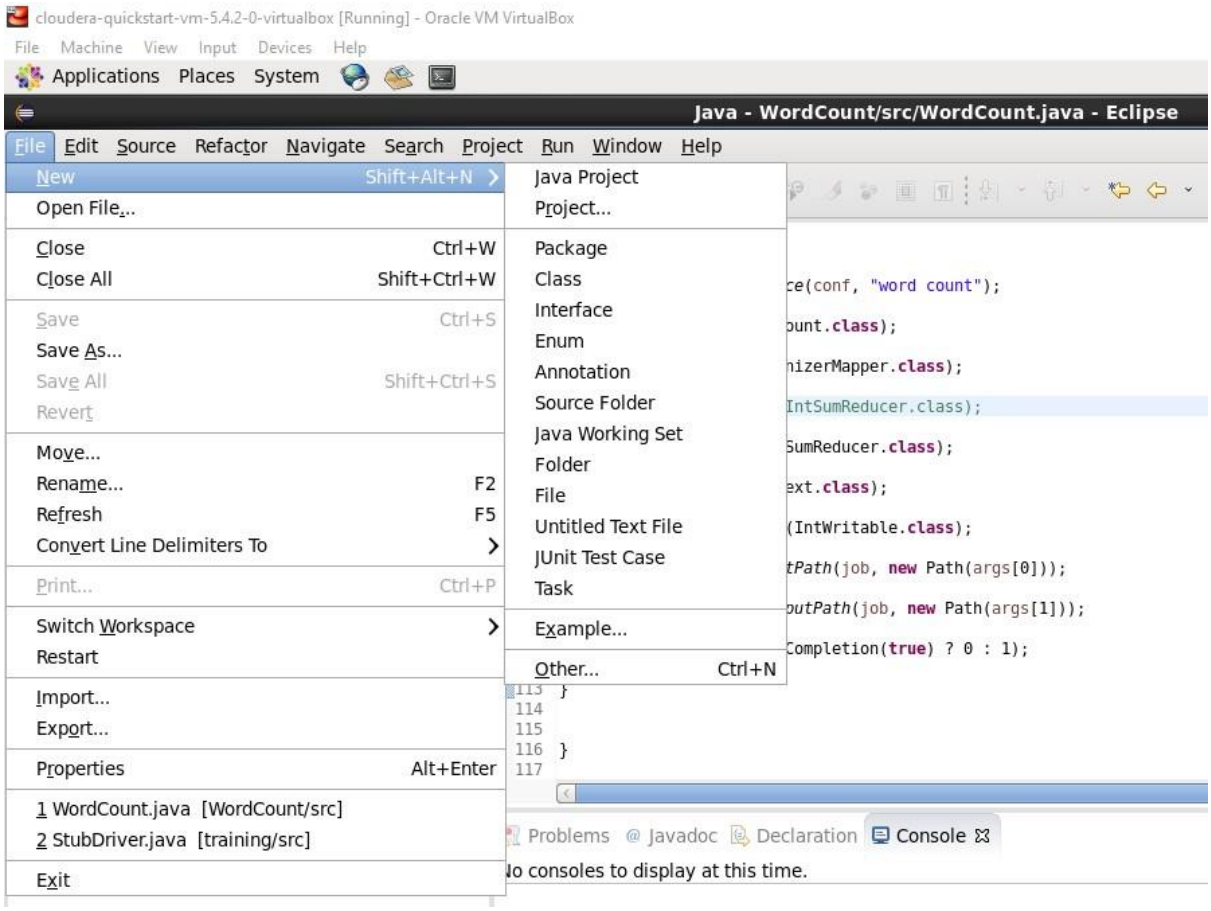
1) Open virtual box and then start cloudera quickstart

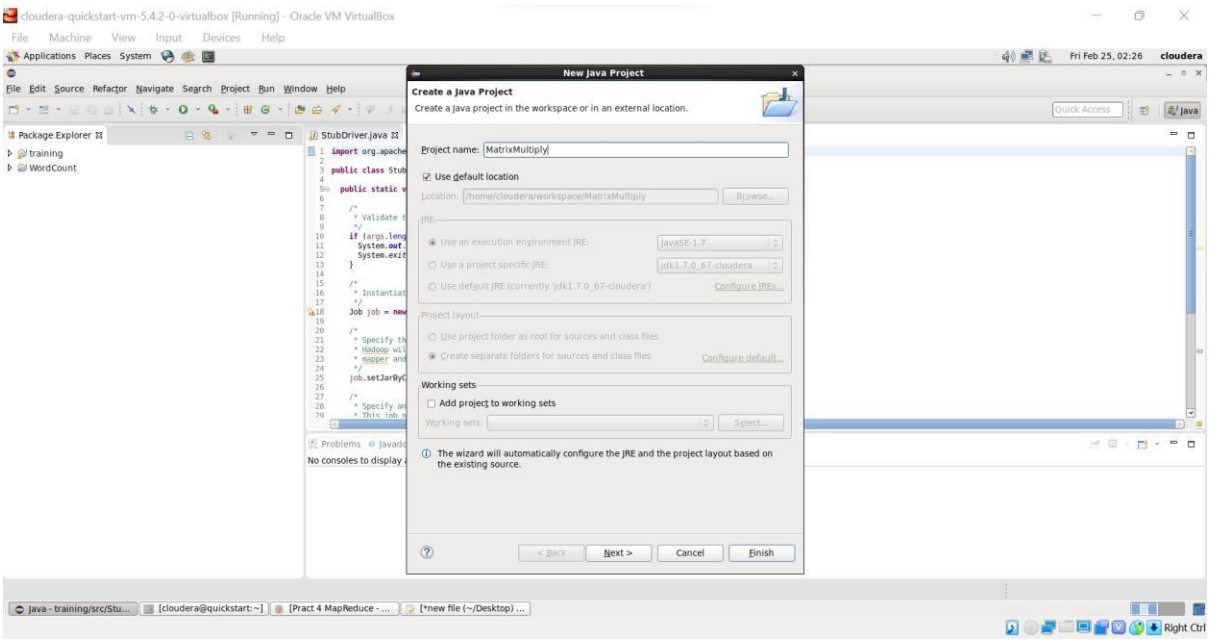


2) Open Eclipse present on the cloudera desktop



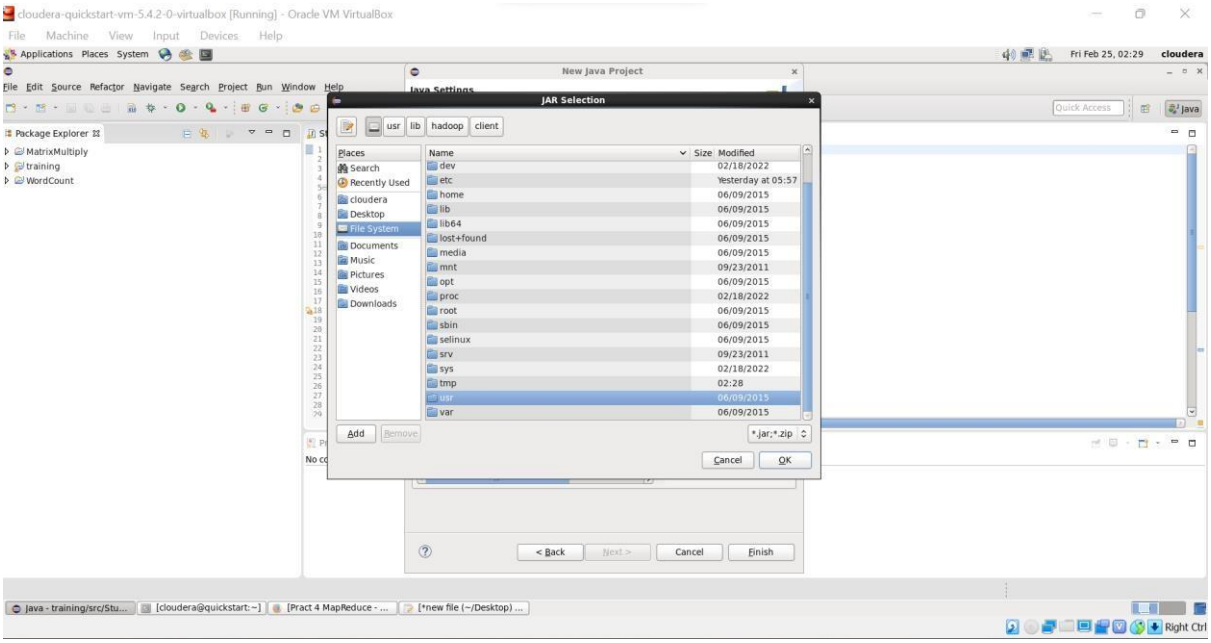
3) Create a new Java project clicking: File -> New -> Project -> Java Project -> Next (“MatrixMultiply” is the project name).

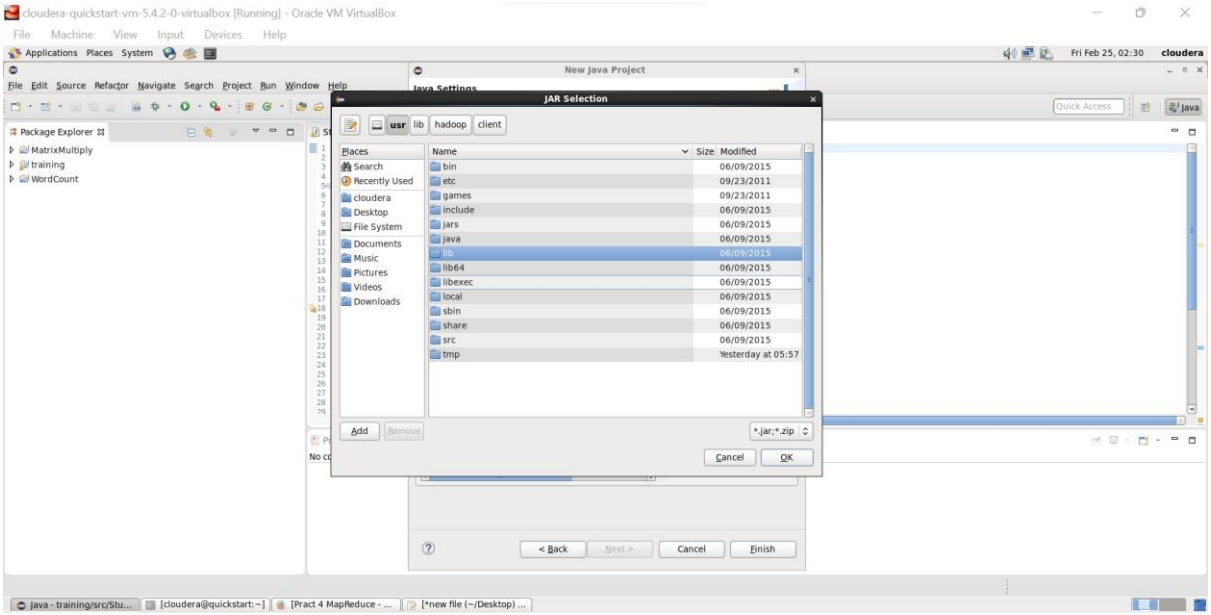


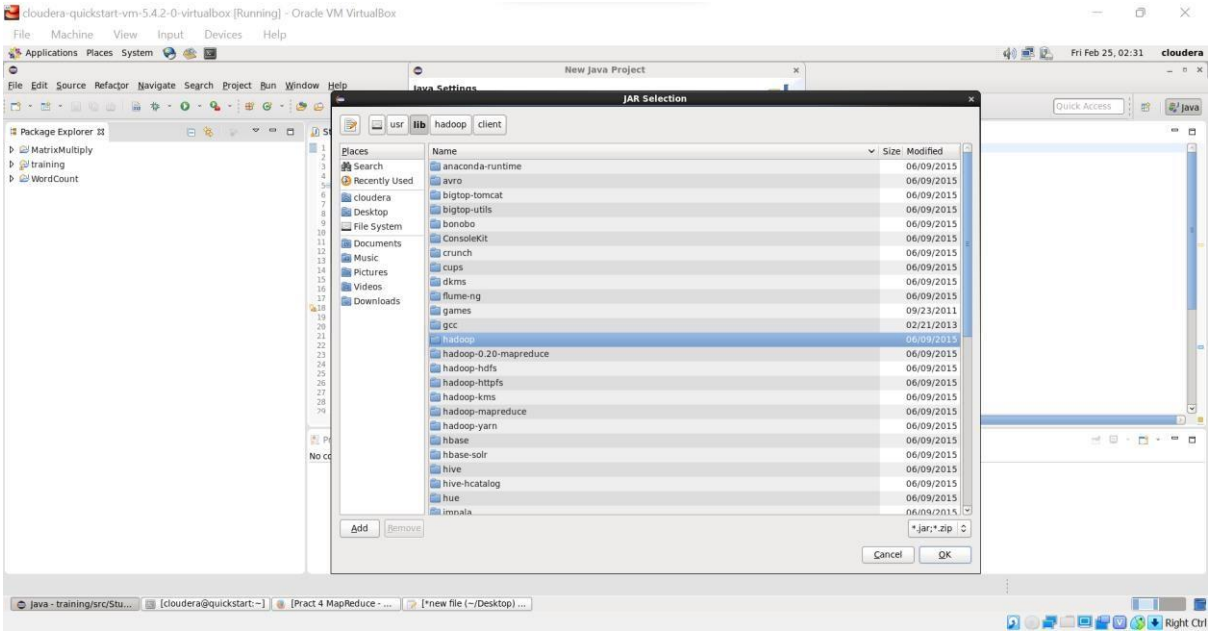


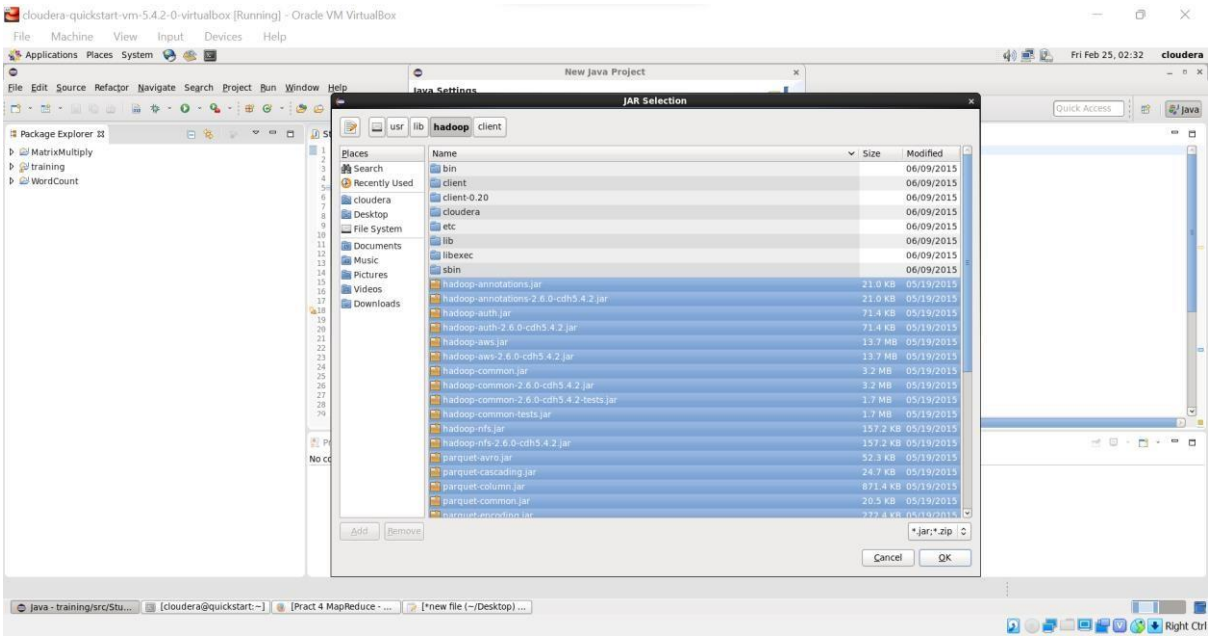
4) Adding the Hadoop libraries to the project Click on Libraries -> Add External JARs Click on File System -> usr -> lib -> hadoop Select all the libraries (JAR Files) -> click OK Click on Add

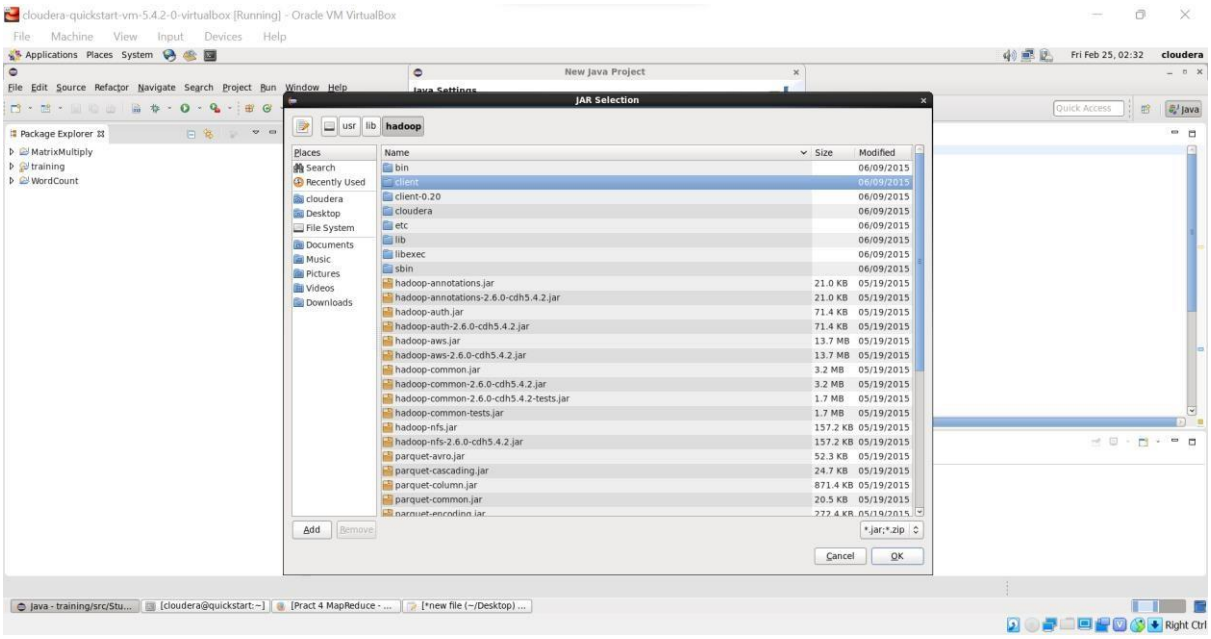
External jars, -> client -> select all jar files -> ok -> Finish

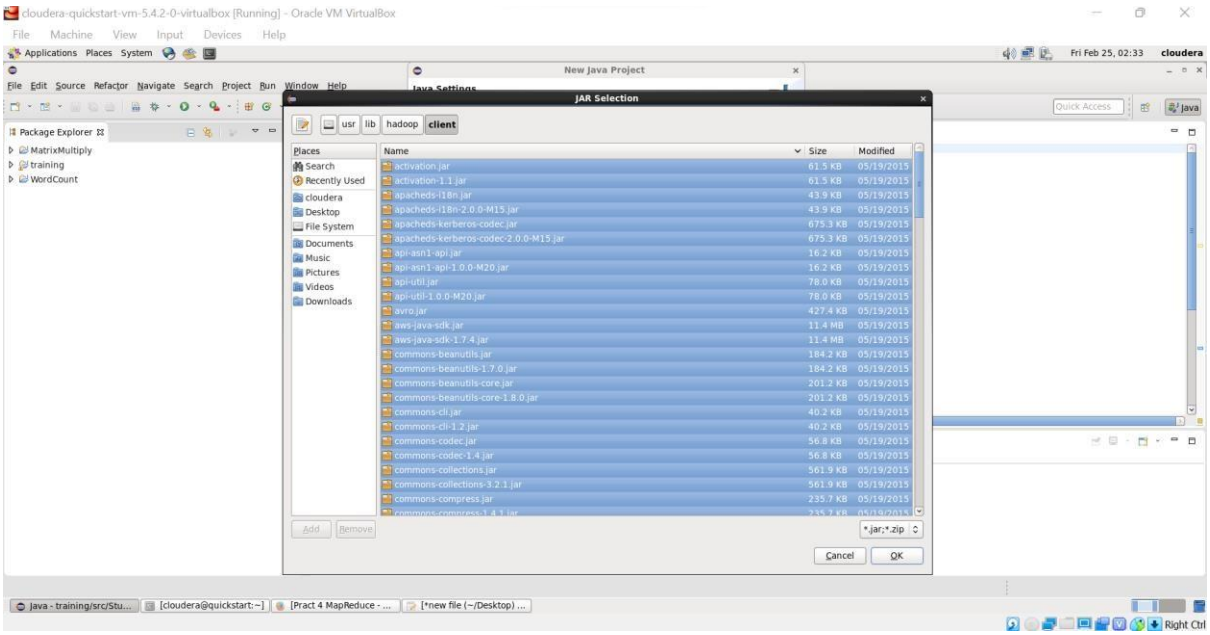




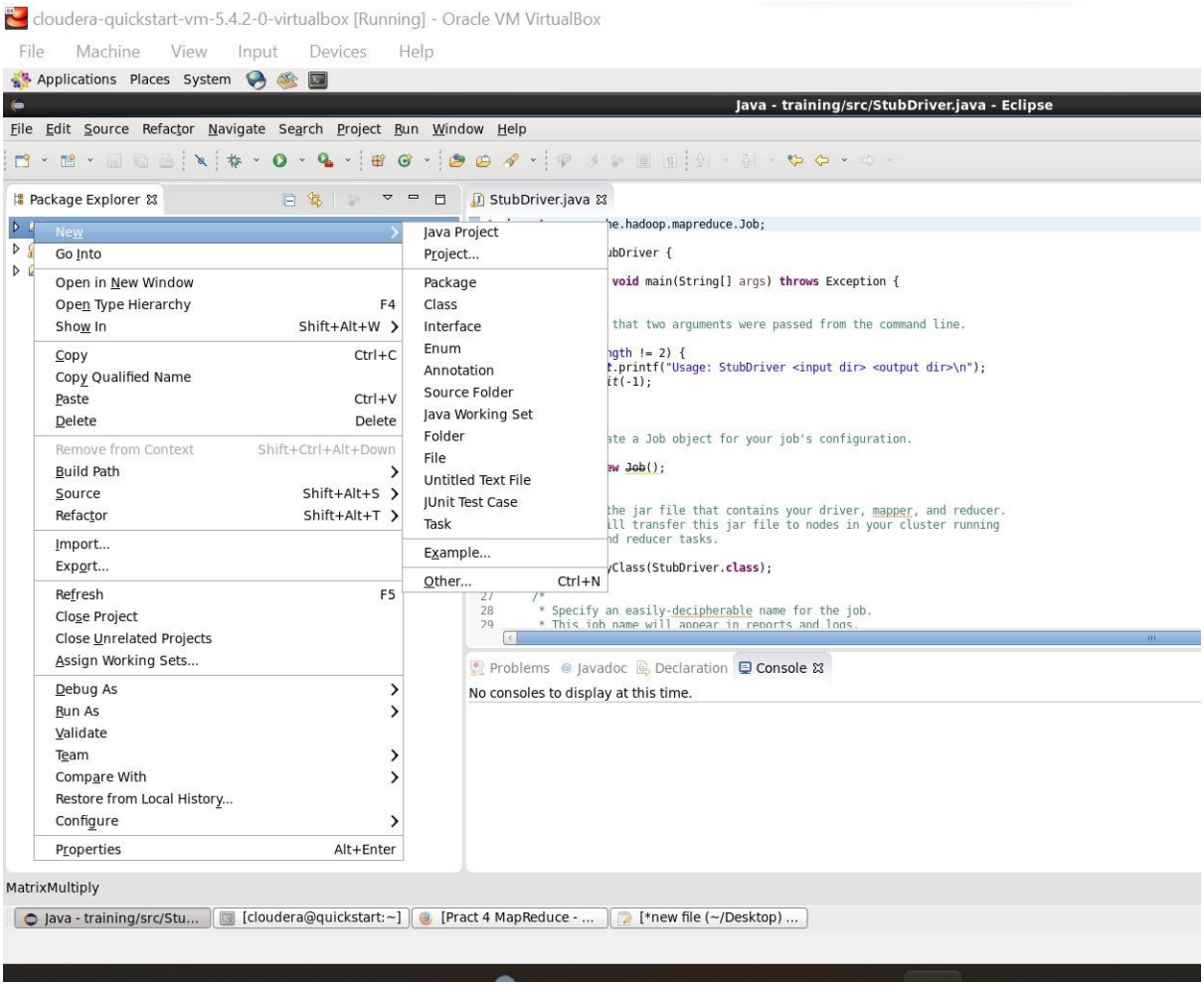

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Fri Feb 25, 02:31
cloudera
New Java Project
JAR Selection
usr lib hadoop client
Places
Name
Size
Modified
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
anaconda-runtime
avro
bigtop-tomcat
bigtop-utils
bonobo
ConsoleKit
crunch
cups
dkms
flume-ng
games
gcc
hadoop
hadoop-0.20-mapreduce
hadoop-hdfs
hadoop-httpfs
hadoop-kms
hadoop-mapreduce
hadoop-yarn
hbase
hbase-solr
hive
hive-hcatalog
hue
Add
Remove
.jar;.zip
Cancel
OK
Package Explorer
MatrixMultiply
training
WordCount
Quick Access
Java
Right Ctrl

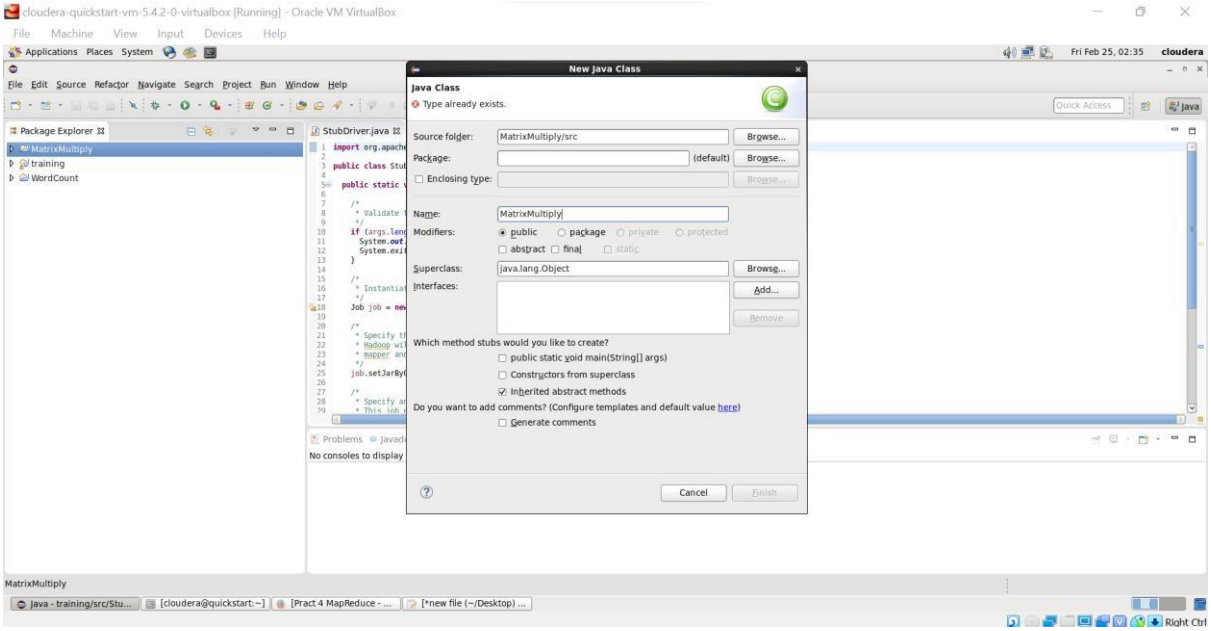

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Fri Feb 25, 02:32
cloudera
New Java Project
JAR Selection
usr lib hadoop client
Places
Name
Size
Modified
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
bin
client
client-0.20
cloudera
etc
lib
libexec
sbin
hadoop-annotations.jar
hadoop-annotations-2.6.0-cdh5.4.2.jar
hadoop-auth.jar
hadoop-auth-2.6.0-cdh5.4.2.jar
hadoop-aws.jar
hadoop-aws-2.6.0-cdh5.4.2.jar
hadoop-common.jar
hadoop-common-2.6.0-cdh5.4.2.jar
hadoop-common-2.6.0-cdh5.4.2-tests.jar
hadoop-common-tests.jar
hadoop-nfs.jar
hadoop-nfs-2.6.0-cdh5.4.2.jar
parquet-avro.jar
parquet-cascading.jar
parquet-column.jar
parquet-common.jar
Add
Remove
.jar;.zip
Cancel
OK
Package Explorer
MatrixMultiply
training
WordCount
Quick Access
Java
Right Ctrl

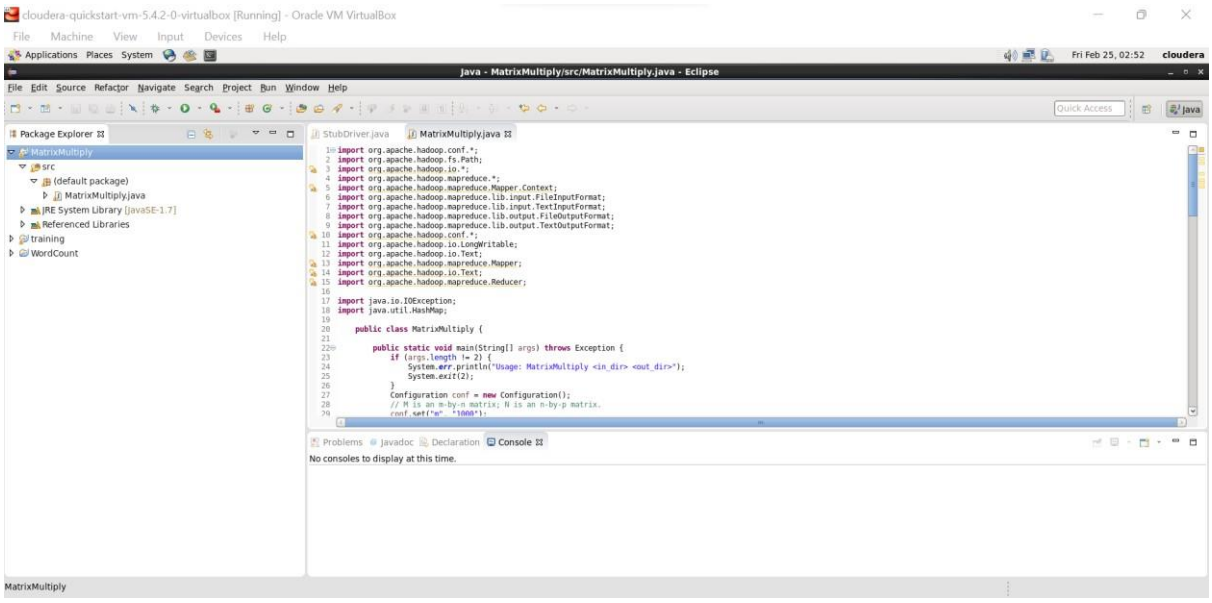




5) Right Click on the name of Project “MatrixMultiply” -> New -> class “MatrixMultiply” -> Finish Then MatrixMultiply.java window will pop up

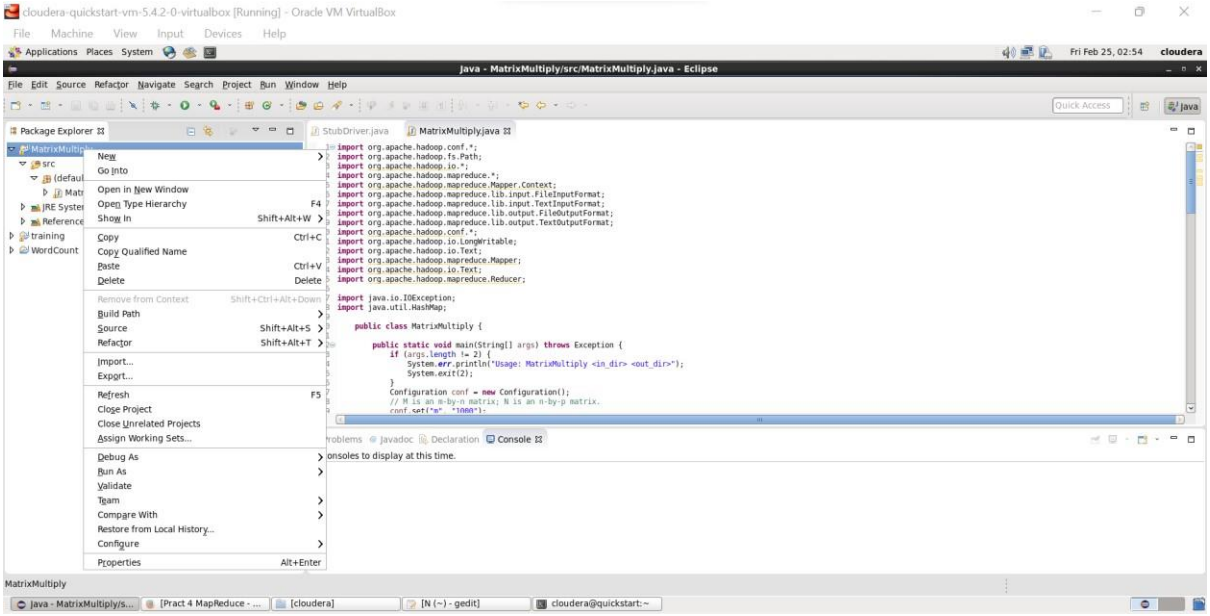

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Java - training/src/StubDriver.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
StubDriver.java
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Shift+Alt+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Shift+Ctrl+Alt+Down
Build Path
Source Shift+Alt+S
Refactor Shift+Alt+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Debug As
Run As
Validate
Team
Compare With
Restore from Local History...
Configure
Properties Alt+Enter
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Task
Example...
Other... Ctrl+N
Problems Javadoc Declaration Console
No consoles to display at this time.
MatrixMultiply
Java - training/src/Stu...
[cloudera@quickstart:~]
[Pract 4 MapReduce - ...
[*new file (~/Desktop) ...

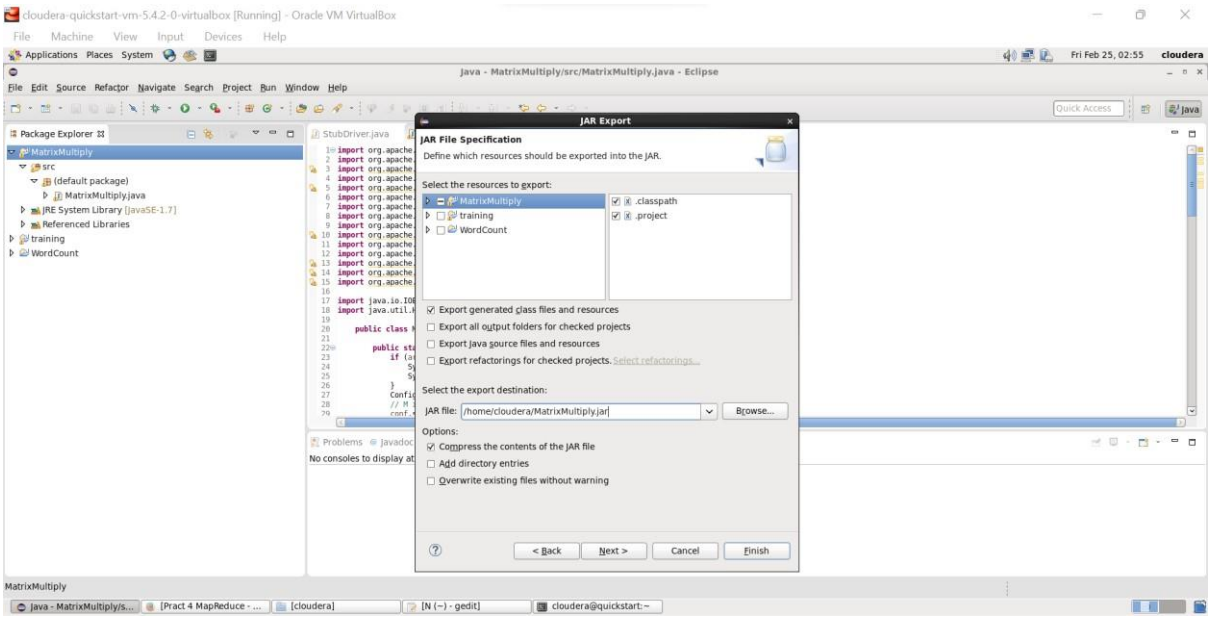

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
New Java Class
Java Class
Type already exists.
Source folder: MatrixMultiply/src
Package:
Enclosing type:
Name: MatrixMultiply
Modifiers: public package private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Cancel
Finish

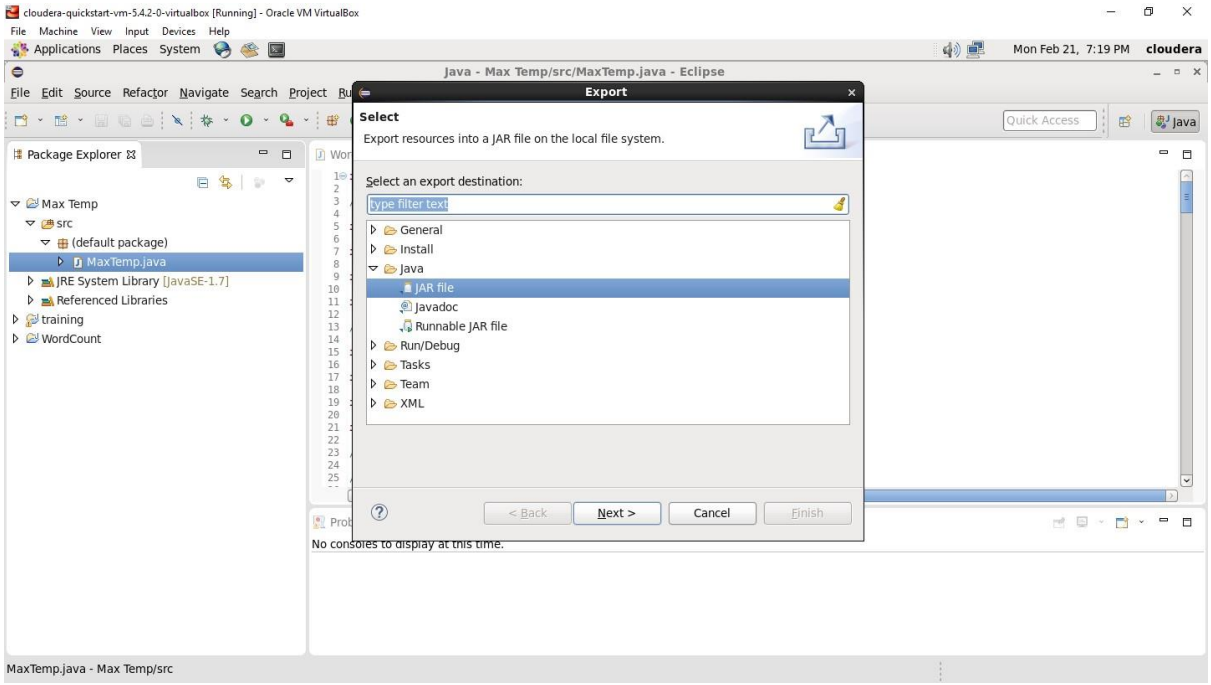


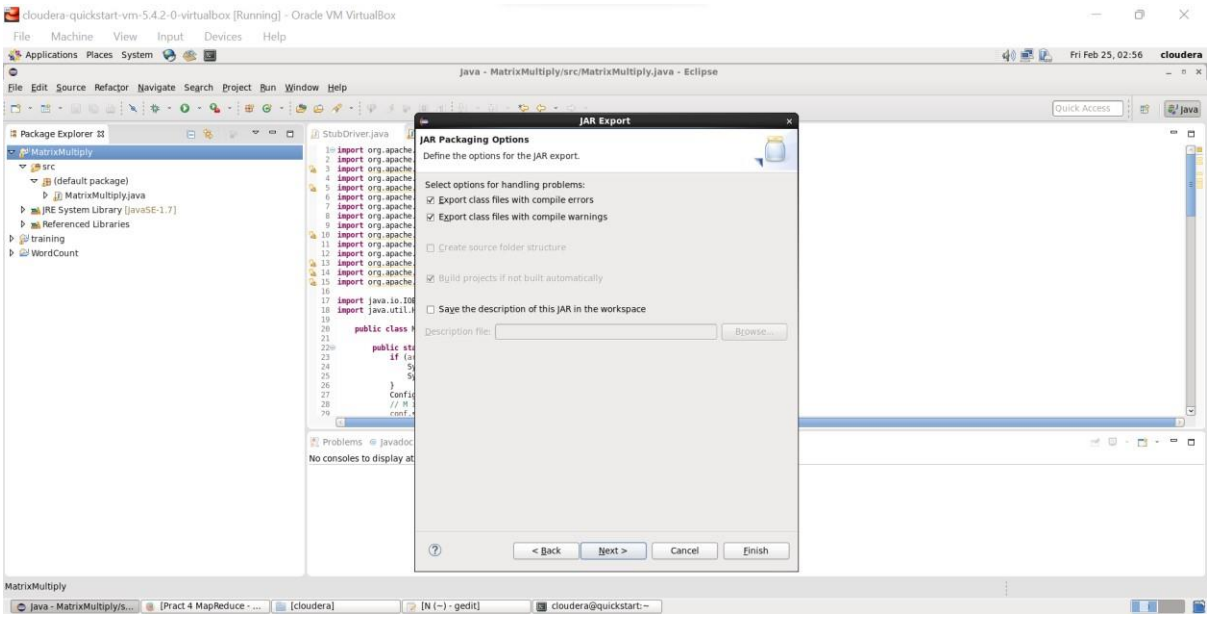
6) Right Click on the project name MatrixMultiply -> Export -> Java -> JAR

File -> Next -> for select the export destination for JAR file: browse -> Name : MatrixMultiply.jar -> save in folder -> cloudera -> Finish -> OK



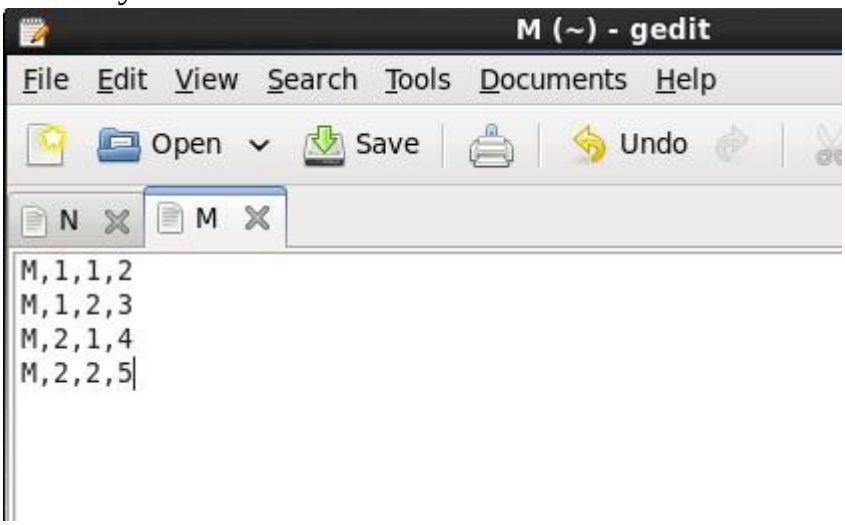

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
Java - MatrixMultiply/src/MatrixMultiply.java - Eclipse
JAR Export
JAR File Specification
Define which resources should be exported into the JAR.
Select the resources to export:
MatrixMultiply
training
WordCount
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects.
Select the export destination:
JAR file: /home/cloudera/MatrixMultiply.jar
Browse...
Options:
Compress the contents of the JAR file
Add directory entries
Overwrite existing files without warning
< Back
Next >
Cancel
Finish

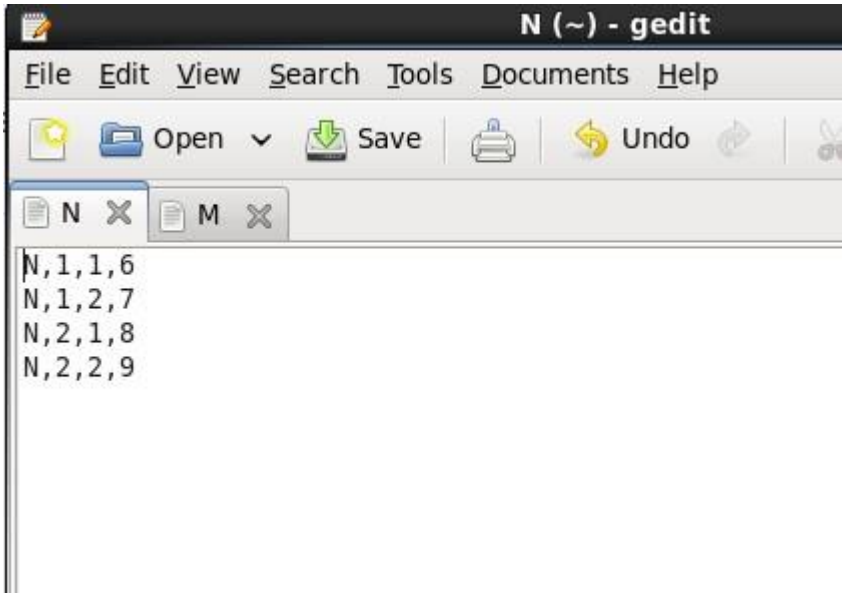

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
Java - Max Temp/src/MaxTemp.java - Eclipse
Export
Select
Export resources into a JAR file on the local file system.
Select an export destination:
type filter text
General
Install
Java
JAR file
Javadoc
Runnable JAR file
Run/Debug
Tasks
Team
XML
< Back
Next >
Cancel
Finish
Max Temp
src
(default package)
MaxTemp.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
training
WordCount
No consoles to display at this time.
MaxTemp.java - Max Temp/src



7) Open terminal and type hdfs dfs -ls/ command Here listing all the directory present in hdfs.

8) Input file named as M and N which is present on desktop i.e. in local file system.





9) Now we have to move this input file to hdfs. For this we create a direcory on hdfs using command **hdfs dfs -mkdir /testpip**.

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /testip
[cloudera@quickstart ~]$ hdfs dfs -ls  /
Found 15 items
drwxr-xr-x   - hbase    supergroup          0 2022-02-10 20:00 /hbase
drwxr-xr-x   - cloudera supergroup          0 2022-02-17 19:52 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2022-02-17 19:55 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 18:51 /outputdir1
drwxr-xr-x   - cloudera supergroup          0 2022-02-14 20:55 /rjc
drwxr-xr-x   - solr     solr                0 2015-06-09 03:38 /solr
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:41 /tempip
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:14 /tempop
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:42 /tempop1
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:04 /tempop2
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:05 /tempop3
drwxr-xr-x   - cloudera supergroup          0 2022-02-23 22:32 /testip
drwxrwxrwx   - hdfs     supergroup          0 2022-02-07 21:01 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2015-06-09 03:38 /user
drwxr-xr-x   - hdfs     supergroup          0 2015-06-09 03:36 /var
[cloudera@quickstart ~]$
```

10) Move the input file i.e. temperature to this directory created in hdfs by using either put command or copyFromLocal command.

11) Now checking whether the **"M and N"** present in /testip directory of hdfs or not using **hdfs dfs -ls /testpip** command

```
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/M /testip
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/N /testip
[cloudera@quickstart ~]$
```

12) As we can see **"M and N"** file is present in /testip directory of hdfs. Now we will see the content of this file using **hdfs dfs –cat /tempip/temperature** command

```
[cloudera@quickstart ~]$ hdfs dfs -cat /tempip/M
M,1,1,2
M,1,2,3
M,2,1,4
M,2,2,5
```

```
[cloudera@quickstart ~]$ hdfs dfs -cat /tempip/N
N,1,1,6
N,1,2,7
N,2,1,8
N,2,2,9
[cloudera@quickstart ~]$
```

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

```
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/MatrixMultiply.jar MatrixMultiply /tempip  /tempop2
22/02/24 19:56:29 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
22/02/24 19:56:30 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your applicatio
22/02/24 19:56:30 INFO input.FileInputFormat: Total input paths to process : 2
22/02/24 19:56:30 INFO mapreduce.JobSubmitter: number of splits:2
22/02/24 19:56:30 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1645544927537_0006
22/02/24 19:56:30 INFO impl.YarnClientImpl: Submitted application application_1645544927537_0006
22/02/24 19:56:30 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1645544927537_0006/
22/02/24 19:56:30 INFO mapreduce.Job: Running job: job_1645544927537_0006
22/02/24 19:56:37 INFO mapreduce.Job: Job job_1645544927537_0006 running in uber mode : false
22/02/24 19:56:37 INFO mapreduce.Job:  map 0% reduce 0%
22/02/24 19:56:47 INFO mapreduce.Job:  map 50% reduce 0%
22/02/24 19:56:48 INFO mapreduce.Job:  map 100% reduce 0%
22/02/24 19:56:54 INFO mapreduce.Job:  map 100% reduce 100%
22/02/24 19:56:54 INFO mapreduce.Job: Job job_1645544927537_0006 completed successfully
22/02/24 19:56:54 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=111126
                FILE: Number of bytes written=554795
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=274
                HDFS: Number of bytes written=36
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=14429
                Total time spent by all reduces in occupied slots (ms)=4356
                Total time spent by all map tasks (ms)=14429
                Total time spent by all reduce tasks (ms)=4356
                Total vcore-seconds taken by all map tasks=14429
                Total vcore-seconds taken by all reduce tasks=4356
                Total megabyte-seconds taken by all map tasks=14775296
                Total megabyte-seconds taken by all reduce tasks=4460544
        Map-Reduce Framework
                Map input records=8
                Map output records=8000
                Map output bytes=95120
                Map output materialized bytes=111132
                Input split bytes=210
                Combine input records=0
                Combine output records=0
                Reduce input groups=3996
                Reduce shuffle bytes=111132
```

```
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=14429
                Total time spent by all reduces in occupied slots (ms)=4356
                Total time spent by all map tasks (ms)=14429
                Total time spent by all reduce tasks (ms)=4356
                Total vcore-seconds taken by all map tasks=14429
                Total vcore-seconds taken by all reduce tasks=4356
                Total megabyte-seconds taken by all map tasks=14775296
                Total megabyte-seconds taken by all reduce tasks=4460544
        Map-Reduce Framework
                Map input records=8
                Map output records=8000
                Map output bytes=95120
                Map output materialized bytes=111132
                Input split bytes=210
                Combine input records=0
                Combine output records=0
                Reduce input groups=3996
                Reduce shuffle bytes=111132
                Reduce input records=8000
                Reduce output records=4
                Spilled Records=16000
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=170
                CPU time spent (ms)=1970
                Physical memory (bytes) snapshot=561012736
                Virtual memory (bytes) snapshot=4507758592
                Total committed heap usage (bytes)=391979008
        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=64
        File Output Format Counters
                Bytes Written=36
```

```
[cloudera@quickstart ~]$ hdfs dfs -cat /tempop2/part-r-00000
1,1,36.0
1,2,41.0
2,1,64.0
2,2,73.0
```

14) Then we can verify the content of tempop2 directory and in that part-r file has the actual output by using the command Hdfs dfs -cat /tempop2/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 tempop1 directory is created in hdfs or not using command **hdfs dfs -ls /**

15) Now let's check what we have inside this **tempop2** directory using command as **hdfs dfs -ls /tempop2**

16) Now we want to read the content of the **part-r-00000** file which present inside the **tempop2** using command **hdfs dfs -cat /tempop2/part-r-00000**

18) Browse the Directory by
Hadoop->HDFS Namenode->Ultilities ->Browse the file system
And thendownloading the **part-r-00000** file**.**