



School of Computer Engineering and Technology

Course : PE1 Big Data Analytics

Year : TY BTech Tri:IX

Lab Assignment 4

Problem Statement : Implement Map-reduce operation in Hadoop

Demo Program : Word Count Application

Pre-requisite : Apache Hadoop should be already installed.

Map Reduce Program for counting the occurrences of words in a text document available on HDFS.

STEPS :

1. Create the input source file and store it in a drive in local file system.

Example: E:\input.txt

Example :

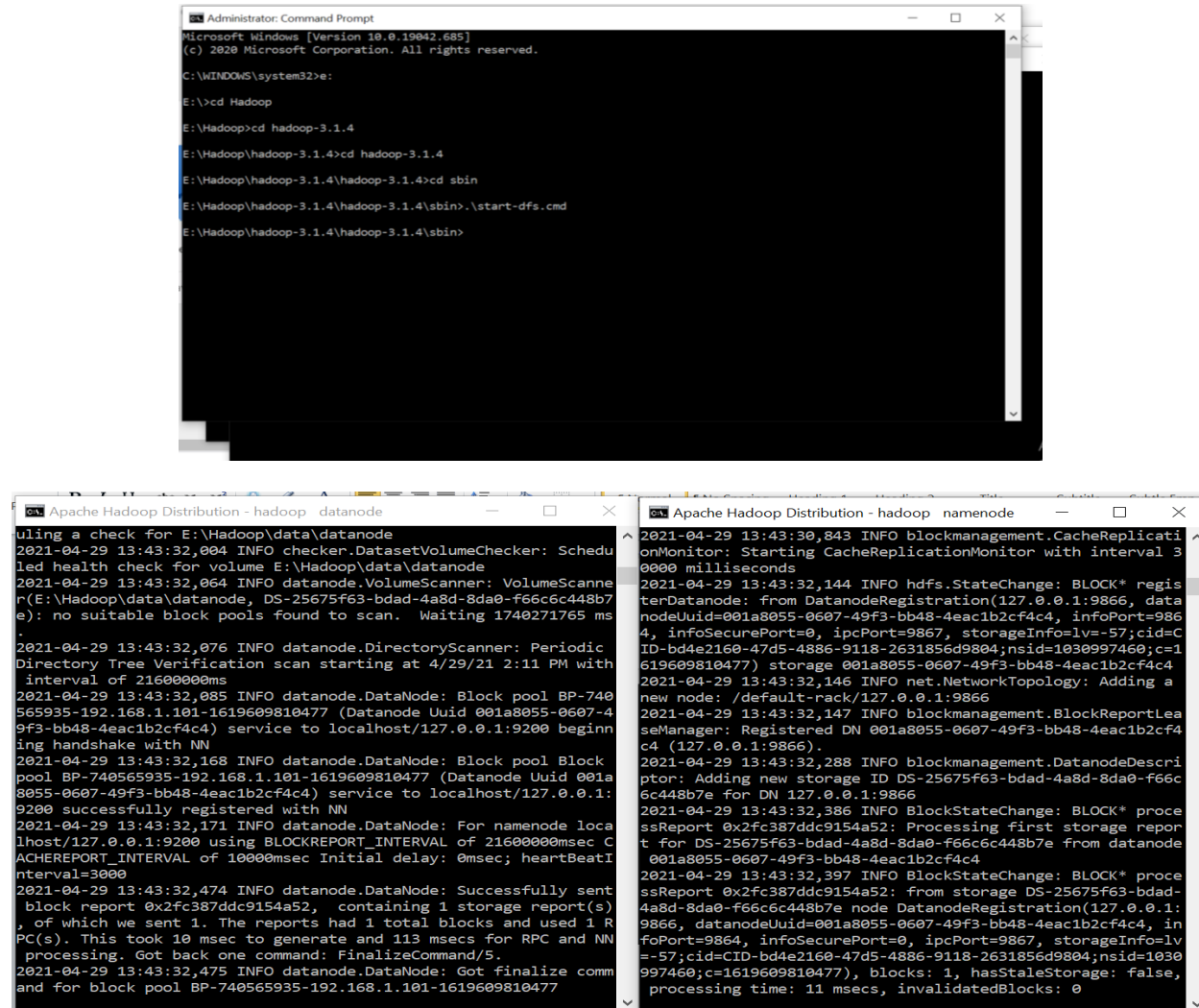
```
input - Notepad
File Edit Format View Help
The course name is Big Data Analytics.
The course focuses on analysing Big Data to find insights from the data.
Big Data sources are IoT,Social Media,E-Commerce etc.
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

2. Start the Hadoop File system Service using following command from command prompt.

Go to the sbin folder path in Hadoop :

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin and type following command to start the namenode and datanode processes :

.\start-dfs.cmd



The first screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The user navigates to the Hadoop sbin directory and runs the command `.\start-dfs.cmd`. The second screenshot shows two separate command prompt windows displaying the logs for the `datanode` and `namenode` processes. The `datanode` logs show the process starting, performing a health check, and successfully sending a block report. The `namenode` logs show the process starting, registering the datanode, and successfully processing the block report.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>e:
E:\>cd Hadoop
E:\Hadoop>cd hadoop-3.1.4
E:\Hadoop\hadoop-3.1.4>cd hadoop-3.1.4
E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4>cd sbin
E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin>.\start-dfs.cmd
E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin>
```

```
Apache Hadoop Distribution - hadoop datanode
2021-04-29 13:43:32,084 INFO checker.DatasetVolumeChecker: Scheduled health check for volume E:\Hadoop\data\datanode
2021-04-29 13:43:32,064 INFO datanode.VolumeScanner: VolumeScanner(E:\Hadoop\data\datanode, DS-25675f63-bdad-4a8d-8da0-f66c6c448b7e): no suitable block pools found to scan. Waiting 1740271765 ms
2021-04-29 13:43:32,076 INFO datanode.DirectoryScanner: Periodic Directory Tree Verification scan starting at 4/29/21 2:11 PM with interval of 2160000ms
2021-04-29 13:43:32,085 INFO datanode.DataNode: Block pool BP-740565935-192.168.1.101-1619609810477 (Datanode Uuid 001a8055-0607-49f3-bb48-4eac1b2cf4c4) service to localhost/127.0.0.1:9200 beginning handshake with NN
2021-04-29 13:43:32,168 INFO datanode.DataNode: Block pool Block pool BP-740565935-192.168.1.101-1619609810477 (Datanode Uuid 001a8055-0607-49f3-bb48-4eac1b2cf4c4) service to localhost/127.0.0.1:9200 successfully registered with NN
2021-04-29 13:43:32,171 INFO datanode.DataNode: For namenode localhost/127.0.0.1:9200 using BLOCKREPORT_INTERVAL of 2160000msec CACHEREPORT_INTERVAL of 10000msec Initial delay: 0msec; heartBeatInterval=3000
2021-04-29 13:43:32,474 INFO datanode.DataNode: Successfully sent block report 0x2fc387ddc9154a52, containing 1 storage report(s), of which we sent 1. The reports had 1 total blocks and used 1 RPC(s). This took 10 msec to generate and 113 msec for RPC and NN processing. Got back one command: FinalizeCommand/S.
2021-04-29 13:43:32,475 INFO datanode.DataNode: Got finalize command for block pool BP-740565935-192.168.1.101-1619609810477

Apache Hadoop Distribution - hadoop namenode
2021-04-29 13:43:30,843 INFO blockmanagement.CacheReplicationMonitor: Starting CacheReplicationMonitor with interval 30000 milliseconds
2021-04-29 13:43:32,144 INFO hdfs.StateChange: BLOCK* registerDatanode: from DatanodeRegistration(127.0.0.1:9866, datanodeUuid=001a8055-0607-49f3-bb48-4eac1b2cf4c4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-bd4e2160-47d5-4886-9118-2631856d9804;nsid=1030997460;c=1619609810477) storage 001a8055-0607-49f3-bb48-4eac1b2cf4c4
2021-04-29 13:43:32,146 INFO net.NetworkTopology: Adding a new node: /default-rack/127.0.0.1:9866
2021-04-29 13:43:32,147 INFO blockmanagement.BlockReportLeaseManager: Registered DN 001a8055-0607-49f3-bb48-4eac1b2cf4c4 (127.0.0.1:9866).
2021-04-29 13:43:32,288 INFO blockmanagement.DatanodeDescriptor: Adding new storage ID DS-25675f63-bdad-4a8d-8da0-f66c6c448b7e for DN 127.0.0.1:9866
2021-04-29 13:43:32,386 INFO BlockStateChange: BLOCK* processReport 0x2fc387ddc9154a52: Processing first storage report for DS-25675f63-bdad-4a8d-8da0-f66c6c448b7e from datanode 001a8055-0607-49f3-bb48-4eac1b2cf4c4
2021-04-29 13:43:32,397 INFO BlockStateChange: BLOCK* processReport 0x2fc387ddc9154a52: from storage DS-25675f63-bdad-4a8d-8da0-f66c6c448b7e node DatanodeRegistration(127.0.0.1:9866, datanodeUuid=001a8055-0607-49f3-bb48-4eac1b2cf4c4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-bd4e2160-47d5-4886-9118-2631856d9804;nsid=1030997460;c=1619609810477), blocks: 1, hasStaleStorage: false, processing time: 11 msec, invalidatedBlocks: 0
```

Once the namenode and datanode processes have successfully started,

3. **Start the yarn service using following command:**

.\start-yarn.cmd

This command will start execution of two processes: ResourceManager and NodeManager respectively.

```
Administrator: Command Prompt - \start-yarn.cmd
Microsoft Windows [Version 10.0.19042.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>e:

E:\>cd Hadoop

E:\Hadoop>cd hadoop-3.1.4

E:\Hadoop\hadoop-3.1.4>cd hadoop-3.1.4

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4>cd sbin

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin>.\start-dfs.cmd

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin>.\start-yarn.cmd
starting yarn daemons
```

```
Apache Hadoop Distribution - yarn  nodemanager
/node/*
2021-04-29 13:46:24,723 INFO http.HttpServer2: adding path spec:
/ws/*
2021-04-29 13:46:25,170 INFO webapp.WebApps: Registered webapp g
uice modules
2021-04-29 13:46:25,172 INFO http.HttpServer2: Jetty bound to po
rt 8042
2021-04-29 13:46:25,176 INFO server.Server: jetty-9.4.20.v201908
13; built: 2019-08-13T21:28:18.144Z; git: 84700530e645e812b33674
7464d6fbbf370c9a20; jvm 1.8.0_161-b12
2021-04-29 13:46:25,216 INFO server.session: DefaultSessionIdMan
ager workerName=node0
2021-04-29 13:46:25,216 INFO server.session: No SessionScavenger
set, using defaults
2021-04-29 13:46:25,220 INFO server.session: node0 Scavenging ev
ery 660000ms
2021-04-29 13:46:25,232 INFO server.AuthenticationFilter: Unable
to initialize FileSignerSecretProvider, falling back to use ran
dom secrets.
2021-04-29 13:46:25,236 INFO handler.ContextHandler: Started o.e
.j.s.ServletContextHandler@f74e835{/logs,/logs,file:///E:/Hadoop/
hadoop-3.1.4/hadoop-3.1.4/logs/,AVAILABLE}
2021-04-29 13:46:25,237 INFO handler.ContextHandler: Started o.e
.j.s.ServletContextHandler@19fe4644{/static,/static,jar:file:/E:/
Hadoop/hadoop-3.1.4/hadoop-3.1.4/share/hadoop/yarn/hadoop-yarn-c
ommon-3.1.4.jar!/webapps/static,AVAILABLE}
2021-04-29 13:46:25,252 WARN webapp.WebInfConfiguration: Can't g
enerate resourceBase as part of webapp tmp dir name: java.lang.N
ullPointerException

Apache Hadoop Distribution - yarn  resourcemanager
1-04-29 13:46:19,733 INFO pb.RpcServerFactoryPBImpl: Adding protocol o
apache.hadoop.yarn.server.api.ResourceTrackerPB to the server
1-04-29 13:46:19,734 INFO ipc.Server: IPC Server Responder: starting
1-04-29 13:46:19,735 INFO ipc.Server: IPC Server listener on 8031: sta
ng
1-04-29 13:46:19,782 INFO util.JvmPauseMonitor: Starting JVM pause mon
r
1-04-29 13:46:19,977 INFO ipc.CallQueueManager: Using callQueue: class
va.util.concurrent.LinkedBlockingQueue queueCapacity: 5000 scheduler:
ss org.apache.hadoop.ipc.DefaultRpcScheduler
1-04-29 13:46:20,026 INFO ipc.Server: Starting Socket Reader #1 for po
8030
1-04-29 13:46:20,228 INFO pb.RpcServerFactoryPBImpl: Adding protocol o
apache.hadoop.yarn.api.ApplicationMasterProtocolPB to the server
1-04-29 13:46:20,252 INFO ipc.Server: IPC Server listener on 8030: sta
ng
1-04-29 13:46:20,267 INFO ipc.Server: IPC Server Responder: starting
1-04-29 13:46:20,476 INFO ipc.CallQueueManager: Using callQueue: class
va.util.concurrent.LinkedBlockingQueue queueCapacity: 5000 scheduler:
ss org.apache.hadoop.ipc.DefaultRpcScheduler
1-04-29 13:46:20,483 INFO ipc.Server: Starting Socket Reader #1 for po
8032
1-04-29 13:46:20,484 INFO pb.RpcServerFactoryPBImpl: Adding protocol o
apache.hadoop.yarn.api.ApplicationClientProtocolPB to the server
1-04-29 13:46:20,485 INFO ipc.Server: IPC Server listener on 8032: sta
ng
1-04-29 13:46:20,486 INFO ipc.Server: IPC Server Responder: starting
1-04-29 13:46:20,494 INFO resourcemanager.ResourceManager: Transitione
o active state
```

After both services have started perform following:

4. Copy the input.text file from local system to Hadoop File System using following command

```
hadoop dfs -copyFromLocal E:\input.txt /Vasundhara/
```

Confirm whether file is copied using following command

```
hadoop dfs -ls /Vasundhara/
```

5. Once file is available on HDFS, write the Map-reduce program in Eclipse/Netbeans IDE.

Number of programs to be created : 3 under the same project

1. word_Mapper : to provide mapper functionality code

2. word_Reducer : to provide reducer functionality code

JAR files to be included in the Java Project : 3

Following JAR files are required to be added in the Java Project.

1. `hadoop-mapreduce-client-core-3.1.4.jar`

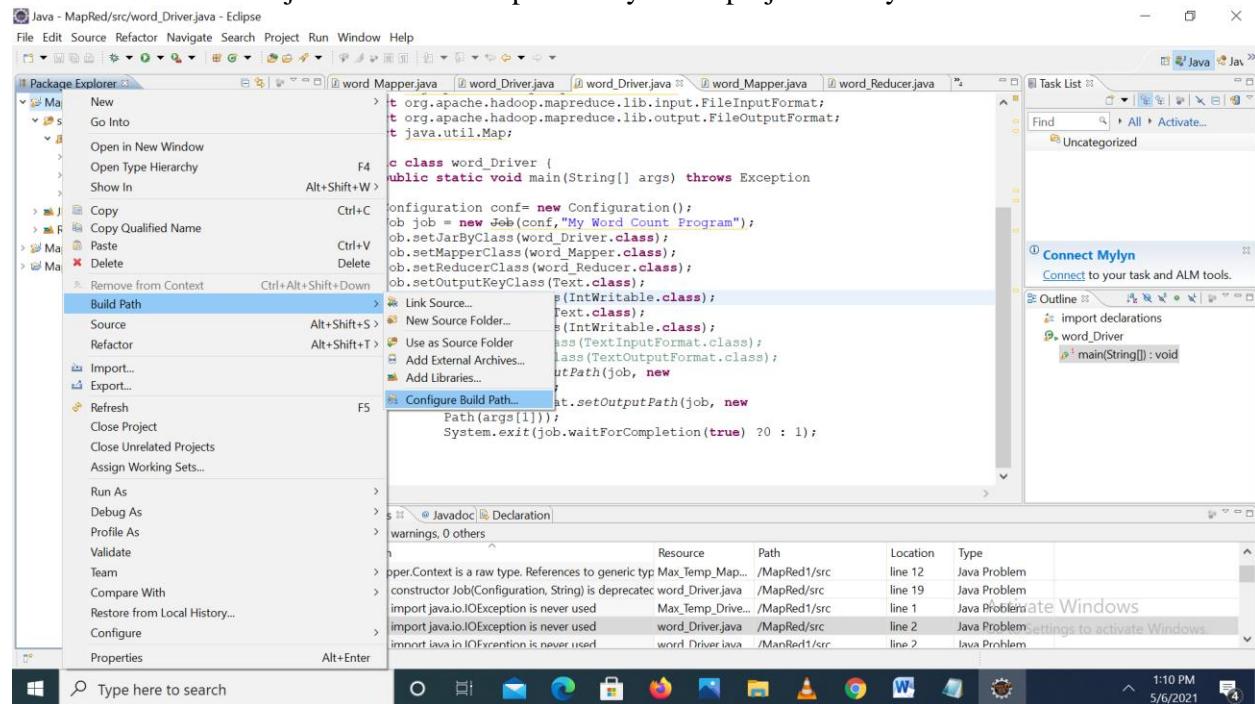
Above file is available in the following folder path :

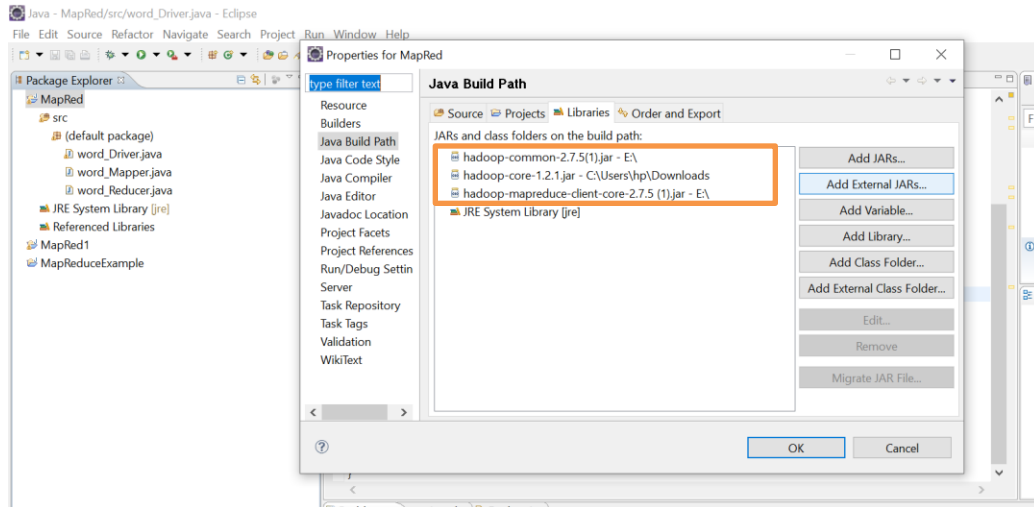
E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\share\hadoop\mapreduce

2. `hadoop-core-1.2.1` (Download this jar file)
3. `hadoop-common-2.7.5.jar`(Download this jar file)

Open Netbeans/eclipse->create new java project

Include the 3 external jar files discussed previously in the project library as follows :





In the Project src folder create 3 classes as follows :

- 1.word_Mapper (without main method)
- 2.word_Reducer (without main method)
- 3.word_Driver(with main method)

1. word_Mapper.java

```
import java.io.IOException;
import java.util.StringTokenizer;
```

```
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 word_Mapper extends Mapper<LongWritable,Text,Text,IntWritable>
{
    public void map(LongWritable key, Text value, Context context) throws IOException,InterruptedException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            value.set(tokenizer.nextToken());
            context.write(value, new IntWritable(1));
        }
    }
}
```

2. word_Reducer.java

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
```

```
public class word_Reducer extends Reducer <Text,IntWritable,Text,IntWritable>
```

```

{
    public void reduce(Text key, Iterable <IntWritable> values, Context context)
        throws IOException, InterruptedException {
        int sum=0;
        for(IntWritable x: values)
        {
            sum+=x.get();
        }
        context.write(key, new IntWritable(sum));
    }
}

```

3. word_Driver.java

```

import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import java.util.Map;

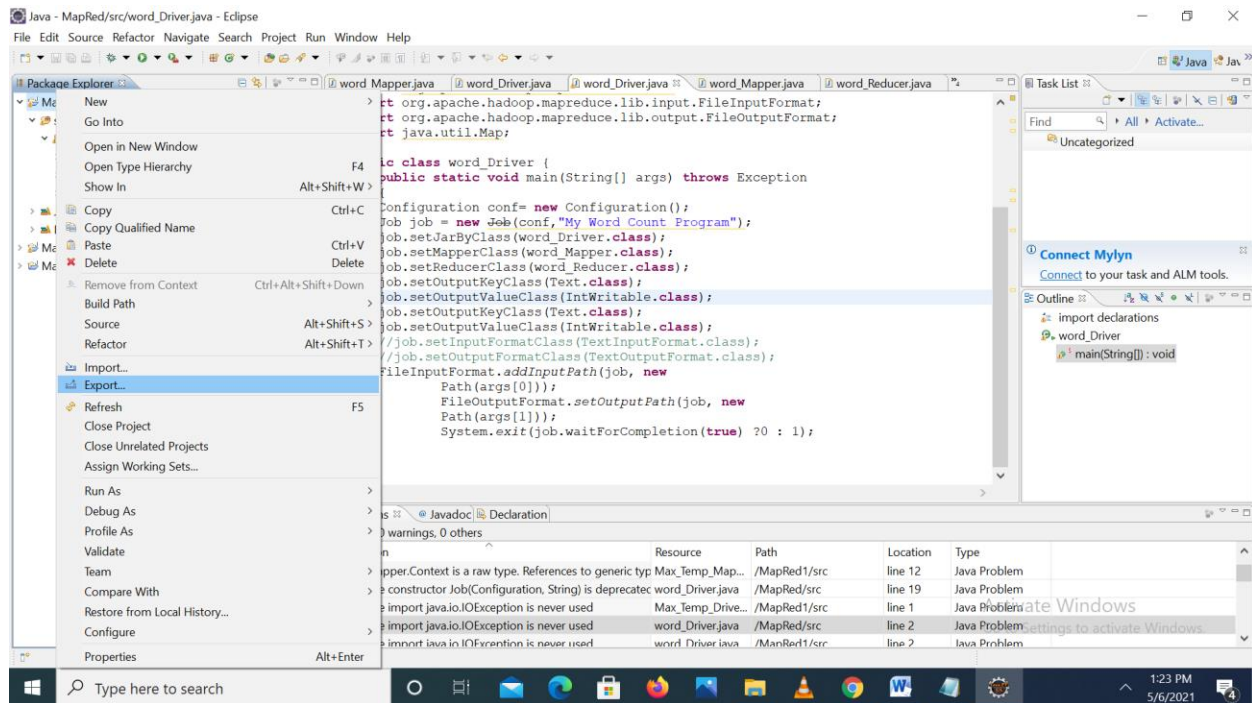
public class word_Driver {
    public static void main(String[] args) throws Exception
    {
        Configuration conf= new Configuration();
        Job job = new Job(conf, "My Word Count Program");
        job.setJarByClass(word_Driver.class);
        job.setMapperClass(word_Mapper.class);
        job.setReducerClass(word_Reducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        //job.setInputFormatClass(TextInputFormat.class);
        //job.setOutputFormatClass(TextOutputFormat.class);
        FileInputFormat.addInputPath(job, new
            Path(args[0]));
        FileOutputFormat.setOutputPath(job, new
            Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

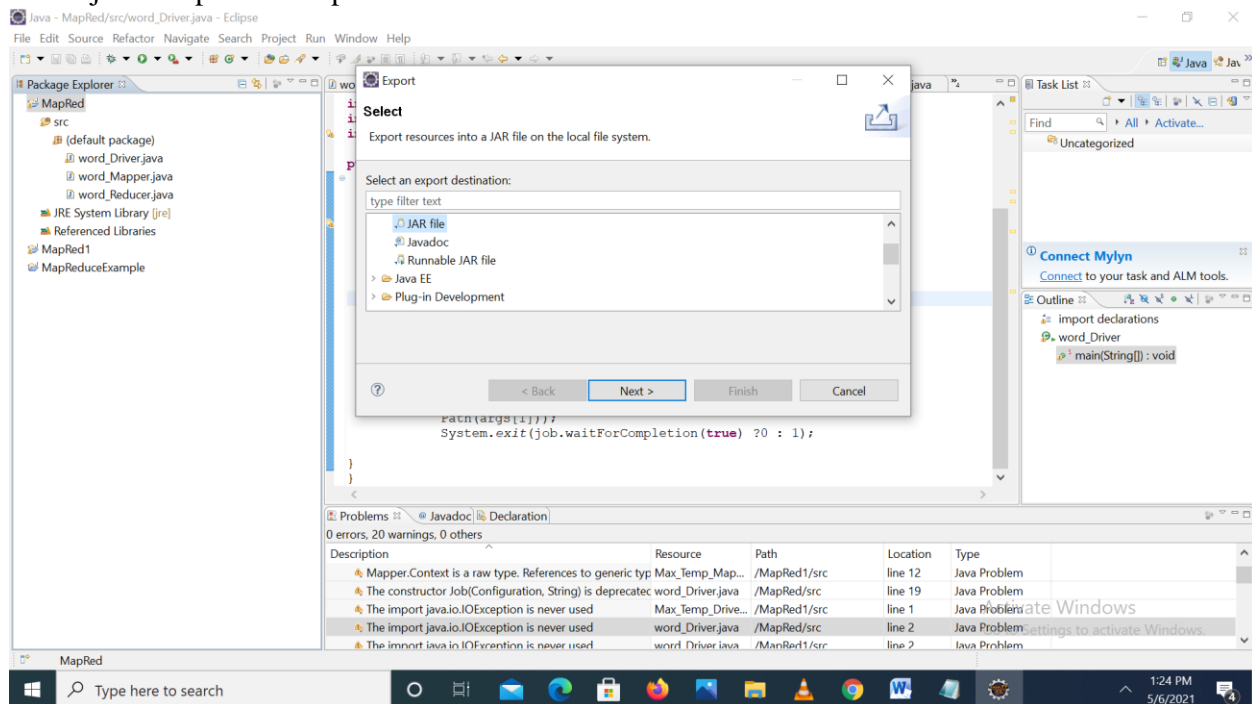
Save all the three programs if no errors.

5. Create a jar file of your project.

Right click on Project in Package explorer and select Export menu

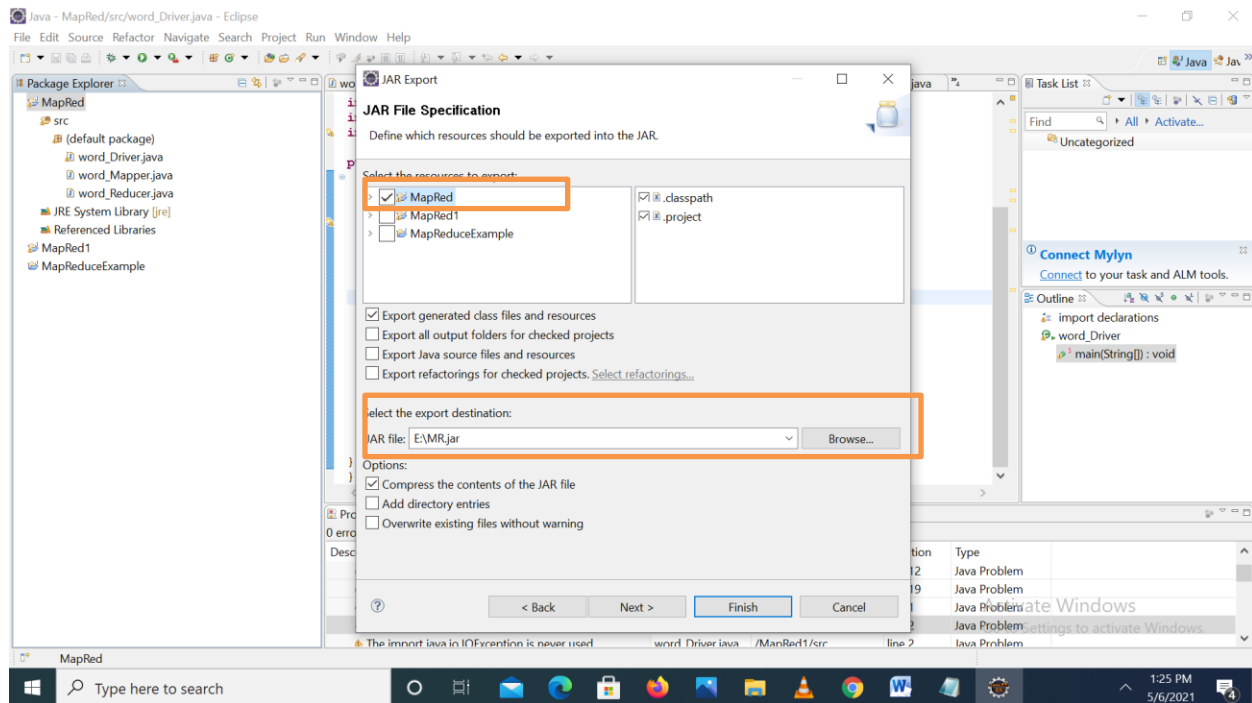


Select jar file option in Export Window:



Click on Next

Select your specific project from the list of projects and specify the path to store the jar file:



Click on finish.

After the jar file is successfully created in the mentioned path ,

6. Execute the Project from command prompt window.

Change the directory in command prompt window to /bin folder instead of sbin folder:

```
cd E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\bin
```

Syntax for executing the project :

```
hadoop jar Path_of_jar_file Name_of_Driver_class input_file_path_hdfs output_folder_name
```

```
E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\bin>hadoop jar E:\Project1\MR.jar word_Driver
/Vasundhara/input.txt MR_Out
```

Where :

E:\Project1\MR.jar : is the exported project jar file path

word_Driver : is the class name of Driver class which has the main() method

/Vasundhara/input.txt : indicates the source file path available on HDFS

MR_Out : User-defined Folder name for storing Map-reduce Output

The screenshot shows a Windows desktop environment. A terminal window is open, displaying the execution of Hadoop commands and their output. The commands include setting the HADOOP_HOME environment variable and running a Hadoop job. The output shows the job's progress, including the number of splits, the number of map and reduce tasks, and the final status of the job.

The web browser window displays the Hadoop Web User Interface (WUI) at the URL <http://localhost:8088/cluster/apps/RUNNING>. The WUI shows a sidebar with navigation options and a main area with application status details. The application status is shown as 'RUNNING' with a final status of 'UNDEFINED' and 2 running containers.

7. To access the status of the application we can use the Hadoop Web User Interface by providing following link in browser:

<http://localhost:8088/cluster>

It shows information related to active nodes and application status.

The screenshot shows the Hadoop Web User Interface (WUI) at the URL <http://localhost:8088/cluster/apps/RUNNING>. The WUI displays a table of running applications. The application 'application_1620066094221_0001' is shown as 'RUNNING' with a final status of 'UNDEFINED' and 2 running containers.

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers
application_1620066094221_0001	hp	My Word Count Program	MAPREDUCE	default	0	Tue May 4 00:08:16 +0550 2021	Tue May 4 00:09:10 +0550 2021	N/A	RUNNING	UNDEFINED	2

Showing 1 to 1 of 1 entries

8. After the project finishes 100% execution of both map and reduce tasks we can access the output folder from Hadoop HDFS UI by providing following link in the browser:

<http://localhost:9870>

The screenshot shows the Hadoop HDFS UI in a web browser. The address bar shows `localhost:9870/explorer.html#/user/hp`. The navigation bar includes links for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main section is titled "Browse Directory" and shows the path `/user/hp`. Below the path is a search bar and a "Go!" button. A table lists the directory contents with columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. A single entry is listed: `drwxr-xr-x` (Permission), `hp` (Owner), `supergroup` (Group), `0 B` (Size), `May 04 00:13` (Last Modified), `0` (Replication), `0 B` (Block Size), and `MR_out` (Name). The entry is highlighted with an orange border. Below the table, it says "Showing 1 to 1 of 1 entries" and "Previous 1 Next". At the bottom, there is a footer with "Hadoop, 2020." and an "Activate Windows" watermark.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hp	supergroup	0 B	May 04 00:13	0	0 B	MR_out

9. Click on the output folder: MR_out .It will display 2 files:

_SUCCESS : Shows the status and

part-r-00000 : has the actual output of Map-reduce

localhost:9870/explorer.html#/user/hp/MR_out

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/hp/MR_out Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hp	supergroup	0 B	May 04 00:13	1	128 MB	_SUCCESS
-rw-r--r--	hp	supergroup	179 B	May 04 00:13	1	128 MB	part-r-00000

Showing 1 to 2 of 2 entries Previous 1 Next

Hadoop, 2020.

Activate Windows
Go to Settings to activate Windows.

10. Click on part-r-00000 file : It will provide a window with option to download the file as follows :

localhost:9870/explorer.html#/user/hp/MR_out

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Browse Directory

/user/hp/MR_out Go!

Show 25 entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hp	supergroup	0 B	May 04 00:13	1	128 MB	_SUCCESS
-rw-r--r--	hp	supergroup	179 B	May 04 00:13	1	128 MB	part-r-00000

Showing 1 to 2 of 2 entries Previous 1 Next

File information - part-r-00000

Download Head the file (first 32K) Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073741872
Block Pool ID: BP-772885956-192.168.1.103-1620056817645
Generation Stamp: 1048
Size: 179
Availability:

11. Open the downloaded file to view the program output :

```
part-r-00000 - Notepad
File Edit Format View Help
Analytics. 1
Big 3
Data 3
IoT,Social 1
Media,E-Commerce 1
The 2
analysing 1
are 1
course 2
data. 1
etc. 1
find 1
focuses 1
from 1
insights 1
is 1
name 1
on 1
sources 1
the 1
to 1

Ln 1, Col 1 100% Unix (LF) UTF-8
```

12. Another way of displaying the output is by reading the file from HDFS using following command syntax in command prompt :

`hadoop fs -cat Path_of_part-r-00000_file_hdfs`

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin>`hadoop fs -cat /user/hp/MR_out/part-r-00000`

Administrator: Command Prompt - hadoop dfs -copyFromLocal E:\input.txt /Vasundhara/ - hadoop dfs -copyFromLocal E:\input.txt /V...

Terminate batch job (Y/N)? Y

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\bin>cd..

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4>cd sbin

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin>hadoop fs -cat /user/hp/MR_out/part-r-00000

Analytics. 1
Big 3
Data 3
IoT,Social 1
Media,E-Commerce 1
The 2
analysing 1
are 1
course 2
data. 1
etc. 1
find 1
focuses 1
from 1
insights 1
is 1
name 1
on 1
sources 1
the 1
to 1

E:\Hadoop\hadoop-3.1.4\hadoop-3.1.4\sbin>

File Explorer

Home

Share

File

Name

Sort

_SUCCESS

part-r-00000

Previous

1

Next

Activate Windows

