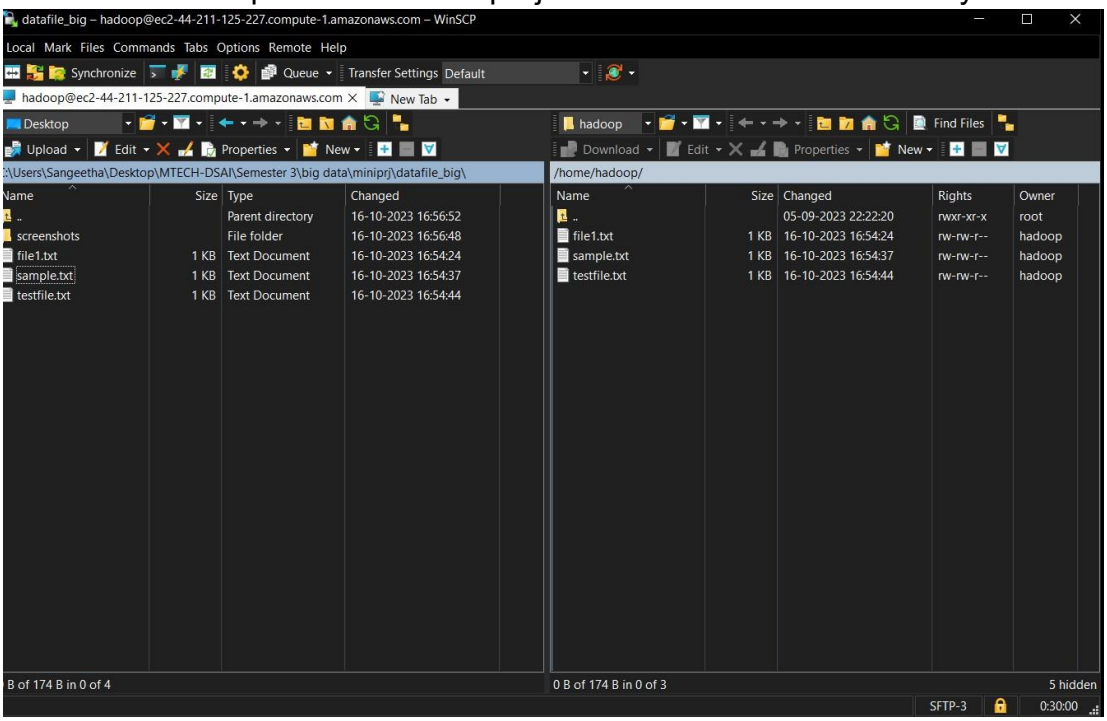
Assessment BDML (Big Data & Machine Learning)

Shafeena Farheen

Tools:

i) **Putty:** Putty is a popular open-source terminal emulator, serial console, and network file transfer application. It is commonly used on Windows systems to connect to remote servers or network devices using various network protocols such as SSH, Telnet, and Rlogin. Putty provides a user-friendly interface for accessing and managing remote systems.

ii) **WinSCP** : WinSCP is a popular open-source SFTP, FTP, WebDAV, Amazon S3, and SCP client for Windows. It provides a user-friendly interface for securely transferring files between local and remote systems. Using the WinSCP we will transfer the required file for our project from windows to remote system.



Marks (25 Marks)

1. Write HDFS shell commands for the following . (18 marks)

a) To Print Version of installed Hadoop (1 mark)

```
hadoop@ip-172-31-83-37:~
[hadoop@ip-172-31-83-37 ~]$ hadoop version
Hadoop 3.3.3-amzn-5
Source code repository Unknown -r Unknown
Compiled by release on 2023-08-22T18:41Z
Compiled with protoc 3.7.1
From source with checksum 6973de6e7dffbfe8a85d8d376349339
This command was run using /usr/lib/hadoop/hadoop-common-3.3.3-amzn-5.jar
[hadoop@ip-172-31-83-37 ~]$
```

Inference:

- To check the Hadoop version installed on your system use the command "**hadoop version**"
- Running this command will display the Hadoop version number along with other information about the Hadoop distribution installed on your system.

b) For listing the files and directories present in HDFS „path" directory present under root (/path). (2 marks)

```
hadoop@ip-172-31-83-37:~
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -mkdir path
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls
Found 1 items
drwxr-xr-x   - hadoop hdfsadmingroup          0 2023-10-16 11:51 path
```

```
hadoop@ip-172-31-83-37:~
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls /user/hadoop/path/
Found 1 items
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 12:06 /user/hadoop/path/file1.txt
[hadoop@ip-172-31-83-37 ~]$
```

```
hadoop@ip-172-31-83-37:~
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x   - hdfs hdfsadmingroup          0 2023-10-16 11:40 /apps
drwxrwxrwt   - hdfs hdfsadmingroup          0 2023-10-16 11:41 /tmp
drwxr-xr-x   - hdfs hdfsadmingroup          0 2023-10-16 11:40 /user
drwxr-xr-x   - hdfs hdfsadmingroup          0 2023-10-16 11:40 /var
[hadoop@ip-172-31-83-37 ~]$
```

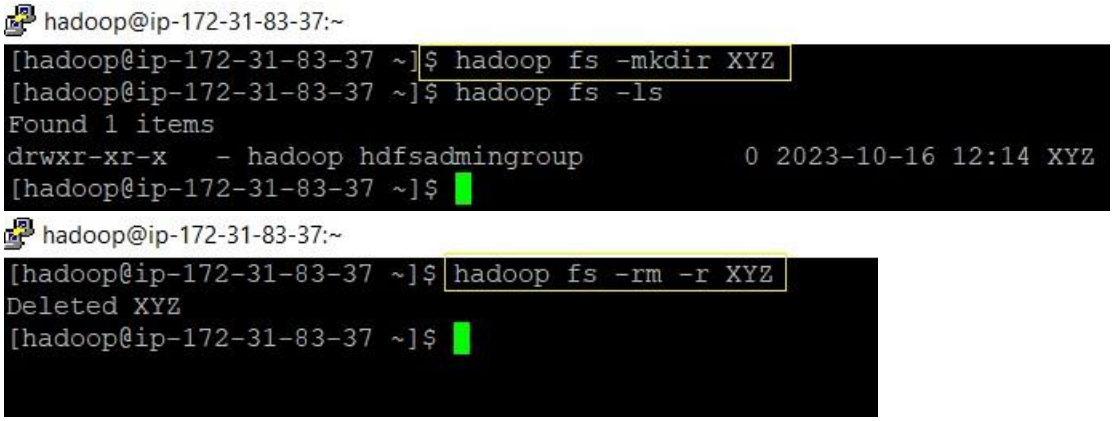
Inference:

- We created the directory name as “path” in hdfs using the command **“hadoop fs -mkdir path”**
- For listing the file and directory in hdfs we use “ls” command **“hadoop fs -ls /user/hadoop/path/”** -here, it listed out the file1.txt present under “path” directory

  “hadoop fs -ls” – here, it listed out the the “path” directory present in the hadoop

- The ”-ls” command will list the files and directories in the root directory of HDFS.
- In HDFS, the command **hadoop fs -ls /** is used to list files and directories in the root directory of HDFS. „/ „ specifies the root directory of HDFS.

c) To Delete an empty directory named as XYZ. (2 marks)

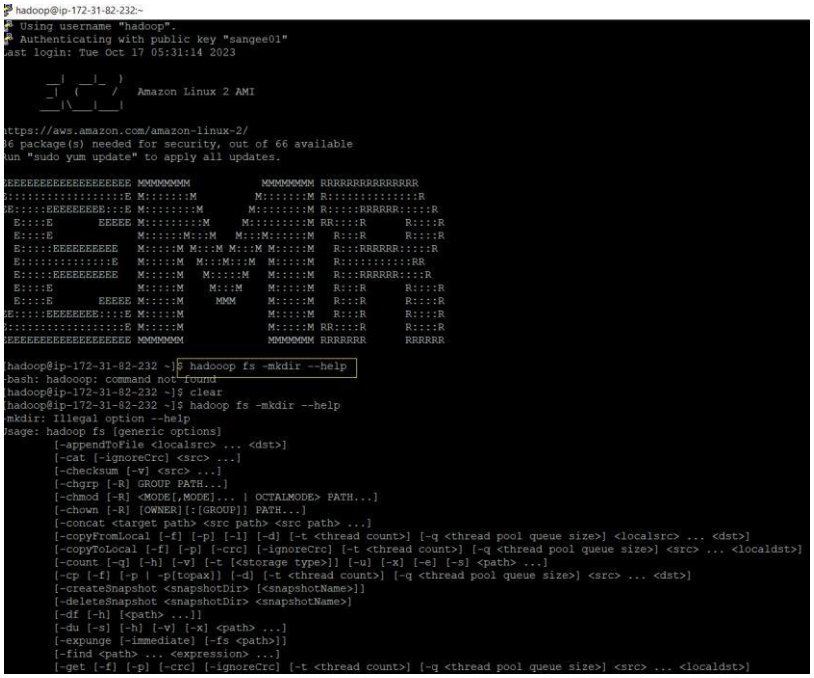


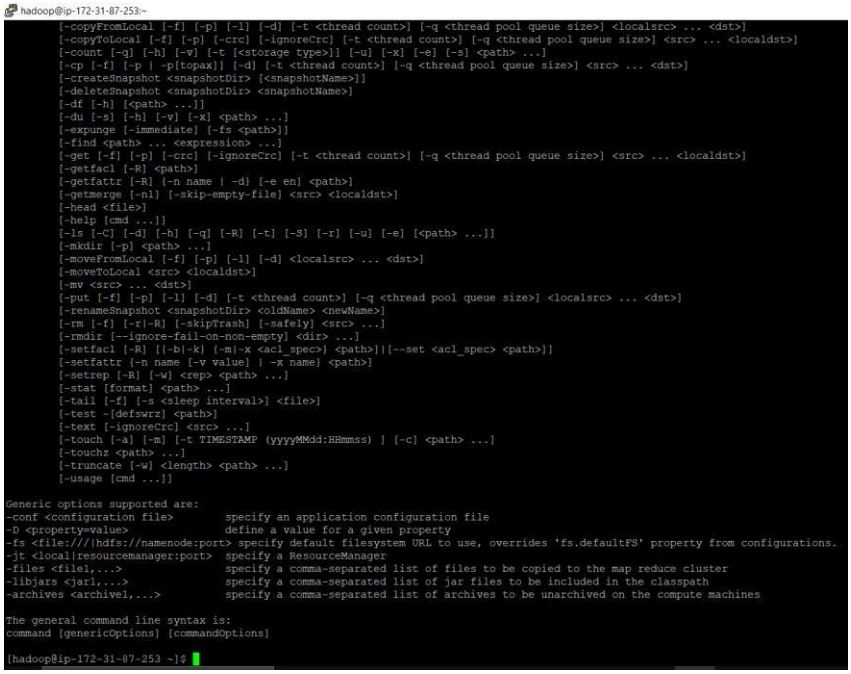
Inference:

- Create a directory named as “XYZ” in hadoop using the below command **“hadoop fs -mkdir XYZ”**
- To remove the empty directory in hadoop use “-rm –r” as below **“hadoop fs –rm –r XYZ”**
- **hdfs dfs -rm <hdfs-path>:** This command is used to delete a file specified by <hdfs-path> in HDFS. It can only delete files and cannot be used to remove directories.
- **hdfs dfs -rm -r <hdfs-path>:** This command is used to recursively delete a directory and its contents specified by <hdfs-path> in HDFS.It can delete both files and directories and is useful when you want to delete an entire directory structure.

d)

To fetch the usage instructions of mkdir command .(2 marks)





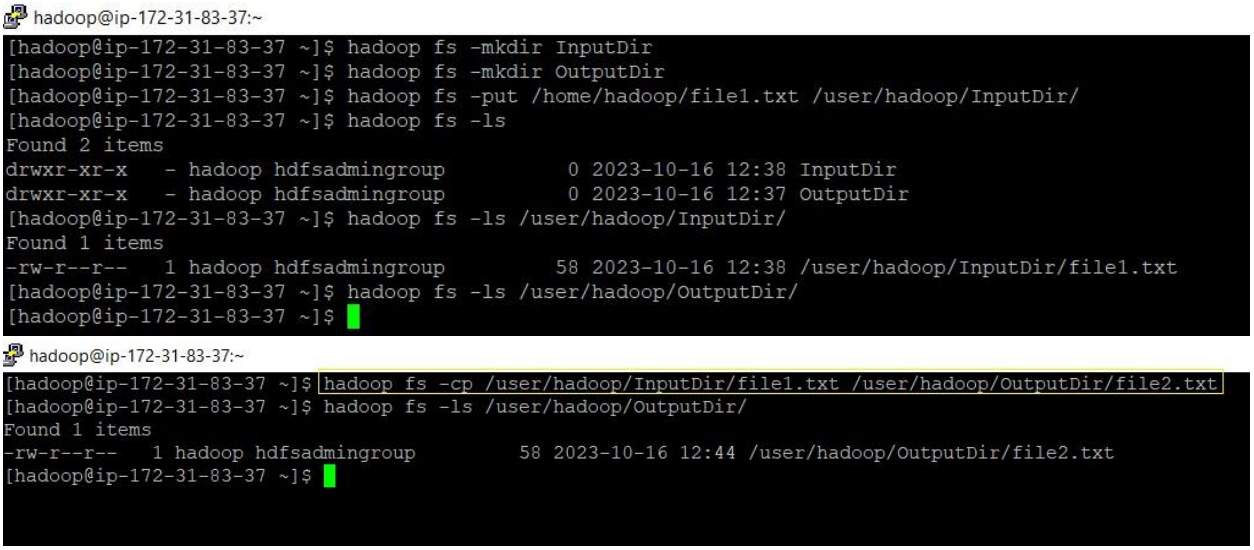
Inference:

- To fetch the usage instructions of the hadoop fs -mkdir command, you can use the hadoop command with the -fs option and the -mkdir subcommand, followed by the --help flag. **“hadoop fs –mkdir - -help”**

- Running this command will display the usage instructions and available options for the hadoop fs -mkdir command, providing detailed information about how to use the command to create directories in Hadoop Distributed File System (HDFS).
- Create a directory named as “XYZ” in hadoop using the below command

  “hadoop fs -mkdir XYZ”

e) To Copy „file1.txt‟ from „InputDir‟ to „OutputDir‟ as file2.txt (2 marks)



Inference:

- Create a directory named as”InputDir” and “OutputDir” using “mkdir” in hadoop

  “**hadoop fs -mkdir InputDir”**

  “**hadoop fs -mkdir OutputDir”**
- Use hadoop “put” command to place the “file1.txt” in hdfs path under “InputDir” directory

  “hadoop fs –put /home/hadoop/file1.txt /user/hadoop/InputDir/”
- Copy command **“-cp**” is used to copy the “**file1.txt**” from “InputDir” to “OutputDir” and save it as **“file2.txt**”

  “hadoop fs –cp /user/hadoop/file1.txt /user/hadoop/OutputDir/file2.txt”
- **/user/hadoop/InputDir/file1.txt** is the source file path in HDFS.
- **/user/hadoop/OutputDir/file2.txt** is the destination file path in HDFS, and file1.txt will be copied and renamed to file2.txt in the destination directory.

f)

Write command for copy the „testfile" of the hadoop filesystem (present under root) to the local file system (pwd) . (3 marks)

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ ls
file1.txt  sample.txt
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:09 testfile.txt
[hadoop@ip-172-31-83-37 ~]$
```

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -get /user/hadoop/testfile.txt /home/hadoop
[hadoop@ip-172-31-83-37 ~]$ ls
file1.txt  sample.txt  testfile.txt
[hadoop@ip-172-31-83-37 ~]$
```

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -copyToLocal /user/hadoop/testfile.txt /home/hadoop
[hadoop@ip-172-31-83-37 ~]$ ls
file1.txt  sample.txt  testfile.txt
[hadoop@ip-172-31-83-37 ~]$
```

Inference:

- The "testfile.txt" is placed in hdfs path using „put" command. And the local file system (pwd) doesn"t have the „testfile.txt"
- In Hadoop, the "**hadoop fs –get"** command is used to copy files or directories from HDFS to the local file system.

  "hadoop fs –get /user/hadoop/testfile.txt" /home/hadoop"
- "-get" command takes two parameter source and destination path The path to the file or directory in HDFS that you want to copy is called as source path (**"/user/hadoop/testfile.txt").** The destination path in the local file system where you want to copy the file or directory**("/home/hadoop")**
- In Hadoop, the **hadoop fs -copyToLocal** command is used to copy files or directories from HDFS (Hadoop Distributed File System) to the local file system. The **copyToLocal** command is an alias for the **-get** command.

  "hadoop fs –copyToLocal /user/hadoop/testfile.txt /home/hadoop"
- It supports various options and can be used to copy files or directories recursively (-get -r or -get -R).

h)

Write command for display the content of the „sample" file present in newDataFlair directory of HDFS (under root). (3 marks)

Inference:

- Create a directory "newDataFlair" using the "mkdir" command in hadoop

  "hadoop fs –mkdir newDataFlair"

- The sample.txt" is placed inside "newDataFlair" directory hdfs path using „put" command.

"hadoop fs –put /home/hadoop/sample.txt /user/hadoop/newDataFlair/"

- The contents of **"sample.txt"** is read using the **"–cat"** command **"hadoop fs –cat /user/hadoop/newDataFlair/sample.txt"**

- The hadoop fs -cat command can be useful when you want to quickly view the content of a file in HDFS without copying it to the local file system. It can also be used in combination with Unix/Linux commands, allowing you to process the data directly from HDFS.

- Here , the "sample.txt " contains below content which read using hadoop "-cat" command.

  „Hello !!!
  Welcome to Big data course
  My first Miniproject"

Write command for copy local file named file1 under the present working directory(pwd) of local file system to the Hadoop filesystem under root. (3 marks)

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ ls
file1.txt  sample.txt  testfile.txt
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls
Found 2 items
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:17 sample.txt
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:09 testfile.txt
[hadoop@ip-172-31-83-37 ~]$
```

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -put /home/hadoop/file1.txt /user/hadoop/
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls
Found 3 items
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:23 file1.txt
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:17 sample.txt
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:09 testfile.txt
[hadoop@ip-172-31-83-37 ~]$
```

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ ls
file1.txt  sample.txt  testfile.txt
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -copyFromLocal /home/hadoop/file1.txt /user/hadoop/
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls
Found 3 items
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:25 file1.txt
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:17 sample.txt
-rw-r--r--   1 hadoop hdfsadmingroup         58 2023-10-16 14:09 testfile.txt
[hadoop@ip-172-31-83-37 ~]$
```

Inference:

- The local file system (pwd) have „file1.txt" and the hadoop root path does not have „file1.txt".
- In Hadoop, the "**hadoop fs –put"** command is used to copy files or directories from local file system to HDFS.

  "hadoop fs –put /home/hadoop/file1.txt" /user/hadoop"
- "-put" command takes two parameter source and destination path The path to the file or directory in local filesystem that you want to copy is called as source path **("/home/hadoop/file1.txt").** The destination path in the HDFS where you want to copy the file or directory**("/user/hadoop")**
- In Hadoop, the **hadoop fs -copyFromLocal** command is used to copy files or directories from Local Filesystem to the HDFS. The **copyFromLocal** command is an alias for the **-put** command. **"hadoop fs –copyFromLocal /home/hadoop/file1.txt /user/hadoop"**
- It supports various options and can be used to copy files or directories recursively (-get -r or -get -R).

j)

2. Illustrate how MapReduce processes WordCount job ? (7 marks)

- Hadoop *MapReduce* is a software framework for developing applications that process Big Data in-parallel on large clusters in a reliable, fault-tolerant manner.
- A MapReduce job has 2 phases – *Map phase* and *Reduce phase.*
- MapReduce *job* divides the input data-set into independent chunks called input splits or simply splits which are processed by the *map tasks* in a completely parallel manner.
- The framework sorts the outputs of the maps, which are then input to the *reduce tasks* which produces the final output.
- The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.
- YARN (Yet Another Resource Negotiator) was introduced in Hadoop 2 to improve the MapReduce implementation.
- MapReduce framework operates exclusively on <key, value> pairs.
- It views the input to the job as a set of <key, value> pairs.
- It produces the output as a set of <key, value> pairs.
- The input key & value and output key & value could be of different data types.

```
(input) <k1, v1> → map → <k2, v2> →
→ reduce → <k3, v3> (output)
```

MapReduce processes WordCount job:

Word Count is a popular example used to demonstrate the MapReduce programming model. In this example, we will consider the input to be a collection of text documents. The goal is to count the frequency of each word in the documents.

1. Input Data:

Let's assume we have text document as input:

"Hello World"
"Hello Hadoop"
"World of Hadoop"

2. Mapper:

- The input data is split into chunks called Input Splits.
- Each Input Split is processed by a Mapper.
- The Mapper processes the input and emits key-value pairs.
- In the Word Count example, the Mapper outputs key-value pairs where the key is a word and the value is 1 for each occurrence of the word

  (Hello, 1)

(World, 1)
(Hello, 1)

(Hadoop, 1)
(World, 1)
(of, 1) (Hadoop,
1)

3. **Shuffle and Sort:**
   - The output from all Mappers is shuffled and sorted based on the
   - keys.

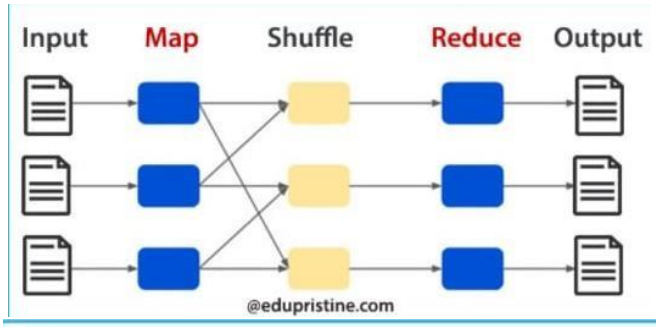
   This ensures that all occurrences of the same word are grouped together.
   (Hello, 1)
   (Hello, 1)
   (Hadoop, 1)
   (Hadoop, 1)
   (of, 1)
   (World, 1)
   (World, 1)
4. **Reducer:**
   - The sorted data is input to Reducers.
   - Reducers aggregate the data based on keys and perform the desired computation.
   - In the Word Count example, Reducers sum up the values for each key (word) to get the total count.

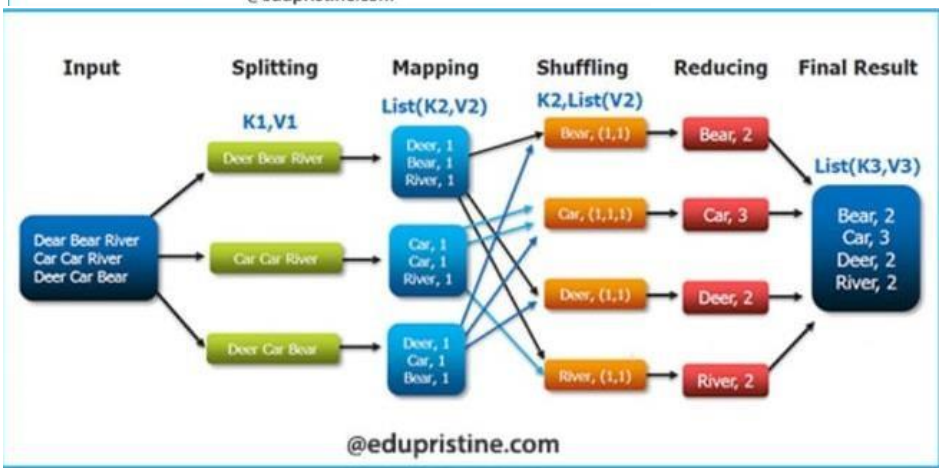
   (Hello, 2)
   (Hadoop, 2)
   (of, 1) (World,
   2)
5. **Final Output:**

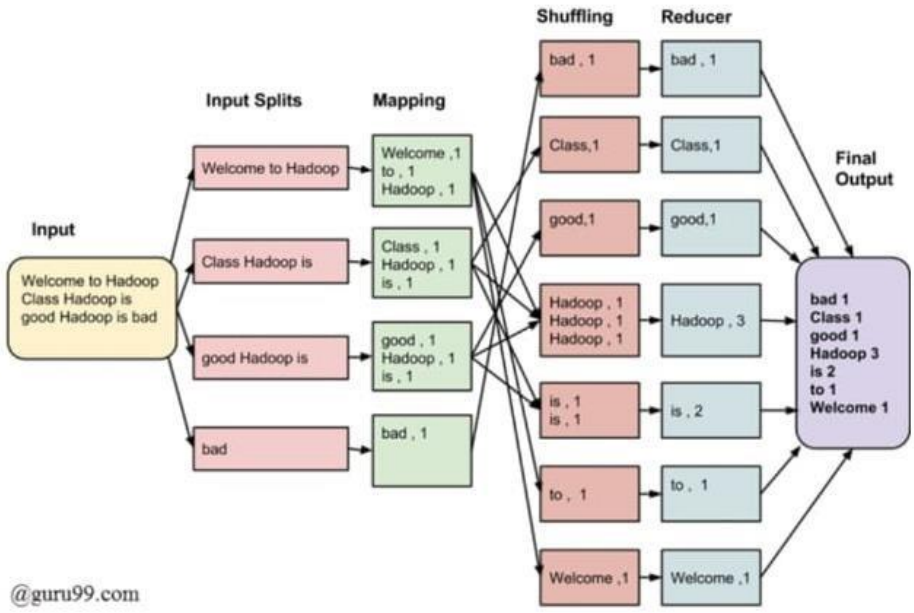
- The output from Reducers is the final result of the Word Count job.
- It shows the frequency of each word in the input documents.

Hello: 2 Hadoop:
2
of: 1
World: 2

Map Reduce word count Architecture:



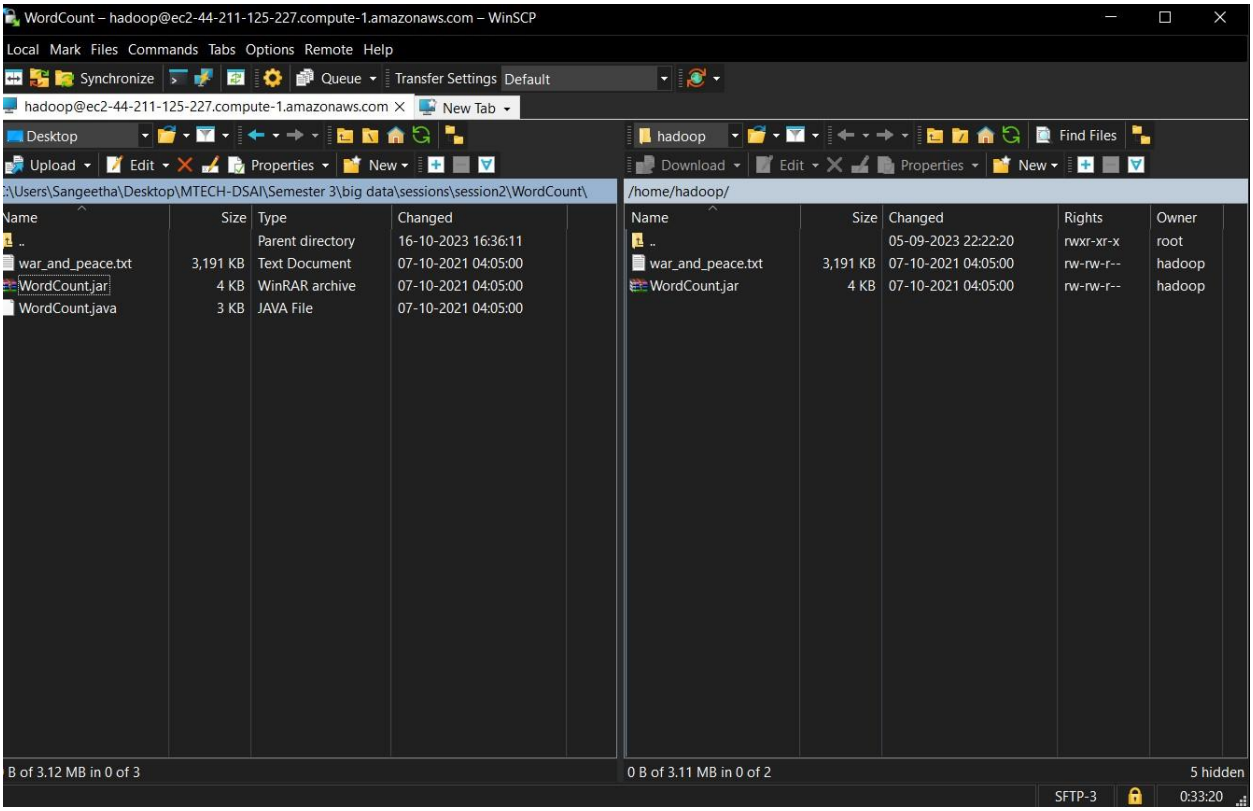




MapReduce Architecture

Implementation of MapReduce job on „war_and_peace.txt" document:

Input:



- First, you need to store the text of "War and Peace" in HDFS. Let's assume you have a file called war_and_peace.txt in HDFS containing the text.
- Using the Winscp tool, the files re transferred from desktop local machine to hadoop .

Using the HDFS commands:

- Create a directory in HDFS named `wcinput`
- Put the file `war_and_peace.txt` into the above HDFS directory

```
hadoop@ip-172-31-83-37:~
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -mkdir wcinput
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -put /home/hadoop/war_and_peace.txt /user/hadoop/wcinput/
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls /user/hadoop/wcinput/
Found 1 items
-rw-r--r--   1 hadoop hdfsadmingroup    3266939 2023-10-16 14:34 /user/hadoop/wcinput/war_and_peace.txt
[hadoop@ip-172-31-83-37 ~]$
```

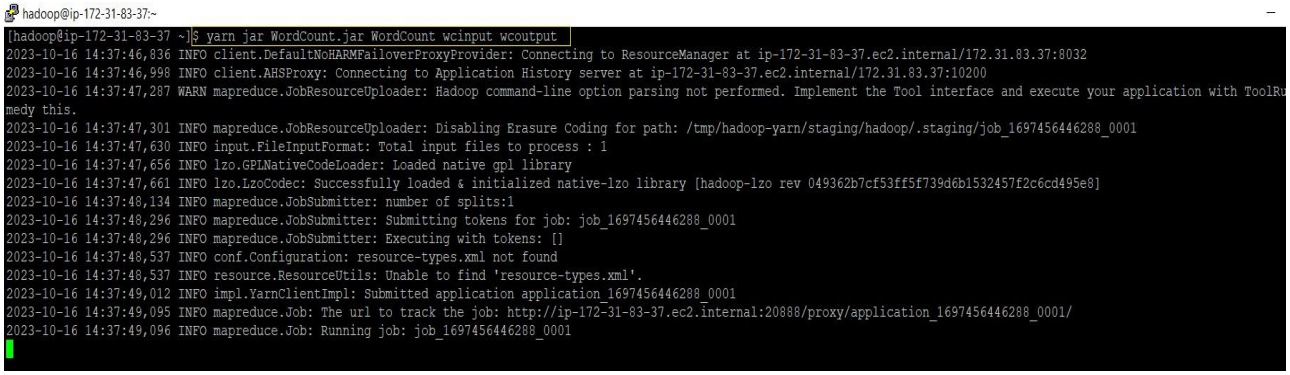
MapReduce Job Configuration:

You can configure the MapReduce job with the following settings:

- **Input Path:** HDFS path to the war_and_peace.txt file.
- **Mapper Class:** A Java class that reads input lines, tokenizes them into words, and emits key-value pairs.

- **Reducer Class:** A Java class that receives word counts from the Mappers, sums them up, and emits the final word counts.
- **Output Path:** HDFS path where the output (word counts) will be stored.
- Run the mapreduce job given as the executable jar file **WordCount.jar,** Note that the above jar file takes 2 arguments – input data location and output data location.

Running the WordCount.jar file using yarn command



$ yarn jar WordCount.jar WordCount wcinput wcoutput

- Input location can be the name of the directory also as mentioned earlier.
- Files in this directory will considered as the input for the job.
- Note that the output location will have 2 output files.
- Below screenshot shows that mapreduce job ran successfully and output is written to the wcoutput directory

Output:

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls
Found 2 items
drwxr-xr-x   - hadoop hdfsadmingroup          0 2023-10-16 14:34 wcinput
drwxr-xr-x   - hadoop hdfsadmingroup          0 2023-10-16 14:38 wcoutput
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -ls /user/hadoop/wcoutput/
Found 3 items
-rw-r--r--   1 hadoop hdfsadmingroup          0 2023-10-16 14:38 /user/hadoop/wcoutput/_SUCCESS
-rw-r--r--   1 hadoop hdfsadmingroup     234177 2023-10-16 14:38 /user/hadoop/wcoutput/part-r-00000
-rw-r--r--   1 hadoop hdfsadmingroup     229165 2023-10-16 14:38 /user/hadoop/wcoutput/part-r-00001
[hadoop@ip-172-31-83-37 ~]$
```

- The output is written to the **"wcoutput"** directory
- There are two files generated that contains output of wordcount mapreduce job
  "part-r-00000"
  "part-r-00001"

hadoop@ip-172-31-83-37:~

```
[hadoop@ip-172-31-83-37 ~]$ hadoop fs -cat  /user/hadoop/wcoutput/part-r-00001 | head -n 10
''Dieu  1
''I     4
''No    1
''Russia        1
''Sergey        1
''The   1
''To    1
''Told  1
''You   1
'800    2
cat: Unable to write to output stream.
[hadoop@ip-172-31-83-37 ~]$
```

- The **"-ls"** command is used to read the output file of wordcount mapreduce job, "**head –n 10**" displays the ten row or line from the output

In summary, MapReduce processes the Word Count job by splitting the input data, mapping words to 1, shuffling and sorting the intermediate key-value pairs, reducing and aggregating the counts for each word, and providing the final word count output