Data System Architecture Assignment 2.

HDFS and Map Reduce

Name : Bhavesh Waghela

Student ID : 200532173

Table of Content

6. dfs -rmr

ii. FSCK Commands

1. fsck block
2. fsck file
3. fsck location
4. fsck racks
5. fsck openforwrite

2. Run Map Reduce Jar File

a. Follow the slides (Week 7 – Hadoop Map Reduce Framework page 73-77). Copy salesjan2009 and jar file to your Hadoop cluster then run the following command.

i. Use WinSCP File Transfer to Local System.

ii. Login to putty.

iii. hadoop command execution for map reduce.

Part 1 - Transfer a file to HDP and run few HDFS commands

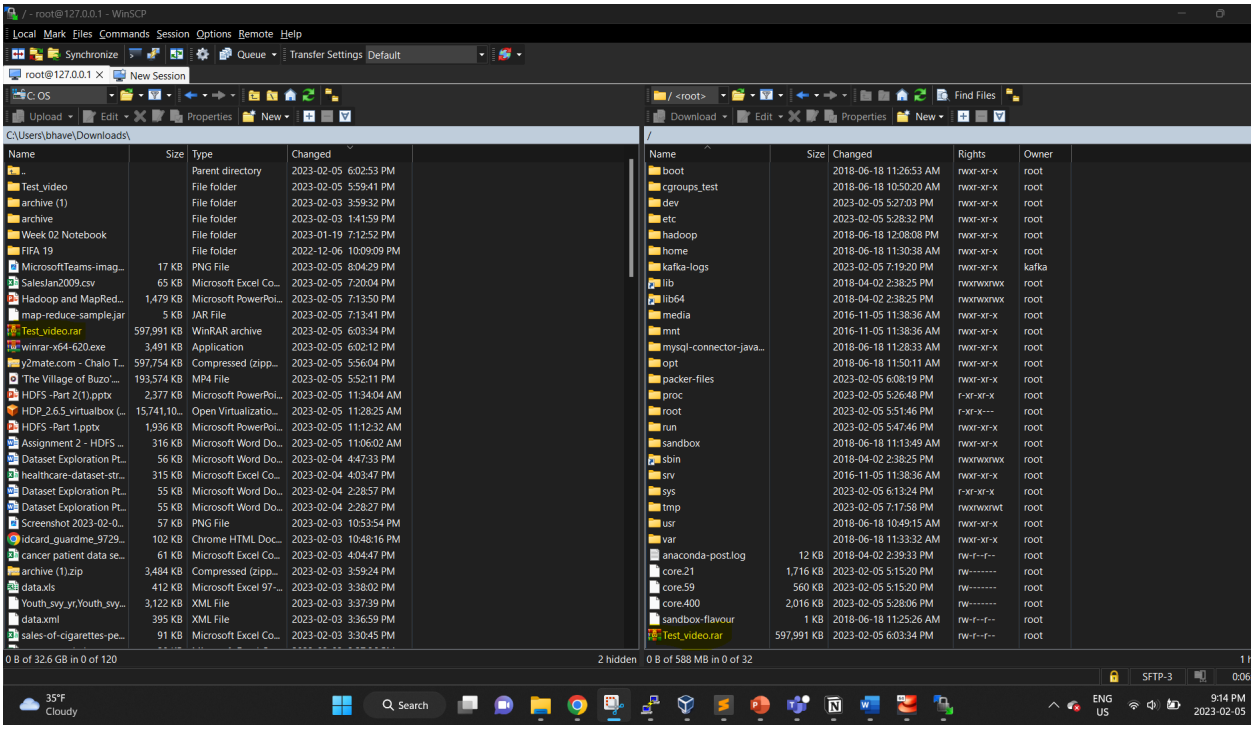
A. Create a test file which is more than 500 MB (for example zip a movie) then upload it to HDP then run the command mentioned in the lecture (HDFS -Part 2 Lecture - Slide 48 -53)

1. Use WinSCP File Transfer to Local System.

a. Download a video of above 500MB and save the same in a .rar file.

b. Connect to WinSCP with the local IP address and transfer the file to linux system.

c. Connect to the local system and check if the file is available where it was dropped.

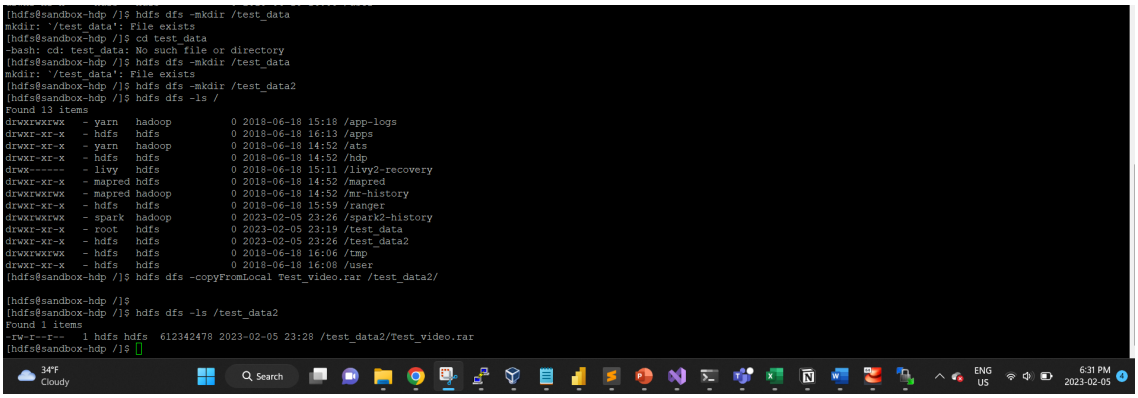


2. HDFS create a directory and transfer the RAR file to the server.

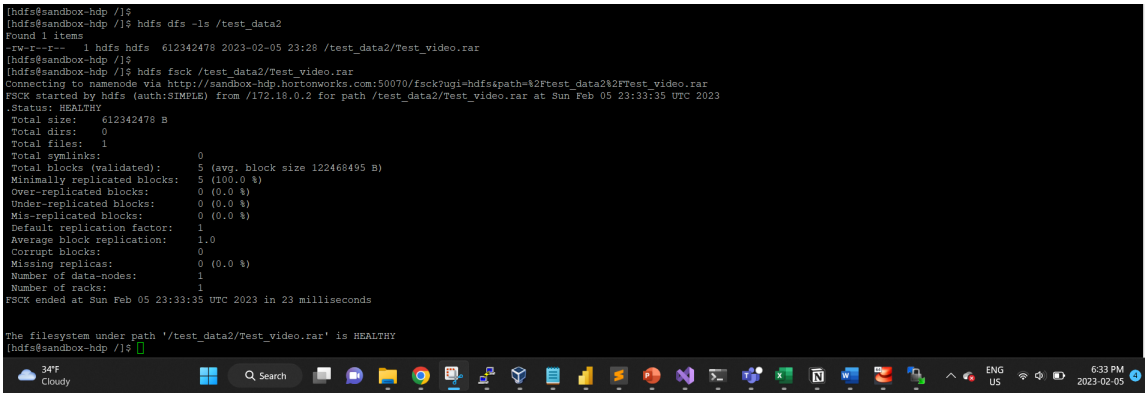
a. Log in to putty with the hdfs root user and password.

b. Switch the user to hdfs user and create a directory to copy the file into the same.

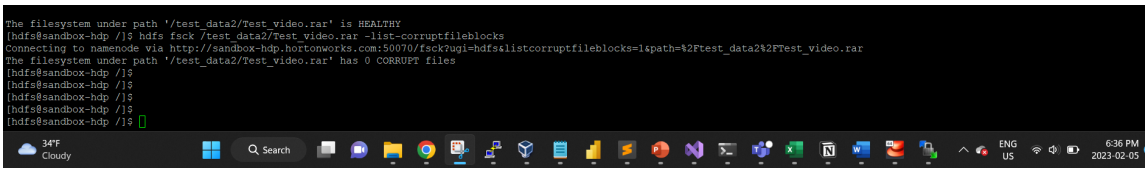
c. Copy file from the local machine to hdfs for further processing the file into chunks.



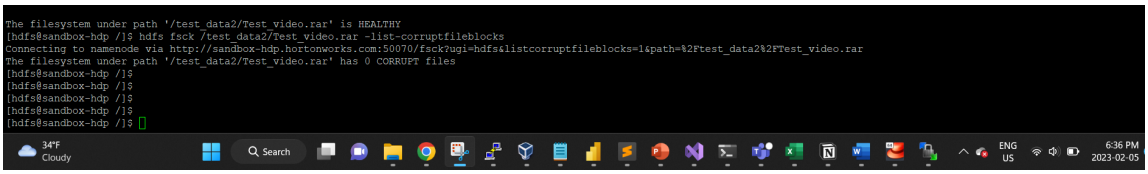
d. Using fsck command



f. Check if the list is corrupted.



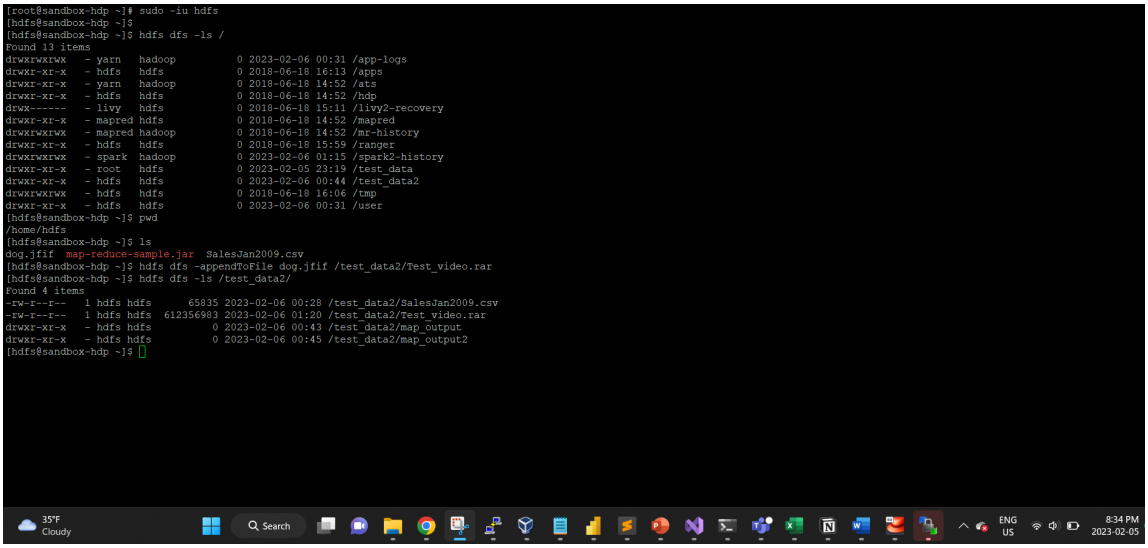
g. check the summary of the transfer with the help of -file and -blocks parameters



B. Use the following link and explore the commands for fsck and dfs . Choose five commands from fsk and five commands from dfs. make sure you explain the command along with screen shots)

DSF Commands

1. Append Command - Appends the content to the file which is present on HDFS. Append single source. or multiple sources from the local file system to the destination file system.



2. dfs du - Du command is used to How much file Occupied in the disk. The field is the base size of the file or directory before replication.

3. dfs -ls - HDFS ls command is used to display the list of Files and Directories in HDFS, This ls command shows the files with permissions, user, group, and other details

```
[hdfs@sandbox-hdp ~]$ hdfs dfs -ls /
Found 12 items
drwxrwxrwx   - yarn   hadoop          0 2018-06-18 15:18 /app-logs
drwxr-xr-x   - hdfs   hdfs            0 2018-06-18 16:13 /apps
drwxr-xr-x   - yarn   hadoop          0 2018-06-18 14:52 /ats
drwxr-xr-x   - hdfs   hdfs            0 2018-06-18 14:52 /hdp
drwx------   - livy   hdfs            0 2018-06-18 15:11 /livy2-recovery
drwxr-xr-x   - mapred hdfs            0 2018-06-18 14:52 /mapred
drwxrwxrwx   - mapred hadoop          0 2018-06-18 14:52 /mr-history
drwxr-xr-x   - hdfs   hdfs            0 2018-06-18 15:59 /ranger
drwxrwxrwx   - spark  hadoop          0 2023-02-05 23:11 /spark2-history
drwxr-xr-x   - root   hdfs            0 2023-02-05 22:40 /test_data
drwxrwxrwx   - hdfs   hdfs            0 2018-06-18 16:06 /tmp
drwxr-xr-x   - hdfs   hdfs            0 2018-06-18 16:08 /user
[hdfs@sandbox-hdp ~]$ hdfs dfs -copyFromLocal test_file.rar /test_data/
```

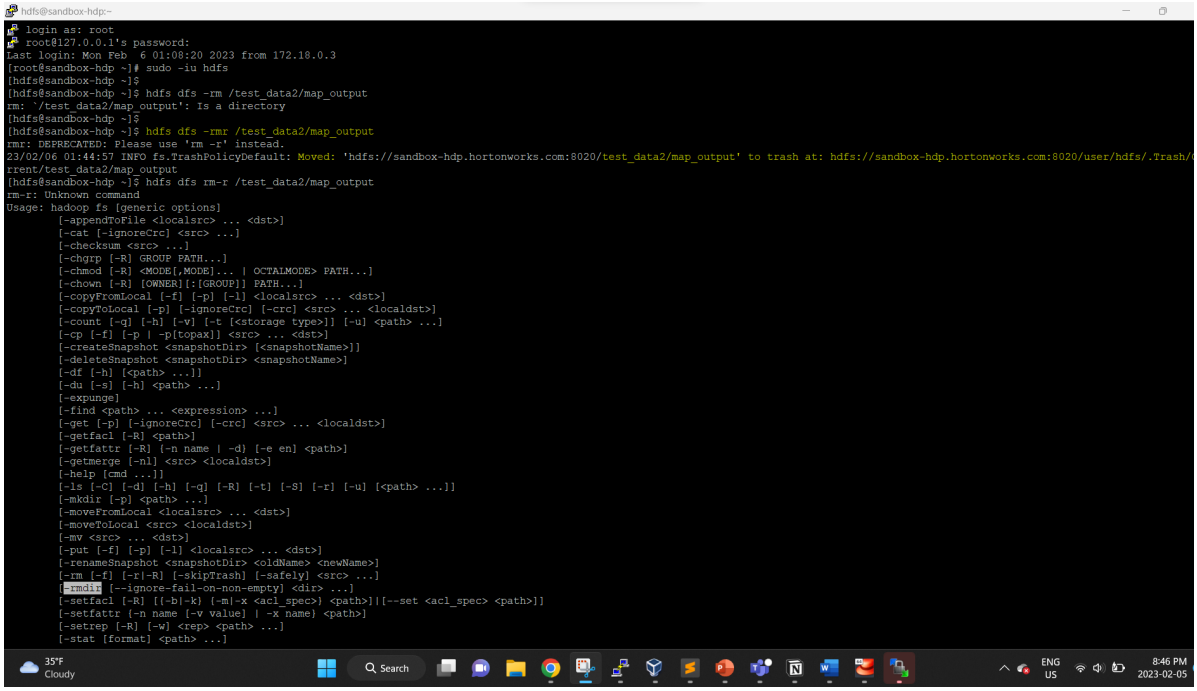
4. dfs -mkdir - HDFS `mkdir` command is used to create a directory in HDFS. By default, this directory would be owned by the user who is creating it.

```
[hdfs@sandbox-hdp /]$ hdfs dfs -mkdir /test_data2
[hdfs@sandbox-hdp /]$ hdfs dfs -ls /
Found 13 items
drwxrwxrwx   - yarn   hadoop          0 2018-06-18 15:18 /app-logs
drwxr-xr-x   - hdfs   hdfs            0 2018-06-18 16:13 /apps
drwxr-xr-x   - yarn   hadoop          0 2018-06-18 14:52 /ats
drwxr-xr-x   - hdfs   hdfs            0 2018-06-18 14:52 /hdp
drwx------   - livy   hdfs            0 2018-06-18 15:11 /livy2-recovery
drwxr-xr-x   - mapred hdfs            0 2018-06-18 14:52 /mapred
drwxrwxrwx   - mapred hadoop          0 2018-06-18 14:52 /mr-history
drwxr-xr-x   - hdfs   hdfs            0 2018-06-18 15:59 /ranger
drwxrwxrwx   - spark  hadoop          0 2023-02-05 23:26 /spark2-history
```

5. dfs put - Copy file/folder from local disk to HDFS. On `put` command specifies the `local-file-path` where you wanted to copy from and then `hdfs-file-path` where you wanted to copy to on hdfs.

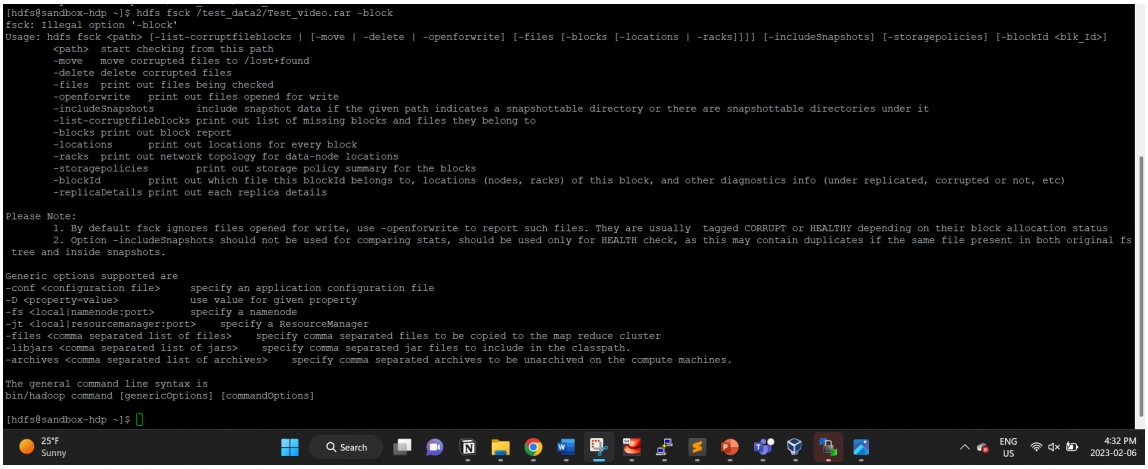
```
[hdfs@sandbox-hdp ~]$
[hdfs@sandbox-hdp ~]$
[hdfs@sandbox-hdp ~]$ hdfs dfs -put dog.jfif /tmp
[hdfs@sandbox-hdp ~]$ hdfs dfs -ls /tmp
Found 4 items
-rw-r--r--   1 hdfs      hdfs        14505 2023-02-06 01:49 /tmp/dog.jfif
drwxrwxr-x   - druid     hadoop          0 2018-06-18 16:06 /tmp/druid-indexing
drwxr-xr-x   - hdfs      hdfs            0 2018-06-18 14:52 /tmp/entity-file-history
drwx-wx-wx   - ambari-qa hdfs            0 2018-06-18 15:12 /tmp/hive
[hdfs@sandbox-hdp ~]$
```

6. dfs rmr - HDFS rm command deletes a file and a directory from HDFS recursively.

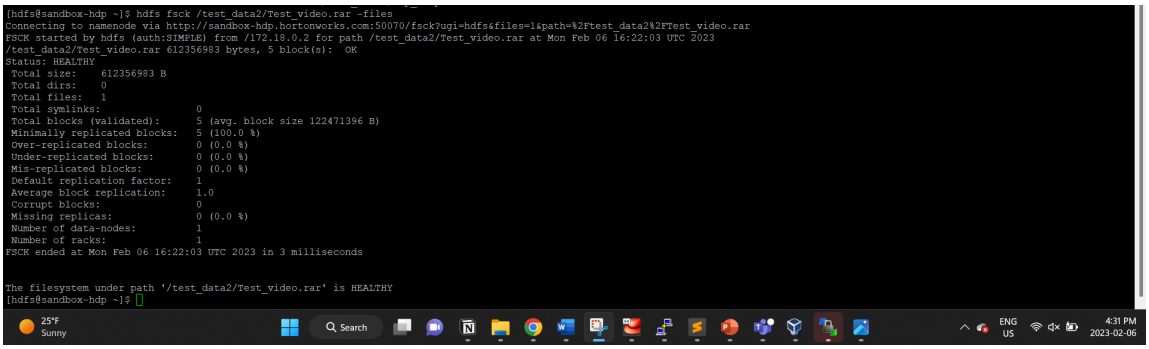


FSCK Commands

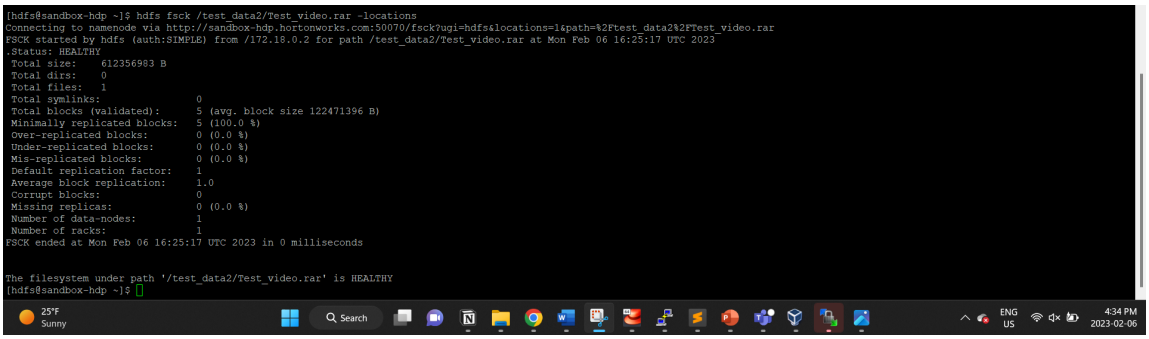
1. fsck block - Print out block report.



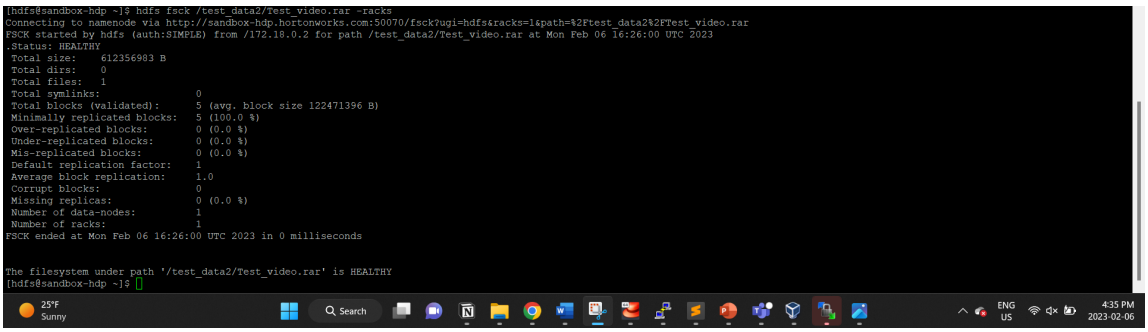
2. fsck file - Print out files being checked.



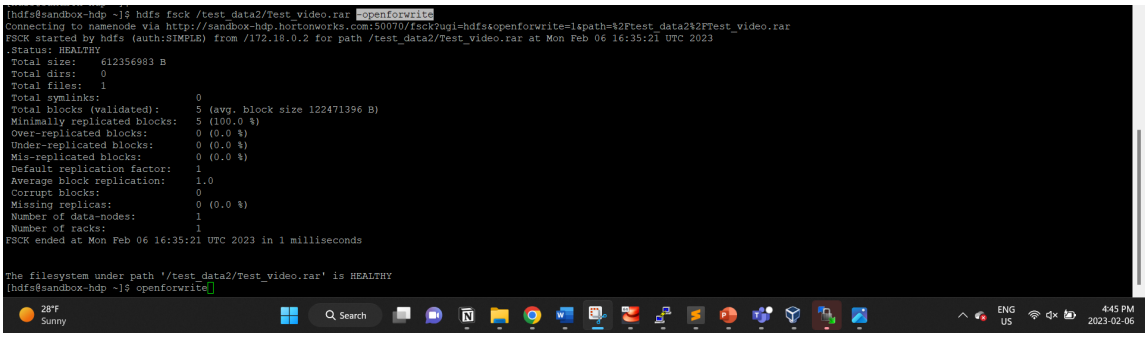
3. fsck location - Print out locations for every block.



4. fsck racks - Print out network topology for data-node locations.

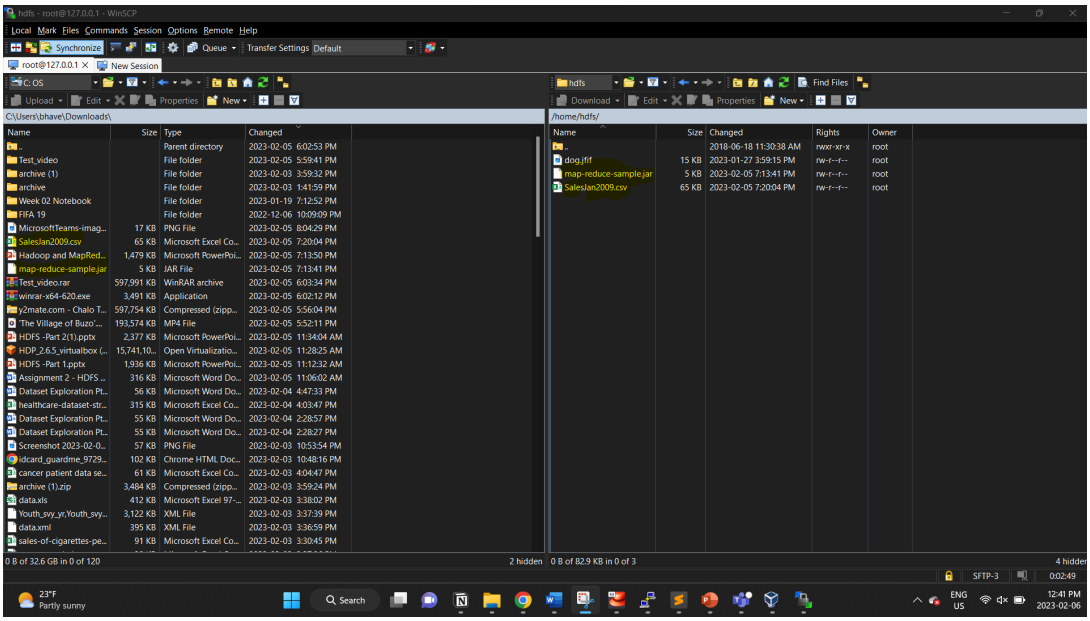


5. fsck openforwrite - Print out files opened for write.



Part 2 - Run Map Reduce Jar File

1. Use WinSCP File Transfer to Local System.
    a. Download the jar file and the csv file from the blackboard link.
    b. Transfer the file from your windows to local machine via winscp to home/hdfs folder check for files with ls command.

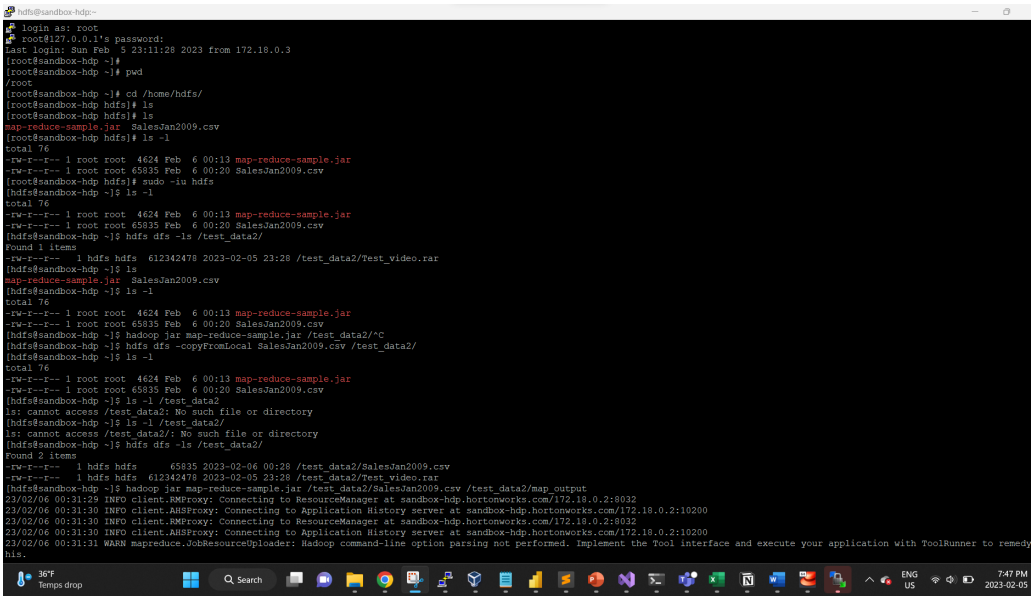


2. Login to putty and run the following commands

a. login into hdfs - sudo -iu hdfs

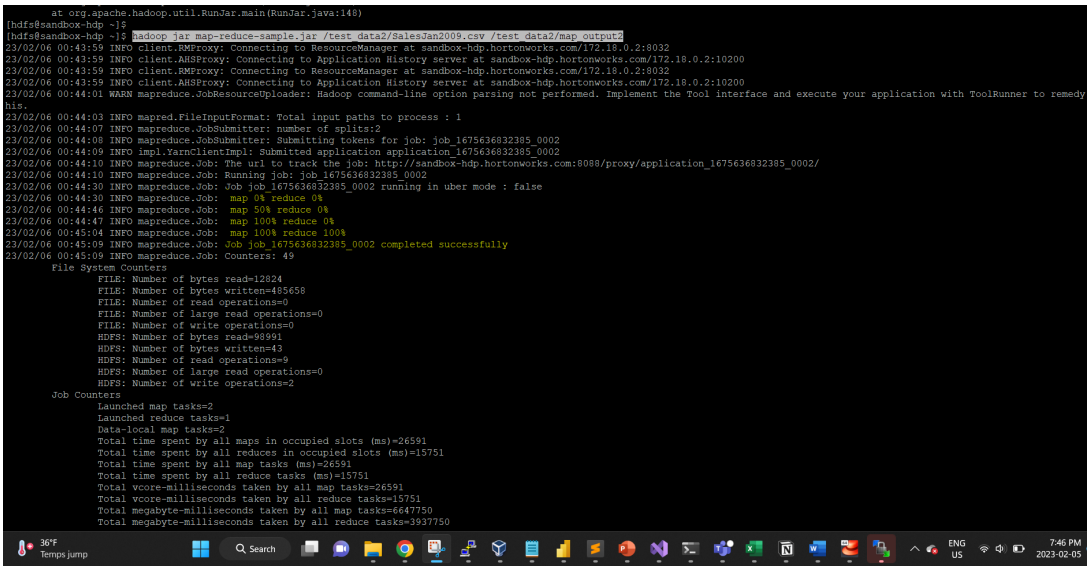
b. Check the contents of the folder test_data - hdfs dfs -ls /test_data/

c. Move the csv file to test_data folder - hdfs dfs -copyFromLocal SalesJan2009.csv /test_data/



d. Check the contents of the folder test_data - hdfs dfs -ls /test_data/

3. Run the following command to get the output - hadoop jar map-reduce-sample.jar /test_data2/SalesJan2009.csv /test_data/map_output


at org.apache.hadoop.util.RunJar.main(RunJar.java:148)
[hdfs@sandbox-hdp ~]$
[hdfs@sandbox-hdp ~]$ hadoop jar map-reduce-sample.jar /test_data2/SalesJan2009.csv /test_data2/map_output2
23/02/06 00:43:59 INFO client.RMProxy: Connecting to ResourceManager at sandbox-hdp.hortonworks.com/172.18.0.2:8032
23/02/06 00:43:59 INFO client.AHSProxy: Connecting to Application History server at sandbox-hdp.hortonworks.com/172.18.0.2:10200
23/02/06 00:43:59 INFO client.RMProxy: Connecting to ResourceManager at sandbox-hdp.hortonworks.com/172.18.0.2:8032
23/02/06 00:43:59 INFO client.AHSProxy: Connecting to Application History server at sandbox-hdp.hortonworks.com/172.18.0.2:10200
23/02/06 00:44:01 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy
his
23/02/06 00:44:03 INFO mapred.FileInputFormat: Total input paths to process : 1
23/02/06 00:44:07 INFO mapreduce.JobSubmitter: number of splits:2
23/02/06 00:44:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1675636832385_0002
23/02/06 00:44:09 INFO impl.YarnClientImpl: Submitted application application_1675636832385_0002
23/02/06 00:44:10 INFO mapreduce.Job: The url to track the job: http://sandbox-hdp.hortonworks.com:8088/proxy/application_1675636832385_0002/
23/02/06 00:44:10 INFO mapreduce.Job: Running job: job_1675636832385_0002
23/02/06 00:44:30 INFO mapreduce.Job: Job job_1675636832385_0002 running in uber mode : false
23/02/06 00:44:30 INFO mapreduce.Job: map 0% reduce 0%
23/02/06 00:44:46 INFO mapreduce.Job: map 50% reduce 0%
23/02/06 00:44:47 INFO mapreduce.Job: map 100% reduce 0%
23/02/06 00:45:04 INFO mapreduce.Job: map 100% reduce 100%
23/02/06 00:45:09 INFO mapreduce.Job: Job job_1675636832385_0002 completed successfully
23/02/06 00:45:09 INFO mapreduce.Job: Counters: 49
File System Counters
FILE: Number of bytes read=12824
FILE: Number of bytes written=485658
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=98991
HDFS: Number of bytes written=43
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)=26591
Total time spent by all reduces in occupied slots (ms)=15751
Total time spent by all map tasks (ms)=26591
Total time spent by all reduce tasks (ms)=15751
Total vcore-milliseconds taken by all map tasks=26591
Total vcore-milliseconds taken by all reduce tasks=15751
Total megabyte-milliseconds taken by all map tasks=6647750
Total megabyte-milliseconds taken by all reduce tasks=3937750
36°F
Temps jump
Search
ENG
US
7:46 PM
2023-02-05