VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

BIG DATA ANALYTICS

(20CS6PEBDA)

Submitted by

SHIVANI GAHLOT (1BM19CS150)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

May-2022 to July-2022

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**BIG DATA ANALYTICS**" carried out by **SHIVANI GAHLOT (1BM19CS150),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of Big Data Analytics **- (20CS6PEBDA)** work prescribed for the said degree.

Antara Roy Choudhury Assistant Professor Department of CSE BMSCE, Bengaluru

Dr. Jyothi S Nayak Professor and Head Department of CSE BMSCE, Bengaluru

`

Course Outcome

CO1	Apply the concept of NoSQL, Hadoop or Spark for a given task
CO2	Analyze the Big Data and obtain insight using data analytics mechanisms.
CO3	Design and implement Big data applications by applying NoSQL, Hadoop or Spark

1. MongoDB:

I. CREATE DATABASE IN MONGODB.

use myDB;

Confirm the existence of your database

db;

To list all databases

show dbs;

II. CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS

1. To create a collection by the name "Student".

db.createCollection(“Student”);

2. To drop a collection by the name “Student”.

db.Student.drop();

3. Create a collection by the name “Students” and store the following data in it.

db.Student.insert({_id:1,StudName:”MichelleJacintha”,Grade:”VII”,Hobbies:”InternetS

urfing”});

4. Insert the document for “AryanDavid” in to the Students collection only if it does not already exist in the collection. However, if it is already present in the collection, then update the document with new values. (Update his hobbies from “Skating” to “Chess”). Use “Update else insert” (if there is an existing document, it will attempt to update it, if there is no existing document then it will insert it).

db.Student.update({_id:3,StudName:”AryanDavid”,Grade:”VII”},{$set:{Hobbies:”Skatin

g”}},{upsert:true});

5. FIND METHOD

A. To search for documents from the “Students” collection based on certain search

criteria.

db.Student.find({StudName:”Aryan David”});

B. To display only the StudName and Grade from all the documents of the Students

collection. The identifier_id should be suppressed and NOT displayed.

db.Student.find({},{StudName:1,Grade:1,_id:0});

C. To find those documents where the Grade is set to ‘VII’

db.Student.find({Grade:{$eq:’VII’}}).pretty();

D. To find those documents from the Students collection where the Hobbies is set to

either ‘Chess’ or is set to ‘Skating’.

db.Student.find({Hobbies :{ $in: [’Chess’,’Skating’]}}).pretty ();

E. To find documents from the Students collection where the StudName begins with “M”.

db.Student.find({StudName:/^M/}).pretty();

F. To find documents from the Students collection where the StudNamehas an “e” in any position.

db.Student.find({StudName:/e/}).pretty();

G. To find the number of documents in the Students collection.

db.Student.count();

H. To sort the documents from the Students collection in the descending order of StudName.

db.Student.find().sort({StudName:-1}).pretty();

III. Import data from a CSV file

Given a CSV file "sample.txt" in the D:drive, import the file into the MongoDB collection, "SampleJSON". The collection is in the database "test".

mongoimport --db Student --collection airlines --type csv –headerline --file /home/hduser/Desktop/airline.csv

IV. Export data to a CSV file

This command used at the command prompt exports MongoDB JSON documents from "Customers" collection in the "test" database into a CSV file "Output.txt" in the D:drive.

mongoexport --host localhost --db Student --collection airlines --csv --out /home/hduser/Desktop/output.txt –fields "Year","Quarter"

V. Save Method :

Save() method will insert a new document, if the document with the _id does not exist. If it exists it will replace the exisiting document.

db.Students.save({StudName:"Vamsi", Grade:"VI"})

VI. Add a new field to existing Document:

db.Students.update({_id:4},{$set:{Location:"Network"}})

VII. Remove the field in an existing Document

db.Students.update({_id:4},{$unset:{Location:"Network"}})

VIII. Finding Document based on search criteria suppressing few fields

db.Student.find({_id:1},{StudName:1,Grade:1,_id:0});

To find those documents where the Grade is not set to 'VII'

db.Student.find({Grade:{$ne:'VII'}}).pretty();

To find documents from the Students collection where the StudName ends with s.

db.Student.find({StudName:/s$/}).pretty();

IX. to set a particular field value to NULL

db.Students.update({_id:3},{$set:{Location:null}})

X. Count the number of documents in Student Collections

db.Students.count()

XI. Count the number of documents in Student Collections with grade :VII

db.Students.count({Grade:"VII"})

Retrieve first 3 documents

db.Students.find({Grade:"VII"}).limit(3).pretty();

Sort the document in Ascending order

db.Students.find().sort({StudName:1}).pretty();

for descending order:

db.Students.find().sort({StudName:-1}).pretty();

to Skip the 1st two documents from the Students Collections

db.Students.find().skip(2).pretty()

XII. Create a collection by name "food" and add to each document add a "fruits" array

db.food.insert({ _id:1, fruits:['grapes','mango','apple'] })

db.food.insert({ _id:2, fruits:['grapes','mango','cherry'] })

db.food.insert({ _id:3, fruits:['banana','mango'] })

To find those documents from the "food" collection which has the "fruits array" constitute of "grapes", "mango" and "apple".

db.food.find ({fruits: ['grapes','mango','apple'] }). pretty().

To find in "fruits" array having "mango" in the first index position.
db.food.find ({'fruits.1':'grapes'})

To find those documents from the "food" collection where the size of the array is two.

db.food.find ({"fruits": {$size:2}})

To find the document with a particular id and display the first two elements from the array "fruits"

db.food.find({_id:1},{"fruits":{$slice:2}})

To find all the documets from the food collection which have elements mango and grapes in the array "fruits"

db.food.find({fruits:{$all:["mango","grapes"]}})

Update on Array:

Using particular id replace the element present in the 1 st index position of the fruits array with apple

db.food.update({_id:3},{$set:{'fruits.1':'apple'}})

Insert new key value pairs in the fruits array

db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}}

) **XII. Aggregate Function :**

Create a collection Customers with fields custID, AcctBal, AcctType.

Now group on "custID" and compute the sum of "AccBal".

db.Customers.aggregate ({$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } });

Match on AcctType:"S" then group on "CustID" and compute the sum of "AccBal".

db.Customers.aggregate ({$match:{AcctType:"S"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } });

Match on AcctType:"S" then group on "CustID" and compute the sum of "AccBal" and total balance greater than 1200.

db.Customers.aggregate ({$match:{AcctType:"S"}},{$group : { _id : "$custID",TotAccBal : {$sum:"$AccBal"} } }, {$match:{TotAccBal:{$gt:1200}}});

2. Perform the following DB operations using Cassandra.

1.Create a keyspace by name Employee

```
CREATE KEYSPACE employee123 WITH REPLICATION =
{'class':'SimpleStrategy','replication_factor':1};
```

2. Create a column family by name Employee-Info with attributes

Emp_Id Primary Key, Emp_Name,

Designation, Date_of_Joining, Salary,

Dept_Name

```
CREATE TABLE EMPLOYEEINFO( EMPID INT PRIMARY KEY, EMPNAME TEXT, DESIGNATION TEXT, DATEOFJOINING
TIMESTAMP, SALARY DOUBLE, DEPTNAME TEXT);
```

3. Insert the values into the table in batch

Begin Batch

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY,

DEPTNAME) VALUES(1,'ABHISHEK','ASSISTANT MANAGER', '2010-04-26', 75000, 'MARKETING')

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY, DEPTNAME) VALUES(2,'BHASKAR','ASSISTANT MANAGER', '2010-04-26', 75000, 'MARKETING')

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY, DEPTNAME) VALUES(3,'CHIRAG','ASSISTANT MANAGER', '2010-04-26', 75000, 'MARKETING')

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY, DEPTNAME)VALUES(4,'DHANUSH','ASSISTANT MANAGER', '2010-04-26', 75000, 'MARKETING')

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY, DEPTNAME) VALUES(5,'ESHAAN','ASSISTANT MANAGER', '2010-04-26', 85000, 'TECHNICAL')

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY, DEPTNAME) VALUES(6,'FARAH','MANAGER', '2010-04-26', 95000, 'TECHNICAL')

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY, DEPTNAME) VALUES(7,'GEMMA','MANAGER', '2010-04-26', 95000, 'PR')

INSERT INTO EMPLOYEEINFO (EMPID, EMPNAME, DESIGNATION, DATEOFJOINING, SALARY, DEPTNAME)VALUES(121,'HARRY','REGIONAL MANAGER', '2010-04-26', 99000, 'MANAGEMENT')

APPLY BATCH;

SELECT * FROM EMPLOYEEINFO;

empid | dateofjoining | deptname | designation | empname | salary

-------+---------------------------------+-----------+--------------------+----------+--------
5 | 2010-04-25 18:30:00.000000+0000 | TECHNICAL | ASSISTANT MANAGER | ESHAAN | 85000 1 |

2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | ABHISHEK | 75000 2 |

2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | BHASKAR | 75000 4 |

2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | DHANUSH | 75000 121 |

2010-04-25 18:30:00.000000+0000 | MANAGEMENT | REGIONAL MANAGER | HARRY | 99000 7 |

2010-04-25 18:30:00.000000+0000 | PR | MANAGER | GEMMA | 95000

6 | 2010-04-25 18:30:00.000000+0000 | TECHNICAL | MANAGER | FARAH | 95000

3 | 2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | CHIRAG | 75000 **4.**

Update Employee name and Department of Emp-Id 121

UPDATE EMPLOYEEINFO SET EMPNAME='HARISH', DEPTNAME='PR' WHERE EMPID=121;

5. Sort the details of Employee records based on salary

SELECT * FROM EMPLOYEE_IN WHERE EMP_ID IN(1,2,3,4) ORDER BY SALARY DESC ALLOW FILTERING;

6. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.

ALTER TABLE EMPLOYEEINFO ADD PROJECTS LIST<TEXT>;

7. Update the altered table to add project names.

UPDATE EMPLOYEEINFO SET PROJECTS=['FACEBOOK','SNAPCHAT'] WHERE EMPID=1;

UPDATE EMPLOYEEINFO SET PROJECTS=['FACEBOOK','SNAPCHAT'] WHERE EMPID=7;

UPDATE EMPLOYEEINFO SET PROJECTS=['PINTEREST','INSTAGRAM'] WHERE EMPID=121;

UPDATE EMPLOYEEINFO SET PROJECTS=['PINTEREST','INSTAGRAM'] WHERE EMPID=4;

UPDATE EMPLOYEEINFO SET PROJECTS=['YOUTUBE','SPOTIFY'] WHERE EMPID=2;

UPDATE EMPLOYEEINFO SET PROJECTS=['YOUTUBE','SPOTIFY'] WHERE EMPID=3;

UPDATE EMPLOYEEINFO SET PROJECTS=['YOUTUBE','SPOTIFY'] WHERE EMPID=6;

UPDATE EMPLOYEEINFO SET PROJECTS=['TWITTER','REDDIT'] WHERE EMPID=5; **SELECT * FROM EMPLOYEEINFO;**

empid | dateofjoining | deptname | designation | empname | projects | salary

--------+----------+------------+----------+----------+----------+--------

5 | 2010-04-25 18:30:00.000000+0000 | TECHNICAL | ASSISTANT MANAGER | ESHAAN | ['TWITTER', 'REDDIT'] | 85000
1 | 2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | ABHISHEK | ['FACEBOOK', 'SNAPCHAT'] | 75000

2 | 2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | BHASKAR | ['YOUTUBE', 'SPOTIFY'] | 75000

4 | 2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | DHANUSH | ['PINTEREST', 'INSTAGRAM'] | 75000

121 | 2010-04-25 18:30:00.000000+0000 | PR | REGIONAL MANAGER | HARISH | ['PINTEREST', 'INSTAGRAM'] | 99000

7 | 2010-04-25 18:30:00.000000+0000 | PR | MANAGER | GEMMA | ['FACEBOOK', 'SNAPCHAT'] | 95000

6 | 2010-04-25 18:30:00.000000+0000 | TECHNICAL | MANAGER | FARAH | ['YOUTUBE', 'SPOTIFY'] | 95000

3 | 2010-04-25 18:30:00.000000+0000 | MARKETING | ASSISTANT MANAGER | CHIRAG | ['YOUTUBE', 'SPOTIFY'] | 75000

3. Perform the following DB operations using Cassandra.

1.Create a keyspace by name Library

```
CREATE KEYSPACE Library WITH REPLICATION =
        {'class':'SimpleStrategy','replication_factor':1};
```

2. Create a column family by

name Library-Info with attributes

Stud_Id Primary Key,

Counter_value of type Counter,

Stud_Name, Book-Name, Book-Id,

Date_of_issue

CREATE TABLE LIBRARY_INFO_4 (STUD_ID INT, COUNTER_VALUE COUNTER, STUD_NAME TEXT, BOOK_NAME TEXT, BOOK_ID INT, DATE_OF_ISSUE TIMESTAMP, PRIMARY KEY(STUD_ID, STUD_NAME, BOOK_NAME, BOOK_ID, DATE_OF_ISSUE));

3. Insert the values into the table in batch

UPDATE LIBRARY_INFO_4 SET COUNTER_VALUE+1 WHERE STUD_ID=121 AND STUD_NAME='SNEHA' AND BOOK_NAME='BDA' AND BOOK_ID=110 AND DATE_OF_ISSUE='2022-04-01';

UPDATE LIBRARY_INFO_4 SET COUNTER_VALUE+1 WHERE STUD_ID=122 AND STUD_NAME='RAHUL' AND BOOK_NAME='OOMD' AND BOOK_ID=111 AND DATE_OF_ISSUE='2022-07-03';

UPDATE LIBRARY_INFO_4 SET COUNTER_VALUE+1 WHERE STUD_ID=123 AND STUD_NAME='RITIKA' AND BOOK_NAME='ML' AND BOOK_ID=112 AND DATE_OF_ISSUE='2022-02-21';

UPDATE LIBRARY_INFO_4 SET COUNTER_VALUE+1 WHERE STUD_ID=124 AND STUD_NAME='ISHA' AND BOOK_NAME='AI' AND BOOK_ID=113 AND DATE_OF_ISSUE='2022-09-02';

4. Display the details of the table created and increase the value of the counter.

SELECT * FROM LIBRARY_INFO_4;

5. Write a query to show that a student with id 112 has taken a book "BDA" 2 times.

SELECT * FROM LIBRARY_INFO_4 WHERE STUD_ID=112;

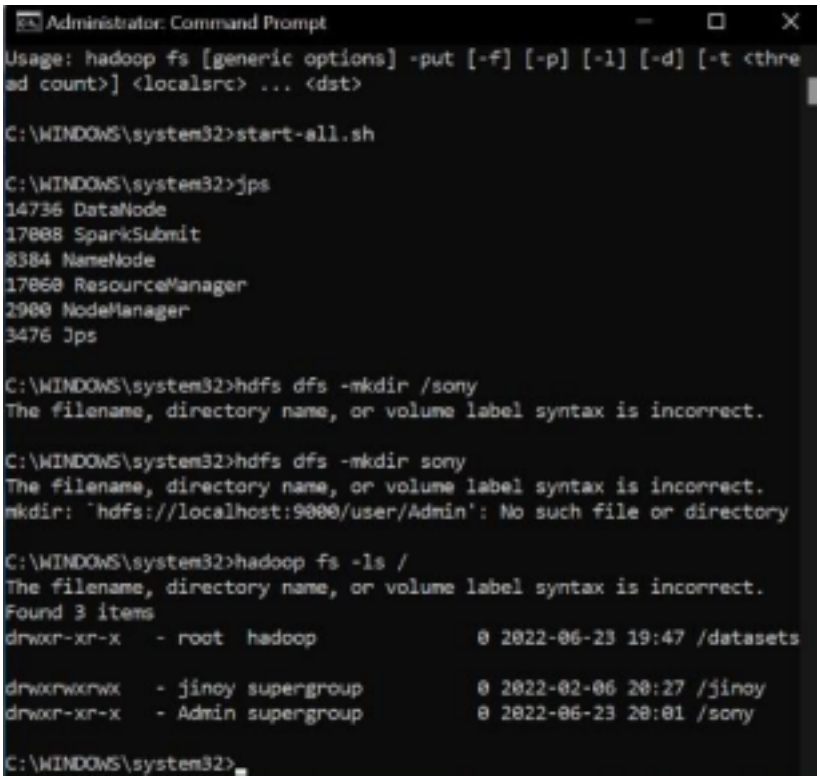
6. Export the created column to a csv file.

COPY LIBRARY_INFO_4 (STUD_ID, STUD_NAME, BOOK_NAME, BOOK_ID, DATE_OF_ISSUE, COUNTER_VALUE) TO 'C:\Users\Admin\OneDrive\Desktop\BDA Lab\data.csv';

7. Import a given csv dataset from local file system into Cassandra column family.

COPY LIBRARY_INFO_4 (STUD_ID, STUD_NAME, BOOK_NAME, BOOK_ID, DATE_OF_ISSUE, COUNTER_VALUE) FROM 'C:\Users\Admin\OneDrive\Desktop\BDA Lab\data.csv';

Lab Program 4:Hadoop Installation



Lab Program 5: Execution of HDFS Commandsfor interaction with HadoopEnvironment. (Minimum 10 commands to be executed)

c:\hadoop_new\sbin>hdfs dfs-mkdir/temp

c:\hadoop_new\sbin>hdfs dfs-copyFromLocal E:\Desktop\sample.txt\temp

c:\hadoop_new\sbin>hdfs dfs -ls \temp

Found 1 items

-rw-r--r-- 1 Admin supergroup 11 2021-06-11 21:12

/temp/sample.txtc:\hadoop_new\sbin>hdfs dfs -cat \temp\sample.txt hello

world

c:\hadoop_new\sbin>hdfs dfs-get\temp\sample.txt E:\Desktop\tempc:\hadoop_new\sbin>hdfs dfs -put E:\Desktop\temp \temp c:\hadoop_new\sbin>hdfs dfs -ls \temp

Found 2 items

-rw-r--r-- 1 Admin supergroup 11 2021-06-11 21:12 /temp/sample.txt drwxr-xr-x -

Admin supergroup 0 2021-06-11 21:15 /temp/temp c:\hadoop_new\sbin>hdfs dfs -

mv \lab1 \temp c:\hadoop_new\sbin>hdfs dfs-ls\temp Found 3 items drwxr-xr-x -

Adminsupergroup 0 2021-04-19 15:07 /temp/lab1 -rw-r--r-- 1 Admin 7 supergroup 11

2021-06-11 21:12 /temp/sample.txt drwxr-xr-x -

Admin supergroup 0 2021-06-11 21:15 /temp/temp

c:\hadoop_new\sbin>hdfs dfs -rm /temp/sample.txtDeleted

/temp/sample.txt

c:\hadoop_new\sbin>hdfs dfs-ls\temp Found 2 items drwxr-xr-x – Admin supergroup 0 2021-04-19 15:07 /temp/lab1 drwxr-xr-x – Admin supergroup 0 2021-06-11 21:15 /temp/temp

c:\hadoop_new\sbin>hdfs dfs-copyFromLocal E:\Desktop\sample.txt\temp

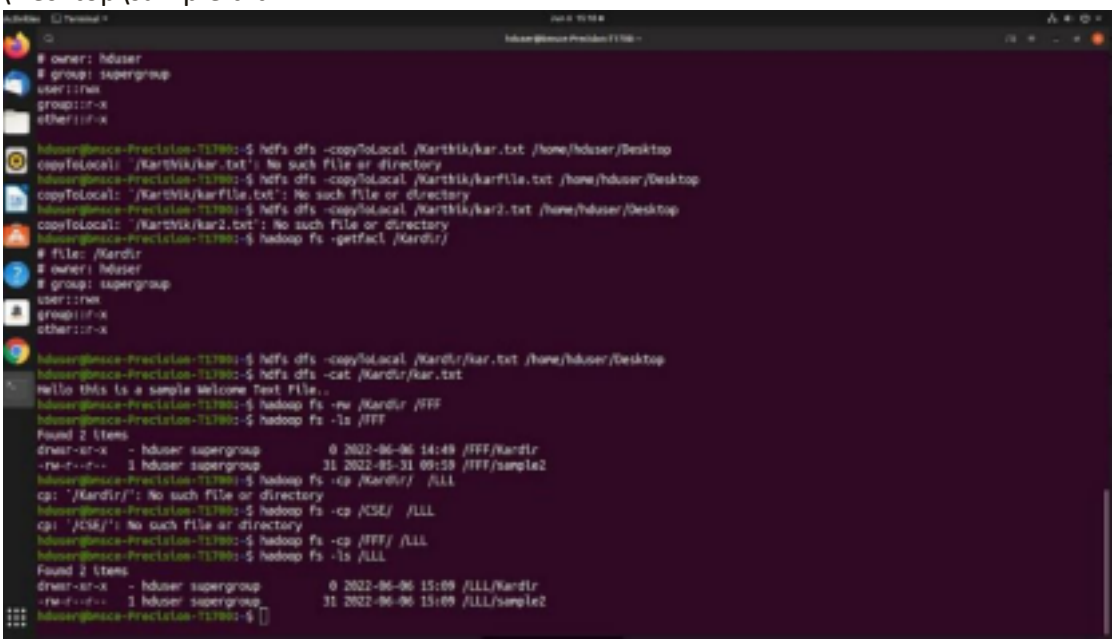
c:\hadoop_new\sbin>hdfs dfs -ls \temp Found 3 items drwxr-xr-x - Admin supergroup 0 2021-04-

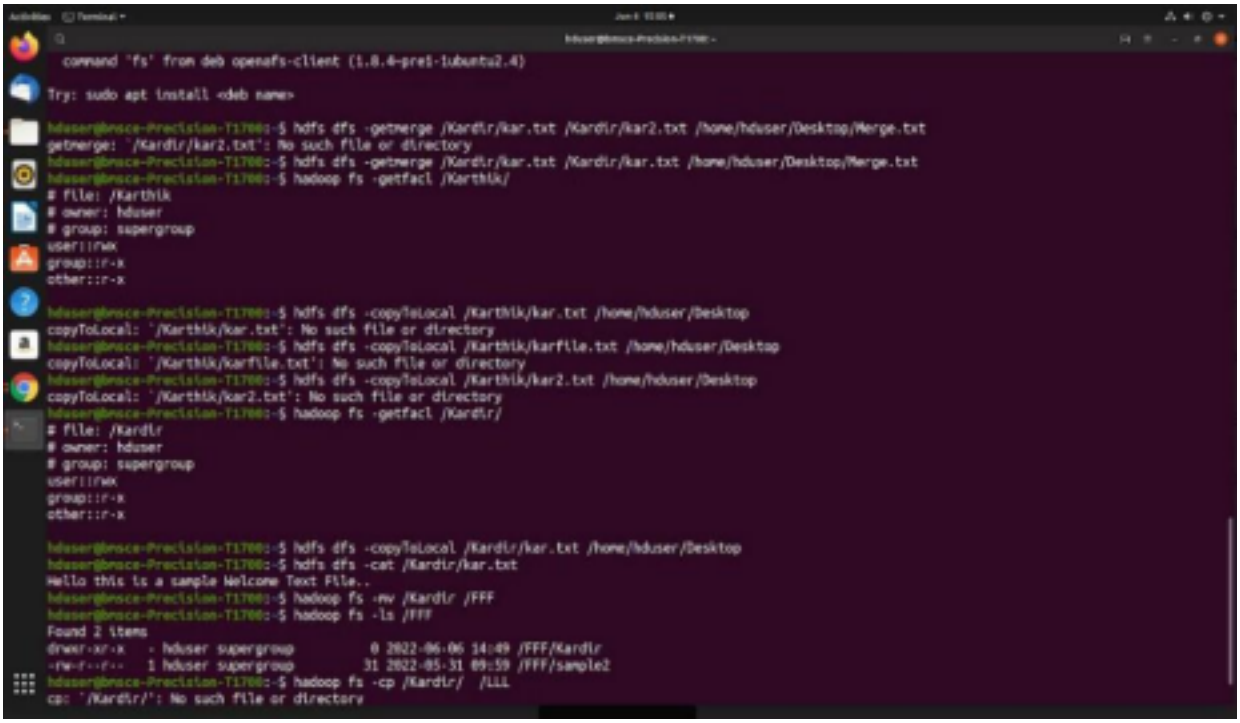
19 15:07 /temp/lab1 -rw-r--r-- 1 Admin supergroup

11 2021-06-11 21:17 /temp/sample.txt drwxr-xr-x - Admin supergroup 0

2021-06-11 21:15 /temp/temp

c:\hadoop_new\sbin>hdfs dfs -copyToLocal \temp\sample.txt E:\Desktop\sample.txt





LAB 6 : For the given file, Create a Map Reduce program to

a) Find the average temperature for each year from the NCDC data set.

· **Program**

AverageDriver

package temp;

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.lib.input.FileInputFormat; import

org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;public class

AverageDriver {

public static void main(String[] args) throws Exception {if

(args.length != 2) {

System.err.println("Please Enterthe input and outputparameters");

System.exit(-1);
}

```
Job job = new Job(); job.setJarByClass(AverageDriver.class);

job.setJobName(“Max temperature”);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.setMapperClass(AverageMapper.class);

job.setReducerClass(AverageReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class); System.exit(job.waitForCompletion(true) ? 0 : 1);

}
}
```

AverageMapper

```
package temp;
importjava.io.IOException;

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 AverageMapper extends Mapper&lt;LongWritable,

Text,Text, IntWritable&gt; {

public static final int MISSING = 9999;

public void map(LongWritable key, Text value,
Mapper&lt;LongWritable, Text, Text, IntWritable&gt;.Context

context)throws IOException, InterruptedException {

int temperature;

String line = value.toString();

String year = line.substring(15, 19); if

(line.charAt(87) == &#39;+&#39;) {

temperature = Integer.parseInt(line.substring(88, 92));
```

```
} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);
if (temperature != 9999&amp;amp;&amp;amp; quality.matches(&amp;quot;[01459]&amp;quot;))context.write(new
Text(year), new
IntWritable(temperature));
}
}
```

AverageReducer

```
package temp;
importjava.io.IOException;
import org.apache.hadoop.io.IntWritable;import
org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class AverageReducer extends Reducer&amp;lt;Text,
IntWritable,Text, IntWritable&amp;gt; {
public void reduce(Text key, Iterable&amp;lt;IntWritable&amp;gt; values,
Reducer&amp;lt;Text, IntWritable, Text, IntWritable&amp;gt;.Context
context)throws IOException, InterruptedException {
int max_temp = 0;int
count = 0;
for (IntWritable value : values) {
max_temp += value.get(); count++;
}
context.write(key, new IntWritable(max_temp / count));
```

}

}

·**Output**

hduser@bmsce-Precision-T1700:~$ sudo su hduser[sudo]

password for hduser:

hduser@bmsce-Precision-T1700:~$ start-all.sh

Thisscript is Deprecated. Instead use start-dfs.sh and start-yarn.shStarting

namenodes on [localhost]

hduser@localhost's password:

localhost:starting namenode, logging to
/usr/local/hadoop/logs/hadoop-hduser-namenode-bmsce-Precision-T1700.ou

t hduser@localhost's password:

localhost:starting datanode, logging to
/usr/local/hadoop/logs/hadoop-hduser-datanode-bmsce-Precision-T1700.outStarting

secondary namenodes [0.0.0.0]

hduser@0.0.0.0's password:

0.0.0.0:starting secondarynamenode, logging to
/usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-bmsce-Precision-T1700.ou

t starting yarn daemons

starting resourcemanager, logging to
/usr/local/hadoop/logs/yarn-hduser-resourcemanager-bmsce-Precision-T1700.ou

t hduser@localhost's password:

localhost:starting nodemanager, logging to
/usr/local/hadoop/logs/yarn-hduser-nodemanager-bmsce-Precision-T1700.ou

t hduser@bmsce-Precision-T1700:~$ jps

7376 DataNode

8212 Jps

8090 NodeManager
3725 org.eclipse.equinox.launcher_1.5.600.v20191014-2022.jar

7758 ResourceManager

7199 NameNode

7599 SecondaryNameNode

hduser@bmsce-Precision-T1700:~$ hadoop fs-mkdir/input_kundana

hduser@bmsce-Precision-T1700:~$ hadoop fs-put Downloads/1901
/input_kundana/1901.txt

hduser@bmsce-Precision-T1700:~$ hadoop jar Desktop/temp.jar
Temperature.AverageDriver /input_kundana/1901.txt/output_1901

Exception in thread "main" java.lang.ClassNotFoundException:
Temperature.AverageDriver

at java.net.URLClassLoader.findClass(URLClassLoader.java:382)at

java.lang.ClassLoader.loadClass(ClassLoader.java:418)

at java.lang.ClassLoader.loadClass(ClassLoader.java:351)at

java.lang.Class.forName0(Native Method)

atjava.lang.Class.forName(Class.java:348)

at org.apache.hadoop.util.RunJar.run(RunJar.java:214)
at org.apache.hadoop.util.RunJar.main(RunJar.java:136)

hduser@bmsce-Precision-T1700:~$ hadoop jar Desktop/temp.jar
AverageDriver /input_kundana/1901.txt/output_1901

22/06/21 10:26:05 INFOConfiguration.deprecation:session.id is deprecated.Instead, use
dfs.metrics.session-id

22/06/21 10:26:05 INFO jvm.JvmMetrics: Initializing JVM Metrics
withprocessName=JobTracker, sessionId=

22/06/21 10:26:05 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing
not performed. Implement the Tool interface and execute yourapplication with ToolRunner to
remedy this.

22/06/21 10:26:05 INFO input.FileInputFormat: Total input pathsto process: 122/06/21 10:26:05

INFO mapreduce.JobSubmitter: number of splits:1

22/06/21 10:26:05 INFO mapreduce.JobSubmitter: Submitting tokensfor
job: job_local1195965365_0001

22/06/21 10:26:05 INFO mapreduce.Job: The url to track the

job:http://localhost:8080/ 22/06/21 10:26:05 INFO mapreduce.Job: Running job:

job_local1195965365_0001 22/06/21 10:26:05 INFO mapred.LocalJobRunner:

OutputCommitter set in configNull

22/06/21 10:26:05 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter

22/06/21 10:26:05 INFO mapred.LocalJobRunner: Waiting for map tasks

22/06/21 10:26:05 INFO mapred.LocalJobRunner: Starting task:
attempt_local1195965365_0001_m_000000_0

22/06/21 10:26:05 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]

22/06/21 10:26:05 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/input_kundana/1901.txt:0+888190

22/06/21 10:26:06 INFO mapred.MapTask: (EQUATOR) 0 kvi
26214396(104857584)

22/06/21 10:26:06 INFOmapred.MapTask: mapreduce.task.io.sort.mb: 10022/06/21

10:26:06 INFO mapred.MapTask: soft limit at 83886080

22/06/21 10:26:06 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600 22/06/21 10:26:06

INFO mapred.MapTask: kvstart = 26214396; length = 6553600

22/06/21 10:26:06 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask$MapOutputBuffer

22/06/21 10:26:06 INFO mapred.LocalJobRunner:

22/06/21 10:26:06 INFO mapred.MapTask: Starting flush of map

output22/06/21 10:26:06 INFO mapred.MapTask: Spilling map output

22/06/21 10:26:06 INFO mapred.MapTask: bufstart = 0; bufend = 59076; bufvoid =104857600

22/06/21 10:26:06 INFO mapred.MapTask: kvstart = 26214396(104857584);kvend
= 26188144(104752576); length = 26253/6553600

22/06/21 10:26:06 INFO mapred.MapTask: Finished spill 022/06/21

10:26:06 INFO mapred.Task:
Task:attempt_local1195965365_0001_m_000000_0 is done. And isin the
process of committing

22/06/21 10:26:06 INFO mapred.LocalJobRunner: map

22/06/21 10:26:06 INFO mapred.Task: Task 'attempt_local1195965365_0001_m_000000_0'
done.
22/06/21 10:26:06 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1195965365_0001_m_000000_0

22/06/21 10:26:06 INFO mapred.LocalJobRunner: map task executor

complete.22/06/21 10:26:06 INFO mapred.LocalJobRunner: Waiting for reduce tasks

22/06/21 10:26:06 INFO mapred.LocalJobRunner: Starting task:
attempt_local1195965365_0001_r_000000_0

22/06/21 10:26:06 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]

22/06/21 10:26:06 INFOmapred.ReduceTask: Using
ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@65367f35

22/06/21 10:26:06 INFOreduce.MergeManagerImpl: MergerManager:memoryLimit=349752512,
maxSingleShuffleLimit=87438128, mergeThreshold=230836672, ioSortFactor=10,
memToMemMergeOutputsThreshold=10

22/06/21 10:26:06 INFO reduce.EventFetcher:
attempt_local1195965365_0001_r_000000_0 Thread started: EventFetcher forfetching Map
Completion Events

22/06/21 10:26:06 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of
map attempt_local1195965365_0001_m_000000_0 decomp: 72206len: 72210 to MEMORY

22/06/21 10:26:06 INFO reduce.InMemoryMapOutput: Read 72206
bytesfrommap-output for attempt_local1195965365_0001_m_000000_0

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: closeInMemoryFile ->
map-output of size: 72206, inMemoryMapOutputs.size() -> 1, commitMemory
->0, usedMemory ->72206

22/06/21 10:26:06 INFO reduce.EventFetcher: EventFetcherisinterrupted..Returning

22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: finalMerge called with

1in-memory map-outputs and 0 on-disk map-outputs

22/06/21 10:26:06 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:26:06 INFO mapred.Merger: Down to the last merge-pass, with 1segments left of total size: 72199 bytes

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merged 1 segments, 72206bytes to disk to satisfy reduce memory limit

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 1 files, 72210 bytesfrom disk

22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytesfrom memory into reduce

22/06/21 10:26:06 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:26:06 INFO mapred.Merger: Down to the last merge-pass, with 1segments left of total size: 72199 bytes

22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:26:06 INFOConfiguration.deprecation: mapred.skip.on isdeprecated. Instead, use mapreduce.job.skiprecords

22/06/21 10:26:06 INFO mapred.Task: Task:attempt_local1195965365_0001_r_000000_0 is done. And isin the processof committing

22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:26:06 INFO mapred.Task: Task attempt_local1195965365_0001_r_000000_0 is allowed to commit now

22/06/21 10:26:06 INFO output.FileOutputCommitter: Saved output of task 'attempt_local1195965365_0001_r_000000_0' to hdfs://localhost:54310/output_1901/_temporary/0/task_local1195965365_000 1 _r_000000

22/06/21 10:26:06 INFO mapred.LocalJobRunner: reduce > reduce

22/06/21 10:26:06 INFO mapred.Task: Task 'attempt_local1195965365_0001_r_000000_0' done.

22/06/21 10:26:06 INFO mapred.LocalJobRunner: Finishing task: attempt_local1195965365_0001_r_000000_0

22/06/21 10:26:06 INFO mapred.LocalJobRunner: reduce task executor complete.

22/06/21 10:26:06 INFO mapreduce.Job: Job job_local1195965365_0001 runningin uber mode : false

22/06/21 10:26:06 INFO mapreduce.Job: map 100% reduce 100%

22/06/21 10:26:06 INFO mapreduce.Job: Job job_local1195965365_0001completed successfully
22/06/21 10:26:06 INFO mapreduce.Job: Counters:

38File System Counters

FILE: Number of bytes read=152940 FILE:

Number of bytes written=725372FILE: Number

of read operations=0

FILE: Number of large read operations=0FILE:

Number of write operations=0 HDFS: Number

of bytes read=1776380 HDFS: Number of bytes

written=8

HDFS: Number of read operations=13 HDFS:

Number of large read operations=0HDFS: Number

of write operations=4

Map-Reduce Framework

Map input records=6565 Map

output records=6564Map

output bytes=59076

Map output materialized bytes=72210Input split

bytes=110

Combine input records=0

Combine output records=0

Reduce input groups=1 Reduce

shuffle bytes=72210Reduce

input records=6564 Reduce

output records=1 Spilled

Records=13128 Shuffled Maps

=1

Failed Shuffles=0 Merged
Map outputs=1 GC time

elapsed (ms)=63CPU time

spent (ms)=0

Physical memory (bytes)snapshot=0Virtual

memory (bytes) snapshot=0

Total committed heap usage (bytes)=999292928

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=888190File

Output Format Counters

Bytes Written=8

hduser@bmsce-Precision-T1700:~$ hadoop fs-cat/output_1901/part-r-000001901 46

hduser@bmsce-Precision-T1700:~$

b) <u>find the mean max temperature for every month</u>

· **MeanMaxDriver.class**

package meanmax;

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.lib.input.FileInputFormat; import

org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;public class

MeanMaxDriver {

public static void main(String[] args) throws Exception {if

(args.length != 2) {

System.err.println(&quot;Please Enterthe input and outputparameters&quot;);

System.exit(-1);

}

Job job = new Job(); job.setJarByClass(MeanMaxDriver.class);

job.setJobName(&quot;Max temperature&quot;);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

job.setMapperClass(MeanMaxMapper.class);

job.setReducerClass(MeanMaxReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

System.exit(job.waitForCompletion(true) ? 0 : 1);

}}

MeanMaxMapper.class

package meanmax;


importjava.io.IOException;

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 MeanMaxMapper extends Mapper&lt;LongWritable,

Text,Text, IntWritable&gt; {

public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Mapper&lt;LongWritable, Text,

Text, IntWritable&gt;.Context context)throws IOException, InterruptedException

{

int temperature;

String line = value.toString();

String month = line.substring(19, 21);if

(line.charAt(87) == &#39;+&#39;) {

temperature = Integer.parseInt(line.substring(88, 92));

} else {
temperature = Integer.parseInt(line.substring(87, 92));
}
String quality = line.substring(92, 93);

if (temperature != 9999&amp;&amp; quality.matches(&quot;[01459]&quot;))context.write(new

Text(month), new

IntWritable(temperature));

}

}

MeanMaxReducer.class

package meanmax;

importjava.io.IOException;

import org.apache.hadoop.io.IntWritable;import

org.apache.hadoop.io.Text;
```

```
import org.apache.hadoop.mapreduce.Reducer;

public class MeanMaxReducer extends Reducer&lt;Text,

IntWritable,Text, IntWritable&gt; {

public void reduce(Text key, Iterable&lt;IntWritable&gt; values,

Reducer&lt;Text, IntWritable, Text, IntWritable&gt;.Context

context)throws IOException, InterruptedException {

int max_temp = 0;int

total_temp = 0;

int count = 0;

int days = 0;

for (IntWritable value : values) {int

temp = value.get();

if (temp &gt; max_temp)

max_temp = temp; count++;

if (count == 3) { total_temp +=

max_temp;max_temp = 0;

count = 0;

days++;

}

}

context.write(key, new IntWritable(total_temp / days));

}

}
```

· Output

```
hduser@bmsce-OptiPlex-3060:~$ hadoop jar
/home/hduser/Desktop/mean_max_temp.jar meanmax.MeanMaxDriver
/input_pranav/temp_1901.txt/avg_temp_output_meanmax_1901

22/06/21 10:17:01 INFOConfiguration.deprecation:session.id is deprecated.Instead, use
```

dfs.metrics.session-id

22/06/21 10:17:01 INFO jvm.JvmMetrics: Initializing JVM Metrics withprocessName=JobTracker, sessionId=

22/06/21 10:17:01 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute yourapplication with ToolRunner to remedy this.

22/06/21 10:17:01 INFO input.FileInputFormat: Total input pathsto process: 122/06/21 10:17:01

INFO mapreduce.JobSubmitter: number of splits:1

22/06/21 10:17:01 INFO mapreduce.JobSubmitter: Submitting tokensfor job: job_local232634845_0001

22/06/21 10:17:01 INFO mapreduce.Job: The url to track the

job:http://localhost:8080/ 22/06/21 10:17:01 INFO mapreduce.Job: Running job:

job_local232634845_0001 22/06/21 10:17:01 INFO mapred.LocalJobRunner:

OutputCommitter set in configNull

22/06/21 10:17:01 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Waiting for map tasks

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Starting task: attempt_local232634845_0001_m_000000_0

22/06/21 10:17:01 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ] 22/06/21 10:17:01 INFO mapred.MapTask: Processing split: hdfs://localhost:54310/input_pranav/temp_1901.txt:0+888190

22/06/21 10:17:01 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)

22/06/21 10:17:01 INFOmapred.MapTask: mapreduce.task.io.sort.mb: 10022/06/21

10:17:01 INFO mapred.MapTask: soft limit at 83886080

22/06/21 10:17:01 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/21 10:17:01 INFO mapred.MapTask: kvstart = 26214396; length = 6553600

22/06/21 10:17:01 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer

22/06/21 10:17:01 INFO mapred.LocalJobRunner:

22/06/21 10:17:01 INFO mapred.MapTask: Starting flush of map

output22/06/21 10:17:01 INFO mapred.MapTask: Spilling map output

22/06/21 10:17:01 INFO mapred.MapTask: bufstart = 0; bufend = 45948; bufvoid =104857600

22/06/21 10:17:01 INFO mapred.MapTask: kvstart = 26214396(104857584);kvend
= 26188144(104752576); length = 26253/6553600

22/06/21 10:17:01 INFO mapred.MapTask: Finished spill 022/06/21

10:17:01 INFO mapred.Task:
Task:attempt_local232634845_0001_m_000000_0 is done. And isin the
process of committing

22/06/21 10:17:01 INFO mapred.LocalJobRunner: map

22/06/21 10:17:01 INFO mapred.Task: Task 'attempt_local232634845_0001_m_000000_0' done.
22/06/21 10:17:01 INFO mapred.LocalJobRunner: Finishing task:
attempt_local232634845_0001_m_000000_0

22/06/21 10:17:01 INFO mapred.LocalJobRunner: map task executor

complete.22/06/21 10:17:01 INFO mapred.LocalJobRunner: Waiting for reduce tasks

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Starting task:
attempt_local232634845_0001_r_000000_0

22/06/21 10:17:01 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]

22/06/21 10:17:01 INFOmapred.ReduceTask: Using
ShuffleConsumerPlugin:
org.apache.hadoop.mapreduce.task.reduce.Shuffle@1a055244

22/06/21 10:17:01 INFOreduce.MergeManagerImpl: MergerManager:memoryLimit=349752512,
maxSingleShuffleLimit=87438128, mergeThreshold=230836672, ioSortFactor=10,
memToMemMergeOutputsThreshold=10

22/06/21 10:17:01 INFO reduce.EventFetcher:
attempt_local232634845_0001_r_000000_0 Thread started: EventFetcher forfetching Map
Completion Events

22/06/21 10:17:01 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map
attempt_local232634845_0001_m_000000_0 decomp: 59078 len:59082 to MEMORY

22/06/21 10:17:01 INFO reduce.InMemoryMapOutput: Read 59078

bytesfrommap-output for attempt_local232634845_0001_m_000000_0

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: closeInMemoryFile ->
map-output of size: 59078, inMemoryMapOutputs.size() -> 1, commitMemory
->0, usedMemory ->59078

22/06/21 10:17:01 INFO reduce.EventFetcher: EventFetcherisinterrupted..Returning

22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: finalMerge called with
1in-memory map-outputs and 0 on-disk map-outputs

22/06/21 10:17:01 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:17:01 INFO mapred.Merger: Down to the last merge-pass, with 1segments left
of total size: 59073 bytes

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merged 1 segments, 59078bytes to disk
to satisfy reduce memory limit

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 1 files, 59082 bytesfrom disk

22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytesfrom
memory into reduce

22/06/21 10:17:01 INFO mapred.Merger: Merging 1 sorted segments

22/06/21 10:17:01 INFO mapred.Merger: Down to the last merge-pass, with 1segments left
of total size: 59073 bytes

22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:17:01 INFOConfiguration.deprecation: mapred.skip.on isdeprecated.
Instead, use mapreduce.job.skiprecords

22/06/21 10:17:01 INFO mapred.Task: Task:attempt_local232634845_0001_r_000000_0 is done.
And isin the process ofcommitting

22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:17:01 INFO mapred.Task: Task attempt_local232634845_0001_r_000000_0 is
allowed to commit now

22/06/21 10:17:01 INFO output.FileOutputCommitter: Saved output of task
'attempt_local232634845_0001_r_000000_0' to
hdfs://localhost:54310/avg_temp_output_meanmax_1901/_temporary/0/task_lo
cal232634845_0001_r_000000
22/06/21 10:17:01 INFO mapred.LocalJobRunner: reduce > reduce

22/06/21 10:17:01 INFO mapred.Task: Task 'attempt_local232634845_0001_r_000000_0' done.

22/06/21 10:17:01 INFO mapred.LocalJobRunner: Finishing task: attempt_local232634845_0001_r_000000_0

22/06/21 10:17:01 INFO mapred.LocalJobRunner: reduce task executor complete.

22/06/21 10:17:02 INFO mapreduce.Job: Job job_local232634845_0001 runningin uber mode : false

22/06/21 10:17:02 INFO mapreduce.Job: map 100% reduce 100%

22/06/21 10:17:02 INFO mapreduce.Job: Job job_local232634845_0001completed successfully

22/06/21 10:17:02 INFO mapreduce.Job: Counters: 38File

System Counters

FILE: Number of bytes read=125588 FILE:

Number of bytes written=682332FILE: Number

of read operations=0

FILE: Number of large read operations=0FILE:

Number of write operations=0 HDFS: Number

of bytes read=1776380 HDFS: Number of bytes

written=74 HDFS: Number of read

operations=13

HDFS: Number of large read operations=0HDFS:

Number of write operations=4

Map-Reduce Framework

Map input records=6565 Map

output records=6564Map

output bytes=45948

Map output materialized bytes=59082Input split bytes=114

Combine input records=0

Combine output records=0

Reduce input groups=12 Reduce

shuffle bytes=59082Reduce

input records=6564 Reduce

output records=12 Spilled

Records=13128 Shuffled Maps

=1

Failed Shuffles=0 Merged

Map outputs=1 GC time

elapsed (ms)=54CPU time

spent (ms)=0

Physical memory (bytes)snapshot=0Virtual

memory (bytes) snapshot=0

Total committed heap usage
(bytes)=999292928 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=888190File

Output Format Counters

Bytes Written=74
hduser@bmsce-OptiPlex-3060:~$ hdfs dfs-ls/avg_temp_meanmax_outputls:

`/avg_temp_meanmax_output': No such file or directory

hduser@bmsce-OptiPlex-3060:~$ hdfs dfs-ls/avg_temp_output_meanmax_1901Found 2 items

-rw-r--r-- 1 hduser supergroup 0 2022-06-21 10:17
/avg_temp_output_meanmax_1901/_SUCCESS

-rw-r--r-- 1 hduser supergroup 74 2022-06-21 10:17
/avg_temp_output_meanmax_1901/part-r-00000

hduser@bmsce-OptiPlex-3060:~$ hdfs dfs-cat
/avg_temp_output_meanmax/part-r-00000

cat: `/avg_temp_output_meanmax/part-r-00000': No such file or
directory hduser@bmsce-OptiPlex-3060:~$ hdfs dfs-cat
/avg_temp_output_meanmax_1901/part-r-00000

01 4

02 0

03 7

04 44

05 100

06 168

07 219

08 198

09 141

10 100

11 19

12 3

LAB 7

For a given Text file, create a Map Reduce program to sort the content in analphabetic order listing only top 'n' maximum occurrence of words.

// TopN.java package sortWords;

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
org.apache.hadoop.util.GenericOptionsParser; import
utils.MiscUtils;import java.io.IOException; import java.util.*;
public class TopN {
 public static void main(String[] args) throws Exception {Configuration
 conf = new Configuration();
 String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
if (otherArgs.length != 2) {
 System.err.println("Usage: TopN <in> <out>");
 System.exit(2);
 }
 Job job = Job.getInstance(conf); job.setJobName("Top N");job.setJarByClass(TopN.class);
job.setMapperClass(TopNMapper.class);
//job.setCombinerClass(TopNReducer.class);
job.setReducerClass(TopNReducer.class); job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class); FileInputFormat.addInputPath(job, new
Path(otherArgs[0])); FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);
 }
 /**
 * The mapper reads one line at the time, splits it into an array of single words andemits every
* word to the reducers with the value of 1.
 */
 public static class TopNMapper extends Mapper<Object, Text, Text, IntWritable> {
private finalstatic IntWritable one = new IntWritable(1); private Text word = newText();
private String tokens = "[_|$#<>\\^=\\[\\]\\*/\\\\,;,.\\-:()?!\"']";@Override 17
 public void map(Object key, Text value, Context context)throwsIOException,
InterruptedException {
```

```
 String cleanLine = value.toString().toLowerCase().replaceAll(tokens, " ");StringTokenizer itr =
new StringTokenizer(cleanLine); while (itr.hasMoreTokens()) {word.set(itr.nextToken().trim());
context.write(word, one);
 }
 }
 }
 /**
 * The reducer retrieves every word and puts it into a Map: if the word alreadyexists in the
* map,
increments its value, otherwise sets it to 1.
 */
 public static class TopNReducer extends Reducer<Text, IntWritable, Text,IntWritable> { private
Map<Text, IntWritable> countMap = new HashMap<>();@Override public void reduce(Text
key, Iterable<IntWritable> values, Context context)throwsIOException, InterruptedException {
 // computes the number of occurrences of a single word int sum = 0; for(IntWritable val :
values) { sum += val.get();
 }
 // puts the number of occurrences of this word into the map.
 // We need to create another Text object because the Text instance
 // we receive isthe same for all the words countMap.put(new Text(key), newIntWritable(sum));
 }
@Override
 protected void cleanup(Context context) throwsIOException,InterruptedException {
 Map<Text, IntWritable> sortedMap = MiscUtils.sortByValues(countMap);int counter
 =
 0; for (Text key :sortedMap.keySet()) { if (counter++ == 3) {
break;
 }
 context.write(key,sortedMap.get(key));
 }
 }
```

```
 }
 /**
 * The combiner retrieves every word and puts it into a Map: if the word alreadyexists in the
* map, increments its value, otherwise sets it to 1.
 */
 public static class TopNCombiner extends Reducer<Text, IntWritable, Text,IntWritable>
{ 18
 @Override
 public void reduce(Text key, Iterable<IntWritable> values, Context
context)throwsIOException, InterruptedException {
 // computes the number of occurrences of a single word int sum = 0; for(IntWritable val :
values) { sum += val.get();
 }
 context.write(key, new IntWritable(sum));
}
 }
}
// MiscUtils.java package utils;
import java.util.*;
public class MiscUtils {
 /**
sortsthe map by values. Taken from:
http://javarevisited.blogspot.it/2012/12/how-to-sort-hashmap-java-by-key-and-va
 lue.html */
 public static <K extends Comparable, V extends Comparable> Map<K, V>sortByValues(Map<K,
V> map) {
 List<Map.Entry<K, V>> entries = new
 LinkedList<Map.Entry<K,V>>(map.entrySet()); Collections.sort(entries, new
 Comparator<Map.Entry<K, V>>() {
 @Override public int compare(Map.Entry<K, V> o1, Map.Entry<K, V> o2) {
```

```
return o2.getValue().compareTo(o1.getValue());
 }
 });
 //LinkedHashMap will keep the keys in the order they are inserted
 //which is currently sorted on natural ordering Map<K, V>
sortedMap = new LinkedHashMap<K, V>();for (Map.Entry<K, V>
entry : entries) { sortedMap.put(entry.getKey(),
entry.getValue());
 }
 return sortedMap;
 }
}
```

```
C:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \sortwordsOutput\part-r-00000
car     7
deer    6
bear    3
```

LAB 8:-Create a Hadoop Map Reduce program to combine information from the users file along with Information from the posts file by using the concept of joinand display user_id, Reputation and Score.

```
// JoinDriver.java import org.apache.hadoop.conf.Configured; import
org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.*; import
org.apache.hadoop.mapred.lib.MultipleInputs; import org.apache.hadoop.util.*; public class
JoinDriver extends Configured implements Tool {
public static class KeyPartitionerimplements Partitioner<TextPair, Text>
 {@Override public void configure(JobConf job) {}

 @Override
 public int getPartition(TextPair key, Text value, int numPartitions) {return
(key.getFirst().hashCode() & Integer.MAX_VALUE) % numPartitions;
 }
 }
@Override public int run(String[] args) throws Exception { if (args.length != 3)
```

```
{ System.out.println("Usage: <Department Emp Strength input>
<Department Name input> <output>");return -1;
}
JobConf conf = new JobConf(getConf(), getClass()); conf.setJobName("Join'Department
Emp Strength input' with 'Department Name input'");
Path AInputPath = new Path(args[0]);
Path BInputPath = new Path(args[1]);Path
outputPath = new Path(args[2]);
MultipleInputs.addInputPath(conf, AInputPath,
TextInputFormat.class,Posts.class); MultipleInputs.addInputPath(conf,
BInputPath, TextInputFormat.class,User.class);
FileOutputFormat.setOutputPath(conf, outputPath);
conf.setPartitionerClass(KeyPartitioner.class);
conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);
 conf.setMapOutputKeyClass(TextPair.class);21
 conf.setReducerClass(JoinReducer.class);
 conf.setOutputKeyClass(Text.class);
 JobClient.runJob(conf);
return 0;
}
public static void main(String[] args) throws Exception {int
exitCode = ToolRunner.run(new JoinDriver(), args);
System.exit(exitCode);
}
}
//JoinReducer.java import java.io.IOException; import java.util.Iterator;
import org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.*;
public classJoinReducer extends MapReduceBase implements Reducer<TextPair,Text, Text, Text>
{
@Override
public void reduce (TextPair key, Iterator<Text> values, OutputCollector<Text,Text>
```

```
output, Reporter reporter)
 throwsIOException
{
Text nodeId = new Text(values.next()); while (values.hasNext()) {Text node
= values.next();
Text outValue = new Text(nodeId.toString() + "\t\t" +
node.toString());output.collect(key.getFirst(), outValue);
}
}
}
// User.java import java.io.IOException; import java.util.Iterator; import
org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.FSDataInputStream; import
org.apache.hadoop.fs.FSDataOutputStream; import
org.apache.hadoop.fs.FileSystem; import
org.apache.hadoop.fs.Path; import org.apache.hadoop.io.LongWritable; import
org.apache.hadoop.io.Text; import org.apache.hadoop.mapred.*;
import org.apache.hadoop.io.IntWritable;
public class User extends MapReduceBase implements Mapper<LongWritable,Text, TextPair,
Text> {
22
@Override
public void map(LongWritable key, Text value, OutputCollector<TextPair, Text>output,
Reporter reporter)
throwsIOException
{
String valueString = value.toString();
String[] SingleNodeData = valueString.split("\t"); output.collect(new
TextPair(SingleNodeData[0], "1"), newText(SingleNodeData[1]));
}
}
//Posts.java importjava.io.IOException;
import org.apache.hadoop.io.*; import org.apache.hadoop.mapred.*;
```

```
public class Posts extends MapReduceBase implements Mapper<LongWritable,Text, TextPair,
Text> {

@Override

public void map(LongWritable key, Text value, OutputCollector<TextPair, Text>output,

Reporter reporter)

 throwsIOException

{

String valueString = value.toString();

String[] SingleNodeData = valueString.split("\t"); output.collect(new
TextPair(SingleNodeData[3], "0"), new

Text(SingleNodeData[9]));

}

}

// TextPair.java import java.io.*;
import org.apache.hadoop.io.*;

public class TextPairimplements WritableComparable<TextPair> {private Text

 first; private Text second;

 public TextPair() {set(new Text(), new Text());

 }

 public TextPair(String first, String second) {set(new Text(first), new

 Text(second)); }

 public TextPair(Textfirst, Textsecond) {set(first,second);

 }
```



```
 public void set(Text first, Textsecond) {this.first = first; this.second = second;

 }

 public Text getFirst() { return first;

 }

 public Text getSecond() { return second;
 }
```

```
@Override
public void write(DataOutput out) throwsIOException {
first.write(out);second.write(out); }
@Override public void readFields(DataInput in) throwsIOException {first.readFields(in);
second.readFields(in);
}
@Override public int hashCode() { return first.hashCode() * 163
+second.hashCode(); }
@Override public boolean equals(Object o) { if (o instanceof TextPair) { TextPair
tp = (TextPair) o;
return first.equals(tp.first) && second.equals(tp.second);
} return false;
}
@Override public String toString() { return first + "\t" + second;
}

@Override
public int compareTo(TextPairtp) { int cmp = first.compareTo(tp.first); if (cmp !=
0) { return
cmp;
}
return second.compareTo(tp.second);
}
// ^^ TextPair
// vv TextPairComparator public static class Comparator extendsWritableComparator
{ private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();
public Comparator() {super(TextPair.class);
}
@Override public int compare(byte[] b1, int s1, int l1, byte[] b2, int s2, int l2) {try
```

{ 24

int firstL1 = WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1, s1); int firstL2 =
WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2, s2); int cmp =
TEXT_COMPARATOR.compare(b1,s1, firstL1, b2,s2, firstL2); if (cmp != 0) { returncmp;
}

return TEXT_COMPARATOR.compare(b1,s1 + firstL1, l1 - firstL1,b2, s2
+ firstL2, l2 - firstL2);

} catch (IOException e) { throw new IllegalArgumentException(e);

}

}
}

static {

WritableComparator.define(TextPair.class, new Comparator());

}
public static class FirstComparator extends WritableComparator {

private static final Text.Comparator TEXT_COMPARATOR = new Text.Comparator();

public FirstComparator() {super(TextPair.class);

}

@Override public int compare(byte[] b1, int s1, int l1, byte[] b2, int s2, int l2) {try { int firstL1 =
WritableUtils.decodeVIntSize(b1[s1]) + readVInt(b1,s1); int firstL2 =
WritableUtils.decodeVIntSize(b2[s2]) + readVInt(b2,s2); returnTEXT_COMPARATOR.compare(b1,
s1, firstL1, b2,s2, firstL2);

} catch (IOException e) { throw new IllegalArgumentException(e);

}

}

@Override

public int compare(WritableComparable a, WritableComparable b) { if (ainstanceof TextPair
&& b
instanceof TextPair){return ((TextPair) a).first.compareTo(((TextPair) b).first);

}
return super.compare(a, b);

```
 }
 }
}
```

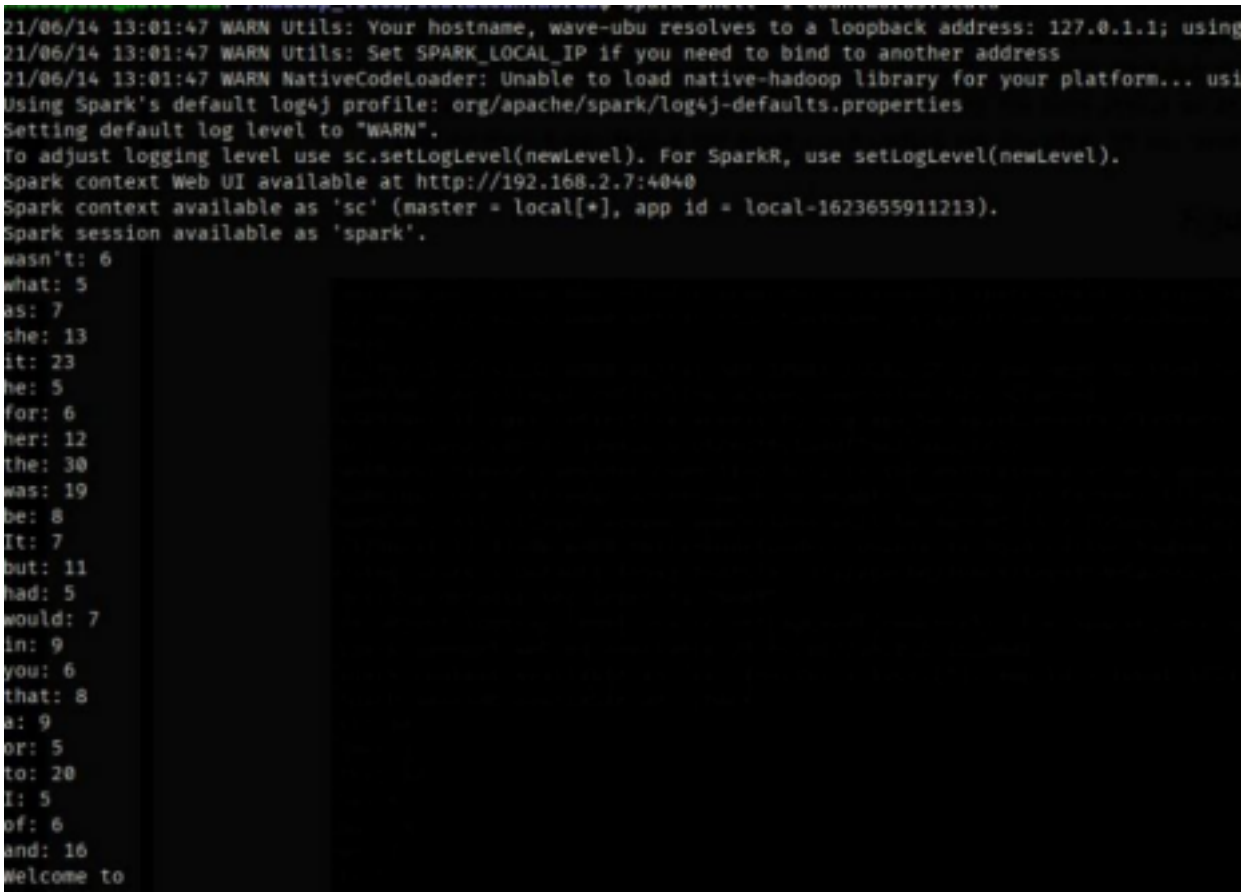
```
c:\hadoop_new\share\hadoop\mapreduce>hdfs dfs -cat \joinOutput\part-00000
"100005361"     "2"             "36134"
"100018705"     "2"             "76"
"100022094"     "0"             "6354"
```

LAB 9 Program to print word count on scala shell and print “Hello world” onscala

IDE scala> println("Hello World!");

Hello World!

```
val data=sc.textFile("sparkdata.txt")data.collect;
valsplitdata = data.flatMap(line => line.split(" "));splitdata.collect;
val mapdata = splitdata.map(word => (word,1));mapdata.collect;
val reducedata = mapdata.reduceByKey(_+_);
reducedata.collect
```



LAB-10 :-Using RDD and FlaMap count how many times each word appears in afile and write out a list of words whose count is strictly greater than 4 using Spark

- commands and output:

cala> val textFile=sc.textFile("/home/hduser/Desktop/sample.txt");

textFile: org.apache.spark.rdd.RDD[String] =
/home/hduser/Desktop/sample.txt MapPartitionsRDD[8] at textFile at
<console>:24
scala> val counts=textFile.flatMap(line=>line.split("
")).map(word=>(word,1)).reduceByKey(_=_)

<console>:25: error: reassignment to val

val
counts=textFile.flatMap(line=>line.split("")).map(word=>(word,1)).reduceByKey(_=_) scala>
val counts=textFile.flatMap(line=>line.split("
")).map(word=>(word,1)).reduceByKey(_+_)

counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[11]
atreduceByKeyat <console>:25
scala> import scala.collection.immutable.ListMapimport

scala.collection.immutable.ListMap

```
scala> val sorted=ListMap(counts.collect.sortWith(_._2>_._2):_*)

sorted: scala.collection.immutable.ListMap[String,Int] = Map(is -> 4, how -> 4, your -> 4,
are -> 1, brother -> 1, sister -> 1, family -> 1, ypu -> 1, job -> 1, hi -> 1,

hw -> 1)
scala> println(sorted)

Map(is -> 4, how -> 4, your -> 4, are -> 1, brother -> 1, sister -> 1, family -> 1,
ypu -> 1, job -> 1, hi -> 1, hw -> 1)
scala> for((k,v)<-sorted)

     | {

     | if(v>4)

     | {

     | print(k+",")

     | print(v)

     | println()

     | }| }
```