

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**

## **BIG DATA ANALYTICS**

*Submitted by*

**Shashank Sharma (1BM19CS147)**

*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 Shashank Sharma (1BM19CS147), who is 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 a BIG DATA ANALYTICS - (20CS6PEBDA) work prescribed for the said degree.

**Antara Roy Choudhary**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

## LAB-1

### 1 Perform the following DB operations using Cassandra.

1. Create a keyspace by name Employee
2. Create a column family by name  
Employee-Info with attributes  
Emp\_Id Primary Key, Emp\_Name,  
Designation, Date\_of\_Joining, Salary, Dept\_Name
3. Insert the values into the table in batch
4. Update Employee name and Department of Emp-Id 121
5. Sort the details of Employee records based on salary
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.
7. Update the altered table to add project names.
8. Create a TTL of 15 seconds to display the values of Employees.

- **COMMANDS AND OUTPUT:**

```
cqlsh> CREATE KEYSPACE Employee WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
USE employee;
cqlsh:employee> CREATE TABLE Employee_info(Emp_id int PRIMARY KEY,Emp_name text,Desg text,Doj
timestamp,Salary float,dept text);
cqlsh:employee> BEGIN BATCH
... INSERT INTO
... Employee_info(Emp_id,Emp_name,Desg,Doj,Salary,dept)
... VALUES(123,'Sakshi','Manager','2000-09-24',650000,'Export')
... INSERT INTO Employee_info(Emp_id,Emp_name,Desg,Doj,Salary,dept)
... VALUES(121,'Ritvika','AsstManager','2001-01-04',620000,'Export')
... INSERT INTO Employee_info(Emp_id,Emp_name,Desg,Doj,Salary,dept)
... VALUES(131,'Priya','HR','1999-05-14',780000,'HR')
... APPLY BATCH;
cqlsh:employee> SELECT * FROM Employee_info;
```

emp_id	dept	desg	doj	emp_name	salary
123	Export	Manager	2000-09-24 00:00:00.000000+0000	Sakshi	6.5e+05
121	Export	AsstManager	2001-01-04 00:00:00.000000+0000	Ritvika	6.2e+05
131	HR	HR	1999-05-14 00:00:00.000000+0000	Priya	7.8e+05

(3 rows)

```
cqlsh:employee> UPDATE Employee_info SET Emp_name='Ritvika_Singh' WHERE Emp_id=121;
cqlsh:employee> SELECT * FROM Employee_info;
```

emp_id	dept	desg	doj	emp_name	salary
123	Export	Manager	2000-09-24 00:00:00.000000+0000	Sakshi	6.5e+05
121	Export	AsstManager	2001-01-04 00:00:00.000000+0000	Ritvika_Singh	6.2e+05
131	HR	HR	1999-05-14 00:00:00.000000+0000	Priya	7.8e+05

(3 rows)

```
cqlsh:employee> UPDATE Employee_info SET dept='import' WHERE Emp_id=121;
cqlsh:employee> SELECT * FROM Employee_info;
```

emp_id	dept	desg	doj	emp_name	salary
123	Export	Manager	2000-09-24 00:00:00.000000+0000	Sakshi	6.5e+05
121	import	AsstManager	2001-01-04 00:00:00.000000+0000	Ritvika_Singh	6.2e+05
131	HR	HR	1999-05-14 00:00:00.000000+0000	Priya	7.8e+05

```
cqlsh:employee> ALTER TABLE Employee_info ADD projects set<text>;
cqlsh:employee> SELECT * FROM Employee_info;
```

emp_id	dept	desg	doj	emp_name	projects	salary
123	Export	Manager	2000-09-24 00:00:00.000000+0000	Sakshi	null	6.5e+05
121	import	AsstManager	2001-01-04 00:00:00.000000+0000	Ritvika_Singh	null	6.2e+05
131	HR	HR	1999-05-14 00:00:00.000000+0000	Priya	null	7.8e+05

(3 rows)

```
cqlsh:employee> UPDATE Employee_info SET projects={'proj1','proj2'} WHERE Emp_id=123;
cqlsh:employee> SELECT * FROM Employee_info;
```

emp_id	dept	desg	doj	emp_name	projects	salary
123	Export	Manager	2000-09-24 00:00:00.000000+0000	Sakshi	{'proj1','proj2'}	6.5e+05
121	import	AsstManager	2001-01-04 00:00:00.000000+0000	Ritvika_Singh	null	6.2e+05
131	HR	HR	1999-05-14 00:00:00.000000+0000	Priya	null	7.8e+05

(3 rows)

## LAB-2

### 2 Perform the following DB operations using Cassandra.

1. Create a keyspace by name Library
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
3. Insert the values into the table in batch
4. Display the details of the table created and increase the value of the counter
5. Write a query to show that a student with id 112 has taken a book "BDA" 2 times.
6. Export the created column to a csv file
7. Import a given csv dataset from local file system into Cassandra column family

- **COMMANDS AND OUTPUT:**

```
cqlsh> create keyspace library with replication={ 'class': 'SimpleStrategy', 'replication_factor': 1 };
cqlsh> use library;
cqlsh:library> create table library_info(stud_id varchar, counter_value counter, stud_name
text, book_name varchar,
book_id varchar, date_of_issue timestamp, primary
key(stud_id, stud_name, book_name, book_id, date_of_issue));
cqlsh:library> update library.library_info set counter_value=counter_value+1 where
stud_id='cs112' and stud_name='kundana' and book_name='spooky' and book_id='bk121' and
date_of_issue='2020-11-19'; cqlsh:library> update library.library_info set
counter_value=counter_value+1 where stud_id='cs121' and stud_name='bobby' and
book_name='spooky boy' and book_id='bk131' and date_of_issue='2020-11-16'; cqlsh:library>
update library.library_info set counter_value=counter_value+1 where stud_id='cs144' and
stud_name='felix' and book_name='spooky man' and book_id='bk141' and
date_of_issue='2019-03-03';
cqlsh:library> select * from library_info;
```

stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
cs121	bobby	spooky boy	bk131	2020-11-16 00:00:00.000000+0000	1
cs112	kundana	spooky	bk121	2020-11-19 00:00:00.000000+0000	1
cs144	felix	spooky man	bk141	2019-03-03 00:00:00.000000+0000	1

(3 rows)

```
cqlsh:library> update library.library_info set counter_value=counter_value+1 where
stud_id='cs112' and stud_name='kundana' and book_name='spooky' and book_id='bk121' and
date_of_issue='2020-11-19';
cqlsh:library> select * from library_info;
```

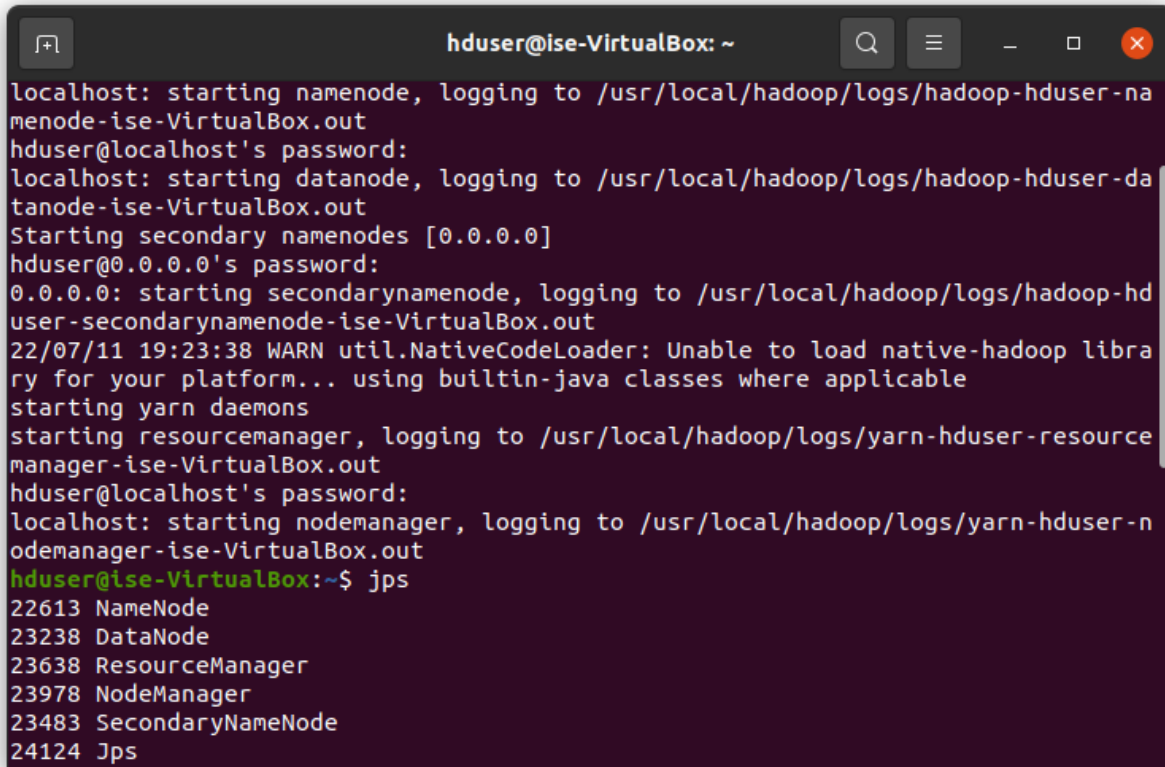
stud_id	stud_name	book_name	book_id	date_of_issue	counter_value
cs121	bobby	spooky boy	bk131	2020-11-16 00:00:00.000000+0000	1

cs112		kundana		spooky		bk121		2020-11-19 00:00:00.000000+0000		2
cs144		felix		spooky man		bk141		2019-03-03 00:00:00.000000+0000		1

(3 rows)

## LAB-4

### 4. Screenshot of Hadoop installed



```
hduser@ise-VirtualBox: ~  
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-na  
menode-ise-VirtualBox.out  
hduser@localhost's password:  
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-da  
tanode-ise-VirtualBox.out  
Starting 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-hd  
user-secondarynamenode-ise-VirtualBox.out  
22/07/11 19:23:38 WARN util.NativeCodeLoader: Unable to load native-hadoop libra  
ry for your platform... using builtin-java classes where applicable  
starting yarn daemons  
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resource  
manager-ise-VirtualBox.out  
hduser@localhost's password:  
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-n  
odemanager-ise-VirtualBox.out  
hduser@ise-VirtualBox:~$ jps  
22613 NameNode  
23238 DataNode  
23638 ResourceManager  
23978 NodeManager  
23483 SecondaryNameNode  
24124 Jps
```

## LAB-5

### 5 Execution of HDFS Commands for interaction with Hadoop Environment.

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

This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh

Starting namenodes on [localhost]

hduser@localhost's password:

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

hduser@localhost's password:

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

Starting 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.out

starting yarn daemons

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

hduser@localhost's password:

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

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

4644 NameNode

5450 SecondaryNameNode

6666 NodeManager

4827 DataNode

5710 ResourceManager

6799 Jps

```
hduser@bmsce-Precision-T1700:~$ ls
```

b           'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz'

c           Pictures

derby.log     pig\_1564816082257.log

Desktop      pt

Documents    PT72Installer

Downloads    Public

eclipse-workspace   R

examples.desktop   snap

hadoop-2.6.0.tar.gz   Templates



hive            toinstalledlist

metastore\_db    Videos

Music

hduser@bmsce-Precision-T1700:~\$ hadoop fs -ls /

Found 2 items

drwxrwxr-x - hduser supergroup    0 2019-08-01 16:19 /tmp

drwxr-xr-x - hduser supergroup    0 2019-08-01 16:03 /user

hduser@bmsce-Precision-T1700:~\$ hdfs dfs -mkdir /abc

hduser@bmsce-Precision-T1700:~\$ hdfs dfs -ls /

Found 3 items

drwxr-xr-x - hduser supergroup    0 2022-05-31 09:38 /abc

drwxrwxr-x - hduser supergroup    0 2019-08-01 16:19 /tmp

drwxr-xr-x - hduser supergroup    0 2019-08-01 16:03 /user

hduser@bmsce-Precision-T1700:~\$ hdfs dfs -touchz /abc/lab.txt

hduser@bmsce-Precision-T1700:~\$ hdfs dfs -ls /abc

Found 1 items

-rw-r--r-- 1 hduser supergroup    0 2022-05-31 09:39 /abc/lab.txt

hduser@bmsce-Precision-T1700:~\$ ls

b            'Packet Tracer 7.2.1 for Linux 64 bit.tar.gz'

c            Pictures

derby.log    pig\_1564816082257.log

Desktop    pt

Documents    PT72Installer

Downloads    Public

eclipse-workspace    R

examples.desktop    snap

hadoop-2.6.0.tar.gz    Templates

hive            toinstalledlist

metastore\_db    Videos

Music

hduser@bmsce-Precision-T1700:~\$ vi new.txt

hduser@bmsce-Precision-T1700:~\$ hdfs dfs -put new.txt /abc/newhadoop.txt

hduser@bmsce-Precision-T1700:~\$ hdfs dfs -cat /abc/newhadoop.txt

Cbbbbbb

fgggjyujhcvdgrbghh

```
hduser@bmsce-Precision-T1700:~$ cd /Desktop
bash: cd: /Desktop: No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x - hduser supergroup      0 2022-05-31 09:48 /abc
drwxrwxr-x - hduser supergroup      0 2019-08-01 16:19 /tmp
drwxr-xr-x - hduser supergroup      0 2019-08-01 16:03 /user
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyFromLocal /home/hduser/Desktop/Welcome.txt /abc/newWelcome.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -cat /abc/newWelcome.txt
nnkjkndngdmgic
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/wc.txt /home/hduser/Downloads/wcc.txt
get: `/abc/wc.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -get /abc/newWelcome.txt /home/hduser/Downloads/wcc.txt
hduser@bmsce-Precision-T1700:~$ hdfs dfs -copyToLocal /abc/newWelcome.txt /home/hduser/Downloads
hduser@bmsce-Precision-T1700:~$ hadoop fs -mv /abc /FFF
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x - hduser supergroup      0 2022-05-31 10:08 /FFF
drwxrwxr-x - hduser supergroup      0 2019-08-01 16:19 /tmp
drwxr-xr-x - hduser supergroup      0 2019-08-01 16:03 /user
hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /FFF/new.txt /tmp
cp: `/FFF/new.txt': No such file or directory
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /FFF
Found 3 items
-rw-r--r-- 1 hduser supergroup      0 2022-05-31 09:39 /FFF/lab.txt
-rw-r--r-- 1 hduser supergroup     14 2022-05-31 10:08 /FFF/newWelcome.txt
-rw-r--r-- 1 hduser supergroup     27 2022-05-31 09:48 /FFF/newhadoop.txt
hduser@bmsce-Precision-T1700:~$ hadoop fs -cp /FFF/lab.txt /tmp
hduser@bmsce-Precision-T1700:~$ hdfs dfs -ls /tmp
Found 2 items
drwx-wx-wx - hduser supergroup      0 2019-08-01 16:19 /tmp/hive
-rw-r--r-- 1 hduser supergroup      0 2022-05-31 10:19 /tmp/lab.txt
hduser@bmsce-Precision-T1700:~$
```

## LAB-6

6. From the following link extract the weather data <https://github.com/tomwhite/hadoop-book/tree/master/input/ncdc/all>. Create a Map Reduce program to

a) find average temperature for each year from 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 Enter the input and output
            parameters"");
            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;
import java.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<LongWritable, Text,
Text, IntWritable> {
```

```

public static final int MISSING = 9999;
public void map(LongWritable key, Text value,
Mapper<LongWritable, Text, Text, IntWritable>.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 && quality.matches("&quot;[01459]&quot;"))
context.write(new Text(year), new
IntWritable(temperature));
}
}

```

#### AverageReducer

```

package temp;
import java.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<Text, IntWritable,
Text, IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values,
Reducer<Text, IntWritable, Text, IntWritable>.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
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
hduser@localhost's password:
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-
bmsce-Precision-T1700.out
hduser@localhost's password:
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-
bmsce-Precision-T1700.out
Starting 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.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-
bmsce-Precision-T1700.out
hduser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-
nodemanager-bmsce-Precision-T1700.out
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)
    at java.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 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/21 10:26:05 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
```

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

22/06/21 10:26:05 INFO input.FileInputFormat: Total input paths to process : 1

22/06/21 10:26:05 INFO mapreduce.JobSubmitter: number of splits:1

22/06/21 10:26:05 INFO mapreduce.JobSubmitter: Submitting tokens for 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 config null

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 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100

22/06/21 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 output

22/06/21 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 0

22/06/21 10:26:06 INFO mapred.Task: Task:attempt\_local1195965365\_0001\_m\_000000\_0 is  
done. And is in 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 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:  
org.apache.hadoop.mapreduce.task.reduce.Shuffle@65367f35  
22/06/21 10:26:06 INFO reduce.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 for fetching 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: 72206 len: 72210 to MEMORY  
22/06/21 10:26:06 INFO reduce.InMemoryMapOutput: Read 72206 bytes from map-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: EventFetcher is interrupted.. 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 1 in-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 1 segments left of  
total size: 72199 bytes  
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merged 1 segments, 72206 bytes to disk  
to satisfy reduce memory limit  
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 1 files, 72210 bytes from disk  
22/06/21 10:26:06 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from  
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 1 segments left of  
total size: 72199 bytes  
22/06/21 10:26:06 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/21 10:26:06 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead,  
use mapreduce.job.skiprecords  
22/06/21 10:26:06 INFO mapred.Task: Task:attempt\_local1195965365\_0001\_r\_000000\_0 is  
done. And is in the process of 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\_0001\_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 running in 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\_0001 completed successfully

22/06/21 10:26:06 INFO mapreduce.Job: Counters: 38

#### File System Counters

FILE: Number of bytes read=152940

FILE: Number of bytes written=725372

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: 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=0

HDFS: Number of write operations=4

#### Map-Reduce Framework

Map input records=6565

Map output records=6564

Map output bytes=59076

Map output materialized bytes=72210

Input split bytes=110

Combine input records=0

Combine output records=0

Reduce input groups=1

Reduce shuffle bytes=72210

Reduce input records=6564

Reduce output records=1

Spilled Records=13128

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=63

CPU time spent (ms)=0

Physical memory (bytes) snapshot=0

Virtual 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=888190

File Output Format Counters

Bytes Written=8

hduser@bmsce-Precision-T1700:~\$ hadoop fs -cat /output\_1901/part-r-00000

1901 46

hduser@bmsce-Precision-T1700:~\$

**b) find the mean max temperature for every month**

- Program

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 Enter the input and output  
parameters&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;
```

```
import java.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<LongWritable, Text,
Text, IntWritable> {
    public static final int MISSING = 9999;
    public void map(LongWritable key, Text value,
Mapper<LongWritable, Text, Text, IntWritable>.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 && quality.matches("&quot;[01459]&quot;"))
            context.write(new Text(month), new
IntWritable(temperature));
    }
}

MeanMaxReducer.class
package meanmax;
import java.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<Text, IntWritable,
Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable> values,
Reducer<Text, IntWritable, Text, IntWritable>.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 > 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 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/21 10:17:01 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
22/06/21 10:17:01 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing
not performed. Implement the Tool interface and execute your application with ToolRunner
to remedy this.
22/06/21 10:17:01 INFO input.FileInputFormat: Total input paths to process : 1
22/06/21 10:17:01 INFO mapreduce.JobSubmitter: number of splits:1
22/06/21 10:17:01 INFO mapreduce.JobSubmitter: Submitting tokens for 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 config null
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 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/21 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 output
22/06/21 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 0  
22/06/21 10:17:01 INFO mapred.Task: Task:attempt\_local232634845\_0001\_m\_000000\_0 is done. And is in 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 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@1a055244  
22/06/21 10:17:01 INFO reduce.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 for fetching 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 bytes from map-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: EventFetcher is interrupted.. 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 1 in-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 1 segments left of total size: 59073 bytes  
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merged 1 segments, 59078 bytes to disk to satisfy reduce memory limit  
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 1 files, 59082 bytes from disk  
22/06/21 10:17:01 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from 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 1 segments left of total size: 59073 bytes  
22/06/21 10:17:01 INFO mapred.LocalJobRunner: 1 / 1 copied.

22/06/21 10:17:01 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use mapreduce.job.skiprecords

22/06/21 10:17:01 INFO mapred.Task: Task:attempt\_local232634845\_0001\_r\_000000\_0 is done. And is in the process of committing

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\_local232634845\_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 running in 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\_0001 completed successfully

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

File System Counters

- FILE: Number of bytes read=125588
- FILE: Number of bytes written=682332
- FILE: Number of read operations=0
- FILE: Number of large read operations=0
- FILE: 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=0
- HDFS: Number of write operations=4

Map-Reduce Framework

- Map input records=6565
- Map output records=6564
- Map output bytes=45948
- Map output materialized bytes=59082
- Input split bytes=114
- Combine input records=0
- Combine output records=0
- Reduce input groups=12
- Reduce shuffle bytes=59082
- Reduce input records=6564

Reduce output records=12  
Spilled Records=13128  
Shuffled Maps =1  
Failed Shuffles=0  
Merged Map outputs=1  
GC time elapsed (ms)=54  
CPU time spent (ms)=0  
Physical memory (bytes) snapshot=0  
Virtual 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=888190

File Output Format Counters

Bytes Written=74

hduser@bmsce-OptiPlex-3060:~\$ hdfs dfs -ls /avg\_temp\_meanmax\_output

ls: `/avg\_temp\_meanmax\_output': No such file or directory

hduser@bmsce-OptiPlex-3060:~\$ hdfs dfs -ls /avg\_temp\_output\_meanmax\_1901

Found 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 an alphabetic order

listing only top 10 maximum occurrences of words.

- Program

```
Driver-TopN.class
package samples.topn;
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.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
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.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);
    }
    public static class TopNMapper extends Mapper<Object, Text,
        Text, IntWritable> {
        private static final IntWritable one = new IntWritable(1);
```

```

private Text word = new Text();
private String tokens = "&quot;[_|${&lt;&gt;\\^=\\[\\]\\|\\*\\/\\\\\\,;\\.\\|-
:()?!\\&quot;&#39;}&quot;;
public void map(Object key, Text value, Mapper&lt;Object,
Text, Text, IntWritable&gt;.Context context) throws IOException,
InterruptedException {
String cleanLine =
value.toString().toLowerCase().replaceAll(this.tokens, "&quot; &quot;;);
StringTokenizer itr = new StringTokenizer(cleanLine);
while (itr.hasMoreTokens()) {
this.word.set(itr.nextToken().trim());
context.write(this.word, one);
}
}
}
}

```

TopNCombiner.class

```

package samples.topn;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class TopNCombiner 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 sum = 0;
for (IntWritable val : values)
sum += val.get();
context.write(key, new IntWritable(sum));
}
}

```

TopNMapper.class

```

package samples.topn;
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class TopNMapper extends Mapper&lt;Object, Text, Text,
IntWritable&gt; {
private static final IntWritable one = new IntWritable(1);

```



```

private Text word = new Text();
private String tokens = "&quot;[_|${&lt;&gt;\\^=\\[\\]\\*\\/\\\\\\,;\\.\\-
:()?!\\&quot;&#39;}&quot;;
public void map(Object key, Text value, Mapper<Object,
Text, Text, IntWritable>.Context context) throws IOException,
InterruptedException {
    String cleanLine =
    value.toString().toLowerCase().replaceAll(this.tokens, "&quot; &quot;;);
    StringTokenizer itr = new StringTokenizer(cleanLine);
    while (itr.hasMoreTokens()) {
        this.word.set(itr.nextToken().trim());
        context.write(this.word, one);
    }
}
}
}

```

TopNReducer.class

```

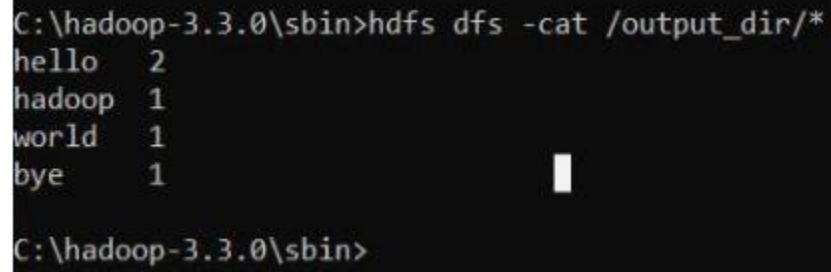
package samples.topn;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import utils.MiscUtils;

public class TopNReducer extends Reducer<Text, IntWritable,
Text, IntWritable> {
    private Map<Text, IntWritable> countMap = new HashMap<>();
    public void reduce(Text key, Iterable<IntWritable> values,
Reducer<Text, IntWritable, Text, IntWritable>.Context context)
throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values)
            sum += val.get();
        this.countMap.put(new Text(key), new IntWritable(sum));
    }
    protected void cleanup(Reducer<Text, IntWritable, Text,
IntWritable>.Context context) throws IOException,
InterruptedException {
        Map<Text, IntWritable> sortedMap =
MiscUtils.sortByValues(this.countMap);
        int counter = 0;
        for (Text key : sortedMap.keySet()) {
            if (counter++ == 20)

```

```
break;  
context.write(key, sortedMap.get(key));  
}  
}  
}
```

- **Output**

A terminal window with a black background and white text. The prompt is 'C:\hadoop-3.3.0\sbin>'. The command entered is 'hdfs dfs -cat /output\_dir/\*'. The output is a list of words and their counts: 'hello 2', 'hadoop 1', 'world 1', and 'bye 1'. The prompt 'C:\hadoop-3.3.0\sbin>' is shown again at the bottom.

```
C:\hadoop-3.3.0\sbin>hdfs dfs -cat /output_dir/*  
hello 2  
hadoop 1  
world 1  
bye 1  
  
C:\hadoop-3.3.0\sbin>
```

## LAB-8

### Create a Map Reduce program to demonstrating join operation

- Program

```
// 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 KeyPartitioner implements 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 &#39;Department Emp Strength input&#39; with
        &#39;Department Name
        input&#39;>");
        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.cl
ass);
conf.setMapOutputKeyClass(TextPair.class);
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 class JoinReducer extends MapReduceBase implements
Reducer<TextPair, Text, Text,
Text>; {
@Override
public void reduce (TextPair key, Iterator<Text> values,
OutputCollector<Text, Text>
output, Reporter reporter)
throws IOException
{
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>; {
@Override
public void map(LongWritable key, Text value,
OutputCollector<TextPair, Text> output,
Reporter reporter)
throws IOException
{
String valueString = value.toString();
String[] SingleNodeData = valueString.split(""\t"");
output.collect(new TextPair(SingleNodeData[0], ""1""), new
Text(SingleNodeData[1]));
}
}
//Posts.java
import java.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)
throws IOException
{
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 TextPair implements 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(Text first, Text second) {
    set(first, second);
}
public void set(Text first, Text second) {
    this.first = first;
    this.second = second;
}
public Text getFirst() {
    return first;
}
public Text getSecond() {
    return second;
}
@Override
public void write(DataOutput out) throws IOException {
    first.write(out);
    second.write(out);
}
@Override
public void readFields(DataInput in) throws IOException {
    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 + "&quot;\t&quot; + second;
}
@Override
public int compareTo(TextPair tp) {
int cmp = first.compareTo(tp.first);
if (cmp != 0) {
return cmp;
}
return second.compareTo(tp.second);
}
// ^^ TextPair
// vv TextPairComparator
public static class Comparator extends WritableComparator {
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 {
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) {
return cmp;
}
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);
            return TEXT_COMPARATOR.compare(b1, s1, firstL1, b2, s2, firstL2);
        } catch (IOException e) {
            throw new IllegalArgumentException(e);
        }
    }
    @Override
    public int compare(WritableComparable a, WritableComparable b) {
        if (a instanceof TextPair && b instanceof TextPair) {
            return ((TextPair) a).first.compareTo(((TextPair) b).first);
        }
        return super.compare(a, b);
    }
}

```

- output

```

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /join8_output/
Found 2 items
-rw-r--r--  1 Anusree supergroup          0 2021-06-13 12:16 /join8_output/_SUCCESS
-rw-r--r--  1 Anusree supergroup       71 2021-06-13 12:16 /join8_output/part-00000

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /join8_output/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" on scala IDE

- commands and outline:

```
hduser@bmsce-OptiPlex-3060:~$ spark-shell
```

```
22/06/28 09:34:37 WARN Utils: Your hostname, bmsce-OptiPlex-3060 resolves to a loopback address: 127.0.1.1; using 10.124.7.72 instead (on interface enp1s0)
```

```
22/06/28 09:34:37 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
```

```
22/06/28 09:34:37 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

```
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
```

```
Setting default log level to "WARN".
```

```
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
```

```
Spark context Web UI available at http://10.124.7.72:4040
```

```
Spark context available as 'sc' (master = local[*], app id = local-1656389082904).
```

```
Spark session available as 'spark'.
```

```
Welcome to
```

```
 _ _ _ _ _  
/ _ \ _ _ _ \ _/  
 _\ V _ V _ \' _ \'  
/ _ \ . _ \ _ \ \ / _ \ version 2.4.8  
/ _ \
```

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0\_312)

Type in expressions to have them evaluated.

Type :help for more information.

```
scala> println("hello");
```

```
hello
```

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

```
data: org.apache.spark.rdd.RDD[String] = /home/hduser/Desktop/sample.txt
```

```
MapPartitionsRDD[1] at textFile at <console>:24
```

```
scala> data.collect;
```

```
res1: Array[String] = Array(hi hw are ypu, how is your job, how is your family, how is your  
brother, how is your sister)
```

```
scala> val splitdata=data.flatMap(line=>line.split(" "));
```

```
splitdata: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at flatMap at
```

```
<console>:25
```

```
scala> splitdata.collect;
```

```
res2: Array[String] = Array(hi, hw, are, ypu, how, is, your, job, how, is, your, family, how, is,  
your, brother, how, is, your, sister)
```

```
scala> val mapdata=splitdata.map(word=>(word,1));  
mapdata: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at  
<console>:25
```

```
scala> mapdata.collect;  
res3: Array[(String, Int)] = Array((hi,1), (hw,1), (are,1), (ypu,1), (how,1), (is,1), (your,1), (job,1),  
(how,1), (is,1), (your,1), (family,1), (how,1), (is,1), (your,1), (brother,1), (how,1), (is,1), (your,1),  
(sister,1))
```

```
scala> val reducedata=mapdata.reduceByKey(_+_);  
reducedata: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at  
<console>:25
```

```
scala> reducedata.collect;  
res4: Array[(String, Int)] = Array((are,1), (brother,1), (is,4), (sister,1), (family,1), (how,4),  
(ypu,1), (job,1), (hi,1), (hw,1), (your,4))
```

```
scala>
```

## LAB-10

Using RDD and FlatMap count how many times each word appears in a file 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] at reduceByKey at
<console>:25
```

```
scala> import scala.collection.immutable.ListMap
import 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()
|   }
| }
```

//SINCE SAMPLE TEXT FILE DOESNT HAVE WORD WITH FREQUENCY >4,NO OUTPUT