

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**

## **BIG DATA ANALYTICS**

*Submitted by*

**YASHASVINI M R (1BM21CS252)**

*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**

**Feb-2024 to July-2024**

**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 “LAB COURSE **BIG DATA ANALYTICS**” carried out by **YASHASVINI M R (IBM21CS252)**, 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 2024. The Lab report has been approved as it satisfies the academic requirements in respect of a **Big Data Analytics - (22CS6PEBDA)** work prescribed for the said degree.

**Saritha A. N**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

### **Index Sheet**

<b>Sl. No.</b>	<b>Experiment Title</b>	<b>Page No.</b>
<b>1</b>	<b>MongoDB- CRUD Demonstration ( Practice and Self Study)</b>	<b>5</b>
<b>2</b>	<b>Perform the following DB operations using Cassandra-Student Database</b>	<b>8</b>
<b>3</b>	<b>Cassandra-Library Database</b>	<b>12</b>
<b>4</b>	<b>Hadoop Installation</b>	<b>14</b>
<b>5</b>	<b>Implement WordCount Program on Hadoop framework</b>	<b>15</b>
<b>6</b>	<b>HDFS Commands</b>	<b>19</b>
<b>7</b>	<b>Create a Map Reduce program to a) find average temperature for each year from NCDC data set. b) find the mean max temperature for every month</b>	<b>23</b>
<b>8</b>	<b>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</b>	<b>31</b>

## Course Outcome

C0	Apply the concepts of NoSQL, Hadoop, Spark for a given task
C1	Analyse data analytic techniques for a given problem .
C2	Analyse data analytic techniques for a given problem .

## 1. MongoDB- CRUD Demonstration( Practice and Self Study)

Inserting into database

```
test> use Student
switched to db Student
Student> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
```

Displaying inserted values

```
}
Student> db.Student.find()
[
  {
    _id: ObjectId('660a86053f257f0a2b66fd9b'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86063f257f0a2b66fd9c'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86063f257f0a2b66fd9d'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86063f257f0a2b66fd9e'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('660a86083f257f0a2b66fd9f'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'abhinav@gmail.com'
  }
]
```

Updating values

```
Student> db.Student.update({RollNo:10},{ $set:{email:"abhinav@gmail.com"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Student> db.Student.update({RollNo:11, Name:"ABC"},{ $set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
Student> db.Student.find()
```

Creating Customers database and inserting.

```
Student> db.createCollection("Customers");
{ ok: 1 }
Student> db.Customers.insert({cust_id:1,Balance:200, Type:"S"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda0') }
}
Student>

Student> db.Customers.insert({cust_id:1,Balance:1000, Type:"Z"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda1') }
}
Student>

Student> db.Customers.insert({cust_id:2,Balance:100, Type:"Z"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda2') }
}
Student>

Student> db.Customers.insert({cust_id:2,Balance:1000, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda3') }
}
Student>

Student> db.Customers.insert({cust_id:2,Balance:500, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a87f33f257f0a2b66fda4') }
}
Student>
```

Updating.

```
Student> db.Customers.aggregate (
...
... {$group : { _id : "$cust_id",
...
... minAccBal :{$min:"$Balance"},
... maxAccBal :{$max:"$Balance"} }));
[
  { _id: 3, minAccBal: 500, maxAccBal: 500 },
  { _id: 2, minAccBal: 50, maxAccBal: 1000 },
  { _id: 1, minAccBal: 200, maxAccBal: 1000 }
]
Student> db.Customers.aggregate(
... {$match:{Type:"Z"}},
... {$group:{_id:"$cust_id",
... TotAccBal:{$sum:"$Balance"}}},
... {$match:{TotAccBal:{$gt:1200}}});
```

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

bmscece@bmscece-HP-Elite-Tower-800-G9-Desktop-PC:~\$ cqlsh

Connected to Test Cluster at 127.0.0.1:9042

[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]

Use HELP for help.

cqlsh> CREATE KEYSPACE Students WITH REPLICATION={

... 'class': 'SimpleStrategy', 'replication\_factor': 1 };

cqlsh> DESCRIBE KEYSPACES

students system\_auth system\_schema system\_views

system system\_distributed system\_traces system\_virtual\_schema

cqlsh> SELECT \* FROM system.schema\_keyspaces;

InvalidRequest: Error from server: code=2200 [Invalid query] message="table schema\_keyspaces does not exist"

cqlsh> use Students;

cqlsh:students> create table Students\_info(Roll\_No int Primary key, StudName text, DateOfJoining timestamp, last\_exam\_Percent double);

cqlsh:students> describe tables;

students\_info

cqlsh:students> describe table students;

Table 'students' not found in keyspace 'students'

cqlsh:students> describe table students\_info;

CREATE TABLE students.students\_info (

roll\_no int PRIMARY KEY,

dateofjoining timestamp,

last\_exam\_percent double,

studname text

) WITH additional\_write\_policy = '99p'

AND bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND cdc = false

AND comment = "

AND compaction = {'class':

'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '16', 'class':

'org.apache.cassandra.io.compress.LZ4Compressor'}



```

AND memtable = 'default'
AND crc_check_chance = 1.0
AND default_time_to_live = 0
AND extensions = {}
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair = 'BLOCKING'
AND speculative_retry = '99p';

```

```

cqlsh:students> Begin batch insert into Students_info(Roll_no,
StudName,DateOfJoining, last_exam_Percent) values(1,'Sadhana','2023-10-09', 98)
insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent)
values(2,'Rutu','2023-10-10', 97)
insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent)
values(3,'Rachana','2023-10-10', 97.5)
insert into Students_info(Roll_no, StudName,DateOfJoining, last_exam_Percent)
values(4,'Charu','2023-10-06', 96.5) apply batch;
cqlsh:students> select * from students_info;

```

roll_no	dateofjoining	last_exam_percent	studname
1	2023-10-08 18:30:00.000000+0000	98	Sadhana
2	2023-10-09 18:30:00.000000+0000	97	Rutu
4	2023-10-05 18:30:00.000000+0000	96.5	Charu
3	2023-10-09 18:30:00.000000+0000	97.5	Rachana

(4 rows)

```

cqlsh:students> select * from students_info where roll_no in (1,2,3);

```

roll_no	dateofjoining	last_exam_percent	studname
1	2023-10-08 18:30:00.000000+0000	98	Sadhana
2	2023-10-09 18:30:00.000000+0000	97	Rutu
3	2023-10-09 18:30:00.000000+0000	97.5	Rachana

```

cqlsh:students> select * from students_info where Studname='Charu';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute
this query as it might involve data filtering and thus may have unpredictable
performance. If you want to execute this query despite the performance unpredictability,
use ALLOW FILTERING"

```

```

cqlsh:students> create index on Students_info(StudName);

```

```

cqlsh:students> select * from students_info where Studname='Charu';

```

roll_no	dateofjoining	last_exam_percent	studname
4	2023-10-05 18:30:00.000000+0000	96.5	Charu

(1 rows)

cqlsh:students> select Roll\_no,StudName from students\_info LIMIT 2;

roll_no	studname
1	Sadhana
2	Rutu

(2 rows)

cqlsh:students> SELECT Roll\_no as "USN" from Students\_info;

USN
1
2
4
3

(4 rows)

cqlsh:students> update students\_info set StudName='Shreya' where Roll\_no=3;  
cqlsh:students> select \* from students\_info;

roll_no	dateofjoining	last_exam_percent	studname
1	2023-10-08 18:30:00.000000+0000	98	Sadhana
2	2023-10-09 18:30:00.000000+0000	97	Rutu
4	2023-10-05 18:30:00.000000+0000	96.5	Charu
3	2023-10-09 18:30:00.000000+0000	97.5	Shreya

(4 rows)

cqlsh:students> update students\_info set roll\_no=8 where Roll\_no=3;  
InvalidRequest: Error from server: code=2200 [Invalid query] message="PRIMARY KEY part roll\_no found in SET part"  
cqlsh:students> delete last\_exam\_percent from students\_info where roll\_no=2;  
cqlsh:students> select \* from students\_info;

roll_no	dateofjoining	last_exam_percent	studname
1	2023-10-08 18:30:00.000000+0000	98	Sadhana
2	2023-10-09 18:30:00.000000+0000	null	Rutu

4	2023-10-05 18:30:00.000000+0000	96.5	Charu
3	2023-10-09 18:30:00.000000+0000	97.5	Shreya

(4 rows)

cqlsh:students> delete from students\_info where roll\_no=2;

cqlsh:students> select \* from students\_info;

roll_no	dateofjoining	last_exam_percent	studname
1	2023-10-08 18:30:00.000000+0000	98	Sadhana
4	2023-10-05 18:30:00.000000+0000	96.5	Charu
3	2023-10-09 18:30:00.000000+0000	97.5	Shreya

(3 rows)

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

#### 1. Create a keyspace by name Library

```
USE HELP FOR help;
cqlsh> CREATE KEYSPACE IF NOT EXISTS 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.

```
INFO: Request error from server: Code=2200 [unavailable query] message= table libraryinfo does not exist
cqlsh:library> CREATE TABLE libraryInfo (BookValue COUNTER, Stud_Id INT, Stud_Name TEXT, Book_Name TEXT, Book_Id TEXT, Date_of_issue TIMESTAMP, PRIMARY KEY(Stud_Id, Stud_Name, Book_Name, Book_Id, Date_of_issue));
cqlsh:library> DESCRIBE BATCH
```

#### 3. Insert the values into the table in batch

```
INFO: Request error from server: Code=2200 [unavailable query] message= table libraryinfo does not exist
cqlsh:library> UPDATE libraryInfo SET bookvalue = bookvalue + 1 WHERE Stud_Id = 101 AND Stud_Name = 'Alice' AND Book_Name = 'History of India' AND Book_Id = '201' AND Date_of_issue = '2024-05-09';
cqlsh:library> UPDATE libraryInfo SET bookvalue = bookvalue + 1 WHERE Stud_Id = 102 AND Stud_Name = 'John' AND Book_Name = 'Python' AND Book_Id = '203' AND Date_of_issue = '2024-02-09';
cqlsh:library> UPDATE libraryInfo SET bookvalue = bookvalue + 1 WHERE Stud_Id = 103 AND Stud_Name = 'Priya' AND Book_Name = 'C Fundamentals' AND Book_Id = '206' AND Date_of_issue = '2024-02-18';
cqlsh:library> UPDATE libraryInfo SET bookvalue = bookvalue + 1 WHERE Stud_Id = 104 AND Stud_Name = 'Shreya' AND Book_Name = 'Mechanical Engineering' AND Book_Id = '205' AND Date_of_issue = '2024-01-18';
```

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

```
cqlsh:library> select * from libraryInfo;

stud_id | stud_name | book_name | book_id | date_of_issue | bookvalue
-----|-----|-----|-----|-----|-----
104 | Shreya | Mechanical Engineering | 205 | 2024-01-17 18:30:00.000000+0000 | 1
102 | John | Python | 203 | 2024-02-08 18:30:00.000000+0000 | 1
101 | Alice | History of India | 201 | 2024-05-08 18:30:00.000000+0000 | 1
103 | Priya | C Fundamentals | 206 | 2024-02-17 18:30:00.000000+0000 | 1
(4 rows)
cqlsh:library> UPDATE libraryInfo SET bookvalue = bookvalue + 1 WHERE Stud_Id = 112 AND Stud_Name = 'Ashok' AND Book_Name = 'BDA' AND Book_Id = '210' AND Date_of_issue = '2023-08-18';
INFO: Request error from server: Code=2200 [unavailable query] message= table libraryinfo does not exist
```

#### 5. Write a query to show that a student with id 112 has taken a book “BDA” 2 time

```
(5 rows)
cqlsh:library> select * from libraryInfo where Stud_Id=112;

stud_id | stud_name | book_name | book_id | date_of_issue | bookvalue
-----|-----|-----|-----|-----|-----
112 | Ashok | BDA | 210 | 2023-08-17 18:30:00.000000+0000 | 2
(1 rows)
```

#### 6. Export the created column to a csv file

```
(5 rows)
cqlsh:library> copy libraryInfo (bookvalue,stud_id,stud_name,book_name,book_id,date_of_issue) TO 'Documents:\library.csv';
Using 16 child processes

Starting copy of library.libraryInfo with columns [bookvalue, stud_id, stud_name, book_name, book_id, date_of_issue].
Processed: 5 rows; Rate: 76 rows/s; Avg. rate: 76 rows/s
5 rows exported to 1 files in 0.100 seconds.
cqlsh:library>
```

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

```
cqlsh:library> copy libraryinfo (bookvalue,stud_id,stud_name,book_name,book_id,date_of_issue) FROM 'Documents:\library.csv';  
Using 16 child processes  
  
Starting copy of library.libraryinfo with columns [bookvalue, stud_id, stud_name, book_name, book_id, date_of_issue].
```

### 3. Hadoop Installation

```
jj@chakraubuntu22:/usr/local/hadoop$ ./usr/local/hadoop/hadoop-3.4.0/bin/hadoop version
Hadoop 3.4.0
Source code repository git@github.com:apache/hadoop.git -r bd8b77f398f626bb7791783192ee7a5dfaee760
Compiled by root on 2024-03-04T06:35Z
Compiled on platform linux-x86_64
Compiled with protoc 3.21.12
From source with checksum f7fe694a3613358b38812ae9c31114e
This command was run using /usr/local/hadoop/hadoop-3.4.0/share/hadoop/common/hadoop-common-3.4.0.jar
```

## 4. Hadoop Hdfs commands

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
```

```
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
```

```
WARNING: This is not a recommended production deployment configuration.
```

```
WARNING: Use CTRL-C to abort.
```

```
Starting namenodes on [localhost]
```

```
Starting datanodes
```

```
Starting secondary namenodes [bmscecse-HP-Elite-Tower-800-G9-Desktop-PC]
```

```
Starting resourcemanager
```

```
Starting nodemanagers
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop dfs -mkdir /sadh
```

```
WARNING: Use of this script to execute dfs is deprecated.
```

```
WARNING: Attempting to execute replacement "hdfs dfs" instead.
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -mkdir /sadh
```

```
mkdir: `/sadh': File exists
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
```

```
Found 1 items
```

```
drwxr-xr-x - hadoop supergroup    0 2024-05-13 14:27 /sadh
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /sadh
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put  
/home/hadoop/Desktop/example/Welcome.txt /sadh/WC.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /sadh/WC.txt
```

```
hiiii
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -get /sadh/WC.txt  
/home/hadoop/Desktop/example/WWC.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -get /sadh/WC.txt  
/home/hadoop/Desktop/example/WWC2.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put  
/home/hadoop/Desktop/example/Welcome.txt /sadh/WC2.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -getmerge  
/sadh/WC.txt /sadh/WC2.txt /home/hadoop/Desktop/example/Merge.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -getfacl /sadh/
```

```
# file: /sadh
```

```
# owner: hadoop
```

```
# group: supergroup
```

```
user::rwx
```

```
group::r-x
```

```
other::r-x
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mv /sadh  
/WC2.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /sadh  
/WC2.txt
```

```
ls: `/sadh': No such file or directory
```

```
Found 2 items
```

```
-rw-r--r--  1 hadoop supergroup    6 2024-05-13 14:51 /WC2.txt/WC.txt
```

```
-rw-r--r--  1 hadoop supergroup    6 2024-05-13 15:03 /WC2.txt/WC2.txt
```

```
hadoop@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cp /WC2.txt/  
/WC.txt
```



```

hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bmscscse-HP-Elite-Tower-800-G9-Desktop-PC]
Starting resourcemanager
Starting nodemanagers

```

```

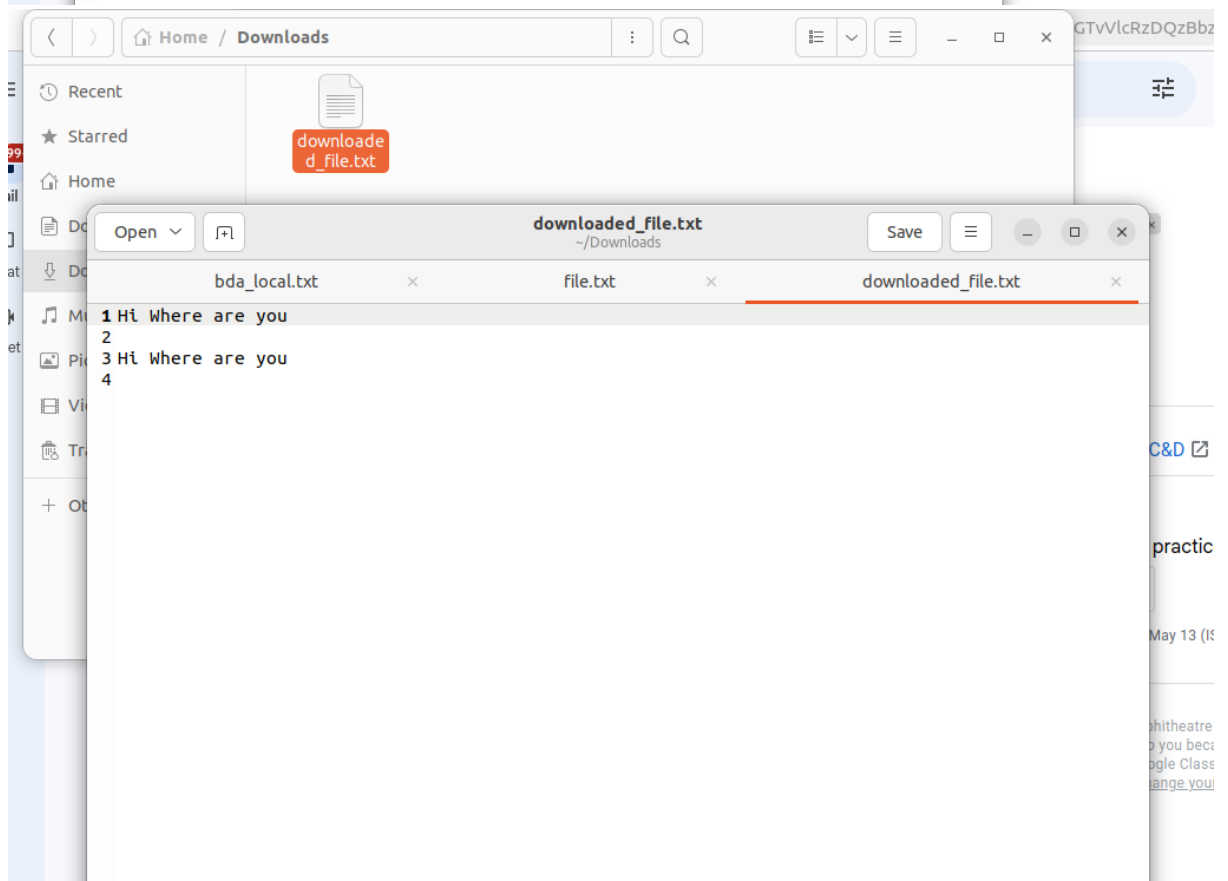
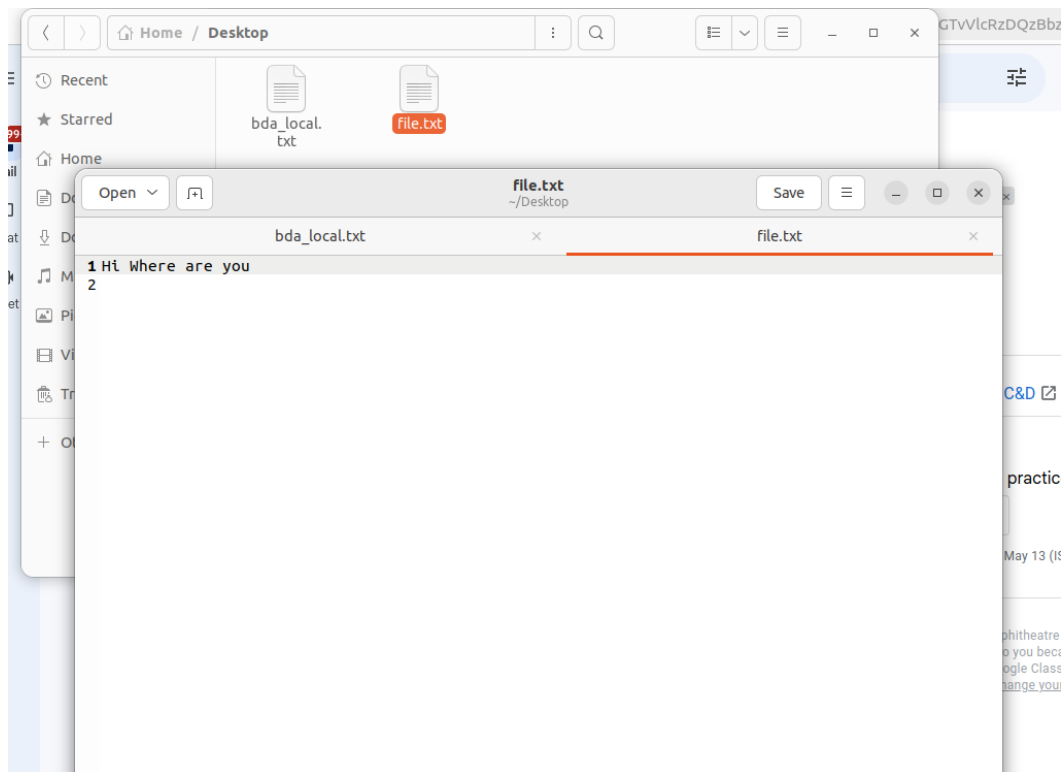
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC: ~
[-touchz <path> ...]
[-truncate [-w] <length> <path> ...]
[-usage [CMD ...]]

Generic options supported are:
-conf <configuration file>      specify an application configuration file
-D <property=value>             define a value for a given property
-fs <file:///hdfs://namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jb <localResourceManager.port> specify a ResourceManager
-files <file1,...>              specify a comma-separated list of files to be copied to the map reduce cluster
-lbjars <jar1,...>              specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...>       specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /
Found 1 items
drwxr-xr-x  - hadoop supergroup          0 2024-05-13 14:34 /bda_hadoop
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/bda_local.txt /bda_hadoop/bda_local.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /bda_hadoop/file.txt
cat: '/bda_hadoop/file.txt': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -put /home/hadoop/Desktop/bda_local.txt /bda_hadoop/file.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /bda_hadoop/file.txt
Hl Where are you
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -copyFromLocal /home/hadoop/Desktop/bda_local.txt /bda_hadoop/file_cp_local.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /bda_hadoop/file_cp_local.txt
Hl Where are you
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -cat /bda_hadoop/file_cp_local.txt
Hl Where are you
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -get /bda_hadoop/file.txt /home/hadoop/Downloads/downloaded_file.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -getmerge /bda_hadoop/file.txt /bda_hadoop/file_cp_local.txt /home/hduser/Downloads/downloaded_file.txt
getmerge: Mkdirs failed to create file:/home/hduser/Downloads (exists=false, cwd=file:/home/hadoop)
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -getmerge /bda_hadoop/file.txt /bda_hadoop/file_cp_local.txt /home/hadoop/Downloads/downloaded_file.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -getfacl /bda_hadoop/
# file: /bda_hadoop
# owner: hadoop
# group: supergroup
user::rwx
group::r-x
other::r-x
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hdfs dfs -copyToLocal /bda_hadoop/file.txt /home/hadoop/Desktop
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -mv /bda_hadoop /abc
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -ls /abc
Found 3 items
-rw-r--r--  1 hadoop supergroup          18 2024-05-13 14:48 /abc/bda_local.txt
-rw-r--r--  1 hadoop supergroup          18 2024-05-13 14:55 /abc/file.txt
-rw-r--r--  1 hadoop supergroup          18 2024-05-13 14:58 /abc/file_cp_local.txt
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$ hadoop fs -cp /hello/ /hadoop_lab
cp: '/hello/': No such file or directory
hadoop@bmscscse-HP-Elite-Tower-800-G9-Desktop-PC:~$

```



## 5. Implement WordCount Program on Hadoop framework

Mapper Code:

```
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.mapred.MapReduceBase;

import org.apache.hadoop.mapred.Mapper;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper<LongWritable,
Text, Text,
IntWritable> {

public void map(LongWritable key, Text value, OutputCollector<Text,
IntWritable> output, Reporter rep) throws IOException
{
String line = value.toString();

for (String word : line.split(" "))
{
if (word.length() > 0)
{
output.collect(new Text(word), new IntWritable(1));
} } } }
```

Reducer Code:

```
// Importing libraries

import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;

import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer<Text,
IntWritable, Text, IntWritable> {

// Reduce function

public void reduce(Text key, Iterator<IntWritable> value,
OutputCollector<Text, IntWritable> output,
Reporter rep) throws IOException
{
int count = 0;

// Counting the frequency of each words

while (value.hasNext())
{
IntWritable i = value.next();

count += i.get();
}
```

```
}  
output.collect(key, new IntWritable(count));  
} }
```

Driver Code: You have to copy paste this program into the WCDriver Java Class file.

```
// Importing libraries  
import java.io.IOException;  
import org.apache.hadoop.conf.Configured;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.IntWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapred.FileInputFormat;  
import org.apache.hadoop.mapred.FileOutputFormat;  
import org.apache.hadoop.mapred.JobClient;  
import org.apache.hadoop.mapred.JobConf;  
import org.apache.hadoop.util.Tool;  
import org.apache.hadoop.util.ToolRunner;  
public class WCDriver extends Configured implements Tool {  
    public int run(String args[]) throws IOException  
    {  
        if (args.length < 2)  
        {  
            System.out.println("Please give valid inputs");  
            return -1;  
        }  
    }  
}
```

```
JobConf conf = new JobConf(WCDriver.class);

FileInputFormat.setInputPaths(conf, new Path(args[0]));

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

conf.setMapperClass(WCMapper.class);

conf.setReducerClass(WCReducer.class);

conf.setMapOutputKeyClass(Text.class);

conf.setMapOutputValueClass(IntWritable.class);

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

JobClient.runJob(conf);

return 0;

}

// Main Method

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

int exitCode = ToolRunner.run(new WCDriver(), args);

System.out.println(exitCode);

}

}
```

**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.**

**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) == '+') {

            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("[01459]"))
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));
    }
}

```

```
C:\hadoop-3.3.0\sbin>hadoop jar C:\avgtemp.jar temp.AverageDriver /input_dir/temp.txt /avgtemp_outputdir
2021-05-15 14:52:50,635 INFO client.DefaultHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-15 14:52:51,005 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-15 14:52:51,111 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621060230696_0005
2021-05-15 14:52:51,735 INFO input.FileInputFormat: Total input files to process : 1
2021-05-15 14:52:52,751 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621060230696_0005
2021-05-15 14:52:53,073 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-15 14:52:53,237 INFO conf.Configuration: resource-types.xml not found
2021-05-15 14:52:53,238 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-15 14:52:53,312 INFO impl.YarnClientImpl: Submitted application application_1621060230696_0005
2021-05-15 14:52:53,352 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1621060230696_0005/
2021-05-15 14:52:53,353 INFO mapreduce.Job: Running job: job_1621060230696_0005
2021-05-15 14:53:06,640 INFO mapreduce.Job: Job job_1621060230696_0005 running in user mode : false
2021-05-15 14:53:06,643 INFO mapreduce.Job: map 0% reduce 0%
2021-05-15 14:53:12,758 INFO mapreduce.Job: map 100% reduce 0%
2021-05-15 14:53:19,060 INFO mapreduce.Job: map 100% reduce 100%
2021-05-15 14:53:25,967 INFO mapreduce.Job: Job job_1621060230696_0005 completed successfully
2021-05-15 14:53:26,096 INFO mapreduce.Job: Counters: 54
File System Counters
  FILE: Number of bytes read=72210
  FILE: Number of bytes written=674341
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=894860
  HDFS: Number of bytes written=8
  HDFS: Number of read operations=8
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
  HDFS: Number of bytes read erasure-coded=0
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=3782
```

```
C:\hadoop-3.3.0\sbin>hdfs dfs -ls /avgtemp_outputdir
Found 2 items
-rw-r--r--  1 Anusree supergroup          0 2021-05-15 14:53 /avgtemp_outputdir/_SUCCESS
-rw-r--r--  1 Anusree supergroup          8 2021-05-15 14:53 /avgtemp_outputdir/part-r-00000

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /avgtemp_outputdir/part-r-00000
1901    46

C:\hadoop-3.3.0\sbin>
```

## b) Find the mean max temperature for every month

### 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("Please Enter the input and output parameters");

            System.exit(-1);

        }

        Job job = new Job();

        job.setJarByClass(MeanMaxDriver.class);

        job.setJobName("Max temperature");

        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) == '+') {

            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("[01459]"))

            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));

    }

}

```

```

C:\hadoop-3.3.0\sbin>hadoop jar C:\meanmax.jar meanmax.MeanMaxDriver /input_dir/temp.txt /meanmax_output
2021-05-21 20:28:05,250 INFO client.DefaultHadoopFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-21 20:28:06,662 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2021-05-21 20:28:06,916 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1621608943095_0001
2021-05-21 20:28:08,426 INFO input.FileInputFormat: Total input files to process : 1
2021-05-21 20:28:09,107 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1621608943095_0001
2021-05-21 20:28:09,741 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-21 20:28:10,029 INFO conf.Configuration: resource-types.xml not found
2021-05-21 20:28:10,030 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-21 20:28:10,676 INFO impl.YarnClientImpl: Submitted application application_1621608943095_0001
2021-05-21 20:28:11,005 INFO mapreduce.Job: The url to track the job: http://LAPTOP-3G329ESD:8088/proxy/application_1621608943095_0001/
2021-05-21 20:28:11,006 INFO mapreduce.Job: Running job: job_1621608943095_0001
2021-05-21 20:28:29,385 INFO mapreduce.Job: Job job_1621608943095_0001 running in uber mode : false
2021-05-21 20:28:29,389 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-21 20:28:40,664 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-21 20:28:50,832 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-21 20:28:50,965 INFO mapreduce.Job: Job job_1621608943095_0001 completed successfully
2021-05-21 20:28:59,178 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=59082
    FILE: Number of bytes written=648091
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=894860
    HDFS: Number of bytes written=74
    HDFS: Number of read operations=0
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=8077
    Total time spent by all reduces in occupied slots (ms)=7511
    Total time spent by all map tasks (ms)=8077
    Total time spent by all reduce tasks (ms)=7511
    Total vcore-milliseconds taken by all map tasks=8077
    Total vcore-milliseconds taken by all reduce tasks=7511
    Total megabyte-milliseconds taken by all map tasks=8270848
    Total megabyte-milliseconds taken by all reduce tasks=7691264

```

```

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /meanmax_output/*

```

```

01      4
02      0
03      7
04     44
05    100
06    168
07    219
08    198
09    141
10    100
11     19
12      3

```

```

C:\hadoop-3.3.0\sbin>

```

**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.**

**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 = "[_!$#<>\\^=\\[\\]\\*\\/\\\\\\.,;\\.\\-:()?!\"'"]";

    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, " ");
        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<Text, IntWritable, Text, IntWritable> {

    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();

        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<Object, Text, Text, IntWritable> {
```

```

private static final IntWritable one = new IntWritable(1);

private Text word = new Text();

private String tokens = "[_!$#<>\\^=\\[\\]\\*\\/\\\\\\,;\\.\\|\\:()?!\"'"]";

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, " ");

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));
    }
}
}

```

```

C:\hadoop-3.3.0\sbin>jps
11072 DataNode
20528 Jps
5620 ResourceManager
15532 NodeManager
6140 NameNode

C:\hadoop-3.3.0\sbin>hdfs dfs -mkdir /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /
Found 1 items
drwxr-xr-x - Anusree supergroup          0 2021-05-08 19:46 /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -copyFromLocal C:\input.txt /input_dir

C:\hadoop-3.3.0\sbin>hdfs dfs -ls /input_dir
Found 1 items
-rw-r--r-- 1 Anusree supergroup          36 2021-05-08 19:48 /input_dir/input.txt

C:\hadoop-3.3.0\sbin>hdfs dfs -cat /input_dir/input.txt
hello
world
hello
hadoop
bye

```

```

C:\hadoop-3.3.0\sbin>hadoop jar C:\sort.jar samples.topn.TopN /input_dir/input.txt /output_dir
2021-05-08 19:54:54,582 INFO client.DefaultHadoopFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2021-05-08 19:54:55,291 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/Anusree/.staging/job_1620483374279_0001
2021-05-08 19:54:55,821 INFO input.FileInputFormat: Total input files to process : 1
2021-05-08 19:54:56,261 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1620483374279_0001
2021-05-08 19:54:56,552 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-08 19:54:56,843 INFO conf.Configuration: resource-types.xml not found
2021-05-08 19:54:56,843 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-08 19:54:57,387 INFO impl.YarnClientImpl: Submitted application application_1620483374279_0001
2021-05-08 19:54:57,507 INFO mapreduce.Job: The url to track the job: http://LAPTOP-JG329ESD:8088/proxy/application_1620483374279_0001/
2021-05-08 19:54:57,508 INFO mapreduce.Job: Running job: job_1620483374279_0001
2021-05-08 19:55:13,792 INFO mapreduce.Job: Job job_1620483374279_0001 running in uber mode : false
2021-05-08 19:55:13,794 INFO mapreduce.Job: map 0% reduce 0%
2021-05-08 19:55:20,020 INFO mapreduce.Job: map 100% reduce 0%
2021-05-08 19:55:27,116 INFO mapreduce.Job: map 100% reduce 100%
2021-05-08 19:55:33,199 INFO mapreduce.Job: Job job_1620483374279_0001 completed successfully
2021-05-08 19:55:33,334 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=65
    FILE: Number of bytes written=530397
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=142
    HDFS: Number of bytes written=31
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0

```