

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**

## **BIG DATA ANALYTICS** **(20CS6PEBDA)**

*Submitted by*

**ANUSHREE HARRISH (1BM20CS020)**

*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-2023 to July-2023**

**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 **ANUSHREE HARRISH (1BM20CS020)**, 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 2023. 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.

**Dr. SYED AKRAM**  
Associate Professor  
Department of CSE  
BMSCE, Bengaluru

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

### Index Sheet

Sl. No.	Experiment Title	Page No.
1	MongoDB-CRUD DEMONSTATION	4
2	Cassandra	9
3	Operations Using Cassandra	12
4	Screenshot of Hadoop Installation	14
5	HDFS Commands	14
6	Temperature MapReduce Program	17
7	MapReduce Program to Sort	24
8	Demonstrating Join Operation in Map Reduce	28
9	Print WordCount on Scala Shell	31
10	Use RDD and FlatMap to count	32

### Course Outcome

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

## LAB 1:

### CREATE DATABASE IN MONGODB.

```
> use newdb
```

```
switched to db newdb db;
```

```
newdb
```

```
show dbs; admin          0.000GB config 0.000GB local          0.000GB
```

### CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS

To create a collection by the name "Student". Let us take a look at the collection list prior to the creation of the new collection "Student".

```
db.createCollection("Student");      =>      sql equivalent
```

```
CREATE TABLE STUDENT(...);
```

```
{ "ok" : 1 }
```

To drop a collection by the name "Student".

```
db.Student.drop();
```

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

```
db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing"});
```

```
WriteResult({ "nInserted" : 1 })
```

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

```
db.Student.update({_id:3,StudName:"AryanDavid",Grade:" VII"},{$set:{Hobbies:"Skating"}},{upsert:true});
```

```
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 3 })
```

### FIND METHOD

To search for documents from the "Students" collection based on certain search criteria.

```
db.Student.find({StudName:"AryanDavid"}); ({cond..},{columns.. column:1, columnname:0}      )
```

```
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : "Skating" }
```

To display only the StudName and Grade from all the documents of the Students collection. The identifier\_id should be suppressed and NOT displayed. db.Student.find({}, {StudName:1,Grade:1,\_id:0});

```
{ "StudName" : "MichelleJacintha", "Grade" : "VII" }
```

```
{ "Grade" : "VII", "StudName" : "AryanDavid" }
```

To find those documents where the Grade is set to 'VII'

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

```
{
  "_id" : 1,
  "StudName" : "MichelleJacintha", "Grade" : "VII",
  "Hobbies" : "InternetSurfing"
}
```

```
{
  "_id" : 3, "Grade" : "VII",
  "StudName" : "AryanDavid", "Hobbies" : "Skating"
}
```

To find those documents from the Students collection where the Hobbies is set to either 'Chess' or is set to 'Skating'.

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

```
{
  "_id" : 3, "Grade" : "VII",
  "StudName" : "AryanDavid", "Hobbies" : "Skating" }
```

To find documents from the Students collection where the StudName begins with "M".

```
db.Student.find({StudName:/^M/}).pretty();
{
  "_id" : 1,
  "StudName" : "MichelleJacintha", "Grade" : "VII",
  "Hobbies" : "InternetSurfing"
}
```

To find documents from the Students collection where the StudName has an "e" in any position.

```
db.Student.find({StudName:/e/}).pretty();
{
  "_id" : 1,
  "StudName" : "MichelleJacintha", "Grade" : "VII",
  "Hobbies" : "InternetSurfing"
}
```

To find the number of documents in the Students collection.

```
db.Student.count(); 2
```

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

```
db.Student.find().sort({StudName:-1}).pretty();
{
  "_id" : 1,
  "StudName" : "MichelleJacintha", "Grade" : "VII",
  "Hobbies" : "InternetSurfing"
}
{
  "_id" : 3, "Grade" : "VII",
  "StudName" : "AryanDavid", "Hobbies" : "Skating"
}
```

Import data from a CSV file

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

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

Export data to a CSV file

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

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

Save Method :

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

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

```
WriteResult({ "nInserted" : 1 })
```

Add a new field to existing Document:

```
db.Student.update({_id:ObjectId("625695cc7d129fb98b44c8a1")}
```

```
{ $set: { Location: "Network" } }
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Remove the field in an existing Document db.Student.update({\_id:ObjectId("625695cc7d129fb98b44c8a1 ")}, { \$unset: { Location: "Network" } })

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Finding Document based on search criteria suppressing few fields

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

```
{ "StudName" : "MichelleJacintha", "Grade" : "VII" }
```

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

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

```
{
  "_id" : ObjectId("625695cc7d129fb98b44c8a1"), "StudName" : "Vamsi",
```

```
  "Grade" : "VI"
```

```
}
```

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

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

```
{
  "_id" : 1,
  "StudName" : "MichelleJacintha", "Grade" : "VII",
  "Hobbies" : "InternetSurfing"
}
```

to set a particular field value to NULL

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

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Count the number of documents in Student Collections

```
db.Student.count() 3
```

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

```
db.Student.count({Grade:"VII"})
```

2 retrieve first 3 documents

```
db.Student.find({Grade:"VII"}).limit(1).pretty();
```

```
{
  "_id" : 1,
  "StudName" : "MichelleJacintha", "Grade" : "VII",
  "Hobbies" : "InternetSurfing"
}
```

Sort the document in Ascending order

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

```
{
  "_id" : 3, "Grade" : "VII",
  "StudName" : "AryanDavid", "Hobbies" : "Skating", "Location" : null
}
```

```

{
  "_id" : 1,
  "StudName" : "MichelleJacintha", "Grade" : "VII",
  "Hobbies" : "InternetSurfing"
}
{
  "_id" : ObjectId("625695cc7d129fb98b44c8a1"), "StudName" : "Vamsi",
  "Grade" : "VI"
}

```

Note: for descending order : db.Students.find().sort({StudName:- 1}).pretty();

to Skip the 1<sup>st</sup> two documents from the Students Collections

```

db.Student.find().skip(2).pretty()
{
  "_id" : ObjectId("625695cc7d129fb98b44c8a1"), "StudName" : "Vamsi",
  "Grade" : "VI"
}

```

Create a collection by name “food” and add to each document add a “fruits” array

```

db.food.insert( { _id:1, fruits:['grapes','mango','apple'] } )
db.food.insert( { _id:2, fruits:['grapes','mango','cherry'] } )
db.food.insert( { _id:3, fruits:['banana','mango'] } )
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
{ "_id" : 3, "fruits" : [ "banana", "mango" ] }

```

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

```

db.food.find ( {fruits: ['grapes','mango','apple'] } ). pretty();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }

```

To find in “fruits” array having “mango” in the first index position.

```

db.food.find ( {“fruits.1”:grapes} )

```

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

```

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

```

```

{ "_id" : 3, "fruits" : [ "banana", "mango" ] }

```

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

```

db.food.find({_id:1},{“fruits”:$slice:2}})
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }

```

To find all the documents from the food collection which have elements mango and grapes in the array “fruits”

```

db.food.find({fruits:{$all:["mango","grapes"]}})
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }

```

update on Array: using particular id replace the element present in the 1<sup>st</sup> index position of the fruits array with apple

```

db.food.update({_id:3},{“$set:{‘fruits.1’:‘apple’}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

```

insert new key value pairs in the fruits array

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

```
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
```

```
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
```

```
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
```

Note: perform query operations using - pop, addToSet, pullAll and pull



## LAB 2:

Perform the following DB operations using Cassandra.

Create a key space by name Employee

```
create keyspace "Employee" with replication =  
{'class':'SimpleStrategy','replication_factor':1}; cqlsh> use Employee;
```

Create a column family by name Employee-Info with attributes Emp\_Id Primary Key, Emp\_Name, Designation, Date\_of\_Joining, Salary, Dept\_Name

```
create table Employee_Info(Emp_id int PRIMARY KEY,Emp_name text,Date_of_Joining timestamp,Salary  
float,Dept_Name text) ;
```

Insert the values into the table in batch cqlsh:employee> begin batch

... insert into

```
Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(1,'Jack','2021-04-23',50000,'CSE')
```

```
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(2,'Tarun','2020-06-21',10000,'ISE')
```

```
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(3,'Suresh','2011-02-12',30000,'ECE')
```

```
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(4,'Yuresh','2015-09-02',90000,'EEE')... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name)
```

```
values(5,'Dharmesh','2016-01-09',70000,'CSE')
```

... apply batch;

```
cqlsh> create keyspace Employee with replication = {'class':'SimpleStrategy',  
'replication_factor':1};  
cqlsh> use Employee  
... ;  
cqlsh:employee> create table Employee_Info(Emp_id int PRIMARY KEY,Emp_name text,  
Date_of_Joining timestamp,Salary float,Dept_Name text);  
cqlsh:employee> begin batch  
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(1,'Nithin','2021-04-23',50000,'CSE')  
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(2,'Tarun','2020-06-21',10000,'ISE')  
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(3,'Suresh','2011-02-12',30000,'ECE')  
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(4,'Yuresh','2015-09-02',90000,'EEE')  
... insert into Employee_Info(Emp_id,Emp_name,Date_of_Joining,Salary,Dept_Name) values(5,'Dharmesh','2016-01-09',70000,'CSE')  
... apply batch;  
cqlsh:employee> select * from Employee_info;
```

emp_id	date_of_joining	dept_name	emp_name	salary
5	2016-01-09 00:00:00.000000+0000	CSE	Dharmesh	70000
1	2021-04-23 00:00:00.000000+0000	CSE	Nithin	50000
2	2020-06-21 00:00:00.000000+0000	ISE	Tarun	10000
4	2015-09-02 00:00:00.000000+0000	EEE	Yuresh	90000
3	2011-02-12 00:00:00.000000+0000	ECE	Suresh	30000

Update Employee name and Department of Emp-Id 1 update employee\_info set

Dept\_Name='Mech',emp\_name='Sreekar'where emp\_id=1;

```
cqlsh:employee> select * from employee_info;
```

```
cqlsh:employee> select * from employee_info;
```

emp_id	date_of_joining	dept_name	emp_name	salary
5	2016-01-09 00:00:00.000000+0000	CSE	Dharmesh	70000
1	2021-04-23 00:00:00.000000+0000	Mech	Sreekar	50000
2	2020-06-21 00:00:00.000000+0000	ISE	Tarun	10000
4	2015-09-02 00:00:00.000000+0000	EEE	Yuresh	90000
3	2011-02-12 00:00:00.000000+0000	ECE	Suresh	30000

(5 rows)

Sort the details of Employee records based on salary

(0 rows)

```
cqlsh:employee> begin batch
```

```
... insert into Employee_information(Emp_id,Emp_name,Date_of_Joi  
ning,Salary,Dept_Name) values(1,'Nithin','2021-04-23',50000,'CSE')
```

```
... insert into Employee_information(Emp_id,Emp_name,Date_of_Joi  
ning,Salary,Dept_Name) values(2,'Tarun','2020-06-21',10000,'ISE')
```

```
... insert into Employee_information(Emp_id,Emp_name,Date_of_Joi  
ning,Salary,Dept_Name) values(3,'Suresh','2011-02-12',30000,'ECE')
```

```
... apply batch;
```

```
cqlsh:employee> select * from Employee_information;
```

emp_id	salary	date_of_joining	dept_name	emp_name
1	50000	2021-04-23 00:00:00.000000+0000	CSE	Nithin
2	10000	2020-06-21 00:00:00.000000+0000	ISE	Tarun
3	30000	2011-02-12 00:00:00.000000+0000	ECE	Suresh

(3 rows)

```
cqlsh:employee> describe Employee_information;
```

```
CREATE TABLE employee.employee_information (  
  emp_id int,  
  salary float,  
  date_of_joining timestamp,  
  dept_name text,  
  emp_name text,  
  PRIMARY KEY (emp_id, salary)  
) WITH CLUSTERING ORDER BY (salary ASC)
```



cqlsh:employee> select \* from Employee\_information where emp\_id in (1,2,3) order by Salary;

cqlsh:employee> paging off  
Disabled Query paging.  
cqlsh:employee> select \* from Employee\_information where emp\_id in (1,2,3) order by Salary;

emp_id	salary	date_of_joining	dept_name	emp_name
2	10000	2020-06-21 00:00:00.000000+0000	ISE	Tarun
3	30000	2011-02-12 00:00:00.000000+0000	ECE	Suresh
1	50000	2021-04-23 00:00:00.000000+0000	CSE	Nithin

(3 rows)

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

cqlsh:employee> alter table employee\_info add projects set<text>;

Update the altered table to add project names. cqlsh:employee> update employee\_info set projects=projects+{'project1','project2','project3'} where emp\_id=1;

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

emp_id	date_of_joining	dept_name	emp_name	projects	salary
5	2016-01-09 00:00:00.000000+0000	CSE	Dharmesh	null	70000
1	2021-04-23 00:00:00.000000+0000	Mech	Sreekar	{'project1', 'project2', 'project3'}	50000
2	2020-06-21 00:00:00.000000+0000	ISE	Tarun	null	10000
4	2015-09-02 00:00:00.000000+0000	EEE	Yuresh	null	90000
3	2011-02-12 00:00:00.000000+0000	ECE	Suresh	null	30000

(5 rows)

8 Create a TTL of 15 seconds to display the values of Employees.

cqlsh:employee> begin batch  
... insert into Employee\_Info(Emp\_id,Emp\_name,Date\_of\_Joining,Salary,Dept\_Name) values(6,'Rahul','2021-05-03',10000,'ISE') USING TTL 15;  
... apply batch;  
cqlsh:employee> select \* from employee\_info;

emp_id	date_of_joining	dept_name	emp_name	projects	salary
5	2016-01-09 00:00:00.000000+0000	CSE	Dharmesh	null	70000
1	2021-04-23 00:00:00.000000+0000	Mech	Sreekar	{'project1', 'project2', 'project3'}	50000
2	2020-06-21 00:00:00.000000+0000	ISE	Tarun	{'project4', 'project5'}	10000
4	2015-09-02 00:00:00.000000+0000	EEE	Yuresh	null	90000
6	2021-05-03 00:00:00.000000+0000	ISE	Rahul	null	10000
3	2011-02-12 00:00:00.000000+0000	ECE	Suresh	null	30000

(6 rows)

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

emp_id	date_of_joining	dept_name	emp_name	projects	salary
5	2016-01-09 00:00:00.000000+0000	CSE	Dharmesh	null	70000
1	2021-04-23 00:00:00.000000+0000	Mech	Sreekar	{'project1', 'project2', 'project3'}	50000
2	2020-06-21 00:00:00.000000+0000	ISE	Tarun	{'project4', 'project5'}	10000
4	2015-09-02 00:00:00.000000+0000	EEE	Yuresh	null	90000
3	2011-02-12 00:00:00.000000+0000	ECE	Suresh	null	30000

(5 rows)

### LAB 3:

Create a key space by name Library

```
cqlsh> create keyspace Library WITH REPLICATION = {'class' : 'SimpleStrategy', 'replication_factor' : 1};
cqlsh> use Library;
```

Create a column family by name Library-Info with attributes Stud\_Id Primary Key, Counter\_value of type Counter,

```
cqlsh:library> create table Library_Info(Stud_Id int,Counter_value counter,Stud_Name varchar,Book_name varchar,Book_id int,Date_of_Issue date,primary key(Stud_Id,Stud_name,Book_name,Book_id,Date_of_Issue));
```

Insert the values into the table in batch

```
cqlsh:library> insert into Library_Info (Stud_Id,Counter_value,Stud_Name,Book_name,Book_id,Date_of_Issue) values (1,1,'naman','abc',123,'2022-05-04');
cqlsh:library> insert into Library_Info (Stud_Id,Counter_value,Stud_Name,Book_name,Book_id,Date_of_Issue) values (1,2,'naman','abc',123,'2022-05-04');
```

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

```
cqlsh:library> update library_info set Counter_value = Counter_value + 1 where Stud_Id = 1 AND Stud_name = 'naman' AND Book_name='abc' AND Book_id = 123 AND Date_of_Issue = '2022-05-04';
cqlsh:library> select * from Library_Info;
```

Stud_Id	stud_name	book_name	book_id	date_of_Issue	counter_value
1	naman	abc	123	2022-05-04	2

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

```
cqlsh:library> select counter_value as borrow_count from library_info where stud_id=1 AND book_id=123;
```

borrow_count
2

Export the created column to a csv file

```
cqlsh:library> COPY library.library_info (stud_id,book_id,counter_value,stud_name,book_name,date_of_issue) TO '/home/bmsce/CASSANDRA-NAMAN/data.csv' WITH HEADER = TRUE;
Using 11 child processes

Starting copy of library.library_info with columns [stud_id, book_id, counter_value, stud_name, book_name, date_of_issue].
Processed: 1 rows; Rate:      6 rows/s; Avg. rate:      6 rows/s
1 rows exported to 1 files in 0.176 seconds.
```

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

```
cqlsh:library> COPY library.library_info (stud_id,book_id,counter_value,stud_name,book_name,date_of_issue) FROM '/home/bmsce/CASSANDRA-NAMAN/data.csv' WITH HEADER = TRUE;
Using 11 child processes

Starting copy of library.library_info with columns [stud_id, book_id, counter_value, stud_name, book_name, date_of_issue].
Processed: 1 rows; Rate:      2 rows/s; Avg. rate:      3 rows/s
1 rows imported from 1 files in 0.379 seconds (0 skipped).
```



[illegible]

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -du -h -v /sample1/test.txt
SIZE      DISK_SPACE_CONSUMED_WITH_ALL_REPLICAS  FULL_PATH_NAME
1.0 K      1.0 K                                  /sample1/test.txt

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -find /sample1 -name *.txt -print
/sample1/test.txt

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -ls /sample1
Found 1 items
-rw-r--r--   1 Avi supergroup      1032 2021-04-19 15:26 /sample1/test.txt
```

C:\> Administrator: Command Prompt

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>start-dfs.cmd

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>start-yarn.cmd
starting yarn daemons

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -mkdir /dir1

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -ls /
Found 2 items
drwxr-xr-x   - Avi supergroup          0 2021-04-19 14:33 /dir1
drwxr-xr-x   - Avi supergroup          0 2021-04-19 14:19 /sample

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -mkdir /dir1

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -mv /sample1/test.txt /dir1

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -ls /dir1
Found 1 items
-rw-r--r--   1 Avi supergroup        1232 2021-04-19 15:35 /dir1/test.txt
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -rm /dir1/test.txt
Deleted /dir1/test.txt
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -count /sample1/test.txt
      0      1      1032 /sample1/test.txt

C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -count /sample1
      1      1      1032 /sample1
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -cat /sample1/test.txt
Hi, You are champ
```

#### HOW TO INSTALL APACHE HADOOP 2.6.0 IN UBUNTU (SINGLE NODE SETUP)

Since we know it's the time for parallel computation to tackle large amount of dataset, we will require Apache Hadoop (here the name is derived from Elephant). As Apache Hadoop is the top most contributed Apache project, more and more features are implemented as well as more and more bugs are getting fixed in new coming versions. So, by considering this situation we need to follow slightly different steps than previous version. Here, I am trying to covering full fledged Hadoop installation steps for BigData enthusiasts who wish to install Apache Hadoop on their Ubuntu / Linux machine.

This blog post teaches how to install Apache Hadoop 2.6 over Ubuntu machine. (You can follow the same blog post for installation over Ubuntu server machine). To get started with Apache Hadoop install, I recommend that you should have knowledge of basic Linux commands which will be helpful in normal operations while installation task.

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hadoop fs -count -e /sample1/test.txt
count: '/sample1/test.txt': No such file or directory
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -count /sample1/test.txt
0          1          1032 /sample1/test.txt
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -copyFromLocal D:\Rohit\test.txt /sample1
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -ls /sample1
Found 1 items
-rw-r--r--  1 Avi supergroup      1032 2021-04-19 15:26 /sample1/test.txt
```

```
C:\Users\avina\Downloads\hadoop-3.1.0.tar\hadoop-3.1.0\sbin>hdfs dfs -cat /sample1/test.txt
```

Hi, You are champ

#### HOW TO INSTALL APACHE HADOOP 2.6.0 IN UBUNTU (SINGLE NODE SETUP)

Since we know it's the time for parallel computation to tackle large amount of dataset, we will require Apache Hadoop (here the name is derived from Elephant). As Apache Hadoop is the top most contributed Apache project, more and more features are implemented as well as more and more bugs are getting fixed in new coming versions. So, by considering this situation we need to follow slightly different steps than previous version. Here, I am trying to covering full fledged Hadoop installation steps for BigData enthusiasts who wish to install Apache Hadoop on their Ubuntu / Linux machine.

This blog post teaches how to install Apache Hadoop 2.6 over Ubuntu machine. (You can follow the same blog post for installation over Ubuntu server machine). To get started with Apache Hadoop install, I recommend that you should have knowledge of basic Linux commands which will be helpful in normal operations while installation task.

It will print all the directories present in HDFS.

bin directory contains executables so, bin/hdfs means

we want the executables of hdfs particularly dfs(Distributed File System) commands.



LAB 6:

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ 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-bmscePrecision-T1700.out
```

```
hduser@localhost's password:
```

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

starting yarn daemons

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

```
hduser@localhost's password:
```

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

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ jps
```

6832 NodeManager

6498 ResourceManager

6339 SecondaryNameNode

4887 org.eclipse.equinox.launcher\_1.5.600.v20191014-2022.jar

6954 Jps

6123 DataNode

5951 NameNode

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /
```

-ls: Unknown command

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /
```

Found 31 items

drwxr-xr-x - hduser supergroup 0 2022-06-06 12:35 /CSE

drwxr-xr-x - hduser supergroup 0 2022-06-06 12:23 /FFF

drwxr-xr-x - hduser supergroup 0 2022-06-06 12:36 /LLL

drwxr-xr-x - hduser supergroup 0 2022-06-20 12:06 /amit\_bda

drwxr-xr-x - hduser supergroup 0 2022-06-27 11:42 /amit\_lab

drwxr-xr-x - hduser supergroup 0 2022-06-03 14:52 /bharath

drwxr-xr-x - hduser supergroup 0 2022-06-03 14:43 /bharath035

drwxr-xr-x - hduser supergroup 0 2022-06-24 14:54 /chi

drwxr-xr-x - hduser supergroup 0 2022-05-31 10:21 /example

drwxr-xr-x - hduser supergroup 0 2022-06-01 15:13 /foldernew

drwxr-xr-x - hduser supergroup 0 2022-06-06 15:04 /hemang061

drwxr-xr-x - hduser supergroup 0 2022-06-20 15:16 /input\_Jack

drwxr-xr-x - hduser supergroup 0 2022-06-03 12:27 /irfan

drwxr-xr-x - hduser supergroup 0 2022-06-22 10:44 /lwde

drwxr-xr-x - hduser supergroup 0 2022-06-27 13:03 /mapreducejoin\_amit

drwxr-xr-x - hduser supergroup 0 2022-06-22 15:32 /muskan

drwxr-xr-x - hduser supergroup 0 2022-06-22 15:06 /muskan\_op

drwxr-xr-x - hduser supergroup 0 2022-06-22 15:35 /muskan\_output

drwxr-xr-x - hduser supergroup 0 2022-06-06 15:04 /new\_folder

drwxr-xr-x - hduser supergroup 0 2022-05-31 10:26 /one

drwxr-xr-x - hduser supergroup 0 2022-06-24 15:30 /out55

drwxr-xr-x - hduser supergroup 0 2022-06-20 12:17 /output

```

drwxr-xr-x - hduser supergroup 0 2022-06-27 13:04 /output_TOPn
drwxr-xr-x - hduser supergroup 0 2022-06-27 12:14 /output_Topn
drwxr-xr-x - hduser supergroup 0 2022-06-24 12:42 /r1
drwxr-xr-x - hduser supergroup 0 2022-06-24 12:24 /rgs
drwxr-xr-x - hduser supergroup 0 2022-06-03 12:08 /saurab
drwxrwxr-x - hduser supergroup 0 2019-08-01 16:19 /tmp
drwxr-xr-x - hduser supergroup 0 2019-08-01 16:03 /user
drwxr-xr-x - hduser supergroup 0 2022-06-01 09:46 /user1
-rw-r--r-- 1 hduser supergroup 2436 2022-06-24 12:17 /wc.jar
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -mkdir /Jack_temperature
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -put ./1901 /Jack_temperature
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -put ./1902 /Jack_temperature
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /Jack_temperature
Found 2 items
-rw-r--r-- 1 hduser supergroup 888190 2022-06-27 14:47 /Jack_temperature/1901
-rw-r--r-- 1 hduser supergroup 888978 2022-06-27 14:47 /Jack_temperature/1902
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hadoop jar ./avgtemp.jar AverageDriver
/Jack_temperature/1901 /Jack_temperature/output/
Exception in thread "main" java.lang.ClassNotFoundException: 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:~/Desktop/temperature$ hadoop jar ./avgtemp.jar
temperature.AverageDriver /Jack_temperature/1901 /Jack_temperature/output/
22/06/27 14:53:27 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/27 14:53:27 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker,
sessionId=
22/06/27 14:53:27 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
22/06/27 14:53:27 INFO input.FileInputFormat: Total input paths to process : 1
22/06/27 14:53:27 INFO mapreduce.JobSubmitter: number of splits:1
22/06/27 14:53:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local254968295_0001
22/06/27 14:53:28 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/27 14:53:28 INFO mapreduce.Job: Running job: job_local254968295_0001
22/06/27 14:53:28 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/27 14:53:28 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Starting task:
attempt_local254968295_0001_m_000000_0

22/06/27 14:53:28 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
22/06/27 14:53:28 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/Jack_temperature/1901:0+888190
22/06/27 14:53:28 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/27 14:53:28 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/27 14:53:28 INFO mapred.MapTask: soft limit at 83886080
22/06/27 14:53:28 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/27 14:53:28 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/27 14:53:28 INFO mapred.MapTask: Map output collector class =

```

org.apache.hadoop.mapred.MapTask\$MapOutputBuffer  
22/06/27 14:53:28 INFO mapred.LocalJobRunner:  
22/06/27 14:53:28 INFO mapred.MapTask: Starting flush of map output  
22/06/27 14:53:28 INFO mapred.MapTask: Spilling map output  
22/06/27 14:53:28 INFO mapred.MapTask: bufstart = 0; bufend = 59076; bufvoid = 104857600  
22/06/27 14:53:28 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend =  
26188144(104752576);  
length = 26253/6553600  
22/06/27 14:53:28 INFO mapred.MapTask: Finished spill 0  
22/06/27 14:53:28 INFO mapred.Task: Task:attempt\_local254968295\_0001\_m\_000000\_0 is done. And is in  
the process of committing  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: map  
22/06/27 14:53:28 INFO mapred.Task: Task 'attempt\_local254968295\_0001\_m\_000000\_0' done.  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Finishing task:  
attempt\_local254968295\_0001\_m\_000000\_0  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: map task executor complete.  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Waiting for reduce tasks  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: Starting task: attempt\_local254968295\_0001\_r\_000000\_0  
22/06/27 14:53:28 INFO mapred.Task: Using ResourceCalculatorProcessTree : []  
22/06/27 14:53:28 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:  
org.apache.hadoop.mapreduce.task.reduce.Shuffle@262cb2a9  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=349752512,  
maxSingleShuffleLimit=87438128, mergeThreshold=230836672, ioSortFactor=10,  
memToMemMergeOutputsThreshold=10  
22/06/27 14:53:28 INFO reduce.EventFetcher: attempt\_local254968295\_0001\_r\_000000\_0 Thread started:  
EventFetcher for fetching Map Completion Events  
22/06/27 14:53:28 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map  
attempt\_local254968295\_0001\_m\_000000\_0 decomp: 72206 len: 72210 to MEMORY  
22/06/27 14:53:28 INFO reduce.InMemoryMapOutput: Read 72206 bytes from map-output for  
attempt\_local254968295\_0001\_m\_000000\_0  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 72206,  
inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory -> 72206  
22/06/27 14:53:28 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0  
on-disk map-outputs  
22/06/27 14:53:28 INFO mapred.Merger: Merging 1 sorted segments  
  
22/06/27 14:53:28 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total  
size: 72199 bytes  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merged 1 segments, 72206 bytes to disk to satisfy  
reduce memory limit  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merging 1 files, 72210 bytes from disk  
22/06/27 14:53:28 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce  
22/06/27 14:53:28 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 14:53:28 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total  
size: 72199 bytes  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 14:53:28 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use  
mapreduce.job.skiprecords  
22/06/27 14:53:28 INFO mapred.Task: Task:attempt\_local254968295\_0001\_r\_000000\_0 is done. And is in  
the process of committing  
22/06/27 14:53:28 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 14:53:28 INFO mapred.Task: Task attempt\_local254968295\_0001\_r\_000000\_0 is allowed to  
commit

now

22/06/27 14:53:28 INFO output.FileOutputCommitter: Saved output of task

'attempt\_local254968295\_0001\_r\_000000\_0' to

hdfs://localhost:54310/Jack\_temperature/output/\_temporary/0/task\_local254968295\_0001\_r\_000000

22/06/27 14:53:28 INFO mapred.LocalJobRunner: reduce > reduce

22/06/27 14:53:28 INFO mapred.Task: Task 'attempt\_local254968295\_0001\_r\_000000\_0' done.

22/06/27 14:53:28 INFO mapred.LocalJobRunner: Finishing task:

attempt\_local254968295\_0001\_r\_000000\_0

22/06/27 14:53:28 INFO mapred.LocalJobRunner: reduce task executor complete.

22/06/27 14:53:29 INFO mapreduce.Job: Job job\_local254968295\_0001 running in uber mode : false

22/06/27 14:53:29 INFO mapreduce.Job: map 100% reduce 100%

22/06/27 14:53:29 INFO mapreduce.Job: Job job\_local254968295\_0001 completed successfully

22/06/27 14:53:29 INFO mapreduce.Job: Counters: 38

File System Counters

FILE: Number of bytes read=153102

FILE: Number of bytes written=723014

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=112

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)=55

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:~/Desktop/temperature$ hdfs dfs -ls /Jack_temperature/output/
Found 2 items
```

```
-rw-r--r-- 1 hduser supergroup 0 2022-06-27 14:53 /Jack_temperature/output/_SUCCESS
```

```
-rw-r--r-- 1 hduser supergroup 8 2022-06-27 14:53 /Jack_temperature/output/part-r00000
```

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -cat /Jack_temperature/output/part-r-00000
```

```
1901 46
```

```
FILE: Number of bytes written=723014
FILE: Number of read operations=0 FILE:
Number of large read operations=0 FILE:
Number of write operations=0 HDFS:
Number of bytes read=1776360 HDFS:
Number of bytes written=0
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=6563
Map output records=6564
Map output bytes=59076
Map output materialized bytes=72210
Input split bytes=112
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=13126
Shuffled Maps =1
Failed Shuffles=0 Merged
Map outputs=1 GC time
elapsed (ms)=65 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=688190
File Output Format Counters
Bytes Written=8
```

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /jack_temperature/output/ Found 2
```

```
items
```

```
-rw-r--r-- 1 hduser supergroup 0 2022-06-27 14:53 /jack_temperature/output/_SUCCESS
```

```
-rw-r--r-- 1 hduser supergroup 8 2022-06-27 14:53 /jack_temperature/output/part-r-00000
```

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -cat /jack_temperature/output/part-r-00000
```

```
1901 46
```

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$
```

Max Temp:  
Driver class:

```
package temperatureMax;
```

```
import org.apache.hadoop.io.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

```
public class TempDriver
```

```
{
    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(TempDriver.class);
        job.setJobName("Max temperature");

        FileInputFormat.addInputPath(job,new Path(args[0]));
        FileOutputFormat.setOutputPath(job,new Path (args[1]));

        job.setMapperClass(TempMapper.class);
        job.setReducerClass(TempReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        System.exit(job.waitForCompletion(true)?0:1);
    }
}
```

Mapper Class;

```
package temperatureMax;
```

```
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;
```

```
public class TempMapper extends Mapper <LongWritable, Text, Text, IntWritable>
```

```
{
    public static final int MISSING = 9999;
```

```
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException
```

```
{
        String line = value.toString();
        String month = line.substring(19,21);
        int temperature;
        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 != MISSING && quality.matches("[01459]"))
```

```

        context.write(new Text(month),new IntWritable(temperature));
    }
}

```

Reducer Class:

```
package temperatureMax;
```

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;
```

```
public class TempReducer extends Reducer <Text, IntWritable,Text, IntWritable>
```

```
{
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws
IOException,InterruptedException
    {
        int max_temp = 0;

        for (IntWritable value : values)
        {
            if(max_temp<value.get()) {
                max_temp = value.get();
            }
        }
        context.write(key, new IntWritable(max_temp));
    }
}

```

```

hadoopgokankshitaz510:~/hadoop-3.2.1/sbin$ hdfs dfs -cat /output_max/temp/part-r-000000
2021-05-21 20:39:59,103 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2021-05-21 20:40:00,791 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
01      44
02      17
03      50
04     194
05     256
06     278
07     317
08     283
09     211
10     156
11      89
12     117

```

## LAB 7:

```
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -mkdir /Jack_topn
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -put ./input.txt /Jack_topn/
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hdfs dfs -ls /Jack_topn/
Found 1 items
-rw-r--r-- 1 hduser supergroup 103 2022-06-27 15:43 /Jack_topn/input.txt
hduser@bmsce-Precision-T1700:~/Desktop/temperature$ hadoop jar topn.jar TopNDriver
/Jack_topn/input.txt /Jack_topn/output
Exception in thread "main" java.lang.ClassNotFoundException: TopNDriver
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:~/Desktop/temperature$ hadoop jar topn.jar topn.TopNDriver
/Jack_topn/input.txt /Jack_topn/output
22/06/27 15:45:22 INFO Configuration.deprecation: session.id is deprecated. Instead, use
dfs.metrics.session-id
22/06/27 15:45:22 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker,
sessionId=
22/06/27 15:45:22 INFO input.FileInputFormat: Total input paths to process : 1
22/06/27 15:45:22 INFO mapreduce.JobSubmitter: number of splits:1
22/06/27 15:45:22 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local691635730_0001
22/06/27 15:45:22 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
22/06/27 15:45:22 INFO mapreduce.Job: Running job: job_local691635730_0001
22/06/27 15:45:22 INFO mapred.LocalJobRunner: OutputCommitter set in config null
22/06/27 15:45:22 INFO mapred.LocalJobRunner: OutputCommitter is
org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Waiting for map tasks
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Starting task:
attempt_local691635730_0001_m_000000_0
22/06/27 15:45:22 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
22/06/27 15:45:22 INFO mapred.MapTask: Processing split:
hdfs://localhost:54310/Jack_topn/input.txt:0+103
22/06/27 15:45:22 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
22/06/27 15:45:22 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
22/06/27 15:45:22 INFO mapred.MapTask: soft limit at 83886080
22/06/27 15:45:22 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
22/06/27 15:45:22 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
22/06/27 15:45:22 INFO mapred.MapTask: Map output collector class =
org.apache.hadoop.mapred.MapTask$MapOutputBuffer
22/06/27 15:45:22 INFO mapred.LocalJobRunner:
22/06/27 15:45:22 INFO mapred.MapTask: Starting flush of map output
22/06/27 15:45:22 INFO mapred.MapTask: Spilling map output
22/06/27 15:45:22 INFO mapred.MapTask: bufstart = 0; bufend = 187; bufvoid = 104857600
22/06/27 15:45:22 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend =
26214316(104857264);

length = 81/6553600
22/06/27 15:45:22 INFO mapred.MapTask: Finished spill 0
22/06/27 15:45:22 INFO mapred.Task: Task:attempt_local691635730_0001_m_000000_0 is done. And is in
the process of committing
22/06/27 15:45:22 INFO mapred.LocalJobRunner: map
```



22/06/27 15:45:22 INFO mapred.Task: Task 'attempt\_local691635730\_0001\_m\_000000\_0' done.  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Finishing task:  
attempt\_local691635730\_0001\_m\_000000\_0  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: map task executor complete.  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Waiting for reduce tasks  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: Starting task: attempt\_local691635730\_0001\_r\_000000\_0  
22/06/27 15:45:22 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]  
22/06/27 15:45:22 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin:  
org.apache.hadoop.mapreduce.task.reduce.Shuffle@40a5e65a  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=334338464,  
maxSingleShuffleLimit=83584616, mergeThreshold=220663392, ioSortFactor=10,  
memToMemMergeOutputsThreshold=10  
22/06/27 15:45:22 INFO reduce.EventFetcher: attempt\_local691635730\_0001\_r\_000000\_0 Thread started:  
EventFetcher for fetching Map Completion Events  
22/06/27 15:45:22 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map  
attempt\_local691635730\_0001\_m\_000000\_0 decomp: 231 len: 235 to MEMORY  
22/06/27 15:45:22 INFO reduce.InMemoryMapOutput: Read 231 bytes from map-output for  
attempt\_local691635730\_0001\_m\_000000\_0  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 231,  
inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->231  
22/06/27 15:45:22 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0  
on-disk map-outputs  
22/06/27 15:45:22 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 15:45:22 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total  
size: 226 bytes  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: Merged 1 segments, 231 bytes to disk to satisfy  
reduce memory limit  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: Merging 1 files, 235 bytes from disk  
22/06/27 15:45:22 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce  
22/06/27 15:45:22 INFO mapred.Merger: Merging 1 sorted segments  
22/06/27 15:45:22 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total  
size: 226 bytes  
22/06/27 15:45:22 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 15:45:22 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use  
mapreduce.job.skiprecords  
22/06/27 15:45:23 INFO mapred.Task: Task:attempt\_local691635730\_0001\_r\_000000\_0 is done. And is in  
the process of committing  
  
22/06/27 15:45:23 INFO mapred.LocalJobRunner: 1 / 1 copied.  
22/06/27 15:45:23 INFO mapred.Task: Task attempt\_local691635730\_0001\_r\_000000\_0 is allowed to  
commit  
now  
22/06/27 15:45:23 INFO output.FileOutputCommitter: Saved output of task  
'attempt\_local691635730\_0001\_r\_000000\_0' to  
hdfs://localhost:54310/Jack\_topn/output/\_temporary/0/task\_local691635730\_0001\_r\_000000  
22/06/27 15:45:23 INFO mapred.LocalJobRunner: reduce > reduce  
22/06/27 15:45:23 INFO mapred.Task: Task 'attempt\_local691635730\_0001\_r\_000000\_0' done.  
22/06/27 15:45:23 INFO mapred.LocalJobRunner: Finishing task:  
attempt\_local691635730\_0001\_r\_000000\_0  
22/06/27 15:45:23 INFO mapred.LocalJobRunner: reduce task executor complete.  
22/06/27 15:45:23 INFO mapreduce.Job: Job job\_local691635730\_0001 running in uber mode : false  
22/06/27 15:45:23 INFO mapreduce.Job: map 100% reduce 100%  
22/06/27 15:45:23 INFO mapreduce.Job: Job job\_local691635730\_0001 completed successfully

22/06/27 15:45:23 INFO mapreduce.Job: Counters: 38

File System Counters

FILE: Number of bytes read=18078

FILE: Number of bytes written=516697

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=206

HDFS: Number of bytes written=105

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=6

Map output records=21

Map output bytes=187

Map output materialized bytes=235

Input split bytes=110

Combine input records=0

Combine output records=0

Reduce input groups=15

Reduce shuffle bytes=235

Reduce input records=21

Reduce output records=15

Spilled Records=42

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=42

CPU time spent (ms)=0

Physical memory (bytes) snapshot=0

Virtual memory (bytes) snapshot=0

Total committed heap usage (bytes)=578289664

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=103

File Output Format Counters

Bytes Written=105

hduser@bmsce-Precision-T1700:~/Desktop/temperature\$ hdfs dfs -ls /Jack\_topn/output/

Found 2 items

-rw-r--r-- 1 hduser supergroup 0 2022-06-27 15:45 /Jack\_topn/output/\_SUCCESS

-rw-r--r-- 1 hduser supergroup 105 2022-06-27 15:45 /Jack\_topn/output/part-r-00000

hduser@bmsce-Precision-T1700:~/Desktop/temperature\$ hdfs dfs -cat /Jack\_topn/output/part-r-00000

hadoop 4

i3

am 2

hi 1

im 1  
is 1  
there 1  
bye 1  
learing 1  
awesome 1  
love 1  
Jack 1  
cool 1  
and 1  
using 1

```
-rw-r--r--      1 hduser supergroup      63 2022-06-20 15:16
_jack/output_jack/part-00000
hduser@bmsce-Precision-T1700: $ hdfs dfs -cat /input_jack/output_jack/part-0000 cat:
_jack/output_jack/part-0000: No such file or directory
hduser@bmsce-Precision-T1700: $ hdfs dfs -cat /input_jack/output_jack/part-00000 am      1
awesome      1
hadoop2
hi      1
i      1
im      1
is      1
jack      1
learing      1
```

LAB8:

DeptEmpStrengthMapper

```
package MapReduceJoin;
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 DeptEmpStrengthMapper 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]));
    }
}
```

DeptNameStrengthMapper:

```
package MapReduceJoin;
```

```
import java.io.IOException;
```

```
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
```

```
public class DeptNameMapper 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], "0"), new Text(SingleNodeData[1]));
    }
}
```

```
}
```

Join Driver

```
package MapReduceJoin;
```

```
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 'Department Emp Strength input' with 'Department Name input'");
```

```
        Path AInputPath = new Path(args[0]);
```

```
        Path BInputPath = new Path(args[1]);
```

```
        Path outputPath = new Path(args[2]);
```

```
        MultipleInputs.addInputPath(conf, AInputPath, TextInputFormat.class,  
DeptNameMapper.class);
```

```
        MultipleInputs.addInputPath(conf, BInputPath, TextInputFormat.class,  
DeptEmpStrengthMapper.class);
```

```
        FileOutputFormat.setOutputPath(conf, outputPath);
```

```
        conf.setPartitionerClass(KeyPartitioner.class);
```

```
        conf.setOutputValueGroupingComparator(TextPair.FirstComparator.class);
```

```
        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);
    }
}
Join Reducer

package MapReduceJoin;

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);
        }
    }
}

```

Input Split bytes=140

Combine input records=0  
Combine output records=0  
Reduce input groups=4  
Reduce shuffle bytes=140  
Reduce input records=8  
Reduce output records=4  
Spilled Records=16  
Shuffled Maps=2  
Failed Shuffles=0  
Merged Map outputs=2  
GC time elapsed (ms)=2  
CPU time spent (ms)=0

Physical memory (bytes) snapshot=0  
Virtual memory (bytes) snapshot=0  
Total committed heap usage (bytes)=913833984  
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=0

File Output Format Counters  
Bytes Written=80

hduser@bmsce-Precision-T1700:~/jack/join/MapReduceJoin\$ nois dis -cat /jack/join/output2/part-00000

A11	50	Finance
B12	100	HR
C13	250	Manufacturing
Dept_ID	Total_Employee	Dept_Name

hduser@bmsce-Precision-T1700:~/jack/join/MapReduceJoin\$

#### LAB9:

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

