Geetanjali Institute of Technical Studies

(Approved by AICTE, New Delhi and Affiliated to Rajasthan Technical University Kota (Raj.))

DABOK, UDAIPUR, RAJASTHAN 313022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B. Tech - VIII SEMESTER

ACADEMIC YEAR – 2021-22

BIG DATA ANALYTICS LAB
(8CS4-21)

Submitted To:
Ms.Monika Bhatt

Submitted by:

VISION & MISSION OF INSTITUTE

INSTITUTE VISION

TO ACHIEVE EXCELLENCE IN TECHNICAL AND MANAGEMENT EDUCATION THROUGH QUALITY TEACHING, RESEARCH AND INNOVATION.

INSTITUTE MISSION

TO PROVIDE A CONDUCIVE ENVIRONMENT IN ORDER TO PRODUCE SOCIALLY RESPONSIBLE AND PRODUCTIVE PROFESSIONALS.

VISION & MISSION OF DEPARTMENT

VISION

To nurture the students to become employable graduates who can provide solutions to the societal issues through ICT.

MISSION

To nurture knowledge of students in theoretical and practical aspects in collaboration with industries.

To inculcate the students towards research and innovation to fulfill the need of industry & society.

To develop socially responsible professionals with values and ethics.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

The Programme Educational Objectives of the programme offered by the department are listed below:

- **PEO1:**ANALYTICAL SKILLS
  1. To facilitate the graduates with the ability to visualize, gather information, articulate, analyze, solve complex problems, and make decisions. These are essential to address the challenges of complex and computation intensive problems increasing their productivity
- **PEO2:** – TECHNICAL SKILLS

  2. To facilitate the graduates with the technical skills that prepare them for immediate employment and pursue certification providing a deeper understanding of the technology in advanced areas of computer science and related fields.
- **PEO3:**SOFT SKILLS

  To facilitate the graduates with the soft skills that include fulfilling the mission, setting goals, showing self-confidence by communicating effectively, having a positive attitude, get involved in team-work, being a leader, managing their career and their life.

COURSE OUTCOMES (COs)

CO1	Optimize business decisions and create competitive advantage with Big data analytics
CO2	Practice java concepts required for developing map reduce programs.
CO3	Impart the architectural concepts of Hadoop and introducing map reduce paradigm.
CO4	Practice programming tools PIG and HIVE in Hadoop eco system.
CO5	Implement best practices for Hadoop development.

INDEX

S. No.	LIST OF EXPERIMENT	DATE	SIGN
1.	Implement the following Data structures in Java i) Linked Lists ii) Stacks iii) Queues iv) Set v) Map .	**7-3-22**	
2.	Perform setting up and Installing Hadoop in its three operating modes: Standalone, Pseudodistributed, Fully distributed.	**14-3-22**	
3.	Implement the following file management tasks in Hadoop: Adding files and directories • Retrieving files • Deleting files Hint: A typical Hadoop workflow creates data files (such• as log files) elsewhere and copies them into HDFS using one of the above command line utilities.	**21-3-22**	
4.	Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.	**28-3-22**	
5.	Write a Map Reduce program that mines weather data. Weather sensors collecting data everyhour at many locations across the globe gather a large volume of log data, which is a goodcandidate for analysis with MapReduce, since it is semi structured and record-oriented	**4-4-22**	
6.	Implement Matrix Multiplication with Hadoop Map Reduce	**11-4-22**	
7.	Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.	**18-4-22** **25-4-22**	
8.	Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes..	**2-5-22**	
9.	Solve some real life big data problems	**9-5-22**	

-

EXPERIMENT -1

INSTALL

VMWARE

OBJECTIVE:

To Install VMWare.

RESOURCES:

VMWare stack, 4 GB RAM, Web browser,Hard Disk 80 GB.

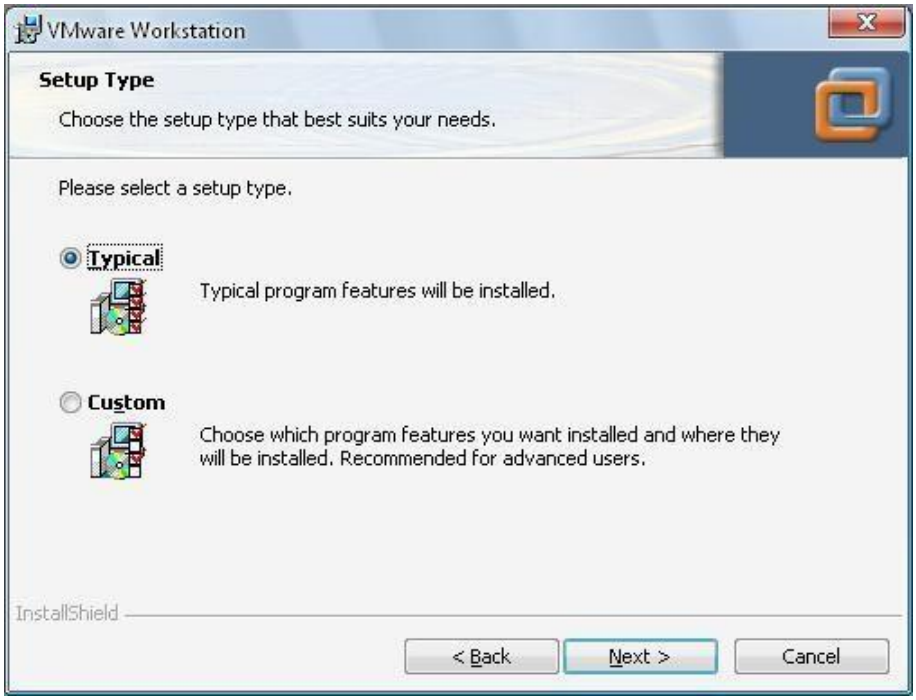
PROGRAM LOGIC:

STEP 1. First of all, enter to the official site of VMware and download VMware Workstationhttps://www.vmware.com/tryvmware/?p=workstation-w

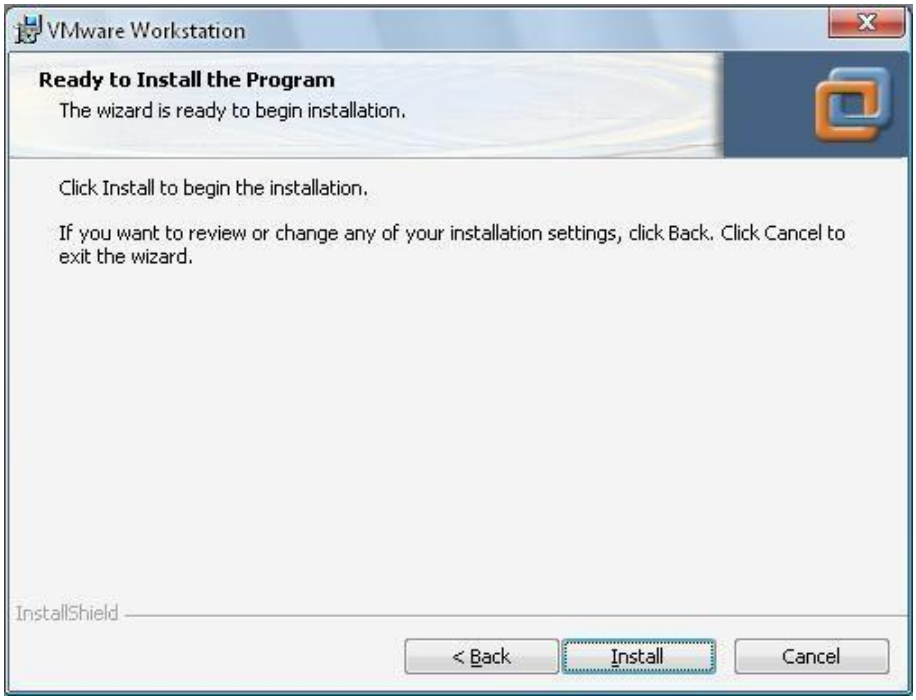
STEP 2. After downloading VMware workstation, install it on your PC

STEP 3. Setup will open Welcome Screen

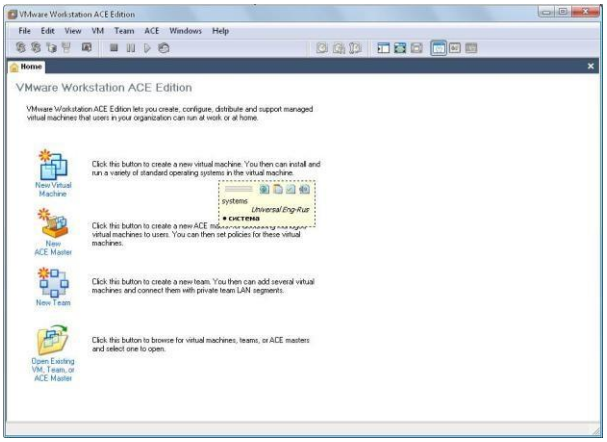
Click on **Next** button and choose **Typical** option



STEP 4. By clicking "**Next**" buttons, to begin the installation, click on **Install** button at the end

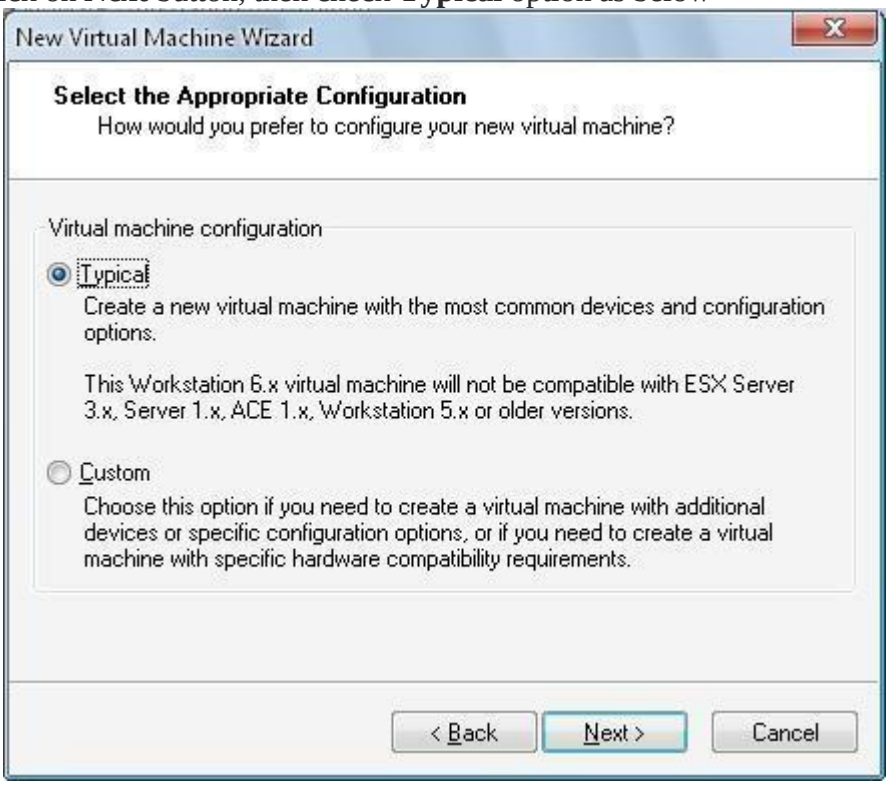


STEP 5. This will install VMware Workstation software on your PC, After installation complete, click on **Finish** button. Then restart your PC. Then open this software



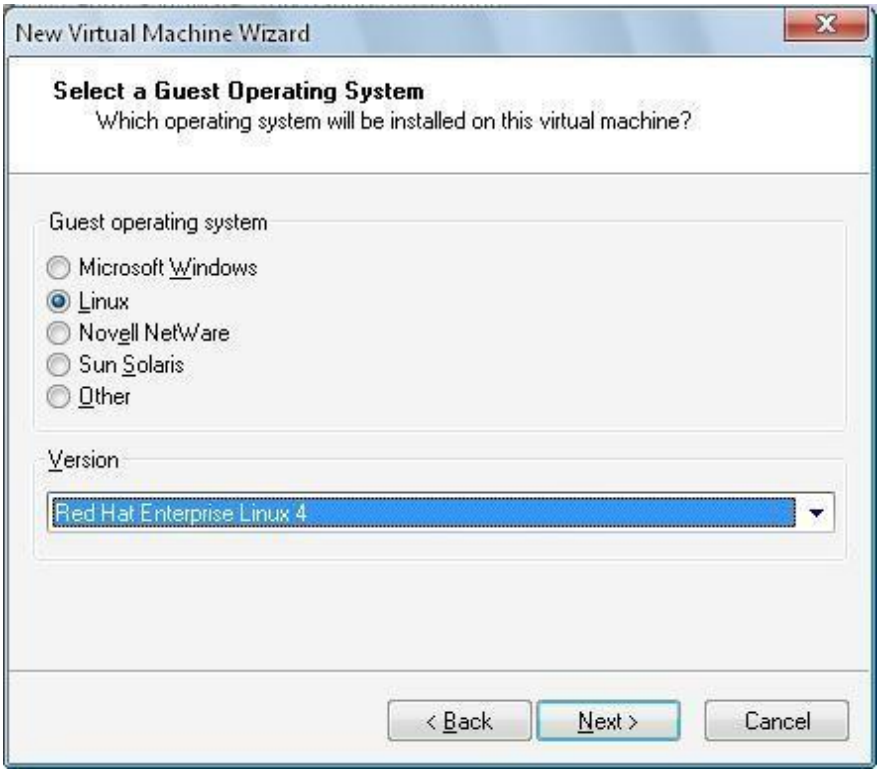
6. In this step we try to create new “virtual machine”. Enter to File menu, then New-> Virtual Machine

Click on **Next** button, then check **Typical** option as below

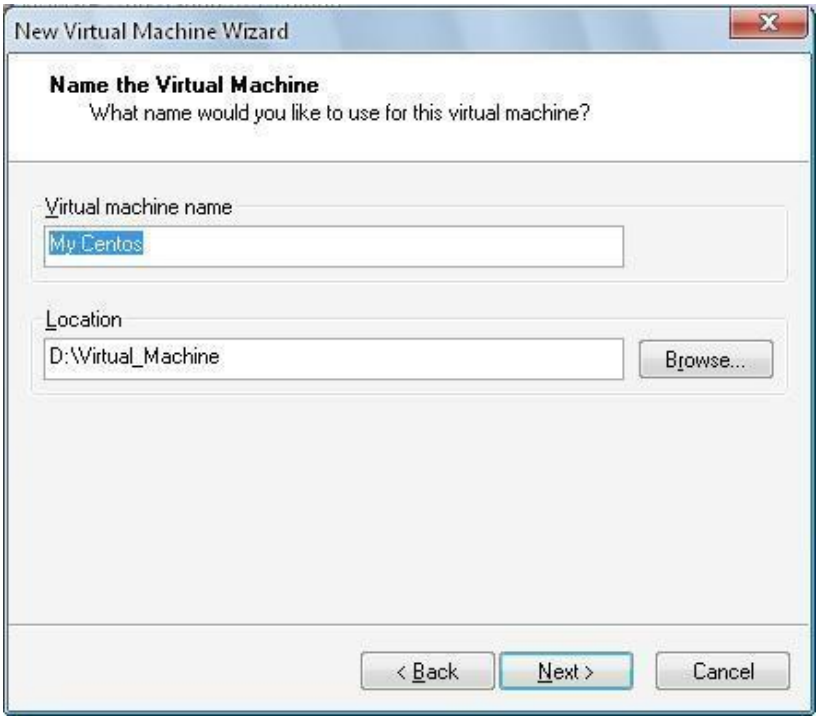


Then click **Next** button, and check your OS version. In this example, as we‟re going to setup Oracle server on CentOS, we‟ll check **Linux** option and from “*version*” option we‟ll check **Red Hat**

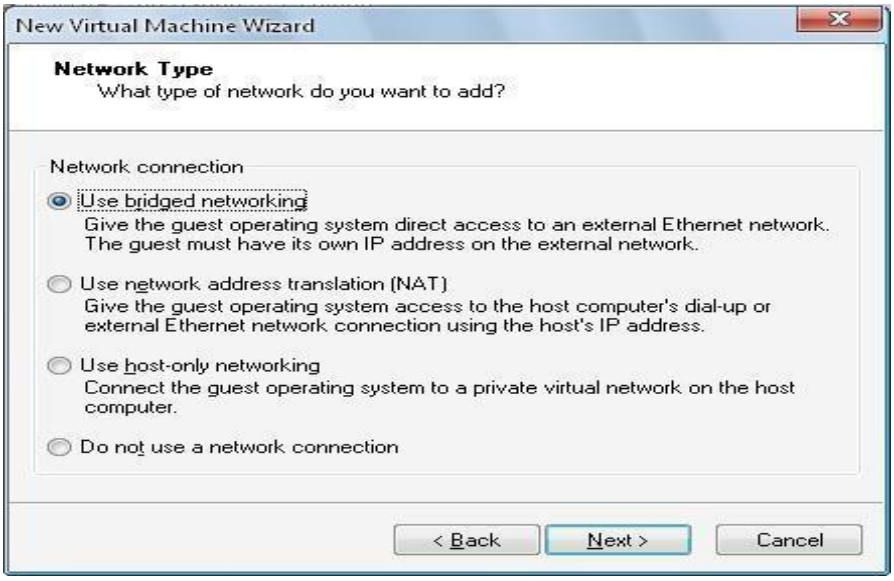
Enterprise Linux 4



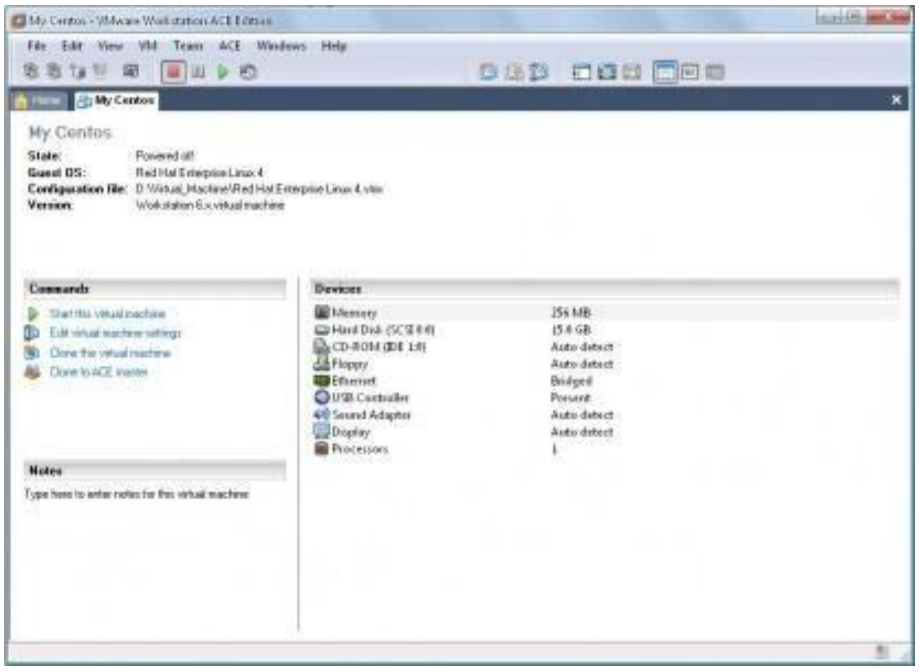
By clicking **Next** button, we‟ll give a name to our virtual machine, and give directory to create this new virtual machine



Then select **Use bridged networking** option and click **Next**.

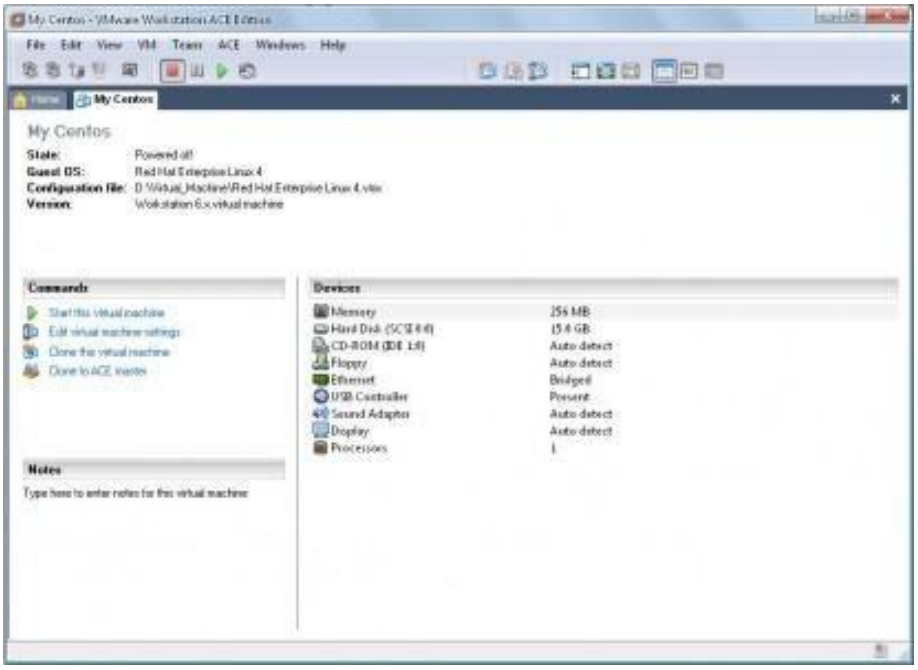


Then you‟ve to define size of hard disk by entering its size. I‟ll give 15 GB hard disk space and please check **Allocate all disk space now** option



Here, you can delete **Sound Adapter, Floppy and USB Controller** by entering “Edit virtual machine settings”**.** If you‟re going to setup Oracle Server, please make sure you‟ve increased your Memory (RAM) to 1GB.

INPUT/OUTPUT



PRE LAB VIVA QUESTIONS:

1. What is VMWare stack?
2. List out various data formats?
3. List out the characteristics of big data?

LAB ASSIGNMENT:

1. Install Pig?
2. Install Hive?

POST LAB VIVA QUESTIONS:

1. List out various terminologies in Big Data environments?
2. Define big data analytics?

EXPERIMENT -2

HADOOP MODES

OBJECTIVE:

1) Perform setting up and Installing Hadoop in its three operating modes.
   Standalone.
   Pseudo distributed
   Fully distributed.

2) Use web based tools to monitor your Hadoop setup.

RESOURCES:

VMWare stack, 4 GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

a) STANDALONE MODE:

- Installation of jdk 7

Command: sudo apt-get install openjdk-7-jdk

- Download and extract Hadoop

Command: wget http://archive.apache.org/dist/hadoop/core/hadoop-1.2.0/hadoop-1.2.0.tar.gz
Command: tar -xvf hadoop-1.2.0.tar.gz
Command: sudo mv hadoop-1.2.0 /usr/lib/hadoop

- Set the path for java and hadoop

Command: sudo gedit $HOME/.bashrc

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
export PATH=$PATH:$JAVA_HOME/bin
export        HADOOP_COMMON_HOME=/usr/lib/hadoop
export         HADOOP_MAPRED_HOME=/usr/lib/hadoop
export PATH=$PATH:$HADOOP_COMMON_HOME/bin
export PATH=$PATH:$HADOOP_COMMON_HOME/Sbin
```

- Checking of java and hadoop

Command: java -version
Command: hadoop version

b) PSEUDO MODE:

Hadoop single node cluster runs on single machine. The namenodes and datanodes are performing on the one machine. The installation and configuration steps as given below:

- Installation of secured shell

Command: sudo apt-get install openssh-server

- Create a ssh key for passwordless ssh configuration

Command: ssh-keygen -t rsa –P ""
- Moving the key to authorized key

Command: cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
/**********RESTART THE COMPUTER**********/
- Checking of secured shell login

Command: ssh localhost
- Add JAVA_HOME directory in hadoop-env.sh file

Command: sudo gedit /usr/lib/hadoop/conf/hadoop-env.sh
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
- Creating namenode and datanode directories for hadoop

Command: sudo mkdir -p /usr/lib/hadoop/dfs/namenode
Command: sudo mkdir -p /usr/lib/hadoop/dfs/datanode
- Configure core-site.xml

Command: sudo gedit /usr/lib/hadoop/conf/core-site.xml
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:8020</value>
</property>
- Configure hdfs-site.xml

Command: sudo gedit /usr/lib/hadoop/conf/hdfs-site.xml
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permissions</name>
<value>false</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>/usr/lib/hadoop/dfs/namenode</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/usr/lib/hadoop/dfs/datanode</value>
</property>
- Configure mapred-site.xml

Command: sudo gedit /usr/lib/hadoop/conf/mapred-site.xml
<property>
<name>mapred.job.tracker</name>

<value>localhost:8021</value>
</property>

- Format the name node

Command: hadoop namenode -format

- Start the namenode, datanode

Command: start-dfs.sh

- Start the task tracker and job tracker

Command: start-mapred.sh

- To check if Hadoop started correctly

Command: jps
namenode
secondarynamenode
datanode
jobtracker
tasktracker

c) FULLY DISTRIBUTED MODE:

All the demons like namenodes and datanodes are runs on different machines. The data will replicate according to the replication factor in client machines. The secondary namenode will store the mirror images of namenode periodically. The namenode having the metadata where the blocks are stored and number of replicas in the client machines. The slaves and master communicate each other periodically. The configurations of multinode cluster are given below:

- Configure the hosts in all nodes/machines

Command: sudo gedit /etc/hosts/
192.168.1.58 pcetcse1
pcetcse2
pcetcse3
pcetcse4
pcetcse5

- Passwordless Ssh Configuration

Create ssh key on namenode/master.

Command: ssh-keygen -t rsa -p “”
Copy the generated public key all datanodes/slaves.

Command: ssh-copy-id -i ~/.ssh/id_rsa.pub huser@pcetcse2
Command: ssh-copy-id -i ~/.ssh/id_rsa.pub huser@pcetcse3
Command: ssh-copy-id -i ~/.ssh/id_rsa.pub huser@pcetcse4
Command: ssh-copy-id -i ~/.ssh/id_rsa.pub huser@pcetcse5
/**********RESTART ALL NODES/COMPUTERS/MACHINES **********/

NOTE: Verify the passwordless ssh environment from namenode to all datanodes as "huser" user.

- Login to master node

Command: ssh pcetcse1
Command: ssh pcetcse2
Command: ssh pcetcse3
Command: ssh pcetcse4
Command: ssh pcetcse5

- Add JAVA_HOME directory in hadoop-env.sh file in all nodes/machines

Command: sudo gedit /usr/lib/hadoop/conf/hadoop-env.sh
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386

- Creating namenode directory in namenode/master

Command: sudo mkdir -p /usr/lib/hadoop/dfs/namenode

- Creating namenode directory in datanonodes/slaves

Command: sudo mkdir -p /usr/lib/hadoop/dfs/datanode Close HTML tag.

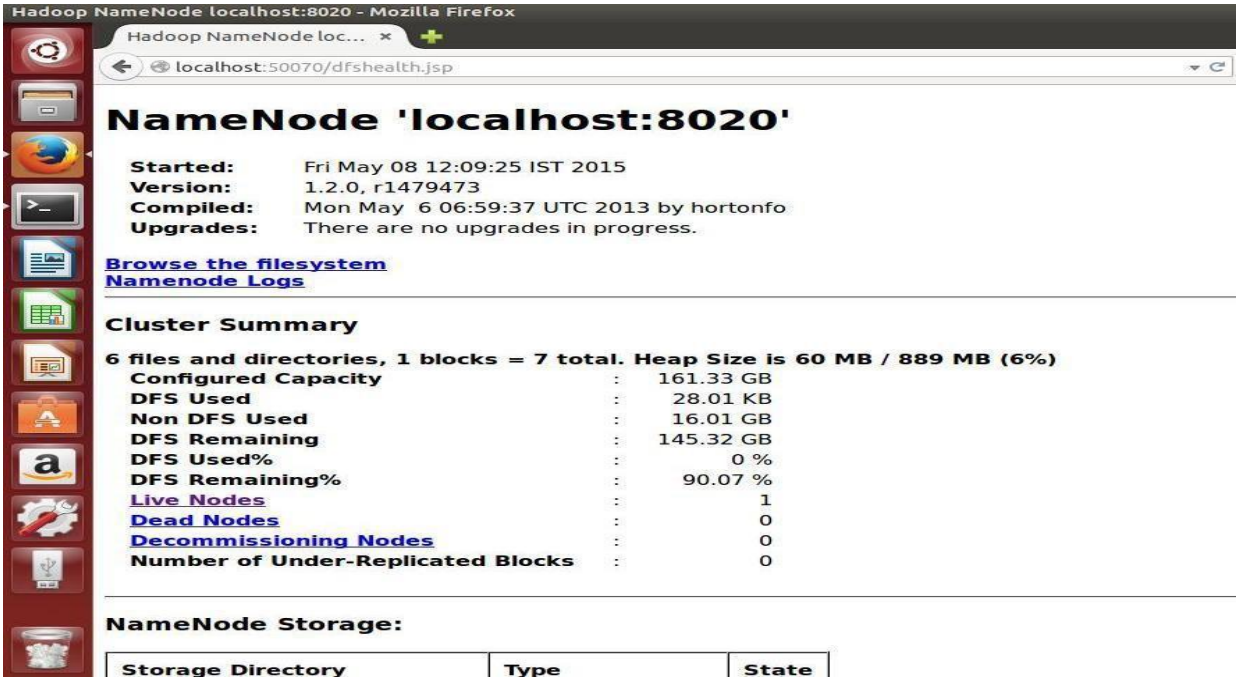
Use web based tools to monitor your Hadoop setup.

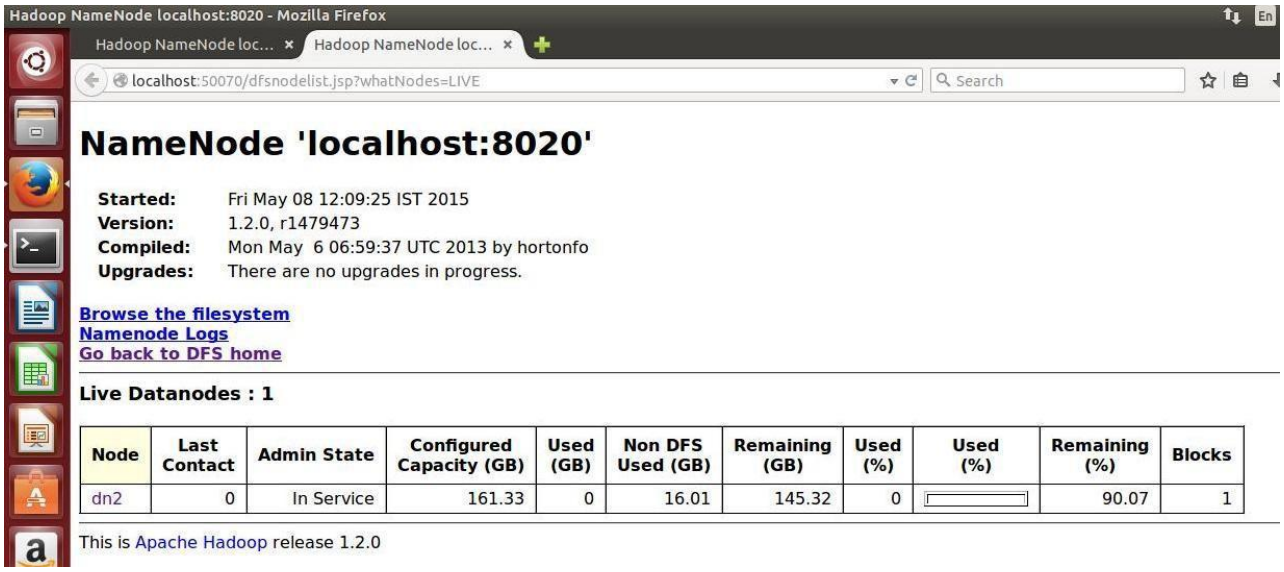
HDFS Namenode on UI
http://locahost:50070/

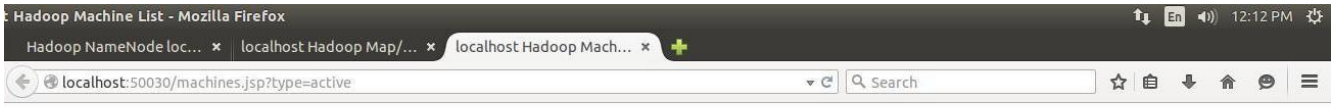
INPUT/OUTPUT:

ubuntu @localhost> jps

Data node, name nodem
Secondary name node, NodeManager, Resource Manager


Hadoop NameNode localhost:8020 - Mozilla Firefox
Hadoop NameNode loc...
localhost:50070/dfshealth.jsp
NameNode 'localhost:8020'
Started: Fri May 08 12:09:25 IST 2015
Version: 1.2.0, r1479473
Compiled: Mon May 6 06:59:37 UTC 2013 by hortonfo
Upgrades: There are no upgrades in progress.
Browse the filesystem
Namenode Logs
Cluster Summary
6 files and directories, 1 blocks = 7 total. Heap Size is 60 MB / 889 MB (6%)
Configured Capacity : 161.33 GB
DFS Used : 28.01 KB
Non DFS Used : 16.01 GB
DFS Remaining : 145.32 GB
DFS Used% : 0 %
DFS Remaining% : 90.07 %
Live Nodes : 1
Dead Nodes : 0
Decommissioning Nodes : 0
Number of Under-Replicated Blocks : 0
NameNode Storage:
Storage Directory
Type
State


Hadoop NameNode localhost:8020 - Mozilla Firefox
Hadoop NameNode loc...
Hadoop NameNode loc...
localhost:50070/dfsnodelist.jsp?whatNodes=LIVE
Search
NameNode 'localhost:8020'
Started: Fri May 08 12:09:25 IST 2015
Version: 1.2.0, r1479473
Compiled: Mon May 6 06:59:37 UTC 2013 by hortonfo
Upgrades: There are no upgrades in progress.
Browse the filesystem
Namenode Logs
Go back to DFS home
Live Datanodes : 1
Node
Last Contact
Admin State
Configured Capacity (GB)
Used (GB)
Non DFS Used (GB)
Remaining (GB)
Used (%)
Used (%)
Remaining (%)
Blocks
dn2
0
In Service
161.33
0
16.01
145.32
0
90.07
1
This is Apache Hadoop release 1.2.0



localhost Hadoop Machine List

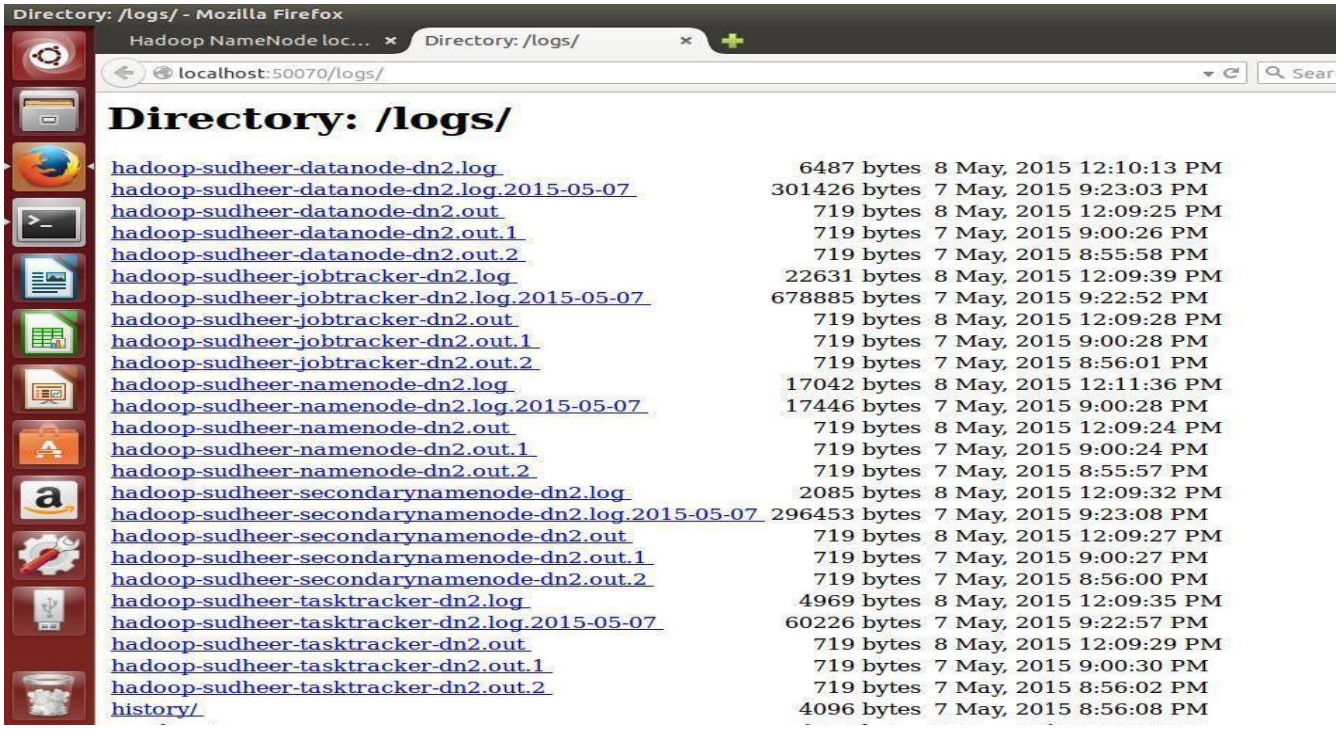
Active Task Trackers

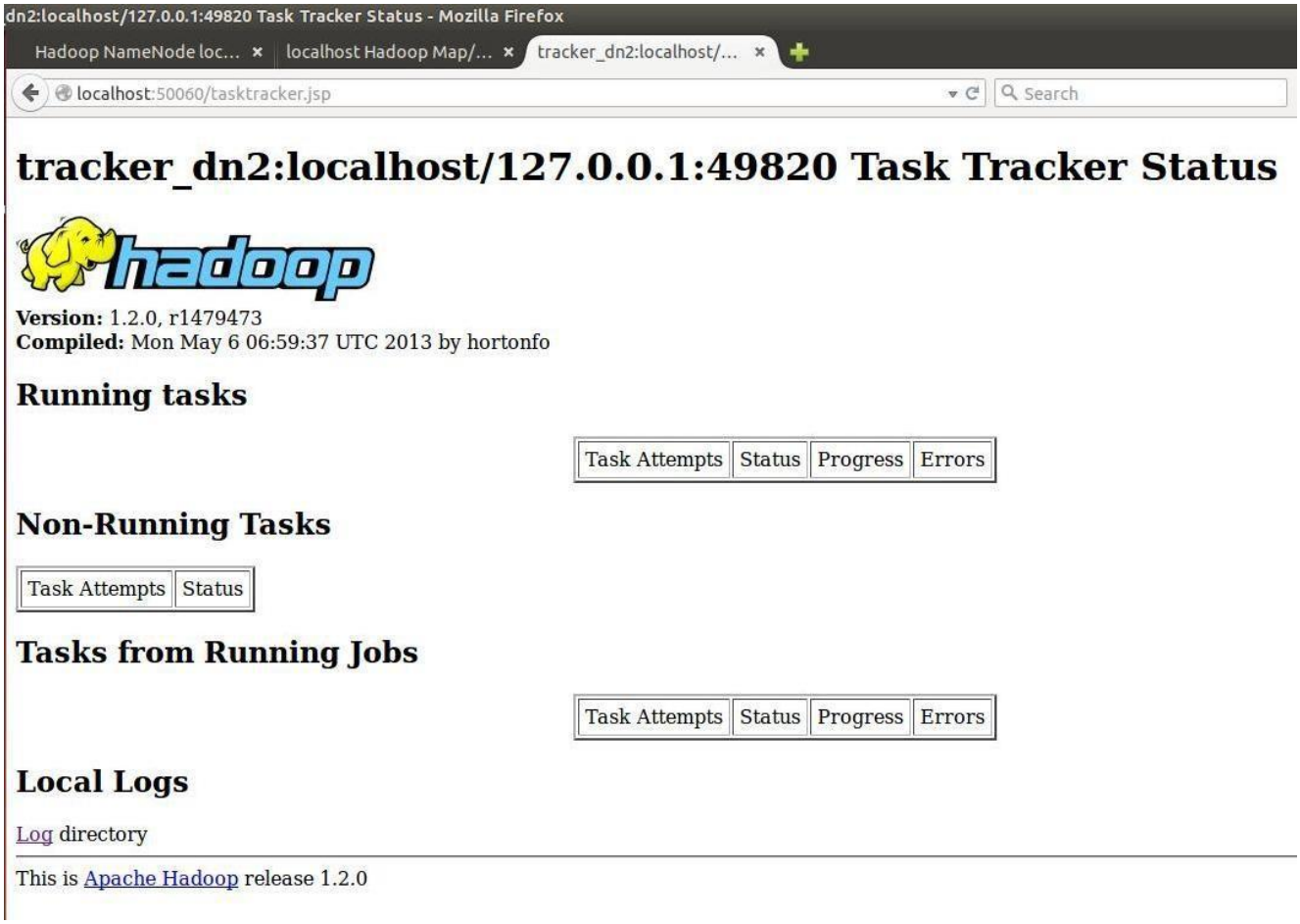
Task Trackers												
Name	Host	# running tasks	Max Map Tasks	Max Reduce Tasks	Task Failures	Directory Failures	Node Health Status	Seconds Since Node Last Healthy	Total Tasks Since Start	Succeeded Tasks Since Start	Total Tasks Last Day	Succeeded Tasks Last Day
tracker_dn2:localhost/127.0.0.1:49820	dn2	0	2	2	0	0	N/A	0	0	0	0	0

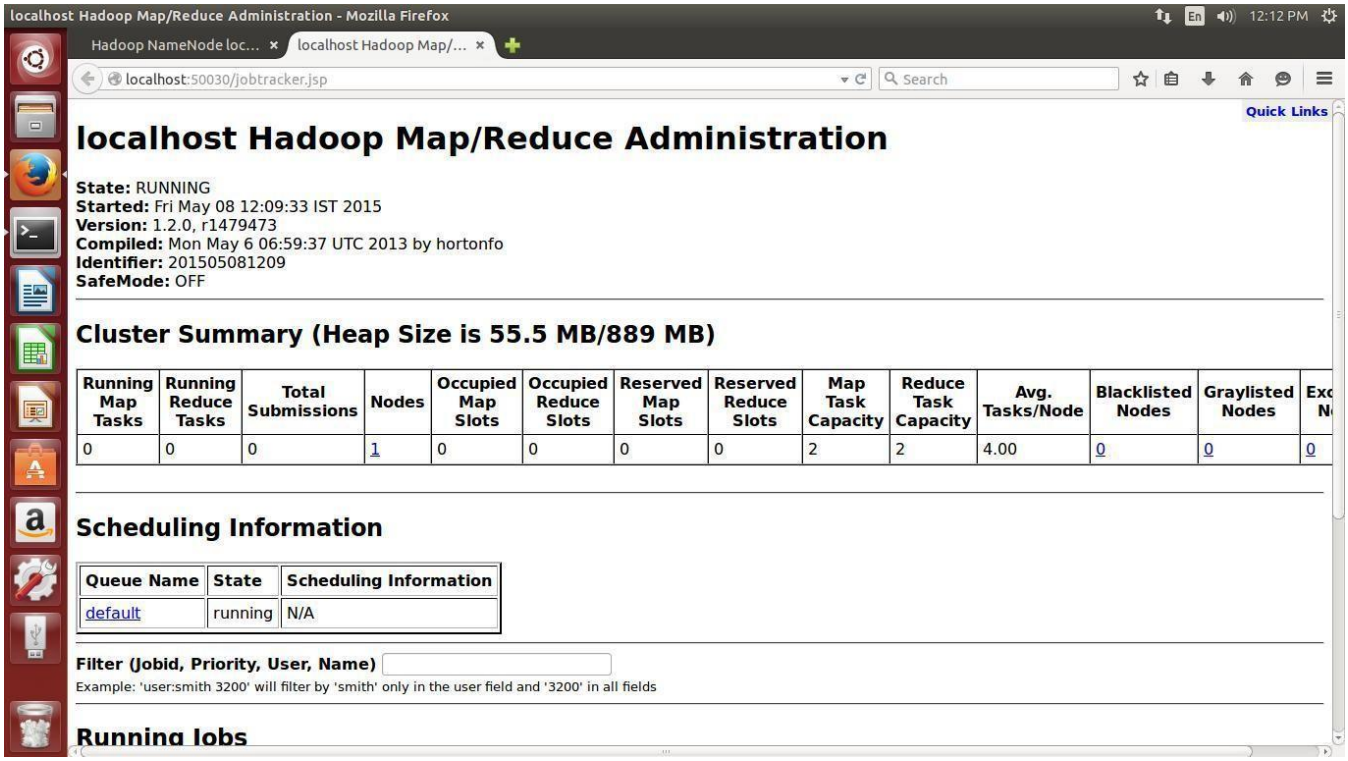
This is Apache Hadoop release 1.2.0

HDFS Jobtracker
http://locahost:50030/

HDFS Logs
http://locahost:50070/logs/




dn2:localhost/127.0.0.1:49820 Task Tracker Status - Mozilla Firefox
Hadoop NameNode loc...
localhost Hadoop Map/...
tracker_dn2:localhost/...
localhost:50060/tasktracker.jsp
Search
tracker_dn2:localhost/127.0.0.1:49820 Task Tracker Status
hadoop
Version: 1.2.0, r1479473
Compiled: Mon May 6 06:59:37 UTC 2013 by hortonfo
Running tasks
Task Attempts
Status
Progress
Errors
Non-Running Tasks
Task Attempts
Status
Tasks from Running Jobs
Task Attempts
Status
Progress
Errors
Local Logs
Log directory
This is Apache Hadoop release 1.2.0



PRE LAB VIVA QUESTIONS:

1. What does _jps,, command do?
2. How to restart Namenode?
3. Differentiate between Structured and Unstructured data?

LAB ASSIGNMENT:

1 How to configure the daemons in the browser.

POST LAB VIVA QUESTIONS:

1. What are the main components of a Hadoop Application?
2. Explain the difference between NameNode, Backup Node and Checkpoint NameNode.

EXPERIMENT -3

USING LINUX OPERATING SYSTEM

OBJECTIVE:

1. Implementing the basic commands of LINUX Operating System – File/Directory creation, deletion, update operations.

RESOURCES:

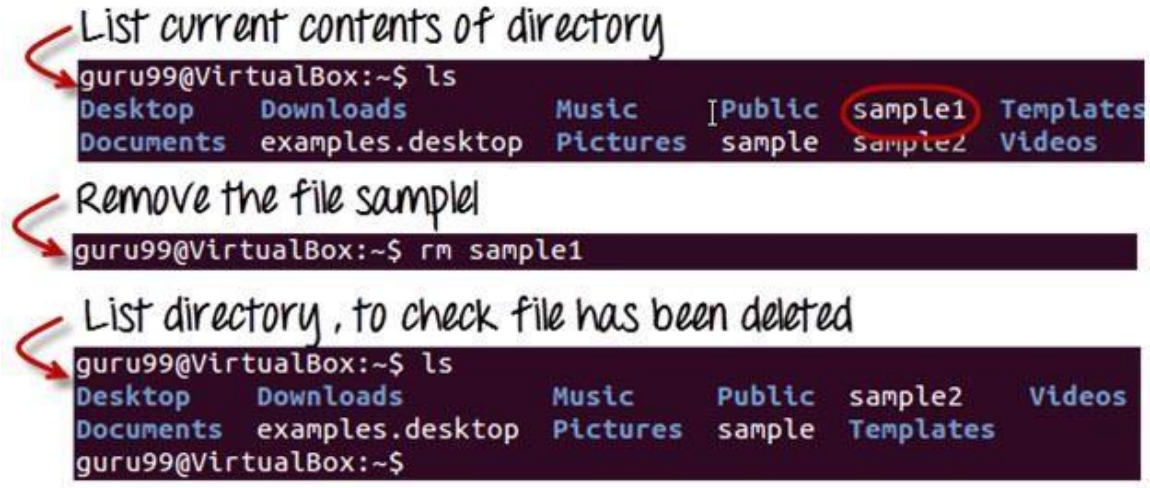
VMWare stack, 4 GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

1. cat > filename
2. Add content
3. Press 'ctrl + d' to return to command prompt.

To remove a file use syntax - rm filename

1.4 INPUT/OUTPUT:



PRE-LAB VIVA QUESTIONS:

1. What is ls command?
2. What are the attributes of ls command?

LAB ASSIGNMENT:

1 Write a linux commands for Sed operations?
2 Write the linux commands for renaming a file?

POST-LAB VIVA QUESTIONS:

1. What is the purpose of rm command?
2. What is the difference between Linux and windows commands?

FILE MANAGEMENT IN HADOOP

OBJECTIVE:

Implement the following file management tasks in Hadoop:

i. Adding files and directories
ii. Retrieving files
iii. Deleting files

Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.

RESOURCES:

VMWare stack, 4 GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

Adding Files and Directories to HDFS

Before you can run Hadoop programs on data stored in HDFS, you„ll need to put the data into HDFS first. Let„s create a directory and put a file in it. HDFS has a default working directory of /user/$USER, where $USER is your login user name. This directory isn„t automatically created for you, though, so let„s create it with the mkdir command. For the purpose of illustration, we use chuck. You should substitute your user name in the example commands.

hadoop fs -mkdir /user/chuck

hadoop fs -put

hadoop fs -put example.txt /user/chuck

Retrieving Files from HDFS

The Hadoop command get copies files from HDFS back to the local filesystem. To retrieve example.txt, we can run the following command:

hadoop fs -cat

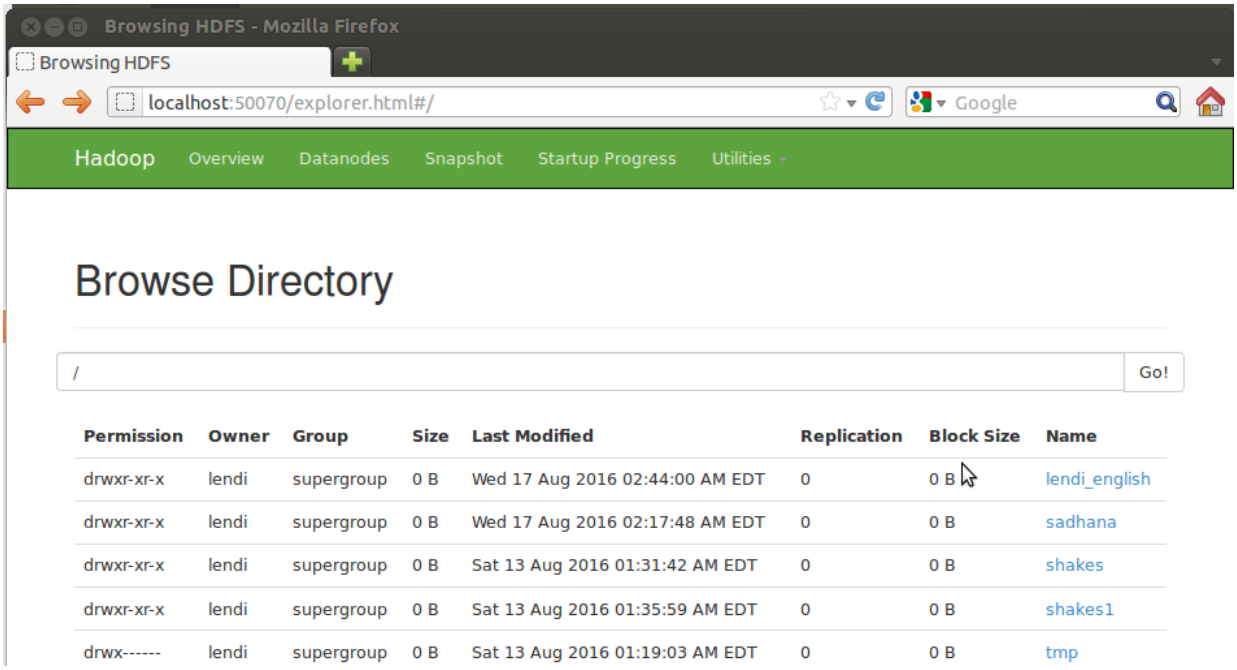
example.txt Deleting

files from HDFS

hadoop fs -rm example.txt

- Command for creating a directory in hdfs is “hdfs dfs –mkdir /lendicse”.
- Adding directory is done through the command “hdfs dfs –put lendi_english /”

INPUT/OUTPUT:



PRE LAB VIVA QUESTIONS:

1) Define Hadoop?

2) List out the various use cases of Hadoop?

LAB ASSIGNMENT

1) What is command used to list out directories of Data Node through web tool

POST LAB VIVA QUESTIONS:

1. Distinguish the Hadoop Ecosystem?
2. Demonstrate divide and conquer philosophy in Hadoop Cluster?

EXPERIMENT-4

MAPREDUCE PROGRAM 1

OBJECTIVE:

Run a basic word count Map Reduce program to understand Map Reduce Paradigm.

RESOURCES:

VMWare stack, 4 GB RAM, Web browser, Hard Disk 80 GB.

PROGRAM LOGIC:

WordCount is a simple program which counts the number of occurrences of each word in a given text input data set. WordCount fits very well with the MapReduce programming model making it a great example to understand the Hadoop Map/Reduce programming style. Our implementation consists of three main parts:

1. Mapper
2. Reducer
3. Driver

Step-1. Write a Mapper

A Mapper overrides the ‒map‖ function from the Class "org.apache.hadoop.mapreduce.Mapper" which provides <key, value> pairs as the input. A Mapper implementation may output <key,value> pairs using the provided Context .

Input value of the WordCount Map task will be a line of text from the input data file and the key would be the line number <line_number, line_of_text> . Map task outputs <word, one> for each word in the line of text.

Pseudo-code

```
void Map (key, value){
        for each word x in value:
            output.collect(x,1);
}
```

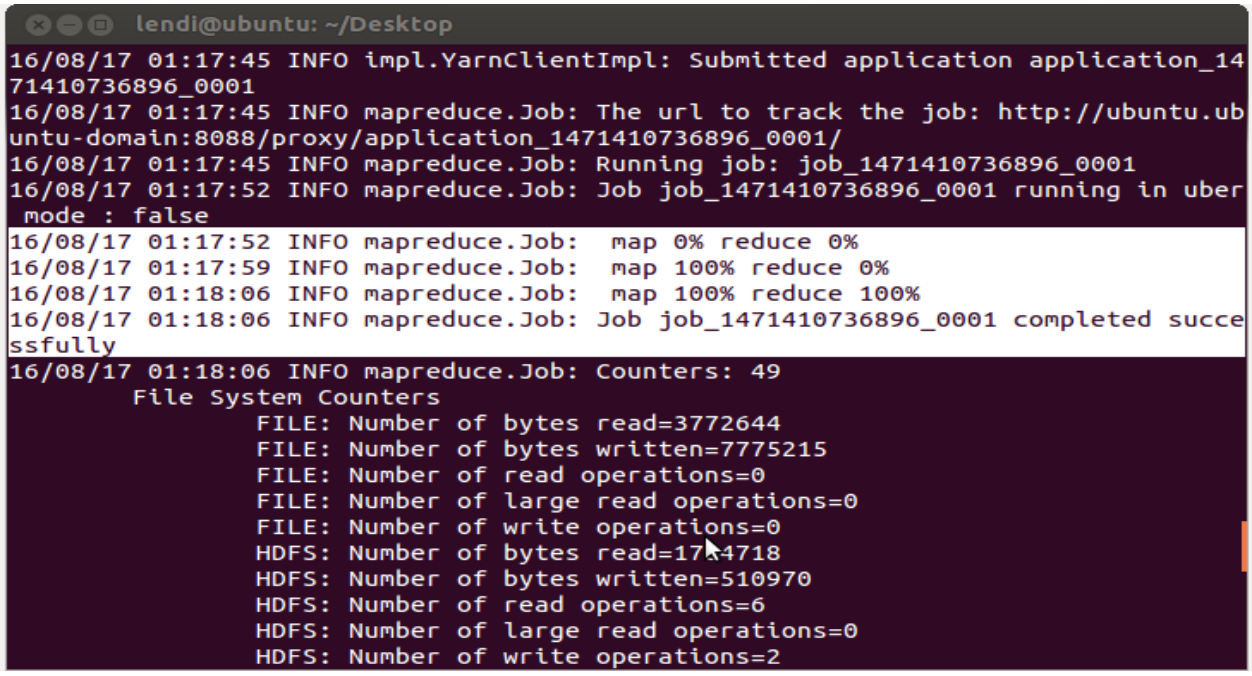
Step-2. Write a Reducer

A Reducer collects the intermediate <key,value> output from multiple map tasks and assemble a single result. Here, the WordCount program will sum up the occurrence of each word to pairs as <word, occurrence>.

Pseudo-code

```
void Reduce (keyword, <list of value>){ for
    each x in <list of value>:
      sum+=x;
   final_output.collect(keyword, sum);
}
```

INPUT/OUTPUT:



PRE-LAB VIVA QUESTIONS:

1. Justify how hadoop technology satisfies the business insights now -a –days?
2. Define Filesystem?

LAB ASSIGNMENT:

Run a basic word count Map Reduce program to understand Map Reduce Paradigm.

POST-LAB VIVA QUESTIONS:

1. Define what is block in HDFS?
2. Why is a block in HDFS so large?

MAPREDUCE PROGRAM 2

OBJECTIVE:

Write a Map Reduce program that mines weather data. Hint: Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with Map Reduce, since it is semi structured and record-oriented.

RESOURCES:

VMWare, Web browser, 4 GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

WordCount is a simple program which counts the number of occurrences of each word in a given text input data set. WordCount fits very well with the MapReduce programming model making it a great example to understand the Hadoop Map/Reduce programming style. Our implementation consists of three main parts:

1. Mapper
2. Reducer
3. Main program

Step-1. Write a Mapper

A Mapper overrides the ―map function from the Class "org.apache.hadoop.mapreduce.Mapper" which provides <key, value> pairs as the input. A Mapper implementation may output <key,value> pairs using the provided Context .
Input value of the WordCount Map task will be a line of text from the input data file and the key would be the line number <line_number, line_of_text> . Map task outputs <word, one> for each word in the line of text.

Pseudo-code

```
void Map (key, value){
for each max_temp x in value:
output.collect(x, 1);
}
void Map (key, value){
    for each min_temp x in value:

output.collect(x, 1);
}
```

Step-2 Write a Reducer

A Reducer collects the intermediate <key,value> output from multiple map tasks and assemble a single result. Here, the WordCount program will sum up the occurrence of each word to pairs as <word, occurrence>.

Pseudo-code

```
void Reduce (max_temp, <list of value>){
for each x in <list of value>:
sum+=x;
final_output.collect(max_temp, sum);
}
void Reduce (min_temp, <list of value>){
for each x in <list of value>:
sum+=x;
final_output.collect(min_temp, sum);
}
```

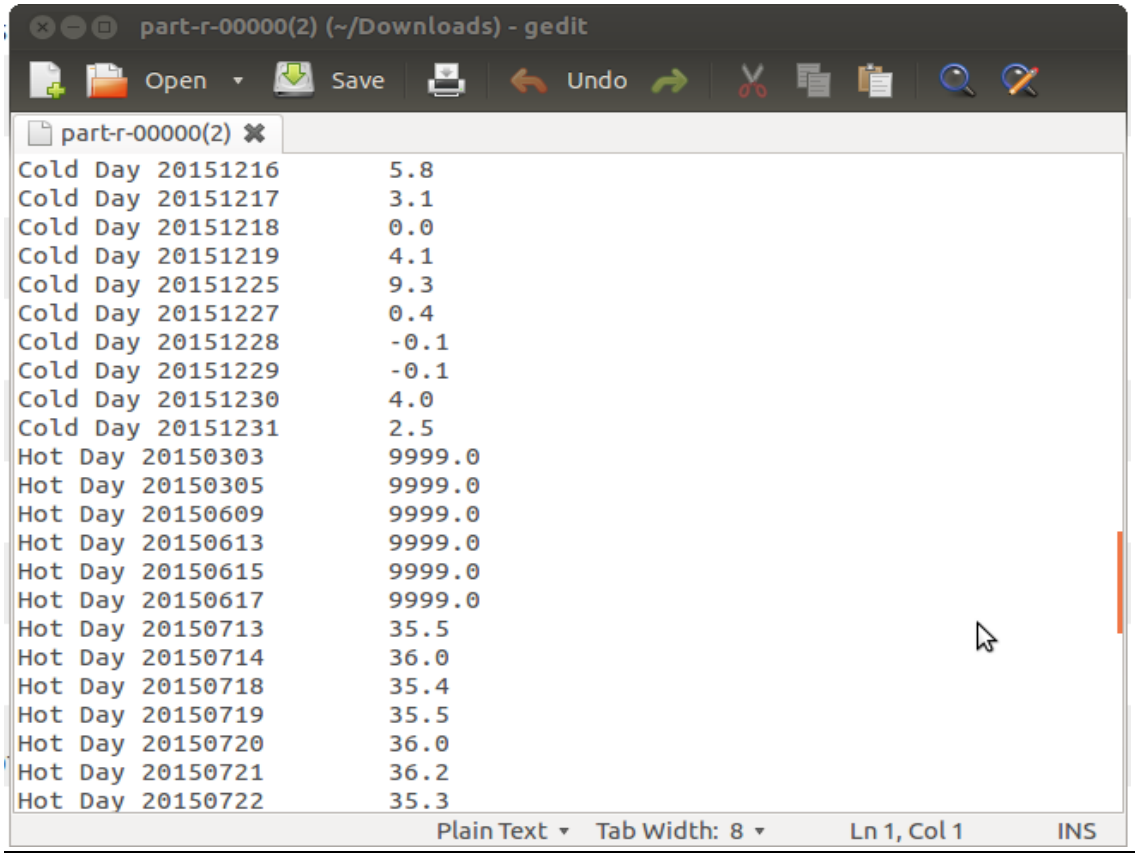
3. Write Driver

The Driver program configures and run the MapReduce job. We use the main program to perform basic configurations such as:

Job Name : name of this Job Executable (Jar)
Class: the main executable class. For here, WordCount.
Mapper Class: class which overrides the "map" function. For here, Map.
Reducer: class which override the "reduce" function. For here , Reduce.
Output Key: type of output key. For here, Text.Output
Value: type of output value. For here, IntWritable.
File Input Path
File Output Path

INPUT/OUTPUT:

Set of Weather Data over the years



PRE-LAB VIVA QUESTIONS:

1) Explain the function of MapReducer partitioner?
2) What is the difference between an Input Split and HDFS Block?
3) What is Sequencefileinputformat?

LAB ASSIGNMENT:

1. Using Map Reduce job to Identify language by merging multi language dictionary files into a single dictionary file.
2. Join multiple datasets using a MapReduce Job.

POST-LAB VIVA QUESTIONS:

1) In Hadoop what is InputSplit?
2) Explain what is a sequence file in Hadoop?

EXPERIMENT-6

OBJECTIVE :

Implement matrix multiplication with Hadoop Map Reduce.

RESOURCES:

VMWare, Web browser, 4 GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

We assume that the input files for A and B are streams of (key,value) pairs in sparse matrix format, where each key is a pair of indices (i,j) and each value is the corresponding matrix element value. The output files for matrix C=A*B are in the same format.

We have the following input parameters:

The path of the input file or directory for matrix A.
The path of the input file or directory for matrix B.

The path of the directory for the output files for matrix C.
strategy = 1, 2, 3 or 4.
R = the number of reducers.
I = the number of rows in A and C.
K = the number of columns in A and rows in B.
J = the number of columns in B and C.
IB = the number of rows per A block and C block.
KB = the number of columns per A block and rows per B block.
JB = the number of columns per B block and C block.

In the pseudo-code for the individual strategies below, we have intentionally avoided factoring common code for the purposes of clarity. Note that in all the strategies the memory footprint of both the mappers and the reducers is flat at scale.

Note that the strategies all work reasonably well with both dense and sparse matrices. For sparse matrices we do not emit zero elements. That said, the simple pseudo-code for multiplying the individual blocks shown here is certainly not optimal for sparse matrices. As a learning exercise, our focus here is on mastering the MapReduce complexities, not on optimizing the sequential matrix multipliation algorithm for the individual blocks.

Steps

1. setup ()
2. var NIB = (I-1)/IB+1
3. var NKB = (K-1)/KB+1
4. var NJB = (J-1)/JB+1
5. map (key, value)

6. if from matrix A with key=(i,k) and value=a(i,k)
7. for 0 <= jb < NJB
8. emit (i/IB, k/KB, jb, 0), (i mod IB, k mod KB, a(i,k))
9. if from matrix B with key=(k,j) and value=b(k,j)
10. for 0 <= ib < NIB
emit (ib, k/KB, j/JB, 1), (k mod KB, j mod JB, b(k,j))
Intermediate keys (ib, kb, jb, m) sort in increasing order first by ib, then by kb, then by jb, then by m. Note that m = 0 for A data and m = 1 for B data.

The partitioner maps intermediate key (ib, kb, jb, m) to a reducer r as follows:

11. r = ((ib*JB + jb)*KB + kb) mod R
12. These definitions for the sorting order and partitioner guarantee that each reducer R[ib,kb,jb] receives the data it needs for blocks A[ib,kb] and B[kb,jb], with the data for the A block immediately preceding the data for the B block.
13. var A = new matrix of dimension IBxKB
14. var B = new matrix of dimension KBxJB
15. var sib = -1
16. var skb = -1

Reduce (key, valueList)

17. if key is (ib, kb, jb, 0)
18. // Save the A block.
19. sib = ib
20. skb = kb
21. Zero matrix A
22. for each value = (i, k, v) in valueList A(i,k) = v
23. if key is (ib, kb, jb, 1)
24. if ib != sib or kb != skb return // A[ib,kb] must be zero!
25. // Build the B block.
26. Zero matrix B
27. for each value = (k, j, v) in valueList B(k,j) = v
28. // Multiply the blocks and emit the result.
29. ibase = ib*IB
30. jbase = jb*JB
31. for 0 <= i < row dimension of A
32. for 0 <= j < column dimension of B
33. sum = 0
34. for 0 <= k < column dimension of A = row dimension of B
a. sum += A(i,k)*B(k,j)
35. if sum != 0 emit (ibase+i, jbase+j), sum

Set of Data sets over different Clusters are taken as Rows and Columns

INPUT/OUTPUT:

PRE-LAB VIVA QUESTIONS:

1. Explain what is "map" and what is "reducer" in Hadoop?
2. Mention what daemons run on a master node and slave nodes?
3. Mention what is the use of Context Object?

LAB ASSIGNMENT:

1. Implement matrix addition with Hadoop Map Reduce.

POST-LAB VIVA QUESTIONS:

1. What is partitioner in Hadoop?
2. Explain of RecordReader in Hadoop?

EXPERIMENT-8

PIG LATIN LANGUAGE - PIG

OBJECTIVE:

1. Installation of PIG.

RESOURCES:

VMWare, Web browser, 4 GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

STEPS FOR INSTALLING APACHE PIG

1) Extract the pig-0.15.0.tar.gz and move to home directory

2) Set the environment of PIG in bashrc file.

3) Pig can run in two modes

Local Mode and Hadoop Mode

Pig –x local and pig

4) Grunt Shell

Grunt >

5) LOADING Data into Grunt Shell

DATA = LOAD <CLASSPATH> USING PigStorage(DELIMITER) as (ATTRIBUTE :

DataType1, ATTRIBUTE : DataType2…..)

6) Describe Data

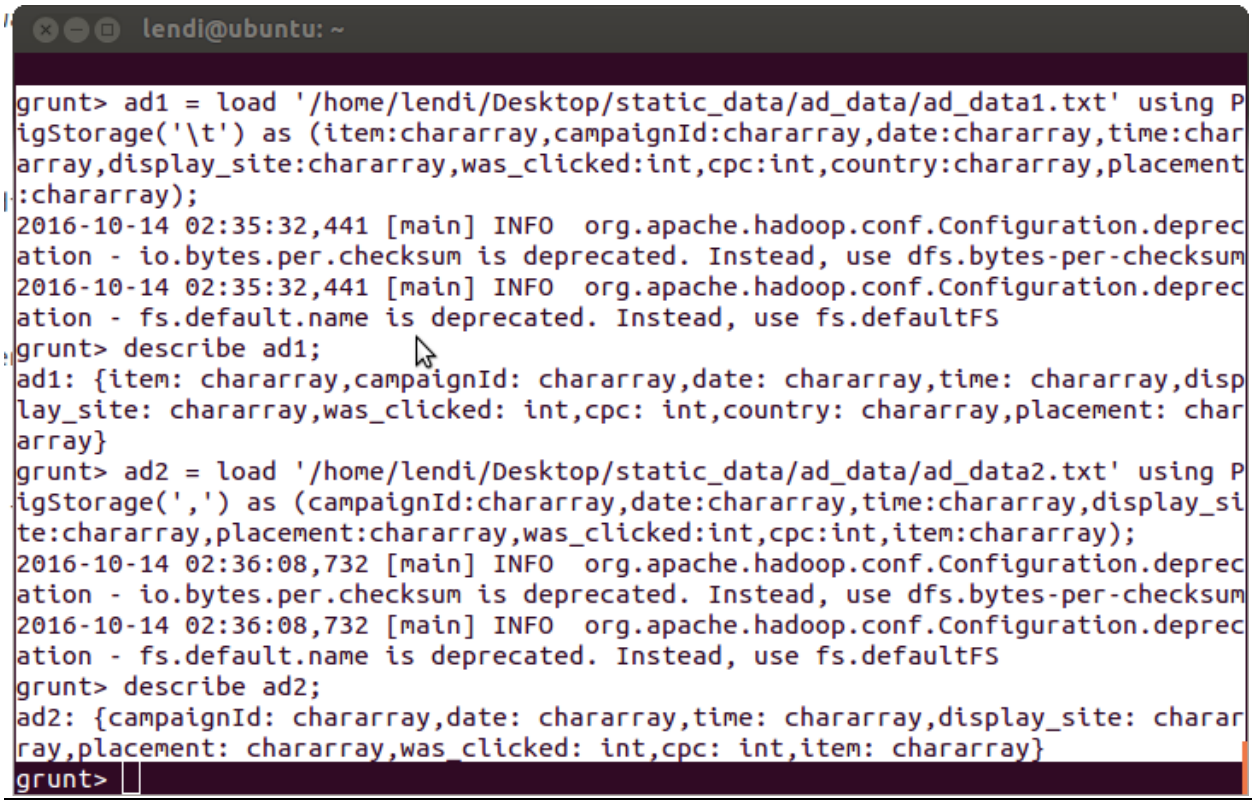
Describe DATA;

7) DUMP Data

Dump DATA;

INPUT/OUTPUT:

Input as Website Click Count Data



PRE-LAB VIVA QUESTIONS:

1) What do you mean by a bag in Pig?
2) Differentiate between PigLatin and HiveQL
3) How will you merge the contents of two or more relations and divide a single relation into two or more relations?

LAB ASSIGNMENT:

1. Process baseball data using Apache Pig.

POST-LAB VIVA QUESTIONS:

1. What is the usage of foreach operation in Pig scripts?
2. What does Flatten do in Pig

PIG COMMANDS

OBJECTIVE:

Write Pig Latin scripts sort, group, join, project, and filter your data.

RESOURCES:

VMWare, Web browser, 4 GB RAM, Hard Disk 80 GB.

PROGRAM

LOGIC: FILTER

Data

FDATA = FILTER DATA by ATTRIBUTE = VALUE;

GROUP Data

GDATA = GROUP DATA by ATTRIBUTE;

Iterating Data

FOR_DATA = FOREACH DATA GENERATE GROUP AS GROUP_FUN,

ATTRIBUTE = <VALUE>

Sorting Data

SORT_DATA = ORDER DATA BY ATTRIBUTE WITH CONDITION;

LIMIT Data

LIMIT_DATA = LIMIT DATA COUNT;

JOIN Data

JOIN DATA1 BY (ATTRIBUTE1,ATTRIBUTE2….) , DATA2 BY (ATTRIBUTE3,ATTRIBUTE….N)

INPUT / OUTPUT :

PRE-LAB VIVA QUESTIONS:

1. How will you merge the contents of two or more relations and divide a single relation into two or more relations?
2. What is the usage of foreach operation in Pig scripts?
3. What does Flatten do in Pig?

LAB ASSIGNMENT:

1. Using Apache Pig to develop User Defined Functions for student data.

PRE-LAB VIVA QUESTIONS:

1. What do you mean by a bag in Pig?
2. Differentiate between PigLatin and HiveQL

EXPERIMENT-10

OBJECTIVE: PIG LATIN MODES, PROGRAMS

a. Run the Pig Latin Scripts to find Word Count.
b. Run the Pig Latin Scripts to find a max temp for each and every year.

RESOURCES:

VMWare, Web Browser, 4 GB RAM, 80 GB Hard Disk.

PROGRAM LOGIC:

Run the Pig Latin Scripts to find Word Count.

```
lines = LOAD '/user/hadoop/HDFS_File.txt' AS (line:chararray);
words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) as word;
grouped = GROUP words BY word;
wordcount = FOREACH grouped GENERATE group, COUNT(words);
DUMP wordcount;
```

Run the Pig Latin Scripts to find a max temp for each and every year

```
-- max_temp.pig: Finds the maximum temperature by year
records = LOAD 'input/ncdc/micro-tab/sample.txt'
AS (year:chararray, temperature:int, quality:int);
filtered_records = FILTER records BY temperature != 9999 AND
(quality == 0 OR quality == 1 OR quality == 4 OR quality == 5 OR quality == 9);
grouped_records = GROUP filtered_records BY year;
max_temp = FOREACH grouped_records GENERATE group,
MAX(filtered_records.temperature);
DUMP max_temp;
```

INPUT / OUTPUT:

```
(1950,0,1)
(1950,22,1)
(1950,-11,1)
(1949,111,1)
(1949,78,1)
```

PRE-LAB VIVA QUESTIONS:

1. List out the benefits of Pig?
2. Classify Pig Latin commands in Pig?

LAB ASSIGNMENT:

1. Analyzing average stock price from the stock data using Apache Pig

POST-LAB VIVA QUESTIONS:

1. Discuss the modes of Pig scripts?
2. Explain the Pig Latin application flow?

EXPERIMENT-9

HIVE

OBJECTIVE:

Installation of HIVE.

RESOURCES:

VMWare, Web Browser, 1GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

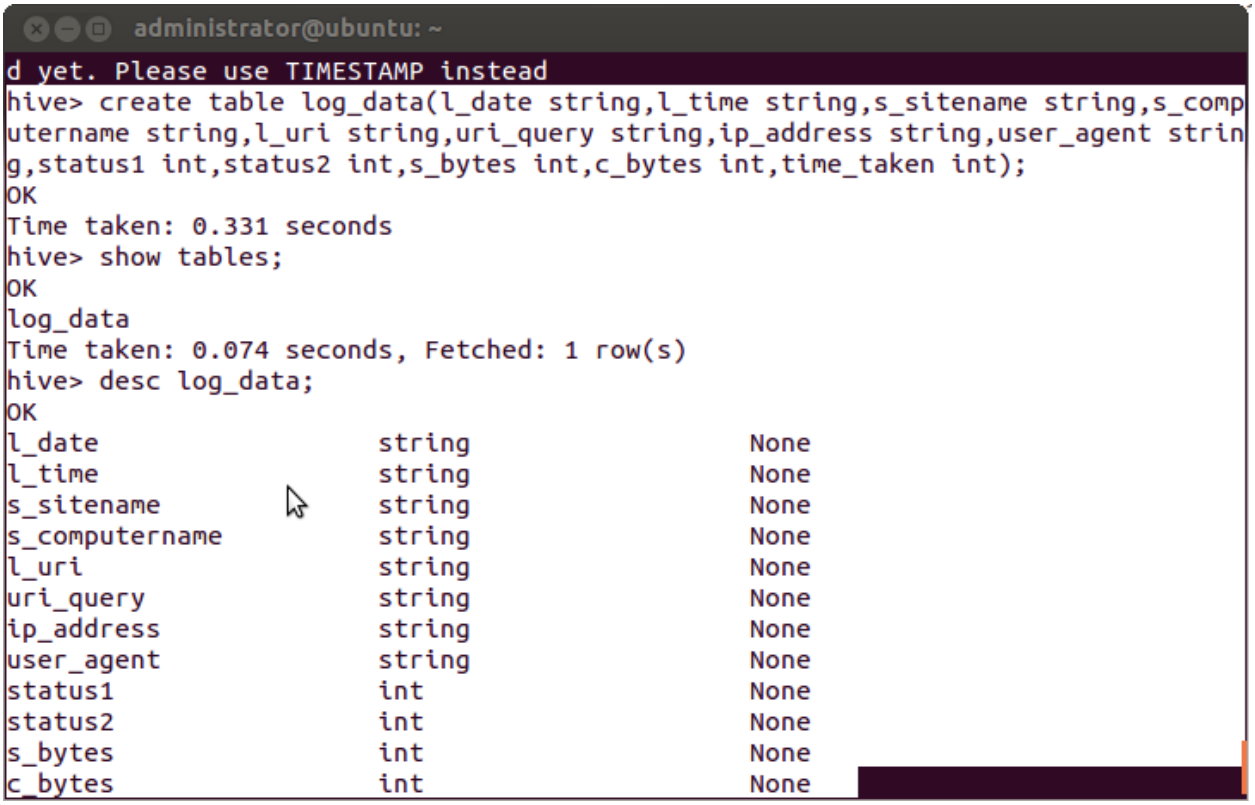
Install MySQL-Server

1) Sudo apt-get install mysql-server
2) Configuring MySQL UserName and Password
3) Creating User and granting all Privileges
Mysql –uroot –proot
Create user <USER_NAME> identified by <PASSWORD>
4) Extract and Configure Apache Hive
tar xvfz apache-hive-1.0.1.bin.tar.gz
5) Move Apache Hive from Local directory to Home directory
6) Set CLASSPATH in bashrc
Export HIVE_HOME = /home/apache-hive
Export PATH = $PATH:$HIVE_HOME/bin
7) Configuring hive-default.xml by adding My SQL Server Credentials

```
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>
jdbc:mysql://localhost:3306/hive?createDatabaseIfNotExist=true
</value>
</property>
<property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>com.mysql.jdbc.Driver</value>
</property>
<property>
<name>javax.jdo.option.ConnectionUserName</name>
<value>hadoop</value>
</property>
<property>
<name>javax.jdo.option.ConnectionPassword</name>
<value>hadoop</value>
</property>
```

8) Copying mysql-java-connector.jar to hive/lib directory.

INPUT/OUTPUT:



PRE-LAB VIVA QUESTIONS:

1. In Hive, explain the term „aggregation“ and its uses?
2. List out the Data types in Hive?

LAB ASSIGNMENT:

1. Analyze twitter data using Apache Hive.

POST-LAB VIVA QUESTIONS:

1. Explain the Built-in Functions in Hive?
2. Describe the various Hive Data types?

HIVE OPERATIONS

OBJECTIVE

Use Hive to create, alter, and drop databases, tables, views, functions, and indexes.

RESOURCES:

VMWare, XAMPP Server, Web Browser, 1GB RAM, Hard Disk 80 GB.

PROGRAM LOGIC:

SYNTAX for HIVE Database Operations

DATABASE Creation

CREATE DATABASE|SCHEMA [IF NOT EXISTS] <database name>

Drop Database Statement

DROP DATABASE StatementDROP (DATABASE|SCHEMA) [IF EXISTS] database_name [RESTRICT|CASCADE];

Creating and Dropping Table in HIVE

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.] table_name
[(col_name data_type [COMMENT col_comment], ...)]
[COMMENT table_comment] [ROW FORMAT row_format] [STORED AS file_format]

Loading Data into table log_dataSyntax:

LOAD DATA LOCAL INPATH '<path>/u.data' OVERWRITE INTO TABLE u_data;

Alter Table in HIVE

Syntax

ALTER TABLE name RENAME TO new_name
ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])
ALTER TABLE name DROP [COLUMN] column_name
ALTER TABLE name CHANGE column_name new_name new_type
ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])

Creating and Dropping View

CREATE VIEW [IF NOT EXISTS] view_name [(column_name [COMMENT column_comment], ...)] [COMMENT table_comment] AS SELECT ...

Dropping ViewSyntax:

DROP VIEW view_name

Functions in HIVE

String Functions:- round(), ceil(), substr(), upper(), reg_exp() etc
Date and Time Functions:- year(), month(), day(), to_date() etc

Aggregate Functions :- sum(), min(), max(), count(), avg() etc

INDEXES

```
CREATE INDEX index_name ON TABLE base_table_name (col_name, ...)
AS 'index.handler.class.name'
[WITH DEFERRED REBUILD]
[IDXPROPERTIES (property_name=property_value, ...)]
[IN TABLE index_table_name]
[PARTITIONED BY (col_name, ...)]
[
[ ROW FORMAT ...] STORED AS ...
| STORED BY ...
]
[LOCATION  hdfs_path]
[TBLPROPERTIES (...)]
```

Creating Index

```
CREATE INDEX index_ip ON TABLE log_data(ip_address) AS
'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler' WITH DEFERRED
REBUILD;
```

Altering and Inserting Index

```
ALTER INDEX index_ip_address ON log_data REBUILD;
```

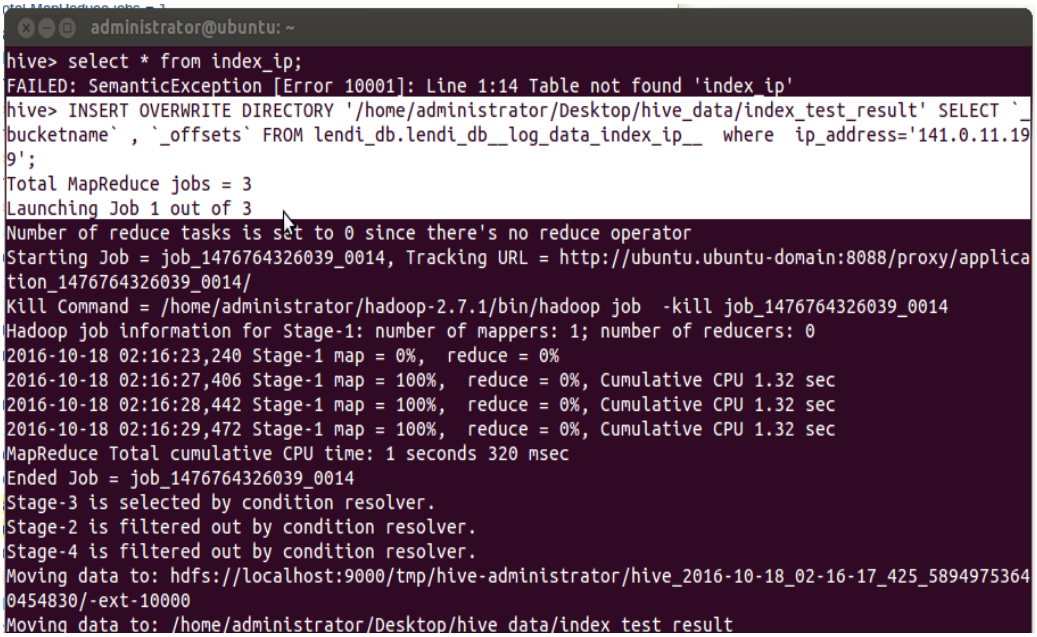
Storing Index Data in Metastore

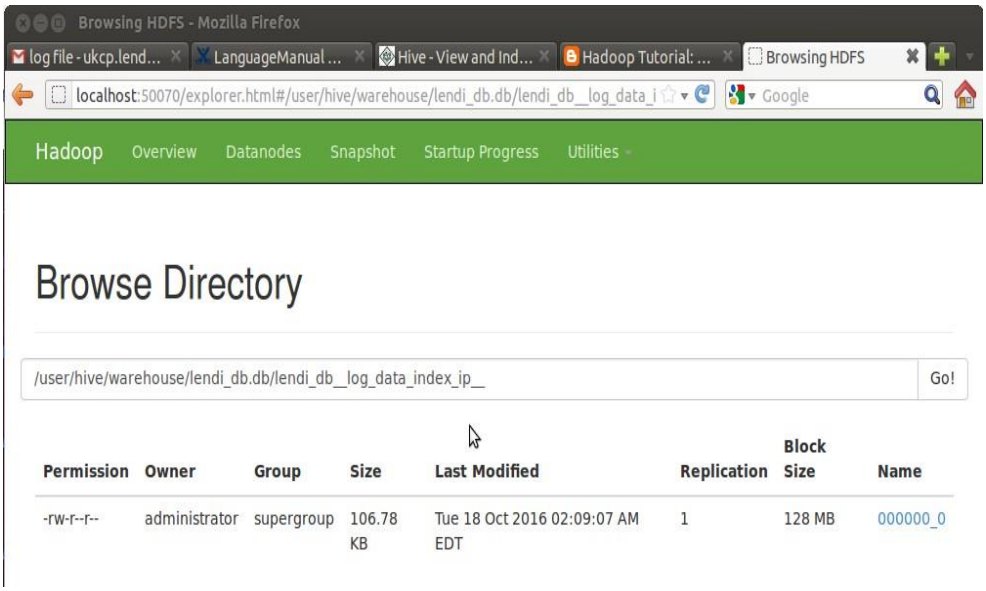
```
SET
hive.index.compact.file=/home/administrator/Desktop/big/metastore_db/tmp/index_ipadd
ress_result;
SET
hive.input.format=org.apache.hadoop.hive.ql.index.compact.HiveCompactIndexInputFor
mat;
```

Dropping Index

```
DROP INDEX INDEX_NAME on TABLE_NAME;
```

INPUT/OUTPUT:





PRE-LAB VIVA QUESTIONS:

1. How many types of joins are there in Pig Latin with an examples?
2. Write the Hive command to create a table with four columns: First name, last name,age, and income?

LAB ASSIGNMENT:

1. Analyze stock data using Apache Hive.

POST-LAB VIVA QUESTIONS:

1. Write a shell command in Hive to list all the files in the current directory?
2. List the collection types provided by Hive for the purpose a start-up companywant to use Hive for storing its data.

RUBRICS EVALUATION

Performance Criteria	Scale 1 (0-25%)	Scale 2 (26-50%)	Scale 3 (51-75%)	Scale 4 (76-100%)	Score (Numerical)
Understandability Ability to analyse Problem and Identify solution	Unable to understand the problem.	Able to understand the problem partially and unable to identify the solution	Able to understand the problem completely but unable to identify the solution	Able to understand the problem completely and able to provide alternative solution too.	
Logic Ability to specify Conditions & control flow that are appropriate for the problem domain.	Program logic is incorrect	Program logic is on the right track but has several errors	Program logic is mostly correct, but may contain an occasional boundary error or redundant or contradictory condition.	Program logic is correct, with no known boundary errors, and no redundant or contradictory conditions.	
Debugging Ability to execute /debug	Unable to execute program	Unable to debug several errors.	Able to execute program with several warnings.	Able to execute program completely	
Correctness Ability to code formulae and algorithms that reliably produce correct answers or appropriate results.	Program does not produce correct answers or appropriate results for most inputs.	Program approaches correct answers or appropriate results for most inputs, but can contain miscalculations in some cases.	Program produces correct answers or appropriate results for most inputs.	Program produces correct answers or appropriate results for all inputs tested.	
Completeness Ability to demonstrate and deliver on time.	Unable to explain the code.and the code was overdue.	Unable to explain the code and the code submission was late.	Able to explain code and the program was delivered within the due date.	Able to explain code and the program was delivered on time.	
TOTAL					

OUTCOMES OF LAB

After Completion of all the practical experiment students have achieved:

- Students learned the definition and significance of the SRS (Software Requirement Specification)
- Students have explored the different platform/software/hardware used for UML
- Students learned to work on Star UML and to create different UML diagrams.

Computer Lab's Do's and Don't and Safety Rules

DO's

- Please switch off the Mobile/Cell phone before entering Lab.
- Check whether all peripheral are available at your desktop before proceeding for the session
- Arrange all the peripheral and seats before leaving the lab.
- Properly shutdown the system before leaving the lab.
- Keep the bag outside in the racks.
- Enter the lab on time and leave at proper time.
- Maintain the decorum of the lab.

DON'TS

- Don't mishandle the system.
- Don't leave the system on standing for long
- Don't bring any external material in the lab.
- Don't make noise in the lab.
- Don't bring the mobile in the lab.
- Don't enter in the lab without permission of lecturer/laboratory technician immediately
- Don't delete or make any modification in system files.
- Don't bring storage devices like pen drive without permission of lecturer/laboratory technician.

Computer Lab Safety Rules

- Know the location of the fire extinguisher and how to use them in case of an emergency.
- Report fires or accidents to your lecturer/laboratory technician immediately
- Report any broken plugs or exposed electrical wires to your lecturer/laboratory technician immediately.
- Avoid stepping on electrical wires or any other computer cables.
- Do not open the system unit casing or monitor casing particularly when the power is turned on.
- Do not touch, connect or disconnect any plug or cable without your lecturer/laboratory technician's permission.
- Do not bring any food or drinks near the machine.