THE NEOTIA UNIVERSITY

Computer Science & Engineering

Bigdata and Hadoop Lab Manual

1. PROGRAM OUTCOMES:

PROGRAM OUTCOMES (POs)	
PO-1:	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (**Engineering Knowledge**).
PO-2:	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (**Problem Analysis**).
PO-3:	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (**Design/Development of Solutions**).
PO-4:	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (**Conduct Investigations of Complex Problems**).
PO-5:	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (**Modern Tool Usage**).
PO-6:	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (**The Engineer and Society**).
PO-7:	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development (**Environment and Sustainability**).
PO-8:	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (**Ethics**).
PO-9:	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (**Individual and Team Work**).
PO-10:	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (**Communication**).
PO-11:	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one"s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO-12:	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (**Life-long learning**).

2. PROGRAM SPECIFIC OUTCOMES (PSOs):

PROGRAM SPECIFIC OUTCOMES (PSO'S)	
PSO – I	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.
PSO – II	**Problem-Solving Skills:** The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.
PSO – III	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

3. INDEX:

		PIG LATIN MODES, PROGRAMS	
WEEK-10	a	Run the Pig Latin Scripts to find Word Count.	41
	b	Run the Pig Latin Scripts to find a max temp for each and every year.	42
WEEK-11		HIVE	
	a	Installation of HIVE.	43
WEEK-12		HIVE OPERATIONS	
	a	Use Hive to create, alter, and drop databases, tables, views, functions, and indexes.	45

INSTALL VMWARE
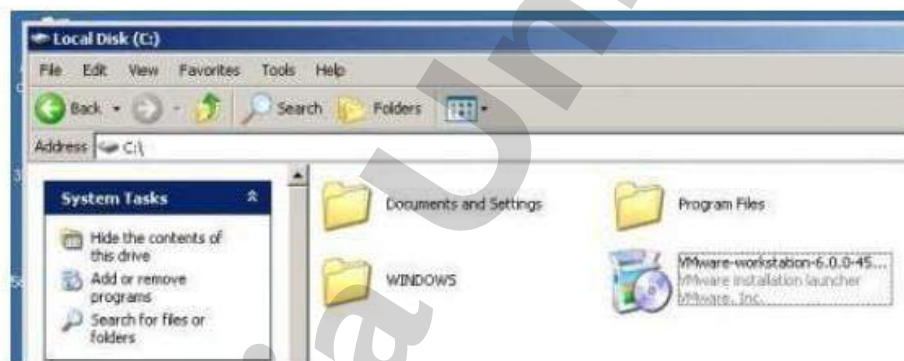
1.1   OBJECTIVE:

To Install VMWare.

1.2   RESOURCES:

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

1.3   PROGRAM LOGIC:

STEP 1. First of all, enter to the official site of VMware and download VMware Workstation https://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
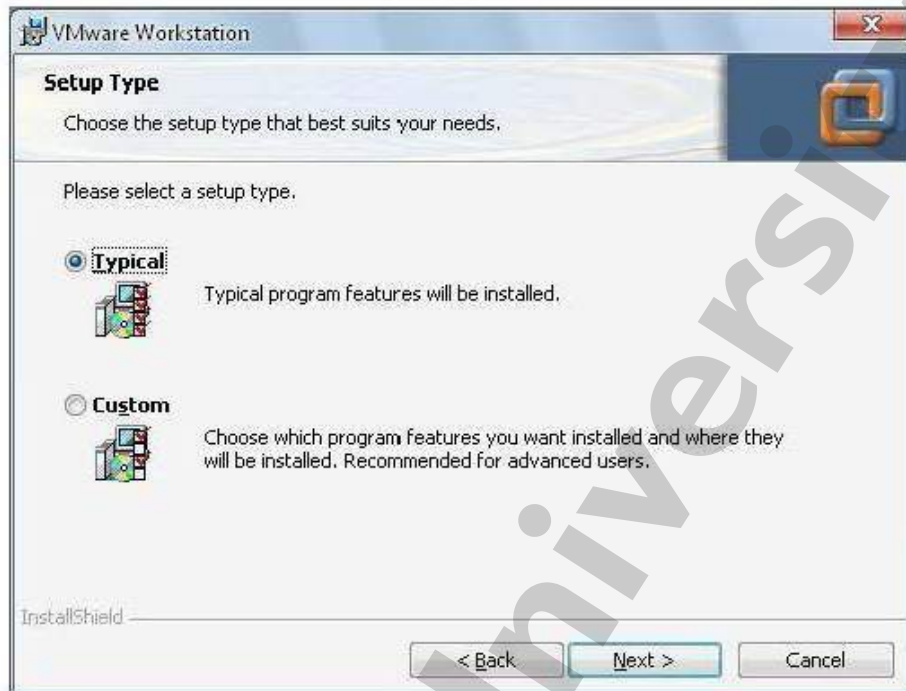
STEP 4. By clicking "**Next**" buttons, to begin the installation, click on **Install** button at the end
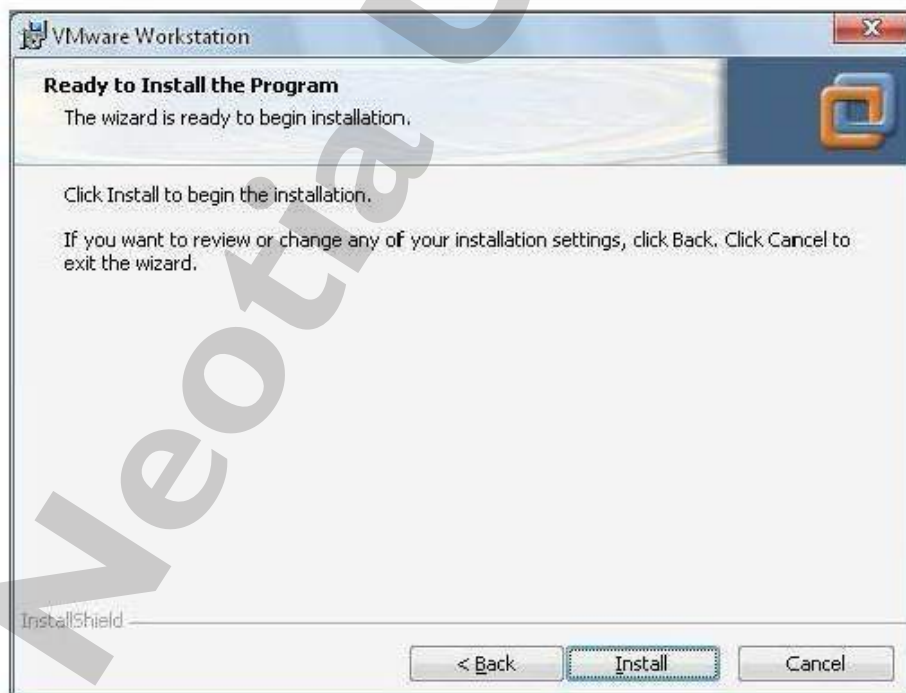
STEP 5. This will install VMware Workstation software on your PC, After installation complete, click on **Finish** button. Then restart your PC. Then open this software
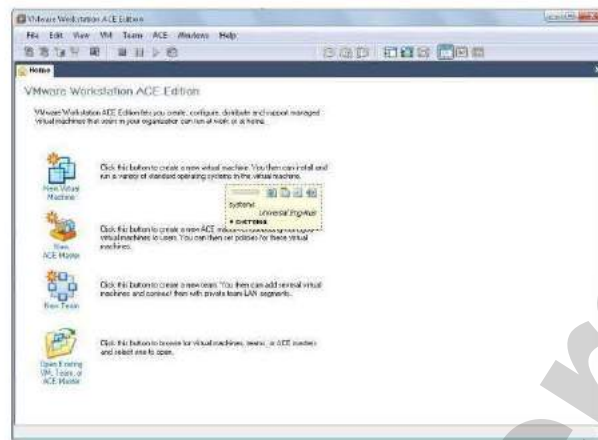
6. In this step we try to create new "virtual machine". Enter to File menu, then New-> Virtual Machine

Click on **Next** button, then check **Typical** option as below
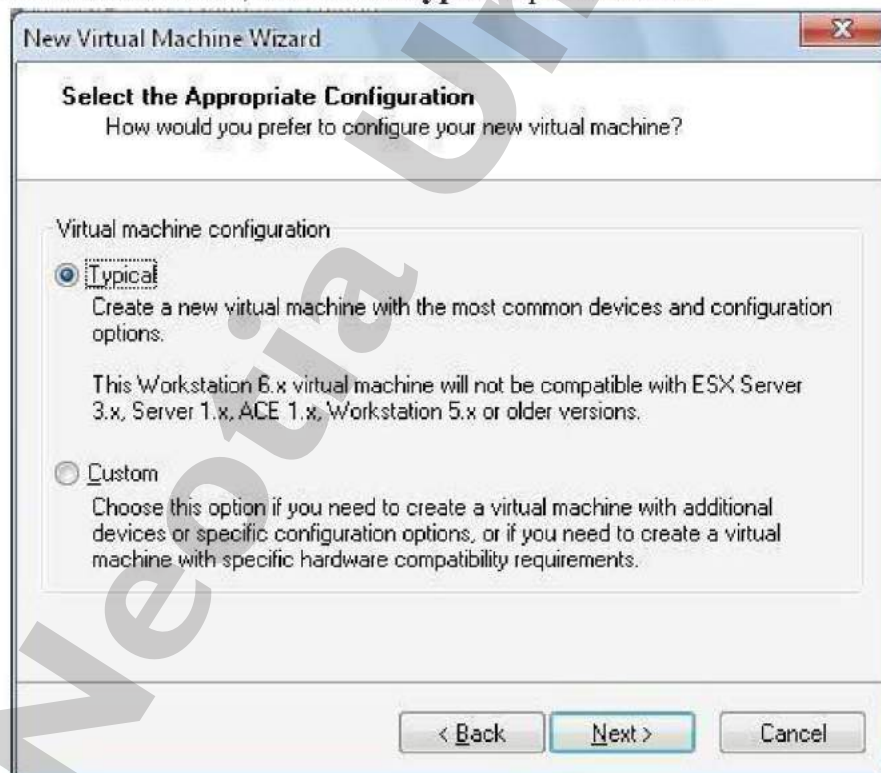
Then click **Next** button, and check your OS version. In this example, as we"re going to setup Oracle server on CentOS, we"ll check **Linux** option and from "*version*" option we"ll check **Red Hat**

Enterprise Linux 4
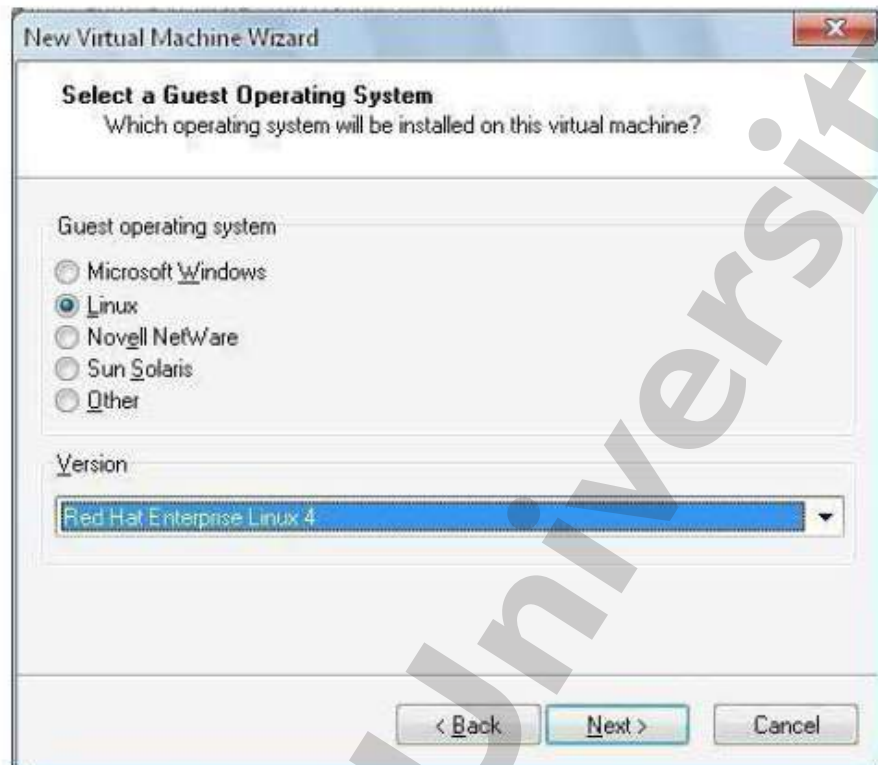
By clicking **Next** button, we'll give a name to our virtual machine, and give directory to create this new virtual machine
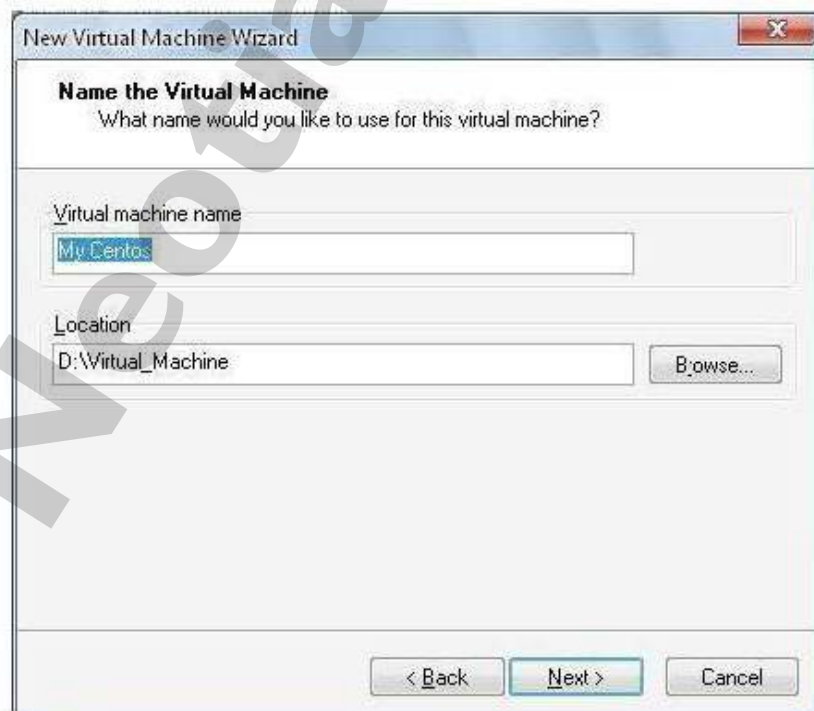
Then select **Use bridged networking** option and click Next.
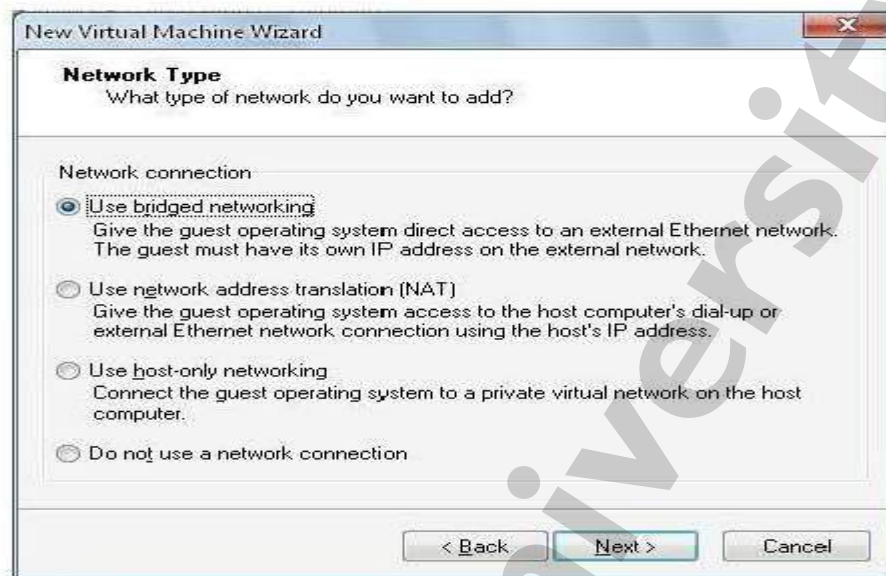
Then you''ve to define size of hard disk by entering its size. I''ll give 15 GB hard disk space and please check **Allocate all disk space now** option
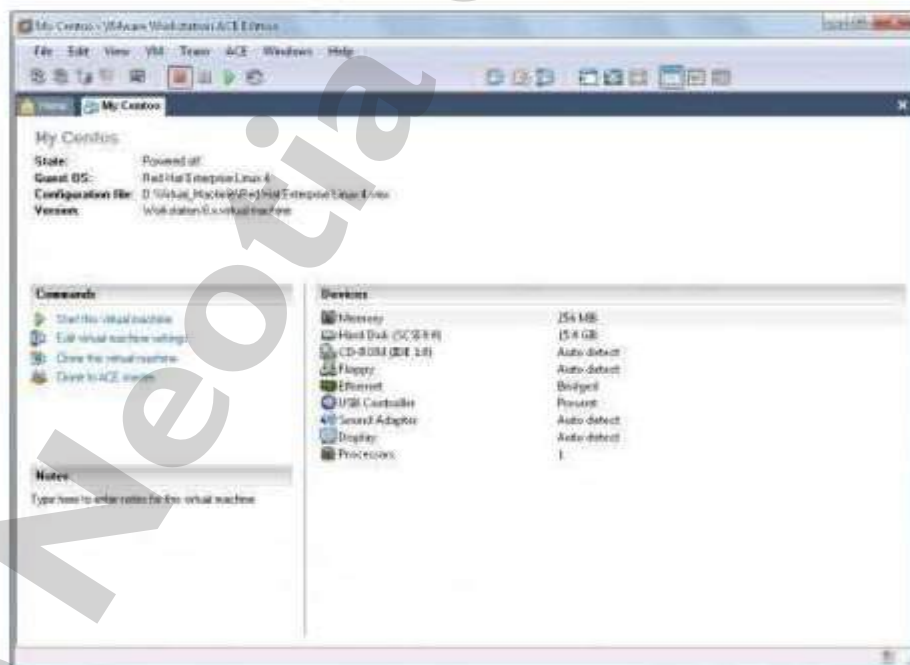
Here, you can delete **Sound Adapter, Floppy and USB Controller** by entering "Edit virtual machine settings". If you''re going to setup Oracle Server, please make sure you''ve increased your Memory (RAM) to 1GB.

1.4   INPUT/OUTPUT
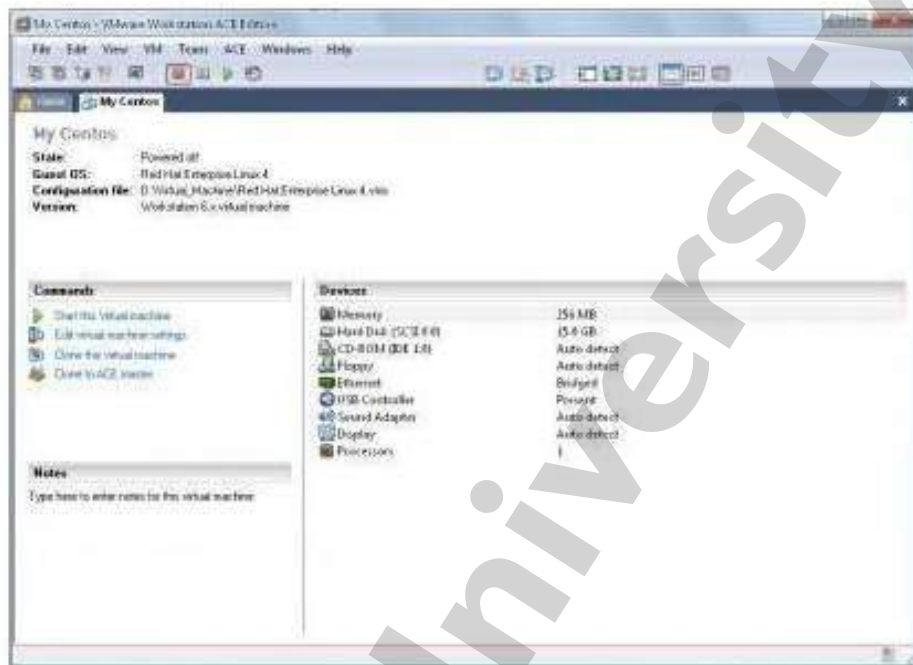
1.5 PRE LAB VIVA QUESTIONS:

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

1.6 LAB ASSIGNMENT:

1. Install Pig?
2. Install Hive?

1.7 POST LAB VIVA QUESTIONS:

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

HADOOP MODES

2.1 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.

2.2 RESOURCES:

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

2.3 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>

<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
192.168.1.4 pcetcse2
192.168.1.5 pcetcse3
192.168.1.7 pcetcse4
192.168.1.8 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/

2.4 INPUT/OUTPUT:
ubuntu @localhost> jps

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

localhost:50030/machines.jsp?type=active

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/

localhost:50070/logs/

Directory: /logs/

hadoop-sudheer-datanode-dn2.log	6487 bytes	8 May, 2015 12:10:13 PM
hadoop-sudheer-datanode-dn2.log.2015-05-07	301426 bytes	7 May, 2015 9:23:03 PM
hadoop-sudheer-datanode-dn2.out	719 bytes	8 May, 2015 12:09:25 PM
hadoop-sudheer-datanode-dn2.out.1	719 bytes	7 May, 2015 9:00:26 PM
hadoop-sudheer-datanode-dn2.out.2	719 bytes	7 May, 2015 8:55:58 PM
hadoop-sudheer-jobtracker-dn2.log	22631 bytes	8 May, 2015 12:09:39 PM
hadoop-sudheer-jobtracker-dn2.log.2015-05-07	678885 bytes	7 May, 2015 9:22:52 PM
hadoop-sudheer-jobtracker-dn2.out	719 bytes	8 May, 2015 12:09:28 PM
hadoop-sudheer-jobtracker-dn2.out.1	719 bytes	7 May, 2015 9:00:28 PM
hadoop-sudheer-jobtracker-dn2.out.2	719 bytes	7 May, 2015 8:56:01 PM
hadoop-sudheer-namenode-dn2.log	17042 bytes	8 May, 2015 12:11:36 PM
hadoop-sudheer-namenode-dn2.log.2015-05-07	17446 bytes	7 May, 2015 9:00:28 PM
hadoop-sudheer-namenode-dn2.out	719 bytes	8 May, 2015 12:09:24 PM
hadoop-sudheer-namenode-dn2.out.1	719 bytes	7 May, 2015 9:00:24 PM
hadoop-sudheer-namenode-dn2.out.2	719 bytes	7 May, 2015 8:55:57 PM
hadoop-sudheer-secondarynamenode-dn2.log	2085 bytes	8 May, 2015 12:09:32 PM
hadoop-sudheer-secondarynamenode-dn2.log.2015-05-07	296453 bytes	7 May, 2015 9:23:08 PM
hadoop-sudheer-secondarynamenode-dn2.out	719 bytes	8 May, 2015 12:09:27 PM
hadoop-sudheer-secondarynamenode-dn2.out.1	719 bytes	7 May, 2015 9:00:27 PM
hadoop-sudheer-secondarynamenode-dn2.out.2	719 bytes	7 May, 2015 8:56:00 PM
hadoop-sudheer-tasktracker-dn2.log	4969 bytes	8 May, 2015 12:09:35 PM
hadoop-sudheer-tasktracker-dn2.log.2015-05-07	60226 bytes	7 May, 2015 9:22:57 PM
hadoop-sudheer-tasktracker-dn2.out	719 bytes	8 May, 2015 12:09:29 PM
hadoop-sudheer-tasktracker-dn2.out.1	719 bytes	7 May, 2015 9:00:30 PM
hadoop-sudheer-tasktracker-dn2.out.2	719 bytes	7 May, 2015 8:56:02 PM
history/	4096 bytes	7 May, 2015 8:56:08 PM

localhost:50060/tasktracker.jsp

tracker_dn2:localhost/127.0.0.1:49820 Task Tracker Status

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

localhost Hadoop Map/Reduce Administration - Mozilla Firefox

Hadoop NameNode loc... × | localhost Hadoop Map/... ×

localhost:50030/jobtracker.jsp

Quick Links

localhost Hadoop Map/Reduce Administration

State: RUNNING
Started: Fri May 08 12:09:33 IST 2015
Version: 1.2.0, r1479473
Compiled: Mon May 6 06:59:37 UTC 2013 by hortonfo
Identifier: 201505081209
SafeMode: OFF

Cluster Summary (Heap Size is 55.5 MB/889 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Graylisted Nodes	Exc N
0	0	0	1	0	0	0	0	2	2	4.00	0	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name) [   ]
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

2.5 PRE LAB VIVA QUESTIONS:

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

2.6   LAB ASSIGNMENT:

1   How to configure the daemons in the browser.

2.7   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.

WEEK- 3

USING LINUX OPERATING SYSTEM

3.1   OBJECTIVE:

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

3.2   RESOURCES:

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

3.3   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:

List current contents of directory

```
guru99@VirtualBox:~$ ls
Desktop      Downloads         Music        [Public   sample1   Templates
Documents    examples.desktop  Pictures     sample    sample2   Videos
```

Remove the file sample1

```
guru99@VirtualBox:~$ rm sample1
```

List directory , to check file has been deleted

```
guru99@VirtualBox:~$ ls
Desktop      Downloads         Music        Public    sample2    Videos
Documents    examples.desktop  Pictures     sample    Templates
guru99@VirtualBox:~$
```

3.5   PRE-LAB VIVA QUESTIONS:

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

3.6   LAB ASSIGNMENT:

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

3.7   POST-LAB VIVA QUESTIONS:

1. What is the purpose of rm command?

2. What is the difference between Linux and windows commands?

4.1 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.

4.2 RESOURCES:

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

4.3 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
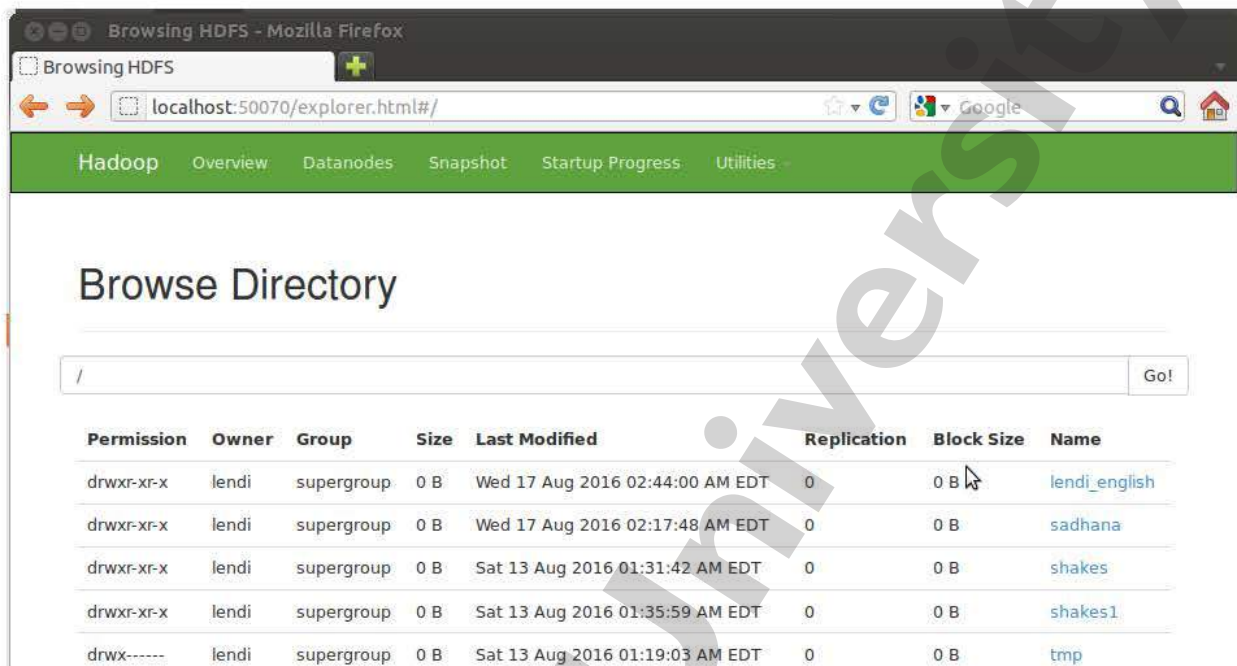
   hadoop fs -rm example.txt

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

4.4 INPUT/OUTPUT:

4.5 PRE LAB VIVA QUESTIONS:
1) Define Hadoop?
2) List out the various use cases of Hadoop?

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

4.7 POST LAB VIVA QUESTIONS:
1. Distinguish the Hadoop Ecosystem?
2. Demonstrate divide and conquer philosophy in Hadoop Cluster?

MAPREDUCE PROGRAM 1

5.1 OBJECTIVE:

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

5.2 RESOURCES:

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

5.3 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);

}
```

5.4 INPUT/OUTPUT:

5.5 PRE-LAB VIVA QUESTIONS:
1. Justify how hadoop technology satisfies the business insights now -a –days?
2. Define Filesystem?

5.6 LAB ASSIGNMENT:

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

5.7 POST-LAB VIVA QUESTIONS:
1. Define what is block in HDFS?
2. Why is a block in HDFS so large?

MAPREDUCE PROGRAM 2

6.1 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.

6.2 RESOURCES:

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

6.3 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 "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

6.4 INPUT/OUTPUT:
Set of Weather Data over the years

6.5 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?

6.6 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.

6.7 POST-LAB VIVA QUESTIONS:

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

7.1 OBJECTIVE:

Implement matrix multiplication with Hadoop Map Reduce.

7.2 RESOURCES:

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

7.3 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

7.4 INPUT/OUTPUT:

7.5 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?

7.6 LAB ASSIGNMENT:
1. Implement matrix addition with Hadoop Map Reduce.

7.7 POST-LAB VIVA QUESTIONS:
1. What is partitioner in Hadoop?
2. Explain of RecordReader in Hadoop?

PIG LATIN LANGUAGE - PIG

8.1 OBJECTIVE:

1. Installation of PIG.

8.2 RESOURCES:

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

8.3 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;

8.4 INPUT/OUTPUT:

Input as Website Click Count Data

8.5 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?

8.6 LAB ASSIGNMENT:

1. Process baseball data using Apache Pig.

8.7 POST-LAB VIVA QUESTIONS:

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

9.1   OBJECTIVE:

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

9.2   RESOURCES:

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

9.3   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)

9.4 INPUT / OUTPUT :

9.5 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?

9.6 LAB ASSIGNMENT:

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

9.7 PRE-LAB VIVA QUESTIONS:

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

10.1 OBJECTIVE:

    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.

10.2 RESOURCES:

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

10.3 PROGRAM LOGIC:

<u>Run the Pig Latin Scripts to find Word Count.</u>

```
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;
```

<u>Run the Pig Latin Scripts to find a max temp for each and every year</u>

```
-- 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;
```

10.4 INPUT / OUTPUT:

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

10.5 PRE-LAB VIVA QUESTIONS:

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

10.6 LAB ASSIGNMENT:

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

10.7 POST-LAB VIVA QUESTIONS:

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

11.1 OBJECTIVE:

Installation of HIVE.

11.2 RESOURCES:

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

11.3   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>

```xml
</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.

11.4 INPUT/OUTPUT:

```
administrator@ubuntu: ~
d yet. Please use TIMESTAMP instead
hive> create table log_data(l_date string,l_time string,s_sitename string,s_comp
utername string,l_uri string,uri_query string,ip_address string,user_agent strin
g,status1 int,status2 int,s_bytes int,c_bytes int,time_taken int);
OK
Time taken: 0.331 seconds
hive> show tables;
OK
log_data
Time taken: 0.074 seconds, Fetched: 1 row(s)
hive> desc log_data;
OK
l_date                  string                  None
l_time                  string                  None
s_sitename              string                  None
s_computername          string                  None
l_uri                   string                  None
uri_query               string                  None
ip_address              string                  None
user_agent              string                  None
status1                 int                     None
status2                 int                     None
s_bytes                 int                     None
c_bytes                 int                     None
```

11.5 PRE-LAB VIVA QUESTIONS:
   1. In Hive, explain the term „aggregation" and its uses?
   2. List out the Data types in Hive?

11.6 LAB ASSIGNMENT:
   1. Analyze twitter data using Apache Hive.

11.7 POST-LAB VIVA QUESTIONS:
   1. Explain the Built-in Functions in Hive?
   2. Describe the various Hive Data types?

12.1 OBJECTIVE:

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

12.2 RESOURCES:

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

12.3 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_data

Syntax:

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 View

Syntax:

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;

```
0.6.20.6          Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.1;+Trident/4.0;+GTB7.5;+SLC
R+2.0.50727;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729;+Media+Center+PC+6.0;+InfoPath.2)   304
11     498     0
2014-12-23     23:08:38     W3SVC1  NEWINTSERV2     /trf/elast/images/small/pic3.jpg
6.20.6           Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.1;+Trident/4.0;+GTB7.5;+SLC
R+2.0.50727;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729;+Media+Center+PC+6.0;+InfoPath.2)   304
10     497     0
2014-12-23     23:16:07     W3SVC1  NEWINTSERV2     /trf/elast/css/demo.css -        10.
ozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.0;+SLCC1;+.NET+CLR+2.0.50727;+.NET+CLR+3.0.0
CLR+1.1.4322;+InfoPath.2)       304     0       210     458     0
2014-12-23     23:16:07     W3SVC1  NEWINTSERV2     /trf/elast/css/elastislide.css  -
0.22    Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.0;+SLCC1;+.NET+CLR+2.0.50727;+.NET+
06;+.NET+CLR+1.1.4322;+InfoPath.2)      304     0       210     465     0
2014-12-23     23:16:07     W3SVC1  NEWINTSERV2     /trf/elast/images/small/pic11.jpg
0.3.20.22        Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.0;+SLCC1;+.NET+CLR+2.0.5072
+3.0.04506;+.NET+CLR+1.1.4322;+InfoPath.2)      304     0       211     469     0
2014-12-23     23:16:07     W3SVC1  NEWINTSERV2     /trf/elast/images/small/pic12.jpg
0.3.20.22        Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.0;+SLCC1;+.NET+CLR+2.0.5072
+3.0.04506;+.NET+CLR+1.1.4322;+InfoPath.2)      304     0       211     469     0
2014-12-23     23:16:07     W3SVC1  NEWINTSERV2     /trf/elast/images/small/pic10.jpg
0.3.20.22        Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.0;+SLCC1;+.NET+CLR+2.0.5072
+3.0.04506;+.NET+CLR+1.1.4322;+InfoPath.2)      304     0       211     469     0
2014-12-23     23:16:07     W3SVC1  NEWINTSERV2     /trf/elast/images/small/pic9.jpg
0.3.20.22        Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+6.0;+SLCC1;+.NET+CLR+2.0.5072
+3.0.04506;+.NET+CLR+1.1.4322;+InfoPath.2)      304     0       210     467     0
2014-12-23     23:16:07     W3SVC1  NEWINTSERV2     /trf/elast/images/small/pica.jpg
```



```
hive> select * from index_ip;
FAILED: SemanticException [Error 10001]: Line 1:14 Table not found 'index_ip'
hive> INSERT OVERWRITE DIRECTORY '/home/administrator/Desktop/hive_data/index_test_result' SELECT `_
bucketname` , `_offsets` FROM lendi_db.lendi_db__log_data_index_ip__  where  ip_address='141.0.11.19
9';
Total MapReduce jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1476764326039_0014, Tracking URL = http://ubuntu.ubuntu-domain:8088/proxy/applica
tion_1476764326039_0014/
Kill Command = /home/administrator/hadoop-2.7.1/bin/hadoop job  -kill job_1476764326039_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2016-10-18 02:16:23,240 Stage-1 map = 0%,  reduce = 0%
2016-10-18 02:16:27,406 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.32 sec
2016-10-18 02:16:28,442 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.32 sec
2016-10-18 02:16:29,472 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.32 sec
MapReduce Total cumulative CPU time: 1 seconds 320 msec
Ended Job = job_1476764326039_0014
Stage-3 is selected by condition resolver.
Stage-2 is filtered out by condition resolver.
Stage-4 is filtered out by condition resolver.
Moving data to: hdfs://localhost:9000/tmp/hive-administrator/hive_2016-10-18_02-16-17_425_5894975364
0454830/-ext-10000
Moving data to: /home/administrator/Desktop/hive_data/index_test_result
```
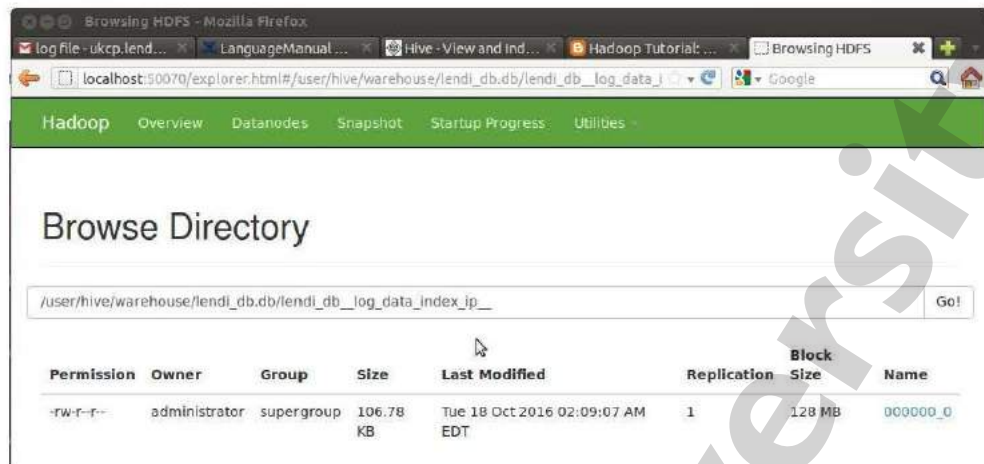
12.5 PRE-LAB VIVA QUESTIONS:

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?

12.6 LAB ASSIGNMENT:

1. Analyze stock data using Apache Hive.

12.7 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 company want to use Hive for storing its data.