	Hope Foundation's **Finolex Academy of Management and Technology, Ratnagiri** **Information Technology Department**		
Subject name: Big Data Lab			Subject Code: ITL801
Class	BE IT	Semester – VIII (CBGS)	Academic year: 2019-20
Name of Student	**Kazi Jawwad A Rahim**	**QUIZ Score :**	
Roll No	**28**	Assignment/Experiment No.	01
Title: Hadoop ecosystem demo in HUE and study of commands required in Hadoop installation			

1. Course objectives applicable COB1. To understand main business drivers and key issues of BDA
COB2. To acquire knowledge about fundamentals of Big Data Analytics

2. Course outcomes applicable:

CO1 : Understand the key issues in big data management and its associated applications in intelligent business and scientific computing.

CO2 - Acquire fundamental enabling techniques and scalable algorithms like Hadoop, Map Reduce and NO SQL in big data analytics.

3. Learning Objectives:

1. To understand the concept Big Data Analytics and Hadoop Ecosystem
2. To study the role of Hadoop in BDA
3. To explore Hadoop through HUE dashboard

4. Practical applications of the assignment/experiment: Hue is a widely used GUI dashboard for Hadoop

5. Prerequisites:

1. Knowledge of Data warehousing
2. SQL Queries

6. Hardware Requirements:

1. PC with 4GB RAM, 500GB HDD

7. Software Requirements:

1. Ubuntu / Windows , access to internet www.gethue.com

8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)

1. What is a Hadoop?
2. List components of Hadoop ecosystem?
3. Enlist four limitations of Hadoop?

9. Experiment/Assignment Evaluation:

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used -	05	6
2	Neatness/presentation	02	2
3	Punctuality	02	2
Date of performance (DOP)	8/1/20	**Total marks obtained**	09 / **10**
Date of checking (DOC)	15/1/20	**Signature of teacher**	

Theory:

Apache Hadoop is an open source framework used for distributed storage and processing of dataset of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurences and should be automatically handled by the framework.

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

The base apache hadoop framework is composed of the following modules-

Hadoop Common: Contains libraries and

utilies needed by other Hadoop modules.
Hadoop Distributed File System (HDFS): A distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
Hadoop YARN: A platform responsible for managing computing resources in clusters and using them for scheduling users application. ~~and~~
Hadoop MapReduce: An implementation of the MapReduce programming model for large-scale data processing.

The term Hadoop has come to refer not just to the aforementioned base modules and sub-modules, but also the ecosystem, or collection of additional software packages that can be installed on top of or alongside Hadoop, such as Apache Pig, Apache Hive, Apache HBase, Apache Phoenix, Apache Spark, Apache Zookeeper, Cloudera Impala, Apache Flume, Apache Flume, Apache Flume, Apache Sqoop, Apache Oozie and Apache Storm.

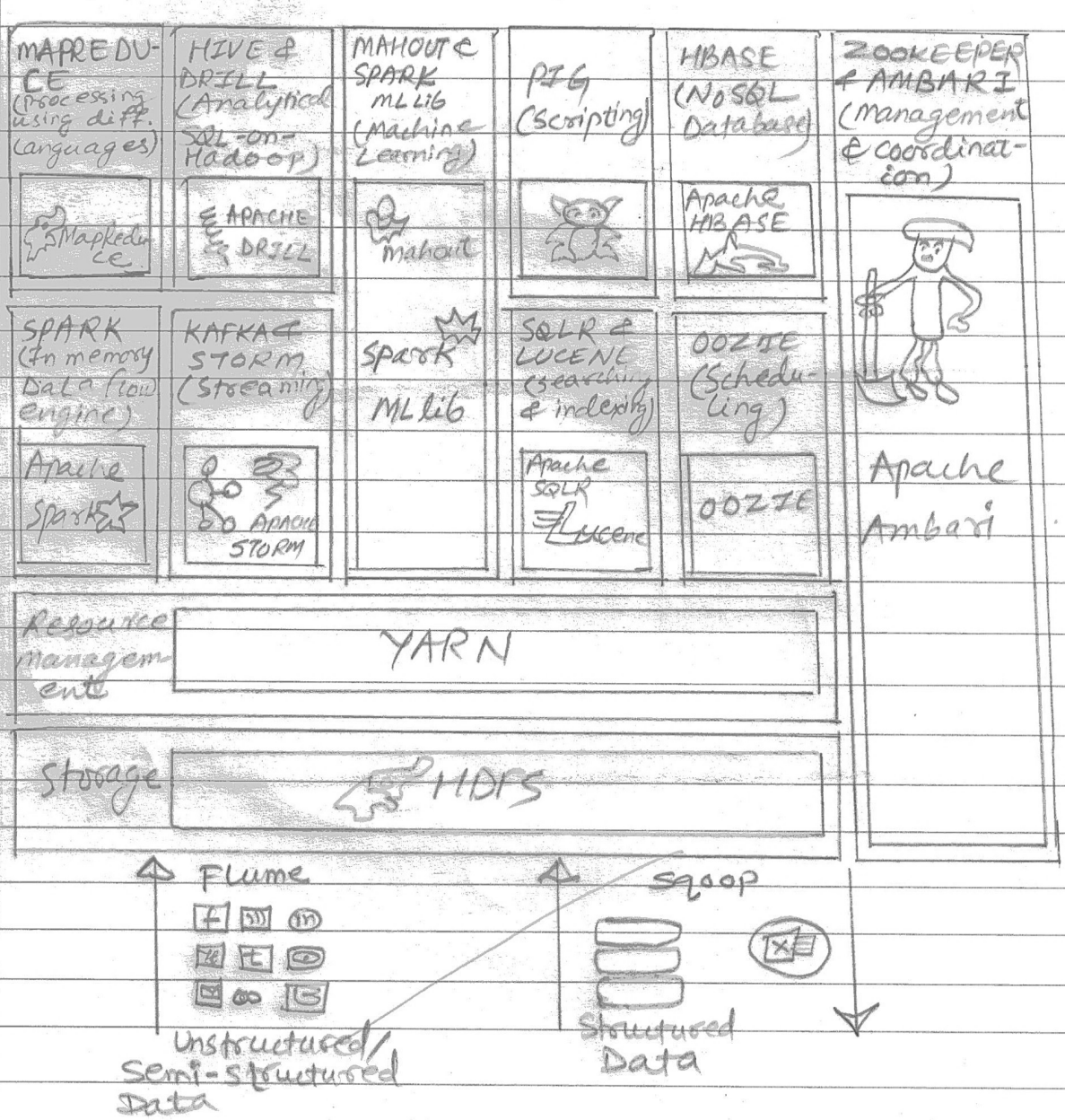


Fig: 1: Hadoop Ecosystem

The Hadoop ecosystem includes both official Apache open source projet and wide range of commercial tools and solutions. Some of the best-known open source examples include Spark, Hive, Pig, Oozie and Sqoop. Commercial Hadoop offerings are even more diverse and include platforms and packaged distribut from vendors such as Cloudera, Hortonworks and MapR.

2. Precautions :

1. Don't format live machine
2. Check the OS version and compatibility of instructions
3. Incase the Hadoop connection give error on Hue, please log out and restart the session

3. Installation Steps / Performance Steps – Important Commands for Hadoop Installation -

Hadoop requires a working Java installation. I will therefore describe the installation of Java 1.7. You can download the tar for **Java SE Development Kit 7u21.** For the sake of the tutorial I am using jdk-7u21-linux-i586.tar.gz. Please choose the right version according to your machine configuration.
We will use /usr/java directory to place our extracted JDK. You may choose any other directory according to your choice. I assume that tar ball has been downloaded in the download directory. To un-tar the tar ball follows the following steps:

```
# Change current directory to /home/username/Downloads. Replace username with your current user.
cd /home/username/Downloads
# Un-tar the tar ball.
tar xzvf /home/username/Downloads/jdk-7u21-linux-i586.tar.gz
# Change the user to root.
su –
Password:
# Change current directory to /usr
cd /usr
# Create a new directory named java, if it is already not present
mkdir java
# Move the extracted tar ball to /usr/local/java
mv /home/username/Downloads/jdk1.7.0_21 /usr/java
```

After this is done exit out of root user.

Dedicated Hadoop User

We will use a dedicated Hadoop user account for running Hadoop. While that's not required it is recommended because it helps to separate the Hadoop installation from other software applications and user accounts running on the same machine.

```
# Switch to root user
su –
Password:
# Create a new group named hadoop and new user named hduser.
groupadd hadoop
useradd hduser –g hadoop
```

This will add the user **hduser** and the group **hadoop** to your local machine.

Configuring SSH

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it (which is what we want to do in this short tutorial). For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost for the hduser user we created in the previous section.
I assume that you have SSH up and running on your machine and configured it to allow SSH public key authentication.
First, we have to generate an SSH key for the hduser user.

```
su – hduser
```

ssh-keygen -t rsa -P ""

The second line will create an RSA key pair with an empty password. Generally, using an empty password is not recommended, but in this case it is needed to unlock the key without your interaction (you don't want to enter the passphrase every time Hadoop interacts with its nodes).
Second, you have to enable SSH access to your local machine with this newly created key.

cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
The final step is to test the SSH setup by connecting to your local machine with the hduser user. The step is also needed to save your local machine's host key fingerprint to the hduser user's known_hosts file. If you have any special SSH configuration for your local machine like a non-standard SSH port, you can define host-specific SSH options in $HOME/.ssh/config
ssh localhost

This should connect you to the localhost.
If the SSH connect should fail, these general tips might help:

- Enable debugging with ssh -vvv localhost and investigate the error in detail.
- Check the SSH server configuration in /etc/ssh/sshd_config, in particular the options PubkeyAuthentication (which should be set to yes) and AllowUsers (if this option is active, add the hduser user to it). If you made any changes to the SSH server configuration file, you can force a configuration reload with sudo /etc/init.d/ssh reload
- Check that your firewall is not blocking the SSH service. If this is the case open your firewall settings and unblock SSH service. Then type service sshd restart and iptables –flush (in case of fedora 17) while being the root user.
- You can also use namp to check if port 22 is open on your machine.
- Make sure that /etc/hosts has proper settings.

Disabling IPv6
One problem with IPv6 is that using 0.0.0.0 for the various networking-related Hadoop configuration options will result in Hadoop binding to the IPv6 addresses. In my case, I realized that there's no practical point in enabling IPv6 when you are not connected to any IPv6 network. Hence, I simply disabled IPv6. Your mileage may vary.
To disable IPv6, open /etc/sysctl.conf in the editor of your choice and add the following lines to the end of the file:
# disable ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
You have to reboot your machine in order to make the changes take effect. You can check whether IPv6 is enabled on your machine with the following command:

cat /proc/sys/net/ipv6/conf/all/disable_ipv6

A return value of 0 means IPv6 is enabled, a value of 1 means disabled (that's what we want).
Hadoop Installation
Download the Apache Hadoop from Apache mirrors and extract the contents of the Hadoop package to a location of your choice.Use a stable release. Currently the stable version is hadoop-1.1.2. I am using /home/user/Downloads directory to extract the contents.
Change your location to the directory where tar was downloaded.
cd /home/user/Downloads
Extract the tar

```
tar xzvf  hadoop-1.1.2.tar.gz
# Rename to Hadoop
mv  hadoop-1.1.2 hadoop
# Change to root user
su –
Password:
```

Change your current directory to the directory where you want to install Hadoop. This location should be kept same for all the nodes in the cluster. For tutorial we will use /usr/local for this task.

```
cd /usr/local
# Move extracted contents of Hadoop from /home/user/Downloads to /usr/local
mv /home/user/Downloads/hadoop /usr/local/
# Change owner to hduser
chown –R hduser:hadoop /usr/local/hadoop
# Exit root user
exit
```

Setting the Environment Variables

Add the following lines to the end of the $HOME/.bashrc file of user hduser. If you use a shell other than bash, you should of course update its appropriate configuration files instead of.bashrc.

```
# Set Hadoop-related environment variables
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_CONF_DIR=${HADOOP_HOME}/conf
# Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop later on)
export JAVA_HOME= /usr/java/jdk1.7.0_21
# Some convenient aliases and functions for running Hadoop-related commands
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"
# If you have LZO compression enabled in your Hadoop cluster and
# compress job outputs with LZOP (not covered in this tutorial):
# Conveniently inspect an LZOP compressed file from the command
# line; run via:
#
# $ lzohead /hdfs/path/to/lzop/compressed/file.lzo
#
# Requires installed 'lzop' command.
#
lzohead () {
   hadoop fs -cat $1 | lzop -dc | head -1000 | less
}
# Add Hadoop bin/ directory to PATH
export PATH=$PATH:$HADOOP_HOME/bin
```

After this is done restart your terminal.

Configuring $HADOOP_CONF_DIR/hadoop-env.sh

The only required environment variable we have to configure for Hadoop in this tutorial is JAVA_HOME. Open $HADOOP_CONF_DIR/hadoop-env.sh on your favorite editor and add the following lines.

```
# The java implementation to use.  Required.
export JAVA_HOME=/usr/java/jdk1.7.0_21
```

Configuring $HADOOP_CONF_DIR/*-site.xml

In this section, we will configure the directory where Hadoop will store its data files, the network ports it listens to, etc. Our setup will use Hadoop's Distributed File System, HDFS, even though our little "cluster" only contains our single local machine.

You can leave the settings below "as is" with the exception of the hadoop.tmp.dir parameter – this parameter you must change to a directory of your choice. We will use the directory /app/hadoop/tmp in

this tutorial. Hadoop's default configurations use hadoop.tmp.dir as the base temporary directory both for the local file system and HDFS, so don't be surprised if you see Hadoop creating the specified directory automatically on HDFS at some later point.
Now we create the directory and set the required ownerships and permissions:

```
# Change user to root
su -
Password:
mkdir -p /app/hadoop/tmp
chown -R hduser:hadoop /app
```

If you forget to set the required ownerships and permissions, you will see a java.io.IOException when you try to format the name node in the next section).
Add the following snippets between the ... tags in the respective configuration XML file.
In file conf/core-site.xml:

```
<property>
<name>hadoop.tmp.dir</name>
<value>/app/hadoop/tmp</value>
<description>A base for other temporary directories.</description>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:54310</value>
<description>The name of the default file system.  A URI whose
scheme and authority determine the FileSystem implementation.  The
uri's scheme determines the config property (fs.SCHEME.impl) naming
the FileSystem implementation class.  The uri's authority is used to
determine the host, port, etc. for a filesystem.</description>
</property>
```

In file conf/hdfs-site.xml:

```
<property>
<name>dfs.replication</name>
<value>1</value>
<description>Default block replication.
The actual number of replications can be specified when the file is created.
The default is used if replication is not specified in create time.
</description>
</property>
```

In file conf/mapred-site.xml:

```
<property>
<name>mapred.job.tracker</name>
<value>localhost:54311</value>
<description>The host and port that the MapReduce job tracker runs
at.  If "local", then jobs are run in-process as a single map
and reduce task.
</description>
</property>
```

Formatting the HDFS filesystem via the NameNode

The first step to starting up your Hadoop installation is formatting the Hadoop filesystem which is implemented on top of the local filesystem of your "cluster" (which includes only your local machine if you followed this tutorial). You need to do this the first time you set up a Hadoop cluster.
Caution: Do not format a running Hadoop filesystem as you will lose all the data currently in the cluster (in HDFS)!
To format the filesystem (which simply initializes the directory specified by the dfs.name.dir variable), run the following command
/usr/local/hadoop/bin/hadoop namenode -format
Starting your single-node cluster
Run the command:

/usr/local/hadoop/bin/start-all.sh
This will startup a Namenode, SecondaryNamenode, Datanode, Jobtracker and a Tasktracker on your machine.
The output will look like this:

[hduser@Fuji ~]$ /usr/local/hadoop/bin/start-all.sh
starting namenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-namenode-Fuji.out
localhost: starting datanode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-datanode-Fuji.out
localhost: starting secondarynamenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-secondarynamenode-Fuji.out
starting jobtracker, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-jobtracker-Fuji.out
localhost: starting tasktracker, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-tasktracker-Fuji.out
A nifty tool for checking whether the expected Hadoop processes are running is jps.
[hduser@Fuji ~]$ jps
6794 NameNode
7362 TaskTracker
7107 SecondaryNameNode
7494 Jps
6938 DataNode
7205 JobTracker

Stopping your single-node cluster
Run the command

/usr/local/hadoop/bin/stop-all.sh
[hduser@Fuji ~]$ /usr/local/hadoop/bin/stop-all.sh
stopping jobtracker
localhost: stopping tasktracker
stopping namenode
localhost: stopping datanode
localhost: stopping secondarynamenode

In the coming posts we will see how to run a Map/Reduce program on the single node cluster and also learn to setup multi-node cluster.

12. Results:

[hduser@Fuji ~]$ /usr/local/hadoop/bin/start-all.sh
starting namenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-namenode-Fuji.out
localhost: starting datanode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-datanode-Fuji.out
localhost: starting secondarynamenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-secondarynamenode-Fuji.out
starting jobtracker, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-jobtracker-Fuji.out
localhost: starting tasktracker, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-tasktracker-Fuji.out
A nifty tool for checking whether the expected Hadoop processes are running is jps.
[hduser@Fuji ~]$ jps

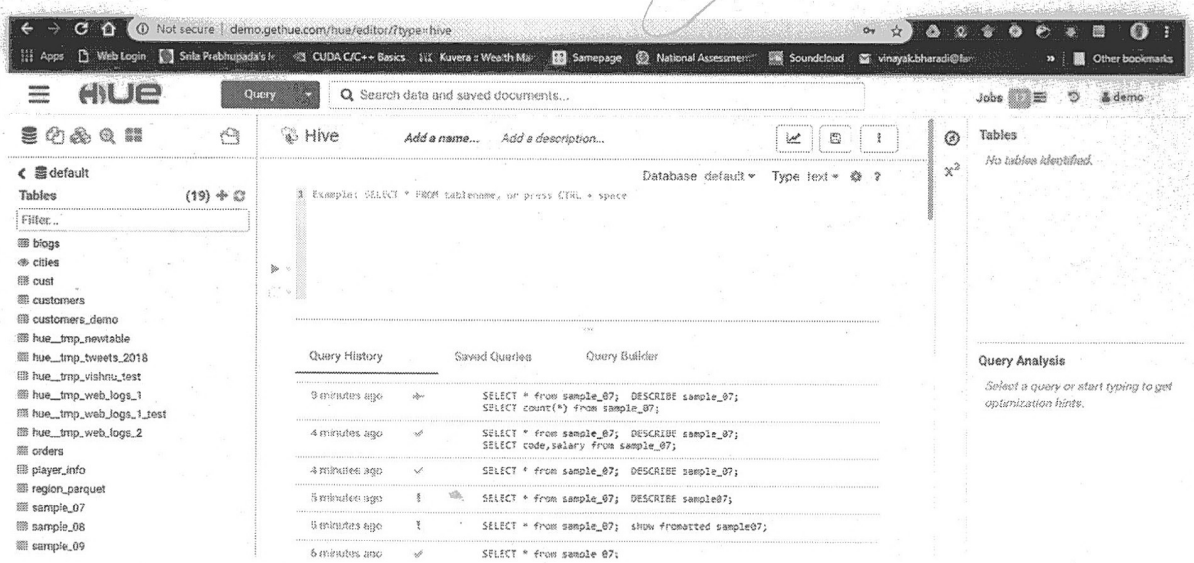
6794 NameNode

7362 TaskTracker

7107 SecondaryNameNode

7494 Jps

6938 DataNode



- Learning Outcomes Achieved:

i) Listing of Hadoop ecosystem components was done.

2) Hadoop was ~~done~~ explored to find the application of Hive, Pig, Hbase, etc.

3) Hue as a dashboard was tested and components were involved.

- Conclusion:

1) Applications of the studied technique in industry
   a. Hadoop is used for Big data analytics as a data storage and processing framework.
   b. parallel processing datasets on multiple nodes.

2) Engineering Relevance
   a. Hadoop Enables Analysis of Big data

3) Skills Developed
   a. HUE dashboard operation
   b. Identifying functionality of each component on Hadoop.

References :

[1] "Hadoop Releases". apache.org. Apache Software Foundation. Retrieved 2014-12-06.
[2] deRoos, Dirk. "Managing Files with the Hadoop File System Commands". dummies.com. For Dummies. Retrieved 21 June 2016.
[3] "Welcome to Apache Hadoop!". hadoop.apache.org. Retrieved 2016-08-25.
[4] Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. John Wiley & Sons. 2014-12-19. p. 300. ISBN 9781118876220. Retrieved 2015-01-29.
[5] Cutting, Mike; Cafarella, Ben; Lorica, Doug (2016-03-31). "The next 10 years of Apache Hadoop". O'Reilly Media. Retrieved 2017-10-12.

Viva Questions

4. What is a Big Data?
5. What is command line?
6. How to change a user in Linux?
7. What is YARN?
8. How Hadoop is processing Big Data?

Teachers Interaction with Students

1. Briefing about installation of Hadoop on linux systems
2. Discussion about Hadoop ecosystem.
3. Discussion about installation steps
4. Discussion about Hadoop functionality