

FINAL REPORT

On

On Demand Automation & Analysis of Hadoop Cluster

Submitted by:

Ujjawal Gupta	CSE-CCVT B4	R110216167
Shubham Gupta	CSE-CCVT B4	R110216153
Sakshi Gupta	CSE-CCVT B3	R110216135

Under the guidance of

Mr. Pravin Dagdee

**Industry Fellow,
School of Computer Science**



**Department of Virtualization,
School of Computer Science and Engineering,
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
Dehradun-248007**

CANDIDATE'S DECLARATION

I/We hereby certify that the project work entitled “ **On Demand Automation & Analysis of Hadoop Cluster**” in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Virtualization and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **January, 2019** to **May, 2019** under the supervision of **Mr. Pravin Dagdee, Industrial Fellow, UPES**.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Ujjawal Gupta (167)
Shubham Gupta (153)
Sakshi Gupta (135)

This is to certify that the above statement made by the candidates, is correct to the best of my knowledge.

Date: 22/05/2019

Mr. Pravin Dagdee
Mentor

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Mr. Pravin Dagdee**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our Head of Department, **Dr. Deepshika Bhargava**, for her great support in doing our **On Demand Automation & Analysis of Hadoop Cluster** in Cloud Computing and Virtualization at SoCS.

We are also grateful to Dean of SoCS, **Dr. Manish Prateek Professor** and Dean of UPES, **Dr. Kamal Bansal** for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Name:	Shubham Gupta	Ujjawal Gupta	Sakshi Gupta
Roll No.:	R110216153	R110216167	R110216135



Certificate of Completion

This certification is hereby bestowed upon

Ujjawal Gupta , **Shubham Gupta** & **Sakshi Gupta**

For the exceptional performance that has led to the successful completion of
The Project - “**On Demand Automation and Analysis of Hadoop Cluster**”
conducted at **University of Petroleum and Energy Studies** between
21st January 2019 to 22nd May, 2019.

This activity was awarded by,

Mr. Pravin Dadgee
(Project Mentor)

ABSTRACT

The aim of this project is to implement Hadoop cluster to make the relevant use of storage. The storage that is distributed along various physical systems can be integrated by using computational cluster designed especially for storing and analyzing huge amount of unstructured data in distributed computing environment. Hadoop cluster is well suited to analyzing Big Data i.e. widely distributed and largely unstructured data. Through the means of this project, we will be using Python programming language to build Hadoop cluster and achieve the above mentioned.



Table of Content

S.NO.	TITLE	PAGE NO.
1.	Introduction	8-9
2.	Problem Statement	9
3.	Literature Review	10
4.	Objective	10
5.	Methodology	11
6.	Flow Chart	12
7.	Data Flow Diagram	13
8.	Use Case	14
9.	Algorithm	15-16
10.	Result - OUTPUT Snaps	17-22
11.	Pert Chart	23
12.	Future Enhancements	23
13.	Conclusion	24
14.	References	24

List of Figures

S.NO.	TITLE	PAGE NO.
0.	Logo – Hadoop	5
1.	Features of Hadoop and its importance	9
2.	Components of Hadoop Cluster	11
3.	Flow of Control in our System	12
4.	Flow of data in HDFS	13
5.	Flow of data in MapReducer	13
6.	USE Cases - Shows the Use Cases of our system	14
7.	PERT Chart - Shows the Timestamp for each stage of production	23



Final Report

Project Title:

On Demand Automation & Analysis of Hadoop Cluster

Introduction:

In fast-paced and hyper-connected world where more and more data is being created, big data analytics has become all the rage. Now it has become a tedious task to store such a large data set as the data generated is in various formats and its processing possesses new challenges. As Big Data being unstructured and growing rapidly, Hadoop is required to put the right Big Data workloads in the right systems and optimize data management structure. Hadoop can store and distribute very large data sets across hundreds of underused servers that operate in parallel. Hadoop technology is used in software known as Hadoop cluster to utilize storage.

A Hadoop cluster is a special type of computational cluster designed specifically for storing and analyzing huge amounts of data in a distributed computing environment. To boosting the speed of data analysis applications, Hadoop cluster are often referred to as "shared nothing" systems because the only thing that is shared between nodes is the network that connects them.

The Hadoop cluster consists of a group of nodes, which are processes running on either a physical or virtual machine in parallel. Typically, one machine in the cluster is designated as the Name node and Job Tracker; these are the masters. The rest of the machines in the cluster act as both Data Node and Task Tracker; these are the slaves.

JobTracker and TaskTracker are two important processes involved in the Hadoop computation cluster in MRv1(Hadoop version1). JobTracker is a service within Hadoop that takes client requests and assigns them to TaskTrackers that are within the DataNodes of the cluster, here the data is locally present. JobTracker selects the TaskTracker containing the data or at least are near the data where the task is to be submitted.

Both JobTracker and TaskTracker are now deprecated in the MRv2 (or Hadoop version 2), they are replaced by Application Master, Resource Manager and Node Manager Daemons.



Figure no.: 1, Features of Hadoop and its importance

To achieve all the above mentioned in this project, we are implementing the complete process of data storage within a network using Hadoop cluster by using python language to automate & analysis it. The inner workings of all the relevant processes involved, will be done using Linux on Master OS. The project will enable us to understand the use of python programming language, concepts of data storage and how Hadoop cluster are configured and automated.

Problem Statement:

In traditional databases when it comes to storing large amount of data i.e. Big Data, cost of the software is the main constraint. As data management technologies often store multiple copies of same data on different system, the total cost might be more like \$30000 to \$40000, per terabyte. When we deal with Petabytes of data and manage multiple copies of it as there are very high chances of data loss if there occurs a system failure, cost further increases. Also manual configuration and setup of Hadoop Cluster is complex and time consuming process which include redundant work that can be automated.

Literature Review:

- ✓ In a research paper by *Samee Ullah Khan*, “**The rise of Big data on cloud computing**” the author introduces to the amount of data continues to increase at an exponential rate, cloud computing and big data are conjoined, only a few tools are available to address the issues of big data processing in cloud, open research issues that require substantial research efforts are summarized.[1]
- ✓ In a research paper by *Konstantin Shvachko, Hairong Kuang, Sanjay Radia and Robert Chansler*, “**The Hadoop Distributed file System**”, The authors discuss the detailed overview of developing a Hadoop Distributed File System (HDFS) to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. Where thousands of servers both host directly attached storage and execute user application tasks. [2]
- ✓ On Studying the Research Paper by *Guanghui Xu and Hongxu Ma*, “**Deploying and researching Hadoop in Virtual machines**”, we understood that this paper firstly introduces some technologies used such as Cloud Stack, Map Reduce and Hadoop. Based on that, a method to deploy Cloud Stack is given. Then it discusses how to deploy Hadoop in virtual machines which can be obtained from Cloud Stack. [3]
- ✓ In a research paper by *Zhang Shao-min, LI Xiao-qiang and WANG Bao-yi*, “**Design of data storage in smart grid based on Hadoop**”, the paper analyzes the characteristics of the existing security storage solutions, then combines with the special occasions in smart grid, and finally designs a security storage solution based on Hadoop. [4]

Objective:

To implement Hadoop cluster to make the relevant use of storage by integrating storage of various physical systems. Store huge amount of data given from the Client to Master system in various slave nodes.

Sub Objectives: -

- Build multi-node Hadoop cluster for distributing massive amount of data.
- Using Python programming language, design an automated solution to reduce the time for manual configuration.

Methodology:

The primary purpose of this project is to provide “Hadoop cluster” to the client according to the demand or amount of data which is to be stored on the distributed file system. Also we will further analyses the cluster & will try to overcome various drawbacks of Hadoop cluster such as: single point failure which results in the data loss etc.

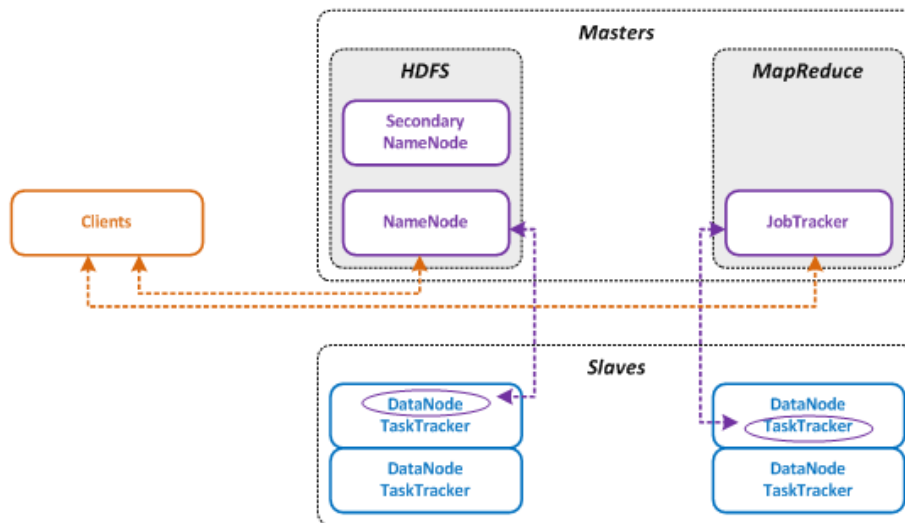


Figure no.: - 2, Components of Hadoop Cluster

Various STEPs involved in the complete procedure:

1. A python script which is to be run on the client side.
2. Client provide the information according to his/her requirements like- number of slaves, block size, no. of replica's, etc.
3. As every system has different specification & configuration so each system will be analysed.
4. The systems which have maximum storage available on it will be configured as a slave.
5. Start the service of Name node (master) & Data nodes (slaves).
6. Check the services & active data nodes on the Hadoop.
7. Hadoop storage cluster is ready to use by the client.
8. Now, we can also setup a Hadoop compute cluster for client.
9. Every Name node (master) can be used as a Job tracker, it will map all the Jobs that are assigned to their respective slave machines.
10. Every Data node (slave) can be used as a Task Tracker, where the tasks are computed.

Flow Chart

This flow chart shows all the steps that are involved in the Hadoopv1.py (Source Code) file.

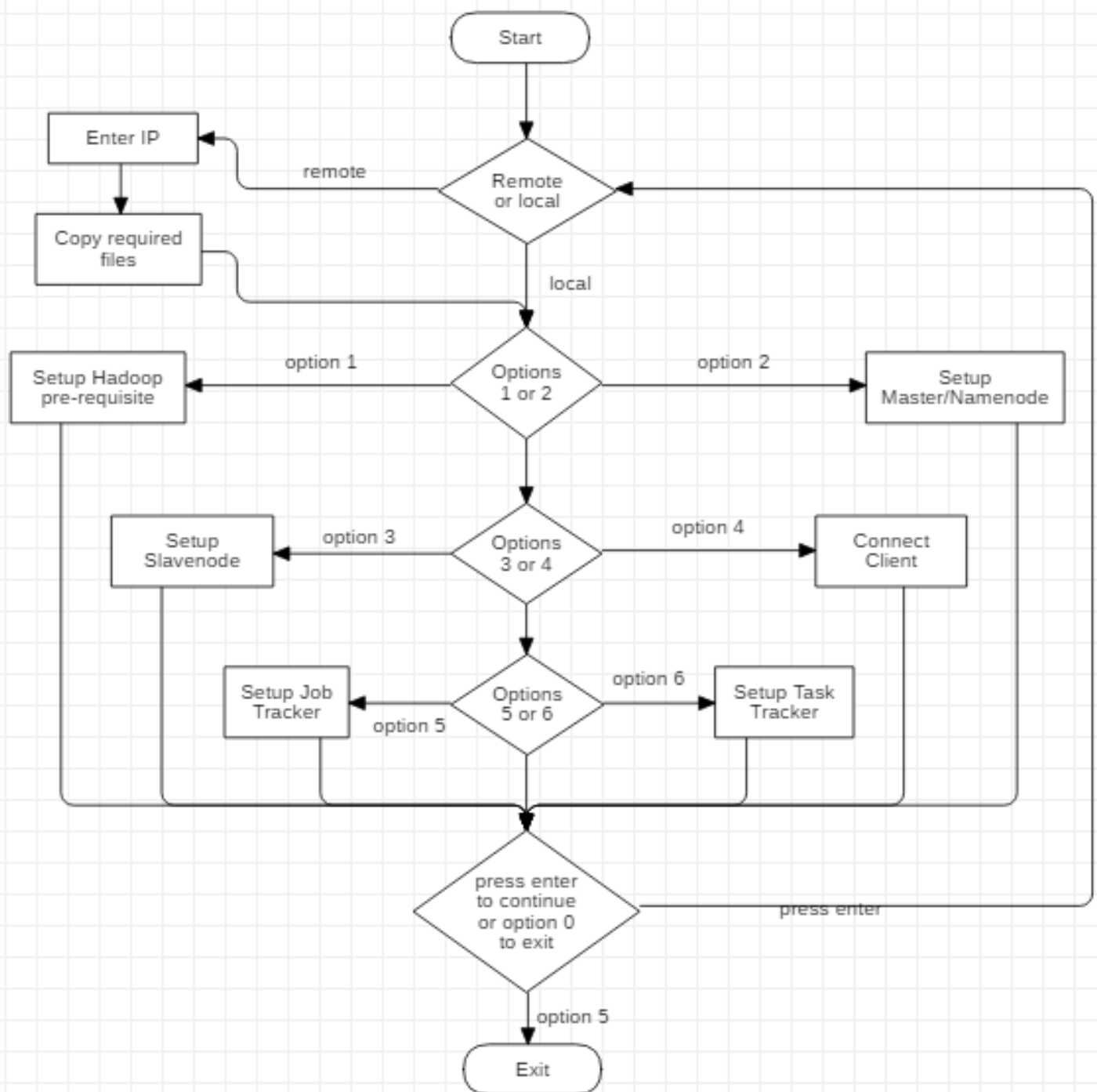


Figure no. – 3, Flow of Control in our System

Data Flow Diagram

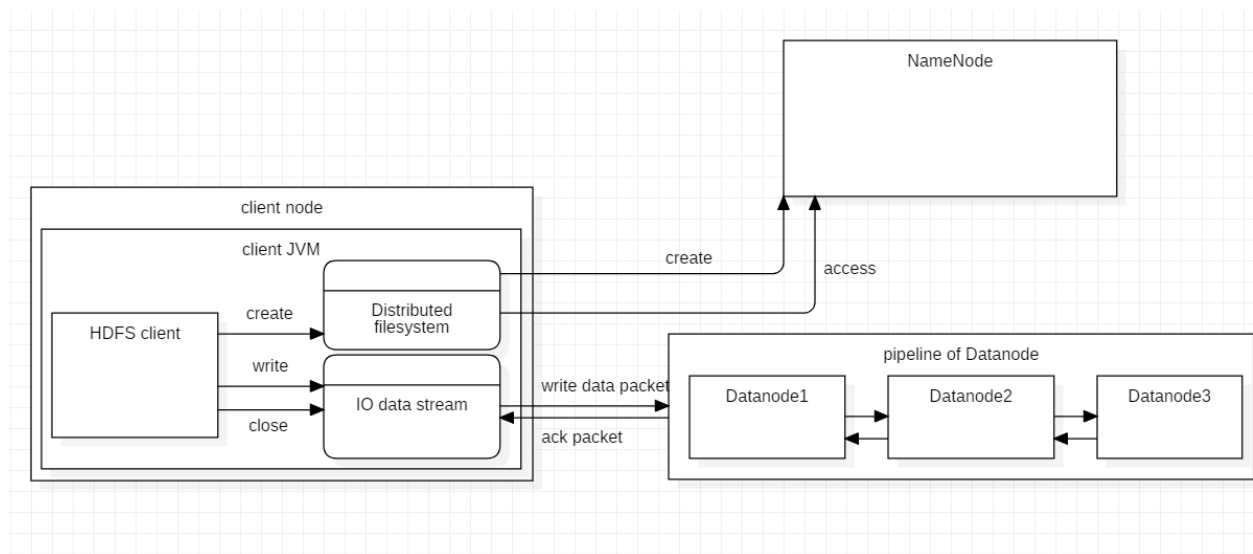


Figure no.- 4 Flow of data in HDFS

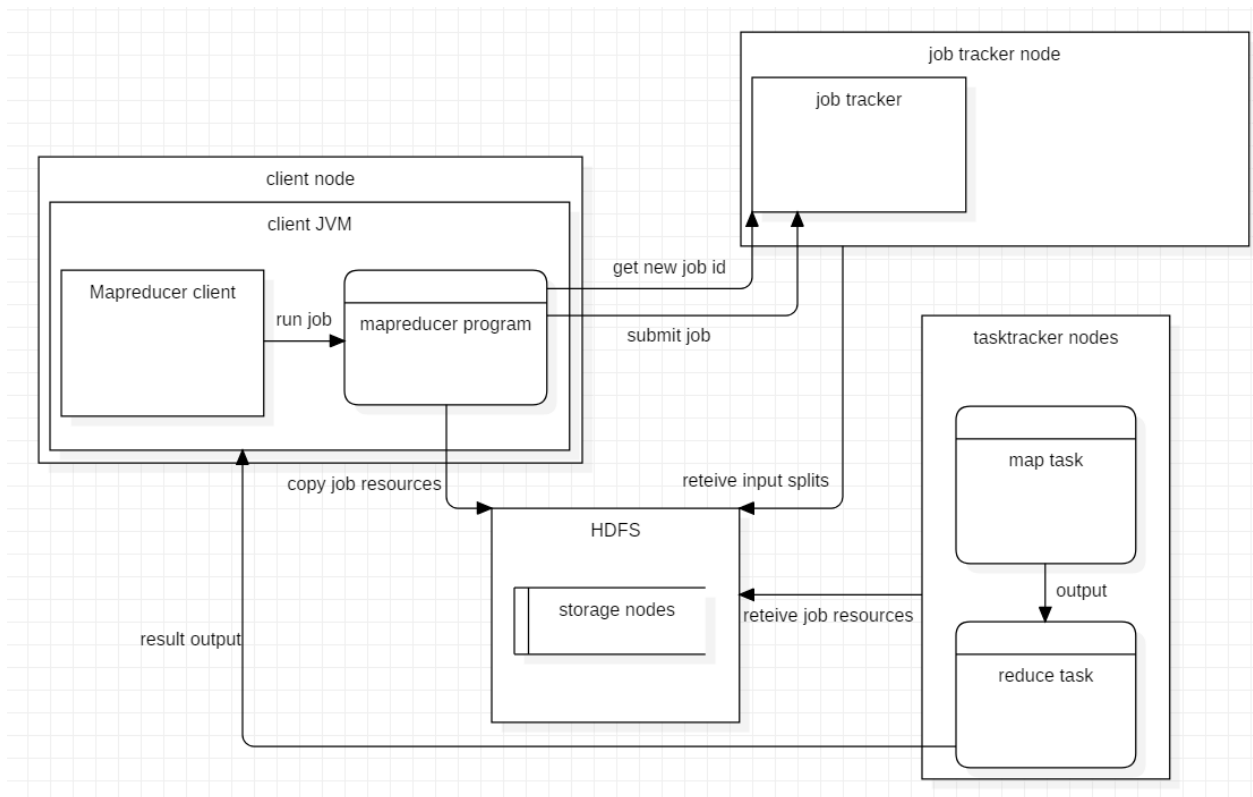


Figure no.- 5 Flow of Data in MapReduce

Use Case Diagram

There are two actors, client and server that accesses the functionalities of different machines such as Master node and slave nodes. It also shows some of the components of these machines.

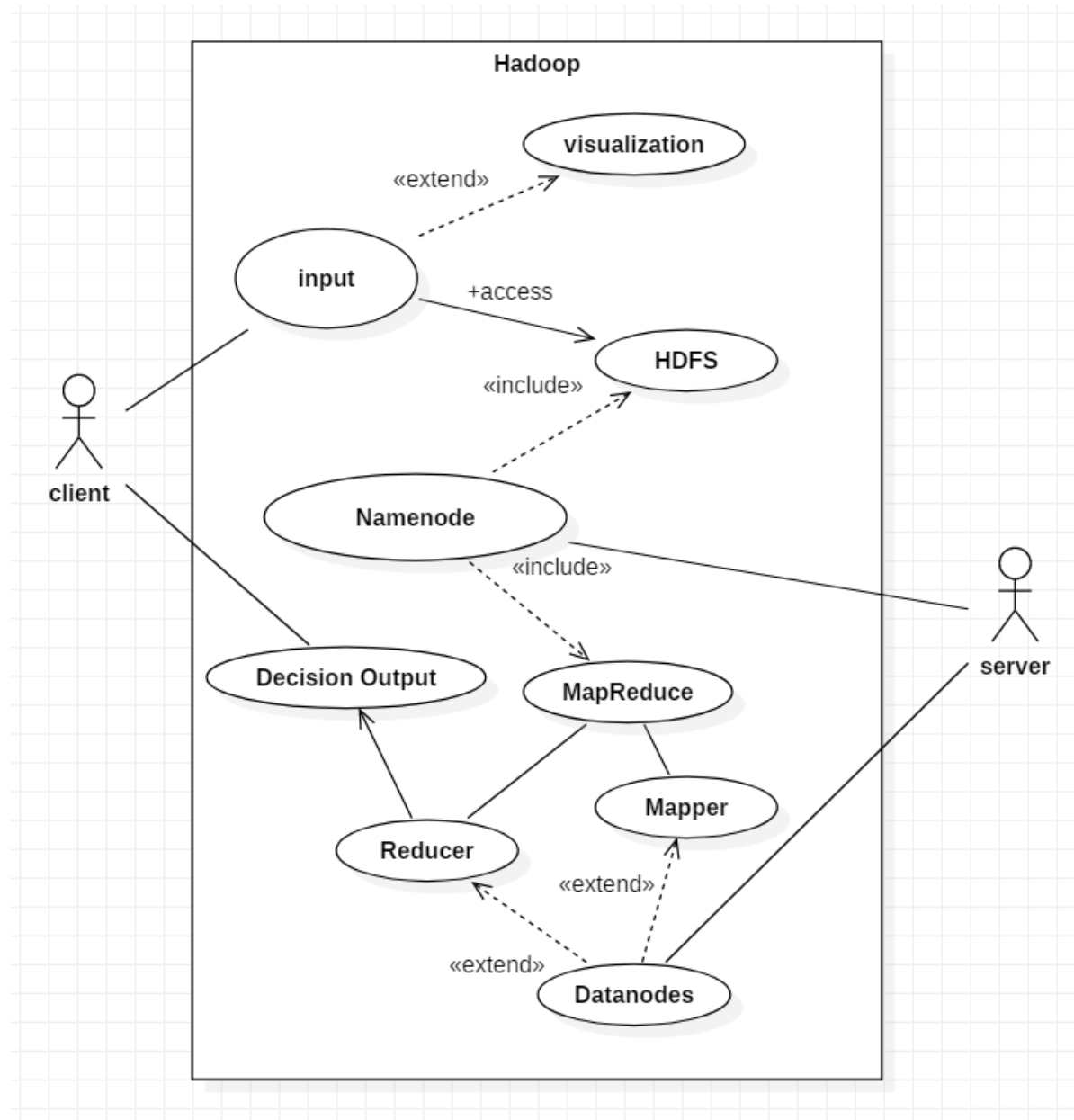


Figure no. -6, shows the Use Cases of our system

Algorithm:

- 1) Start
- 2) user will be asked for local or remote access
- 3) If user chooses LOCAL mode, then following options
 - a) Setup prerequisite to setup hadoop
 - b) Setup master node
 - c) Setup slave node
 - d) Setup client node
 - e) Setup Job Tracker
 - f) Setup Task Tracker
 - g) Exit (0)
- 4) Else user chooses REMOTE mode then above options will be shown
 - a) And user have to enter the IP and password of remote system
- 5) If user chooses Option 1 run case 1.
- 6) Else if user chooses Option 2 run case 2.
- 7) Else if user chooses Option 3 run case 3.
- 8) Else if user chooses Option 4 run case 4.
- 9) Else if user chooses Option 5 run case 5.
- 10) Else if user chooses Option 6 run case 6.
- 11) Else user chooses Option 0, exit from the program.
- 12) END

CASE 1:

Enter the IP address and password of remote machine

- a) If IP and password is correct using SSH command a directory will be created on remote system containing
 - (1) Hadoop installer (this software will be installed)
 - (2) Java development kit (this software will be installed)
 - (3) Hadoop setup python script (Our Source Code)
- b) Else the password or IP is incorrect

CASE 2:

Enter the IP address and password of remote machine

- 1) If IP and password are correct
 - a) SSH the master-setup.py file onto the remote system.
 - b) Execute the master-setup.py file on remote system.
 - c) Configuration of hdfs-site.xml & core-site.xml file of hadoop.
 - d) Namenode is starts on remote system.
- 2) Else the password or IP is incorrect.

CASE 3:

Enter the IP address and password of remote machine

- 1) If IP and password are correct
 - a) SSH the slave-setup.py file onto the remote system.
 - b) Execute the slave-setup.py file on remote system.
 - c) Configuration of hdfs-site.xml & core-site.xml file of Hadoop.
 - d) Datanode is starts on remote system.
- 2) Else the password or IP is incorrect.

CASE 4:

Enter the IP address and password of remote machine

- 1) If IP and password are correct
 - a) SSH the client-setup.py file onto the remote system.
 - b) Execute the client-setup.py file on remote system.
 - c) Configuration of hdfs-site.xml & core-site.xml file of hadoop.
 - d) Client is active & can access the hadoop distributed storage.
- 2) Else the password or IP is incorrect.

CASE 5:

Enter the IP address and password of remote machine

- 1) If IP and password are correct
 - a) SSH the job-setup.py file onto the remote system.
 - b) Execute the job-setup.py file on remote system.
 - c) Configuration of mapred-site.xml & core-site.xml file of hadoop.
 - d) Job Tracker is active on remote system (Master node).
- 2) Else the password or IP

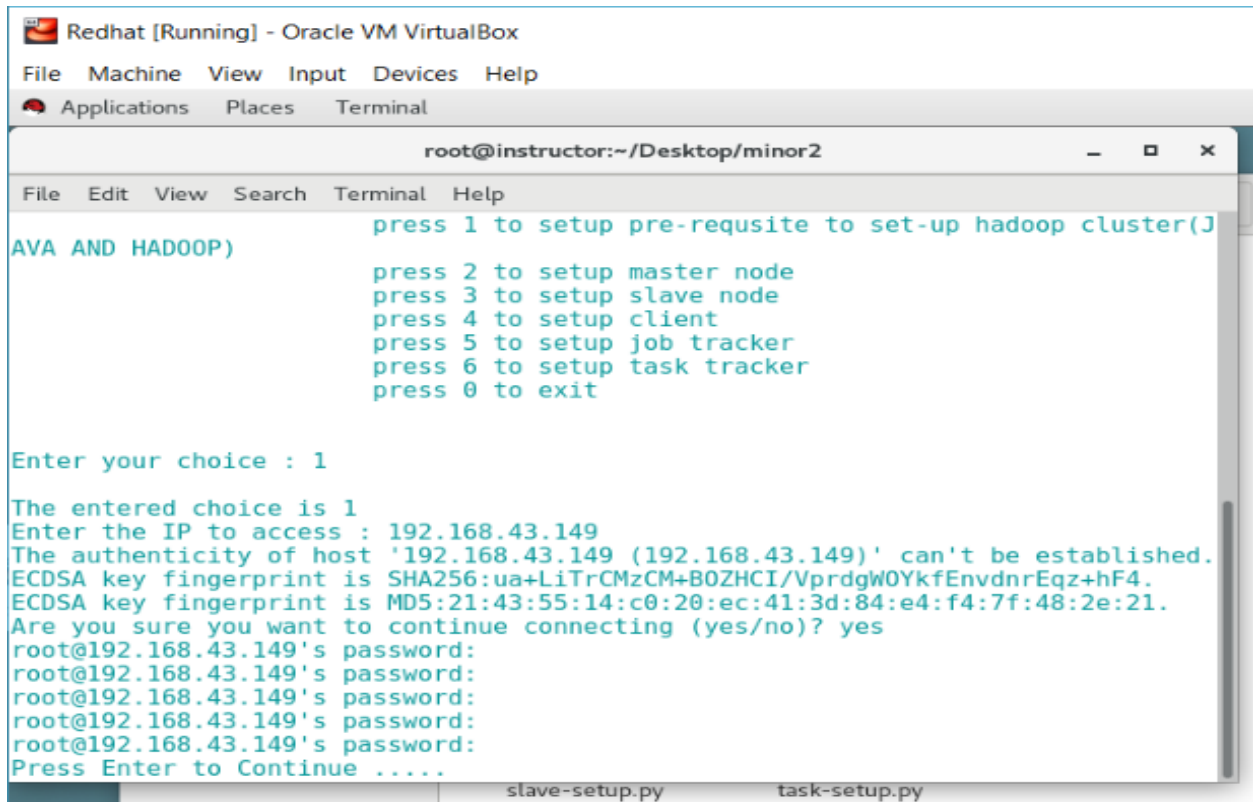
CASE 6:

Enter the IP address and password of remote machine

- 3) If IP and password are correct
 - a) SSH the task-setup.py file onto the remote system.
 - b) Execute the task-setup.py file on remote system.
 - c) Configuration of mapred-site.xml & core-site.xml file of hadoop.
 - d) Task Tracker is active on remote system (Slave node).
- 4) Else the password or IP

OUTPUT Snaps:

Screen 1: shows initial window where we setup Hadoop Cluster Tool on remote system



```
Redhat [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal

root@instructor:~/Desktop/minor2
File Edit View Search Terminal Help

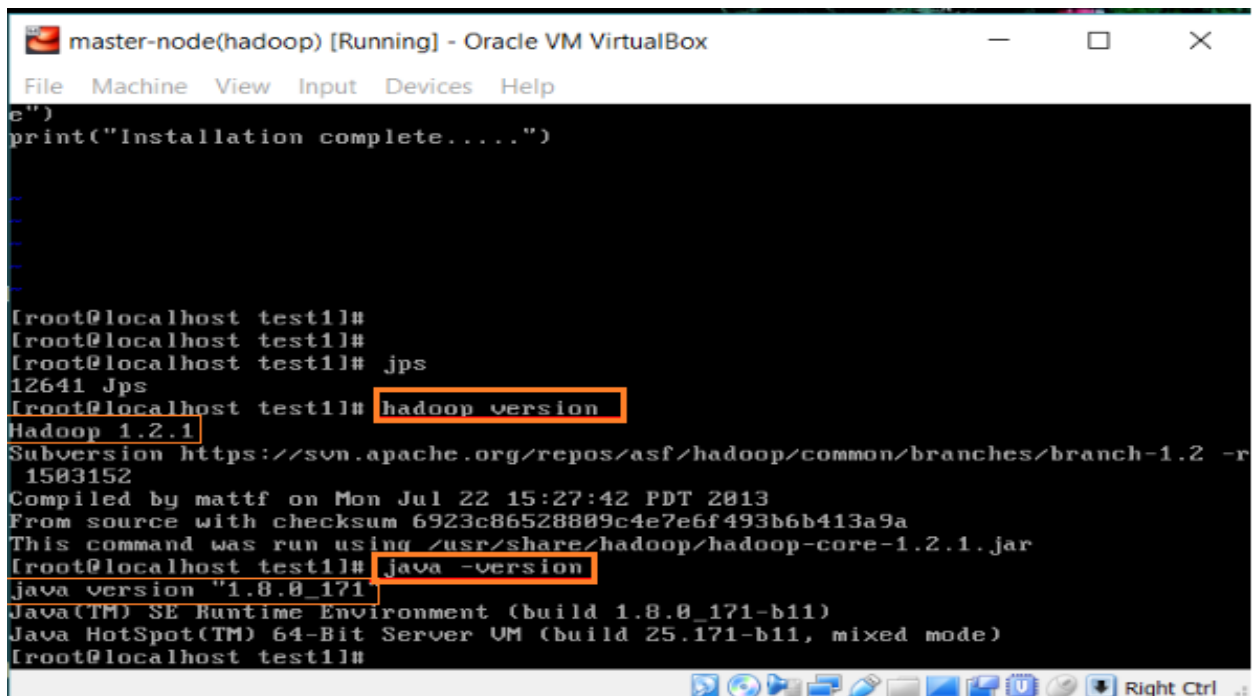
AVA AND HADOOP)
    press 1 to setup pre-requisite to set-up hadoop cluster(J
    press 2 to setup master node
    press 3 to setup slave node
    press 4 to setup client
    press 5 to setup job tracker
    press 6 to setup task tracker
    press 0 to exit

Enter your choice : 1

The entered choice is 1
Enter the IP to access : 192.168.43.149
The authenticity of host '192.168.43.149 (192.168.43.149)' can't be established.
ECDSA key fingerprint is SHA256:ua+LiTrCMzCM+BOZHCI/VprdgWOYkfEnvdnrEqz+hF4.
ECDSA key fingerprint is MD5:21:43:55:14:c0:20:ec:41:3d:84:e4:f4:7f:48:2e:21.
Are you sure you want to continue connecting (yes/no)? yes
root@192.168.43.149's password:
root@192.168.43.149's password:
root@192.168.43.149's password:
root@192.168.43.149's password:
root@192.168.43.149's password:
Press Enter to Continue .....

slave-setup.py task-setup.py
```

Screen 2: Shows the Hadoop pre-requisite setup on MasterNode



```
master-node(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

e")
print("Installation complete.....")

[root@localhost test11#
[root@localhost test11#
[root@localhost test11# jps
12641 Jps
[root@localhost test11# hadoop version
Hadoop 1.2.1
Subversion https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.2 -r
1503152
Compiled by mattf on Mon Jul 22 15:27:42 PDT 2013
From source with checksum 6923c86528809c4e7e6f493b6b413a9a
This command was run using /usr/share/hadoop/hadoop-core-1.2.1.jar
[root@localhost test11# java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
[root@localhost test11#
```

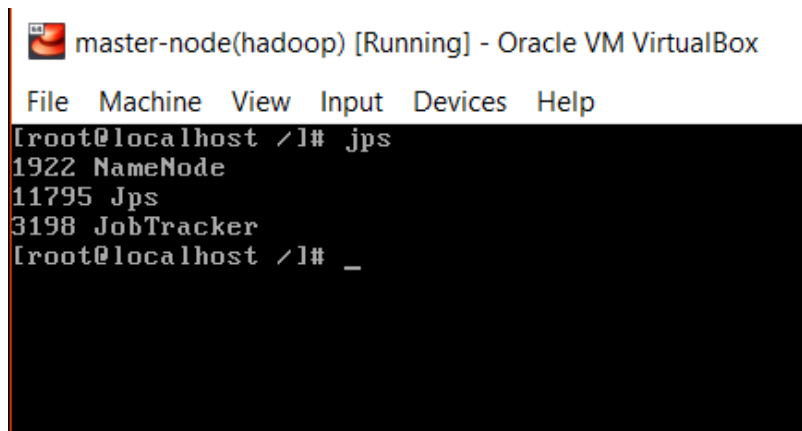
Screen 3: Similarly Hadoop is setup on other 6 different VMs

The screenshot displays six Oracle VM VirtualBox windows, each showing the terminal output of Hadoop setup on a different node. The windows are titled: master-node(hadoop), f-node1(hadoop), data-node2(hadoop), task-tracker(hadoop), task-tracker2(hadoop), and client(hadoop). The terminal outputs show the installation of the Internet Systems Consortium DHCP Client 4.2.5, the configuration of the network interface, and the successful completion of the Hadoop setup process. The outputs also show the Hadoop version (1.2.1) and the location of the Hadoop binaries.

Screen 4: shows the setup of NameNode(Master)

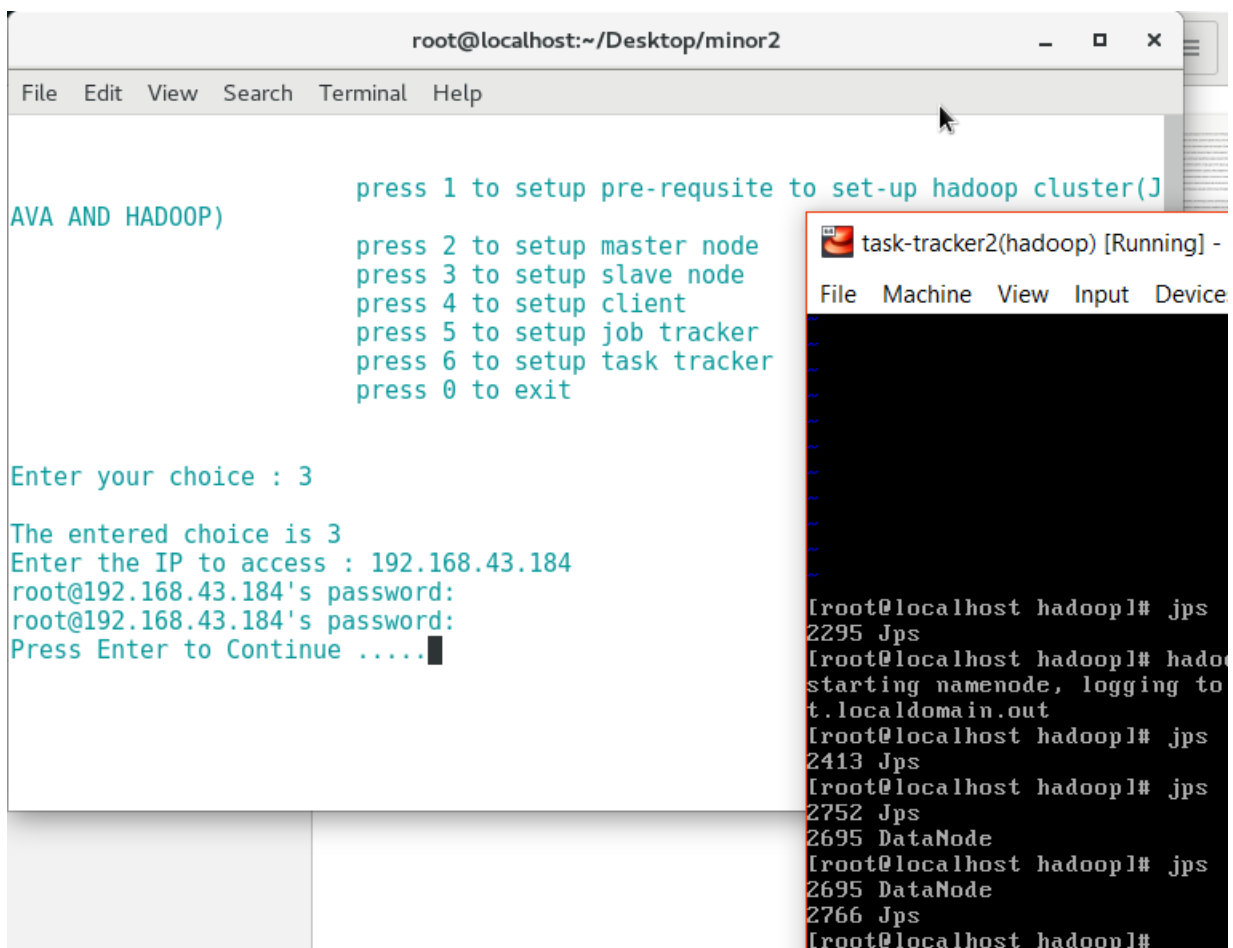
The screenshot shows a terminal window titled "root@localhost:~/Desktop/minor2". The terminal displays a series of instructions for setting up Hadoop, including pressing 1 to setup pre-requisite, 2 to setup master node, 3 to setup slave node, 4 to setup client, 5 to setup job tracker, 6 to setup task tracker, and 0 to exit. The user enters the choice "2". The terminal then displays the IP address "192.168.43.184" and the password "root@192.168.43.184's password:". The user enters the password, and the terminal displays "Press Enter to Continue".

Screen 5: Shows the MasterNode that comprises NameNode and JobTracker.



```
master-node(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[root@localhost ~]# jps
1922 NameNode
11795 Jps
3198 JobTracker
[root@localhost ~]# _
```

Screen 6: Shows the setup of DataNode



```
root@localhost:~/Desktop/minor2
File Edit View Search Terminal Help

AVA AND HADOOP)      press 1 to setup pre-requisite to set-up hadoop cluster(J
                      press 2 to setup master node
                      press 3 to setup slave node
                      press 4 to setup client
                      press 5 to setup job tracker
                      press 6 to setup task tracker
                      press 0 to exit

Enter your choice : 3

The entered choice is 3
Enter the IP to access : 192.168.43.184
root@192.168.43.184's password:
root@192.168.43.184's password:
Press Enter to Continue .....

task-tracker2(hadoop) [Running] -
File Machine View Input Device

[task-tracker2@localhost ~]$ jps
2295 Jps
[task-tracker2@localhost ~]$ hadoop
starting namenode, logging to
t.localdomain.out
[task-tracker2@localhost ~]$ jps
2413 Jps
[task-tracker2@localhost ~]$ jps
2752 Jps
2695 DataNode
[task-tracker2@localhost ~]$ jps
2695 DataNode
2766 Jps
[task-tracker2@localhost ~]$
```

Screen 7: Shows report of two DataNodes are in Active state in cluster

```
master-node(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Datanodes available: 2 (2 total, 0 dead)

Name: 192.168.43.218:50010
Decommission Status : Normal
Configured Capacity: 28968488960 (26.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 1992298496 (1.86 GB)
DFS Remaining: 26976182272(25.12 GB)
DFS Used%: 0%
DFS Remaining%: 93.12%
Last contact: Tue May 21 23:30:41 IST 2019

Name: 192.168.43.93:50010
Decommission Status : Normal
Configured Capacity: 28968488960 (26.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 1991217152 (1.85 GB)
DFS Remaining: 26977263616(25.12 GB)
DFS Used%: 0%
DFS Remaining%: 93.13%
Last contact: Tue May 21 23:30:40 IST 2019

[root@localhost ~]#
```

Screen 8: Complete Hadoop Cluster is setup and in Active state

```
master-node(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Datanodes available: 2 (2 total, 0 dead)

Name: 192.168.43.218:50010
Decommission Status : Normal
Configured Capacity: 28968488960 (26.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 1992298496 (1.86 GB)
DFS Remaining: 26976182272(25.12 GB)
DFS Used%: 0%
DFS Remaining%: 93.12%
Last contact: Tue May 21 23:30:41 IST 2019

Name: 192.168.43.93:50010
Decommission Status : Normal
Configured Capacity: 28968488960 (26.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 1991217152 (1.85 GB)
DFS Remaining: 26977263616(25.12 GB)
DFS Used%: 0%
DFS Remaining%: 93.13%
Last contact: Tue May 21 23:30:40 IST 2019

[root@localhost ~]#

data-node2(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
DFS Remaining: 26977282176 (25.12 GB)
DFS Used: 8192 (8 KB)
DFS Used%: 0%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

Datanodes available: 1 (1 total, 0 dead)

Name: 192.168.43.93:50010
Decommission Status : Normal
Configured Capacity: 28968488960 (26.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 1991278592 (1.85 GB)
DFS Remaining: 26977282176(25.12 GB)
DFS Used%: 0%
DFS Remaining%: 93.13%
Last contact: Tue May 21 23:27:57 IST 2019

[root@localhost hadoop]# jps
2640 DataNode
2932 Jps
[root@localhost hadoop]#

task-tracker1(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

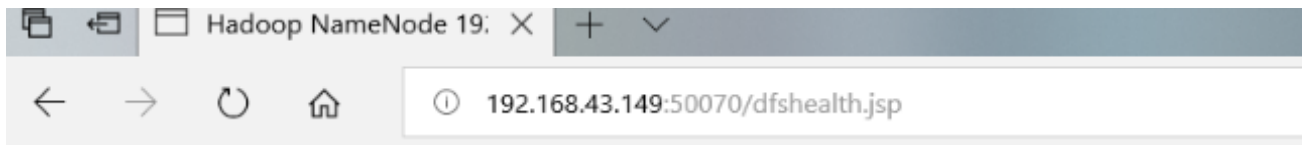
[root@localhost hadoop]# jps
1552 Jps
1490 TaskTracker
[root@localhost hadoop]# jps
1490 TaskTracker
1565 Jps
[root@localhost hadoop]# jps
1490 TaskTracker
1579 Jps

data-node1(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Decommission Status : Normal
Configured Capacity: 28968488960 (26.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 1992298496 (1.86 GB)
DFS Remaining: 26976182272(25.12 GB)
DFS Used%: 0%
DFS Remaining%: 93.12%
Last contact: Tue May 21 23:29:26 IST 2019

Name: 192.168.43.93:50010
Decommission Status : Normal
Configured Capacity: 28968488960 (26.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 1991217152 (1.85 GB)
DFS Remaining: 26977263616(25.12 GB)
DFS Used%: 0%
DFS Remaining%: 93.13%
Last contact: Tue May 21 23:29:28 IST 2019

[root@localhost hadoop]# jps
2917 Jps
2742 DataNode
```

Screen 9: Portal the shows the configuration of our Hadoop storage cluster.



NameNode '192.168.43.149:9001'

Started: Tue May 21 23:14:40 IST 2019
Version: 1.2.1, r1503152
Compiled: Mon Jul 22 15:27:42 PDT 2013 by mattf
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)

Cluster Summary

1 files and directories, 0 blocks = 1 total. Heap Size is 59.5 MB / 114 MB (52%)

Configured Capacity	:	53.96 GB
DFS Used	:	16 KB
Non DFS Used	:	3.71 GB
DFS Remaining	:	50.25 GB
DFS Used%	:	0 %
DFS Remaining%	:	93.12 %

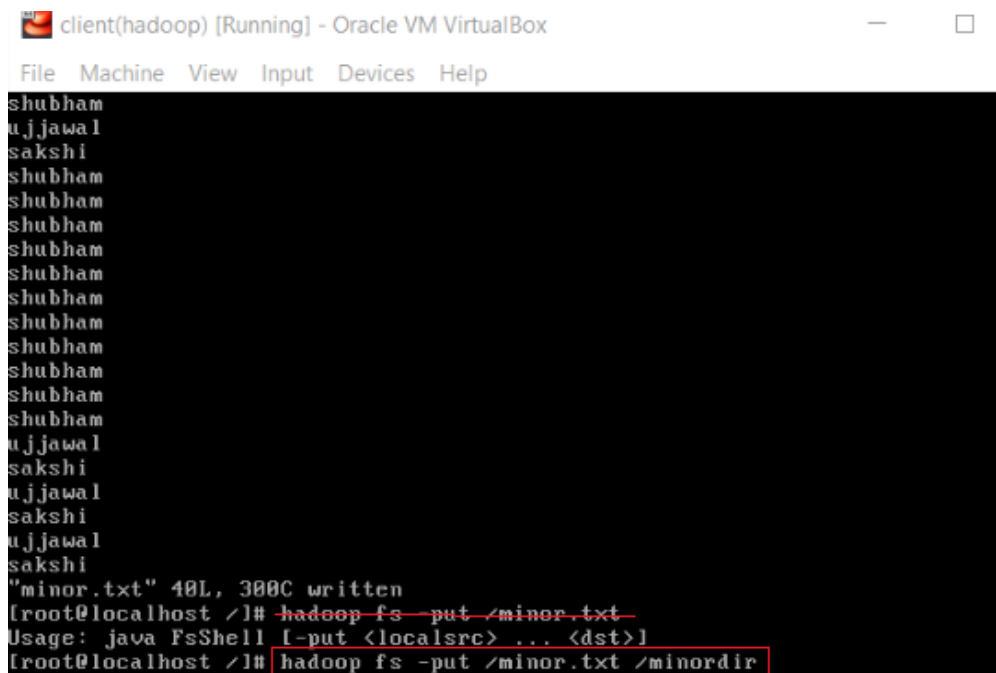
Live Nodes	:	2
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

NameNode Storage:

Storage Directory	Type	State
/mydata	IMAGE_AND_EDITS	Active

This is [Apache Hadoop](#) release 1.2.1

Screen 10: Text file is created on client and Uploaded on Hadoop cluster



```
client(hadoop) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
shubham
ujjawal
sakshi
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
ujjawal
sakshi
ujjawal
sakshi
ujjawal
sakshi
"minor.txt" 40L, 300C written
[root@localhost ~]# hadoop fs -put /minor.txt /minordir
Usage: java FsShell [-put <localsrc> ... <dst>]
```

Screen 11: Portal shows the uploaded text file.

File: [/minordir/minor.txt](#)

Goto :

[Go back to dir listing](#)

[Advanced view/download options](#)

shubham
ujjawal
sakshi

shubham
ujjawal
sakshi
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
shubham
ujjawal
sakshi
ujjawal
sakshi
ujjawal
sakshi
ujjawal

Schedule (Pert Chart):

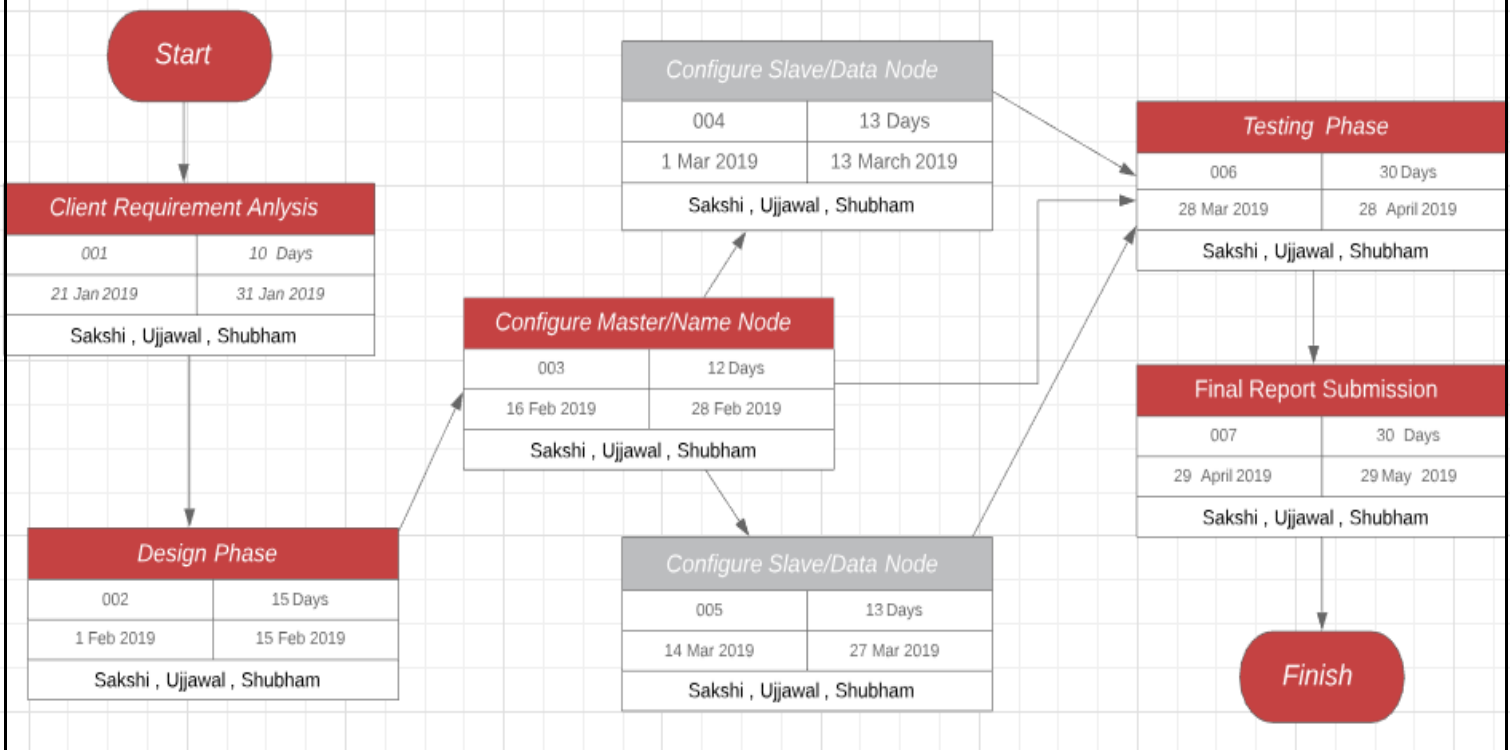


Figure no. – 7, Shows the Timestamp for each stage of production

Future Enhancement:

- 1) **HDFS Transparent Encryption:** HDFS encryption implements transparent, end-to-end encryption between HDFS slave and master blocks across the cluster. *End-to-end* means that data is encrypted at-rest and in-transit and *transparent* means that end-users are unaware of the encryption/decryption processes.
- 2) **HDFS Erasure Coding(EC)** is 3x replication factor which simply protect our data even in the failure of Datanode but needs too much extra storage.
- 3) **Introducing RAID Parity in NameNode:** NameNode (master) is a *single-point-of-failure* in HDFS, it requires a more reliable hardware setup. Therefore, the use of RAID is recommended on namenodes.
- 4) **Scheduling using JBOD:** JBOD (Just a Bunch of Disks) configuration used by HDFS, which round-robins HDFS blocks between all disks. If a disk fails in a JBOD configuration, HDFS can continue to operate without the failed disk.
- 5) **Monitoring & Analyzing Android Application:** Users can get an android app which can be used to monitor the status of their Hadoop cluster. Also, it can analyses the health and efficiency of their cluster.

Conclusion:

We have successfully implemented Hadoop storage cluster that comprises of a NameNode – that contains the metadata of the data stored at different DataNodes and various DataNode – that physically stores the data of the client. With these we dealt with the storage problem of Big data using Hadoop cluster automation.

Also, we have successfully implemented Hadoop computation cluster that comprises of JobTracker – that controls and maintain logs of the assignment of tasks to various DataNodes and TaskTracker – present at the DataNode that uses the compute power of these DataNodes to solve a complex problem that requires a huge computation power to be solved.

Hence, we gave an efficient solution to optimize the usage of compute power and storage of all the NameNode and DataNode machines that are a part of Hadoop Cluster.

References :

[1] Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of “big data” on cloud computing: Review and open research issues. *Information systems*, 47, 98-115.
Available here : <https://www.sciencedirect.com/science/article/abs/pii/S0306437914001288>

[2] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. THE HADOOP DISTRIBUTED FILE SYSTEM.
Available here : <https://ieeexplore.ieee.org/abstract/document/5496972/>

[3] Xu, G., Xu, F., & Ma, H. (2012, August). Deploying and researching Hadoop in virtual machines. In *2012 IEEE International Conference on Automation and Logistics* (pp. 395-399). IEEE.
Available here : <https://ieeexplore.ieee.org/abstract/document/6308241>

[4] ZHANG, Shao-min, Xiao-qiang LI, and Bao-yi WANG. "Design of data security storage in smart grid based on Hadoop [J]." *Power System Protection and Control* 14 (2013): 136-140.
Available here : http://en.cnki.com.cn/Article_en/CJFDTOTAL-JDQW201314025.htm

Report verified by:

Mr. Pravin Dagdee
(Project Guide)