PATTERN AND ANOMALY RECOGNITION IN INDUSTRIAL SCENARIOS

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Yash Agarwal (131230)

Pushpak Passey (131278)

Under the supervision of

Dr. Hemraj Saini

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat,

Solan-173234, Himachal Pradesh

CERTIFICATE

Candidate’s Declaration

We hereby declare that the work presented in this report entitled “Pattern and Anomaly Recognition in Industrial Scenarios” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2016 to April 2017 under the supervision of Dr. Hemraj Saini (Associate Professor, Computer Science & Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Yash Agarwal (131230)   Pushpak Passey (131278)

This is to certify that the above statement made by the candidate is true to the best of our knowledge.

Dr. Hemraj Saini

Associate Professor, Computer Science & Engineering

Dated:

ACKNOWLEDGEMENT

We extend our heartfelt thanks to our final year project supervisor Dr. Hemraj Saini. He gives us valuable advices for our project and how to present well. Also, he reminded us regularly to schedule and compile the work well.

Besides, we would like to thank Puneet Gupta in various stages. He provided us with facilities and technical support when we were setting up our system environment, and algorithms related to R and Hadoop.

Lastly, we would thank Shivi Sharma. She gave us motivation at times which we needed most desperately and also helped us in lab. So, we could perform better and pave the way for a solid platform for our project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPHS

ABSTRACT

This report presents our group's final year project. In this project we will describe the following points:

-Objective of the Project

-Experiments done

-Advantages and Disadvantages of using Hadoop

-Our proposal

The objective of our project is to use R programming and Hadoop to handle massive data processing like industrial data (structured and semi-structured).

So to achieve our objective, we chose the two technologies enlisted. The first one is R programming which allows us to graphically represent the data in an efficient way. The second technology we have utilized here is Hadoop, a software framework to implement the data-intensive programs and algorithms with thousands of machines. Hadoop evenly allocates the storage, computation power and divide large jobs to separate machines with many tiny jobs. Since we are emphasizing on two relatively new technologies (R and Hadoop) to handle a large scale job, we must fully understand the characteristics of these systems. To fully comprehend and understand these two technologies, we did many experiments on testing the performance, especially the scalability of these two technologies and optimize the respective environment.

Hence came the challenges, which were initially tough to tackle but were then one by one solved and the results showed up. The essence of our project deals with the aadhaar scheme and the census of India and tries to tackle some fundamental problems that are faced due to the inefficiency of the governmental systems. The data collected manually is ought to have some discrepancies, especially when done for a country like India, where population acts as a hurdle. There lies in the challenge and the right steps, if taken are bound to provide positive results to improve the data collection and analysis of our nation, which ultimately leads to better solutions for the society.

Chapter 1: INTRODUCTION

1.1 Introduction

All Indian residents have the right to be empowered with a unified and unique identification which can be used anytime and anywhere to accomplish tasks like payment, identification, procurement of benefits etc. All this linked with digital platform

- Deliver Aadhaar numbers universally to residents with a well-enumerated processing time and quality and quantity characteristics.
- Establish partnerships to set up infrastructure and environment that calls for maximum participation from residents to handle and manipulate their digital data accordingly.
- Establishing partnerships with service providers in using Aadhaar as an asset to serve people in an equal and justified manner.
- Call for innovation and empower people with a platform for public and private authorities to design Aadhaar linked platforms and services.
- Try for availability, measurement and resilience of this contemporary technology commodity infrastructure.
- Construct an acute life maintainable organization to take forward with zeal the ideals and fundamentals of the UIDAI.
- Try to construct the scheme attractive for known and established expertise in distinguished fields to form coalitions and give good insights to the UIDAI organization for it's improvisation.

The UIDAI system is designed to assign a 12-digit unique identification (UID) number (called Aadhaar) to all the citizens of our nation. The execution of UID scheme follows the process of identification and allotment of UID to the population; describing mechanisms for combining UID with the established firm's databases; operation and manipulation of all stages and steps of UID lifecycle; designing and compiling policies and procedures for refurbishing mechanism and describing utilization and application of UID for dispersion of varied services. The aadhaar number is connected with the resident's basic regional and biometric data such as a signature, picture, ten finger prints and two eye (iris) scans,that are secured in a national database repository.

Census of India

Census of India is an activity conducted every 10 years and has been conducted 15 times, till 2001 one, which is the last one. The first one was conducted in 1872 under the aegis of the former British Empire and post-independence, from 1949, this power is vested with the office of the Registrar General and Census Commissioner of India under the Ministry of Home Affairs, Government of India.

This comprehensive and extensive report of the Census of India is the best and sole source of the statistical and integral data in all the varied aspects of the population of India. It is a practice deeply engraved in the democracy of our nation since the last 130 years. Various subjects as demography, economics, anthropology, sociology, and statistics use this data extensively and are vital to feed their research work. The truest sense of India's diversity comes out of this activity which has become an essential tool to comprehend the versatility of India.

1.2 Problem Statement

As the Government of India moves forward with a nationally unique identification system, with the aadhaar scheme at its center stage, many computational and analytical problems are set to become easier for the authorities to solve to alleviate and mitigate various problems on an immediate basis. One such activity is the Census of India which is a comprehensive survey and study of pan India, which is conducted every 10 years. Most of the government employees, be it central or state, are allotted duties under this scheme to conduct the surveys by going home to home and collecting the data, which is quite a tedious process. Even after digitization, it is a tough task to collect all the data without pen and paper system due to humongous population and very less penetration of internet services. The loss in the form of physical resources and manpower wasted in the process is huge. The flagship government scheme of Digital India tries to tackle this problem through digital empowerment.

1.3 Objective

The reduction of manpower to reduce manual effort through digitization and automation is a long process and requires initiatives from both the government and the people. To take a step in this direction, aadhaar can perform as a powerful tool to accomplish such a task.

1.4 Methodology

Big data is a technical jargon for data and information blocks that are quite or complex that normally used data execution application software is incapable to handle with the existing techniques. The problems here in the existing system consist of compilation, manipulation, analysis, data maintenance, search, commodity sharing, transmission, digitization, questioning, updating and data privacy. The jargon "big data" seldom implies to the simple to the utilization of characteristic analysis, predictive analysis, end person, or some varied advanced data analytics algorithms that find the technical value from information, and somewhat to a specific size of sample data set.

Hadoop-

Apache Hadoop is an open-source software framework, used by most of the analysts around the world for distributed storage and processing of big data sets using the MapReduce and other such intricate programming models. It consists of large computer clusters built with and from commodity and personal hardware. All the various types of modules in Hadoop are designed with a basic and fundamental assumption that hardware or workstation failures are quite a known and repeated occurrence and thus, these must be automatically taken care of, by the framework.

The inner core of Apache Hadoop consists of storage part, known as Hadoop Distributed File System (HDFS), and executing part which is a MapReduce programming model. Hadoop takes the files and then breaks them into large blocks and distributes all of them across nodes in a cluster. It then transmits enroute code into nodes to process the data in parallel. This technique leverages an advantage of data and information locality, where nodes maintain and manipulate the data they have access to. This just allows the dataset to be executed faster and more efficiently than it would be in a normally used supercomputer architecture that always takes the shadow of or relies on a parallel file system where computation and data are distributed via high-speed networking.

Apache Hadoop software framework has been composed of the following modules:

- Hadoop Common – consists of the specific libraries and all the other distinguished utilities required by different options and divisions of Hadoop;
- Hadoop Distributed File System (HDFS) – a parallel and divided file-system which secures information on shared machines, and gives for extremely high added bandwidth within and outside the cluster;
- Hadoop YARN – a platform that manages all the various available resources for the base that is completely dependent on managing processing and execution of varied implementations in blocks and utilizing these resources for scheduling of all the end programmer applications

- Hadoop MapReduce – an execution of the MapReduce algorithm model for huge scale information processing.

HDFS

Hadoop Distributed File System was initiated and designed utilizing the distributed file system design. It is executed on shared hardware. Unlike most of the contemporary distributed systems, HDFS is exfremely faultresilient and the specific architectureis a low-cost shared hardware.

HDFS harbourslarge amounts of information and gives easier access. To harbor such large information, the stored files are secure using distributed multiple machines. These files are secured in a redundant way to help save the system from possible way of losing the important data when the system goes for a large scale turndown. HDFS also makes applications available to parallel processing.

Namenode

The namenode is the basic used and shared hardware that consists of the GNU/Linux operating system and the namenode software. This is a software that can run on shared hardware. The system which has the namenode acts as the master server and it accomplishes the following tasks:

- Manipulates the file system namespace.
- Defines and establishes client's access to files.
- It also executes and processes file system operations like renaming, closing, and opening files and directories.

Datanode

The datanode is a shared hardware having the GNU/Linux operating system and datanode software. For every single point in block, there is a datanode. These type of nodes manage the data storage of their system.

- Datanodes perform and implement read-write operations on all the file systems, as per the request of the clients.
- These kind of modules can also execute computations like a block initialization, removal, and copying according to the commands of the namenode.

Block

Usually, user information is secreted in the system documentations of HDFS. The document in a file system is distributed respectively into more than a single segment and then secured in individual distributed data nodes. The known division of files are called blocks. The minimum quantity of the information that HDFS can harbor and then manipulate it accordingly is called a Block. The default size of the block is 64MB, but it can be increased as per the need to change in HDFS configuration.

As already elucidated, ResourceManager (RM), the main entity here that mediates all the known divided resources and thus manages the divided applications executing on the YARN system. It co-ordinates together with NodeManagers (NMs) and ApplicationMasters (AMs), both of which work simultaneously on a node to node basis.

1. NodeManagers get the commands from the ResourceManager and manipulate all the resources that are known and retrievable on a single node.
2. ApplicationMasters are required to discuss all the varied resources with the ResourceManager and for constructing works with the NodeManagers to initiate the containers.

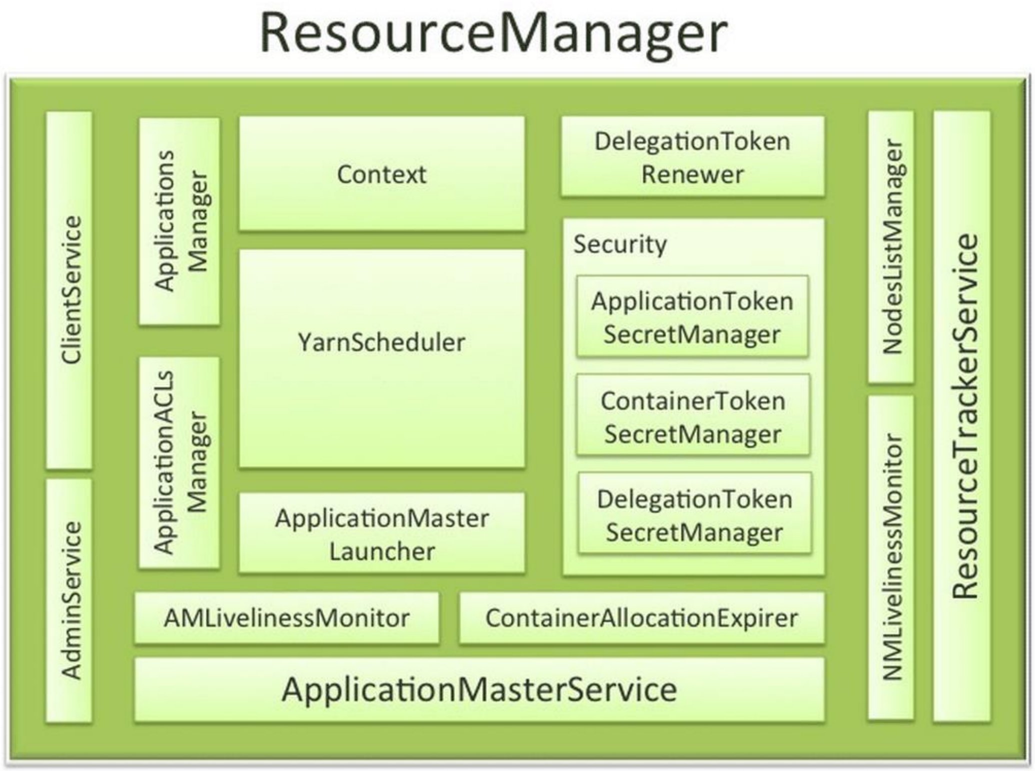


Figure 1: The core of Resource Manager

The NodeManager (NM) is YARN's depends on the node to node basis and takes care of the individual nodes computation in a Hadoop cluster. This involves keeping up pace with the ResourceManager (RM), overseeing containers' life-cycle management; monitoring resource usage (memory, CPU) of individual containers, following node-health, log's manipulation and auxiliary daemons which may be used by different YARN applications.

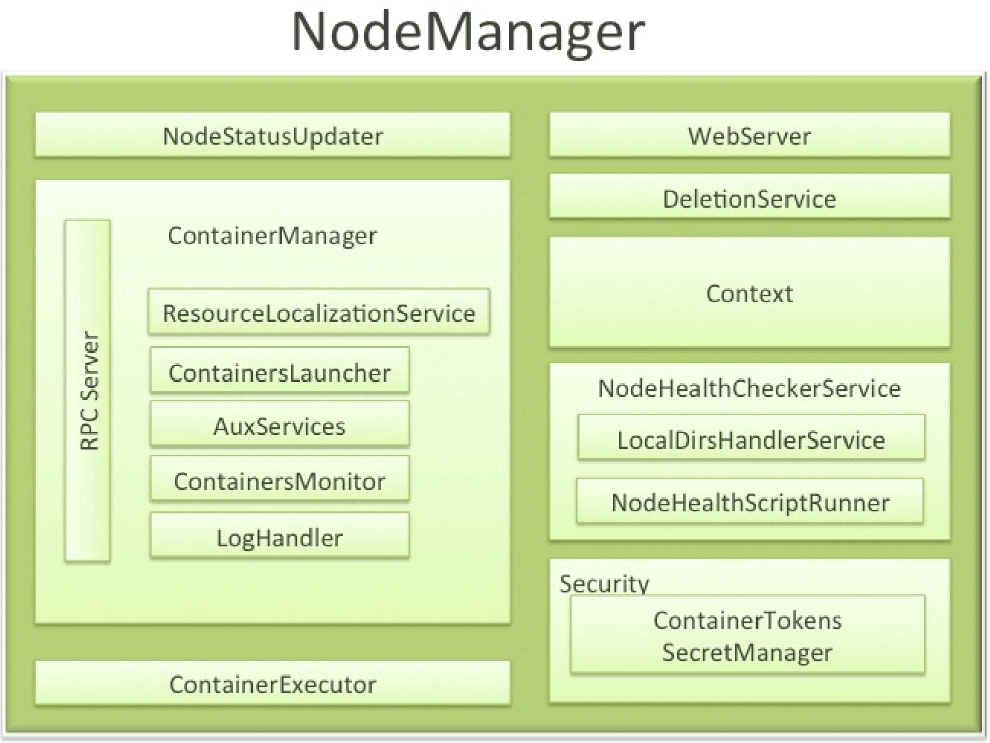


Figure 2: The core of Node Manager

Chapter-2: LITERATURE SURVEY

1. There is a need to find a more suitable technology for the scientists so that they can analysis the pictorial information about maybe weather or to study the changes in geographic features over time. So there can be a way to use an integration of R and Hadoop along with some classification algorithms like BFAST. So the memory limitations of R are overcome by using the features of Hadoop. They together provide a more flexible programming of complex analysis in storage components. In this paper, there is a guide to process analytics streaming approaches as a more generic way in terms of performance and capacity. They indicated that for various amount of pixels and MODIS time series, the processing time was linear for complex algorithms such as those found in deforestation detection applications.There is another technology i.e. Spark framework that is also a promising and efficient approach to be tested in our approach. [10, p.2-4]

2. With the development of remote sensing technology, the sizes of the data are becoming larger with the increase in resolution. It has become very time-consuming and a skill of specialists to cluster a large amount of images considering the limitations of both hardware and software resources. Some solutions were with MPI programming which again requires sophisticated skills of the user. They will demonstrate how robust scalability and the computational time are substantially reduced through increasing the number of nodes and it may inspire new solutions to other similar problems. The results show that their algorithm can effectively process remote sensing images in acceptable time, also that high-performance hardware devices reflect the superiority of MapReduce better. Since ISODATA algorithm is more complex and moreover currently Hadoop cannot support image format, they just transform images into text files and parallel execute partial stage of traditional ISODATA algorithm.[11, p.2-4]

3. To analyze the huge weather analysis data, which amounts to petabytes, the techniques to be used are to be scrutinised first, such as the pig and hive queries. Their performances are compared based on pseudo node and Hadoop Distributed Multinode cluster. With the ever increasing data on a day to day basis, multinode HDFS comes to the rescue. System hive programming proves its worth when used in a multinode environment. This provides the perfect mechanism for analyzing data like that of weather conditions.[12, p.4,5]

4. Structured prediction learning frameworks are costly to train which are widely applied in natural language processing. In this paper, they investigated distributed training strategies for the structured perceptron as a means to reduce training times when computing clusters are available. They presented experiments on two structured prediction problems- dependency parsing and name identity recognition to highlight the efficiency of this method.Their analysis shows that an iterative parameter mixing strategy is both guaranteed to significantly reduces the time required to train high-accuracy classifiers and separate the data (if possible). [13, p.-2,3]

5. Due to the social networking sites like Facebook, Twitter discovered the growth of data which will be uncontrollable in the future. So they proposed a method that will process the data in parallel as small chunks in distributed clusters and aggregate all the data across clusters to obtain the final processed data. The data are preferably refined using collaborative filtering, under the prediction mechanism of particular data needed by the user. They proposed that the unstructured data is structured and processed by using MapReduce technique and the automatic prediction of user’s taste is done through collaborative filtering. Also by using Tagging Techniques and Emoticon Based clustering, the developed method can be enhanced and the recommendation generation process can be even more efficient and optimized efficiently.[14, p.-2]

6. Big Data streaming analysis is the most sought after data to be successfully and efficiently analyzed as all the digital devices produce a large amount of data. Patterns exist in our usage and it is the job of such analysis to correctly tap on such usage, as it provides very sought after industrial data. A new methodology has been proposed to execute such analysis using the novel data structure, LERP reduced suffixed array, and the new, improvised ARPaD algorithm. A data stream of 1 trillion digits composed of 1 thousand subsequences of 1 billion digits has been used to conduct the experiments.1 billion data points have been taken and analyzed in 33 minutes using 10 computers of standard hardware configuration. According to the researchers, this outperforms any known analysis and amounts to data point generation every 2 microseconds. The flexibility of LERP-RSA data structure and the efficiency of ARPaD algorithm allow the detection of the patterns in a very lucid and swift manner. Queries are manipulated to provide the meta-analysis of the results.Trends can also be taken out of the results for forecasting purposes.The data used is sequential frequent and non-frequentitemsets detection, it can also be shifted to very big data streams.[15, p.-5]

7. Business intelligence and analytics are extensively used today by various business organizations to expand their prospects as data related problems that require optimized answers. These avenues are being explored by both practitioners and researchers. There are numerous versions of BI&A in the industry elucidated by their characteristics. More than a decade of research work and study related to the various fields of BI&A were taken into consideration. The BI&A version 1.0, with the critical aspects of the raw, unstructured data presented through the industry are taken into account. There are various fields of interest to BI&A, such as e-commerce, governance, healthcare, market intelligence, security etc. One by one, many of the fields are added to the newer versions of BI&A as published in the quarterly published magazine.[16, p.-3]

Chapter-3 SYSTEM DEVELOPMENT

3.1 Running Hadoop on Ubuntu Linux (Single-Node Cluster)

Prerequisites is Java Environment

To install run the following command

```
$ sudo apt-get update
$ sudo apt-get install openjdk-8-jdk
```

To check the correct installation

```
$ java –version
```

Adding a dedicated Hadoop system user

```
$ sudo addgroup hadoop

$ sudo adduser  --ingroup hadoop hduser
```

Configuring SSH

```
$ su – hduser

$ ssh -keygen -t rsa -P ""

$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

$ ssh localhost
```

Disabling IPv6

```
$ sudo gedit /etc/sysctl.conf (add the following lines )

# disable ipv6
```

```
net.ipv6.conf.all.disable_ipv6 = 1

net.ipv6.conf.default.disable_ipv6 = 1

net.ipv6.conf.lo.disable_ipv6 = 1
```

To check for proper disabling to IPv6

```
$ cat /proc/sys/net/ipv6/conf/all/disable_ipv6 (value must be 1)
```

Hadoop Installation

```
$ cd /usr/local

$ sudo tar xzf hadoop-2.7.2.tar.gz

$ sudo mv hadoop-2.7.2 hadoop

$ sudo chown -R hduser:hadoop hadoop
```

Update .bashrc file for environment variables

```
$ sudo gedit ~/.bashrc (add the following lines to the file)

export HADOOP_HOME=/usr/local/hadoop

export JAVA_HOME=/usr/lib/jvm/ openjdk-8-jdk

export PATH=$PATH:$HADOOP_HOME/bin
```

Configuration

```
$sudo gedit /usr/local/hadoop/share/hadoop/hadoop-env.sh (add the following lines)

export JAVA_HOME=/usr/lib/jvm/openjdk-8-jdk
```

Now create a directory and set the required ownerships and permissions to the directory:

```
$ sudo mkdir -p /app/hadoop/tmp

$ sudo chown hduser:hadoop /app/hadoop/tmp
$ sudo chmod 750 /app/hadoop/tmp
```

core-site.xml

```
$sudo gedit /usr/local/hadoop/share/hadoop/core-site.xml

<property>

<name>hadoop.tmp.dir</name>

<value>/app/hadoop/tmp</value>

</property>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:54310</value>

</property>
```

mapred-site.xml

```
<property>

<name>mapred.job.tracker</name>

<value>localhost:54311</value>

</property>
```

hdfs-site.xml

```
<property>

<name>dfs.replication</name>

<value>1</value>

</property>
```

Formatting the HDFS filesystem via the NameNode

```
$ cd $HADOOP_HOME

$ bin/hadoop namenode –format
```

Starting single-node cluster

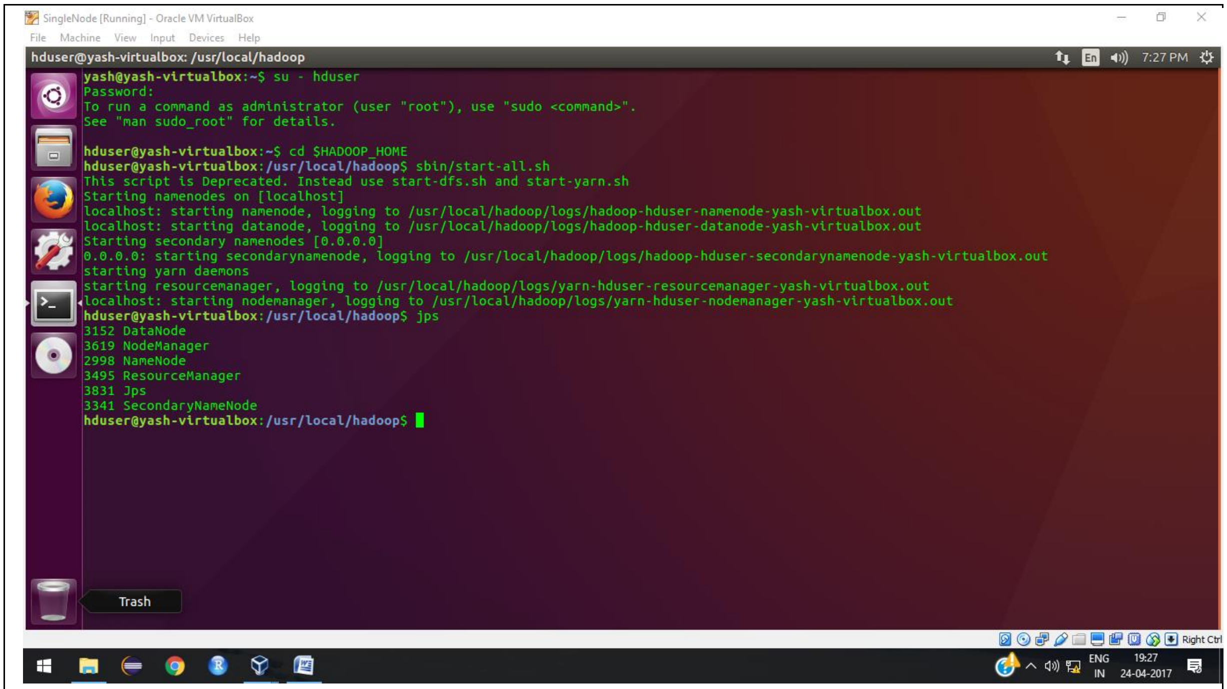
```
$ sbin/start-all.sh
```

To debug MapReduce programs

```
$ jps
```

Output must be like

```
2276TaskTracker

2156JobTracker

1984DataNode

2058SecondaryNameNode

2343Jps

1734NameNode
```



SingleNode [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hduser@yash-virtualbox: /usr/local/hadoop
7:27 PM
Trash
Right Ctrl
ENG
IN
19:27
24-04-2017

```
yash@yash-virtualbox:~$ su - hduser
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hduser@yash-virtualbox:~$ cd $HADOOP_HOME
hduser@yash-virtualbox:/usr/local/hadoop$ sbin/start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namenode-yash-virtualbox.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datanode-yash-virtualbox.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser-secondarynamenode-yash-virtualbox.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourcemanager-yash-virtualbox.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nodemanager-yash-virtualbox.out
hduser@yash-virtualbox:/usr/local/hadoop$ jps
3152 DataNode
3619 NodeManager
2998 NameNode
3495 ResourceManager
3831 Jps
3341 SecondaryNameNode
hduser@yash-virtualbox:/usr/local/hadoop$
```

Figure 3: JPS (Single Node)

Stopping single-node cluster

$ sbin/stop-all.sh

Copy local example data to HDFS

Let there be some files in Documents of the local system so after starting all the demons write the following commands:

$ bin/hdfs dfs – mkdir /user

$ bin/hdfs dfs – mkdir /user/hduser/

$ bin/hadoop dfs –copyFromLocal /home/hduser/Documents/files /user/hduser/

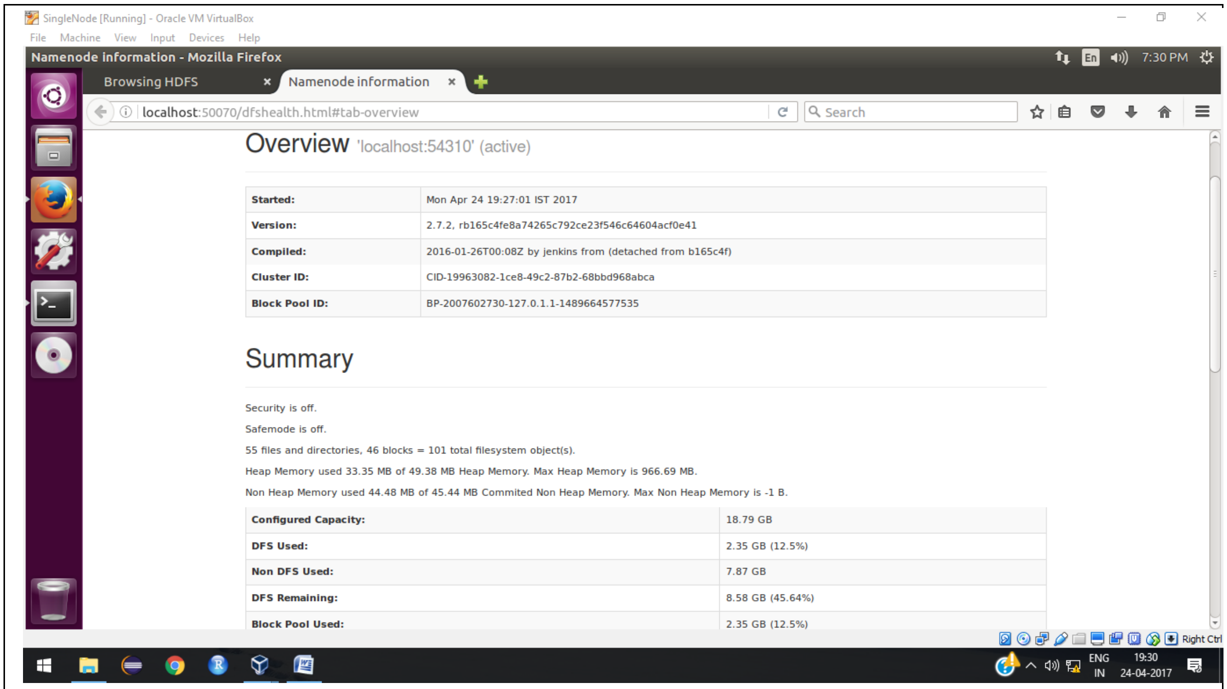
Run the MapReduce job (to run the WordCount example job.)

$ bin/hadoop jar hadoop-examples-1.0.3.jar wordcount /user/hduser/files /user/hduser/files-output

Retrieve the job result from HDFS

```
$ bin/hadoop dfs -cat /user/hduser/files-output/part-r-00000
```

NameNode Web Interface (HDFS layer)


SingleNode [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Namenode information - Mozilla Firefox
7:30 PM
Browsing HDFS
Namenode information
localhost:50070/dfshealth.html#tab-overview
Search
Right Ctrl
ENG IN
19:30
24-04-2017

Overview 'localhost:54310' (active)

Started:	Mon Apr 24 19:27:01 IST 2017
Version:	2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41
Compiled:	2016-01-26T00:08Z by jenkins from (detached from b165c4f)
Cluster ID:	CID-19963082-1ce8-49c2-87b2-68bbd968abca
Block Pool ID:	BP-2007602730-127.0.1.1-1489664577535

Summary

Security is off.

Safemode is off.

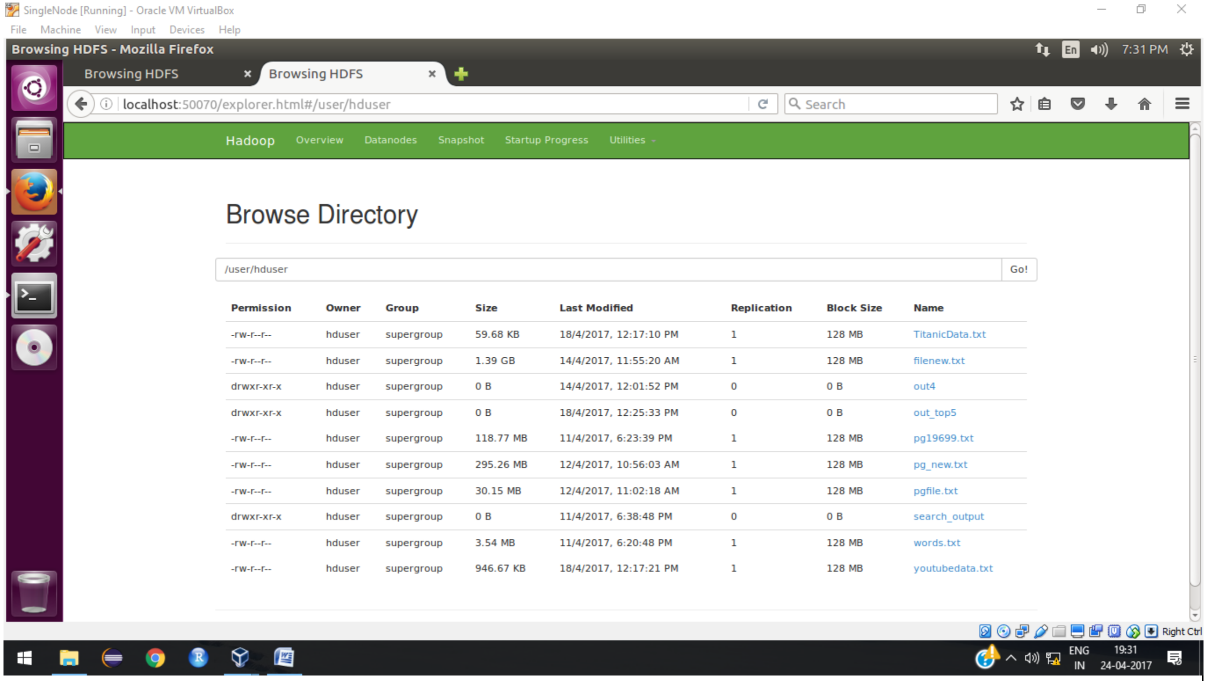
55 files and directories, 46 blocks = 101 total filesystem object(s).

Heap Memory used 33.35 MB of 49.38 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 44.48 MB of 45.44 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	18.79 GB
DFS Used:	2.35 GB (12.5%)
Non DFS Used:	7.87 GB
DFS Remaining:	8.58 GB (45.64%)
Block Pool Used:	2.35 GB (12.5%)

Figure 4: NameNode on localhost (Single Node)


SingleNode [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Browsing HDFS - Mozilla Firefox
7:31 PM
Browsing HDFS
Browsing HDFS
localhost:50070/explorer.html#/user/hduser
Search
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Right Ctrl
ENG IN
19:31 24-04-2017

Browse Directory

/user/hduser Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hduser	supergroup	59.68 KB	18/4/2017, 12:17:10 PM	1	128 MB	TitanicData.txt
-rw-r--r--	hduser	supergroup	1.39 GB	14/4/2017, 11:55:20 AM	1	128 MB	filenew.txt
drwxr-xr-x	hduser	supergroup	0 B	14/4/2017, 12:01:52 PM	0	0 B	out4
drwxr-xr-x	hduser	supergroup	0 B	18/4/2017, 12:25:33 PM	0	0 B	out_top5
-rw-r--r--	hduser	supergroup	118.77 MB	11/4/2017, 6:23:39 PM	1	128 MB	pg19699.txt
-rw-r--r--	hduser	supergroup	295.26 MB	12/4/2017, 10:56:03 AM	1	128 MB	pg_new.txt
-rw-r--r--	hduser	supergroup	30.15 MB	12/4/2017, 11:02:18 AM	1	128 MB	pgfile.txt
drwxr-xr-x	hduser	supergroup	0 B	11/4/2017, 6:38:48 PM	0	0 B	search_output
-rw-r--r--	hduser	supergroup	3.54 MB	11/4/2017, 6:20:48 PM	1	128 MB	words.txt
-rw-r--r--	hduser	supergroup	946.67 KB	18/4/2017, 12:17:21 PM	1	128 MB	youtubedata.txt

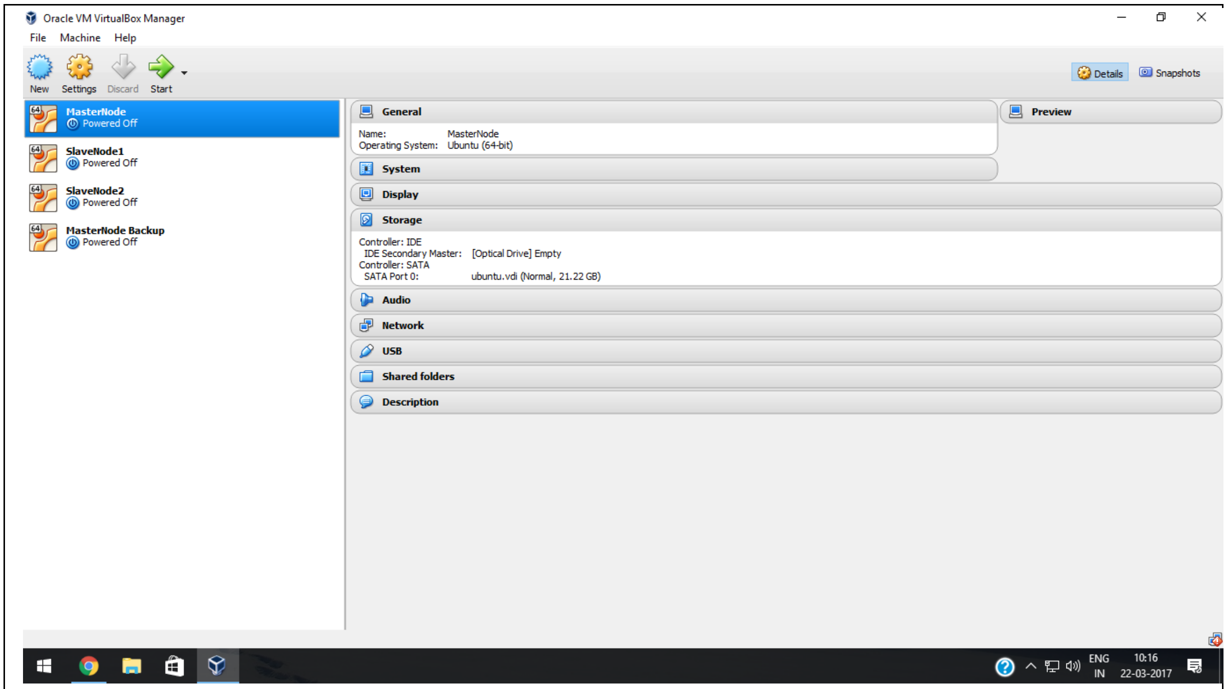
Figure 5: HDFS on localhost (Single Node)

3.2 Running Hadoop on Ubuntu Linux (Multi-Node Cluster)

Prerequisites is Configuring single-node clusters first

Networking (on master)

```
$ sudo gedit/etc/hosts (add the following lines)

192.168.0.1    master

192.168.0.2    slave1

192.168.0.3    slave2
```



MasterNode
Powered Off

SlaveNode1
Powered Off

SlaveNode2
Powered Off

MasterNode Backup
Powered Off

```
Name:               MasterNode
Operating System:   Ubuntu (64-bit)
```

```
Controller: IDE
 IDE Secondary Master:   [Optical Drive] Empty
Controller: SATA
 SATA Port 0:            ubuntu.vdi (Normal, 21.22 GB)
```

Figure 6: MultiNode setup (Multi Node)

SSH access (on master)

```
$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@slave

$ ssh master

$ ssh slave1

$ ssh slave2
```

Hadoop Configuration conf/*-site.xml (all machines)

core-site.xml

```
<property>

<name>fs.default.name</name>

<value>hdfs://master:54310</value>
```

```
</property>
```

mapred-site.xml

```
<property>

<name>mapred.job.tracker</name>

<value>master:54311</value>

</property>
```

hdfs-site.xml

```
<property>

<name>dfs.replication</name>

<value>3</value>

</property>
```

Formatting the HDFS filesystem via the NameNode (on master)

```
$ bin/hadoop namenode–format
```

Starting the multi-node cluster

```
$ sbin/start-all.sh

$ jps
```

Output on master must be like

```
16034Jps

14777NameNode

15654TaskTracker

14876DataNode
```

15566JobTracker

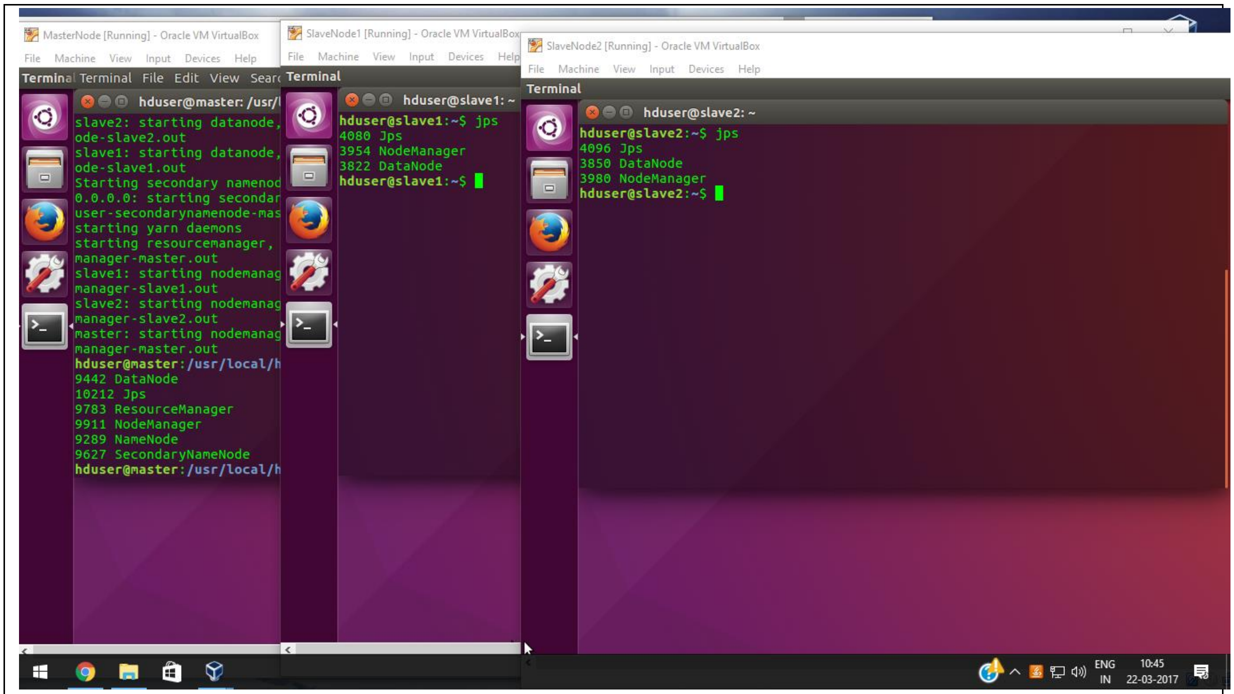
14934SecondaryNameNode

Output on slave must be like

15154DataNode

15845TaskTracker

16290Jps



MasterNode [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal Terminal File Edit View Searc
hduser@master: /usr/l
SlaveNode1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
hduser@slave1: ~
SlaveNode2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
hduser@slave2: ~
ENG
IN
10:45
22-03-2017

```
slave2: starting datanode,
ode-slave2.out
slave1: starting datanode,
ode-slave1.out
Starting secondary namenod
0.0.0.0: starting secondar
user-secondarynamenode-mas
starting yarn daemons
starting resourcemanager,
manager-master.out
slave1: starting nodemanag
manager-slave1.out
slave2: starting nodemanag
manager-slave2.out
master: starting nodemanag
manager-master.out
hduser@master:/usr/local/h
9442 DataNode
10212 Jps
9783 ResourceManager
9911 NodeManager
9289 NameNode
9627 SecondaryNameNode
hduser@master:/usr/local/h
```

```
hduser@slave1:~$ jps
4080 Jps
3954 NodeManager
3822 DataNode
hduser@slave1:~$
```

```
hduser@slave2:~$ jps
4096 Jps
3850 DataNode
3980 NodeManager
hduser@slave2:~$
```

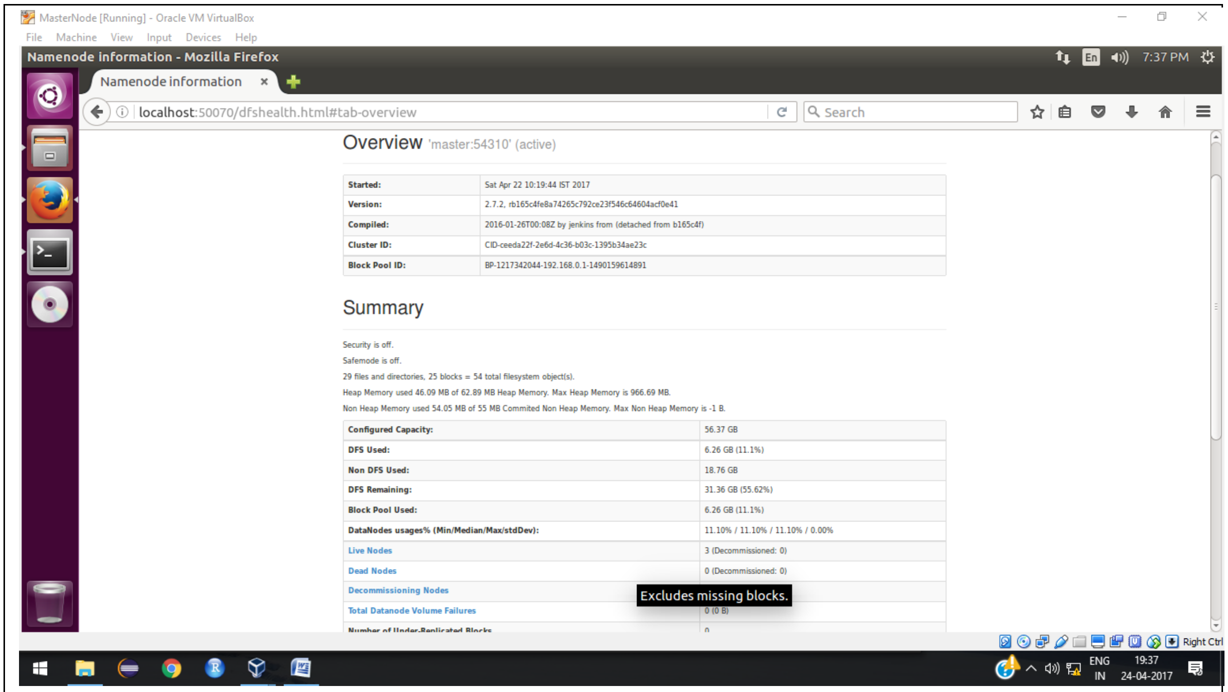
Figure 7: JPS(Multi Node)

Stopping the multi-node cluster

```
$ sbin/stop-all.sh
```

Hadoop Web Interfaces

- http://localhost:50070/ – web UI of the NameNode daemon


MasterNode [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Namenode information - Mozilla Firefox
En
7:37 PM
Namenode information
localhost:50070/dfshealth.html#tab-overview
Search
Right Ctrl
ENG
IN
19:37
24-04-2017

Overview 'master:54310' (active)

Started:	Sat Apr 22 10:19:44 IST 2017
Version:	2.7.2, rb165c4fe8a74265c792ce23f546c64604acf0e41
Compiled:	2016-01-26T00:08Z by jenkins from (detached from b165c4f)
Cluster ID:	CID-ceeda22f-2e6d-4c36-b03c-1395b34ae23c
Block Pool ID:	BP-1217342044-192.168.0.1-1490159614891

Summary

Security is off.

Safemode is off.

29 files and directories, 25 blocks = 54 total filesystem object(s).

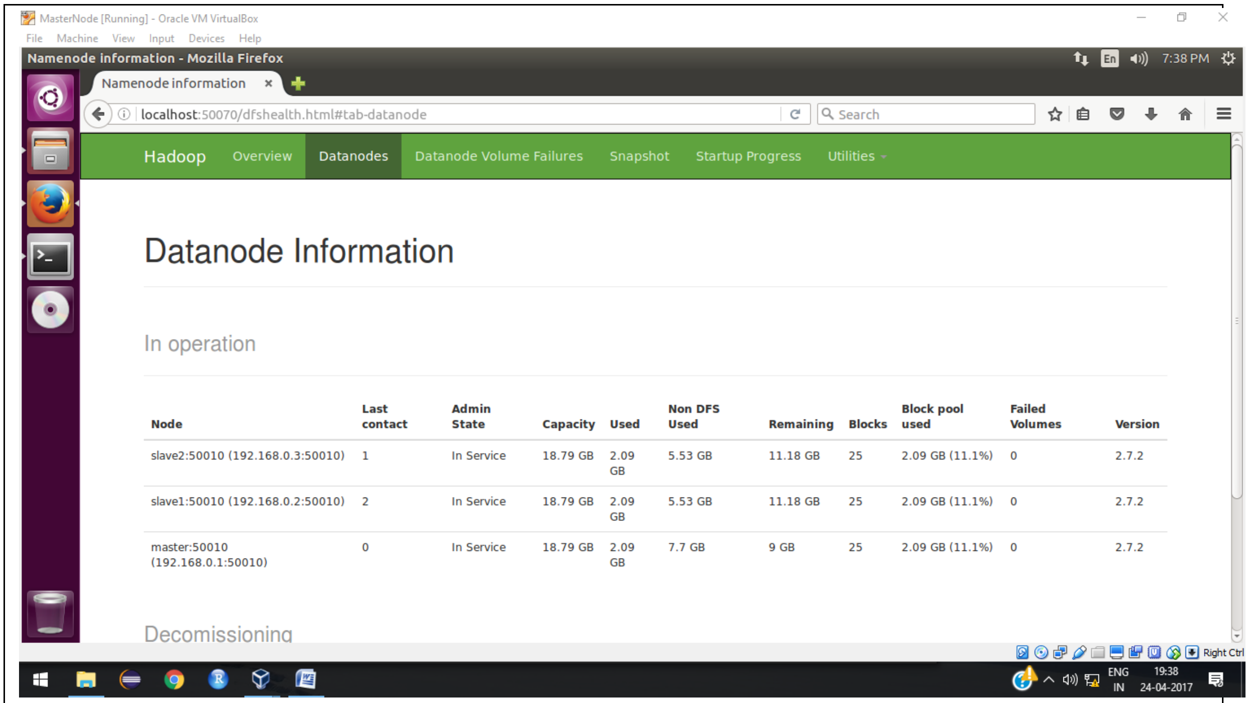
Heap Memory used 46.09 MB of 62.89 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 54.05 MB of 55 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	56.37 GB
DFS Used:	6.26 GB (11.1%)
Non DFS Used:	18.76 GB
DFS Remaining:	31.36 GB (55.62%)
Block Pool Used:	6.26 GB (11.1%)
DataNodes usages% (Min/Median/Max/stdDev):	11.10% / 11.10% / 11.10% / 0.00%
Live Nodes	3 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0

Excludes missing blocks.

Figure 8: Namenode on localhost (Multi Node)


MasterNode [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Namenode information - Mozilla Firefox
7:38 PM
Namenode information
localhost:50070/dfshealth.html#tab-datanode
Search
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Right Ctrl
ENG IN
19:38 24-04-2017

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
slave2:50010 (192.168.0.3:50010)	1	In Service	18.79 GB	2.09 GB	5.53 GB	11.18 GB	25	2.09 GB (11.1%)	0	2.7.2
slave1:50010 (192.168.0.2:50010)	2	In Service	18.79 GB	2.09 GB	5.53 GB	11.18 GB	25	2.09 GB (11.1%)	0	2.7.2
master:50010 (192.168.0.1:50010)	0	In Service	18.79 GB	2.09 GB	7.7 GB	9 GB	25	2.09 GB (11.1%)	0	2.7.2

Decomissioning

Figure 9: DataNode (Multi Node)

3.3 Problems Faced and Solutions

1. DataNode and NameNode were not getting started

   Solution - In the hdfs-site.xml file there should be dfs.data.dir property that points to a local directory. Delete everything under the directory and not the directory itself. Careful!! if you have any data on hdfs you will lose all of it.

   ```
   <property>
   <name>dfs.namenode.name.dir</name>
   <value>/hadoop/data/namenode</value>
   </property>
   <property>
   <name>dfs.datanode.data.dir</name>
   ```

```
<value>/hadoop/data/datanode</value>
</property>
```

2. Unable to format NameNode

   Solution –New hadoop version is compiled with JDK7, and will not run with a lesser JDK version (such as 6, which is no longer supported).

3. HDFS goes into read-only mode and gives error "Name node is in safe mode"

   ```
   $ bin/hdfs dfs dfsadmin -safemode leave
   ```

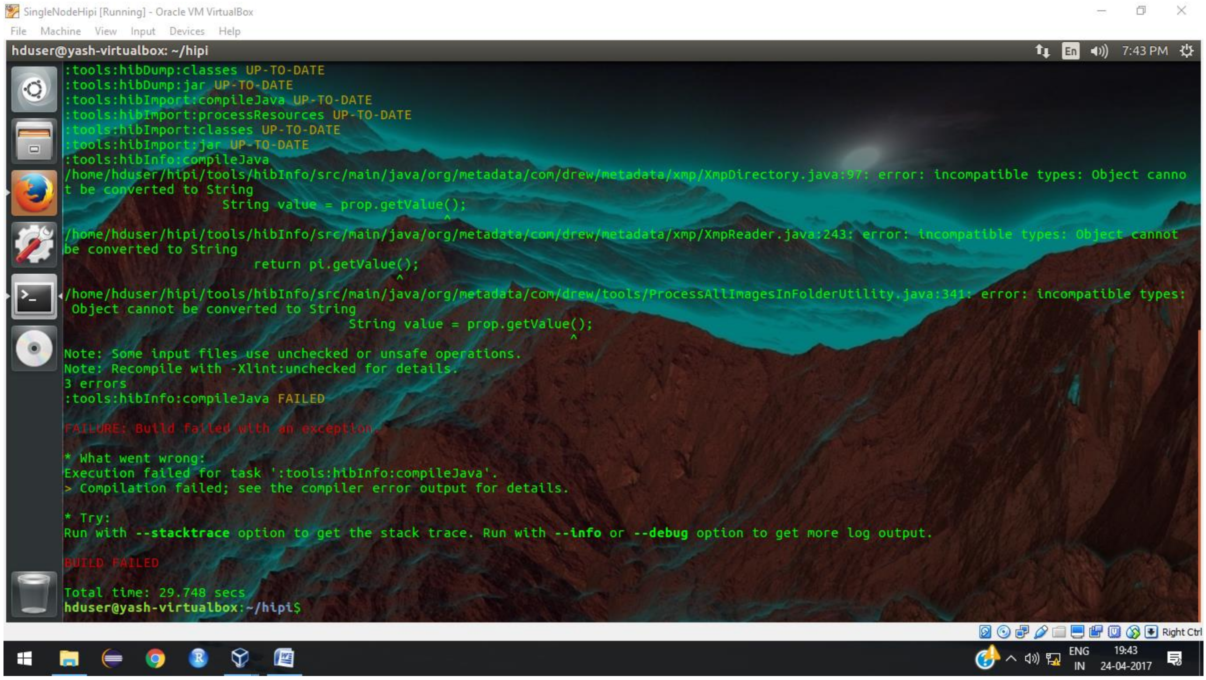
4. During the installation of Hadoop Image Processing Interface

   Firstly the problem occur while cloning the github repository. There were some issues with the SSH key generation and after removing that other errors made it impossible for us to install HIPI, so it was a fail experiment for us.

   There were some errors regarding only some particular class not found

   The classes were:

4.1 While compiling the hipi folder with gradle command :

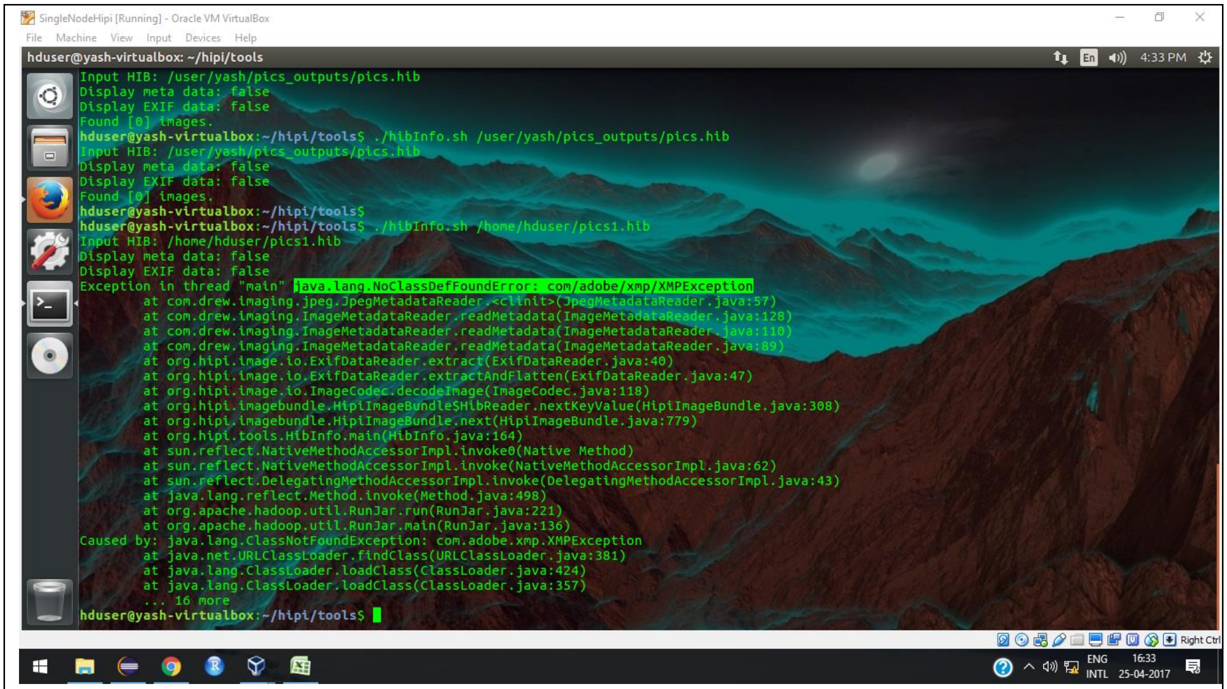

SingleNodeHipi [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hduser@yash-virtualbox: ~/hipi
En
7:43 PM
Right Ctrl
ENG
IN
19:43
24-04-2017

```
:tools:hibDump:classes UP-TO-DATE
:tools:hibDump:jar UP-TO-DATE
:tools:hibImport:compileJava UP-TO-DATE
:tools:hibImport:processResources UP-TO-DATE
:tools:hibImport:classes UP-TO-DATE
:tools:hibImport:jar UP-TO-DATE
:tools:hibInfo:compileJava
/home/hduser/hipi/tools/hibInfo/src/main/java/org/metadata/com/drew/metadata/xmp/XmpDirectory.java:97: error: incompatible types: Object canno
t be converted to String
                    String value = prop.getValue();
                                                ^
/home/hduser/hipi/tools/hibInfo/src/main/java/org/metadata/com/drew/metadata/xmp/XmpReader.java:243: error: incompatible types: Object cannot
be converted to String
                        return pi.getValue();
                                          ^
/home/hduser/hipi/tools/hibInfo/src/main/java/org/metadata/com/drew/tools/ProcessAllImagesInFolderUtility.java:341: error: incompatible types:
 Object cannot be converted to String
                                    String value = prop.getValue();
                                                                ^
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
3 errors
:tools:hibInfo:compileJava FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':tools:hibInfo:compileJava'.
> Compilation failed; see the compiler error output for details.

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output.

BUILD FAILED

Total time: 29.748 secs
hduser@yash-virtualbox:~/hipi$
```

Figure 10: Error in grade

This error is about some conversion. Previously it was working fine and removing other errors led to this error which we could find a solution to and was left due to it.

4.2 While running the ./hibImport.sh command :

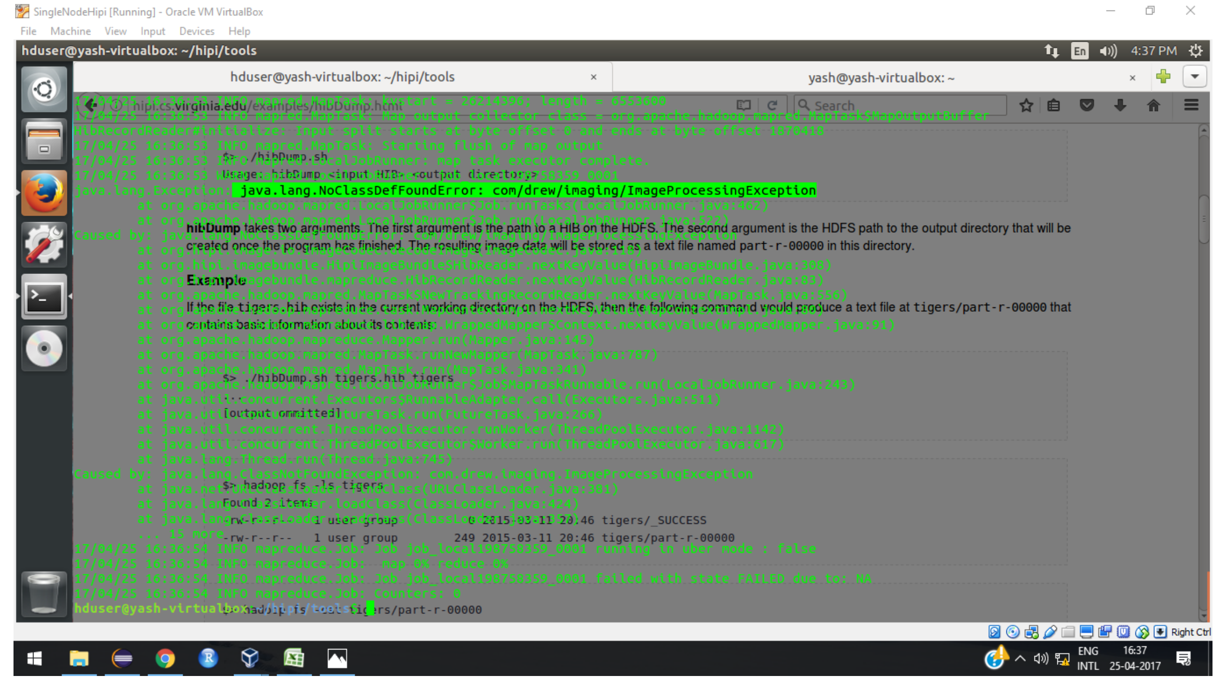

SingleNodeHipi [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hduser@yash-virtualbox: ~/hipi/tools
En
4:33 PM
Right Ctrl
ENG
INTL
16:33
25-04-2017

```
Input HIB: /user/yash/pics_outputs/pics.hib
Display meta data: false
Display EXIF data: false
Found [0] images.
hduser@yash-virtualbox:~/hipi/tools$ ./hibInfo.sh /user/yash/pics_outputs/pics.hib
Input HIB: /user/yash/pics_outputs/pics.hib
Display meta data: false
Display EXIF data: false
Found [0] images.
hduser@yash-virtualbox:~/hipi/tools$
hduser@yash-virtualbox:~/hipi/tools$ ./hibInfo.sh /home/hduser/pics1.hib
Input HIB: /home/hduser/pics1.hib
Display meta data: false
Display EXIF data: false
Exception in thread "main" java.lang.NoClassDefFoundError: com/adobe/xmp/XMPException
        at com.drew.imaging.jpeg.JpegMetadataReader.<clinit>(JpegMetadataReader.java:57)
        at com.drew.imaging.ImageMetadataReader.readMetadata(ImageMetadataReader.java:128)
        at com.drew.imaging.ImageMetadataReader.readMetadata(ImageMetadataReader.java:110)
        at com.drew.imaging.ImageMetadataReader.readMetadata(ImageMetadataReader.java:89)
        at org.hipi.image.io.ExifDataReader.extract(ExifDataReader.java:40)
        at org.hipi.image.io.ExifDataReader.extractAndFlatten(ExifDataReader.java:47)
        at org.hipi.image.io.ImageCodec.decodeImage(ImageCodec.java:118)
        at org.hipi.imagebundle.HipiImageBundle$HibReader.nextKeyValue(HipiImageBundle.java:308)
        at org.hipi.imagebundle.HipiImageBundle.next(HipiImageBundle.java:779)
        at org.hipi.tools.HibInfo.main(HibInfo.java:164)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
Caused by: java.lang.ClassNotFoundException: com.adobe.xmp.XMPException
        at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
        ... 16 more
hduser@yash-virtualbox:~/hipi/tools$
```

Figure 11: XMPException - class not found

In this error, there is an Exception about a Class can’t be found. The class name is XMPException. We do find some solution how to remove this Exception but didn’t work. So it was also a left out error which led to a failure of installation of HIPI.

4.3 While running the ./hibDumb.sh command


SingleNodeHipi [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hduser@yash-virtualbox: ~/hipi/tools
hduser@yash-virtualbox: ~/hipi/tools
yash@yash-virtualbox: ~
hipi.cs.virginia.edu/examples/hibDump.html
Search
4:37 PM
Right Ctrl
ENG INTL
16:37
25-04-2017

```
17/04/25 16:36:53 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
17/04/25 16:36:53 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
HibRecordReader#initialize: Input split starts at byte offset 0 and ends at byte offset 1870418
17/04/25 16:36:53 INFO mapred.MapTask: Starting flush of map output
17/04/25 16:36:53 INFO mapred.LocalJobRunner: map task executor complete.
17/04/25 16:36:53 WARN mapred.LocalJobRunner: job_local198758359_0001
java.lang.Exception: java.lang.NoClassDefFoundError: com/drew/imaging/ImageProcessingException
        at org.apache.hadoop.mapred.LocalJobRunner$Job.runTasks(LocalJobRunner.java:462)
        at org.apache.hadoop.mapred.LocalJobRunner$Job.run(LocalJobRunner.java:522)
Caused by: java.lang.NoClassDefFoundError: com/drew/imaging/ImageProcessingException
        at org.hipi.image.io.ImageCodec.decodeImage(ImageCodec.java:118)
        at org.hipi.imagebundle.HipiImageBundle$HibReader.nextKeyValue(HipiImageBundle.java:308)
        at org.hipi.imagebundle.mapreduce.HibRecordReader.nextKeyValue(HibRecordReader.java:83)
        at org.apache.hadoop.mapred.MapTask$NewTrackingRecordReader.nextKeyValue(MapTask.java:556)
        at org.apache.hadoop.mapreduce.task.MapContextImpl.nextKeyValue(MapContextImpl.java:80)
        at org.apache.hadoop.mapreduce.lib.map.WrappedMapper$Context.nextKeyValue(WrappedMapper.java:91)
        at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:145)
        at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:787)
        at org.apache.hadoop.mapred.MapTask.run(MapTask.java:341)
        at org.apache.hadoop.mapred.LocalJobRunner$Job$MapTaskRunnable.run(LocalJobRunner.java:243)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.run(FutureTask.java:266)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)
Caused by: java.lang.ClassNotFoundException: com.drew.imaging.ImageProcessingException
        at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
        ... 15 more
17/04/25 16:36:54 INFO mapreduce.Job: Job job_local198758359_0001 running in uber mode : false
17/04/25 16:36:54 INFO mapreduce.Job:  map 0% reduce 0%
17/04/25 16:36:54 INFO mapreduce.Job: Job job_local198758359_0001 failed with state FAILED due to: NA
17/04/25 16:36:54 INFO mapreduce.Job: Counters: 0
hduser@yash-virtualbox:~/hipi/tools$
```

```
$> ./hibDump.sh
Usage: hibDump <input HIB> <output directory>
```

hibDump takes two arguments. The first argument is the path to a HIB on the HDFS. The second argument is the HDFS path to the output directory that will be created once the program has finished. The resulting image data will be stored as a text file named `part-r-00000` in this directory.

Example

If the file `tigers.hib` exists in the current working directory on the HDFS, then the following command would produce a text file at `tigers/part-r-00000` that contains basic information about its contents:

```
$> ./hibDump.sh tigers.hib tigers
...
[output omitted]
```

```
$> hadoop fs -ls tigers
Found 2 items
-rw-r--r--   1 user group          0 2015-03-11 20:46 tigers/_SUCCESS
-rw-r--r--   1 user group        249 2015-03-11 20:46 tigers/part-r-00000
```

```
$> hadoop fs -cat tigers/part-r-00000
```

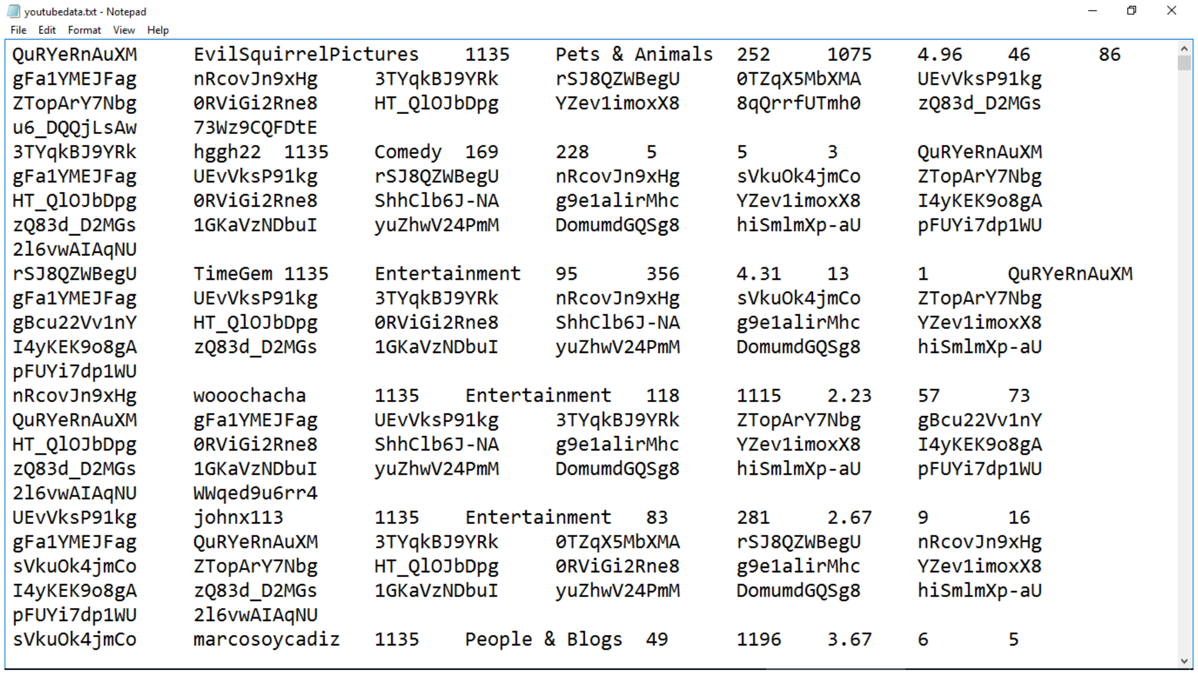
Figure 12: ImageProcessingException class not found

In this error, there is also an Exception of ImageProcessingException class not being found. This was removed with some suitable solutions.

Chapter 4:Performance Analysis

4.1 Analysis Problem 1- MapReduce Use Case-YouTube Data Analysis

YouTube is a major source of revenue and marketplace for firms these days. So the analysis of such a platform becomes increasingly important. This YouTube data is publicly available.



```
QuRYeRnAuXM     EvilSquirrelPictures    1135    Pets & Animals  252     1075    4.96    46      86
gFa1YMEJFag     nRcovJn9xHg     3TYqkBJ9YRk     rSJ8QZWBegU     0TZqX5MbXMA     UEvVksP91kg
ZTopArY7Nbg     0RViGi2Rne8     HT_QlOJbDpg     YZev1imoxX8     8qQrrfUTmh0     zQ83d_D2MGs
u6_DQQjLsAw     73Wz9CQFDtE
3TYqkBJ9YRk     hggh22  1135    Comedy  169     228     5       5       3       QuRYeRnAuXM
gFa1YMEJFag     UEvVksP91kg     rSJ8QZWBegU     nRcovJn9xHg     sVkuOk4jmCo     ZTopArY7Nbg
HT_QlOJbDpg     0RViGi2Rne8     ShhClb6J-NA     g9e1alirMhc     YZev1imoxX8     I4yKEK9o8gA
zQ83d_D2MGs     1GKaVzNDbuI     yuZhwV24PmM     DomumdGQSg8     hiSmlmXp-aU     pFUYi7dp1WU
2l6vwAIAqNU
rSJ8QZWBegU     TimeGem 1135    Entertainment   95      356     4.31    13      1       QuRYeRnAuXM
gFa1YMEJFag     UEvVksP91kg     3TYqkBJ9YRk     nRcovJn9xHg     sVkuOk4jmCo     ZTopArY7Nbg
gBcu22Vv1nY     HT_QlOJbDpg     0RViGi2Rne8     ShhClb6J-NA     g9e1alirMhc     YZev1imoxX8
I4yKEK9o8gA     zQ83d_D2MGs     1GKaVzNDbuI     yuZhwV24PmM     DomumdGQSg8     hiSmlmXp-aU
pFUYi7dp1WU
nRcovJn9xHg     wooochacha      1135    Entertainment   118     1115    2.23    57      73
QuRYeRnAuXM     gFa1YMEJFag     UEvVksP91kg     3TYqkBJ9YRk     ZTopArY7Nbg     gBcu22Vv1nY
HT_QlOJbDpg     0RViGi2Rne8     ShhClb6J-NA     g9e1alirMhc     YZev1imoxX8     I4yKEK9o8gA
zQ83d_D2MGs     1GKaVzNDbuI     yuZhwV24PmM     DomumdGQSg8     hiSmlmXp-aU     pFUYi7dp1WU
2l6vwAIAqNU     WWqed9u6rr4
UEvVksP91kg     johnx113        1135    Entertainment   83      281     2.67    9       16
gFa1YMEJFag     QuRYeRnAuXM     3TYqkBJ9YRk     0TZqX5MbXMA     rSJ8QZWBegU     nRcovJn9xHg
sVkuOk4jmCo     ZTopArY7Nbg     HT_QlOJbDpg     0RViGi2Rne8     g9e1alirMhc     YZev1imoxX8
I4yKEK9o8gA     zQ83d_D2MGs     1GKaVzNDbuI     yuZhwV24PmM     DomumdGQSg8     hiSmlmXp-aU
pFUYi7dp1WU     2l6vwAIAqNU
sVkuOk4jmCo     marcosoycadiz   1135    People & Blogs  49      1196    3.67    6       5
```

Figure 13: Sample Data of YouTube

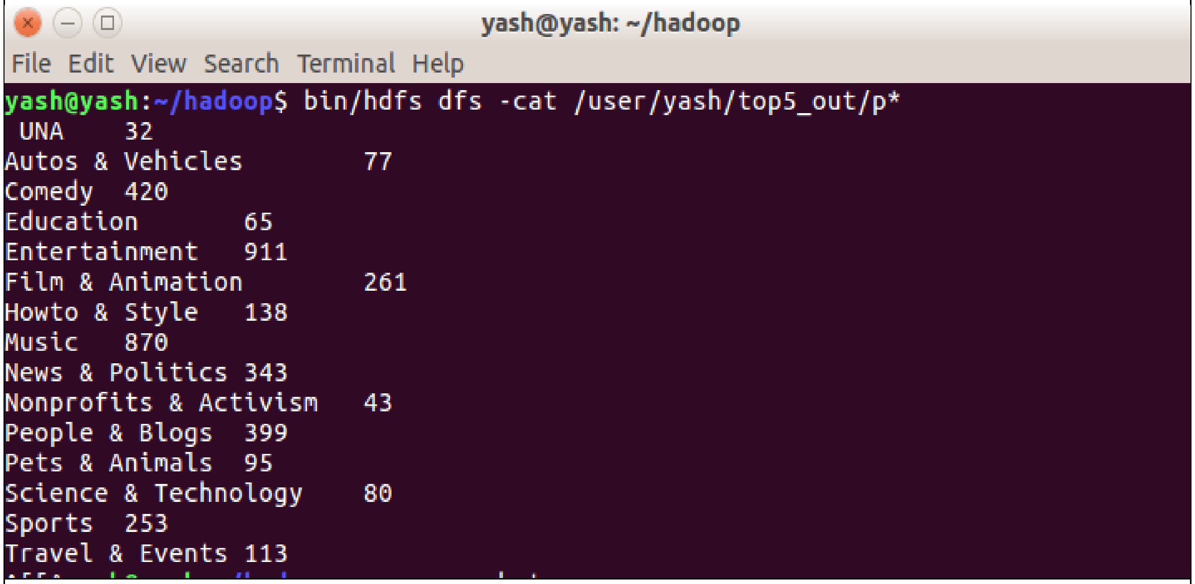
Problem Statement 1

Here we will find out what are the top 5 categories with a maximum number of videos uploaded.

Now from the mapper, we want to get the video category as key and final int value ‘1’ as values which will be passed to the shuffle and sort phase and are further sent to the reducer phase where the aggregation of the values is performed.

Command:-bin/hadoop jar top5.jar Top5_categories /user/yash/youtubedata.txt /user/yash/top5_out

Output:- bin/hdfs fs -cat /user/yash/top5_out/p* | sort -n -k2 -r |head -n5



```
yash@yash:~/hadoop$ bin/hdfs dfs -cat /user/yash/top5_out/p*
 UNA    32
Autos & Vehicles        77
Comedy  420
Education       65
Entertainment   911
Film & Animation        261
Howto & Style   138
Music   870
News & Politics 343
Nonprofits & Activism   43
People & Blogs  399
Pets & Animals  95
Science & Technology    80
Sports  253
Travel & Events 113
```

Figure 14 : Output of the analysis of YouTube data – 1

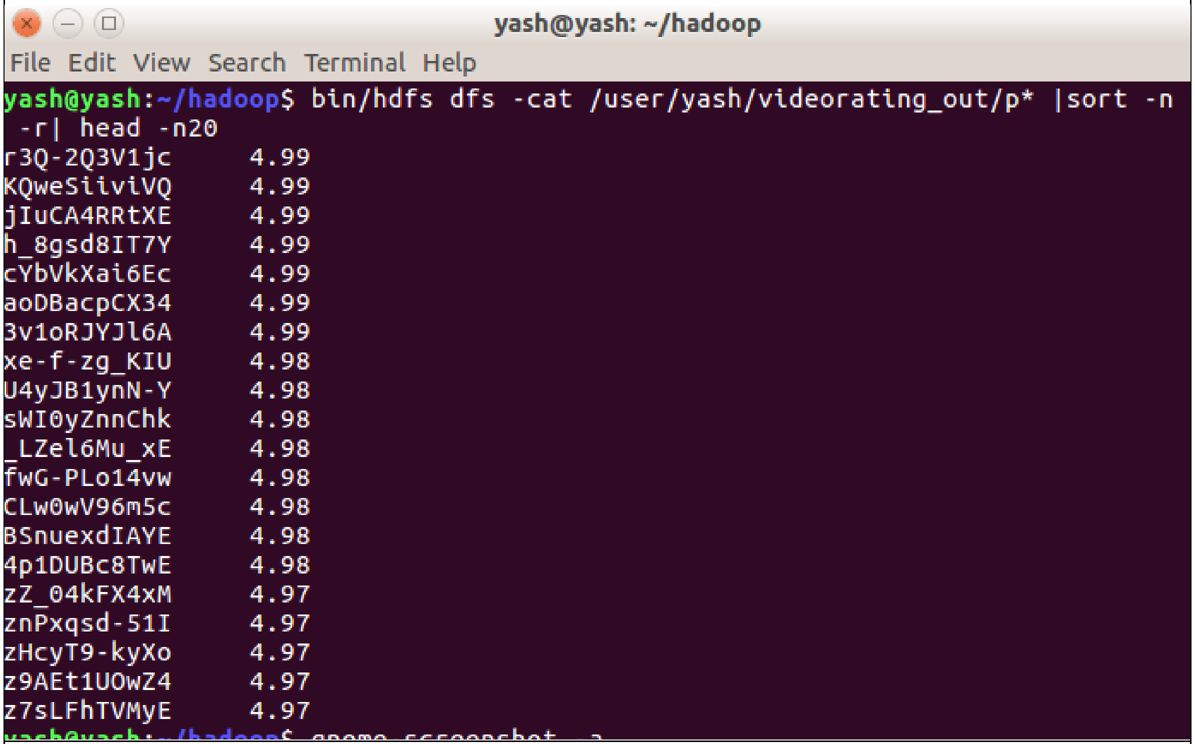
Problem Statement 2

In this problem statement, we will find the top 10 rated videos on YouTube.

Now from the mapper, we want to get the video id as key andratingas a valuewhich will be passed to the shuffle and sort phase and are further sent to the reducer phase where the aggregation of the values is performed.

Command :- bin/hadoop jar video_rating.jar /user/yash/youtubedata.txt /user/yash/videorating_out

output :- bin/hdfs fs -cat /user/yash/videorating_out/p* | sort –n –k2 –r | head –n20



```
yash@yash:~/hadoop$ bin/hdfs dfs -cat /user/yash/videorating_out/p* |sort -n
 -r| head -n20
r3Q-2Q3V1jc     4.99
KQweSiiviVQ     4.99
jIuCA4RRtXE     4.99
h_8gsd8IT7Y     4.99
cYbVkXai6Ec     4.99
aoDBacpCX34     4.99
3v1oRJYJl6A     4.99
xe-f-zg_KIU     4.98
U4yJB1ynN-Y     4.98
sWI0yZnnChk     4.98
_LZel6Mu_xE     4.98
fwG-PLo14vw     4.98
CLw0wV96m5c     4.98
BSnuexdIAYE     4.98
4p1DUBc8TwE     4.98
zZ_04kFX4xM     4.97
znPxqsd-51I     4.97
zHcyT9-kyXo     4.97
z9AEt1UOwZ4     4.97
z7sLFhTVMyE     4.97
yash@yash:~/hadoop$ gnome-screenshot -a
```

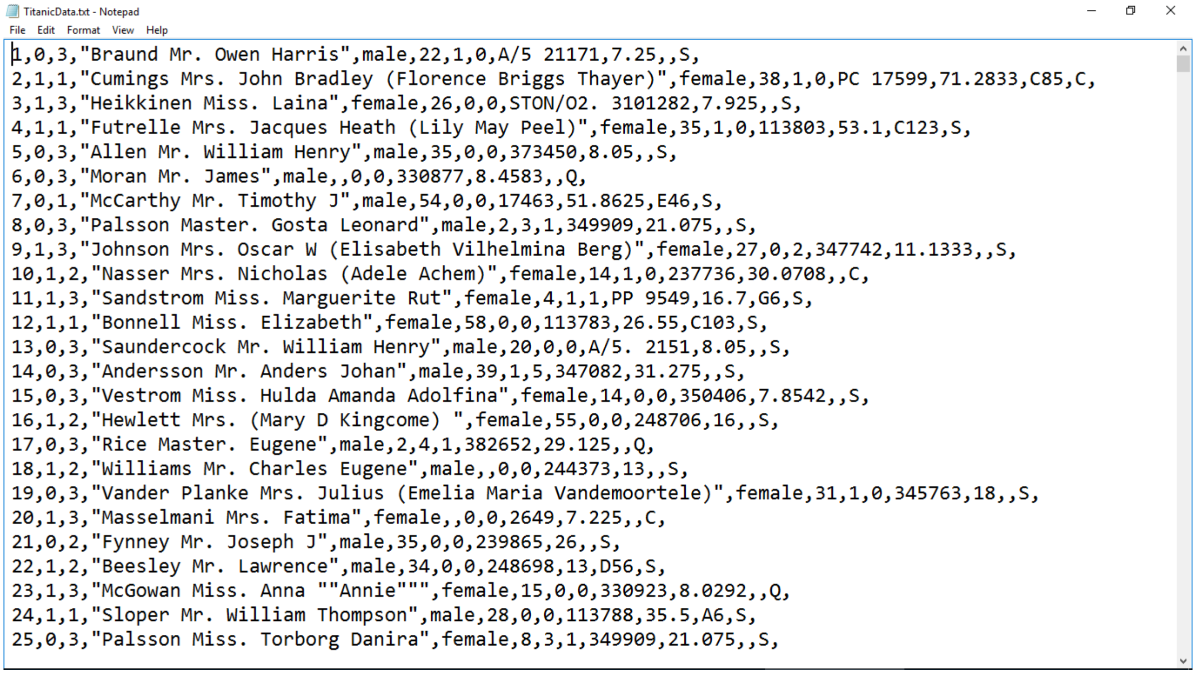
Figure 15: Output of the data of youtube data - 2

4.2 Analysis Problem 2 -Map reduce Use case – Titanic Data Analysis

There have been huge disasters in the history of Map reduce, but the magnitude of the Titanic’s disaster ranks as high as the depth it sank too. So much so that subsequent disasters have always been described as “titanic in proportion” – implying huge losses.

There have been as many inquisitions as there have been questions raised and equally that many types of analysis methods applied to arrive at conclusions. We will analyse the data that is present about the Titanic publicly. It actually uses Hadoop MapReduce to analyze and arrive at:

- The average age of the people (both male and female) who died in the tragedy using Hadoop MapReduce.
- How many persons survived – classwise.



```
1,0,3,"Braund Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S,
2,1,1,"Cumings Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C,
3,1,3,"Heikkinen Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S,
4,1,1,"Futrelle Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S,
5,0,3,"Allen Mr. William Henry",male,35,0,0,373450,8.05,,S,
6,0,3,"Moran Mr. James",male,,0,0,330877,8.4583,,Q,
7,0,1,"McCarthy Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S,
8,0,3,"Palsson Master. Gosta Leonard",male,2,3,1,349909,21.075,,S,
9,1,3,"Johnson Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S,
10,1,2,"Nasser Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C,
11,1,3,"Sandstrom Miss. Marguerite Rut",female,4,1,1,PP 9549,16.7,G6,S,
12,1,1,"Bonnell Miss. Elizabeth",female,58,0,0,113783,26.55,C103,S,
13,0,3,"Saundercock Mr. William Henry",male,20,0,0,A/5. 2151,8.05,,S,
14,0,3,"Andersson Mr. Anders Johan",male,39,1,5,347082,31.275,,S,
15,0,3,"Vestrom Miss. Hulda Amanda Adolfina",female,14,0,0,350406,7.8542,,S,
16,1,2,"Hewlett Mrs. (Mary D Kingcome) ",female,55,0,0,248706,16,,S,
17,0,3,"Rice Master. Eugene",male,2,4,1,382652,29.125,,Q,
18,1,2,"Williams Mr. Charles Eugene",male,,0,0,244373,13,,S,
19,0,3,"Vander Planke Mrs. Julius (Emelia Maria Vandemoortele)",female,31,1,0,345763,18,,S,
20,1,3,"Masselmani Mrs. Fatima",female,,0,0,2649,7.225,,C,
21,0,2,"Fynney Mr. Joseph J",male,35,0,0,239865,26,,S,
22,1,2,"Beesley Mr. Lawrence",male,34,0,0,248698,13,D56,S,
23,1,3,"McGowan Miss. Anna ""Annie""",female,15,0,0,330923,8.0292,,Q,
24,1,1,"Sloper Mr. William Thompson",male,28,0,0,113788,35.5,A6,S,
25,0,3,"Palsson Miss. Torborg Danira",female,8,3,1,349909,21.075,,S,
```

Figure 16 : Sample Data of Titanic

Problem statement 1:

In this problem statement we will find the average age of males and females who died in the Titanic tragedy.

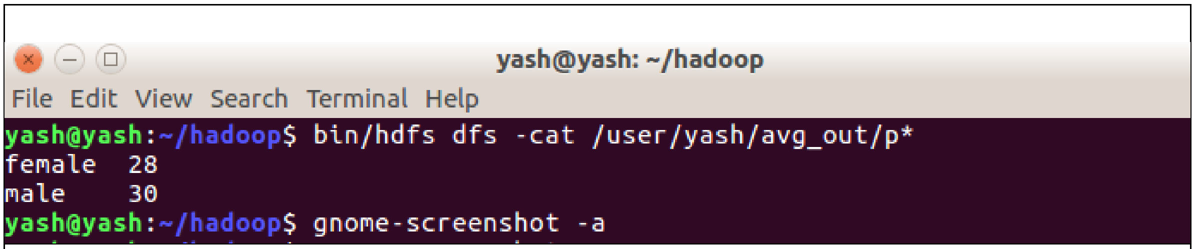
Now from the mapper we will derive:

- the gender as a key
- age as values

These values will be passed to the shuffle and sort phase and are further sent to the reducer phase where the aggregation of the values is performed.

Command :- bin/hadoop jar average.jar Average_age /user/yash/TitanicData.txt /user/yash/avg_out

output :- bin/hdfs dfs -cat /user/yash/avg_out/p*



```
yash@yash:~/hadoop$ bin/hdfs dfs -cat /user/yash/avg_out/p*
female  28
male    30
yash@yash:~/hadoop$ gnome-screenshot -a
```

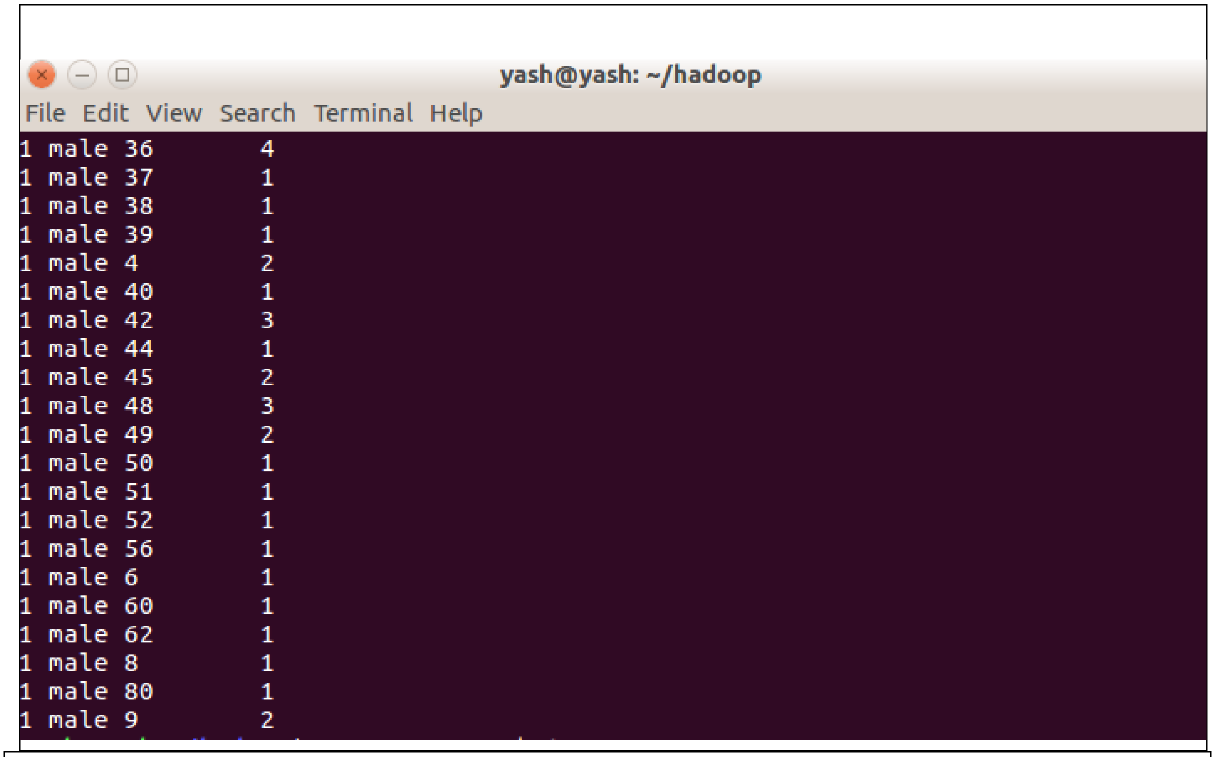
Figure 17 : Output of the analysis of Titanic data – 1

Problem statement 2:

In this problem statement, we will find the number of people died or survived in each class with their genders and ages.

Now from the mapper, we want to get the composite key which is the combination of Passenger class and gender and their age and final int value ‘1’ as values which will be passed to the shuffle and sort phase and are further sent to the reducer phase where the aggregation of the values is performed.

Command: bin/hadoop jar people_survived.jar /user/yash/TitanticData.txt /user/yash/people_out outout : bin/hdfs dfs -cat /user/yash/people_out/p*



```
1 male 36	4
1 male 37	1
1 male 38	1
1 male 39	1
1 male 4	2
1 male 40	1
1 male 42	3
1 male 44	1
1 male 45	2
1 male 48	3
1 male 49	2
1 male 50	1
1 male 51	1
1 male 52	1
1 male 56	1
1 male 6	1
1 male 60	1
1 male 62	1
1 male 8	1
1 male 80	1
1 male 9	2
```

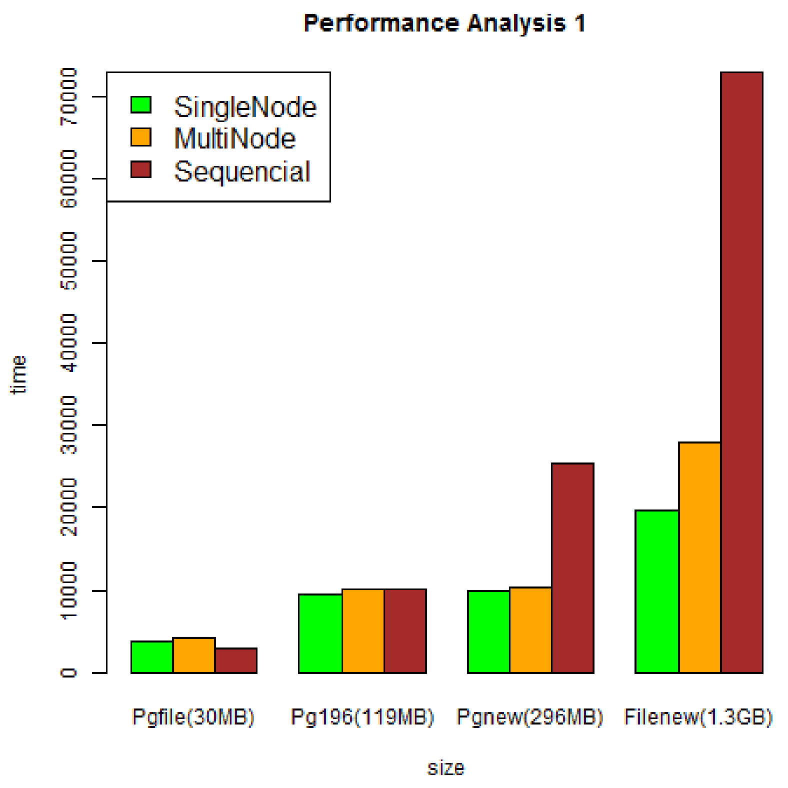
Figure 18 :Output of the analysis of Titanic data – 2

4.3 Performance Analysis 1:Word Frequency-

Table 1 : Single Node Execution Time - 1				
File Size\Time (ms)	Time 1	Time 2	Time 3	Mean
Pgfile (30 MB)	3714	3754	3845	3771
Pg196 (119 MB)	9654	9930	8916	9500
Pgnew (296 MB)	10077	9819	9708	9868
Filenew (1.3 GB)	-	-	-	19749

Table 2 : Multi Node Execution Time -1				
File Size\Time (ms)	Time 1	Time 2	Time 3	Mean
Pgfile (30 MB)	4038	4414	4119	4190
Pg196 (119 MB)	11025	9417	9921	10121
Pgnew (296 MB)	11514	9771	9463	10250
Filenew (1.3 GB)	-	-	-	27992

Table 3 : Sequential Execution Time - 1				
File Size	Pgfile (30 MB)	Pg196 (119 MB)	Pgnew (296 MB)	Filenew (1.3 GB)
Time (ms)	2980	10077	25331	123044



Graph 1: Performance Analysis -1

As evident from the graph, sequential provides decent results in the first two cases. This is due to the fact that in the singlenode and multinode executions, the overheads of mapping, reducing and of the transfer time amongst the various machines is also included, which overshadows the positive effects of the efficiency brought in by HDFS. There is the cost of transfer from the clients to the server and vice versa.

Then if the single node is considered, it proves worthy in all the 4 cases as even in the last case, the multinode's overhead of the transfer is more than the singlenode's computation of mapping and then reducing.

Similarly for the multinode scenario, it is the fastest one for large amount of data but not in this case even for the largest file (1.3 GB) as the overheads consume the profits. So, singlenode performs better.

4.4 Performance Analysis 2 :String Frequency-

File Size\Time(ms)	Time 1	Time 2	Time 3	Mean
Pgfile (30 MB)	630	766	583	660
Pg196 (119 MB)	588	532	592	571
Pgnew (296 MB)	767	587	556	637
Filenew (1.3 GB)	-	-	-	688

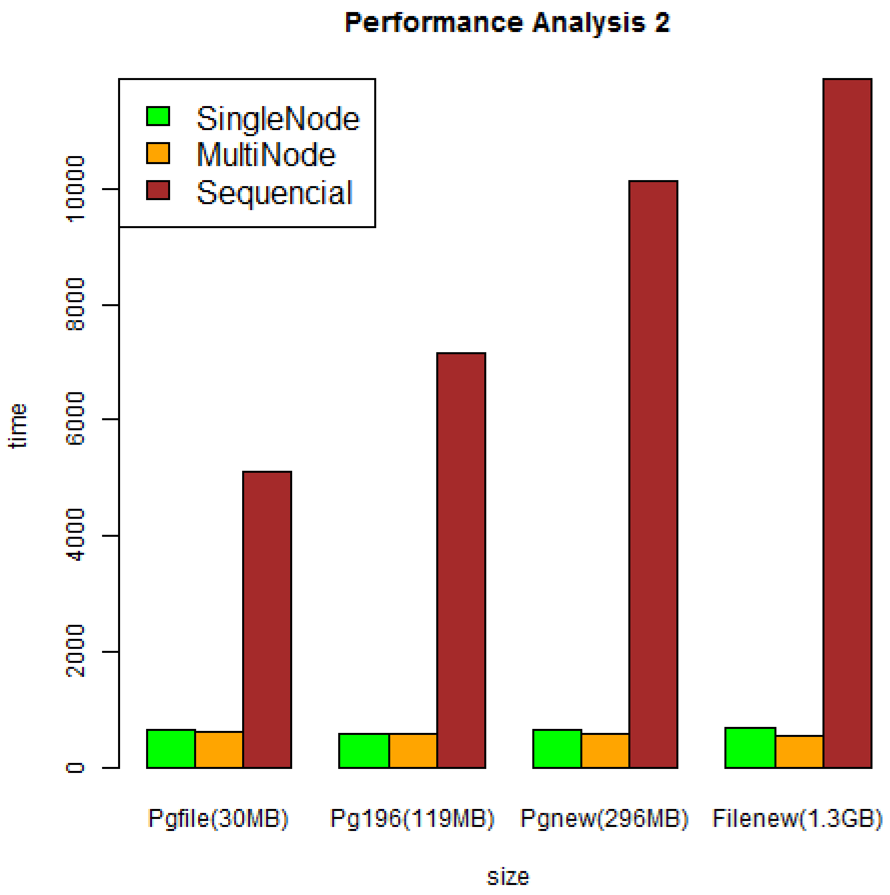
Table 4 : Single Node Execution Time – 2

File Size\Time(ms)	Time 1	Time 2	Time 3	Mean
Pgfile (30 MB)	755	522	582	620
Pg196 (119 MB)	530	592	598	573
Pgnew (296 MB)	470	727	524	573
Filenew (1.3 GB)	-	-	-	542

Table 5: Multi Node Execution Time – 2

File Size	Pgfile (30 MB)	Pg196 (119 MB)	Pgnew (296 MB)	Filenew (1.3 GB)
Time (ms)	5103	7160	10132	11918

Table 6: Sequential Execution Time– 2



Graph 2 : Performance Analysis -2

In this second portion, we go for string frequency calculation, where a specific string is searched for in the whole file in a similar manner as the previous one. Here the results of the previous thread will be compounded in rather lucid manner.

Here, for the first two cases, the performance of singlenode and multinode is comparable, with sequential being out of question here.

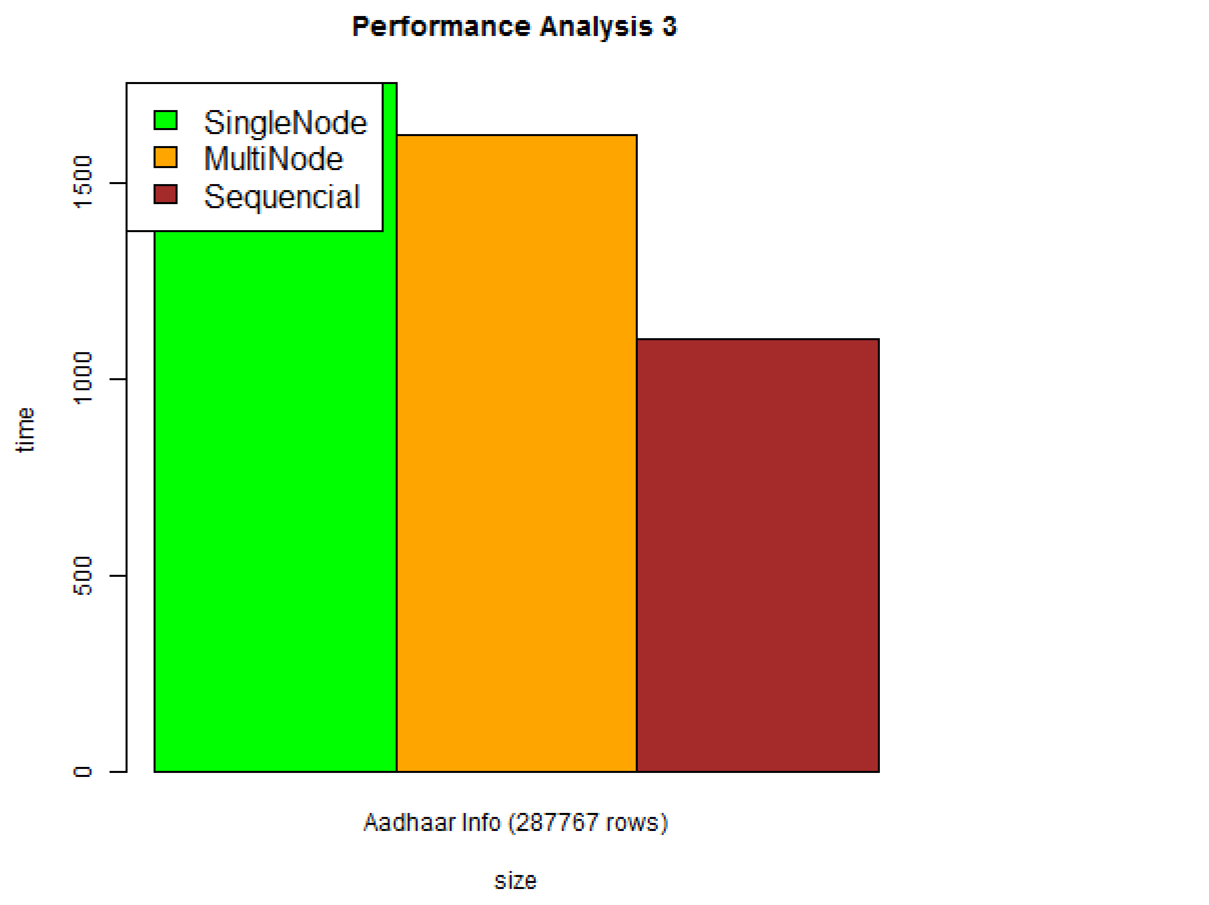
According to the graph, in the last two cases, multinode starts to improve in its performance, as this time the computation is more complex than in the previous case, and thus the overheads this time are compensated by the level of processing. The results, though directly conflicting the results of the first analysis of word frequency are justified due to the complex nature of the processing. The results get compounded as the size of the file increases. This is in sync with the expectations.

4.5 Performance Analysis 3: AadhaarCard :-

4.5.1 Aadhaar Analysis 1 (Age Group)

File Size\Time (ms)	Single Node	Multi Node	Sequential
Pgfile (30 MB)	1654	1632	1211
Pg196 (119 MB)	1769	1821	1064
Pgnew (296 MB)	1479	1806	1033
Filenew (1.3 GB)	1627	1760	1103

Table 7 : Aadhaar card data performance



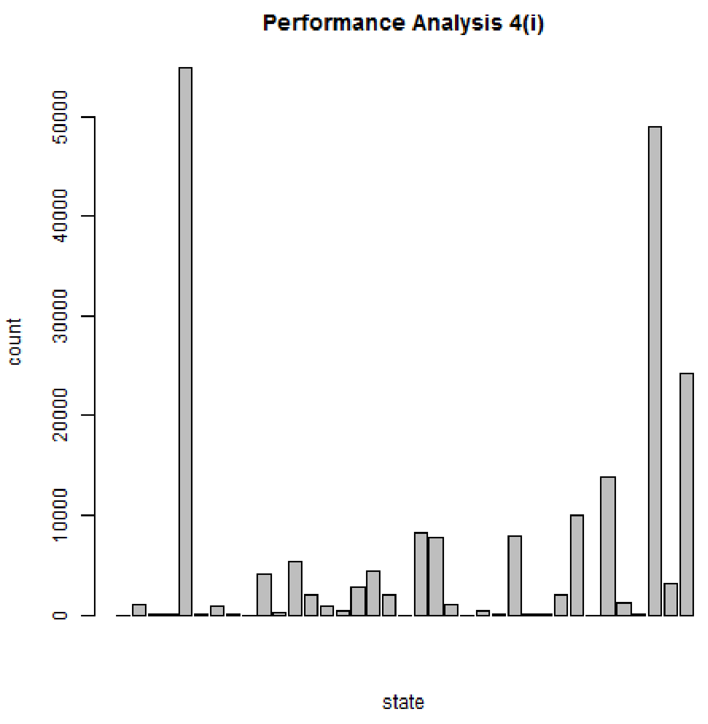
Graph 3 : Performance Analysis - 3

As we move forward with core theme of our project, data analysis of enrollment of people of all the states for the aadhaar card in a single sample month is taken. The data has lakhs of rows with details pertaining to name, city, sex, age,

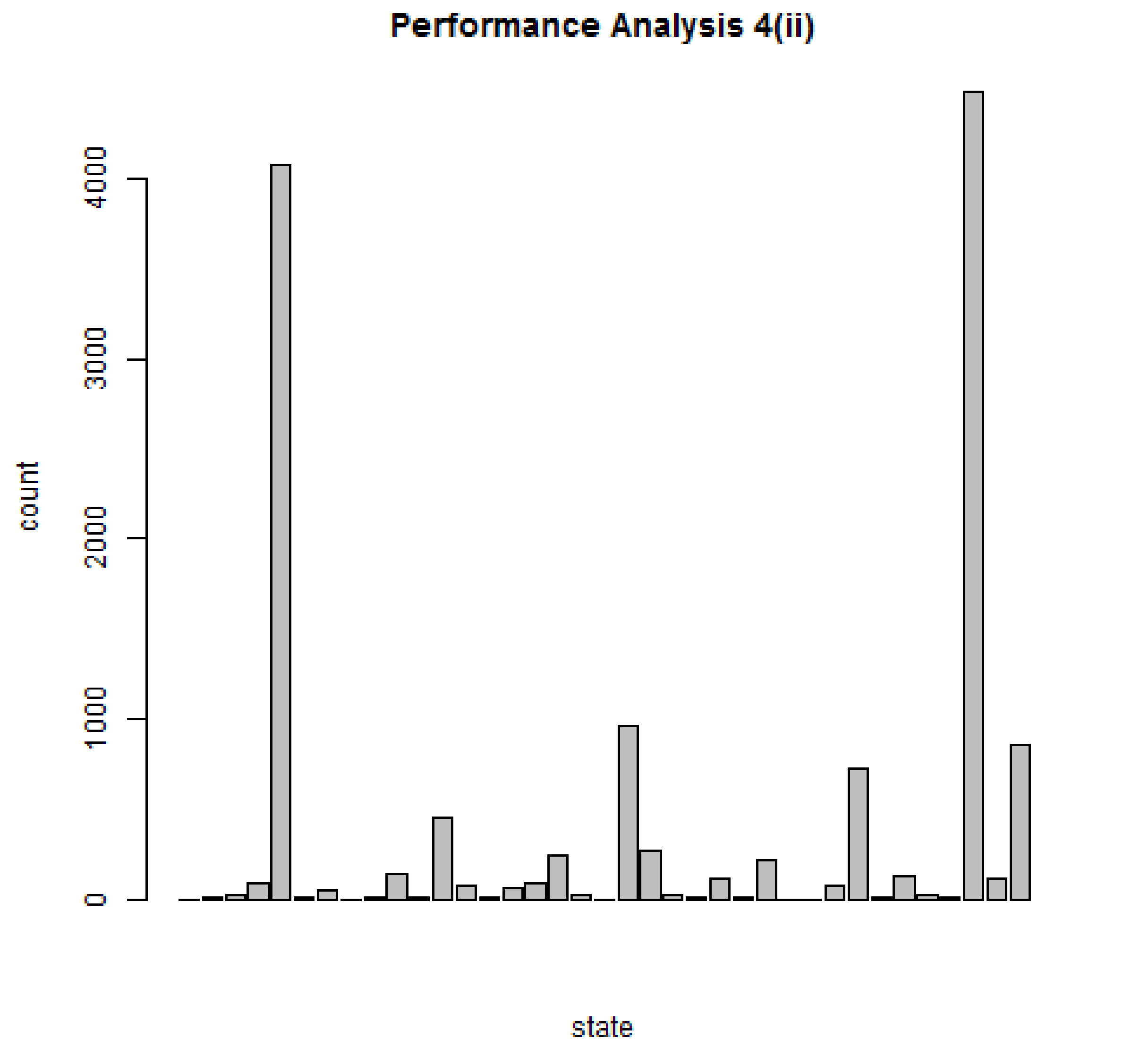
The first analysis we run is of the enrolment of the people state wise. The data is analyzed through Hadoop and the relevant data is then plotted in the form of bar graphs through R programming.

The question here is that if the data contributing the population of each and every region is ready, then the analyzed enrolled data can be compared with the aforementioned data and it can be then known that which areas or regions have less enrolment or penetration of the aadhaar. Similarly, in which areas the enrolment rate has suddenly dropped or increased. This will in turn lead to the investigation of reasons of the same and thus provide the authorities with a good reason to go ahead and tackle those area specific problems.

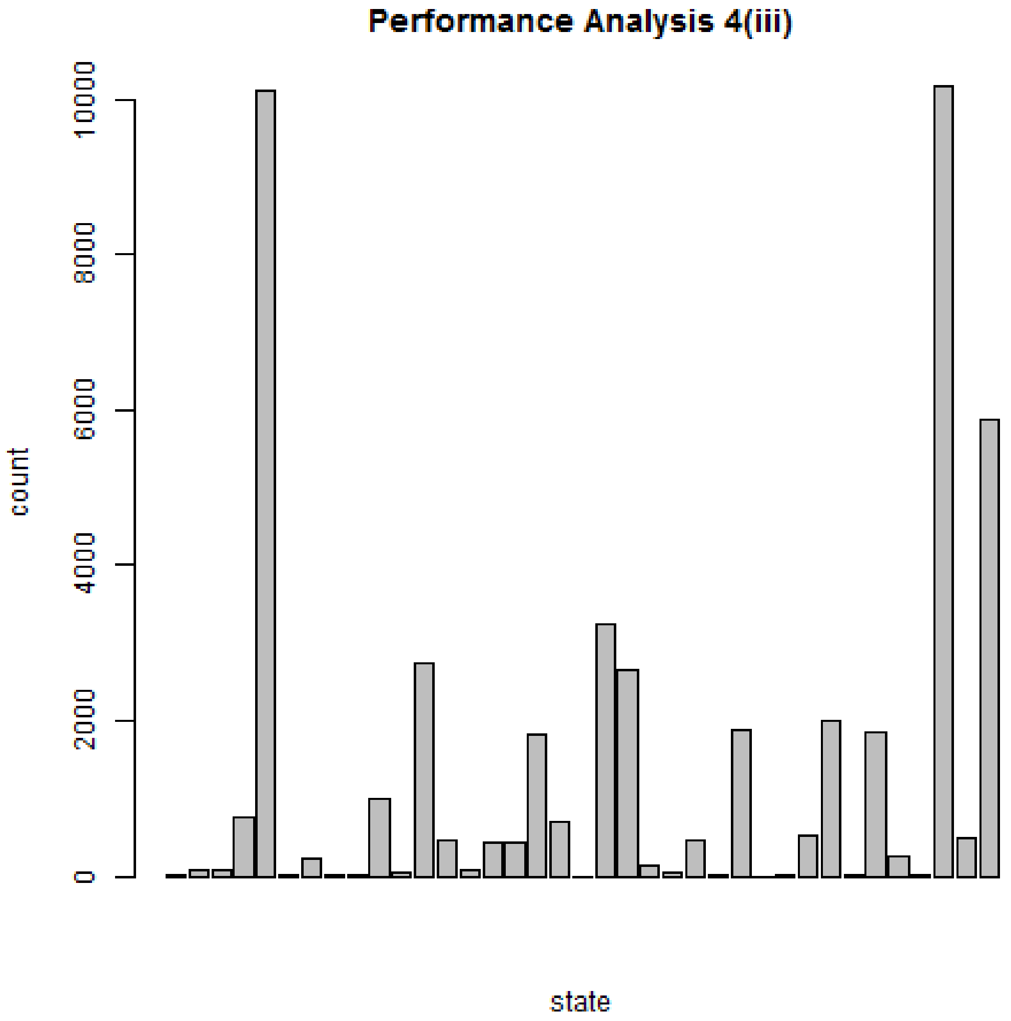
4.5.2 Aadhaar Analysis 2 (Age Group)-



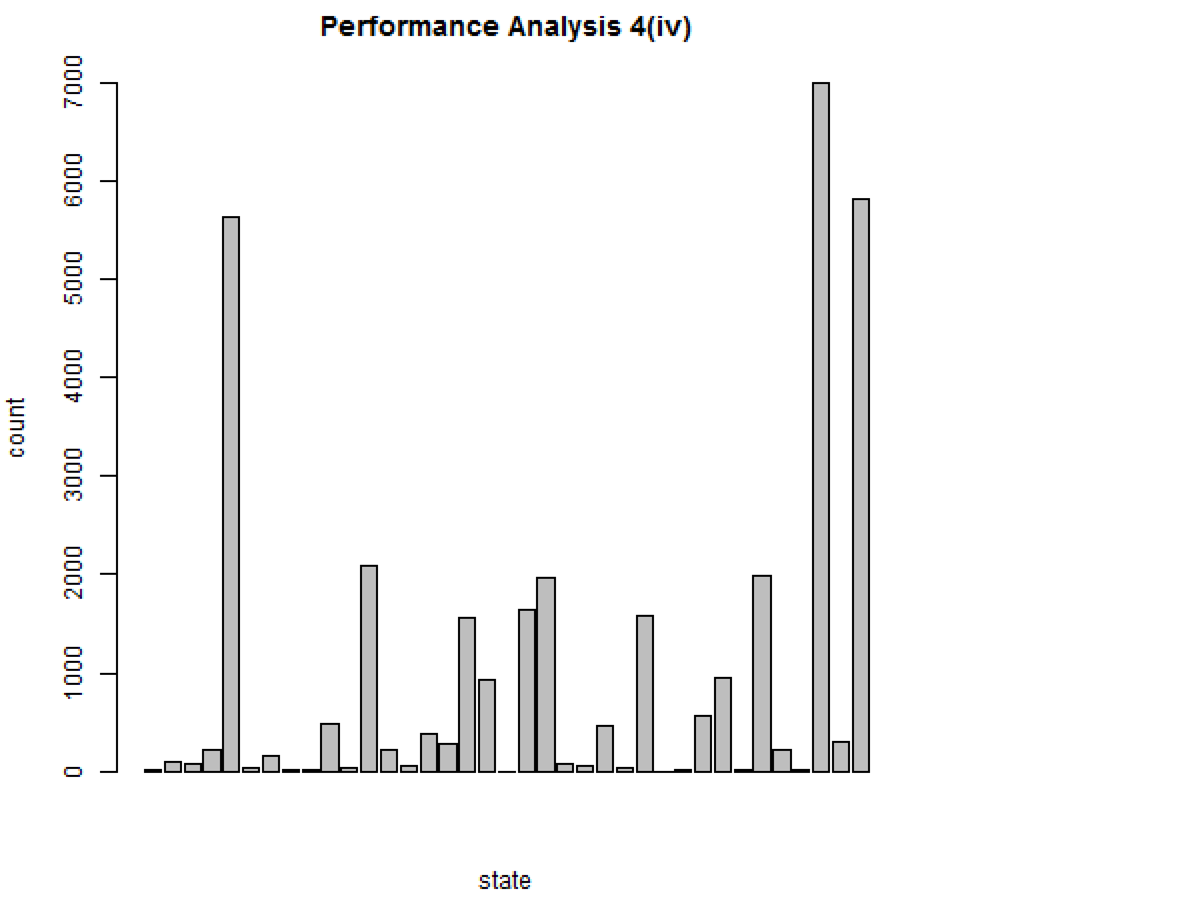
Graph 4 : Age group 1(0 - 14) data



Graph 5: Age group 2 (15-18) data



Graph 6 : Age group 3(19 -35) data



Graph 7: Age group 4(36 and above) data

As we take the previous step of aadhaar analysis forward and try to analyze some other subject of the data, data analysis of enrolment of people of all the states for the aadhaar card in a single sample month is taken. The data has lakhs of rows with details pertaining to Aadhaar generated, State, Age and Gender etc.

This second analysis we run is pertaining to the various ages of people in all the regions/states in question. The data is analyzed through Hadoop and the relevant data is then plotted in the form of bar graphs through R programming.

There are four age groups taken in the analysis-

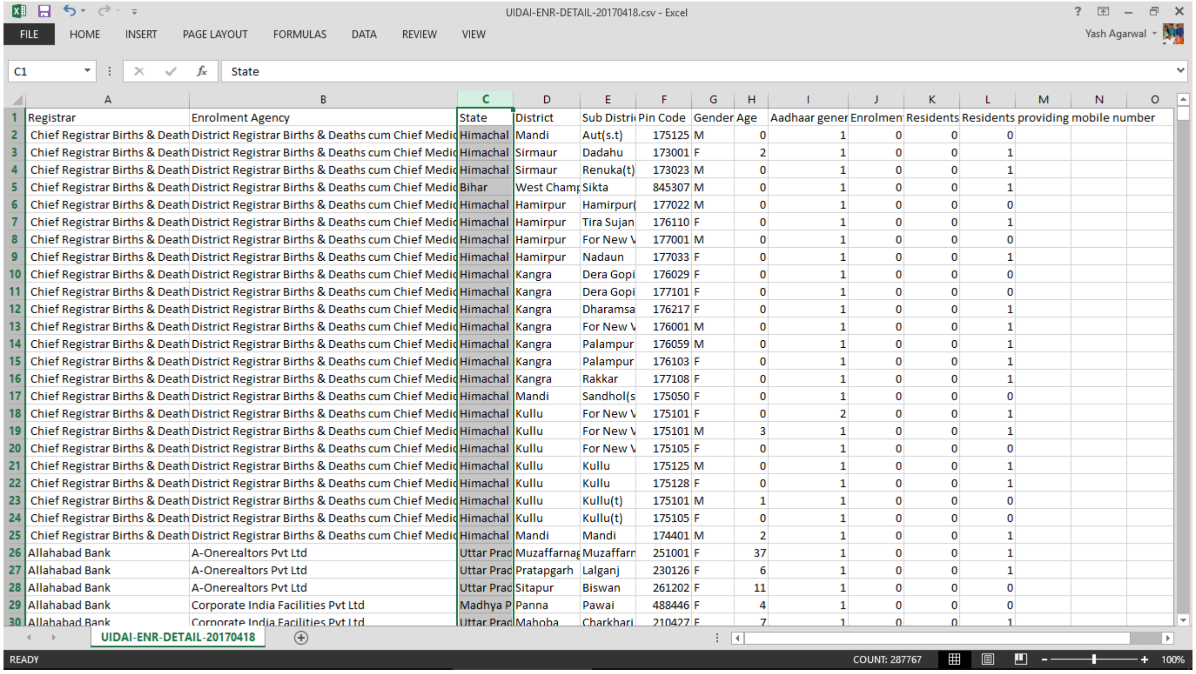
Children (0-14)

Adolescents (15-18)

Young Adults (19-35)

Senior Adults (36 and above)

The question here is that if the data contributing the population of each and every region is ready, then the analyzed data of each and every age group can be of prime importance and if genuinely compared with the aforementioned data, it can be then known that which areas or regions have population of which age group and thus the administration of the region changes with such data. Similarly, in which areas the age of the population shows some irregular pattern and why is it so. This will in turn lead to the investigation of reasons of the same and thus provide the authorities with a good reason to go ahead and tackle those area specific problems.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Registrar	Enrolment Agency	State	District	Sub Distri	Pin Code	Gender	Age	Aadhaar gener	Enrolmen	Residents	Residents providing mobile number			
2	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Mandi	Aut(s.t)	175125	M	0	1	0	0	0			
3	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Sirmaur	Dadahu	173001	F	2	1	0	0	1			
4	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Sirmaur	Renuka(t)	173023	M	0	1	0	0	1			
5	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Bihar	West Champ	Sikta	845307	M	0	1	0	0	1			
6	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Hamirpur	Hamirpur(	177022	M	0	1	0	0	0			
7	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Hamirpur	Tira Sujan	176110	F	0	1	0	0	1			
8	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Hamirpur	For New V	177001	M	0	1	0	0	0			
9	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Hamirpur	Nadaun	177033	F	0	1	0	0	1			
10	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kangra	Dera Gopi	176029	F	0	1	0	0	0			
11	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kangra	Dera Gopi	177101	F	0	1	0	0	0			
12	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kangra	Dharamsa	176217	F	0	1	0	0	1			
13	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kangra	For New V	176001	M	0	1	0	0	1			
14	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kangra	Palampur	176059	M	0	1	0	0	1			
15	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kangra	Palampur	176103	F	0	1	0	0	1			
16	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kangra	Rakkar	177108	F	0	1	0	0	1			
17	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Mandi	Sandhol(s	175050	F	0	1	0	0	0			
18	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kullu	For New V	175101	F	0	2	0	0	1			
19	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kullu	For New V	175101	M	3	1	0	0	1			
20	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kullu	For New V	175105	F	0	1	0	0	0			
21	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kullu	Kullu	175125	M	0	1	0	0	1			
22	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kullu	Kullu	175128	F	0	1	0	0	1			
23	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kullu	Kullu(t)	175101	M	1	1	0	0	0			
24	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Kullu	Kullu(t)	175105	F	0	1	0	0	0			
25	Chief Registrar Births & Death	District Registrar Births & Deaths cum Chief Medic	Himachal	Mandi	Mandi	174401	M	2	1	0	0	1			
26	Allahabad Bank	A-Onerealtors Pvt Ltd	Uttar Prad	Muzaffarnag	Muzaffarn	251001	F	37	1	0	0	1			
27	Allahabad Bank	A-Onerealtors Pvt Ltd	Uttar Prad	Pratapgarh	Lalganj	230126	F	6	1	0	0	1			
28	Allahabad Bank	A-Onerealtors Pvt Ltd	Uttar Prad	Sitapur	Biswan	261202	F	11	1	0	0	0			
29	Allahabad Bank	Corporate India Facilities Pvt Ltd	Madhya P	Panna	Pawai	488446	F	4	1	0	0	0			
30	Allahabad Bank	Corporate India Facilities Pvt Ltd	Uttar Prad	Mahoba	Charkhari	210427	F	7	1	0	0	1			

Figure 19 : Sample Data of Aadhaar

Chapter-5 CONCLUSION

5.1 Inferential conclusion

As we tread towards a more digitized world, the role of technology in our lives has become ever indispensable. The intrusion is of unimaginable scale and we as the creators must strive hard to embrace it in a positive sense. What we have tried to achieve in this project is just a small step on this road of nobility. As the struggle of internet penetration continues, the research pioneering is one trump card that'll make the future of computer science fly high in coming years. The universal identity scheme, better known as the UIDAI or the Aadhaar scheme is the flagship government programed to make our country progress towards a universal unique identity scheme for the procurement of each and every citizen centric scheme. Problems like low tax to population ratio, as in the case of direct taxes still persists. Connecting aadhaar number to each and every bank account, with the PAN number gives a leverage to the government for extensively scrutinizing the transactions. Such progressive planning is a leap that authorities have taken, but the onus of practicing the proposals thoroughly is dependent completely on the denizens. The Aadhaar data used for our calculations is not dynamic in nature and thus it does not represent changing trends in the society. The proposed mechanism can be implemented on dynamic data by the civic authorities to tackle real time problems and alleviate the distressed portion of the society through immediate actions.

5.2 Future Prospects-

The scope of our project is boundless if implemented and taken to use. As the data increases by leaps and bounds, the need to cater to the services of general public use becomes more and more potent. What we’ve tried to do with the data of aadhaar card in an intrinsic manner can be implemented on a wider scope with a larger population and can be emulated on other government schemes.

REFERENCE LIST

[1] Vignesh Prajapati, “Big Data Analytics with R and Hadoop”, UK: Packt Publishing

[2] Tom White foreword by Doug Cutting,” Hadoop: The Definitive Guide”, O’Reilly

[3] “Hadoop 2.6 Installing On Ubuntu 16.04 (Single-Node Cluster)” http://www.bogotobogo.com/Hadoop/BigData_hadoop_Install_on_ubuntu_16_04_single_node_cluster.php

[4] “Integration of R with Hadoop”

https://acadgild.com/blog/integration-r-hadoop/

[5] “Install Ubuntu 14.04 alongside Windows 8.1 in 10 easy steps” http://www.everydaylinuxuser.com/2014/05/install-ubuntu-1404-alongside-windows.html

[6] “Hadoop Tutorial: Developing Big-Data Applications with Apache Hadoop” http://www.coreservlets.com/hadoop-tutorial/

[7] “Map reduce Use case – Titanic Data Analysis” https://acadgild.com/blog/analyzing-titanic-data-with-hadoop-mapreduce/

[8] “MapReduce Use Case-YouTube Data Analysis” https://acadgild.com/blog/mapreduce-use-case-youtube-data-analysis/

[9] “Hadoop Shell Commands to Manage HDFS” https://dzone.com/articles/top-10-hadoop-shell-commands

[10] Luiz Fernando Assisl, Gilberto Ribeiro1, Karine Reis Ferreira1, Lubia Vinhas, Eduardo Llapa1, Alber Sanchez1, Victor Maus1 and Gilberto Camara, "Big data streaming for remote sensing time series analytics using MapReduce" , INPE - National Institute for Space Research, 2016

[11] Bo LI, Hui ZHAO and Zhen Hua LV "Parallel ISODATA Clustering of Remote Sensing Images Based on MapReduce", International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2010

[12] Veershetty Dagade, Mahesh Lagal, Supriya Avadhani and Priya Kalekar,"Big Data Weather Analytics Using Hadoop", International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) Volume 14 Issue 2, APRIL 2015

[13] Ryan McDonald, Keith Hall and Gideon Mann, "Distributed Training Strategies for the Structured Perceptron"

[14] Subramaniyaswamy, Vijayakumar, Logesh and Indragandhi, "Unstructured Data Analysis on Big Data using Map Reduce", 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

[15] Konstantinos F. Xylogiannopoulos, Panagiotis Karampelas and Reda Alhajj, "Frequent and Non-Frequent Pattern Detection in Big Data Streams: An Experimental Simulation in 1 Trillion Data Points", IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016

[16] Hsinchun Chen, Roger H. L. Chiang and Veda C. Storey, "Business Intelligence And Analytics: From Big Data To Big Impact", Business Intelligence Research

[17] Matthew S. Eastin, Nancy H. Brinson, Alexandra Doorey and Gary Wilcox, "Living in a big data world: Predicting mobile commerce activity through privacy concerns", January 2016

[18] “Aadhaar Card Data Set”, Dated (22-Apr-2017 i.e. one month data)

https://data.uidai.gov.in/uiddatacatalog/dataCatalogHome.do