

**VIRTUALIZING HADOOP MAP REDUCE  
FRAMEWORK USING VIRTUAL BOX TO  
ENHANCE THE PERFORMANCE**

*Dissertation submitted in fulfilment of the requirements for the Degree of*

**MASTER OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

By

**MADHURI**

**41300069**

Supervisor

**MR. VIRRAT DEVASER**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

Dec 2016



**TOPIC APPROVAL PERFORMA**

School of Computer Science and Engineering

Program : 1792::M. Tech- CSE(Computer Science and Engineering)(Part Time)

COURSE CODE : CSEPS46

REGULAR/BACKLOG : Backlog

GROUP NUMBER : CSEBGD0331

Supervisor Name : Virrat Devaser

UID : 14591

Designation : Assistant Professor

Qualification : \_\_\_\_\_

Research Experience : \_\_\_\_\_

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Madhuri	41300069	2013	BLI42	9465019860

SPECIALIZATION AREA : Programming-I

Supervisor Signature: \_\_\_\_\_

PROPOSED TOPIC : Data analytics on web characteristics of user behavior to analyze shopping patterns in hadoop

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	7.17
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.17
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.50
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.33
5	Social Applicability: Project work intends to solve a practical problem.	6.83
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	7.00

PAC Committee Members		
PAC Member 1 Name: Janpreet Singh	UID: 11266	Recommended (Y/N): Yes
PAC Member 2 Name: Harjeet Kaur	UID: 12427	Recommended (Y/N): Yes
PAC Member 3 Name: Sawal Tandon	UID: 14770	Recommended (Y/N): Yes
PAC Member 4 Name: Vikas Verma	UID: 11361	Recommended (Y/N): Yes
PAC Member 5 Name: Dr. Ramandeep Singh	UID: 14105	Recommended (Y/N): Yes
DAA Nominee Name: Kanwar Preet Singh	UID: 15367	Recommended (Y/N): Yes

**Final Topic Approved by PAC:** Data analytics on web characteristics of user behavior to analyze shopping patterns in hadoop

**Overall Remarks:** Approved

PAC CHAIRPERSON Name: 11011::Rajeev Sobti

Approval Date: 25 Oct 2016

11/29/2016 1:22:53 PM

## **ABSTRACT**

---

Hadoop Distributed File System is used for storage along with a programming framework MapReduce for processing large datasets allowing parallel processing. The process of handling such complex and vast data and maintaining the performance parameters up to certain level is a difficult task. The paper works towards an approach of improving the performance by virtualizing the Hadoop framework using virtual box. This can be demonstrated by establishing virtual machines for the slave nodes in the physical machine where the master node also resides. The configuration of Hadoop virtual cluster is implemented to achieve high performance.

## DECLARATION STATEMENT

---

I hereby declare that the research work reported in the dissertation entitled “Virtualizing Hadoop Map Reduce Framework Using Virtual Box To Enhance The Performance” in partial fulfillment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr. Virrat Devaser I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

*Signature of Candidate*

**Madhuri**

**41300069**

# SUPERVISOR'S CERTIFICATE

---

This is to certify that the work reported in the M.Tech Dissertation entitled “**Virtualizing Hadoop MapReduce Framework Using Virtual Box To Enhance The Performance**”, submitted by **Madhuri** at **Lovely Professional University, Phagwara, India** is a bonafide record of his / her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

Mr. Virrat Devaser

**Date:**

**Counter Signed by:**

1) **HoD's Signature:** \_\_\_\_\_

HoD Name: \_\_\_\_\_

Date: \_\_\_\_\_

2) **Neutral Examiners:**

(i) **Examiner 1**

Signature: \_\_\_\_\_

Name: \_\_\_\_\_

Date: \_\_\_\_\_

(ii) **Examiner 2**

Signature: \_\_\_\_\_

Name: \_\_\_\_\_

Date: \_\_\_\_\_

## ACKNOWLEDGEMENT

---

I would like to take this noble opportunity to extend my deep-sense of gratitude to all who helped me a lot directly or indirectly during the development of this dissertation proposal.

Fore-mostly I want to express wholehearted thank to my mentor, **Mr. Virrat Devaser** for being such a worthy mentor and best ever adviser. His precious advice, motivation and critics proved the sources of innovative ideology, encouragement and main cause behind the successful completion of this dissertation. I am very much obliged to all the lecturers of computer science and engineering dept. for their heartfelt encouragement and support.

I also extend my sincerest thanks and gratitude towards all mates for their consistent support and invaluable suggestions provided at that time when I required the most. I am very grateful to my lovable family for their support, love and prayers.

**Madhuri**

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE NO.</b>
Inner first page	i
PAC form	ii
Abstract	iii
Declaration by the Scholar	iv
Supervisor's Certificate	v
Acknowledgement	vi
Table of Contents	vii
List of Abbreviations	ix
List of Figures	x
List of Tables	xi
<b>CHAPTER 1: INTRODUCTION</b>	<b>1</b>
<b>1.1 Big Data</b>	<b>1</b>
<b>1.1.1 Three V's of Big Data</b>	<b>1</b>
<b>1.2 Hadoop</b>	<b>2</b>
<b>1.2.1 Why Hadoop</b>	<b>2</b>
<b>1.2.2 HDFS</b>	<b>5</b>
<b>1.2.3 MapReduce</b>	<b>7</b>
<b>1.3 Virtualization</b>	<b>9</b>
<b>1.3.1 Types of Virtualization</b>	<b>10</b>
<b>1.3.2 Benefits of Virtualization</b>	<b>11</b>
<b>1.3.3 Various Platforms for Virtualization</b>	<b>12</b>
<b>CHAPTER 2: REVIEW OF LITERATURE</b>	<b>13</b>
<b>CHAPTER 3: PRESENT WORK</b>	<b>18</b>

<b>3.1</b>	<b>Scope of the Study</b>	<b>18</b>
<b>3.2</b>	<b>Objective of the Study</b>	<b>19</b>
<b>3.3</b>	<b>Research Methodology</b>	<b>19</b>
	<b>3.3.1 Sources of Data</b>	<b>20</b>
	<b>3.3.2 Research Design</b>	<b>20</b>
 <b>CHAPTER 4: RESULTS AND DISCUSSION</b>		<b>21</b>
<b>4.1</b>	<b>Configuration of Hadoop</b>	<b>21</b>
<b>4.2</b>	<b>Creating Hadoop Cluster</b>	<b>23</b>
<b>4.3</b>	<b>Performance</b>	<b>25</b>
	<b>4.3.1 Performance of Hadoop in Virtual Cluster</b>	<b>25</b>
	<b>4.3.2 Performance with Physical Machine</b>	<b>29</b>
	<b>4.3.3 Performance Metrics of Cluster</b>	<b>32</b>
 <b>CHAPTER 5: CONCLUSION AND FUTURE SCOPE</b>		<b>34</b>
 <b>REFERENCES</b>		



## LIST OF ABBREVIATION

---

API	Application Programming Interfaces
BYOD	Bring Your Own Device
ETL	Extract Transforms Load
HDFS	Hadoop Distributed File System
OS	Operating system
VCC	Virtual Client Computing
VMM	Virtual Machine Monitors
VM	Virtual Machine

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>FIGURE DESCRIPTION</b>	<b>PAGE NO.</b>
1.1	Three Vs of Big Data	1
1.2	Architecture of Hadoop System	3
1.3	HDFS Architecture	6
1.4	Job and Task Tracker	9
1.5	Virtual Machine Monitors	10
3.1	Research Design	20
4.1	Hadoop Performance for DataSet 1 - I	26
4.2	Hadoop Performance for DataSet 1 – II	26
4.3	Hadoop Performance for DataSet 1 - III	27
4.4	Hadoop Performance for DataSet 2 - I	27
4.5	Hadoop Performance for DataSet 2 – II	28
4.6	Hadoop Performance for DataSet 3 - I	28
4.7	Performance for DataSet 1 - I	29
4.8	Performance for DataSet 1 - II	29
4.9	Performance for DataSet 2 - I	30
4.10	Performance for DataSet 2 - II	30
4.11	Performance for DataSet 3 - I	31
4.12	Performance for DataSet 3 – II	31
4.13	Time Comparision between Real and Virtual Clusters for WordCount Operation	32
4.14	Time Comparision between Real and Virtual Clusters for Average Operation	33

## LIST OF TABLES

<b>TABLE NO.</b>	<b>TABLE DESCRIPTION</b>	<b>PAGE NO.</b>
<b>Table 1</b>	Time Taken by Physical and Virtual Cluster to Perform WordCount	32
<b>Table 2</b>	Time Taken by Physical and Virtual Cluster to Perform Average	33

## Checklist for Dissertation-II Supervisor

Name: \_\_\_\_\_ UID: \_\_\_\_\_ Domain: \_\_\_\_\_

Registration No: \_\_\_\_\_ Name of student: \_\_\_\_\_

Title of Dissertation:

\_\_\_\_\_

- 
- Front pages are as per the format.
  - Topic on the PAC form and title page are same.
  - Front page numbers are in roman and for report, it is like 1, 2, 3.....
  - TOC, List of Figures, etc. are matching with the actual page numbers in the report.
  - Font, Font Size, Margins, line Spacing, Alignment, etc. are as per the guidelines.
  - Color prints are used for images and implementation snapshots.
  - Captions and citations are provided for all the figures, tables etc. and are numbered and center aligned.
  - All the equations used in the report are numbered.
  - Citations are provided for all the references.
  - Objectives are clearly defined.**
  - Minimum total number of pages of report is 50.
  - Minimum references in report are 30.

Here by, I declare that I had verified the above mentioned points in the final dissertation report.

Signature of Supervisor with UID

# CHAPTER 1

## INTRODUCTION

---

### 1.1 Big Data

Every single minute, Facebook users share nearly 2.5 million of data. Twitter users post tweet nearly 300,000 times. Users of Instagram post new pictures nearly about 220,000 .YouTube users upload 72 hours of new videos. Email users send over 200 million mails. Amazon generates over \$80,000 in online shopping and Google receives queries about 4 billion from global users of around 2.4 billion [17]. So this is how data is increasing, referred to as —Data Tsunami. Such huge datasets and rich mix of data type and format is commonly known as —Big Data. This data is so vast and complex that it is beyond the scope of any traditional data processing application to handle this kind of data within a tolerable elapsed time. Hadoop came as a solution to this problem and helped to extract, transform, analyse and store this big data.

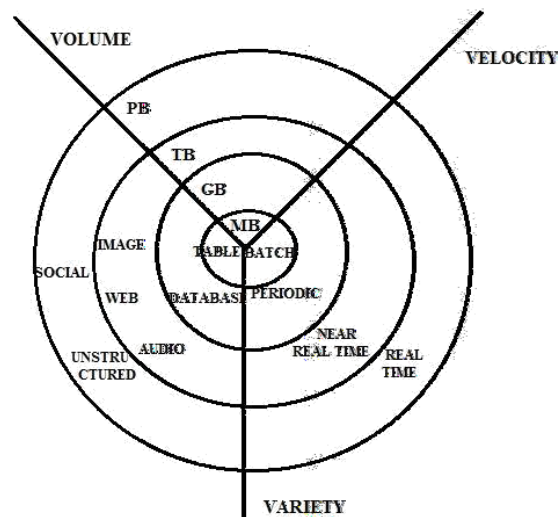


Fig1.1: 3-Vs of Big Data

#### 1.1.1 Three Vs of Big Data

As shown in Fig 1.1, Big Data is about three different Vs

#### Volume

Data storage is growing exponentially because now a days data is not only about text only, it consists of videos, music and large images on our social media channels. It is very common to have Terabytes and Petabytes of the storage system for enterprises.

### **Velocity**

There is tremendous growth in data. Data explosion has changed how we look at the data. There was a time when we used to believe that data of yesterday is recent. There is lot of data movement and updates are within fractions of the seconds. This high velocity data represent Big Data.

### **Variety**

Multiple formats are available to store data e.g. database, excel, csv, access or for the matter of the fact, it can be stored in a simple text file. The real world have data in many different formats.

## **1.2 HADOOP**

Hadoop is a popular, open source, java based software framework that provides a platform for distributed processing and analysis of large datasets across clusters of computers. It is used for scalable, fault tolerant, flexible, reliable and distributed computing. Hadoop was derived from Google's MapReduce, a distributed system and Google File System (GFS), a distributed file system. Hadoop can increase the processing power and the storage by combining many computers into one.

Architecture of Hadoop system is shown in Fig 1.2. The main component of Hadoop includes:

- Hadoop Distributed File System (HDFS)
- Map-Reduce

### **1.2.1 Why HADOOP**

Hadoop enables big data applications for both analytics and operations. Hadoop is one of the popular, fastest growing technologies and a key component of the next generation data architecture which provide a great processing platform and scalable distributed storage as well. The advantages of using Hadoop framework are as follows.

## Hadoop

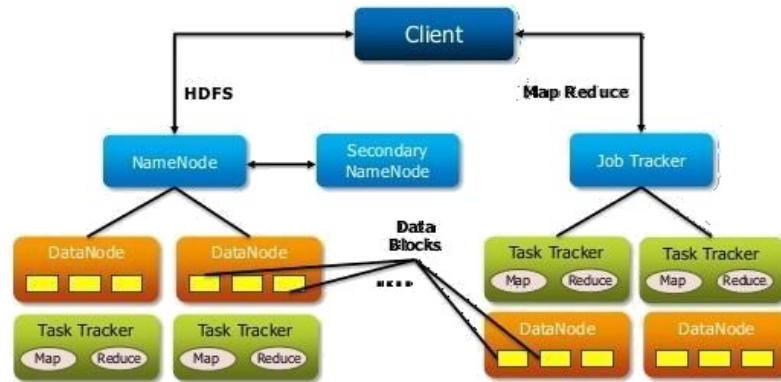


Fig 1.2: Architecture of Hadoop System

1. **Low Cost:** Hadoop is a freely available, open source framework so there is no cost required to get this software. On the other hand, the large datasets and information will be stored by having minimal hardware requirement.
2. **Flexibility of data:** Hadoop can handle different variety of data i.e. structured, semi structured or unstructured and also the compatible with the different resources from where the data is coming. The increasing data from social media and networked devices is the key consideration.
3. **Processing Speed :** Hadoop is mainly used for processing large amount of data as its computation power is very high because the map and reduce jobs are executed on various data chunks of different virtual machine at a time which will automatically reduce the computation time and increase the processing power.
4. **Scalability:** Hadoop is highly scalable system. As the data in organization increases it can handle that by adding large numbers of nodes with the least overhead being a distributed approach.
5. **Failure Tolerant:** Hadoop stores huge data across the various machines in distributed system. Multiple (generally three) copies of that data is stored for high availability and when any node goes down, the same data will be recovered from another node providing the protection against fault.
6. **Fast access:** Hadoop is able to handle large amount of unstructured data efficiently using MapReduce jobs with HDFS in very economical time limit.

Hadoop framework also has some limitations which are as follows

1. **Suitability for small data:** Hadoop framework is not suitable for small datasets. As this platform is compatible and designed for large amount of data so it is not efficient to use it for the organizations having small quantity of data.
2. **Security issues:**Hadoop's security is the main challenge. Because it is the key component for any organization to keep the data safe. In Hadoop encryption is missing at storage and network levels. Although steps are taken towards enhancing the security of Hadoop framework, using Kerberos authentication technique is one of them.
3. **Skill gap:** It is easy to search programmer who are comfortable in SQL programming than to MapReduce programming skills.
4. **Frequent data change:** Hadoop is not suitable for the system where frequent data changes occur so where the basic RDBMS operations like data insertion, deletion and updation is required for frequent changes in data processing, Hadoop is less efficient.

The Apache Hadoop open source software library is a framework which is used for its distributed processing of massive data sets across clusters of computers .a single server is multiplied to many numbers of machines providing local storage capacity and processing of data. It provides high scalability, reliability and distributing computation of data. Hadoop main module includes:

- Common: It is the also termed as Hadoop core as it provides the basic and essential services to all the other modules in the framework.
- HDFS: It is the java based distribution file system that holds very huge amount of data and offers easy access to that data.
- MapReduce: It is the data processing technique and programming model for distributed computing.

Other projects related to Hadoop are as follows:

1. Avro : A project used for data serialization
2. Cassandra: It is a scalable database used for online web and mobile applications with no single point of failure.
3. Hbase: It is a NoSql, distributed database which provides big tables capabilities of Hadoop.



4. Chukwa: It is a system for data collection used for managing in large distributed systems.
5. Hive : It is a data warehouse infrastructure used for querying and analysis of large datasets
6. Mahout: It is an open source machine learning library to provide distributed analytics capabilities.
7. Pig: It provides engine for data flow in parallel.

As discussed earlier, Hadoop has two main components

1. HDFS (Hadoop Distributed File System)
2. MapReduce

### 1.2.2 HDFS

The Hadoop Distributed File System is a scalable and reliable file system that is suitable for distributed storage of data. HDFS is highly cost efficient as it is developed on low cost hardware and is fault tolerant. It can store a very huge amount of data and provide the streaming access to that data .The files of such large data will be distributed across various machines and stored redundantly as blocks in the system.

#### **HDFS Architecture**

HDFS is a master – slave architecture containing two nodes as shown in Fig 1.3

1. Name Node (As single master node)
- 2 .Data node (As Multiple slave nodes)

**Name Node:** It stores the metadata of the file system. It manage the namespace and allows the client to directly access to the files. Name node performs namespace operations i.e. Opening, renaming and closing of files and directories.

**Data Node:** It stores the actual data. Each cluster will have several data node to perform the task assigned by the Name node. Data node performs block operations such as create, delete and replicate as directed by name node. It will provide service to the read and write request of client.

HDFS exposes the namespace of a file system and then stores that user data into files. The file thus created will be divided into multiple blocks (normally block

size is of 64 Mb). Then the replication of blocks is processed and each block is assigned to data node.

As the name node maintain the metadata for every file, in its main memory. The mapping between the name of files stored, blocks and their data nodes will be determined. So that the any request of client to perform any action will pass through Name node and then name node directs the appropriate data node to perform the task. Client can interact with the data node directly to process the request and perform operations on the file system.

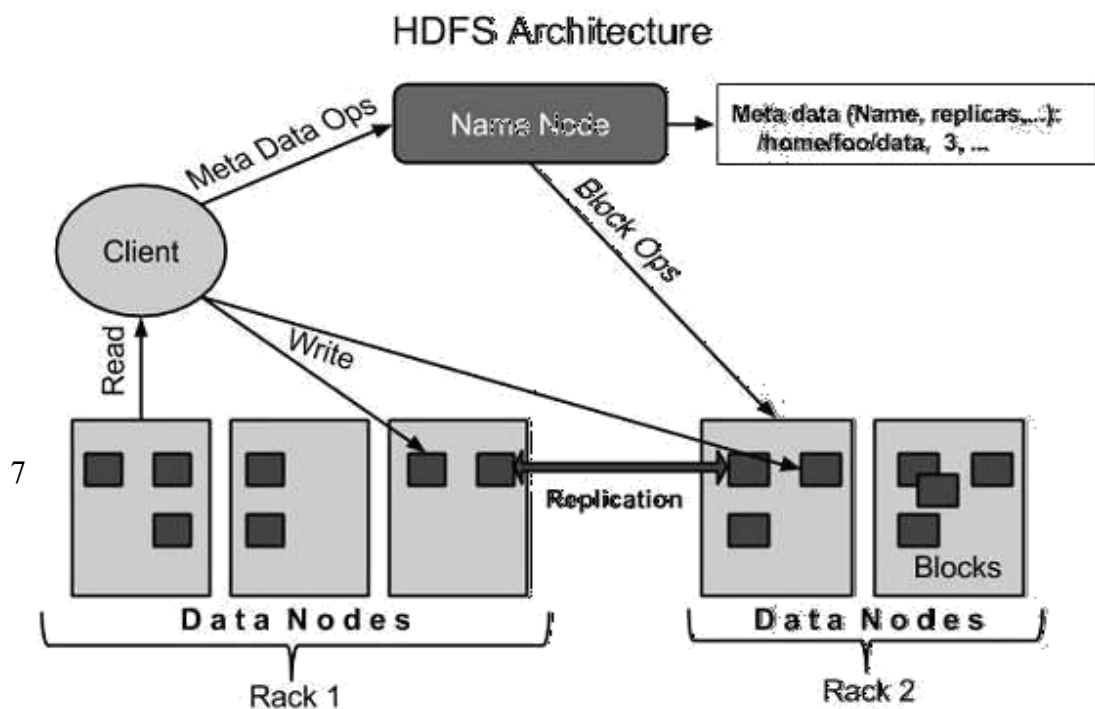


Fig 1.3 HDFS Architecture

### Relationship Status Between Name Node and Data Node

The status of connectivity between Name node and Data node can be tracked using Heartbeat Mechanism. Heartbeat is a signal indicating that whether the data node is alive or not. It carries information about storage capacity, used space in fractions and number of data transfer currently occurring etc. In Heartbeat mechanism the Name node assign task to the data node. In response Data node sends heartbeat signal or message to the Name node confirming that it is in active condition. But in case when the Name not do not receive any heartbeat message from data node then it

will mark it as a dead Data node and did not assign any task to that node in future and removes it from the cluster. The same mechanism also takes place in job tracker and Task tracker.

### **Various Characteristics of HDFS**

- **Huge Data sets:** HDFS will run application having large amount of datasets. This dataset will typically ranges from gigabytes to petabytes in size. The data is stored within distributed machines and commodity hardware.
- **Reliability:** The main goal of HDFS is to provide reliability of storage data. So that any machine failure can occur but data will be still available for use. Failure can be detected using the heartbeat signals.
- **Data Coherency:** It follows the write once Read many model which provides simple coherence of data and gives high throughput access.
- **Rebalancing of blocks:** HDFS creates many replicas (typically 3) of data nodes and place it to the appropriate data nodes. It re-balance the data block in each data node considering the space requirement i.e. data block is moved to another data node for balancing the load or by creating the replicas of data block for space utilization in data node. So it provides fault tolerance and high throughput access to the data.
- **Data integrity:** The data integrity is carried at block level. The client computes the checksum along with data and compares it with the corresponding checksum. If it does not matches than that blocks is removed and data is fetched from its replica.

### **1.2.3 MapReduce**

MapReduce is a processing model and scalable framework used for processing large amount of datasets in parallel on multiple nodes. It follows the divide and conquer technique for processing data. The main components of MapReduce comprise of Map stage and reduce stage .In which map function will map and process one pair of dataset to another in form of tuple (key/value pair) and the reduce function will take the output of map function as input and merge all the intermediate tuples into smaller tuples using corresponding key value and hence reduce the dataset.

**Map Stage:** Map performs the function of data collection and processing. The input taken will be divided into smaller chunks.

Map ( ) function:

Input -----> Output  
(k1, v1)-----> List (k2, v2)

**Reduce Stage:** Reduce function will take the output of map function i.e. the intermediate values as input and combines them to give a final output as a reduced dataset.

Reduce ( ) function:

Input -----> Output  
(k2, List (k3,v2)----->list(k3,v3)

Shuffling is also a step in MapReduce framework which is used to transfer the data from mappers to reducers.

MapReduce consists two components

1. Job Tracker ( A central component)
2. Task Tracker (Distributed Component)

**Job Tracker:** Job tracker is a master that is responsible for the scheduling and processing of all the jobs. Whenever a job is received by job tracker ,it will assign that job to the task tracker and coordinates the execution process of the job. It is run on Master node i.e. Name node.

**Task Tracker:** A task tracker will be assigned to perform map and reduce functions for the particular job by job tracker. It will give report to the job tracker about the progress of the assigned task. Each task tracker has various number of slots (map and reduce function) which is required to perform a task. The balance of map to reduce task is also an important consideration which is managed by JVM. The Task tracker run on slave node i.e. Data node.

In Fig 1.4 the whole function of job tracker and task tracker is shown. The job tracker will assign job to task tracker submitted by a user and that job is processed in task tracker. The Task tracker will update the job Tracker with current status

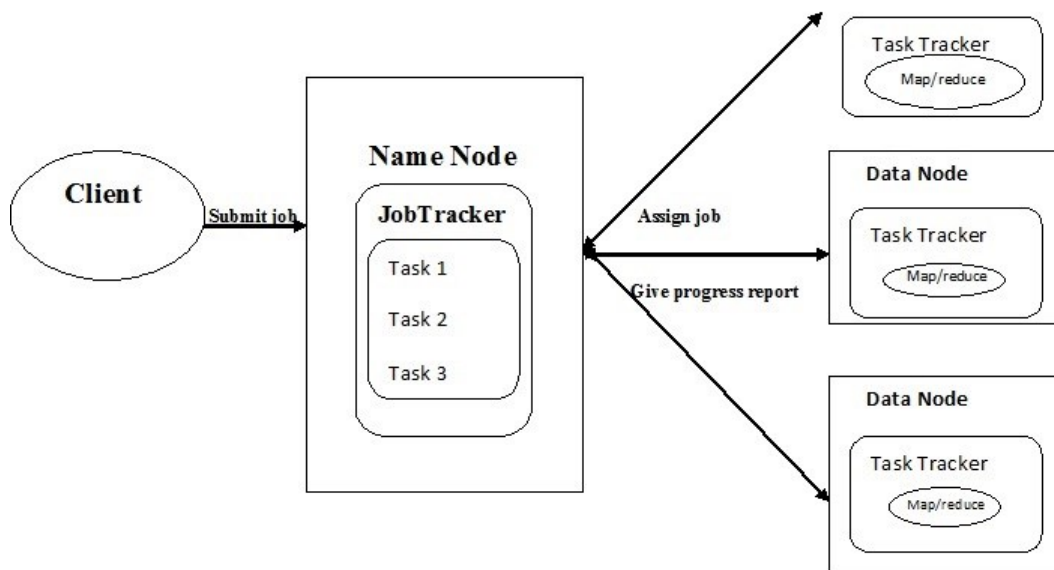


Fig 1.4: Job and Task Tracker

### 1.3 Virtualization

Virtualization is a concept of dividing the resource of the same hardware to multiple operating system (same or different) and applications at same time, achieving higher efficiency and resource utilization. In this the two isolated operating system can run simultaneously. Virtualization has proven as an effective tool for IT industry as it increases its agility, scalability, flexibility and performance ratio. It is transforming the way to utilize techniques and resources. Hypervisor or virtual machine monitor is the manager who carries out the process of virtualization as shown in Fig 1.5

1. Bare metal or Native
2. Hosted Hypervisor.

**Bare metal hypervisor** run directly on top of hardware to have more control power over hardware.

**Hosted hypervisor** run on the top of conventional operating system along with some native processes. It will differentiate the guest and host operating system running on the single hardware.

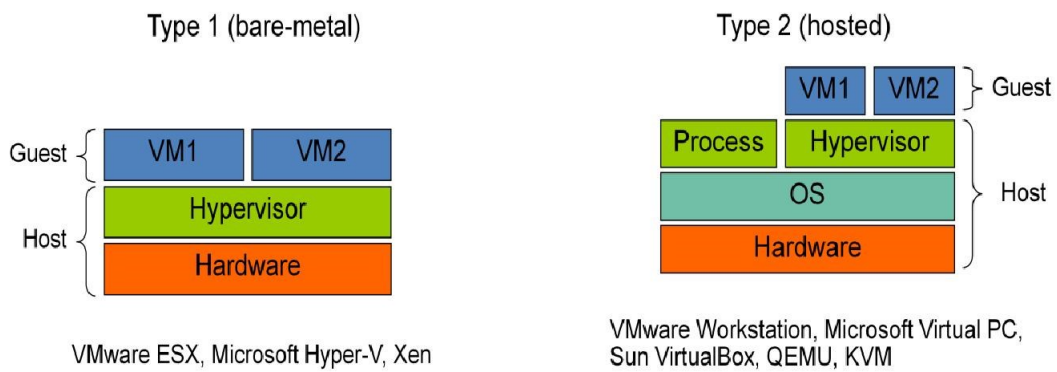


Fig 1.5: Virtual Machine Monitors

### 1.3.1 Types of Virtualization

Virtualization can be utilized in different areas to enhance performance and resource sharing. So various types of virtualizations are as following

**Server Virtualization:** Server virtualization is a process in which multiple OS and applications run on a single server so that resources of that single host will be divided among all the operating system and the various applications. It brings the resource utilization and the cuts the cost by having only one server in place of many.

**Application Virtualization:** Application virtualization enables the end user to access the application from remotely located server. Application is not installed on every user desktop but is available on user's demand. So it is very cost efficient for the organizations.

**Desktop Virtualization:** It is similar to server virtualization. In desktop virtualization the workstation with all its application is virtualized. The hypervisor contain the customization and preferences of an application. It runs on a centralized mechanism so user can access the desktop from any location .It provides higher efficiency and the cost reduction.

**Storage Virtualization:** Storage virtualization is a technique in which the storage from the multiple hardware devices will be combined to act as a single storage device. In this the storage space by SAN and NAS is virtualized along with the disk file of virtual machine. So it provides the disaster recovery management by replication.

**Network Virtualization:** Network virtualization is a concept in which the all the network from different network devices are combined into a single network called as

a virtual network so that its bandwidth will be divided to channels which will be given to the server or device.

### 1.3.2 Benefits of Virtualization

- **Server Consolidation:** In this existing application will be moved to the some servers. As it is one time event, it will reduce the cost and increase the space utility.
- **Cost Reduction:** Virtualization is very appropriate in cutting cost of the organizations. As many servers are combined into one. The only single server will provide the same functionality to the user. Hence the cost of extra hardware, storage and maintenance overhead will be reduced to a greater extent which saves both time and money.
- **High Availability:** The feature of high availability of virtual machine is provided so that in case on any kind of failure of a particular virtual machine the host will provide its replication from another virtual machine. It will automatically reduce the downtime and cut the cost.
- **Security:** As centralized management is available, there is very less chance to data loss. It also provides control over interrupts and device DMA to enhance security.
- **Live Migration:** In this the running virtual machines can be moved to some another host which provides the higher availability of data and hence achieve the greater performance.
- **Legacy hardware:** In this the legacy system will run as a virtual machine on a new hardware and provides the same functionality as the previous server hardware which was outdated. In this way it cuts the cost and increased the performance.
- **Disaster Recovery:** Its recovery process is independent of any OS or hardware. So if any disaster occurs the recovery will be done through image based backup of virtual machines by snapshot mechanism
- **Load balancing:** Virtualization provides the facility to balance the load of virtual machines of over loaded server by transferring them to underutilized server, which makes efficient utilization of servers.

### 1.3.3 Various Platforms for Virtualization

**VM ware:** VMware is a company providing virtualization solution. Its products dominate the virtualization market. It provides full virtualization. VM ware's main product is Vsphere which uses VMwaresESXi hypervisor. It is highly mature, robust, extremely stable and high performer.

**Citrix:** It owns the world's mostly used cloud vendor product Xen. Xen is open source virtualization software that supplies paravirtualization and supports different processors. It allows various guest OS to be carried on single physical machine. In Xen Domain-0 is known as guest operating system. Whenever Xen software boots, Domain-0 boots automatically. In the Linux, Xen is well known software that used for windows virtualization. Amazon use xen for its elastic compute cloud.

**Red hat enterprise:** It is the most successful company in terms of open source technique. It provides reliable and high performance products. Its main product is RHEA (RedhatRnterprise virtualization) which uses the companies own hypervisor KVM (Kernel Based Virtual Machine) .It provides the functionality of hardware virtualization to the user. It is highly scalable system

**Microsoft Hyper V:** Only Microsoft have non linux platform for virtualization. Its Commercial product isHyper V. Hypervisor can run on any OS supported by platform of hardware. It goes well with the Microsoft's product virtual PC. Management is easy in it.

**Oracle Virtual Box:** virtualization product of Sun Microsystems under acquisition of oracle is Virtual Box. Virtual box is open source and provides high performance to the enterprise. It is very portable and supports different host operating system such as windows, Linux, Solaris and Mac.



## CHAPTER 2

### REVIEW OF LITERATURE

---

In today's era data is growing very fast in structured (relational data), semi structured (xml data) or unstructured (audio, videos, data from social sites) form. So to deal with this huge amount of data Hadoop is used. Even though Hadoop has many built in features for data analysis and processing, virtualization is integrated with Hadoop for better resource utilization, flexibility to add or remove cluster and high and higher security.

Many papers have been analysed and explored relevant to proposed work. The papers include various features of Hadoop which are utilized for processing big data and integration of Hadoop with virtualization for enhancing the overall functionality. The necessity for scalable and efficient solution for the component failure and the consistency of data Google file system (GFS) [Ghemawat et. al] and Map Reduce [Dean & Ghemawat][1] came into existence. Their basic feature was to store the data in commodity servers so that computation can be performed where the data is stored without transferring data over network for processing. These two termed as a base for the Apache's project Hadoop [Apache Hadoop][ ] explains the architecture of Hadoop with its main component HDFS (Hadoop Distributed File System) and MapReduce .

[Robert D. Schneider][2] The book revolves around big data, MapReduce and Hadoop environment and explains the relationship between them. The main concentration of the book is towards the working of MapReduce and Hadoop. MapReduce is a programming framework which processes the data using Divide and Conquer technique. In this the large and complex data is divided into smaller chunks in map phase. These units will be processed in parallel providing fast access. The intermediate result will be generated by integration or merging (reduce phase) and will be taken as input. The process is repeated till the desired and reduced output is generated. Using MapReduce alone sometimes becomes complex so Hadoop is used to manage all the process. Hadoop cluster have main components as Master Node which include (Job tracker, Task Tracker, Name Node), Data node, Worker Node. Hadoop provides high scalability, availability, multi tenancy; Easy management of

task, flexibility etc but it can suffer from single point of failure i.e. if the Master node fails.

[Soundarabai, Aravindh S et.al] [3] Explains the approach of processing huge amount of data using Hadoop's capabilities. They have used the functionality of Map Reduce for the computation by distributing large data sets across multiple nodes in small units(Mapping) and then combining or aggregating the output from various nodes to a desired set (Reducing).A business logic on 2GB dataset has been executed. Hive and Pig models are used to work on dataset as pig is used to create MapReduce code. All the queries are executed on Hadoop as well as centralized system. Hadoop gives better performance as compared to centralized system in terms of execution time for each query.

Virtualization has been used from long times for sharing resources among different environment. Virtualization is implemented at different level i.e. Desktop, system, storage, network which brings enhanced security, flexibility, high availability, scalability. [Daniel A. Menasce][4] This paper took forward the issues of system virtualization and the performance issues. According to the paper virtualization can be of two types Full virtualization or Para virtualization. Full virtualization is carried out by direct implementation of the application code and the binary transformation. On the other hand Para virtualization provides the interface to the virtualized component that is similar but not as hardware. It is considered to be more efficient than the full virtualization as the guest is aware with the fact that they are running on the virtual machine. Some issues related to queuing network model in virtualized environment were also discussed in the paper - Desktop Virtualization and Storage Solutions Evolve to Support Mobile Workers and Consumer Devices is gaining the highest level of interest and attention in organizations. Desktop virtualization is investigating by the leaders because it can increase the workforce flexibility with teleworking and with desktop images for mobile users. IT enabled the new planes of mobility, security and cost reduction by rethinking desktop virtualization. Desktop virtualization is the Centralized desktop virtualization in which desktop OS is abstracted from the endpoint and run as a virtual machine in a system.

Sponsored by:[ Citrix and NetApp Brett Waldman, Ashish Nadkarni] [5]Desktop virtualization is rapidly increasing and expanding to new devices growing nearly 12% year over year. Desktop virtualization increasing the business form PC to Data Centric and even up to Cloud. IT need to move from the PC Centric world to the

BYOD and BYMOD and managing the individual pc components and hard drives where the large and cooperate data is stored and managed and giving access to users.

In one more technical paper[6] it summarizes that three different Hadoop applications are run on the different host operating systems. Then the performance of native and several VMware vsphere clusters configurations was compared. The result shows that if there is only single machine per host then the result is same with the native machine but if we increase the virtual machines per hosts by two or three then result is varied. The time is achieved by 13% than the native. Hadoop provides a platform for building distributed systems for large amount of data storage and analysis. In this failure is recovered by the data replication that across racks of hosts. The scheduler executed multiple jobs but still need to virtualize due to several reasons which helps to make to maintain the level of resource utilization. The result of virtualizing Hadoop on v sphere 5.1 for 32 host cluster achieves good performance.

[Bou][7] Final master project developed in Barcelona Supercomputing Centre. This project focuses on the virtualization of Hadoop environments and the design of a piece of software for automatizing the configuration of the shared resources for Hadoop environments. In addition, this project uses a modified internal Hadoop scheduler that adapts the available resources in the cluster for running jobs according to time restrictions. With the adapted internal scheduler and the virtualization capabilities the software developed in this project provides a flexible and self-adaptive service for Hadoop environments. Hadoop has emerged as one of the most popular implementations for MapReduce in the research community as it is open source and is part of the Apache Software Foundation. But not only Academics are taking part of the Hadoop project. Hadoop started with its MapReduce implementation but grew up very fast and nowadays includes other projects to provide the required infrastructure, such as HDFS (Hadoop Distributed File System) that provides the distributed file system required for a MapReduce implementation, becoming in this way a complete software solution.

Big organizations faced with the growing costs and security concerns created by the quantity and diversity of personal computers can deploy a more secure, cost-effective and flexible desktop environment using the Vblock Fast Path Desktop Virtualization Platform[8], provided by VCE: the Virtual Computing Environment Company. Built on VCE's V block Infrastructure Platform integrates best-of-breed technology from industry leaders Cisco, EMC and VMware—the Vblock Fast Path

Desktop Virtualization Platform is a purpose-built desktop virtualization solution for delivering desktops as a managed service. Enterprises find themselves beset with challenges on all sides as they try to manage the ever-evolving desktop environment.

[Jeffrey Shafer, Scott Rixner et.al] [9] analyzes the performance of HDFS with the various issues associated with it. The issues include bottleneck in architectural implementation of hadoop, limitations in portability and the last one is assumptions taken by the hdfs about the portability i.e the way native platform handles the resources of storage as the input output scheduler and the filesysytem of every native is different from each other. As mentioned in this paper the efficiency of mapreduce applications in hadoop environment is increased by the optimization of hdfs. The performance of the hdfs is a challenge in portability. While preserving portability using apllication level input output scheduling the performance of hdfs can be improved.

[VidyasagarS.D ] [10] explains the role of hadoop environment in processing a huge amount of structured or unstructured data. Hadoop is an open source , highly flexible framework used for analysis, storing and computation of various kind of data. It has inbuilt functionality to handle fault tolerance, data replication etc. reducing cost parameters. It explained the architectural design (master/slave) of the hadoop enivronment having name node(master) and data nodes(slave) that provide high throughput access. [Prashant D. Londhe ,Satish S. Kumbhar et. al] [11] In this paper the hadoop framework is described i.e. how the huge datasets are processed under the hadoop environment. It also focuses on the various features of hadoop to bring the availability, fault tolerance and scalability. In the implementation part Hadoop cluster is formed, configuration of master and slave node ios done and then data is stored and processed on the HDFS. Future work includes adding the indexing to the data nodes so that the process of mapping huge number of small files becomes efficient. [Dali Ismail, Steven Harris] [12] has taken Hadoop as a solution to the problems related to the big data in their paper “Performance Comparison of Big Data Analysis using Hadoop in Physical and Virtual Servers”. Hadoop gives performance enhancement by providing high fault tolerance, low computational cost, analyzing and processing large data sets and hence provind high throughput data access. The comparison of the performance is achieved using three benchmarking test i.e. TestDFSIO ,MapResuce Sort and Gridmix benchmarking test. The paper concludes that considering the various performance parameters (network bandwidth and RAM

size etc.) the performance of hadoop cluster is lower than the physical machine. This may be possible for the reason of overhead of various virtual machines on the cpu of physical host.

Jeff Buell [13] explores benchmarking test based on the performance of virtualised hadoop environment VMware vsphere5. In this case study the performance of the hadoop application with the various configurations of virtual machines is compared with the native configuration. The test results show that the average performance of the native and virtual configuration is different by only 4%. As a functionality of virtualization to create multiple nodes of hadoop cluster for each host, the better performance is achieved automatically by it over the native. Aditya B. Patel, Manashavi Birla, Ushma Nair [14] addresses the problem of big data using hadoop and Map reduce environment. This paper explores the big data problem and its solution using HDFS for storing the large data with processing the data parallelly in the MapReduce framework. The experimental results in the word count problem shows the decrease in time with the increase in the number of the nodes and the second test results that the hadoop cluster is highly scalable and gives better results on increasing the no. of nodes. Hae-Duck J. Jeong, WooSeok Hyun et al [15] took a survey on the problem of handling big data considering hadoop as a solution. As various network attacks caused severe damage to the network resources. So this paper focuses on anomaly teletraffic intrusion detection systems based on the hadoop platform to detect them earlier. It also provides solution to the problem to carry out the intrusion detection system based on the attribute like variety, cost, volume of storage, IDS and velocity.

Arantxa Duque Barrachina and Aisling O'Driscoll [16] provides a journal on big data in which hadoop is used for the categorising and processing of the same service calls and identifying the similar calls from a large dataset providing faster access. The solution provides the evaluation of the subset taken from the technical support data with the output and the validity of the clustering algorithms used is examined.

### **3.1 Scope of the Study**

The study mainly focuses on integrating the features of virtualization to the Hadoop environment. Hadoop Map Reduce is used to compute the huge amount of complex data in tolerable elapsed time, because all the processes performed to analyze data will be done in parallel on different nodes. Although Hadoop provides many functionality but when it comes to manage and providing resources for each upcoming request, it will become a difficult task to handle it. So with the built in features of Hadoop there is a need to virtualize it. Virtualizing the Hadoop cluster adds new features to it.

1. It brings elasticity as cluster can be expanded or reduced by adding or removing the nodes on demand. The whole process is very fast.
2. A physical cluster will be shared between multiple virtual cluster so that the physical cluster will be reused which enables resource utilization.
3. Roles of task tracker and data nodes will be separated into different machines for achieving high security as they both have their own access authorization.
4. After virtualization of physical cluster, cloning of single image (for e.g. cloning of data node) can be performed which reduce the cost and enhance the performance.

Hadoop Map Reduce processes huge amount of data quickly in very less time, thus making system highly scalable. Large files are divided into smaller parts so that each part can run parally to achieve high performance. Hadoop Map Reduce works very well but when there is need of high end resources. Because to configure the hadoop on another machine takes a time and this process becomes very time consuming and response to the fast requests becomes slow. To solve this problem virtualization is used that helps in the easily creation of new node and configuration of the hadoop.

## **3.2 Objective of the Study**

The objective of the study is to integrate the concept of virtualization with the features of Hadoop to bring resource utilization. Following lists the objectives for the present work.

- Study Hadoop environment and its various features.
- Configuration of Hadoop for single and multi-node setup as well.
- To study virtualization concept with comparison of various platform of virtual machines and selecting one among them for the given problem.
- Integration of virtualization with Hadoop environment.
- Performance Analysis of the virtual cluster in the given setup.

## **3.3 RESEARCH METHODOLOGY**

Research methodology includes the tools used in study to bring out the solution for a problem. Hadoop MapReduce along with virtual machine is used to bring the high performance, scalability, efficiency, fault tolerance and elasticity. As we have already discussed the various features of Hadoop as well as virtualization technique in detail. So following are the steps, also shown in Fig 3.1, carried out to use given tools in the problem and bring the solution in form of performance.

- As the concept and tools of virtualization have been explained. The next step involves configuration of the physical and virtual cluster and then their performance is examined
- The study of various machines took place for the cluster formation. Integration of machine with cluster is done using virtual box. The configuration of various nodes (single or multimode) in Hadoop is established.
- Process of assigning task to master node is done which further given to slave node in a virtual cluster. Efficiency of the nodes is mapped in terms of time and cost.
- Ubuntu tool is used as a framework in the problem. Virtual box is used for virtualizing cluster and then comparison is done between physical and the virtual cluster.

- For performance evaluation with word count and average calculation problem will be processed in Map Reduce framework and the efficiency is checked.

### 3.3.1 Sources of Data

The present work is made on four different datasets. All data sets are taken online with confirmation that each consists of at least one string field and one numeric field. String field has been utilized for performing word count operation while average operation was applied on numeric values.

DataSet 1: Data of players and their categories (52.1 MB)

DataSet 2: Health issues of persons including vision, hearing and oral problems (53.8 MB)

DataSet 3: Environmental data related to pollution (56.2 MB)

DataSet 4: Census data (61.2 MB)

### 3.3.2 Research Design

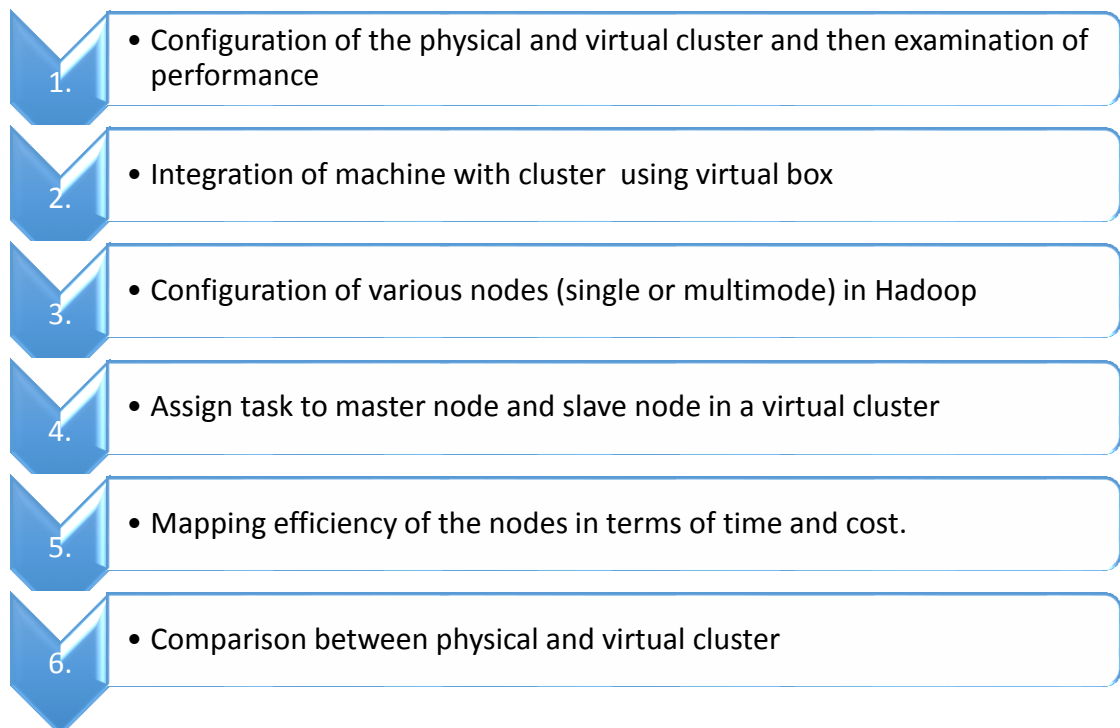


Fig 3.1: Research Design



## CHAPTER 4

### RESULTS AND DISCUSSION

---

#### 4.1 Configuration of Hadoop

Hadoop can be installed on GNU/Linux platform. Therefore, Linux operating system is needed for setting up Hadoop environment. In case operating system other than Linux, install virtual machine and have Linux inside it. To install Hadoop in the Linux environment, Linux is required to be setup using Secure Shell, ssh. Following are the steps for setting up the Linux environment.

##### Make Hadoop User

Following are the steps to create a user:

- Use command “su” to open the root.
- Use “useradd username” command to make user from root account.
- Use command “su username” to open existing user account.

In Linux terminal use following commands to create a user.

```
$ sudo addgroup Hadoop
$ sudo adduser --ingroup Hadoop hduser
$ sudo adduser hduser sudo
```

##### SSH Setup and Key Generation

In order to perform different operations on a cluster like starting, stopping, distributed daemon shell operations SSH setup is required. Give public-private key pair for a Hadoop user and share it with other different users. Following commands are used for generating a key value pair using SSH. Copy the public keys from id\_rsa.pub to authorized\_keys, and give the owner read and write permissions to authorized\_keys file. To setup ssh certificate-

```
$ ssh-keygen -t rsa -P "
$ cat ~/.ssh/id_rsa.pub>> ~/.ssh/authorized_keys
```

```
$ ssh localhost
```

### **Installation of Java**

Java is prerequisite for Hadoop. Check the presence of java in system using following command

```
$ java -version
```

If Java is present, it will give version details of the JDK, JRE otherwise install Java. In order to make Java available to all the users, move it to the location “/usr/local”. Set up PATH and JAVA\_HOME variables.

### **Downloading Hadoop**

Download and extract Hadoop from Apache software foundation using the following commands.

```
$ su
password:
# cd /usr/local
# wget http://apache.claz.org/hadoop/common/hadoop-2.7.3/
hadoop-2.7.3.tar.gz
# tar xzf hadoop-2.7.3.tar.gz
# mv hadoop-2.7.3/* to hadoop/
# exit
```

Hadoop has three different Operation Modes

- Local/Standalone Mode
- Pseudo Distributed Mode
- Fully Distributed

Perform following steps

```
$ sudo tar Hadoop-2.7.3.tar.gz -C /usr/local
$ cd /usr/local
$ sudo mv Hadoop-2.7.3 Hadoop
$ sudo chown -R hduser : Hadoop Hadoop
```

### **Setup Hadoop environment variables**

Open file - .bashrc and add following content in end of that file.

```
#Hadoop variables
export JAVA_HOME=/usr/lib/jvm/jdk/
```

```
export HADOOP_INSTALL=/usr/local/Hadoop
```

```
export YARN_HOME=$HADOOP_INSTALL
```

Open another file named Hadoop-env.sh and modify java\_home

```
export JAVA_HOME=/usr/lib/jvm/jdk/
```

### **Check the version of Hadoop**

```
$ Hadoop version
```

This will ensure that Hadoop is configured on the system and ready to use.

## **4.2 Creating Hadoop Cluster**

### **Virtual machine installation**

Oracle Virtual-box VM has been used to create virtual machine. It is a software collection for maintaining and creating the VMS. Oracle VM VirtualBox is an open source virtualization software that can be installed on x86 systems like Windows, Linux, Mac, or Solaris. After the installation of Virtualbox, virtual machines are created to run guest operating systems, Linux in present work. It is configured on the host operating device and then run as an application. Different devices are virtualized and are started to decrease the load.

In this, three virtualized Machines were created with the base memory 512 MB and maximum storage for each virtual machine is 8GB. To the Reflexive loading administration, actively defined disk file created that is Virtual.

### **Configurations for the Virtual Machine:**

For network, settings in virtual machine needs to be changed. On system install three network utilities - Vtun, uml utilities and Bridge utils. By default Network address translation Adapter already exists in the system while bridge and tap networks are created. After successful network setting on each machine configure Ssh on both the machines.

At the booting time, the setup of the bridge and tap device is done automatically in each machine.

### **Creating the Virtual Machines (Cloning):**

Creating the virtual machine or cloning the machine making the replica of existing one and it is the direct copy of the machine from which it is created. The

cloned machine has the same name and ip addresses from which it is cloned. To change the name of cloned machine it is necessary to change in the different files. For change the name files that are used:

/etc/hosts: in which the address of the master and slave nodes are mentioned.

/etc/host name: to change the host name.

/etc/network/interface: to change the network address.

### **Core-site.xml**

```
<Property>
<name>hadoop.tmp.dir</name>
<Value>/app/Hadoop/tmp</value>
<Description>A base for other temporary
directories</description></property> <Property>
<name>fs.default.name</name>
<Value>hdfs://master:54310</value>
<Description>The name of the default file system
</description> </property>
```

### **Mapred-site.xml**

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

### **Configuring the Capacity Scheduler:**

Capacity Scheduler was improved from beginning by running ant package. The Obtained jar file of capacity scheduler was placed in Hadoop/build/cont rib/ folder. The Hadoop master node was then configured to use capacity scheduler instead of the default scheduler.

### **Mapred-site.xml**

```
<Property>
```

```

<name>mapred.job.tracker.taskScheduler</name>
<value>org.apache.hadoop.Mapred.capacityTaskScheduler</value>
<Description>
The scheduler which is to be utilized by the jobtracker
</description>
</property>

```

HADOOP\_CLASSPATH in conf\_/hadoop-env.sh by identifying the capacity-scheduler.jar.

Below commands are used to run the MapReduce program:

- In /usr/local/hadoop HADOOP-PATH is configured by setting the path.
- To format Name node: the name node formatted to startup the cluster.
- HADOOP\_PATH /bin/hadoop name node -format
- To start hadoop : hadoop daemon job tracker, task tracker, name node, data node
- And secondary Name Node started.
- To start hadoop :/bin/start-all.sh
- To start each node individually:
- Hadoop\_Path/bin/hadoop-daemon.sh start.
- <Daemon-name>
- File that is used to run the map reduce application is first copied into the HDFS.
- \$HADOOP\_PATH/bin/hadoop dfs-copyFromLocal <local-path>

<Hdfs-location>

- Executing Map Reduce: if the data files in HDFS, the MapReduce job is executed.
- \$HADOOP\_PATH/bin/hadoop jar<program-name>
- <input file location><output file location>

## 4.3 Performance

Now, Hadoop in virtual clusters are appropriately configured according to requirements.

### 4.3.1 Performance of Hadoop in Virtual Cluster

Performance of Hadoop in virtual cluster using four different datasets have been

evaluated and presented in Fig 4.1 to Fig 4.6

DataSet 1:

```
15/04/16 22:37:22 INFO mapreduce.Job: Job job_local102355853_0001 completed successfully
15/04/16 22:37:23 INFO mapreduce.Job: Counters: 38
  File System Counters
    FILE: Number of bytes read=1547778
    FILE: Number of bytes written=2557287
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=7970552
    HDFS: Number of bytes written=391850
    HDFS: Number of read operations=13
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=4
  Map-Reduce Framework
    Map input records=37853
    Map output records=572213
    Map output bytes=5822467
    Map output materialized bytes=503389
    Input split bytes=102
    Combine input records=572213
    Combine output records=28830
    Reduce input groups=28830
    Reduce shuffle bytes=503389
```

Fig 4.1: Hadoop Performance for DataSet 1-I

```
master [Running] - Oracle VM VirtualBox
na@master: /usr/local/hadoop/share/hadoop/mapreduce
Total time spent by all reduces in occupied slots (ms)=16296
Total time spent by all map tasks (ms)=47864
Total time spent by all reduce tasks (ms)=16296
Total vcore-seconds taken by all map tasks=47864
Total vcore-seconds taken by all reduce tasks=16296
Total megabyte-seconds taken by all Map tasks=49817756
Total megabyte-seconds taken by all reduce tasks=16687194
Map-Reduce Framework
  Map input records=74189
  Map output records=75298
  Map output bytes=2542955
  Map output materialized bytes=611444
  Input split bytes=97
  Combine input records=75298
  Combine output records=17450
  Reduce input groups=17450
  Reduce shuffle bytes=611444
  Reduce input records=17450
  Reduce output records=17450
  Spilled Records=34900
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=1003
  CPU time spent (ms)=3930
  Physical memory (bytes) snapshot=217628672
  Virtual memory (bytes) snapshot=886699712
  Total committed heap usage (bytes)=133632888
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  ID_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=2260630
File Output Format Counters
```

Fig 4.2: Hadoop Performance for DataSet 1-II

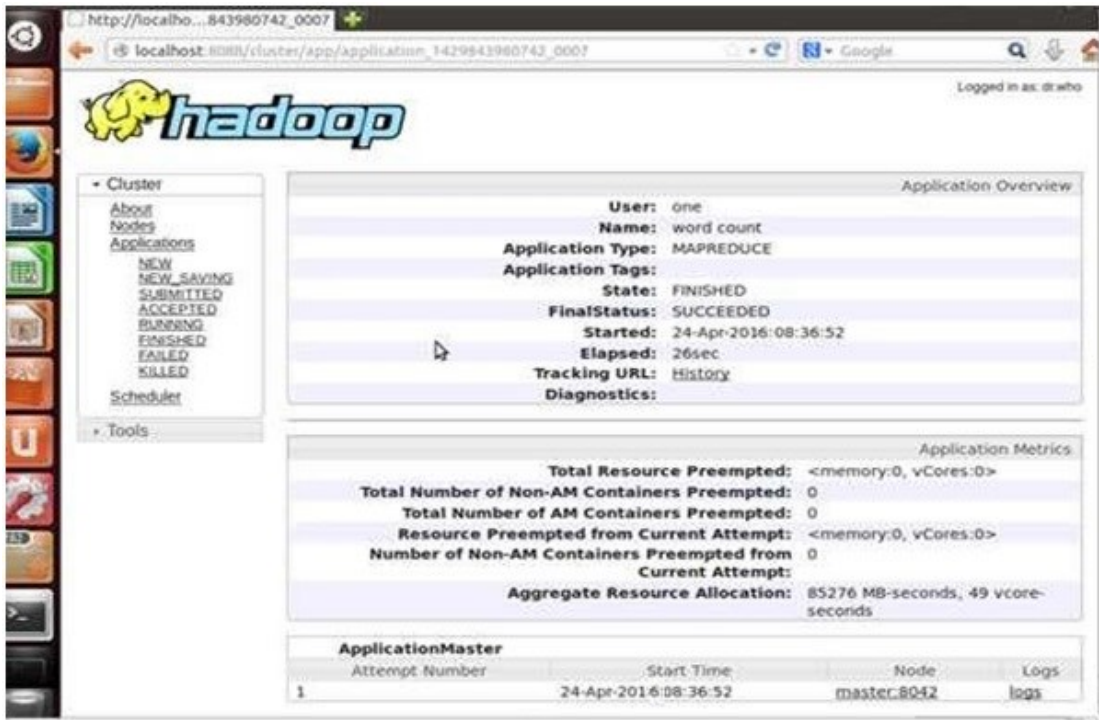


Fig 4.3: Hadoop Performance for DataSet 1-III

Data Set 2:



Fig 4.4: Hadoop Performance for DataSet 2-I

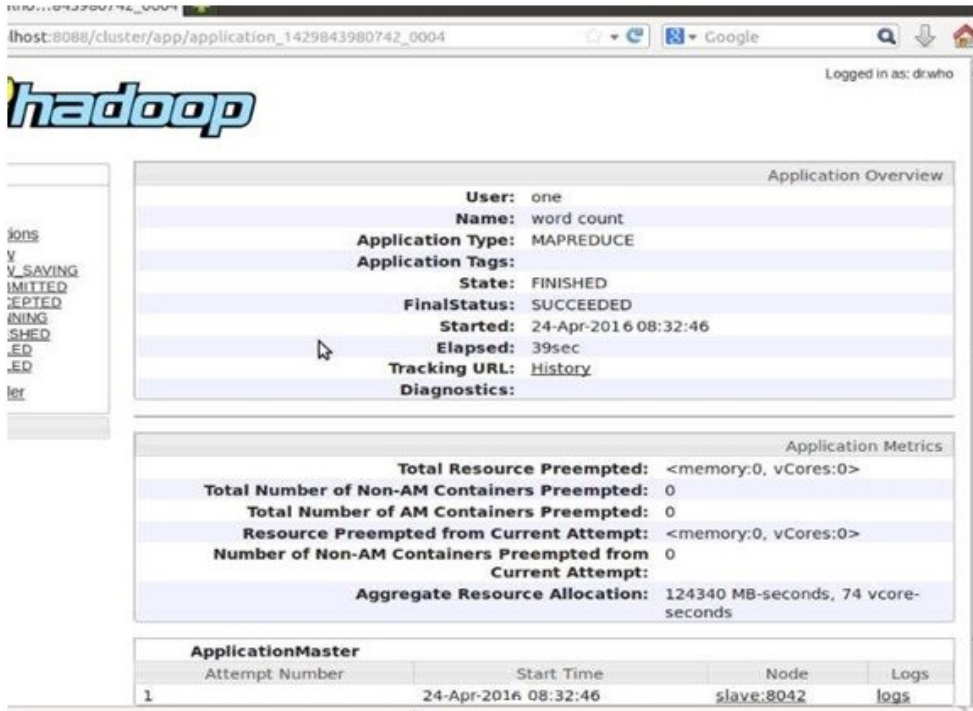


Fig 4.5: Hadoop Performance for DataSet 2-II

Dataset 3:

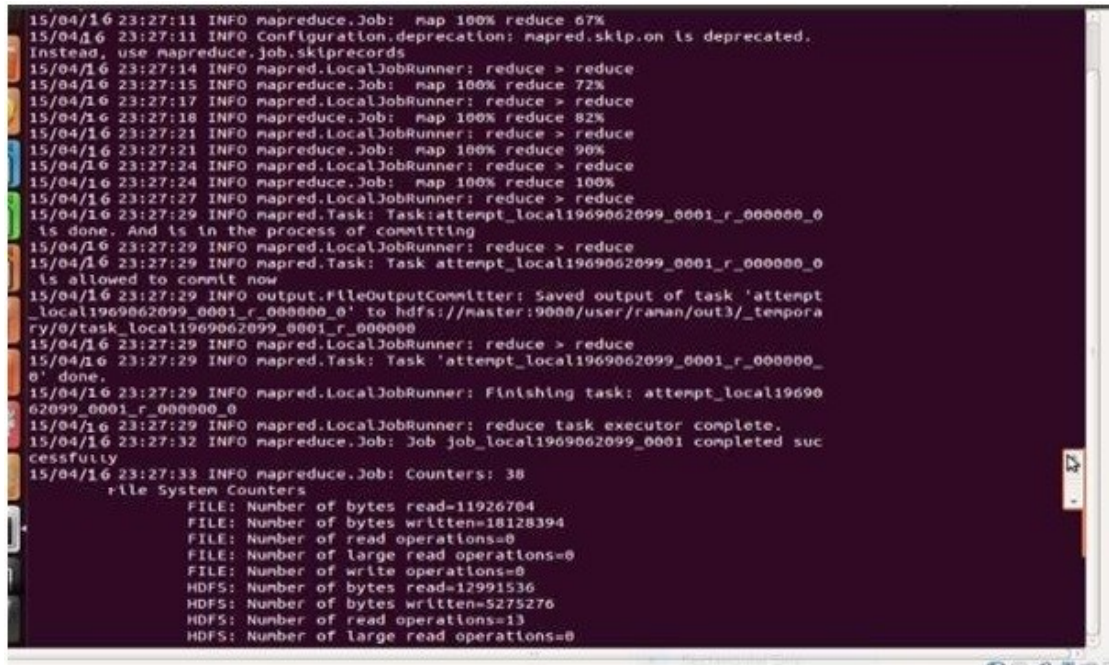


Fig 4.6: Hadoop Performance for DataSet 3



### 4.3.2 Performance with Physical Machine:

Performance of Hadoop on physical machine using same four different datasets used earlier have been evaluated and presented in Fig 4.7 to Fig 4.12

DataSet 1:

```
hduser@raman-Satellite-C600: /usr/local/hadoop/share/hadoop/mapreduce
15/04/24 21:49:53 INFO mapreduce.Job: Job Job_1429891607246_0005 completed successfully
15/04/24 21:49:53 INFO mapreduce.Job: Counters: 43
File System Counters
  FILE: Number of bytes read=503389
  FILE: Number of bytes written=1165137
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=3985382
  HDFS: Number of bytes written=391838
  HDFS: Number of read operations=8
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=36239
  Total time spent by all reduces in occupied slots (ms)=17875
Map-Reduce Framework
  Map input records=37853
  Map output records=572213
  Map output bytes=5822467
  Map output materialized bytes=503389
  Input split bytes=100
  Combine input records=572213
  Combine output records=28838
  Reduce input groups=28838
  Reduce shuffle bytes=503389
  Reduce input records=28838
  Reduce output records=28838
  Spilled Records=57860
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=1035
  CPU time spent (ms)=8160
  Physical memory (bytes) snapshot=253385888
  Virtual memory (bytes) snapshot=972206080
  Total committed heap usage (bytes)=218365952
Shuffle Errors
```

Fig 4.7: Performance for DataSet 1-I

The screenshot shows the Hadoop web interface in Mozilla Firefox. The browser address bar shows the URL: `http://localhost:8020/cluster/app/application_1429891607246_0004`. The page displays the Hadoop logo and a sidebar with navigation options like Cluster, About, Nodes, Applications, Scheduler, and Tools. The main content area shows the 'Application Overview' for a job with the following details:

User:	hduser
Name:	word count
Application Type:	MAPREDUCE
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	24-Apr-2016:45:53
Elapsed:	32sec
Tracking URL:	History
Diagnostics:	

Below this, the 'ApplicationMaster' section shows a table with the following data:

Attempt Number	Start Time	Node	Logs
1	24-Apr-2016:45:53	raman-Satellite-C600:8042	logs

Fig 4.8: Performance for DataSet 1-II

DataSet 2:

```

hduser@aman-Satellite-C600: /usr/local/hadoop/share/hadoop/mapreduce
HDFS: Number of large read operations=0
HDFS: Number of write operations=1
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=17195
  Total time spent by all reduces in occupied slots (ms)=22848
Map-Reduce Framework
  Map input records=133859
  Map output records=133855
  Map output bytes=7031184
  Map output materialized bytes=5692854
  Input split bytes=184
  Combine input records=133855
  Combine output records=184393
  Reduce input groups=184393
  Reduce shuffle bytes=5692854
  Reduce input records=184393
  Reduce output records=184393
  Spilled Records=280788
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=95
  CPU time spent (ms)=7780
  Physical memory (bytes) snapshot=252522496
  Virtual memory (bytes) snapshot=967278496
  Total committed heap usage (bytes)=235929888
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=6495788
File Output Format Counters
  Bytes Written=5275276
hduser@aman-Satellite-C600: /usr/local/hadoop/share/hadoop/mapreduce$

```

Fig 4.9: Performance for DataSet 2-I

The screenshot shows the Hadoop web interface in a Mozilla Firefox browser. The page title is 'Application Overview' and it is logged in as 'hduser'. The main content area displays details for a job named 'word count'.

<b>User:</b>	hduser
<b>Name:</b>	word count
<b>Application Type:</b>	MAPREDUCE
<b>State:</b>	FINISHED
<b>Final Status:</b>	SUCCEEDED
<b>Started:</b>	24-Apr-2016 11:51:46
<b>Elapsed:</b>	54sec
<b>Tracking URL:</b>	History
<b>Diagnostics:</b>	

Below the application details, there is a table for the 'ApplicationMaster' showing the job's progress across nodes.

Attempt Number	Start Time	Node	Logs
1	24-Apr-2016 21:51:46	aman-Satellite-C600-8052	logs

Fig 4.10: Performance for DataSet 2-II

DataSet 3:

```

hduser@raman-Satellite-C600: /usr/local/hadoop/share/hadoop/mapreduce
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=17195
  Total time spent by all reduces in occupied slots (ms)=22848
Map-Reduce Framework
  Map input records=133859
  Map output records=133855
  Map output bytes=7931184
  Map output materialized bytes=5692854
  Input split bytes=184
  Combine input records=133855
  Combine output records=184393
  Reduce input groups=184393
  Reduce shuffle bytes=5692854
  Reduce input records=184393
  Reduce output records=184393
  Spilled Records=288706
  Shuffled Maps =1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=95
  CPU time spent (ms)=7700
  Physical memory (bytes) snapshot=252522496
  Virtual memory (bytes) snapshot=967278400
  Total committed heap usage (bytes)=235929600
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=6495708
File Output Format Counters
  Bytes Written=5275276
hduser@raman-Satellite-C600: /usr/local/hadoop/share/hadoop/mapreduce$

```

Fig 4.11: Performance for DataSet 3-I

The screenshot shows the Hadoop web interface in Mozilla Firefox. The browser address bar shows the URL: `http://localhost:8080/cluster/app/application_1425851807246_0008`. The page title is "Application Overview".

On the left side, there is a navigation menu with the following items: Cluster, About nodes, Applications, NEW, NEW\_SAVING, SUBMITTED, ACCEPTED, RUNNING, REMOVING, FINISHED, FAILED, KILLED, Scheduler, and Tools.

The main content area displays the following application details:

- User:** hduser
- Name:** word count
- Application Type:** MAPREDUCE
- State:** FINISHED
- FinalStatus:** SUCCEEDED
- Started:** 24-Apr-2016 1:57:32
- Elapsed:** 58sec
- Tracking URL:** History
- Diagnostics:**

Below the application details, there is a table for "ApplicationMaster":

Attempt Number	Start Time	Node	Logs
1	24-Apr-2016 1:57:32	raman-Satellite-C600:8042	logs

At the bottom of the page, there is a link: "About Apache Hadoop".

Fig 4.12: Performance for DataSet 3-II

### 4.3.3 Performance Metrics of cluster:

Performance is measured by setting up the Physical and virtual cluster then running the map reduce program on different sizes of input and variation analyzed in results with varying cluster size and configuration of the system. The results and analysis is shown in the Table 1 for word count operation and in Table 2 for average operation. Corresponding graphs are in Fig 4.13 and Fig 4.14.

Table1: Time taken by physical and virtual cluster to perform Word Count

Dataset Name	File Size	Time taken in Physical cluster (Sec)	Time Taken by Virtual cluster(Sec)
Dataset 1	52.15MB	32	26
Dataset 2	53.80MB	54	39
Dataset 3	56.20 MB	58	42
Dataset 4	61.10 MB	61	47

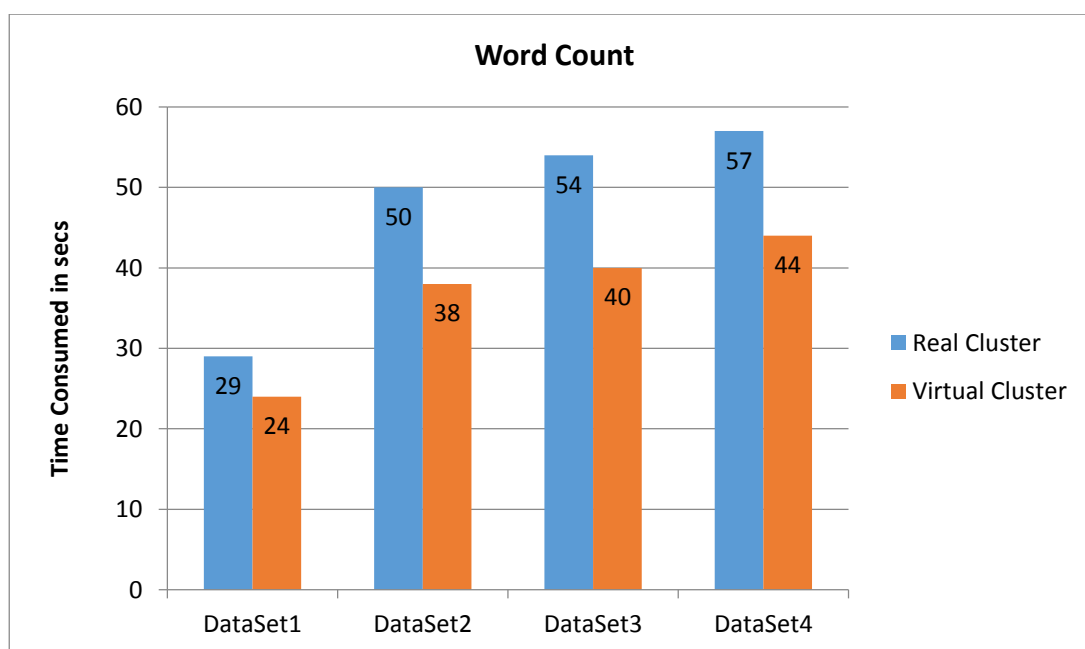


Fig 4.13: Time comparison between real and virtual clusters for word count operation

Table2: Time taken by physical and virtual cluster to perform Average

Dataset Name	File Size	Time taken in Physical cluster (Sec)	Time Taken by Virtual cluster(Sec)
Dataset 1	52.15MB	29	24
Dataset 2	53.80MB	50	38
Dataset 3	56.20 MB	54	40
Dataset 4	61.10 MB	57	44

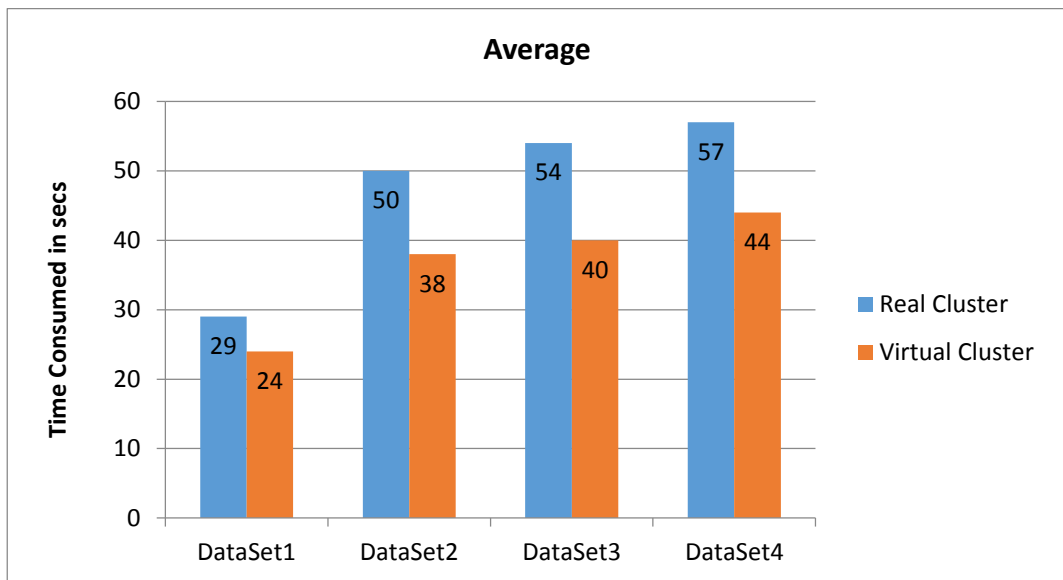


Fig 4.14: Time comparison between real and virtual clusters for average operation

## **CHAPTER 5**

### **CONCLUSION AND FUTURE SCOPE**

---

The presented work analysed and compared the performances of four different data sets on virtual environment and physical machine. Two different operations – word count and average on some numbers are implemented to support the conclusion that performance is better in virtual machines. Gradually increasing the size of data set does not affect performance. But increasing virtual machines may affect adversely on performance. To compensate the loss, high configuration machines are to be taken for installing virtual machines.

The next work in sequence could be the study on scalability. Issue needs to be considered, simply increasing the configuration will not lead to optimum solution as high cost will also be involved in that case.

## REFERENCES

---

- [1] Jeffrey Dean and Sanjay Ghemawat, “MapReduce:Simplified Data Processing On Large Clusters”,(2004)
- [2] Robert D. Schneider , “Hadoop For Dummies”(2012).
- [3] P Beulah Soundarabai, Aravind S, Thriveni J, K .R Venugopal and L. m Patnaik, “Big Data Analytics : An Approach Using Hadoop Distributed File System”
- [4] Daniel A. Menasce, “Virtualization Concepts, applications and performance modelling”
- [5] Ashish Nadkarni and Brett Waldman, “Desktop Virtualization and storage Solutions evolve to support Mobile workers and Consumer Devices”
- [6] T-Systems Enterprise Services GmbH, Germany White Paper on “Desktop Virtualization : The future of the corporate desktop”
- [7] David de Nadal Bou , “Support for Managing Hadoop Dynamically Hadoop Clusters” (2010)
- [8] Vblock Fast Path Desktop Virtualization Platform
- [9] Jeffrey Shafer, Scott Rixner, and Alan L. Cox, “The Hadoop Distributed Filesystem: Balancing Portability and Performance”.
- [10] Vidyasagar S.D “A Study on Role of Hadoop in Information Technology era”  
Vol 2, Issue : 2 ,Feb 2013, ISSN No 2277 – 8160
- [11] Prashant D. Londhe, Satish S. Kumbhar, Ramakant S. Sul, Amit J. Khadse “Processing Big Data using Hadoop Framework”,2012
- [12] Dali Ismail, Steven Harris, “Performance Comparison of Big Data Analysis using Hadoop in Physical and Virtual Servers”, on “Computational Intelligence and Informatics (CINTI)”, *fourteenthIEEE International Symposium, 2013,Pages: 327 - 332, DOI: 10.1109/CINTI.2013.6705215.*
- [13] Jeff Buell, “Virtualized Hadoop Performance with VMware Vsphere5.1”,2012
- [14] Aditya B. Patel ,Manashavi Birla, Ushma Nair, “Addressing Big Data Problem Using Hadoop And Map Reduce” ,*Nirma University International Conference On Engineering,NUiCONE-2012*
- [15] Hae-Duck J. Jeong, WooSeok Hyun,Jiyong Lim, Ilsun You,” “Anomaly Teletraffic Intrusion Detection Systems on Hadoop Based Platforms:A Survey of

Some Problems and Solutions”, *Fifteenth International Conference on Network-Based Information Systems*, 2012.

[16] Arantxa Duque Barrachina and Aisling O’Driscoll, “A Big Data Methodology For Categorising Technical Support Requests Using Hadoop and Mahout”, *journal of Big Data 2014*

[17] <http://aci.info/2014/07/12/the-data-explosion-in-2014-minute-by-minute-infographic/>

[18] [http://www.sas.com/en\\_us/insights/big-data/Hadoop.html](http://www.sas.com/en_us/insights/big-data/Hadoop.html)

[19] <https://www.quora.com/What-are-the-limitations-of-Hadoop>

[20] <https://Hadoop.apache.org>

[21] <http://www.slideshare.net/vinothkumarselvaraj1/hypervisor-and-nova>

[22] <http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/>

[23] <http://www.thegeekstuff.com/2012/02/virtualbox-install-create-vm/>

[24] HOG: Distributed Hadoop MapReduce on the Grid Chen He, Derek Weitzel, David Swanson, Ying Lu

[25] Y. Yang, X. Long, X. Dou, C. Wen, "Impacts of Virtualization Technologies on Hadoop", In *Third International Conference on Intelligent System Design and Engineering Applications*, 2013

[26] Y. Geng, S. Chen, Y. Wu, R. Wu, G. Yang, W. Zheng, "Location-aware MapReduce in Virtual Cloud", *International Conference on Parallel Processing*, 2011

[27] Daniel Schlosser, Michael Duelli, Sebastian Goll”, published in NETWORKING’11, *tenth international IFIP TC 6 conference on Networking*, *Volumn Part I*

[28] Aparna raj, Kamaldeep kaur, Uddipan Dutta, V Venkat Sandeep, Shrisha Rao- “Enhancement of Hadoop clusters with virtualization using the Capacity scheduler”- *Third Conference on Services in Emerging Markets*, 2012.

[29] Keim, Daniel, Huamin Qu and Kwan-Liu Ma. “Big Data Visualizaation” in *Computer Graphics and Applications*, *IEEE 33.4(2013): 20-21*.

[30] Dittrich Jens, Jorge-Arnulfo Quiane-Ruiz “Efficient big data processing in Hadoop MapReduce”, *proceedings of the VLDB Endowment 5.12(2012): 2014-2015*.



[31] Jun Fan, Xinhui Li, Chi Harold Liu, Jeffrey Bull, Gavin Lu, Luke Lu-  
“Diagnosing Virtualized Hadoop Performance from Benchmark Results: An  
Exploratory Study”, in *IEEE International Conference on Big Data*, 2014.