LOVELY
PROFESSIONAL
UNIVERSITY

IMPROVING RESOURCE UTILIZATION BY INTEGRATING HADOOP MAP REDUCE FRAMEWORK WITH VIRTUAL BOX

A Dissertation submitted by

Ramanpal Kaur
(Regd. No. 11307314)

to

Department of Computer Science and Engineering

In partial fulfilment of the requirement for the
Award of the Degree of

Master of Technology in Computer Science and Engineering

Under the guidance of

Mrs. Harjeet Kaur
(Assistant Professor)
(04/2015)

PT

LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

School of: Technology and Sciences

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: RAMANPAL BRAR	Registration No: 11307314
Batch: 2013-2015	Roll No.
Session:	Parent Section: K2305
Details of Supervisor:	Designation: Asst. Professor
Name: Kamaldeep Kaur	Qualification: MTech
U.ID: 18308	Research Experience: Hadoop, Map Reduce, 3/4 etc.

SPECIALIZATION AREA: Software Engineering (pick from list of provided specialization areas by DAA)

PROPOSED TOPICS

1. Integration of virtualization (Such as ~~Xen~~ VM ware) with Hadoop Tools.
2.
3.

Some provided some work is yet due.

Kamaldeep 18308
Signature of Supervisor

PAC Remarks:
Topic 1 is approved. Publication is suggested

19/9/14

APPROVAL OF PAC CHAIRPERSON: Signature: 11011 Date: 19/9/14

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

ABSTRACT

MapReduce is a framework for processing huge volumes of data in parallel, on big clusters of nodes. Processing enormous data requires fast coordination and allocation of resources. Emphasis is on achieving maximum performance with optimal resources. This paper portrait is a technique for accomplishing better resource utilization. The main objective of the work is to incorporate virtualization in Hadoop MapReduce framework and measuring the performance enhancement. In order to realize this master node is setup on physical machine and slave nodes are setup in a common physical machine as virtual machines (VM), by cloning of Hadoop configured VM images. To further enhance the performance Hadoop virtual cluster are configured to use capacity scheduler.

CERTIFICATE

This is to certify that **Ramanpal Kaur (11307314)** has completed M.Tech dissertation titled **“Improving Resource utilization by Integrating Hadoop Map Reduce Framework with Virtual Box”** under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of the dissertation has ever been submitted for any other degree or diploma.

The dissertation is fit for the submission and partial fulfilment of the conditions for the award of M.Tech Computer Science and Engineering.

Date: _____________ Signature of Advisor: ___________

(Mrs. Harjeet Kaur)

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude to the people who helped me directly or indirectly in my thesis work.

Especially, I would like to thank my supervisor Mrs. Harjeet Kaur for being great mentor and best advisor. Her encouraging words, advise and useful suggestions help me throughout the project work. It was pleasure for me to work under her guidance. She has always been a source of ideas and inspiration to me.

I would like to thank Mr. Balraj Singh for his support and guidance. Lastly, I would like to thank all the teaching and non-teaching members of the Department of Computer science, Lovely Professional University, for their support and help throughout my work.

Ramanpal Kaur

Reg. no. 11307314

DECLARATION

I hereby declare that the dissertation entitled **“Improving Resource utilization by Integrating Hadoop Map Reduce Framework with Virtual Box”** submitted for the M.Tech degree is entirely my original work and all references and ideas have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: 29/04/2015

Ramanpal Kaur

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER1

INTRODUCTION

1.1 Virtualization

Virtualization is creation of a real version of something rather than virtual, containing operating system, computer network resources or storage device. It starts in 1960's as a method of logically dividing the system resources. Virtualization is the most effective way to prove the software technology and rapidly changing the way to people compute. Virtualization gives a chance to execute various machines on a single server and distribution the properties of that single machine among the various environments. Dissimilar virtual machines can run on changed operating systems and several applications on the same physical computer. It is achieved with the virtual machine monitor (VMM). VMMs are also known as hypervisors. Hardware abstraction layer is added by system virtualization, called the Virtual Machine Monitor, on top of the scanty hardware as shown in the Fig1. This layer provides an interface that is functionally equivalent to the actual hardware to a number of virtual machines. These virtual machines may then run regular operating systems, which would normally run directly on top of the actual hardware. [1]

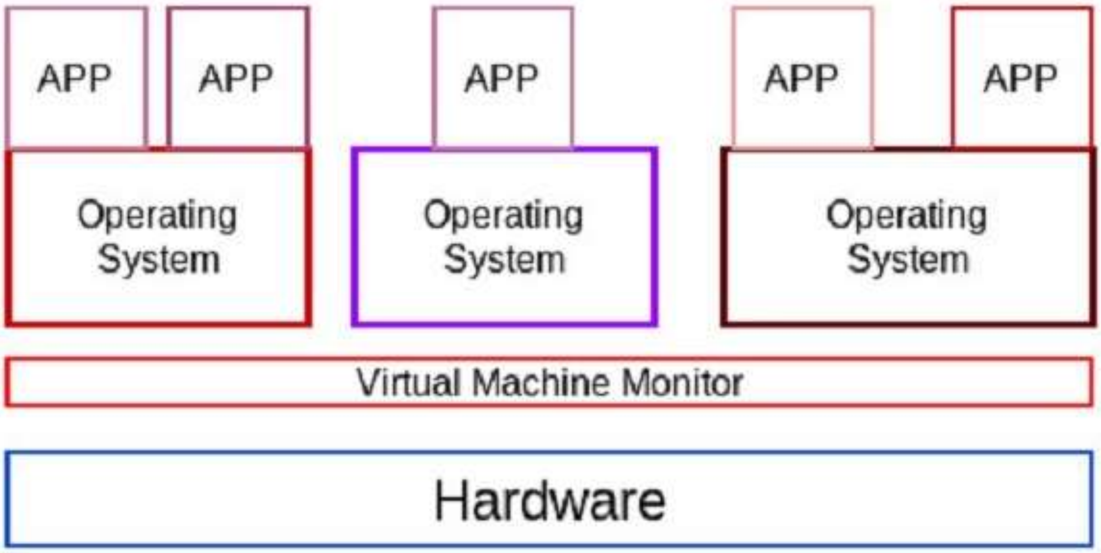


Fig 1: Virtualization basic concepts

VMMs are creditworthy for holding path of every activity executed with virtual machines. Hypervisor are divided into Para virtualization and full virtualization VMMs. Para virtualization VMMs run instantly without the demand for a host OS on hardware, whereas full virtualization VMMs operate on top of a host OS. Using full virtualization Server virtualization can be achieved.

- **Full virtualization-**Full virtualization is accomplished using direct implementation of binary transformation and application code. The VMMs snares and understanding all OS directives and caches the outcomes for succeeding use.
- **Para virtualization—**para virtualization provides the interface to virtualized components that are similar but not the same as the hardware. Efficiency of the para virtualization is more than the full virtualization. Para virtualization not translates the binary requests to the system calls. It is successful because it allows the relocating the direct tasks form the virtual to the host system. The para virtualization where the unmodified operating system, components are available gives the significant advantage in the performance. In Fig 2: it shows the basic knowledge how para and full Virtualization works with the system. Full Virtualization showing the direct execution of requests and para virtualization shows the first binary translation of requests.

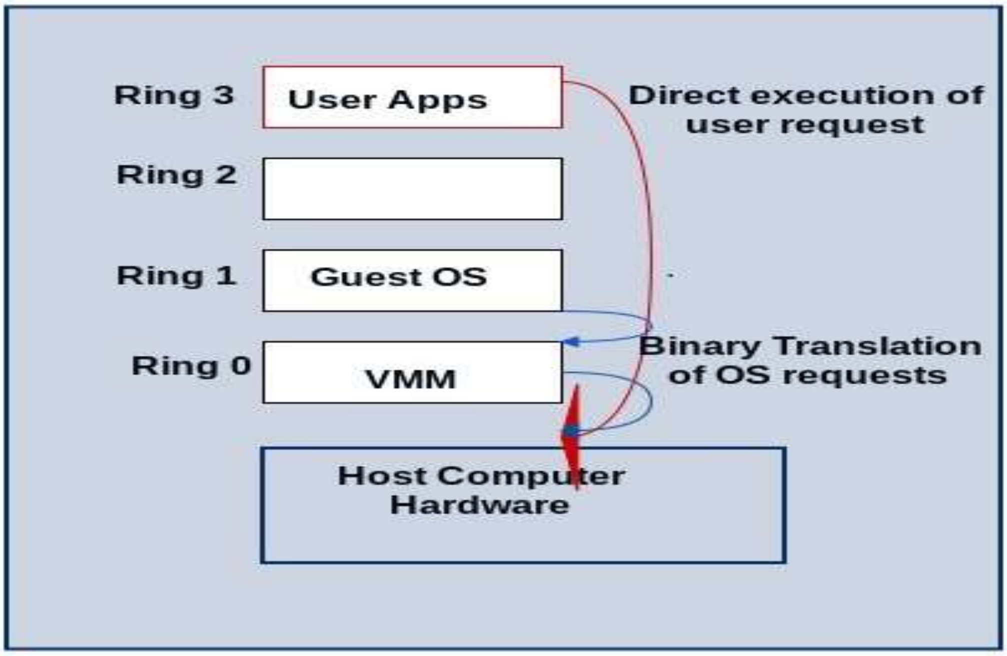


Fig 2: showing the full virtualization and paravirtualization

We are using the virtualization for different reasons that are mentioned below.

- Enhance Server Efficiency to increase the software efficiency as a method.
- Better Disaster Recovery Efforts to increase the dataset.
- Increase Business Continuity
- Test Security Updates and Patches
- Move to Desktop Virtualization
- Computer hardware was planned to run a single application and single operating system. This provides most of the machines immensely underutilized. Virtualization assists to use the resources expeditiously.
- Evolving business and reshaping the modern workplace.

1.2 Significance of Virtualization

Security: by subdivisioning surrounding with separate security essential in unlike virtual machines one can choose the guest operating system and instrument that are extra suitable for each environment. For example, if we want to execute the Apache web server on top of a Linux guest operating system and a backend MS SQL server on top of a guest Windows XP operating system, all in the similar physical plat- form. A surety flack on one virtual machine does not settle with others because of their segregation.

Availability and reliability: software crash does not impact other virtual machines. If one virtual machine is failed it can be recovered from the second virtualization machine.

Cost: It is potential to attain cost reductions by integration smaller servers into more right servers. Cost reductions base from hardware cost reducing operations cost diminution in terms of force, floor space, and software licenses. VMware cites overall cost reductions ranging from 29 to 64%.

Workload Adaptability: Changes in work- load volume levels can be gently handled by unfirms resources and precedence allotment among virtual machines. Autonomic computing-based resource allocation Proficiencies, actively move processors from one to another virtual machine.

To Balancing the load: The software state of an integral artificial machine is entirely enclosed by the VMM; it is comparatively peaceful to relocate virtual machines to other platforms in order to better production through better load balancing.

Legacy System: if any organization determine to migrate the dissimilar operating system, It is conceivable to proceed the test heritage applications as a guest OS within a VM on the old OS running.

1.3 Types of virtualizations:

1. Server Virtualization—server virtualization is the screening of server resources admits the number and the identity of separate physical servers, processors, and operating system, form server users. The server administrator utilizes a software application to divide one physical server into multiple scattered virtual environments. For e.g. we know that each physical server cannot perform up to the level that they can perform. It performs only up to 20% to 30%. We can create the set of partitions from physical server resources such as hard disk, processor and memory. Then assign these virtual resources to different servers.

2. Network Virtualization- The complete procreation of a physical network in software is called Network virtualization. It offers guarantees of a physical network and the similar characteristic, yet they present the hardware and operational benefits. It fragmented useable bandwidth in a network into autonomous channels that can be appointed to particular servers or devices. For example Citrix and Vyatta used a network protocol stack combining Vyatka's routing, firewall and VPN functions with load balancer.

3. **Application Virtualization—**Application virtualization separates application from the hardware and the operating system, putting them in a container that can be resettled without disturbing the other systems. Microsoft offers an application virtualization such as Microsoft office is a suite of applications that are used in the business and in homes and windows file manager is graphical user interfaces through which users could see and manipulate files and folders.

4. **Storage Virtualization-** this virtualization is portion of the software-defined layer that stores and offers progresses in routine and space proficiency without requiring the purchase of additional storage hardware. For example hard disk management, replication and RAID.

5. **Desktop Virtualization**-Strategically desktops gives you the chance to react faster to changing demand and opportunities. You can simplify cost or gain service rapidly or well bringing virtualized desktops and coatings to offices, outsourced, offshore apply mobile actors on iPad. Virtual Machine desktop resolutions are accessible, ordered, completely protected or extremely usable to ensure maximal uptime and productiveness. Efficiently deployment Provide reliable remote approach to teleworkers and irregular workers without giving functioning. The capacity of the software on the one hand: price, protection and employee motive on other hand and the operative environment is an important commercial constituent for businesses for several reasons. The leaner and greater care-free it is. Desktop virtualization makes a terminated interval between terminals and applications. The central position of applications and information in the data center to be retrieved across the network, grant the user to access his or her PC from anyplace. This extends many companies the saint result for their workstation PCs, as shown in the Fig 3 that explains the concept of desktop virtualization.

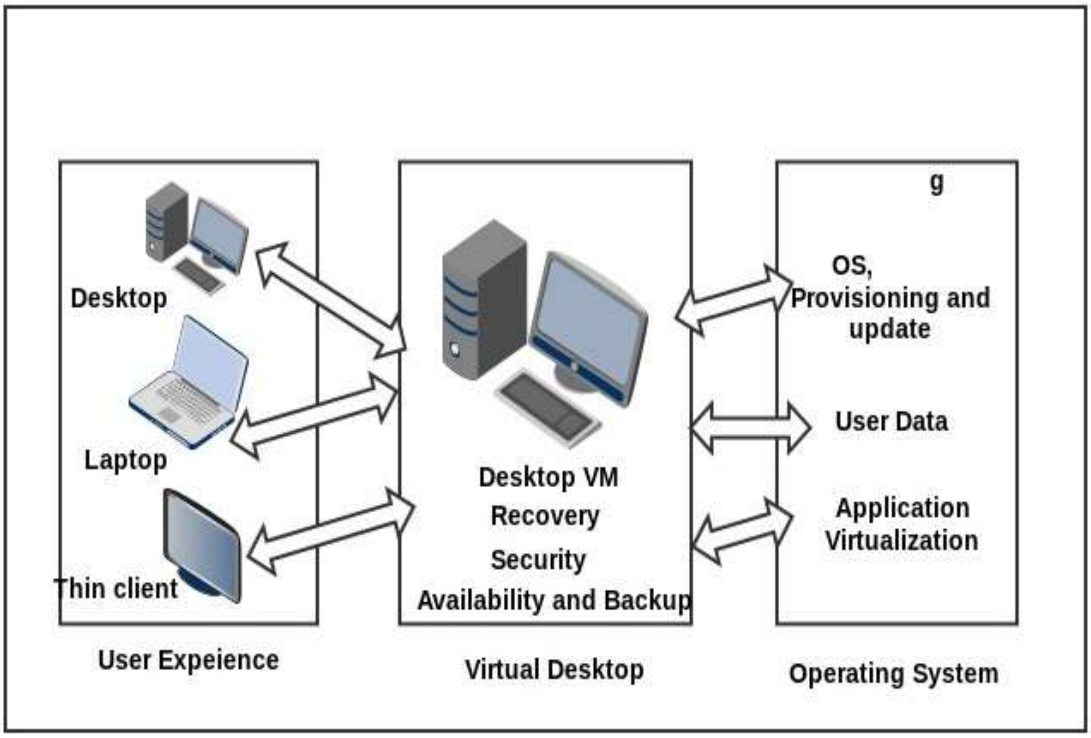


Fig 3: Desktop virtualization in organization

It can benefit workforce productivity and desktop operations:

- To enhance the workforce and flexibility the users invest the telecommuting plan and desktop images for their use on their devices.
- The End user Leverage on devices that tracts the applications. It offers more productivity by devising the apps available on smartphones and other systems.
- IT can easily adapt the change and enable fresh enlargement, offshoring that gives the initiatives with higher security and lower costs.
- By reconsidering the desktop virtualization it gives the way to reduce the cost and more security with the varying level of mobility.it gives the service driven approach rather than asset.

To build the desktop virtualization we need different set of requirements. To realize what base we required, organizations want gather the number or another types of customers that subscribed or the divergent types of VCC technologies that they be installed. Choice of suitable desktop virtualization platform is significant decision that should point for buyers. The features consist of mobility, BYOD scheme, safety and policy-based approach, and deployment experiments. IT configures the proper servers, networking and storage. In addition, IDC has logically create the storage resolution has been the important part of the physical infrastructure because it provides the backbone for many desktop images to boot off and access information. **The desktop storage solution is required to improve the performance at a reduced cost that must provide continuous access, be safe, and offer combined data protection.**

It is worth notable that not all desktop virtualization solvent are equal in terms of performance. Hence, storage form changes usually for different desktops. The performance of one virtual desktop saving model may not match with another. It emphasizes the requirement for close corporations among storage, compute, hypervisor and networking and seller in order to define what should be the configurations and what should be the requirements.

1.4 Virtualization platforms

Review of different virtualization platforms found now days is presented below.

KVM Platform: It is free available platform for virtualization that works on the kernel based. This platform used for the full virtualization. It supports the Linux operating system that uses the Virtual machine monitors. KVM authorized for Linux and windows in which Linux as a guest operating system.

Features of KVM:

- Improved scalability that offers virtual machines can connect to more storage devices.
- Standard command set that adds the new features
- KVM improves the para-virtualized storage block interface.

Virtual Box Platform: a freshman Sun Microsystems to the ranks of virtualization market that provides the Virtual Box as a virtualization platform that allows para virtualization. In Initial the German company developed it. But now Sun Microsystems controls this virtual platform. Virtual box supports different host OS such window, Linux, Mac and Solaris.

Features of Virtual Box:

- Portability is one feature of virtual box.
- In this hardware virtualization required is not mandatory.
- Hardware support is more in Virtual box.

Microsoft Hypervisor platform: For maximum resource and hardware utilization it is necessary to virtualize your servers or single system. It helps in the cost reduction for this we need to virtualize on a single physical host. Hypervisor helps to run different operating system on a single system and runs the application in a parallel.

Few Features Listed below:

- Scalability that helps to add the more no of system with memory management.
- Management is easy.

Xen virtual platform: 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.

Few features listed down below:

- XenApp published apps can be deployed on window operating system.
- XenApp published desktops are low cost that provide the flexibility and mobility.

VMware platform: VMware is licensed virtualization platform that best for full virtualization. VM Ware provides different flavor's that are VM Ware Workstation, VM Ware ESX and server that supply many levels for edibility and functionality. VM Ware was extremely portable as it was autonomous as the fundamentals of physical hardware. In VMware it is possible that creating the single instance of the guest system and then copy it to on different systems.

Few listed Features of VMware:

- It is used for the efficient resource virtualization.
- Advanced memory management.
- Resource management for virtual machines that define the advanced resource allocation polices for virtual machines on single server.
- Interoperability means VMware allows the process to be tested and certified process for server, Desktop and application.

Red hat Enterprise virtual platform: It provides the board virtualization for management of the desktops and servers within the single infrastructure. Red hat enterprise uses the Linux that helps to make the continuity for workloads in the business with reliable scale.

- Application is reliable, scalable and efficient.
- Resource management is easy.
- Application development is stable and production platform is easy.

1.5 Hadoop

Hadoop is open source project written in java by Apache software foundation. Apache software foundation is a library that contains the framework that allows the processing of large unstructured data sets across clusters of computers. Library itself is able to handle the failures at the lower level, so high availability of services on the top of clusters.

Hadoop is progressing environment of components to run the Google Map reduce algorithms on commodity hardware with scalability. Hadoop gives very good results on physical machines and users get respond quickly but if incoming requests are more at that time it is difficult to handle.

1.6 Challenges of Hadoop

With different environments, it is considerable the deployment of the software and server with large machines. These were best managed with the configuration of the hardware and hadoop installation on the operating system. The companies need to ensure that the installation and hardware configuration were held properly on the physical systems when any situation or unexpected event occurs within the Hadoop environment. Hadoop must respond this quickly and solve this. As the Hadoop is growing field so there is no guidance or as such best platform to run it correctly and properly.

Hadoop Uses

Hadoop was developed for implementation of Google Map Reduce and file system. Different purposes and uses for which it used:

- **Compute:** The main use of Hadoop is analyzing and processing large amount of data. Compute is needed for processing the large data that is stored in CPUs and large memory as data warehouse.

  Then Hadoop distribute the workloads on different virtual machines by API. It processes the large calculations in a small time.

- **Storage system:** HDFS: Hadoop distributed file system is component of the Hadoop that runs the existing file system for each node of cluster. It allows user to have a single addressable name space, creating large file system .it mange the repetition and allows the parallel read write of the data. It helps to recover the failure with the replication of data.

- **Database:** Hadoop contains components that allow the data within the HDFS .This allows the standard tools for SELECT, INSERT, UPDATE data in Hadoop environment, with minimum changes in the code.

- **Large Data Sets** – Map Reduce combines with HDFS for processing the large amount of unstructured data in a reasonable time.
- **Scalability of algorithm**– In any physical system if hadoop is able to processing with the as cores are available in the system it is more scalable than any other system in organization helps to virtualize the single organization.
- **Logs** – Hadoop contains two components in which one is HDFS that maintain the logs of the system and allow the parallelism for read and write. Map reduces helps to reduce the size of the logs.
- **Extract-Transform-Load (ETL) Platform** – Hadoop provides the central location for the data like data warehouse so companies mostly using the Hadoop platform for synchronization and make the data update with others.

With all applications, some methods are not suitable for Hadoop for these hadoop less efficient:

- **Small File Archive** –Hadoop is used for the large amount of data, it is difficult to keep with the single namespace because it slowdowns the process when small requests coming again and again and processing these requests takes a time and packets that being transferred over the network.
- **High Availability** – A single Name Node becomes a single point of failure in Hadoop. While a second, passive Name Node can be configured.
- Not suitable for online transactions.
- Not good for low Latency.
- It requires intensive calculations with little data

1.7 Hadoop related Open source Projects

Hadoop started with Map Reduce implementation and but grew rapidly and now includes other projects to provide the required services. Following are the subprojects:

- **Common:** It is hadoop related project that helps in provisioning the other subprojects such as Configuration and RPC.
- **Avro:** Avro is used for the data serialization. It also provides dynamic integration with scripting languages.
- **Chukwa:** It is a data collection system that manages large distributed systems.
- **Hbase:** It is a scalable, distributed database that supports structured data storage for large tables.
- **HDFS:** it is hadoop distributed file system that provides parallel read and write to application data.
- **Hive:** It is a data warehouse infrastructure that provides ad hoc querying and data summarization as well.

1.8 Hadoop Architecture:

Hadoop architecture has two main Components:

1. HDFS (Hadoop Distributed file system)
2. Map Reduce Engine (Framework for processing large amount of data)

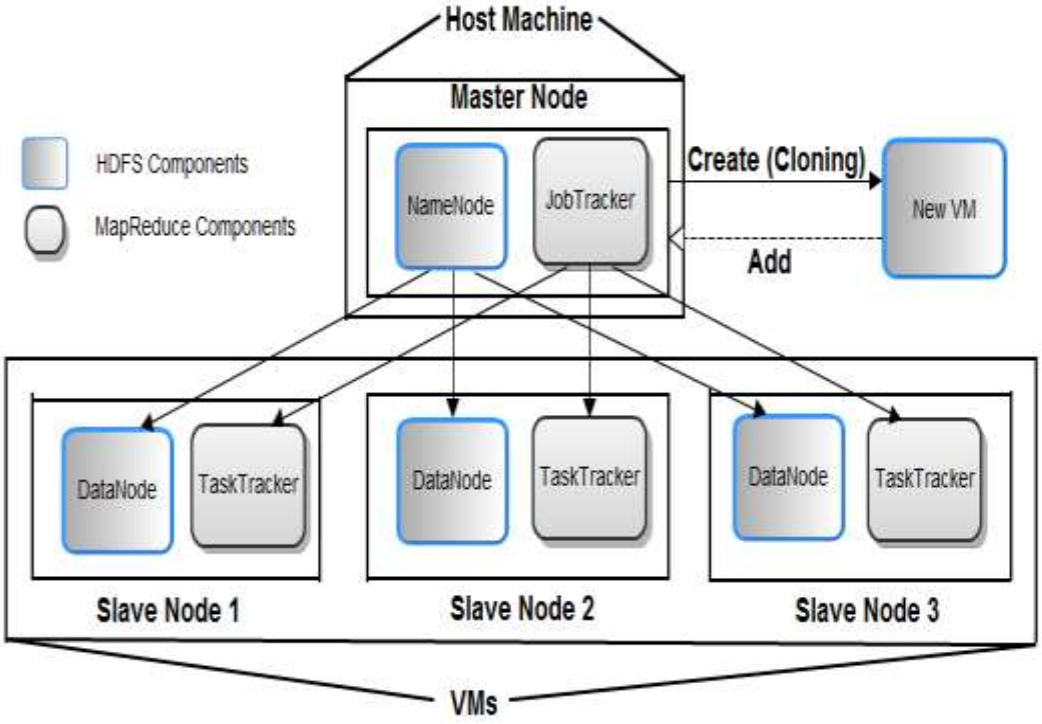


Fig 4: virtual cluster on a physical host machine.

Fig 4 shows master node of the cluster is the host machine or the physical machine. The job tracker, name node and hadoop daemons, all three are run on the master node. Other slave nodes form a cluster on VMs. There are multiple VMs that are set up as slave nodes on which the Hadoop daemons, Data node and Task tracker [1] are run. The addition of a new slave node can be achieved easily by cloning of an already configured VM.

HDFS (Hadoop Distributed file system)

HDFS runs on top of existing file system on each node of cluster to handle very large files with streaming data access Patterns. It uses blocks to store file or parts of a file. It can store large amounts of data, like terabytes and petabytes, and uses HDFS as its storage system. The data file can be accessed and stored as a single file system by the user. HDFS provides fault tolerance and high throughput to large datasets.

HDFS Goals:

- **HDFS heartbeats-** In order to check the connectivity between name and data node. Each data node sends periodic heartbeat messages to its name node, so that the name node can detect the loss the connectivity if it stops receiving messages from the data node. The name node that does not responds to the heartbeat they are marked as dead data nodes and refrains from sending further requests to them. Data is no longer available to an HDFS user from that node which is stored on a dead node, and is removed from the method effectively. If replication factor is caused by the death of the node to drop data blocks under their minimal rate, it initiates further repetition to bring the repetition factor rear to a standardized one.

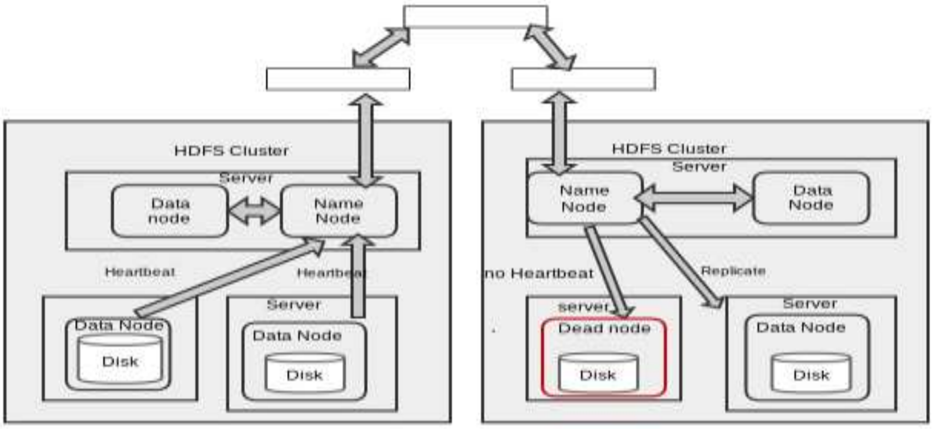


Fig 5: heartbeat between Name and data node

- **Reliability in Data storage:** An essential characteristic of HDFS is that stored is reliable. When any node fail then detection is first step then HDFS makes use of heartbeat message to sense the name and data nodes connectivity.
- **Data block re balancing:** HDFS data blocks may always not place uniformly across the nodes. It means that the space is underutilized so HDFS use block re balancing the blocks. One model move data into other data block to balance the blocks or use the free space .another model dynamically may create the replicas and use to balance the blocks.

- **Data integrity:** It used to safeguard the integrity of data. Checksum method is used on the data that is stored in HDFS file system and computes the checksum in hidden files that are under the same name space. When client receives the data it matches the data in checksum to ensure the integrity. HDFS file system stores the transaction in the file called FsImage. Once name node is ready then, it accesses the file next to with file it also read the FsImage, and applies the transactions and information to the system.

- **Synchronous Meta data updating**: A name node uses a log file known as the Edit Log to record each operation that happens to HDFS file system. If the Edit Log or FsImage files turn out to be tainted, the HDFS case to which they belong come to an end to function. Therefore, a name node provisions multiple imitations of the FsImage and Edit Log files. With manifold copies of these files in place, any alternation to either file propagates synchronously to all of the duplicates.

1.9 Map Reduce

Map Phase-- Map is data gathering stage as it divides the data into smaller parts and then process that pieces of data parallel. Map accepts input as the key/value pair and produces intermediate key/value pair.

Reduce Phase-- Reduce is considered as data transformation or data processing stage as it combines the results of map functions into one final result. Reduce accepts intermediate key/value pair and produces the final output.

Map Reduce is a programming Model that is used to process large amount of data. It is made by Google. If we want to finish the job in a reasonable time the Google decide to concrete the problem and define Map and lessen primitives. The Map piercing the task into minor bits of job and then sends it to the calculating machines and lessen is bringing together all the result form the Map step. Map reduce is not a new but simply used to handle the similar submissions with modest method and evidenced to be more stronger to process huge amount of data within the interval and high scalability.

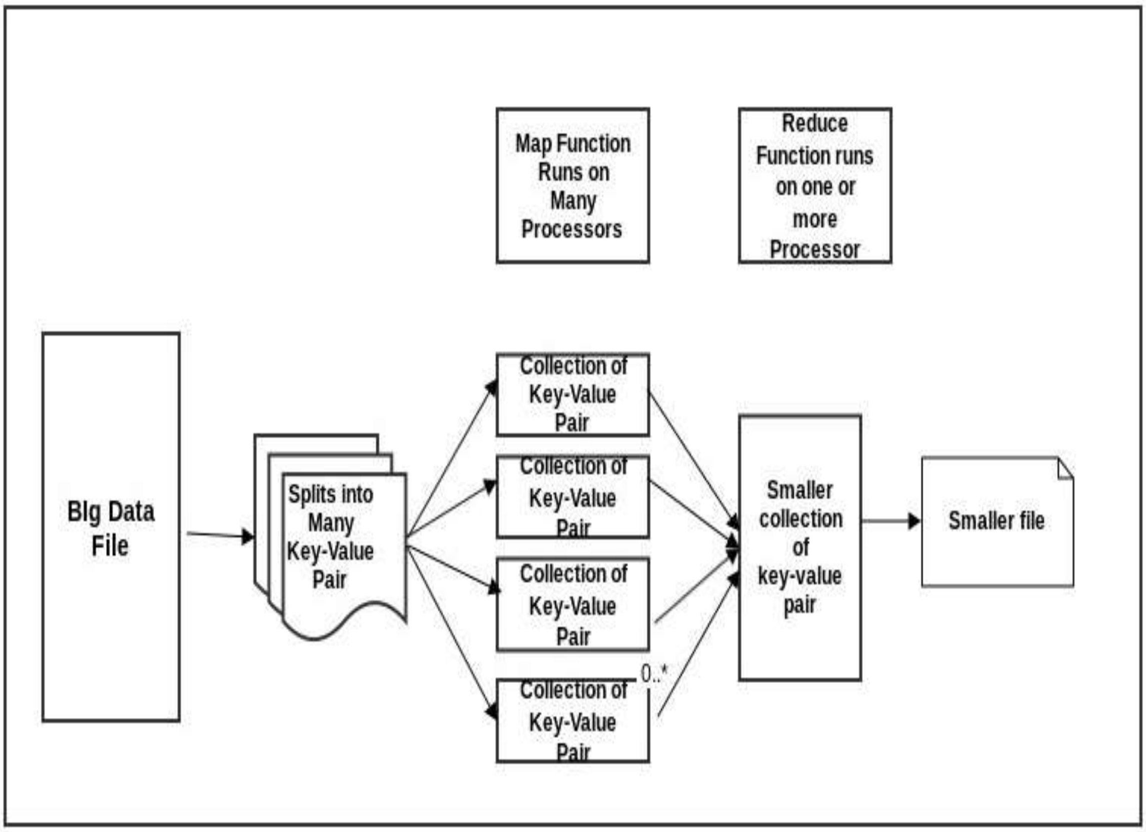

Map Function Runs on Many Processors
Reduce Function runs on one or more Processor
Big Data File
Splits into Many Key-Value Pair
Collection of Key-Value Pair
Collection of Key-Value Pair
Collection of Key-Value Pair
Collection of Key-Value Pair
0..*
Smaller collection of key-value pair
Smaller file

Fig 6: Working of MapReduce Framework

In Fig 6: shows that when any large or big amount file is arrived then map first split it into the small pieces or in different small modules after splitting the task then Map Function performed different operations on that data like sorting, shuffling etc. After the Map phase split output becomes the input for the reduce phase .then reduce phase perform the operations on the data and reduce it into smaller file.

When the map function starts producing output, takes advantage of buffering writes in memory and doing some pre-sorting for efficiency reasons.

The client submit the Map reduce job. Job tracker is one who, which coordinate the job run.

Steps for Map Reduce:

1. New job ID is to be asked by the job tracker

2. Output specifications of the job are checked.

3. Input splits for the job to be computed.

4. Resources needed to run the job are computed.

5. Jobtracker is told that the job is ready for carrying out.

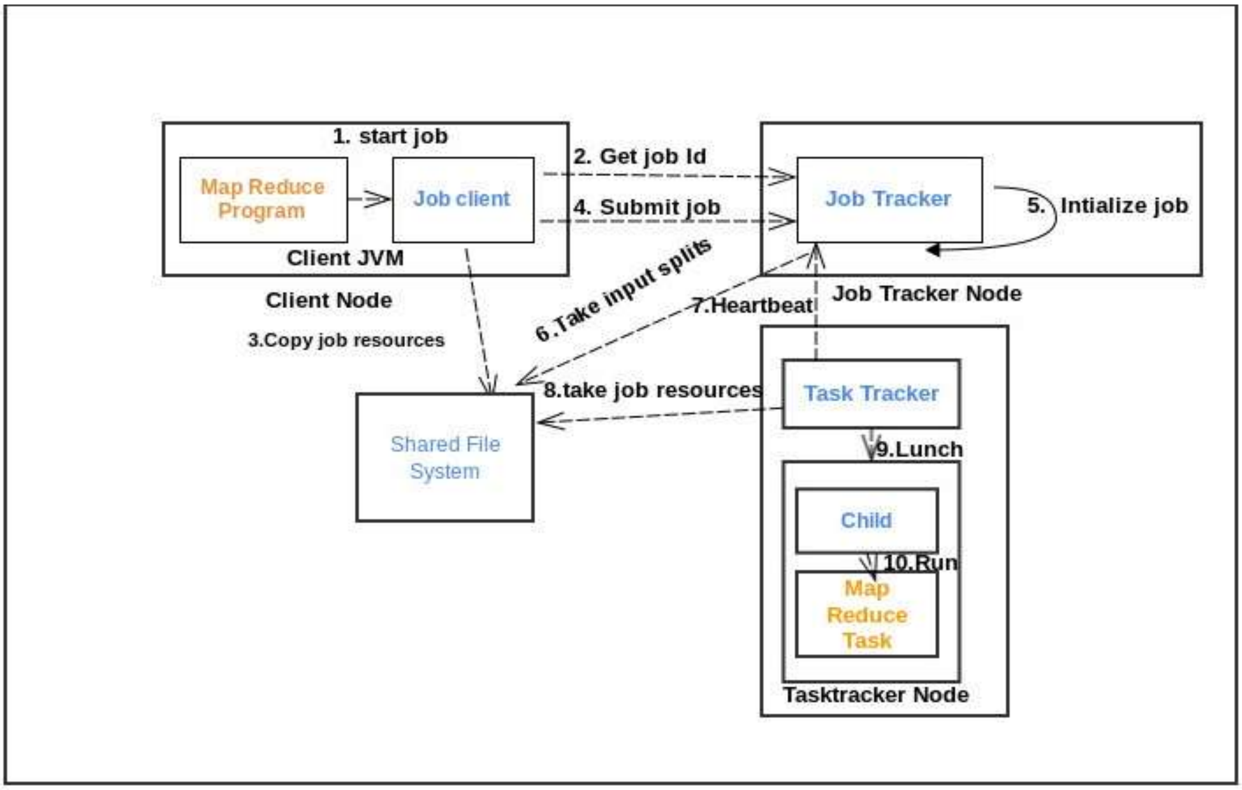


Fig 7: How Hadoop runs a MapReduce job

1.10Types of Nodes:

1. HDFS (Hadoop distributed file system) Node
2. Map Reduce Node

Types of Hadoop nodes:

Each Hadoop cluster has a variety of node types within Hadoop; which include Data node, Name node.

- **Name Node:** Name node is the center position for material about the file. An environment can have one or two Name nodes. It stores the metadata. It manages the File system Namespace.
- **Data Node:** Data Nodes make up the majority of the servers contained in a Hadoop environment. An environment can have more than one data nodes per Hadoop cluster. It manages the data and server to client.

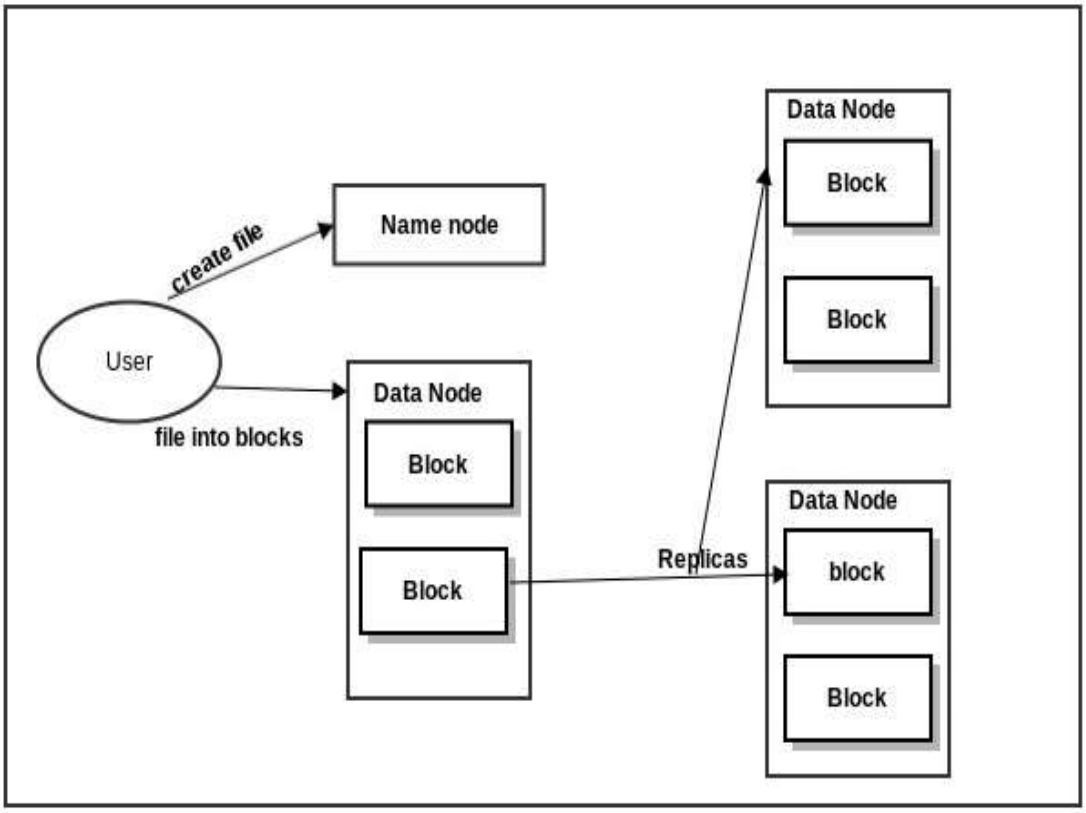


Fig 8: Hadoop Node Types

Map Reduce Nodes:

1. Job Tracker
2. Task Tracker

- **Job Tracker**: Job tracker is considered as a master who is responsible for the creation and execution of the job. Job tracker work on name node. Job tracker discovers the nearest task tracker with free slots and allocate job to it. It schedules and manages the Map Reduce.

- **Task Tracker**: Task tracker is a type of node which takes various map, reduce operations. Each task tracker have different no of slots that shows the no of tasks it can handle. An environment may have many task trackers per cluster. It uses the JVM machine and reschedules it.

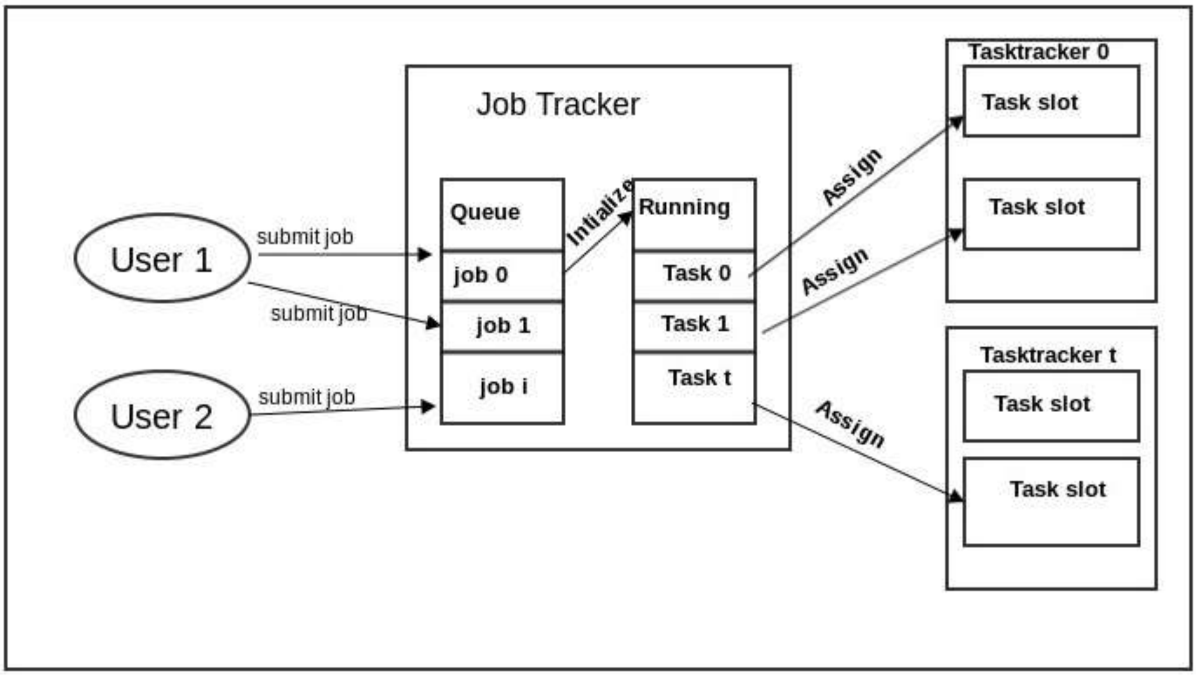


Fig 9: MapReduce Types

In Fig 9 It shows the working of the Jobtracker and task tracker. Users submit the task or job to the Job tracker then job tracker further split down or assign the jobs to the task tracker. In system there is only one job tracker and many task trackers per cluster.

CHAPTER2

LITERATURE REVIEW

With the growth of services and technologies, the data produced may be structured and unstructured and semi structured. To handle the data Hadoop is used.

Big Data University provides online course for Hadoop fundamentals. In which it gives the introduction about Hadoop. Hadoop is utilized by big data. After Hadoop introduction Hadoop architecture that explains the two major parts of Hadoop architecture. One is HDFS and another one is Map Reduce. The distributed file system, the main example of which is the Hadoop Distributed File System, though other file systems are supported. It works for the big files like in terabytes or petabytes as the size of the file increase it seeks the less time to process the data in HDFS system. It uses the streaming of the data that is sequential in nature rather than random. To access the data in a sequential manner means lesser number of times it generally start from the beginning of the block and then further. Hadoop Map reduce is inspired by a paper Google published on the Map Reduce technology. After that it explains how Map Reduce works. In this it first explain the Map and Reduce operations. Then Map Reduce is submitted to the Hadoop. Shuffle connects the output of each Mapper to the input of the reducer. Map Reduce job contains different steps. It explains about the job tracker and Task tracker. It also explains about the different type of nodes that are in the Hadoop.

It gives the information how Map reduce works and when any client put the request for the job then jobtracker start the job and the task tracker split the job into the different tasks.InTom white, “The Definitive Guide” Map split the task and used to run the parallel applications and Reduce is collecting the result of the Map step. At the highest level there are four independent entities: The client submits the Map Reduce job. The job tracker coordinates the job. The job tracker is a Java application whose main class is Jobtracker. The task tracker runs the tasks that the job has been split into. Task trackers are Java applications whose main class is Task Tracker. The distributed file system, which is used for sharing job files between the other entities. In real world it is possible that the process may crash, error in code due to this machine may fail the operation .Hadoop is its ability to handle such type of failures and allow completing the job. Task tracker Failure: failure of task tracker is one type of failure if the task tracker is failed by striking the applications then it does not send heartbeats to job tracker. When Job Tracker notice that task tracker stop sending the heartbeat it simply remove it from the pool. Job tracker Failure: the failure of job tracker is most serious type of failure. Hadoop has no mode to cope up with task tracker failure. Being a single point of failure, job fails here. Hadoop distributed file system: Hadoop comes with distributed file system that is called Hadoop distributed file system. Hadoop actually has a general purpose file system abstraction.

Today companies need to process the terabyte or petabyte dataset efficiently in lesser time. Data may be in unstructured form not following a strict schema so to process this data becomes so expensive. Vidyasagar S. D ," A Study on "Role of Hadoop in Information Technology era" explains that to process this large amount of data common infrastructure is followed given by Hadoop. Hadoop is available for processing large amount of data on commodity hardware. Hadoop is open framework for processing and analyzing large amount of data. In Hadoop large components are available if one component is failed it does not affect the working process of Hadoop .Hadoop implements the map reduce framework in which application is divided into small no of tasks and proceed these tasks parallel and then result is produced by the reducer. Hadoop follows the client server architecture in which HDFS consists single Name node and several no of data nodes.it automatically handles the node failure by data replication .HDFS provides high through put through large dataset .it is efficient and cost saving.

It was invented years ago for allowing the shared resources and expensive mainframes among the different applications. Now virtualization at all levels (desktop, system, network, and server) play significant role for providing the security, flexibility, reliability and reduce costs. With virtualization one can run different environment on same machine. VMM (Virtual Machine Monitor) Virtualizes all the resources and allocates them to the various Virtual machines that run on the top of VMM. VirDaniel A. Menasc ́Dept. of Computer Science George Mason Universitytualizatio,"VIRTUALIZATION: CONCEPTS, APPLICATIONS, AND PERFORMANCE MODELING" states that VMM runs in two Modes one is supervisor Mode and other is user mode. In supervisor Mode VMM control the access to the resource shared by all the Virtual machines and in this mode all the privileged instructions are run which is employed to change the location of the machine's shared resources. Examples of instructions: set program counter. By setting timer, halt the machine.

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,” Desktop virtualization is fastly 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 a access to users.

In this technical paper 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.

- VMware provides the Enhanced availability due to the Fault Tolerance.
- It gives the easier datacenter and faster management of data.
- In hadoop different applications are run that are used for the better resource utilization.

With the virtualization of hadoop on the VMware the results are improved than the native ones. To done the virtualization is not a big issue but considering the size and configuration of the machine to be considered.

This paper gives the environments of virtualized hadoop that automatically configure the resources that are shared for hadoop.it explains that first GRID is used in daily life for personal computer and process the data. After the GRID the cloud computing is used that is less popular but is more realistic than the GRID. Cloud computing gives the chance to see all in one cloud. The virtualization concept is used with it but not worth then the Google start making its own framework MapReduce that helps to process the large amount of data. MapReduce implementation for Google is private and open source that seems difficult so apache hadoop software is best to implement the map reduce. In starting hadoop start with the map reduce but it grows rapidly now it is also implemented with the HDFS (Hadoop distributed file system). Hadoop is used for java and it also supports its other subprojects like as pig, Hive, Hbase.

Large 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, provided by VCE: the Virtual Computing Environment Company. Built on VCE’s Vblock Infrastructure Platform— which 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.

The performance of three Hadoop applications noted for the different VMware and Sphere and compared to the native configuration. The average result shows that the performance difference between the native and simplified virtual configuration is 4%, the performance can be increased by adding multiple nodes per Hadoop. It is a distributed software platform for large amounts of data that manages and transforms for different virtual clusters in a cost effective way.

Jeff Buell iStaff Engineer gives the performance of VMware vSphere5 by analysing the different issues like as the availability and distributed program design. In this users used the algorithms that are highly available and help to improve the performance. As Hadoop can scale up to thousands or hundreds nodes to execute large cluster.in order to get sufficient performance including we are assuming that each disk server, network link, and even rack within the cluster is to be unreliable which allows to use the least luxurious cluster components consistent with delivering sufficient performance, including the use of unprotected local storage.

CHAPTER3

PRESENT WORK

3.1 Significance

Hadoop Map Reduce is highly scalable and it processes the big amount of data in a very less time because MapReduce component divides the large file or data into the lesser no of size that can be processed parallel. Hadoop Map Reduce works very well but when there is need of rapid provisioning of resources then it is a problem. **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.**

In case the computations are really very large then it need lot of resources to be allocated and need quick co-ordination as soon as possible. The follow up condition is applicable specially in two prospects; either at a specific instant huge no. of request need to be processed or need more resources to do so in order to add and configure a physical machine in hadoop environment. To achieve these faster computations and lowering the complexity we need to add more task tracker to our cluster

3.2 Objectives

1. To configure Hadoop in appropriate manner for multinode setup.
2. To study different virtual machine Platforms.
3. To comparison of different tools for identification and suitable for Hadoop.
4. To build the virtualized environment for Hadoop with identified Platform.
5. To analyses the performance of virtual cluster in proposed setup.

3.3 Research Methodology

In order to integrate the Virtualization with Hadoop tools the given methodology is used:

Identification

In this part the requirement analysis of virtualization is done. It involves configuration of different physical and virtual cluster with the different slave nodes. Identification of cluster formation is done to examine the performance.

Conceptualization

In this Study, different machines are required in formation of cluster is determined and Cluster setup is used to understand, measure the performance of map reduce program. The virtual box is utilized here in integration of machines in cluster. The configuration of Hadoop in single node as well as multi node setup is done.

Formalization

It involves the running of virtual cluster in which master node is running along with slave nodes. It includes working of map reduce in measurement of program efficiency in terms of time, CPU utilization.

Implementation

It implies that survey of specify knowledge into framework of the Ubuntu tool(terminal) to progress of working prototype. The specify of Virtual box to make a virtual cluster then running a map reduce word count program on virtual in respect with time and block size then comparing with physical cluster.

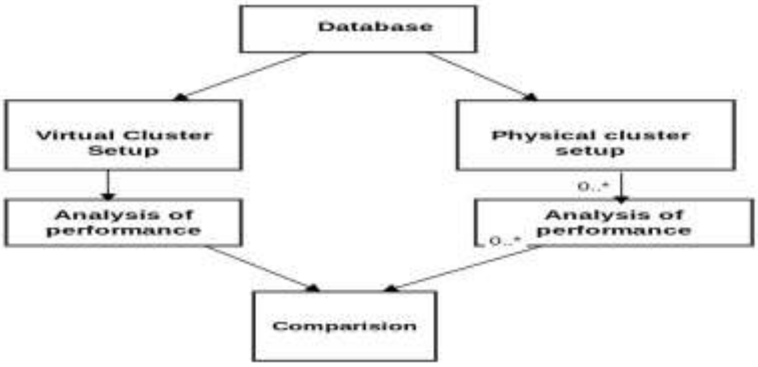


Fig10: Research Methodology

Implementation-

In this module, the program of word count is run on MapReduce platform in suitable manner. The performance related to both is measured in respect of time and then comparisons between them are made. This in return helps in identification of best between them.

3.4 Sources of Data

This study includes the following data sets-

The source of data to run a map reduce program is taken from the online research and data sets in various fields.

1. Dataset1--First data set is from the player categories like (playing, batting,)
2. Dataset2--second data set include different areas like Environmental health(8-2-2012),Family Planning ,Food safety, heart disease and stroke, HIV, Nutrition and overweight, Oral health, occupational health, physical activity and fitness, Vision and hearing.

3. Dataset3--Third data set is taken from the online set which contains the data related to the fielding which contain the player ID, name and other columns.

3.5 Research Design

- Review of literature for defining the virtualization with Hadoop tools.
- Study of different virtualization Platforms for analyzing the performance of system.
- Comparing tools and selecting most appropriate out of them.
- Identification of different parameters.
- Installation of virtualization platform on Hadoop for creating the clusters.
- Adding the virtual machines for making the Master and data node within the virtual platform.
- Selection of the dataset for running the Map reduce program.
- Implementation of proposed system is to integrate the virtualization with Hadoop for better performance.
- Comparing the proposed module with the existing one for analyzing performance.

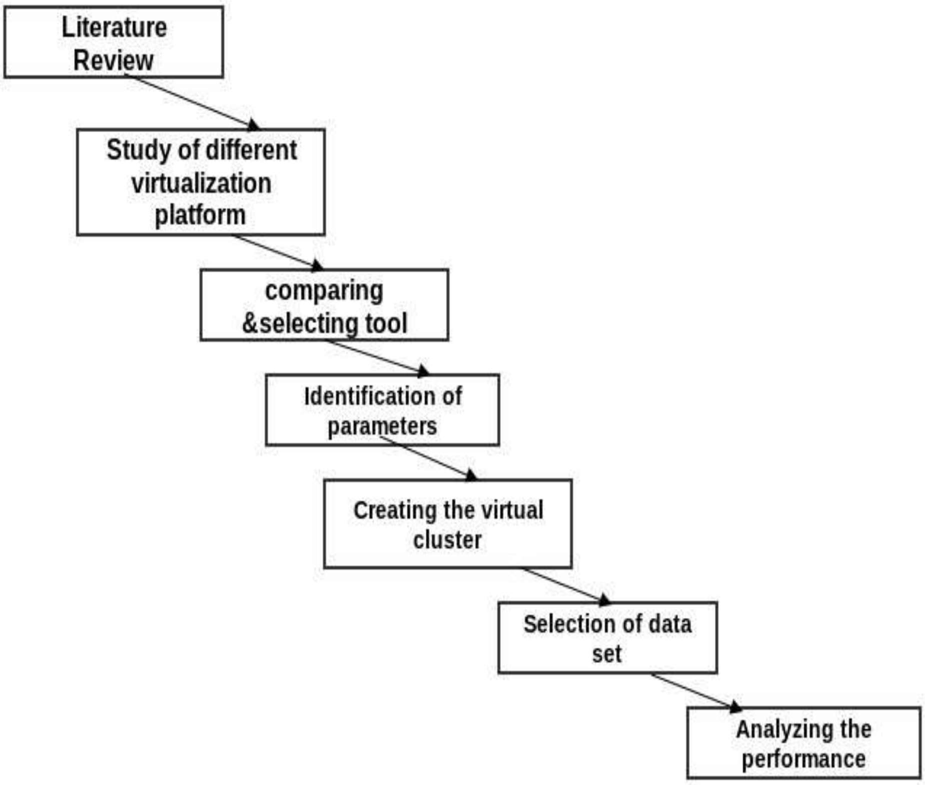
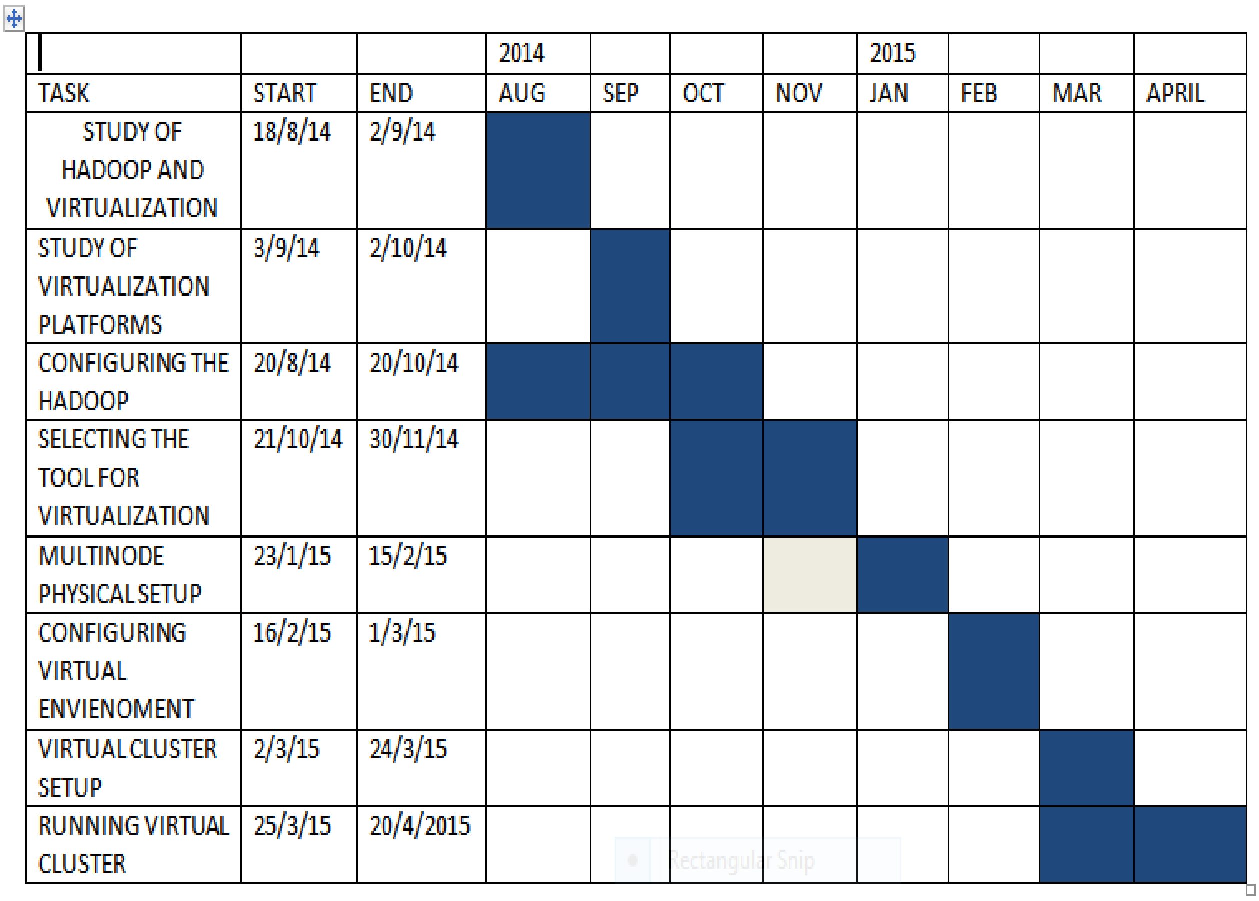

Literature Review
Study of different virtualization platform
comparing &selecting tool
Identification of parameters
Creating the virtual cluster
Selection of data set
Analyzing the performance

Fig11: Research Design

3.6 Monthly Progress Report

Table 1: Gantt Chart



			2014				2015			
TASK	START	END	AUG	SEP	OCT	NOV	JAN	FEB	MAR	APRIL
STUDY OF HADOOP AND VIRTUALIZATION	18/8/14	2/9/14	■							
STUDY OF VIRTUALIZATION PLATFORMS	3/9/14	2/10/14		■						
CONFIGURING THE HADOOP	20/8/14	20/10/14	■	■	■					
SELECTING THE TOOL FOR VIRTUALIZATION	21/10/14	30/11/14			■	■				
MULTINODE PHYSICALSETUP	23/1/15	15/2/15					■			
CONFIGURING VIRTUAL ENVIENOMENT	16/2/15	1/3/15						■		
VIRTUALCLUSTER SETUP	2/3/15	24/3/15							■	
RUNNING VIRTUAL CLUSTER	25/3/15	20/4/2015							■	■

CHAPTER 4

RESULTS AND DISCUSSIONS

The mount up of a virtual cluster requires different steps. They are listed below, with the requisite and surroundings. First step is hadoop configuration on virtual cluster consists of master and slave node.

4.1 Configuration of Hadoop--

Hadoop configuration include following steps-

1. $ sudo apt-get update

This will update the package list to the newer versions.

2. $ sudo apt-get install openjdk-7-jdk

This will used to install Java in the system as Hadoop have Java implementation.

3. To check whether Java is installed or not following command is used.

   $ java -version

4. $ cd/usr/lib/jvm

5. $ $ln -s java-7-openjdk-amd64 jdk

6. $ sudo apt-get install openssh-server

   This command is used to install ssh.

7. To add Hadoop user and group

   $ sudo addgroup Hadoop

   $ sudo adduser –ingroup Hadoop hduser

   $ sudo adduser hduser sudo

8. To setup ssh certificate-

   $ ssh-keygen -t rsa -P ''

```
$ cat ~/.ssh/id_rsa.pub>> ~/.ssh/authorized_keys
$ ssh localhost
```

9. Download Hadoop and perform following steps-

```
$ sudo tar  Hadoop-2.2.0.tqr.gz -C /usr/local
$ cd/usr/local
$ sudo mv Hadoop-2.2.0 Hadoop
$ sudo chown -R hduser : Hadoop Hadoop
```

10. Now 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/
```

11. Now run the following command to check the version of Hadoop-

```
$ Hadoop version
```

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

4.2 To build the Hadoop cluster

Virtual machine installation

Oracle Virtual-box (VM) was used for creating virtual machines. It is a software collection for maintaining and creating the VMS. It is configured on the host operating device and then run as an application. Many different devices are virtualized and run to reduce 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:

To set the Network connections on Virtual machine

For network settings in virtual machine settings need to be changed. Three network Vtun, uml utilities and Bridge utils need to be installed on the system. Network address translation Adapter is exists in the system by default and bridge and tap networks are created. When theses setting are done successfully on each machine then Ssh is configured on both the machines.

In each machine the setup of the bridge and tap device is done automatically at the booting time.

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 mentioned.

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

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

In following files, changes are done--

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>

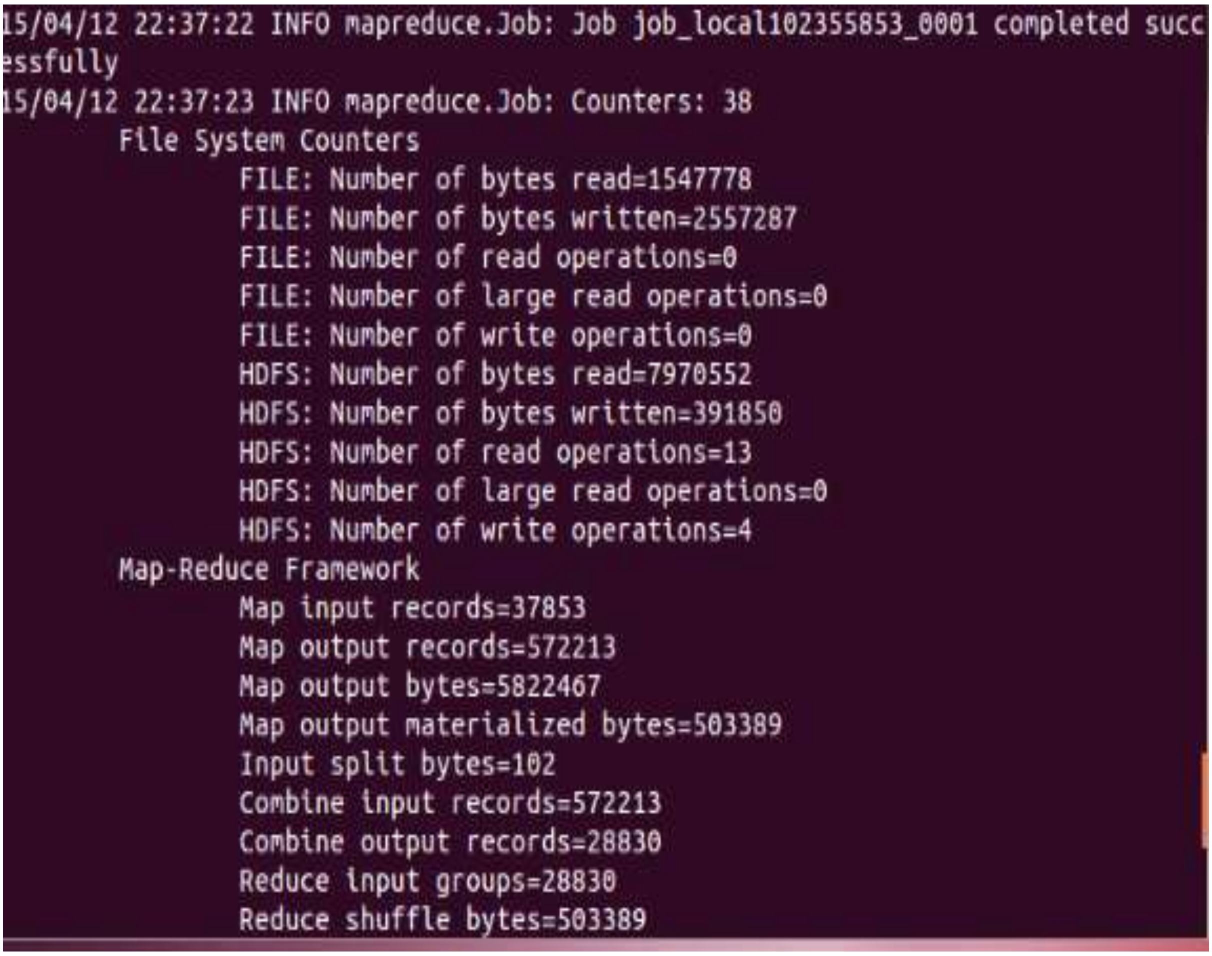
- Executing Map Reduce: if the data files in HDFS, the MapReduce job is executed.
- $HADOOP_PATH/bin/hadoop jar<program-name>
- <input file location><output file location>

4.3 Performance -

Hadoop in virtual cluster are appropriately configured according to requirements.

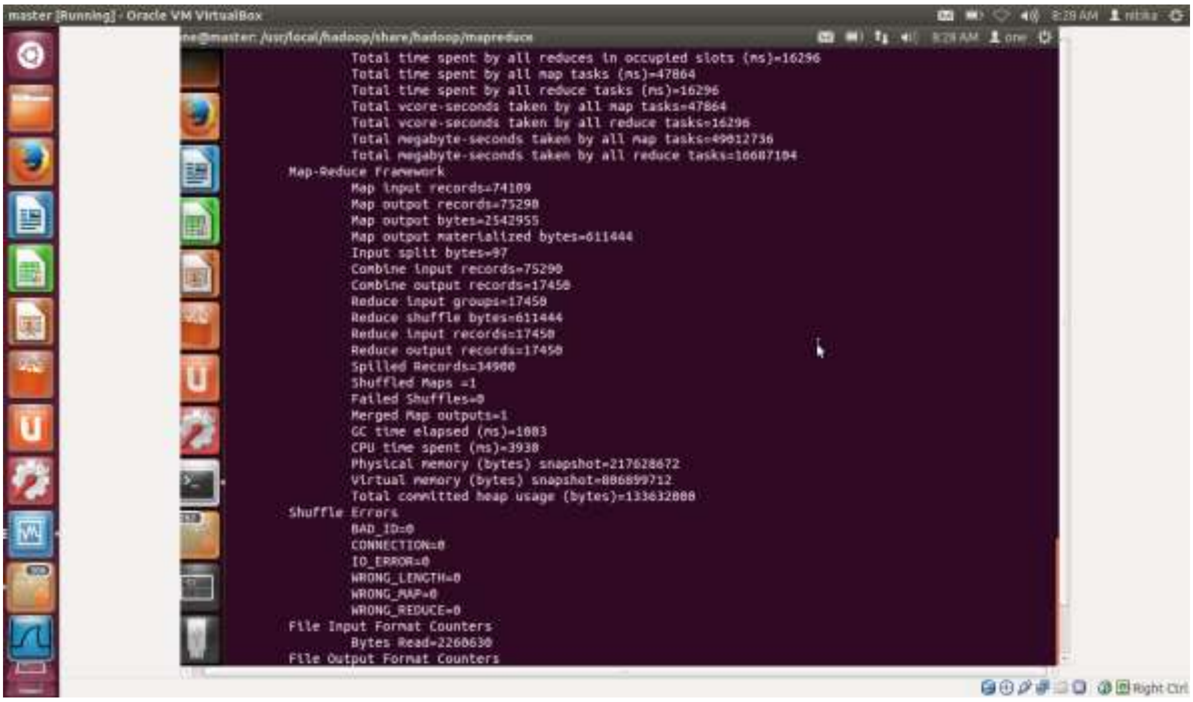
Dataset 1

Performance of Hadoop in Virtual Cluster:



```
15/04/12 22:37:22 INFO mapreduce.Job: Job job_local102355853_0001 completed succ
essfully
15/04/12 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
```

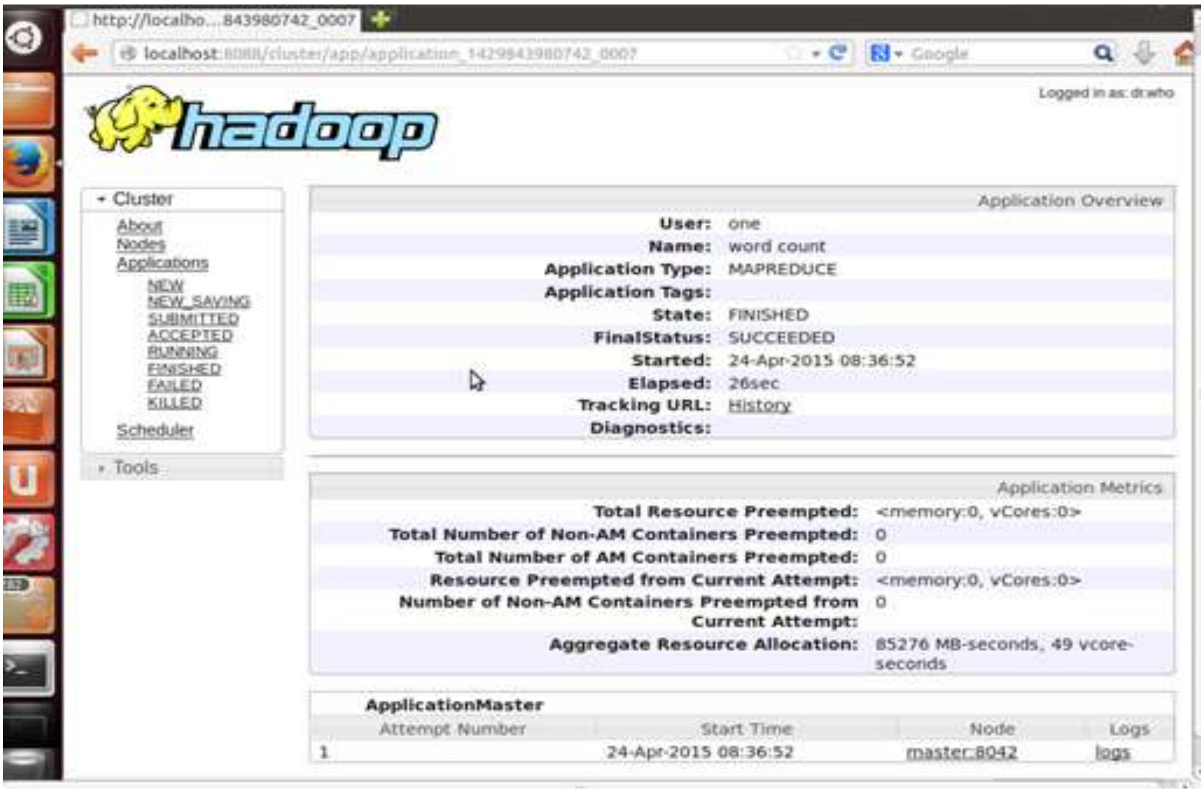
Fig 12a: Hadoop Performance for dataset 1



master [Running] - Oracle VM VirtualBox
one@master: /usr/local/hadoop/share/hadoop/mapreduce
8:28 AM
one

```
		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=49012736
		Total megabyte-seconds taken by all reduce tasks=16687104
	Map-Reduce Framework
		Map input records=74109
		Map output records=75290
		Map output bytes=2542955
		Map output materialized bytes=611444
		Input split bytes=97
		Combine input records=75290
		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=886899712
		Total committed heap usage (bytes)=133632000
	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=2260630
	File Output Format Counters
```

Fig 12b: Hadoop Performance for dataset 1



Application Overview

User: one
Name: word count
Application Type: MAPREDUCE
Application Tags:
State: FINISHED
FinalStatus: SUCCEEDED
Started: 24-Apr-2015 08:36:52
Elapsed: 26sec
Tracking URL: History
Diagnostics:

Application Metrics

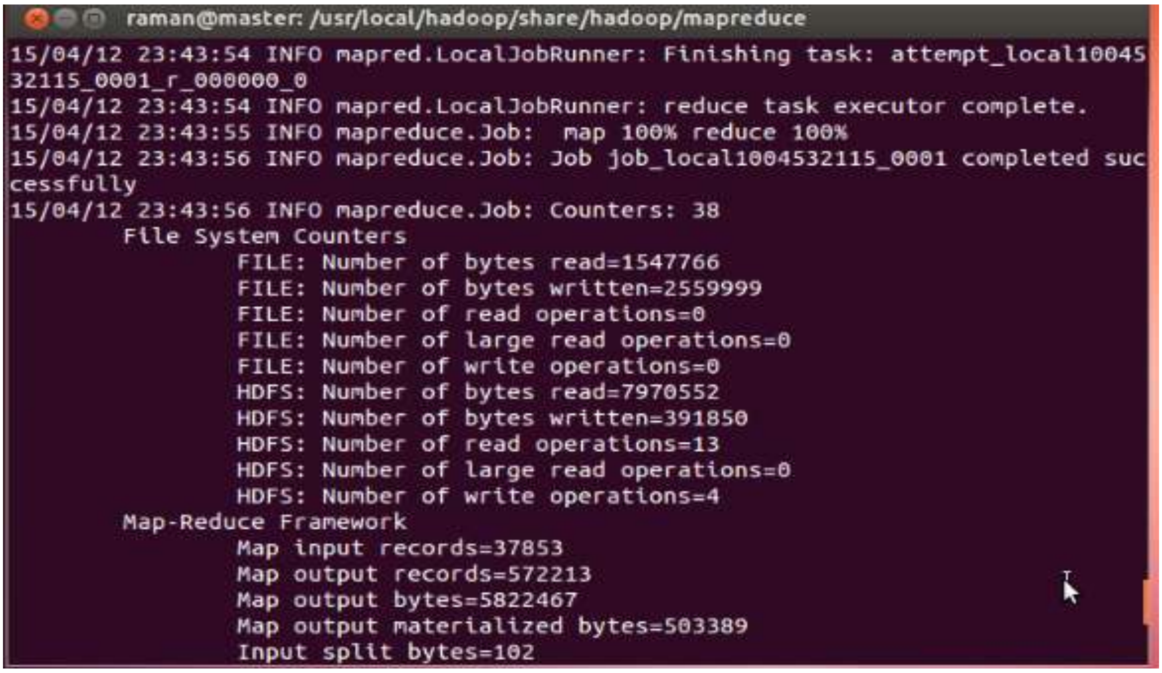
Total Resource Preempted: <memory:0, vCores:0>
Total Number of Non-AM Containers Preempted: 0
Total Number of AM Containers Preempted: 0
Resource Preempted from Current Attempt: <memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt: 0
Aggregate Resource Allocation: 85276 MB-seconds, 49 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	24-Apr-2015 08:36:52	master:8042	logs

Fig 12c: Hadoop Performance for dataset 1

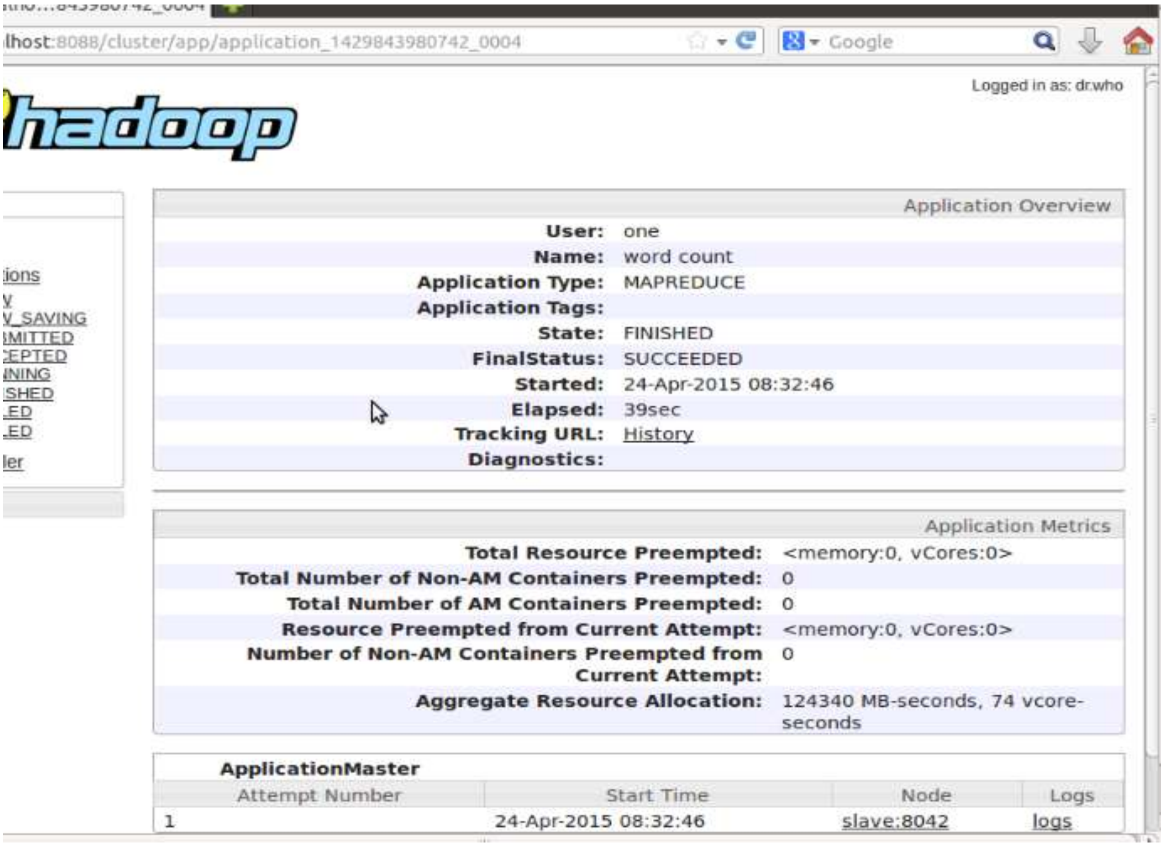
Dataset2:



raman@master: /usr/local/hadoop/share/hadoop/mapreduce

```
15/04/12 23:43:54 INFO mapred.LocalJobRunner: Finishing task: attempt_local10045
32115_0001_r_000000_0
15/04/12 23:43:54 INFO mapred.LocalJobRunner: reduce task executor complete.
15/04/12 23:43:55 INFO mapreduce.Job:  map 100% reduce 100%
15/04/12 23:43:56 INFO mapreduce.Job: Job job_local1004532115_0001 completed suc
cessfully
15/04/12 23:43:56 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=1547766
                FILE: Number of bytes written=2559999
                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
```

Fig 13a: Hadoop Performance for dataset 2



Application Overview

User: one
Name: word count
Application Type: MAPREDUCE
Application Tags:
State: FINISHED
FinalStatus: SUCCEEDED
Started: 24-Apr-2015 08:32:46
Elapsed: 39sec
Tracking URL: History
Diagnostics:

Application Metrics

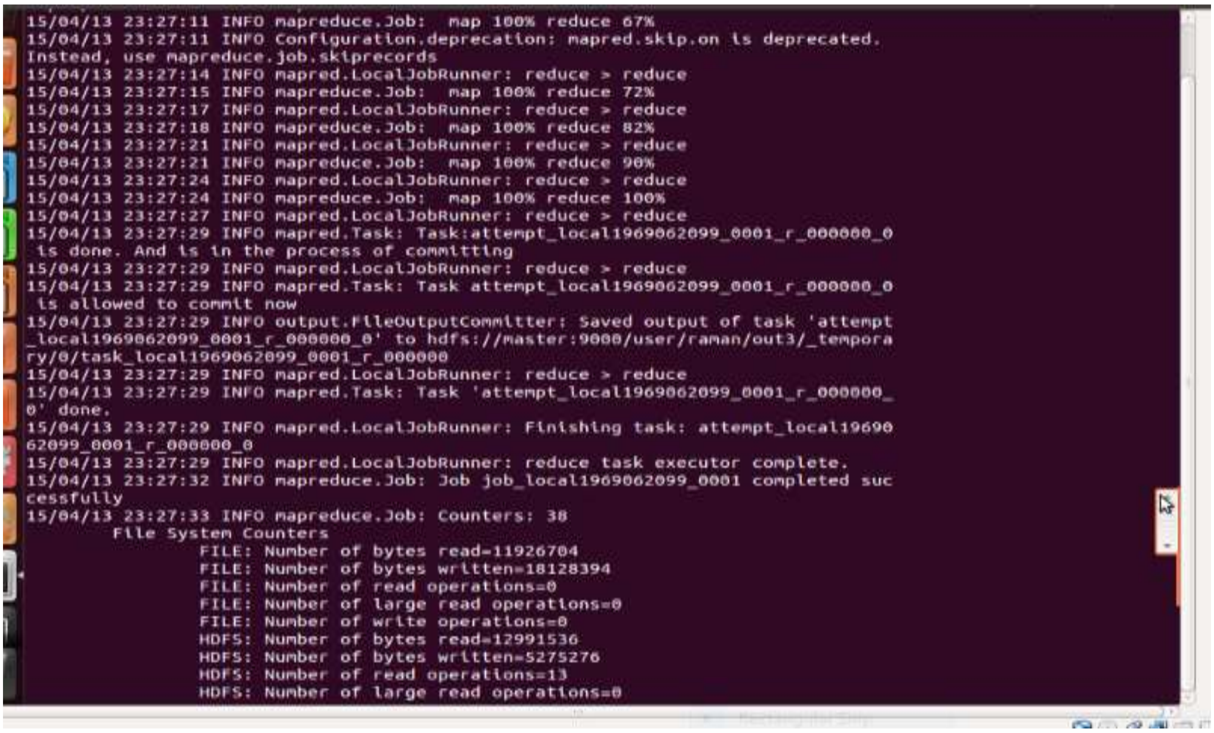
Total Resource Preempted: <memory:0, vCores:0>
Total Number of Non-AM Containers Preempted: 0
Total Number of AM Containers Preempted: 0
Resource Preempted from Current Attempt: <memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt: 0
Aggregate Resource Allocation: 124340 MB-seconds, 74 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	24-Apr-2015 08:32:46	slave:8042	logs

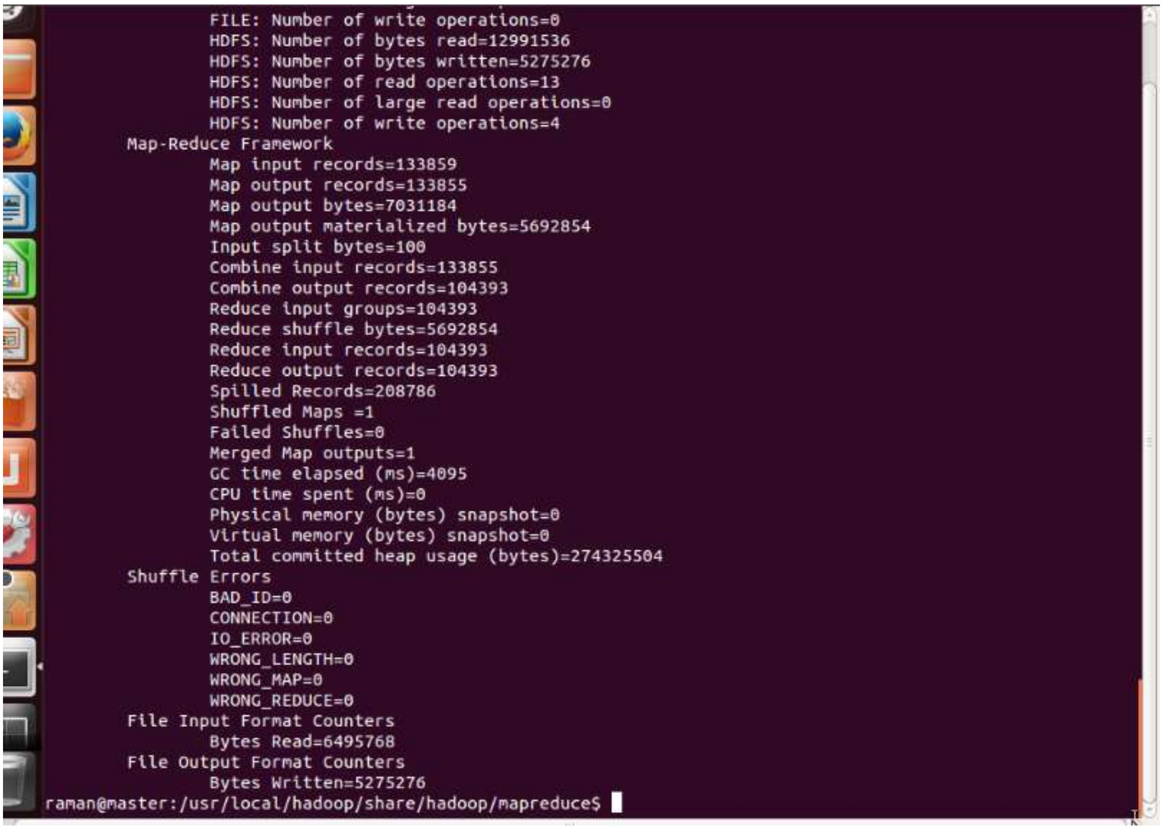
Fig 13b: Hadoop Performance for dataset 2

DataSet3:



```
15/04/13 23:27:11 INFO mapreduce.Job:  map 100% reduce 67%
15/04/13 23:27:11 INFO Configuration.deprecation: mapred.skip.on is deprecated.
Instead, use mapreduce.job.skiprecords
15/04/13 23:27:14 INFO mapred.LocalJobRunner: reduce > reduce
15/04/13 23:27:15 INFO mapreduce.Job:  map 100% reduce 72%
15/04/13 23:27:17 INFO mapred.LocalJobRunner: reduce > reduce
15/04/13 23:27:18 INFO mapreduce.Job:  map 100% reduce 82%
15/04/13 23:27:21 INFO mapred.LocalJobRunner: reduce > reduce
15/04/13 23:27:21 INFO mapreduce.Job:  map 100% reduce 90%
15/04/13 23:27:24 INFO mapred.LocalJobRunner: reduce > reduce
15/04/13 23:27:24 INFO mapreduce.Job:  map 100% reduce 100%
15/04/13 23:27:27 INFO mapred.LocalJobRunner: reduce > reduce
15/04/13 23:27:29 INFO mapred.Task: Task:attempt_local1969062099_0001_r_000000_0
 is done. And is in the process of committing
15/04/13 23:27:29 INFO mapred.LocalJobRunner: reduce > reduce
15/04/13 23:27:29 INFO mapred.Task: Task attempt_local1969062099_0001_r_000000_0
 is allowed to commit now
15/04/13 23:27:29 INFO output.FileOutputCommitter: Saved output of task 'attempt
_local1969062099_0001_r_000000_0' to hdfs://master:9000/user/raman/out3/_tempora
ry/0/task_local1969062099_0001_r_000000
15/04/13 23:27:29 INFO mapred.LocalJobRunner: reduce > reduce
15/04/13 23:27:29 INFO mapred.Task: Task 'attempt_local1969062099_0001_r_000000_
0' done.
15/04/13 23:27:29 INFO mapred.LocalJobRunner: Finishing task: attempt_local19690
62099_0001_r_000000_0
15/04/13 23:27:29 INFO mapred.LocalJobRunner: reduce task executor complete.
15/04/13 23:27:32 INFO mapreduce.Job: Job job_local1969062099_0001 completed suc
cessfully
15/04/13 23:27:33 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=11926704
                FILE: Number of bytes written=18128394
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=12991536
                HDFS: Number of bytes written=5275276
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
```

Fig 14a: Hadoop Performance for dataset 3

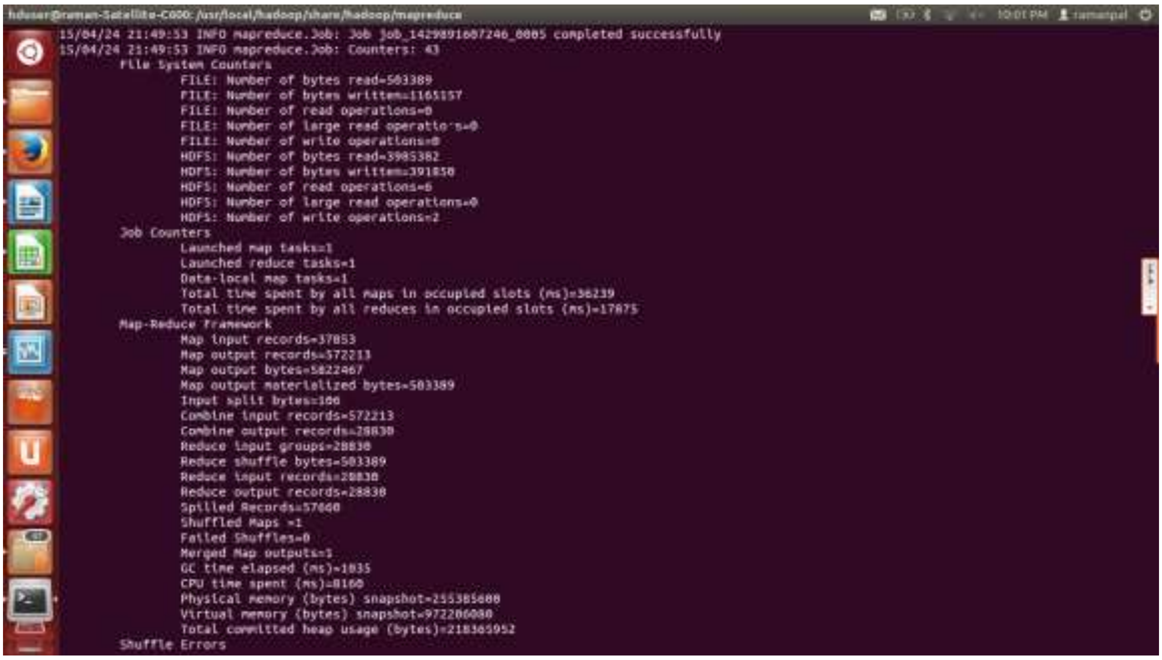


```
                FILE: Number of write operations=0
                HDFS: Number of bytes read=12991536
                HDFS: Number of bytes written=5275276
                HDFS: Number of read operations=13
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=133859
                Map output records=133855
                Map output bytes=7031184
                Map output materialized bytes=5692854
                Input split bytes=100
                Combine input records=133855
                Combine output records=104393
                Reduce input groups=104393
                Reduce shuffle bytes=5692854
                Reduce input records=104393
                Reduce output records=104393
                Spilled Records=208786
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=4095
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=274325504
        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=6495768
        File Output Format Counters
                Bytes Written=5275276
raman@master:/usr/local/hadoop/share/hadoop/mapreduce$
```

Fig 14b: Hadoop Performance for dataset 3

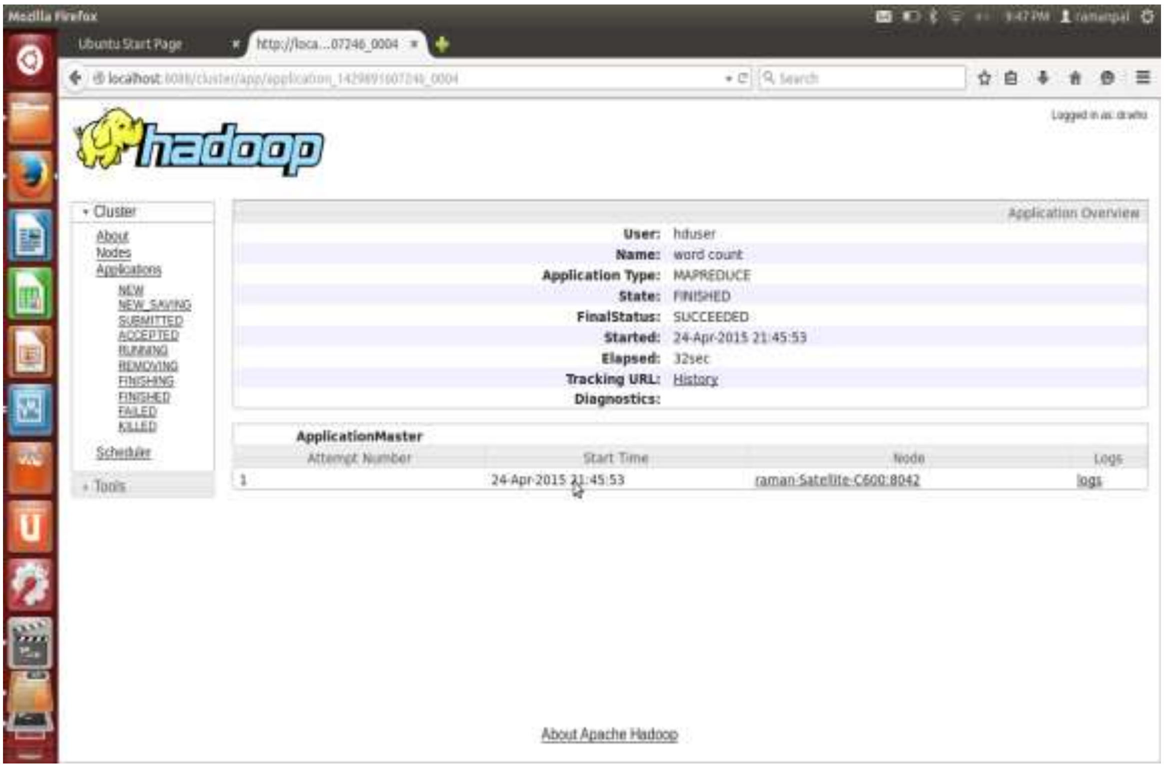
Performance with Physical:

DataSet1:



```
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=1165157
                FILE: Number of read operations=0
                FILE: Number of large read operatio s=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3985382
                HDFS: Number of bytes written=391850
                HDFS: Number of read operations=6
                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=106
                Combine input records=572213
                Combine output records=28830
                Reduce input groups=28830
                Reduce shuffle bytes=503389
                Reduce input records=28830
                Reduce output records=28830
                Spilled Records=57660
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=1035
                CPU time spent (ms)=8160
                Physical memory (bytes) snapshot=255385600
                Virtual memory (bytes) snapshot=972206080
                Total committed heap usage (bytes)=218365952
        Shuffle Errors
```

Fig 15a: Hadoop Performance for dataset 1



Application Overview

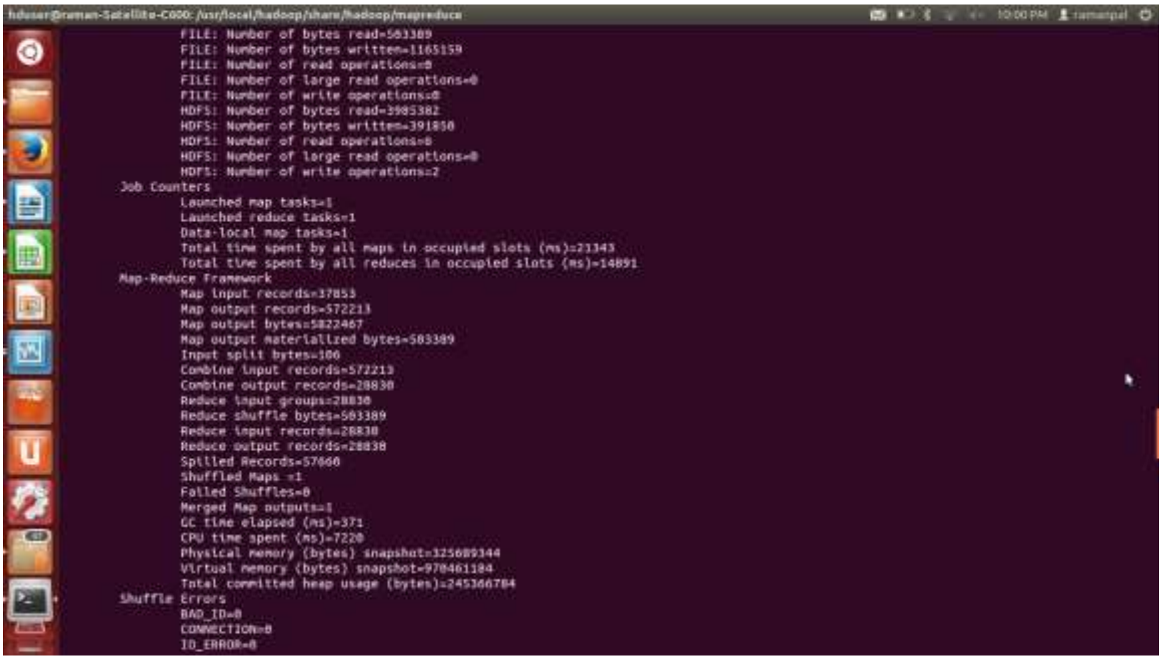
User: hduser
Name: word count
Application Type: MAPREDUCE
State: FINISHED
FinalStatus: SUCCEEDED
Started: 24-Apr-2015 21:45:53
Elapsed: 32sec
Tracking URL: History
Diagnostics:

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	24-Apr-2015 21:45:53	raman-Satellite-C600:8042	logs

Fig15 b: Hadoop Performance for dataset 1

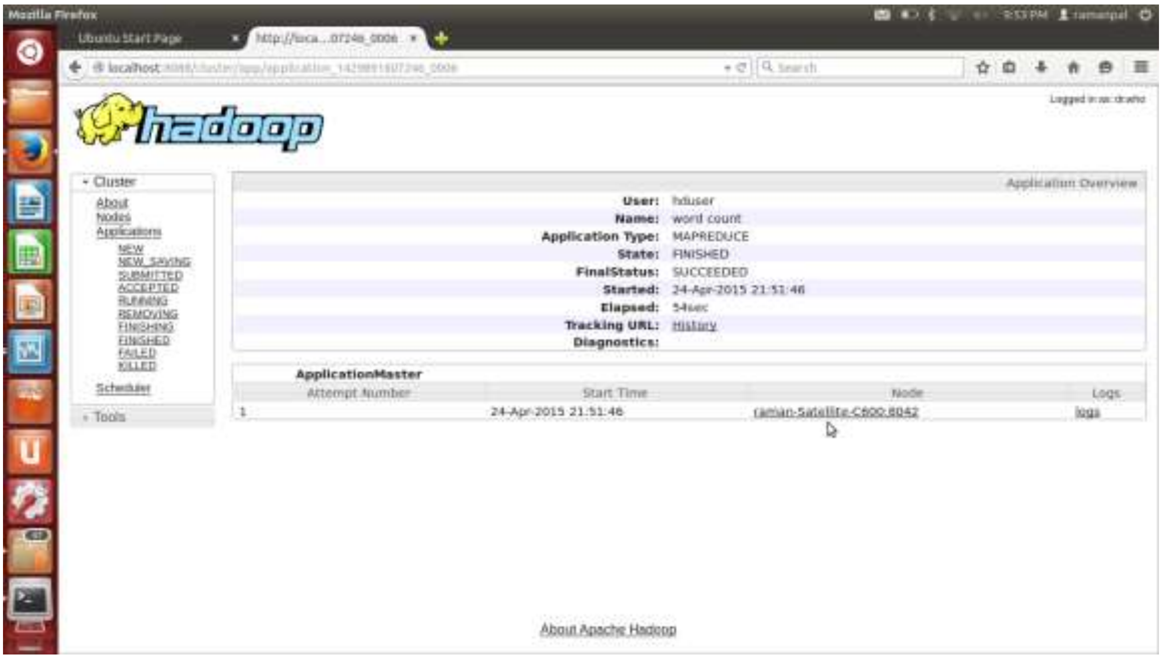
DataSet2:



hduser@raman-Satellite-C600: /usr/local/hadoop/share/hadoop/mapreduce
10:00 PM
ramanpal

```
                FILE: Number of bytes read=583389
                FILE: Number of bytes written=1165159
                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=391850
                HDFS: Number of read operations=6
                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)=21343
                Total time spent by all reduces in occupied slots (ms)=14891
        Map-Reduce Framework
                Map input records=37853
                Map output records=572213
                Map output bytes=5822467
                Map output materialized bytes=583389
                Input split bytes=106
                Combine input records=572213
                Combine output records=28830
                Reduce input groups=28830
                Reduce shuffle bytes=583389
                Reduce input records=28830
                Reduce output records=28830
                Spilled Records=57660
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=371
                CPU time spent (ms)=7220
                Physical memory (bytes) snapshot=325689344
                Virtual memory (bytes) snapshot=970461184
                Total committed heap usage (bytes)=245366784
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
```

Fig 16a: Hadoop Performance for dataset 2



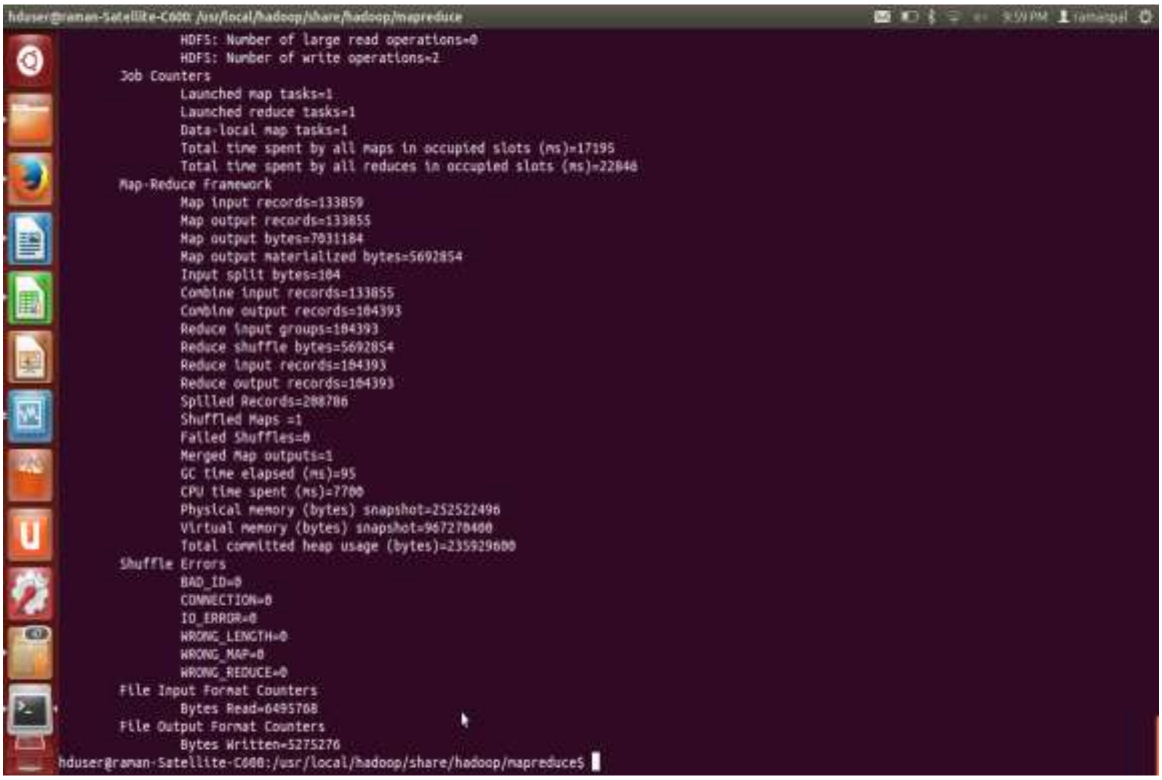
User: hduser
Name: word count
Application Type: MAPREDUCE
State: FINISHED
FinalStatus: SUCCEEDED
Started: 24-Apr-2015 21:51:46
Elapsed: 54sec
Tracking URL: History
Diagnostics:

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	24-Apr-2015 21:51:46	raman-Satellite-C600:8042	logs

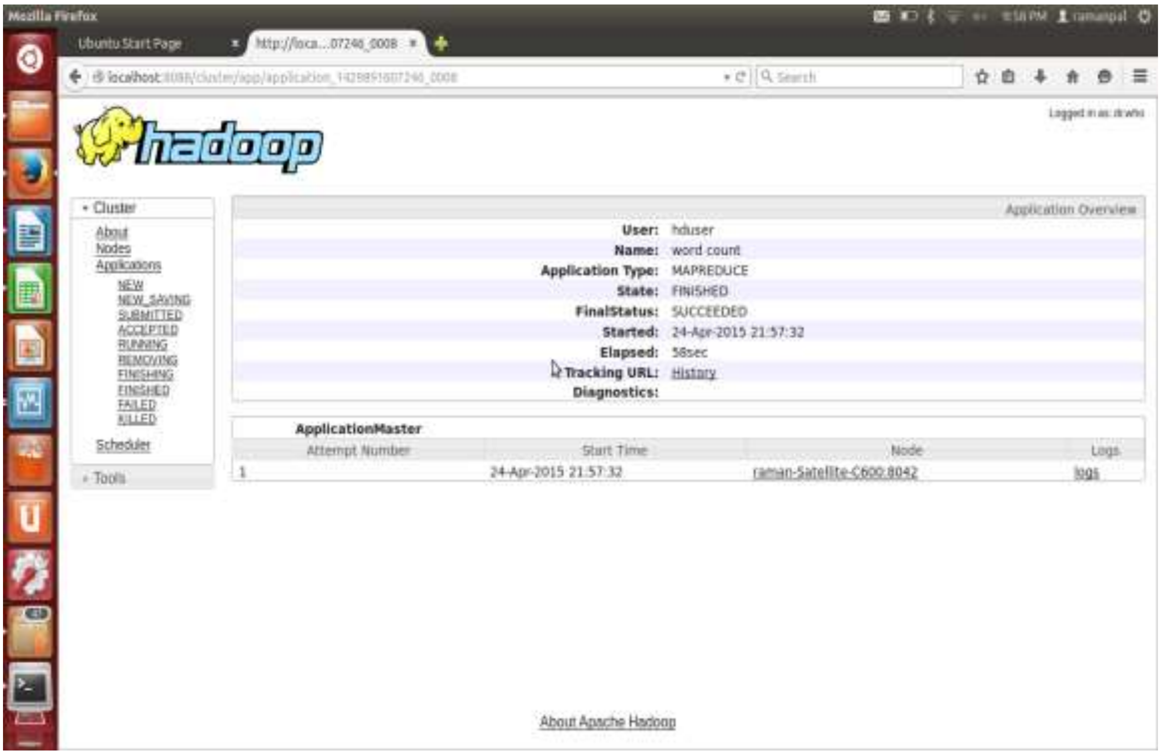
Fig 16b: Hadoop Performance for dataset 2

DataSet3:



```
                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)=22046
        Map-Reduce Framework
                Map input records=133859
                Map output records=133855
                Map output bytes=7031184
                Map output materialized bytes=5692854
                Input split bytes=104
                Combine input records=133855
                Combine output records=104393
                Reduce input groups=104393
                Reduce shuffle bytes=5692854
                Reduce input records=104393
                Reduce output records=104393
                Spilled Records=208786
                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=967270400
                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 17a: Hadoop Performance for dataset 3



User: hduser
Name: word count
Application Type: MAPREDUCE
State: FINISHED
FinalStatus: SUCCEEDED
Started: 24-Apr-2015 21:57:32
Elapsed: 58sec
Tracking URL: History
Diagnostics:

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	24-Apr-2015 21:57:32	raman-Satellite-C600:8042	logs

Fig 17b: Hadoop Performance for dataset 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 shown in the (Table 2).

Table2: Time taken by physical and virtual cluster

Name	Operation	Size of file	Time taken in Physical cluster (Sec)	Time Taken by Virtual cluster(Sec)
Dataset1	Word Count	2.15MB	32	26
Dataset2	Word Count	3.80MB	54	39
Dataset3	Word Count	6.20 MB	58	42

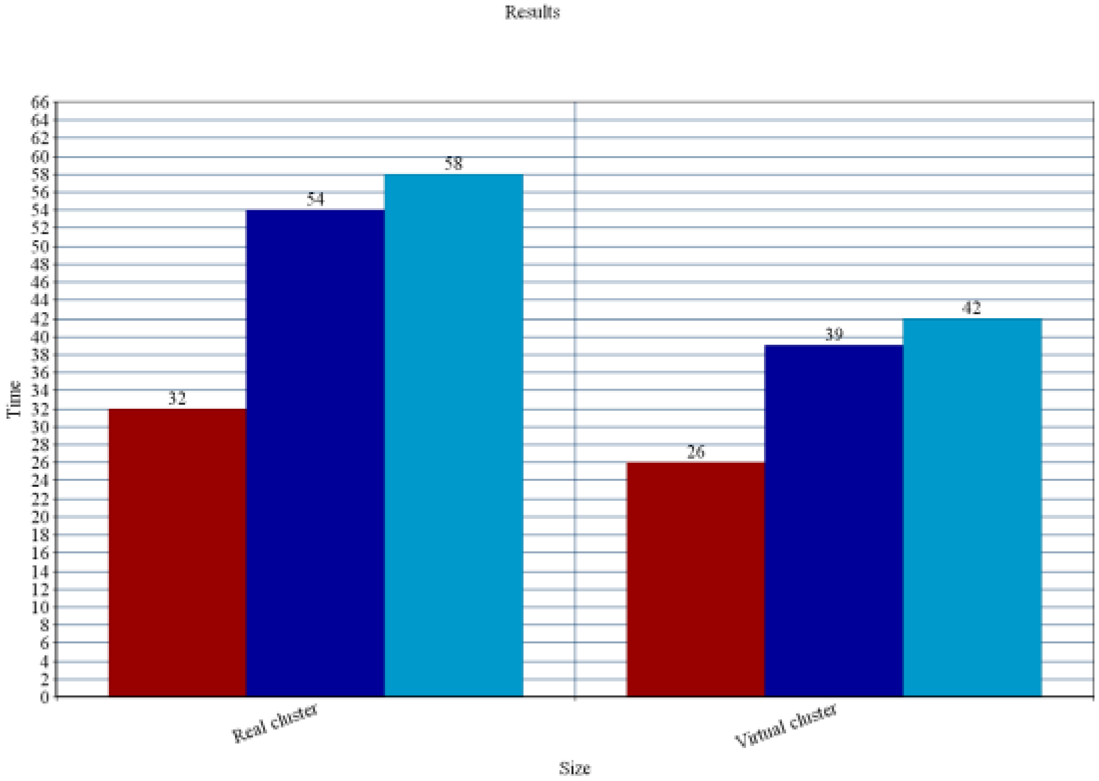


Fig 18: Graph between time taken by physical and virtual cluster.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

In this the performance of the hadoop is analyzed and scalability is also studied as the size of cluster increase with the help of virtualization runtime decreased. Simultaneously running multiple VMs handle the load of the running program and considerable load is divided to the host system on which virtualization is done. The decrease in time is achieved by adding more VMs with limits of the system that use the virtualization helped in better utilization of the resources of the host computer.

CHAPTER 6

REFERENCES

Books--

[1] Jason Verner (2009) Pro Hadoop,Apress,United States of America.

[2] Tom White, (2009) Hadoop:The Definitive Guide ,O'Reilly, United States of America.

Papers--

[3] Ashish Nadkami and Brett Waldman (2013) “Desktop Virtualization and storage Solutions evolve to support Mobile workers and Consumer Devices”

[4]Buell, Jeff (2012) “Virtualized Hadoop Performance with VMware Vsphere5.1”

[5] Daniel A. Menasce, USA “Virtualization Concepts, applications and performance modeling”

[6] David de Nadal Bou (2010) “Support for Managing Hadoop Dynamically Hadoop Clusters”

[7] Vblock Fast Path Desktop Virtualization Platform

[8] Vidyasagar S. D (2013) “A study on “Role of Hadoop in information Technology era””

[9] T-Systems Enterprise Services GmbH, Germany “White Paper Desktop Virtualization. The future of the corporate desktop”

Web pages –

[10] www.centos.org/downaloads.

[11] http://cto.vmware.com/towards-an-elastic-elephant-enabling-hadoop-for-the-cloud/

[12] http://communities.vmware.com/blogs/drummonds/2009/02/17/building-block-architecture-for-superior-performance

[13] http://hadoop.apache.org/.

[14] http://en.wikipedia.org/wiki/Apache_Hadoop

[15] https://pubs.vmware.com/vsphere-51 index.html.

[16] www.vmware.com/solutions/consolidation/mission cri- tical.html

[17] www.vmware.com

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

[19] http://windowsitpro.com/article/articleid/102427/freevirtualizationplatforms.html.

[20] http://wiki.apache.org/hadoop/PoweredBy

Appendix A

Glossary of Terms **Page no**

Appendix B

ABBREVIATIONS

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