ENHANCING PERFORMANCE OF MAPREDUCE TO IMPROVE EFFICIENCY OF DATA ACCESSIBILITY USING HADOOP

A Dissertation submitted by

Archu Dhamija
Reg. no- 11309791

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)

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

School of: Technology & Science

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: ARCHU DHAMIJA Registration No: 11309791

Batch: 2013-15 Roll No. .....

Session: ..... Parent Section: K2305

Details of Supervisor: Designation: Asst Professor

Name: Kamaldeep Kaur Qualification: M.Tech

U.ID: 18308 Research Experience: Hadoop, Software Arch.

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

PROPOSED TOPICS

1. Map Reduce performance enhancements to improve the performance of the sort benchmarks

2. .....

3. .....

Kamaldeep 13/9/14

Signature of Supervisor

PAC Remarks:

Topic 1 is approved. Research publication is also expected from the work.

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

25/9/14

*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

As the amount of data generated by businesses is increasing at a very fast rate, the opportunities around it are exploding in terms of both scale and variety, so it is quite evident that Hadoop and other big data technologies have lot of scope in it. Hadoop is open source framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. Hadoop helps in handling and optimizing the data very effectively in small amount of time. The analysis of data is performed using Apache Pig tool. Pig is a platform for extracting and analyzing the data which helps in performance enhancement of hadoop and helps the businesses in completing their task which involves large data sets in short time. Pig can be used to estimate the performance of MapReduce jobs as well as to find the optimal configuration settings to use when running the jobs.

CERTIFICATE

This is to certify that **Archu Dhamija (11309791)** has completed M.Tech dissertation titled **"Enhancing Performance of MapReduce to Improve Efficiency of Data Accessibility Using Hadoop"** 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 deepest gratitude to Mrs. Harjeet Kaur (Dissertation Mentor) for her valuable knowledge and expertise. It is only with her guidance that I could take up the initiative of such a good topic of thesis and complete it on time.

I would like to thank Mr. Balraj Singh for his support and guidance. I am also very thankful to Lovely Professional University for giving me an opportunity to propose and implement thesis by the course of Dissertation-II. I am gratified for the successful completion of my thesis implementation. I would also like to convey thanks to all my friends who gave their full support and encouraged me for this thesis work.

Archu Dhamija

11309791

DECLARATION

I hereby declare that the dissertation entitled **"Enhancing Performance of MapReduce to Improve Efficiency of Data Accessibility Using Hadoop"** 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

Archu Dhamija

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER1
INTRODUCTION

1.1Big Data

Big data is extremely large sets of data which is analyzed computationally to reveal patterns, threads and their associations. Big data includes data sets of different sizes which are beyond the ability of commonly used software and traditional databases to capture, manage and process data within tolerable elapsed time. It is used to describe exponential growth of the data and its availability which is arriving from different multiple sources at an alarming velocity, volume and variety. Big data consists of set of technologies and techniques that require different forms of integration to find the hidden values from large data sets that are complex and of massive scale. The type of data includes surveillance data, e-mail messages, business transactions, activity logs and the data collected from different sensors. The optimal processing power and analytics capabilities are also needed to extract the meaningful value from big data. Big Data is a special as it represents significant information which can open new doors and the way information is analyzed to help open those doors. Big data has the potential to help companies in improving their operations and to take appropriate decisions. An application that involves big data includes transactional (Facebook) and analytics (ClickFox).

ClickFox Experience Analytics platform give first and foremost scalable analytical application which connects cross-channel interconnection data in meaningful Statistics and knowledge. ClickFox deals with the customer services as mapping of customers interactions. It collects and then integrates the data of customers to analyze which are satisfied and which are dissatisfied and remedial steps are taken in order to satisfy them. With the help of ClickFox, companies can also visualize the experience of their customers with each factor of their company.

Facebook - a social networking website, is considered as one of the largest repository of personal files. It grew 129% from 2011 to 2013. There are about 1.393 billion users uploading their personal images and videos and sharing data with their friends. The Facebook deals with the unstructured data as it include different format of images, videos and documents which is unable to store in tables. One of the challenges faced by Zuckerberg is to increase the speed of Facebook for mobile users so that revenue from them increases in their growth. There are about 1.393 billion users which is around one

fifth population of world, uploading their personal images and videos and sharing data with their friends.

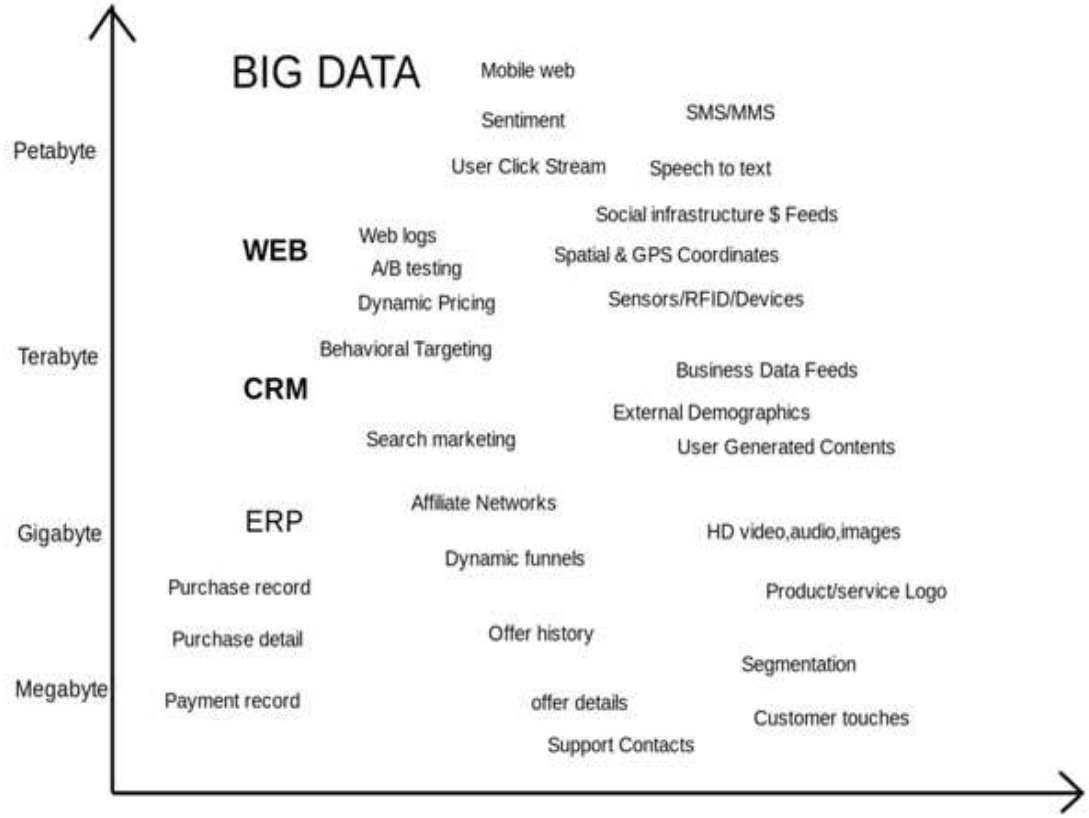

Fig 1: Evolution of Big Data

Some of the characteristics of big data are-

- Petabytes/Exabyte's of data(Fig 1) – it is fifth power of 1000 / two to sixtieth powers of bytes
- Millions/billions of people involved in accessing and manipulating it,
- Billions/trillions of records of different entities involved,
- Loosely-structured and often distributed data includes different physical machines and data which is replicated in machines,
- Flat schema's with few complex interrelationships helps in defining all the records and field characteristics ,
- Often involving time-stamped events as it returns the time when event was created,
- Often made up of incomplete data such as data with missing values.

1.2 Categories of Big Data

Structured data-- Structured data refers to information with a high degree of organization for example inclusion in a relational database is seamless and easily searchable by simple, data is resides in fixed fields within file or a record. The structured data is easily entered, stored, queried and analyzed. Structured data can also be machine readable, is locatable and is predefined. Each entry in structured data has predefined length and is of same order.

Unstructured data: Unstructured data refers to the data or information that does not have a predefined data model, is not organized and is not predictable. This results in irregularities and ambiguities that make it difficult to understand using traditional computer programs. The lack of structure makes compilation a time and energy consuming task. The techniques such as data mining, natural language processing and text analytics are different methods to extract pattern. For example- text, audio, sound, images etc.

Semi-structured data: The semi-structured data may contain rational data made up of records but data may not be organized in a recognizable structure as the missing values can be easily described in database model. Data having similar entities group together but the entities in same group may not have same attributes. The semi structured data is of two types-XML and JSON. Size and type of attributes may differ in different group. The advantages of semi structured data--

- The information of data sources cannot be constrained by the schema.
- It provides a flexible format for data exchange between different types of databases.
- The schema can easily be changed.

1.3 5 V's of big Data

Volume – The data quantity is very important as it is size of the data which helps in determining the value and the potential of the data and help in making decision whether the data is considered as big data or not. The term Big data itself contains a term which refers to size and hence the characteristic of it.

Variety – variety refers to the category to which the data belongs. The variety of data is essential fact and must be known to the data analysts. This helps the persons who analyzing the data and associated with the data as it helps them in effectively use of the data to their advantage and upholding importance of big data.

Velocity – The velocity of the speed refers to the speed of generation of data or how fast the data is produced by different organization (Fig2). It also refers to the speed of processing of data to meet the challenges and demand for the growth and development of different sectors. The flow of data is massive and continuous.

Value – Big data is considered critical to meet the strategic goals and objectives of large organizations. The key use of big data in businesses is assistance in decision making process. Increase number of customers, increase basket margins, Improve market effectiveness, and Increase inventory turns are some of the valuable assets of big data.

Veracity – Big data veracity refers to the noise and abnormality of the data. The data stored or mined must be accurate to make right decisions, so accuracy of data is dealt with veracity of data.

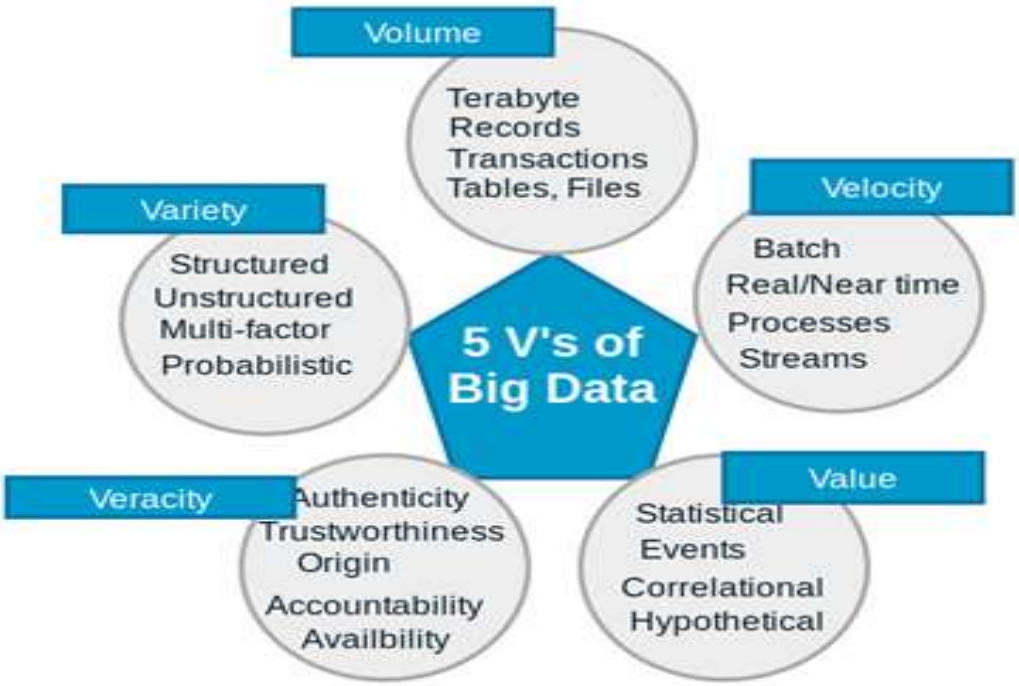

Fig 2: 5V’s of Big Data

1.4 Significance of Big data

In the present era, 2.5 quintillion bytes of data is produced as data is produced from various sources like sensor data, posts to social media sites, digital pictures and videos, transactional data etc. Big data is crucial to businesses as it is even vital for them to analyze the growing data. Big data can help employees in decision making. Companies are able to gain more complete understanding of their customers, business, products when data is effectively and efficiently captured which can lead to efficiency improvements, increased sales, lower costs, better customer service, and improved products and service. With the big data, new growth opportunities and entirely new company processes are created such as analysis of industrial data. Big data gives warranty in innovative processing for variety of new and existing data to provide business benefit. Some of widely cited examples of big data are--

1. Information technology logs, used to improve IT troubleshooting and security breach detection, speed and effectiveness.

2. Historical call center information, used to improve customer interaction and satisfaction.

3. Social media content, used to understand customer sentiment and to improve products, service and customer interaction.

4. Financial transaction information, used to access risk and to take preventive actions.

1.5 Challenges of big data

The major challenges related to big data are workload diversity, data security, data manageability, cost, analysis of data, capture, sharing, storage, transfer, visualization and privacy of data.

A challenge for IT researchers is fast growing rate of data exceeding the capability to design appropriate system to handle the data as well as analyze and extract data for decision making as it is difficult to handle big data with traditional database system.

- Understanding and Utilizing Big Data– It is very important task in industrial field that deal with big data to understand the syntax and semantics of data associated with that data and determining the best use of data according to industrial practices.
- Privacy, Security, and Regulatory Considerations- it is difficult for the companies

and industries to obtain reliable grasp on the content of big data and to capture, secure it, so that the confidential data related to their customers and business is not disclosed to unauthorized parties.

- Repository and scrapping of big data-- the current value of the big data will lose its value over time and have wide range of content and structure so new tools, technologies are employed to archive and delete big data.

In view of security two issues are there- securing customer information and using big data techniques to safely analyze data. In bank management system, data consists of information of customers, partners and systems, billions of the transactions. They need Big data to have insights about customers, their behavior and market. So Banks faced challenge in maintaining workload diversity, data storage and security.

The storage of big data needs devices to store data such as RAM, Optical media devices, magnetic media device etc. which increase in cost.

In order to deal with the challenges of big data, Map Reduce is one of the best solutions as it is capable to handle raw data parallely in reasonable time.

1.6 MapReduce

MapReduce is a programming paradigm and consists of corresponding implementation of handling and production of large data sets with different parallel methods. MapReduce is outlined by Google that focus on resolution of processing huge data such as sensor data, transactional data, web logs etc.

MapReduce is a framework that employs large number of nodes for solving the problems in parallel. MapReduce have the ability of dividing the work in distributed system and take privilege of parallel processing as well as lower down the network bandwidth.

MapReduce consists of two distinct parts--

Map Phase-- Map is data gathering stage as it divides the data into smaller parts and then process that pieces of data parallely. 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.

All the blocks of data in map are accomplished in parallel and independent of others. Map phase is responsible for modifying the blocks of data into key/value pair. All the blocks are then forwarded to the reduce phase. Reduce process that block of data and produce the final output.

MapReduce can also viewed as following step--

- Fork input into smaller parts-- the first step is to divide the input into smaller manageable parts. For M map worker nodes, input is divided into M parts.
- Creation of master and worker nodes as master is able to allocate the jobs to the worker nodes and also keep track of their performance and analyze the results.
- Map function is executed by each block of data and produces corresponding key/value pairs.
- The result of the MapReduce is stored into local disk of map node. Afterwards, data is divided into R sections by Partition function.
- Master informs the reducer to start their work. Here data is sorted by keys as many occurrence of keys and values are grouped together to obtain data according to key.
- The keys and values are passed through Reduce function and final output is obtained.

These steps are executed sequentially as in Fig 3 each step starts only after the completion of previous step.

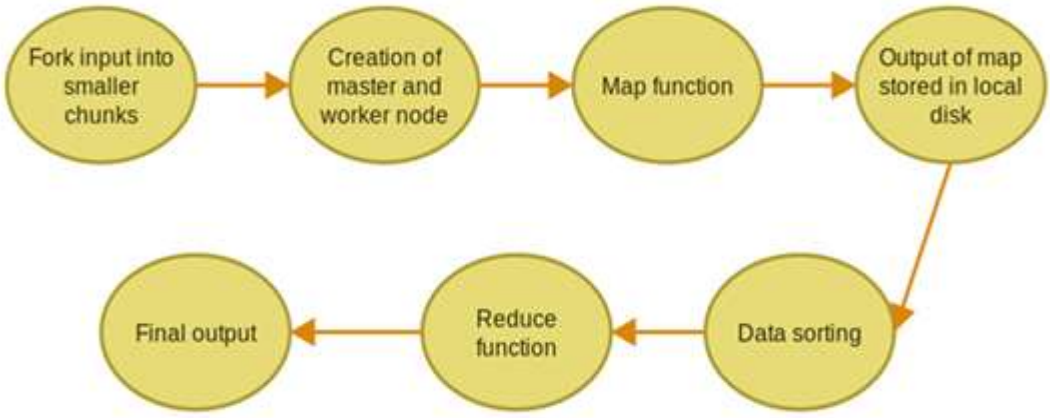

Fig 3: Steps of working of MapReduce

The Map and Reduce Functions can also be explained in terms of data structured in (key, value) pairs. Map uses the data of one domain and returns the output in different domain of data.

MAP (K1, V1) -----> LIST (K2, V2)

This will produces the list of intermediate keys and value for each block of data as shown in Fig4. MapReduce gathers the data of same key for better understanding. The reduce function is applied parallely to each output of map phase.

REDUCE (K2, LIST (V2)) ----> LIST (V3)

Thus, the MapReduce changes a list of (KEY, VALUE) pairs into list of values and this characteristic of MapReduce differentiate it from other programming paradigms.

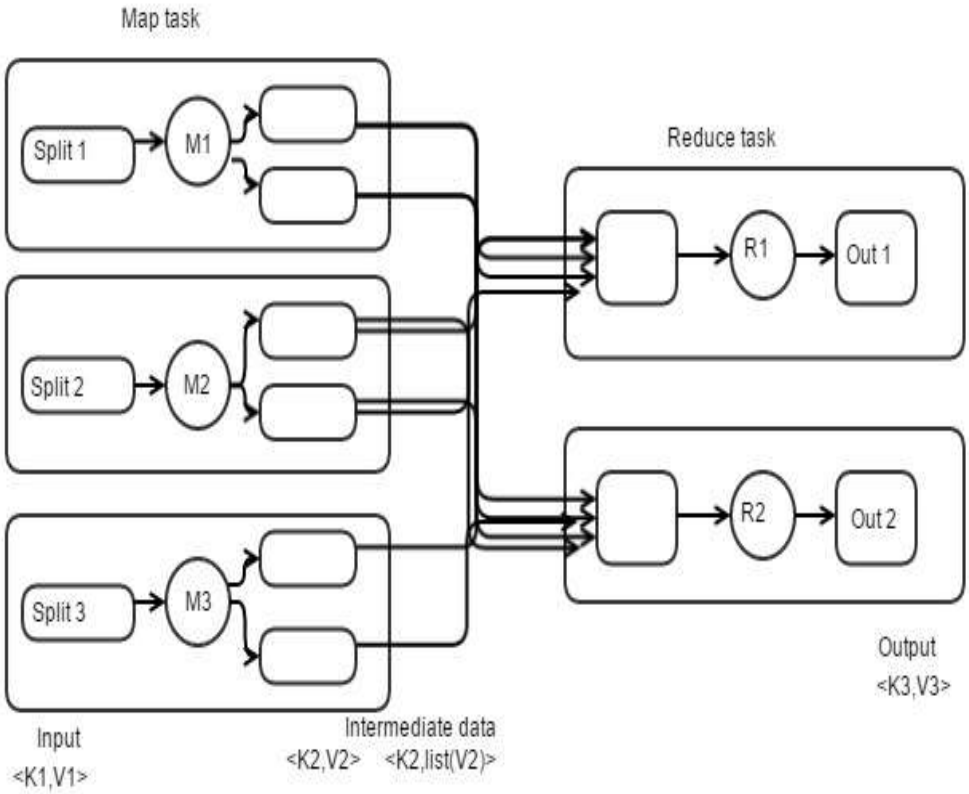

Fig 4: Working of MapReduce

MAP function itself consists of following steps--

1. Read phase- Input split is retrieved from file system and key/value pair are produced.
2. The map Function which is user defined is processed to produce the output.
3. The partitioning and collection of intermediate of the data took place.
4. Data compression phase and data is located to local disk.
5. Data is merged together into single map output file.

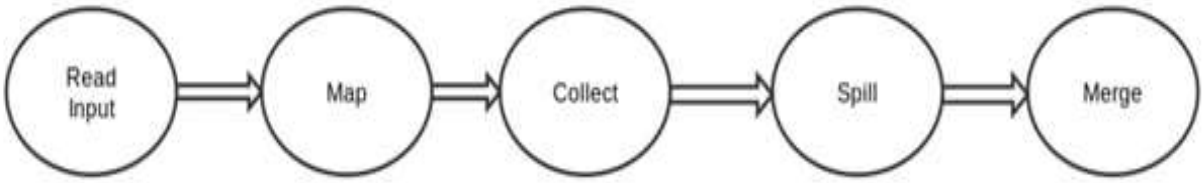

Fig 5: Map Phase

The reduce phase can be viewed as--

1. Shuffling of data occurs when the data from mapper nodes is moved to reducer nodes. Decompression of data took place alongside.
2. Sorting of data is done to combine the redundant data under single key.
3 .In this step, the mapper functions are called to produce final output.
4 .Data compressions may occur on final output and data is stored in file system.

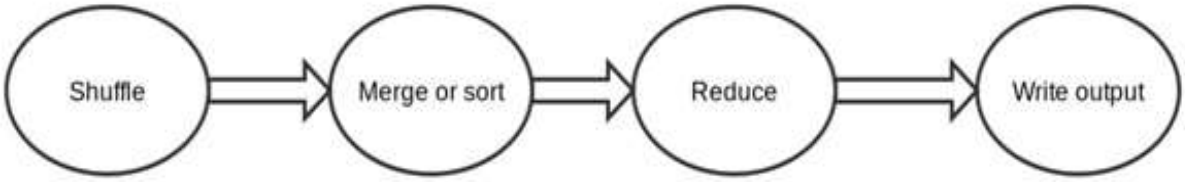

Fig 6: Reduce Phase

Weather data is collected by sensors at every hour at different locations across the universe. A huge amount of data is gathered which is difficult to maintain. MapReduce framework is ideal to handle this massive data as it is semi structured. The sample data is taken from National Climate Data Center. National Climate Data Center is responsible for preserving the data related to the climate and weather received from various sources like radar, satellites etc. The data is stored using a line-oriented ASCII format, in which each line is a record. Some of the piece of data is--

0057
332130 # USAF weather station identifier
99999 # WBAN weather station identifier
19500101 # observation date
0300 # observation time
4
+51317 # latitude (degrees x 1000)
+028783 # longitude (degrees x 1000)
FM-12
+0171 # elevation (meters)
99999
V020
320 # wind direction (degrees)
1 # quality code
00450 # sky ceiling height (meters)
1 # quality code
CN
010000 # visibility distance (meters)
1 # quality code
N9
-0128 # air temperature (degrees Celsius x 10)
1 # quality code
-0139 # dew point temperature (degrees Celsius x 10)
1 # quality code
10268 # atmospheric pressure (hectopascals x 10)
1 # quality code

Here the data is organized in form of date and different locations of whether stations. This record contains directory from year 1901 to 2001.There is a directory for each year from 1901 to 2001, which contained data in form of zipped format. Some of the entries for 1990 are:

% **ls raw/1990 | head**

010010-99999-1990.gz

010014-99999-1990.gz

010015-99999-1990.gz

010016-99999-1990.gz

010017-99999-1990.gz

010030-99999-1990.gz

010040-99999-1990.gz

010080-99999-1990.gz

010100-99999-1990.gz

010150-99999-1990.gz

There are about tens of thousands of the weather stations so the data consists of small correlated files.

The map function is simulated as data preparation as to produce relevant and meaningful data that must be fed into reducer node to find the maximum temperature.

The following input data is considered in order to perform map function-

0067011990999991950051507004...9999999N9+00001+99999999999...

0043011990999991950051512004...9999999N9+00221+99999999999...

0043011990999991950051518004...9999999N9-00111+99999999999...

0043012650999991949032412004...0500001N9+01111+99999999999...

0043012650999991949032418004...0500001N9+00781+99999999999...

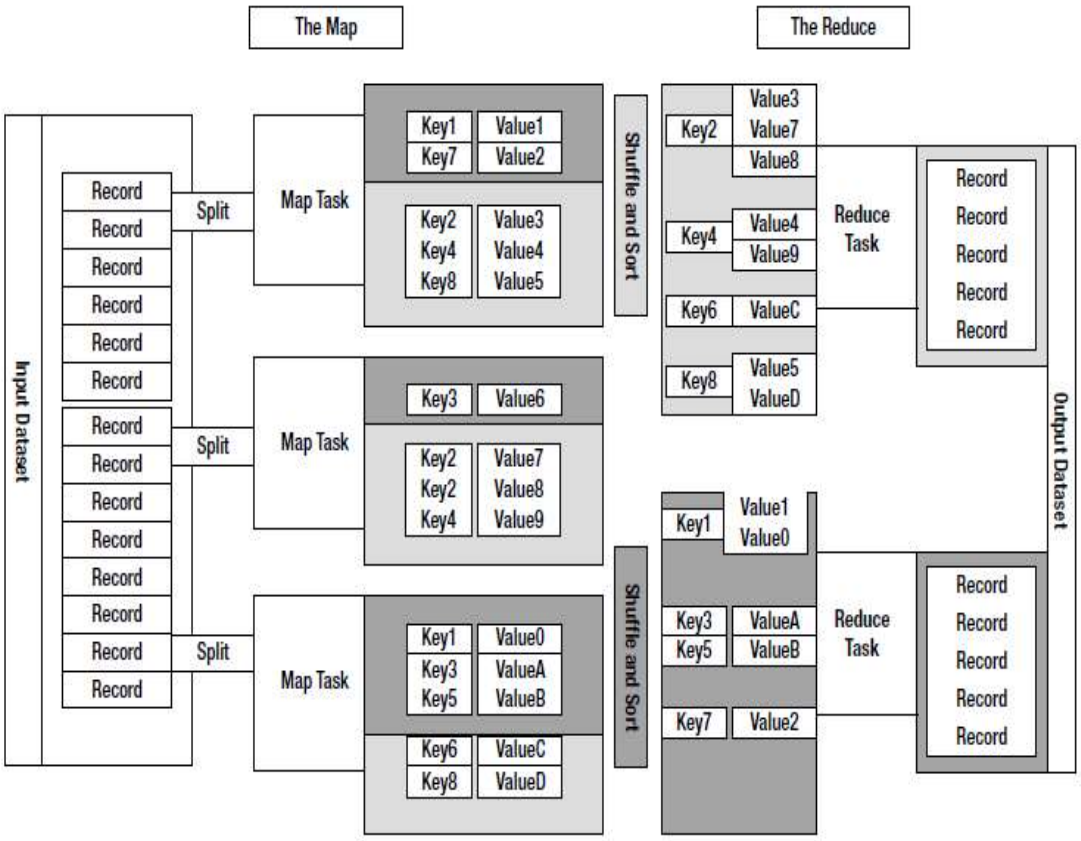

Fig 7: Internal Structure of MapReduce

These lines are presented to the map function as the key-value pairs:

(0, 006701199099999**1950**051507004...9999999N9+**0000**1+99999999999...)

(106, 004301199099999**1950**051512004...9999999N9+**0022**1+99999999999...)

(212, 004301199099999**1950**051518004...9999999N9-**0011**1+99999999999...)

(318, 004301265099999**1949**032412004...0500001N9+**0111**1+99999999999...)

(424, 004301265099999**1949**032418004...0500001N9+**0078**1+99999999999...)

The map function withdraws the year and the air temperature and represents the output--

(1950, 0)

(1950, 22)

(1950, −11)

(1949, 111)

(1949, 78)

The MapReduce framework sorts and groups together the data which have same keys and is as input for reducer (Fig 8).

(1949, [111, 78])

(1950, [0, 22, −11])

Here the year list with their corresponding temperature is generated. The reduce function is now pick up only the highest temperature.

(1950, 22)

This is the final output: the maximum global temperature recorded in each year.

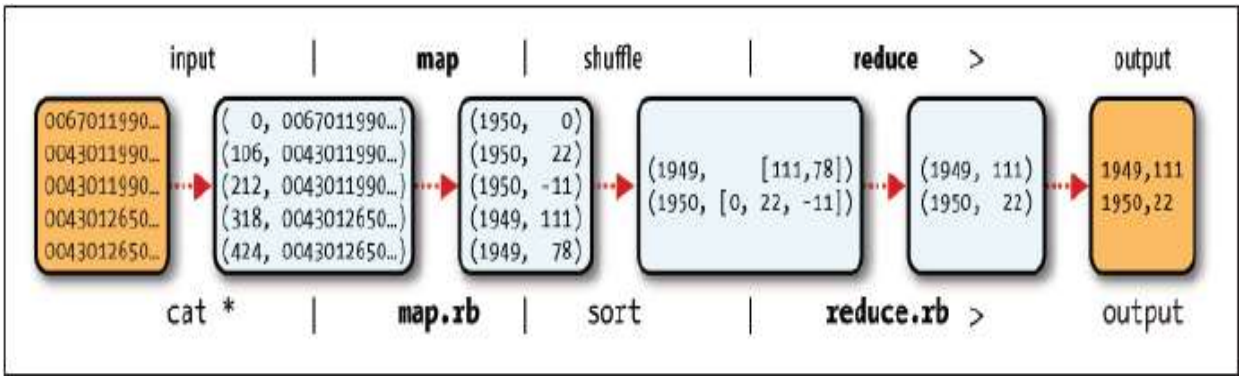

Fig 8: Example of Map Reduce

1.7 Implementation of MapReduce

MapReduce can be implemented by

1. **Hadoop**--Hadoop became apparent as one of the widespread implementations of MapReduce in research section and is a part of Apache Software Foundation. Hadoop is process of using thousands of computer nodes to store big data and operates on that data parallely. The big companies like IBM, adobe, yahoo, Facebook are taking initiative in undertaking projects of Hadoop and also contributing to the technology of Hadoop. Many academics institutions are also contributing to it. This contribution in Hadoop makes it more complete implementation of MapReduce and have potential to compete with Original

Google MapReduce as in April 2009, yahoo published the Terasort benchmark results done with Hadoop which involves 910 nodes and sorted 10 million records(consists of data around 1TB) in 209 seconds. Hadoop can be employed on data centers and on cloud as cloud allows Hadoop organization without any hardware. Hadoop Provides Developers and analysts ability to easy storage, sourcing and handling of data. Logs generated are easy to read.

2. **Infinispan-** Infinispan is scalable, stored data in form of key/value pair and have grid data platform. It is composed in Java, open source and a part of red hat . The main purpose of it is to uncover the data structure which is distributed, concurrent and designed to support the working and architecture of modern multiprocessor. Infinispan holds data to permanent storage and this property is known as cache storage.

3. **Disco**-- It is a light weighted and open source framework based on MapReduce paradigm. Disco built in 2008 by Nokia research center to handle the huge amount of data. The main purpose of the disco is to replicate the data and schedule the jobs effectively and efficiently. Disco is helpful in indexing billions of records and query them in real time.

4. **Riak**--Riak is a distributed database and is an open source. It stores data in key/value pair and is architecture for low latency. Riak have property of replicating the data as data is available in failure conditions. Riak offers high availability, scalability and fault tolerance. Riak MapReduce is meant for batch processing not for real time querying. Riak cluster is formed by grouping of several nodes in communication which ensure continuous availability. In Riak cluster each node has equal responsibilities as no master node is present. Riak is capable in replication of data as one can read/write data if a node is down.

5. **MangoDB**-- MangoDB is based on documented oriented database and NoSQL database. It is open source software. It also delivers high availability with replication. One can make use of it as file system due to its property of load balancing and high availability of data. MangoDB is popular due to its scalability

and is schema less. The indexing property of MangoDB is limits where arithmetic operations and negation operators are used. In the replica set, it handles only 12 nodes.

1.8 Hadoop

Hadoop is open source framework for distributed processing on large data sets on commodity hardware. It is designed to scale from single machine to multiple machines, each machine offers local storage and computation. Hadoop is a framework for storing and processing of data. All the modules of Hadoop are planned with assumption that if any hardware failure occurs, that must be handled by the framework. Hadoop helps business sector in handling enormous amount of structured and unstructured data. Hadoop is originated by Doug cutting in 2005. Hadoop is considered as key part in various web companies like Facebook, yahoo etc. in its computing infrastructure. Many companies use Hadoop in their research and production section. With Hadoop industries can find values in data which was considered as useless. The capability of Hadoop of handling all type of data such as structured data, unstructured data, web logs, emails, pictures, audio etc is remarkable. One of the astonishing features of Hadoop is its faster performance as Hadoop process data parallely using several machines quickly. In yahoo, Hadoop cluster process 1 terabytes of data in 209 seconds for terasort benchmark. Hadoop is considered to be one of the essential tools in industrial sector because of following points-

1. Low cost- Open source, free, uses commodity hardware.
2. Computing power- It works on platform of distributed computing which done work very quickly. It also depends upon the number of machines employed as more the machines, more work done and less time consumed.
3. Scalability- One can easily add nodes to expand the system.
4. Data protection- Here the data is guarded against the hardware failure as if a node fails then the data is send to another node to ensure that failure must not occur.

Architecture of Hadoop

It consists of two main components--

1. HDFS
2. MapReduce

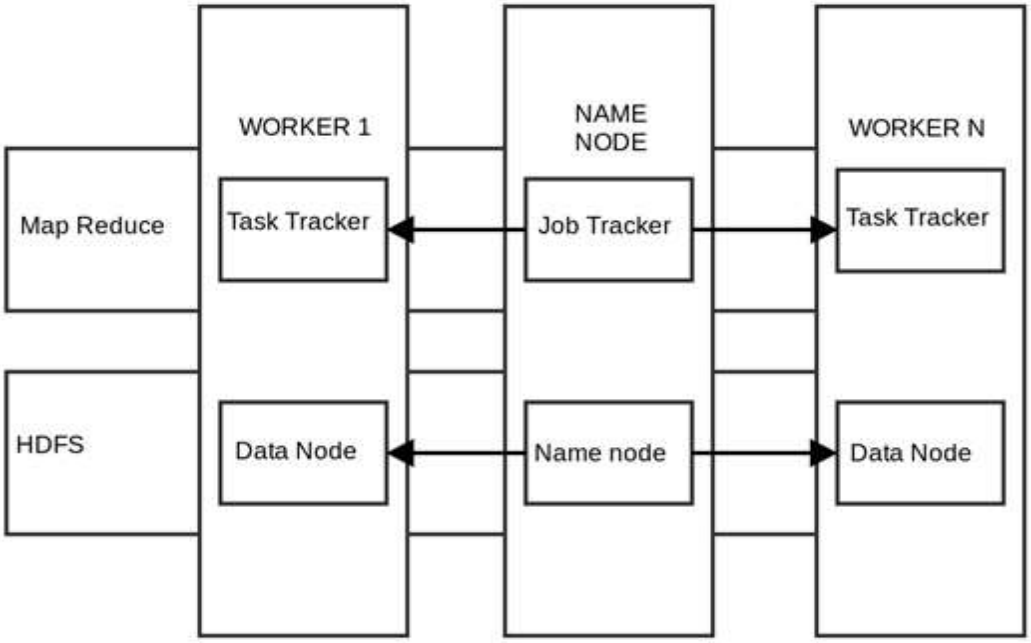

Fig 9: Architecture of Hadoop

HDFS (Hadoop Distributed File System)

HDFS is a Distributed File system used for storing large datasets on commodity hardware. It offers streaming data access and is fault tolerance. It is based on java and includes sequential read and writes. Each file is divided into small chunks and is stored in different data nodes. Some of features are highly available, fault tolerance, highly functional, data replication, write once and read many etc.

Name Node and Data Node

Name Node-- It is considered as the master server responsible for managing the namespace and control access to the files by users. HDFS cluster have only one name node. It keeps track of directory and locating the chunks of data in file. It does not store data itself. It implements file namespace operations such as closing, opening and rename directories.

Data Node-- A functional file system has more than one Data Node, with data replicated across them and HDFS have many data node per cluster (Fig10). It is responsible for handling the data associated with the node. Data nodes serve read and write requests of the users. Data nodes are responsible for block creation and deletion.

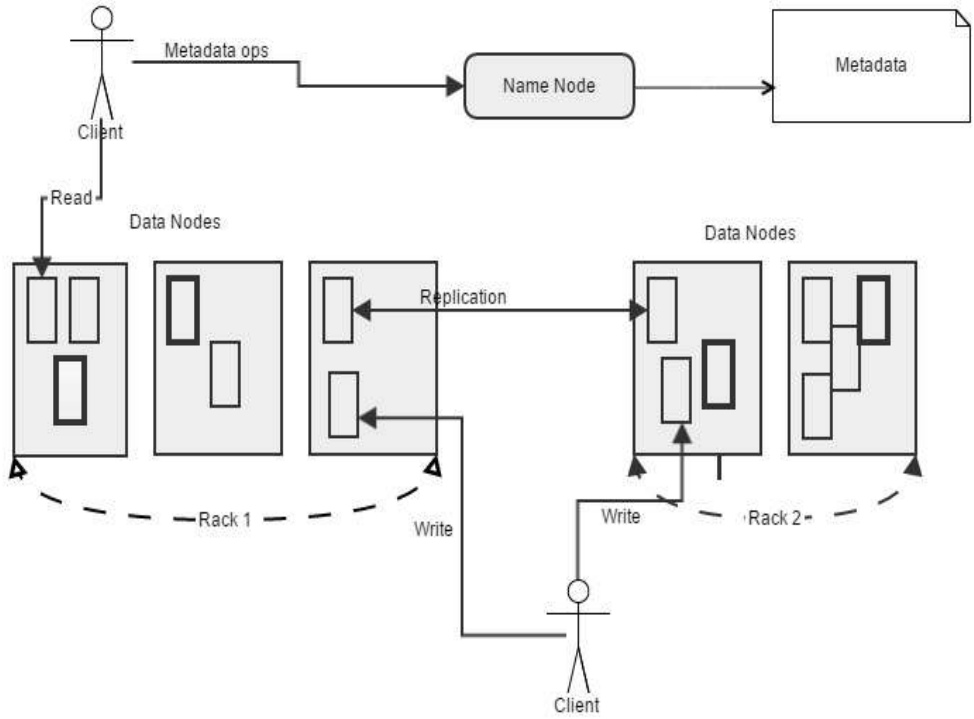

Fig 10: Name Node and Data Node

Name node and data node both are considered as piece of software runs on commodity hardware. As the file system is built by employing java language so any machine has java can run name node and data node. The single name node in HDFS makes its architecture simple. Name node sends instructions to data base when the data node sends heartbeat to Name Node as it does not directly send requests to them.

HDFS Replication process--

HDFS is reliable for storing massive data in large cluster. It stores data as sequence of blocks. To enable fault tolerance feature, replication of data is necessary so the data is replicated for fault tolerance. Each block of data is replicated into multiple nodes and the default value is three. The first copy of block stored into local node and the other two copies are stored into different nodes of remote rack. If there are more copies, then stored into any other rack. Name node is responsible for replication of blocks. In the following diagram block A is replicated in rack1 (node3) and rack3 (node9), similarly block B is replicated in rack1 and rack2.

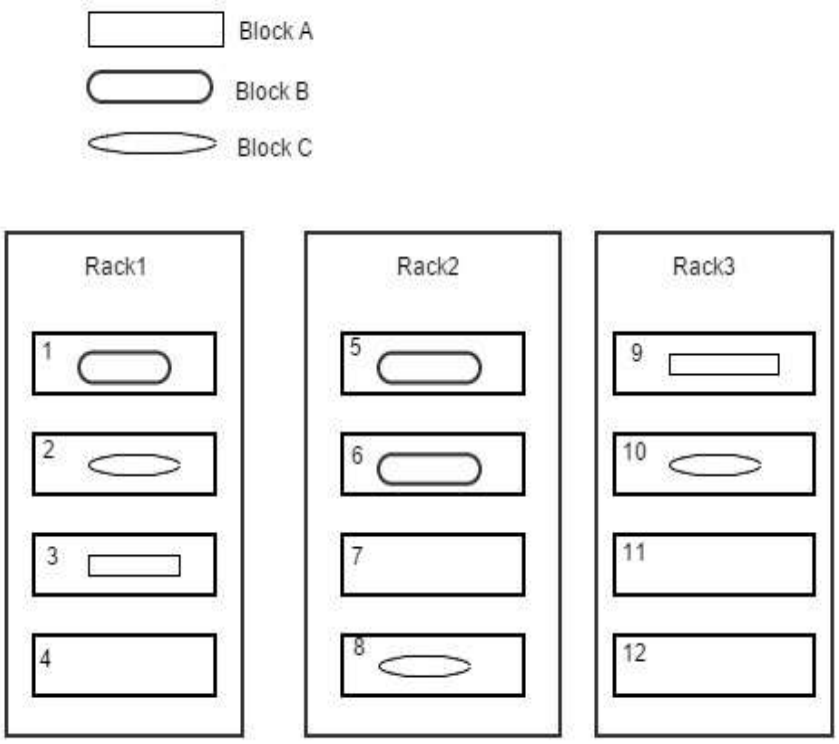

Fig 11: Replication of Data

Job Tracker and Task Tracker

Job tracker is responsible for handling the request from client side and assigning the tasks to the task tracker (Fig 12). The duty of job tracker is to provide task to the task tracker on the same data node or to the nodes present in same rack.

Task tracker is a type of node that handles the tasks- map, reduce and shuffle which is assigned by job tracker. The no of tasks depends upon the no of its slots as they are configured with slots. The task tracker sends heartbeat to job tracker to show its existence. The main job of task tracker is to monitor and start the jobs and then continuously sends the status to the job tracker.

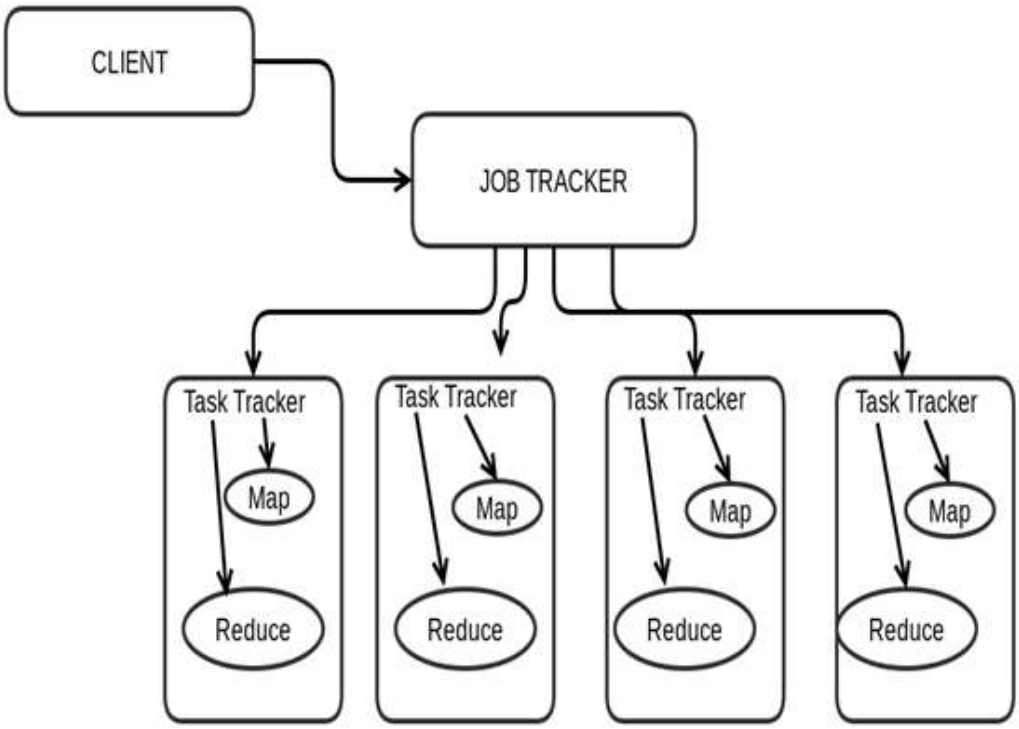

Fig 12: Job Tracker and Task Tracker

1. Client submit its job to the job tracker
2. Name node helps job tracker in locating the data in cluster.
3. Now the job tracker finds the task tracker with free slots.
4. Job tracker assigns that task to selected task tracker.
5. Task tracker continuously sends heartbeat to the job tracker to show its existence, if job tracker does not receives heartbeat from it, then that task tracker considered as fail and its work pass to other task tracker.
6. After the completion of work, job tracker updates its status.
7. Information is passed to the client that work is done.

1.9 Working of MapReduce job

Four entities took part in MapReduce job- Client, Job tracker, task tracker, distributed file system and their working steps are shown in Fig13.

Step1- Job submission- new instance of job client is formed on client node.

Step2- job client gets its job ID from the job tracker

Step3- Job client copies the resourced needed for the job

Step4- It informs the job tracker that job is ready for execution.

Step5- Now job tracker puts job into its queue from where job is initialized by job scheduler

Step6- Job scheduler picks the input split from file system and creates map for each input split. Reduce task are created by mapred.reduce.tasks property.

Step7- Task tracker runs task and send heartbeat to the job tracker. Heartbeat tells the job tracker that task tracker is alive. Task tracker also sends signal to job tracker that it is ready for new task.

Step8- now task tracker run the job and retrieve the job resources from the file system.

Step9and 10-- here new Java virtual machine is created to run each task.

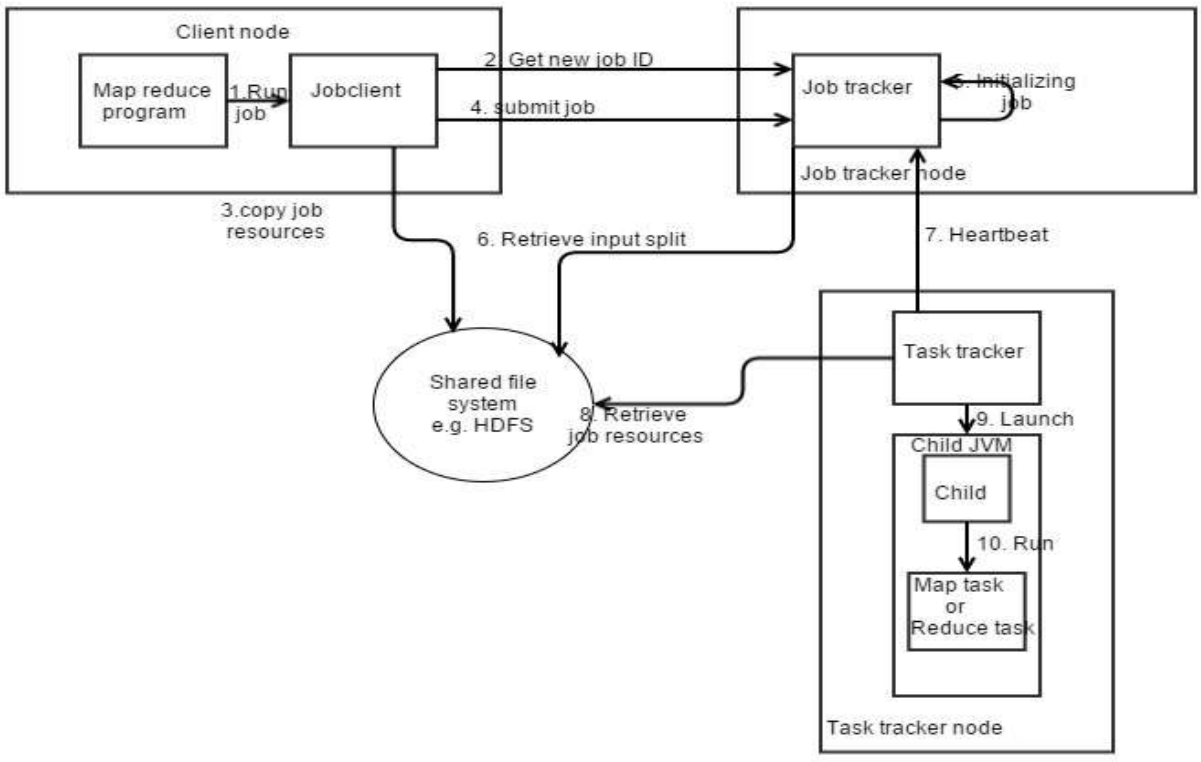

Fig 13- Working of MapReduce

1.10 Pig

Pig is used to analyze the large datasets and its structure allows parallel processing, which in turn handle large datasets easily. Pig is high level platform used for writing MapReduce programs in Hadoop. The pig programs are written in language known as pig Latin, which is a data low language. The key feature of pig is, it provides abstraction layer over the MapReduce programs which are java based, enabling the users to analyze the dataset without having much knowledge of java. So as a result it reduces the time to write map and reduce programs, result is stored in Hadoop File system. Statements of pig Latin follows: Load, Transform, Store which removes the operational burden and complexity.

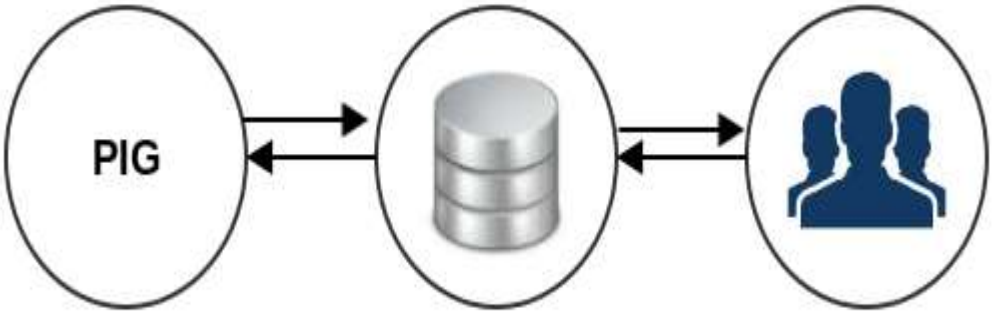

Fig 14- Pig

Advantages of Pig

1. With pig, Pig Latin is used to construct the program instead of Java code as with Pig Latin fewer lines of code is needed, hence minimizing overall development and testing time.
2. Pig script took only 5% of time in writing program as compared to writing in Java.
3. It boosts the productivity for the data analysts and data engineers.
4. Pig allows more control and optimization over the execution of data.
5. Pig is considered as one of the most powerful tool to convert unstructured to structured data.

CHAPTER2

LITERATURE REVIEW

From the earliest days of computing, concept of big data is endemic in field of computer science and data volume increases in range of exabytes and beyond .The amount of information and knowledge that the amount of data is expanding as users come up with new ideas and ways to message and process data. Three fundamental issues or properties related to big data are- data volume, data velocity and data variety. The one of the major issue of big data is its storage for example in field of medical, doctors need past data of patients to understand nature of their disease. Data qualification focuses more on missing values of data or outliers than trying to validate every item and data involved in it is very fine grained such as click stream data. Due to huge volume of data, it is impractical to validate every data item so, new approaches of data qualification and validation are needed. The processing issue of the data required extensively parallel processing. For example-assume the data is divided into blocks of 8 words, so 1 exabyte is equals to 1K petabytes. Stephen Kaisleri and Frank Armour assume a processor expends 100 instructions on one block at 5 gigahertz, the time required for processing would be 20 nanoseconds. To process 1K petabytes would require a total end-to-end processing time of roughly 635 years. One of the problem is to access very large quantities of semi or unstructured data, and to utilize it and problem may neither be solved by dimensional modeling and on-line analytical processing (OLAP), due to its limited functionality and is slow in nature. To make the data computationally tractable, new mechanisms for converting latent, unstructured text, image or audio information into numerical indicators is needed. The different types of data analytics includes descriptive, estimative, predictive, and prescriptive. This paper identified some of the major issues in big data storage, management, and processing.

Map reduce is a platform which processes data in parallel, runs on a large cluster of commodity machines and is scalable. It includes thousands of machines processing terabytes of data. The issues arise is how to parallelize the computation and distribute the data. At Google new abstraction is designed that allows to express the simple computations but hides the messy details of parallelization, fault-tolerance, data

distribution and load balancing. Google employ large clusters of commodity systems attached together with switched Ethernet to implement Map Reduce. The identity of worker machine and its state was stored by master data structure. The map reduce library must tolerate machine failures gracefully. The decisions relates to the debugging in the map reduce is made dynamically by the master but they developed an alternative implementation of the Map Reduce library that sequentially executes all of the work for a MapReduce operation on the local machine. To reduce the computation of map tasks, controls are provided to user. Users invoke their program with a special flag and use any debugging tool according to their convenience and understanding. The locality optimization allows us to read data from local disks, and writing a single copy of the intermediate data to local disk saves network bandwidth. The performance of MapReduce on two computations is examined- One computation searches through approximately one terabyte of data looking for a particular pattern such as shuffling of data. The other computation sorts or extracts the meaningful data which is approximately one terabyte of data. These programs are executed on approximately 1800 machines. They show how sorting took place in normal mode, no backup and when some of the tasks are killed.

Hadoop Distributed File System (HDFS) provides a framework for storing data in a distributed environment and also has set of tools to retrieve and process the data. In this paper, apache pig and hive is used for analytics of big data on the data stored in HDFS. Apache Pig and Hive are two projects which are layered on top of Hadoop, and provide higher-level language to use Hadoop MapReduce library. The standard map reduce programs are written in Java but for rapidly developing MapReduce jobs, Pig is best as it uses pig Latin (which is data flow language)provides the scripting language to describe operations like reading, filtering and transforming, joining, and writing data which are exactly the same operations that MapReduce was originally designed for . Pig does not require any schema and is suitable to process unstructured data. Hive is a technology developed at face book and is similar to SQL queries. HiveQL is a declarative language, relationally complete language and have schema. Both these technologies are used with map reduce for analysis of data.

It is very difficult to handle big data with traditional database management system so to handle data high parallel software is needed. Big data is acquire by the flume as Flume

takes log files as source and after collecting data, it stores the data directly in file system. After acquiring the data, data is organized by GFS or HDFS both have different ways to organize the data. Pig, hive, jaql are used to analyze the data with the map reduce to get fast and efficient results. So as a result lesser time and effort is used in processing big datasets.

Netflow is the network protocol used for assembling network traffic information which is now becoming an industry standard for monitoring the network traffic. Data coming from different routers provide a network-wide view of the traffic and can be made useful in various fields like performance monitoring, capacity planning, accounting and security. This paper comprises of a survey on network traffic analysis using Apache Hadoop Map Reduce framework and pig. The data collected from the network analysis is in big amount and is processed by the Hadoop. One of the problem with map reduce is code complexity and the need arises for a platform which can provide ease of programming along with an enhanced extensibility and Apache Pig is a realistic approach in data analytics. In this, Hadoop is used in analysis of net flow and apache pig is used for programming of map reduce. Pig was tested and proved to be advantageous with a very low computational complexity. Pig can greatly reduce the time consumed for researching on the complex control loops and programming constructs for implementing a typical MapReduce paradigm. So the performance of pig is better in terms of time consumption and code complexity.

Tom White, "Hadoop: The Definitive Guide", 2009. We are in data age as flood of data is coming from many different sources. It is very difficult to measure the data so Hadoop came into existence. Hadoop has its origins in Apache Nutch, which is an open source web search engine and a part of the Lucene project. In April 2008, Hadoop broke a world record to become the fastest system to sort a terabyte of data which include 910-node cluster and sorted one terabyte in 209 seconds .Hadoop is best known for MapReduce and its distributed file system, have some of subprojects as the complementary services. This book helps in understanding the detailed concept of Hadoop and its components and how they work together to achieve a specified goal. It also gives the description about how other components or tools like pig, hive, jaql are involved in the Hadoop processing. It also gives the deep description about the map reduce format and features. This also helps

in understanding how to configure Hadoop.

The on-line Big Data University provides a training courses related to Hadoop. This course helps in basic understanding of the Hadoop, its framework and its associated components. This enables us to understand the step by step execution of Hadoop and its material manages to cover a large spectrum of aspects, both architectural and operational, in dense short lessons. This university conduct on-line test and provide certification.

The book pro Hadoop by Jason Verner, helps in understanding the core of Hadoop and map reduce applications. It explores what is involved in writing the actual code that performs the map and the reduce portions of a Map Reduce job and shows the working with help of suitable examples. It includes the description of HDFS and its applications. It explains the configuring of Hadoop in single node and multi node too. It gives the description of other components related to the Hadoop.

Detection of hardware failures and discovery of the network topology within the Hadoop cluster is done by Manepali V Prabha Satya, Raja Vidya Varsha, Sheeba Samual. They develop an application which collects the information from all data nodes of network and hardware components to analyze them to detect the network, CPU and hard disk failure. It records recent and periodic information from data nodes. The data collected is stored in the Hadoop Distributed File system (HDFS), which is a primary storage system. Pig tool is used for analysis of data, which is platform for analyzing and handling large data sets. Threshold value of failure is used to check failure of data nodes by comparing them and intimates the failed node about the failure so that it can take appropriate actions. The project also discovers the network topology of all the Data Nodes in the Hadoop cluster with the help of a network monitoring tool openNMS and ZenMap.

Impetus Technologies explains tuning of Hadoop configuration parameters which directly affects Map-Reduce job performance under various conditions, to achieve maximum performance. It helps in understanding the parameters and how they affect the performance of Hadoop and these parameters helps in its efficient performance.

Kyong-Ha lee, Hyunsik Choi, Bongki Moon,"Parallel Data Processing with MapReduce", 2011.This enable us to understand the map reduce and its architecture. MapReduce is used in many areas where massive data analysis is required, there are still a big problem lies in its performance, efficiency per node, and simple abstraction, so Kyong-Ha lee, Hyunsik Choi, Bongki Moon explained the pros and cons of MapReduce. In this paper some approaches are mentioned to improve the pitfalls of map reduce framework. This paper tells us that map reduce provides good scalability and fault-tolerance for massive data processing but still there is a problem of efficiency and performance. This efficiency is overcome by improving map reduce and leveraging hardware itself.

CHAPTER3
PRESENT WORK

3.1 Significance

The Internet Archive stores about 2 petabytes data and is growing at a rate of 20 terabytes per month for example Facebook handles around 10 billion of photos, taking up 1 petabytes of storage, Ancestry.com, genealogy site, stores around 2.5 petabytes data. The new business and social science frontier is Big Data, Google's MapReduce is suitable for processing enormous data. MapReduce programming model allows processing and creating large data sets with parallel, distributed algorithm on a cluster. Hadoop is one of the leading adopter of MapReduce. Major challenge in Hadoop MapReduce in the enterprise is lack of performance. The present work tries **to analyze the performance of data manipulation in Hadoop by converting the MapReduce programs to Pig Latin.** The result analysis of this paper shows that Pig can considerably decreases the time consumed for running its script and programming constructs for implementing the typical MapReduce paradigm.

3.2 Objectives

1. To study and analyze the fine granularity of phases within MapReduce in job execution.
2. To compare and identify relevant tool for analyzing data with MapReduce.
3. To configure Hadoop in appropriate manner for identified tool.
4. To increase the performance behavior of the MapReduce framework
5. To analyze the behavior of proposed system.

3.3 Research Methodology

In order to enhance the performance of MapReduce in data accessibility the following methodology is obeyed-

Identification-

In this phase, problem identification is done along with identification of dataset involved in solving the problem. It includes the recognition of working of MapReduce in Hadoop and also involves the identification of appropriate tool out of different tools used in analyzing the performance of Hadoop.

Conceptualization-

In this study, programs are designed to understand study and measure the performance of MapReduce and tool in terms of timing. Performance of word count program is evaluated using Pig tool. The appropriate configuration of Hadoop is set up and the processes involved in Hadoop to run MapReduce is determined. Configuration of pig is done.

Formalization-

It involves the steps used in running the program of word count with the MapReduce and with pig. Different size of data is taken into consideration in analyzing the performance.

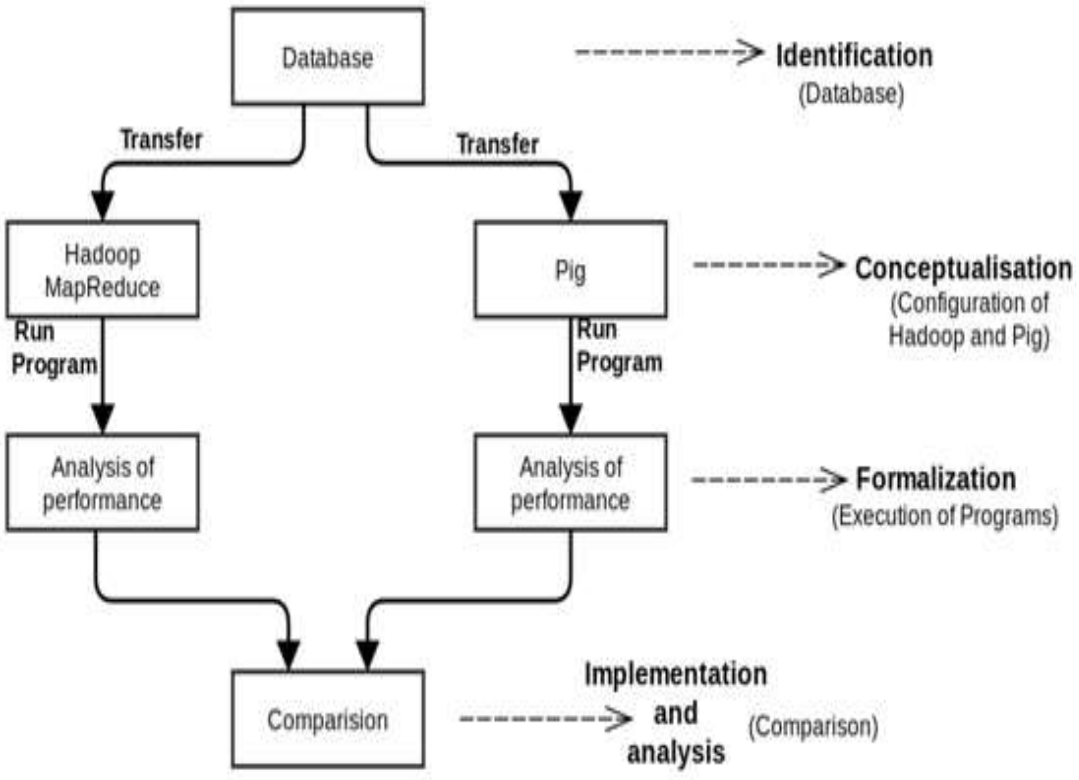

Fig 15: Research Methodology

Implementation-

In this module, the program of word count is run on MapReduce platform and on the pig 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-

1. Dataset 1- US Child Name along with their gender and registered number of their city. The data is from year 1880 to 2103 and random entries are taken into consideration.
2. Dataset 2- Health statistics which include maternal and child health, mental health and mental disorders, tobacco used, vision and hearing, respiratory diseases, nutrition and overweight, injury prevention, etc.
3. Dataset3- School names and its cities.

3.5 Research Design

- Review of literature for defining the phases within the MapReduce.
- Study of different tools for analyzing data with MapReduce.
- Identification of parameters for comparing tools and selecting most appropriate out of them.
- Installation and configuration of tool on Hadoop after configuration of Hadoop.
- Selection of datasets.
- Implementation of proposed system module of enhanced MapReduce in pig
- Comparison of proposed module with existing one for analyzing performance.

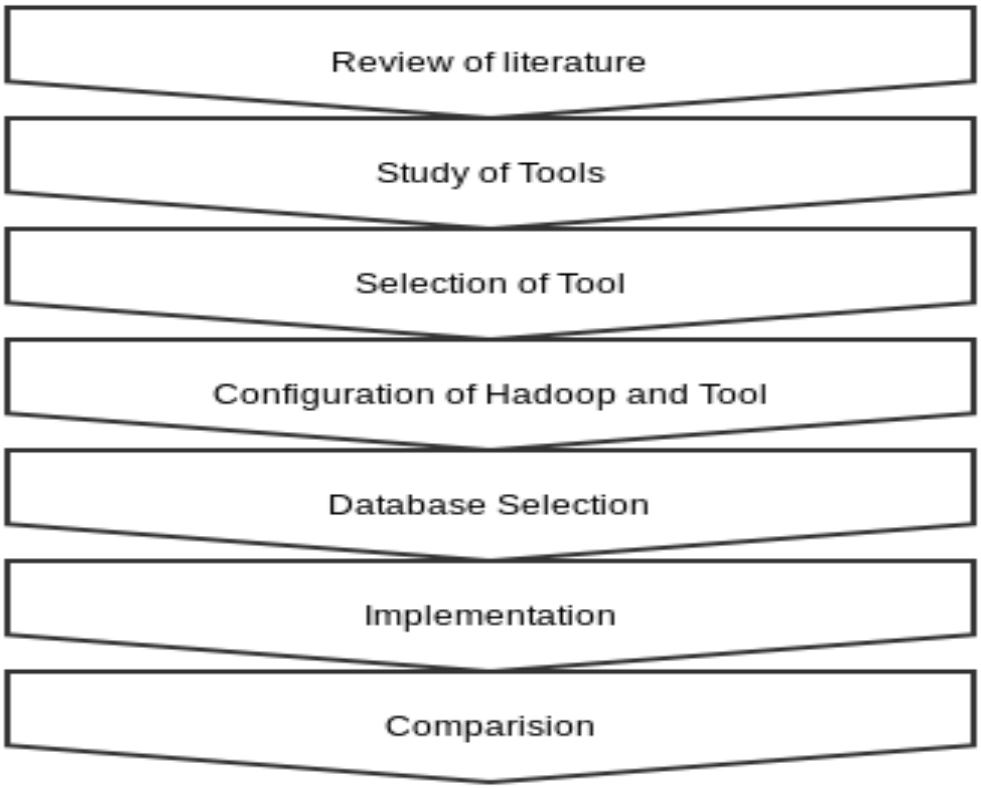

Fig 16: Research Design

3.6 Monthly Progress Report

Table 1: Gantt Chart

Task	Start Date	End Date	Feb 14	Mar 14	Apr 14	May 14	Aug 14	Sep 14	Oct 14	Nov 14	Dec 14	Jan 15	Feb 15	Mar 15	Apr 15
1.Study of big data	1/2/14	20/2/14													
2.Study of Map Reduce	21/2/14	10/3/14													
3.Study of different way of MapReduce implementation	10/3/14	25/3/14													
4.Study of Hadoop	26/3/14	31/4/14													
5.Problem identification and formulization	1/5/14	30/5/14													
6.Confirigutaion of Hadoop	1/8/14	31/9/14													
8.Study and selection of tool	1/10/14	15/11/14													
9.Coding (MapReduce)	16/11/14	31/12/14													
10.Performance Analysis (MapReduce)	1/1/15	15/2/15													
11.Confirugation of tool(PIG)	16/2/15	20/3/15													
12.Perfoamnce Analysis (PIG)	21/3/15	15/4/15													
13.Comparison	16/4/15	30/4/15													

CHAPTER 4

RESULTS AND DISCUSSIONS

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 Pig Configuration

It include following steps-

1. Download the stable release of pig
2. By using export command add pig to selected path in the system.

$export PATH=/<my-path-to-pig>/pig-0.14.0/bin:$PATH

3. Run the following command for testing the installation of pig-

$ pig –help

4.3 Performance comparisons-

Hadoop and Pig are appropriately configured according to requirements. In this work, a sample Hadoop Map Reduce word-count code programmed with java which consist of around 130 lines of code and pig code written in Pig Latin consist of five lines of code. The size of input for both Map reduce and Pig is maintained as a constant.

For making comparison between both, the size of input files is taken as 8.3mb, 1mb and 14.2kb and corresponding execution is plotted in graph.

Dataset 1

Performance of Hadoop

```
15/04/05 01:45:03 INFO mapreduce.Job:  map 100% reduce 100%
15/04/05 01:45:04 INFO mapreduce.Job: Job job_1428177821071_0001 completed successfully
15/04/05 01:45:04 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=5580533
                FILE: Number of bytes written=11319453
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=8254309
                HDFS: Number of bytes written=4354418
                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)=20598
                Total time spent by all reduces in occupied slots (ms)=11607
        Map-Reduce Framework
                Map input records=697288
                Map output records=697281
                Map output bytes=11043318
                Map output materialized bytes=5580533
                Input split bytes=108
                Combine input records=697281
                Combine output records=310519
                Reduce input groups=310519
                Reduce shuffle bytes=5580533
                Reduce input records=310519
                Reduce output records=310519
                Spilled Records=621038
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=680
                CPU time spent (ms)=7890
                Physical memory (bytes) snapshot=309342208
                Virtual memory (bytes) snapshot=968867840
                Total committed heap usage (bytes)=233570304
```

Fig 17a: Hadoop performance for Dataset1

Fig 17b: Hadoop performance for Dataset1

Performance of Pig-

```
onId= - already initialized
2015-04-05 02:39:43,715 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2015-04-05 02:39:43,717 [main] INFO  org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.2.0   0.14.0  hduser  2015-04-05 02:39:26     2015-04-05 02:39:43     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianRe
ducetime        Alias   Feature Outputs
job_local1747269679_0001        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     n/a     A,B,C,D GROUP_BY,COMBINER       f
ile:///home/hduser/output1,

Input(s):
Successfully read 697288 records from: "file:///home/hduser/Hdataset1"

Output(s):
Successfully stored 71431 records in: "file:///home/hduser/output1"

Counters:
Total records written : 71431
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1747269679_0001


2015-04-05 02:39:43,718 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessi
onId= - already initialized
2015-04-05 02:39:43,719 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessi
onId= - already initialized
2015-04-05 02:39:43,720 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with processName=JobTracker, sessi
onId= - already initialized
2015-04-05 02:39:43,728 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2015-04-05 02:39:43,755 [main] INFO  org.apache.pig.Main - Pig script completed in 19 seconds and 921 milliseconds (19921 ms)
hduser@archu-Lenovo-G500:~$
```

Fig 17c: Pig performance for Dataset1

Dataset 2

Performance of Hadoop--

```
15/04/05 01:54:26 INFO mapreduce.Job:  map 100% reduce 100%
15/04/05 01:54:27 INFO mapreduce.Job: Job job_1428177821071_0003 completed successfully
15/04/05 01:54:27 INFO mapreduce.Job: Counters: 43
        File System Counters
                FILE: Number of bytes read=94738
                FILE: Number of bytes written=347863
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1000441
                HDFS: Number of bytes written=65649
                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)=7690
                Total time spent by all reduces in occupied slots (ms)=14334
        Map-Reduce Framework
                Map input records=10685
                Map output records=157107
                Map output bytes=1389760
                Map output materialized bytes=94738
                Input split bytes=108
                Combine input records=157107
                Combine output records=7568
                Reduce input groups=7568
                Reduce shuffle bytes=94738
                Reduce input records=7568
                Reduce output records=7568
                Spilled Records=15136
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=53
                CPU time spent (ms)=3840
                Physical memory (bytes) snapshot=324775936
                Virtual memory (bytes) snapshot=972316672
                Total committed heap usage (bytes)=252182528
```

Fig 18a: Hadoop performance for Dataset 2

Fig 18b: Hadoop performance for Dataset 2

Performance of Pig

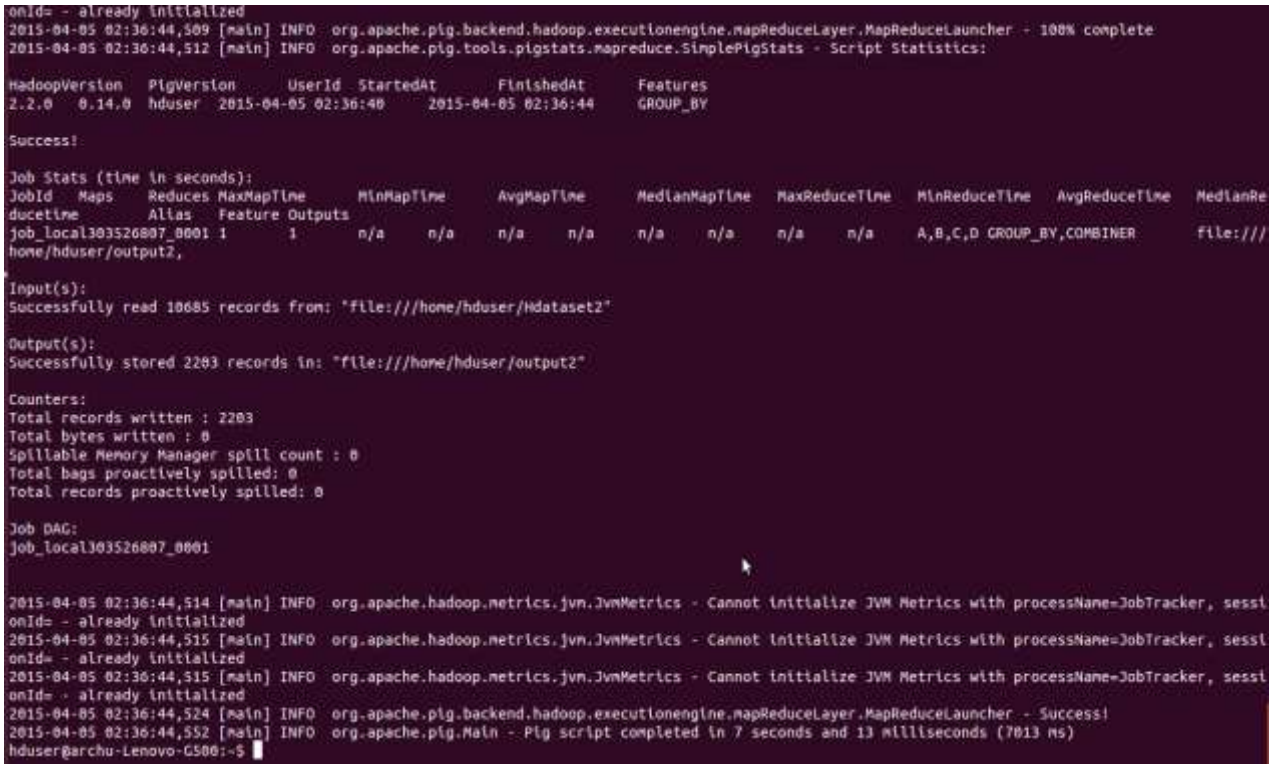

Fig 18c: Hadoop performance for Dataset 2

Dataset 3

Performance of Hadoop—

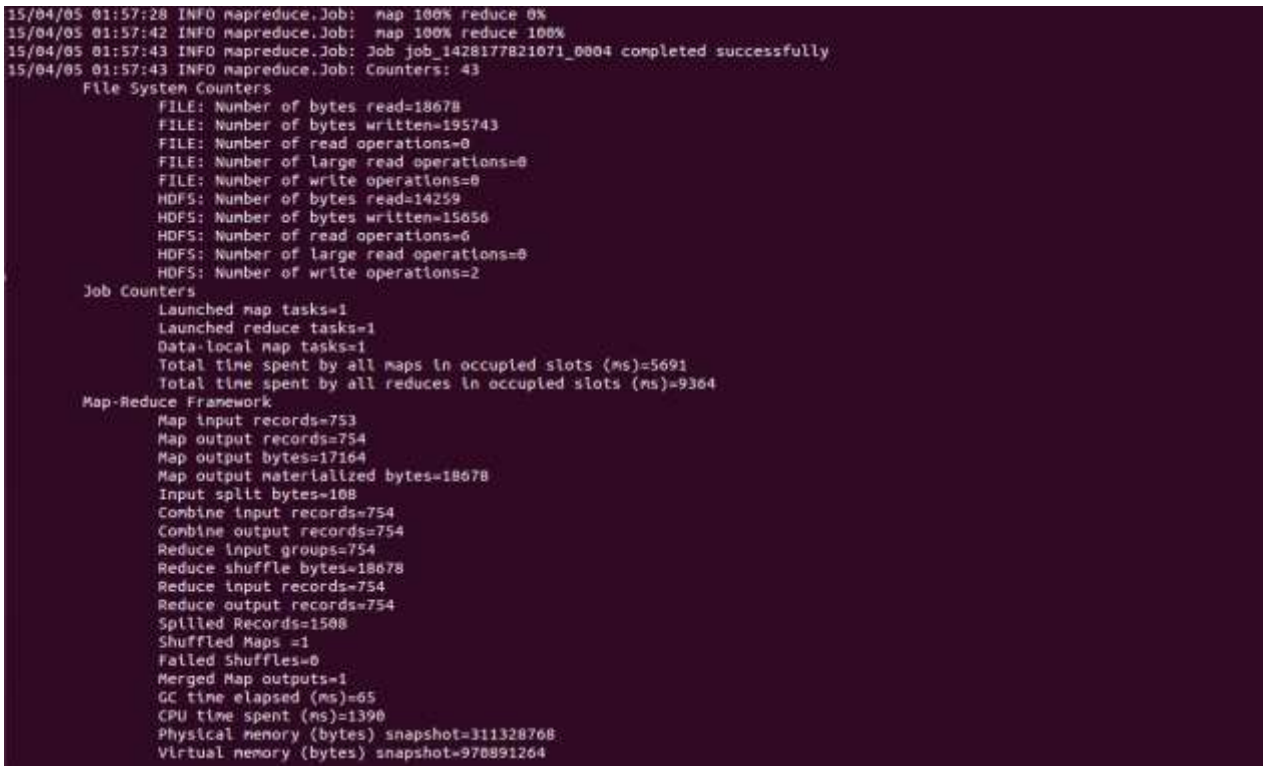

Fig 19a: Hadoop performance for Dataset 3

Fig 19b: Hadoop performance for Dataset3

Performance of pig-

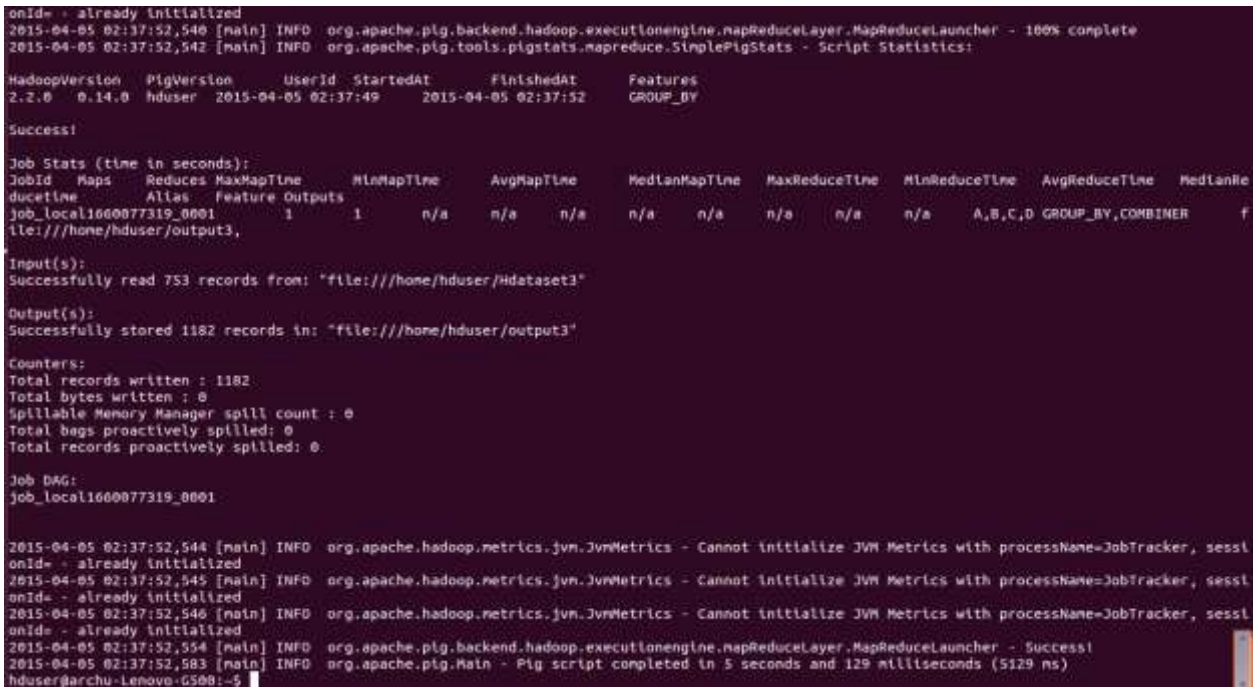

Fig 19c: Hadoop performance for Dataset 3

Comparison graph and Table

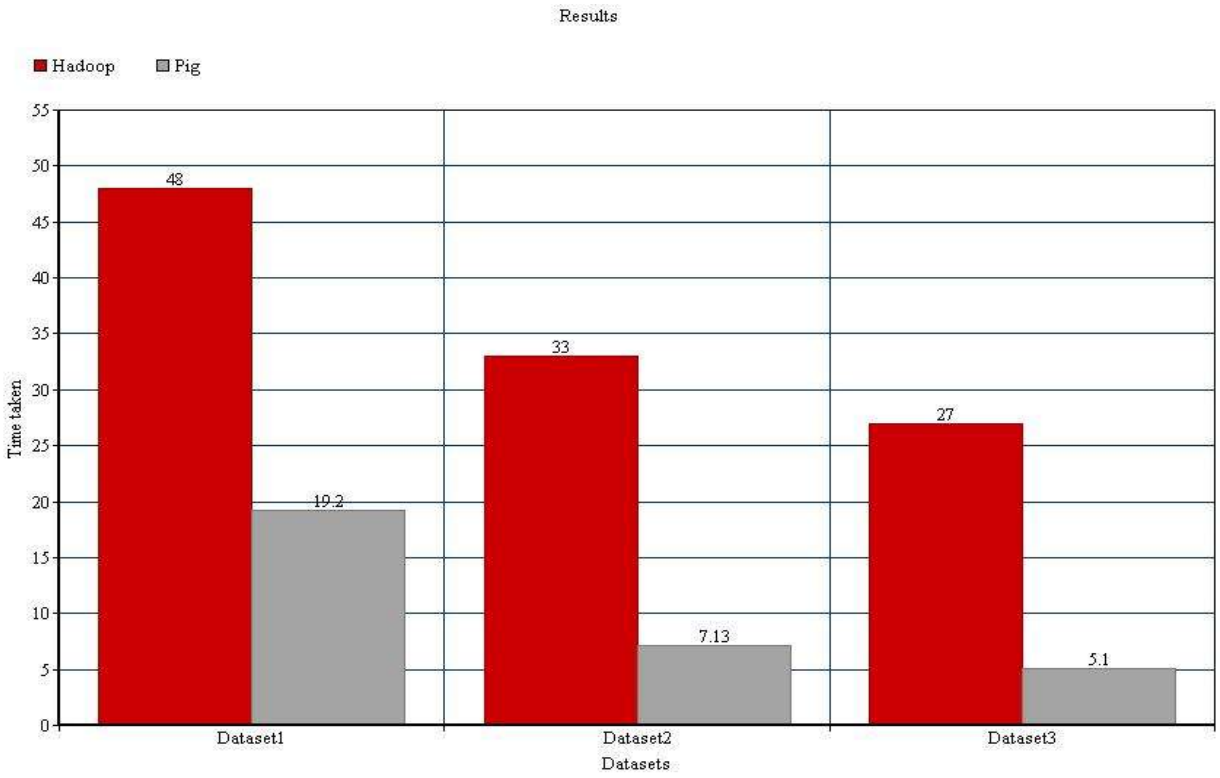

Fig 20: Time taken by MapReduce and Pig

The Time Taken by Map Reduce and pig in execution of Word-Count program is shown in Table 2

Table 2: Time taken by MapReduce and Pig

Name	Operation	Size of file	Time taken by MapReduce	Time Taken by Pig
Dataset1	Word Count	8.3MB	48 sec	19sec 92ms
Dataset2	Word Count	1MB	33 sec	7sec 13ms
Dataset3	Word Count	14.2KB	27 sec	5sec 13ms

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

In order to overcome the major challenge faced by big data community that is performance, so this work tries to convert standard Hadoop MapReduce procedure of word count into Pig Latin scripts. Pig is a platform for analyzing the big data. Pig is considered to have very low computational complexity. From the result analysis of this paper, Pig can greatly decreases the consumed time for running the script of word count and programming constructs involved in implementing a typical MapReduce framework.

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] Anjali P P and Binu A (2014) "A Comparative Survey Based on Processing Network Traffic Using Hadoop Pig and Typical Mapreduce".

[4] Impetus Technologies (2009) "Hadoop Performance Testing".

[5] Jeffrey Dean and Sanjay Ghemawat (2004), "MapReduce: Simplified Data Processing on Large Clusters".

[6] Kyong-Ha lee, Hyunsik Choi, Bongki Moon (2011)"Parallel Data Processing with MapReduce".

[7] Manepali V Prabha Satya, Raja Vidya Varsha, Sheeba Samual(2012) "Hadoop Compatible Framework for Discovering Network Topology and Detecting Hardware Failures".

[8] Sanjeev Dhawan, Sanjay Rathee (2013) "Big Data Analytics using Hadoop Components like Pig and Hive".

[8] Sanjay Rathee (2013) "Big Data and Hadoop Components like Flume, Pig, Hive, Jaql".

[9] Stephen Kaisleri, Frank Armour and J.Alberto Espinosa (2013), "Big Data: Issues and Challenges Moving Forward".

Web pages--

[10] http://airccse.org/journal/ijcses/papers/5114ijcses01

[11] https://docs.jboss.org/author/pages/viewpage.action?pageId=3737165&_sscc=t

[12] http://en.wikipedia.org/wiki/Big_data#cite_note-Editorial-14

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

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

[15] http://iasir.net/AIJRSTEMpapers/AIJRSTEM13-131

[16] http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data

[17] http://spotfire.tibco.com/blog/?p=10941

[18] http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04

[19] http://wikibon.org/wiki/v/Enterprise_Big-data

[20] https://www.cs.rutgers.edu/~pxk/417/notes/content/mapreduce.html

[21]http://www.cse.hcmut.edu.vn/~ttqnguyet/Downloads/SIS/References/Big%20Data/%282%29%20Kaisler2013%20-%20Big%20Data-%20Issues%20and%20Challenges%20Moving%20Forward

[22] http://www.ibm.com/big-data/us/en/

[23] http://www.in.techradar.com/news/world-of-tech/The-importance-of-big-data-analytics-in-business/articleshow/44104837.cms

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

[25] http://www.mongodb.org/about/introduction/

[26] http://www.navint.com/images/Big.Data.pdf

[27] https://www.rgpv.ac.in/iccbdt/papers/44

[28] http://www.sas.com/en_us/insights/big-data/what-is-big-data.html

[29] ftp://ftp.cdc.gov/pub/HealthStatistics/NCHS/Datasets/DATA2010/Standtables/

APPENDIX A

Glossary of Terms **Page no**

APPENDIX B

ABBREVIATIONS

ASCII	American Standard Code for Information interchange
CRM	Customer Relationship Management
ERP	Enterprise Resource Management
GFS	Global File System
GPS	Global Positioning System
HD	High Definition
HDFS	Hadoop Distributive File System
IT	Information Technology
JSON	JavaScript Object Notation
MMS	Multimedia Messaging Service
NCDC	National Climate Data Center
NDFS	New Technology File System
NoSQL	Not Only Structured Query Language
OLAP	On Line Analytical Processing
RAM	Random Access Memory
RFID	Radio Frequency Identification
SMS	Short Message Service
SQL	Structured Query Language
US	United States
USAF	United State Air Force Weather
WBAN	Weather Bureau Air Navy
XML	Extensible Markup Language