Marathwada Shikshan Prasarak Mandal’s

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

**Seminar Report**

**On**

**Apache Hadoop**

Submitted By

**Saurabh Narayanrao Pawar (36062)**

**Dr. BabasahebAmbedkar Technological University**

**Lonere (M.S.)**



Department of Computer Science and Engineering

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

(2019- 2020)

**Seminar Report**

**On**

**Apache Hadoop**

Submitted By

**Saurabh Narayanrao Pawar (36062)**

**In partial fulfillment of**

**Bachelor of Technology**

**(Computer Science & Engineering)**

Guided By

**Prof .Amruta Joshi**

Department of Computer Science & Engineering

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

(2019- 2020)

**CERTIFICATE**

This is to certify that, the Seminar entitled “**Apache Hadoop**” submitted by **Saurabh Narayanrao Pawar** is a bonafide work completed under my supervision and guidance in partial fulfillment for award of Bachelor of Technology (Computer Science and Engineering) Degree of Dr. BabasahebAmbedkar Technological University, Lonere.

Place: Aurangabad

Date:

**Mrs Amruta Joshi Mr. S.B. Kalyankar**

**Guide Head**

**Dr. Ulhas D. Shiurkar**

**Director,**

**Deogiri Institute of Engineering and Management Studies,**

**Aurangabad**

**Abstract**

With the continuous advance in computer technology over the years, the quantity of data being generated is growing exponentially. Some of these data are structured, semi-structured or unstructured. This poses a great challenge when these data are to be analyzed because conventional data processing techniques are not suited to handling such data. This is where Hadoop MapReduce comes in. Hadoop MapReduce is a programming model for developing applications that process large amount of data in parallel across clusters of commodity hardware in a reliable and fault-tolerant manner. This report covers the origin of Hadoop MapReduce, its features and mode of operation, it describes how it is being implemented, as well as reviews how some Information Technology companies are making use of it. It was found out that MapReduce is an efficient model for developing applications that process Big Data.

###### Contents

**List of Figures I**

1. **INTRODUCTION 1** 
   1. Introduction Apache Hadoop 1
   2. What is Hadoop 2
   3. Applications of Apache Hadoop 3

1. **LITERATURE SURVEY 5**

2.1 Review With My Perspective 5

2.2 Others Review 6

1. **BRIF STUDY ON 9**

3.1 Apache Hadoop And Its Architecture 9

3.2 Why is Hadoop important? 11

3.3 How Does Hadoop Work? 11

1. **CONCLUSIONS 13**

4.1 Conclusion 13

4.2 Application 14

4.2.1 Hadoop in Yahoo! 14

4.2.2 Hadoop in Facebook 14

**REFERENCES**

**ACKNOWLEDGEMENT**

**List of Figures**

**Figure Illustration Page**

1.1 Apache Hadoop Ecosystem 2

3.1 HDFS architecture 9

3.2 MapReduce procedure 10

3.3 Cloud 12

1. **Introduction**
   1. **Introduction Apache Hadoop**

Apache Hadoop is an open source software framework used to develop data processing applications which are executed in a distributed computing environment. Applications built using HADOOP are run on large data sets distributed across clusters of commodity computers. Commodity computers are cheap and widely available. These are mainly useful for achieving greater computational power at low cost. Similar to data residing in a local file system of a personal computer system, in Hadoop, data resides in a distributed file system which is called as a **Hadoop Distributed File system**. The processing model is based on **'Data Locality'** concept wherein computational logic is sent to cluster nodes(server) containing data. This computational logic is nothing, but a compiled version of a program written in a high-level language such as Java. Such a program, processes data stored in Hadoop HDFS.

Apache Hadoop consists of two sub-projects –

1. **Hadoop MapReduce:** MapReduce is a computational model and software framework for writing applications which are run on Hadoop. These MapReduce programs are capable of processing enormous data in parallel on large clusters of computation nodes.
2. **HDFS** (**Hadoop Distributed File System**): HDFS takes care of the storage part of Hadoop applications. MapReduce applications consume data from HDFS. HDFS creates multiple replicas of data blocks and distributes them on compute nodes in a cluster. This distribution enables reliable and extremely rapid computations.

In a Hadoop cluster, data is distributed to all the nodes of the cluster as it is being loaded in. The Hadoop Distributed File System (HDFS) will split large data files into chunks which are managed by different nodes in the cluster. In addition to this each chunk is replicated across several machines, so that a single machine failure does not result in any data being unavailable. An active monitoring system then re-replicates the data in response to system failures which can result in partial storage. Even though the file chunks are replicated and distributed across several machines, they form a single namespace, so their contents are universally accessible.

* 1. **What is Hadoop**

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame - worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage. The main reason for its development was advancement of big data. Big data is a term used to describe the voluminous amount of data that would cause too much time and cost to load into a relational database for analysis. In a nutshell, is what Hadoop provides : a reliable shared storage and analysis System. The storage is provided by HDES, and analysis by Map Reduce. The modest cost of commodity hardware makes Hadoop useful for storing and combining data such as transactional , social media , sensor , machine , scientific, click streams, etc. The low - cost storage lets you keep information that is not deemed currently critical but that you might want to analyze later. Because Hadoop was designed to deal with volumes of data in a variety of shapes and forms, it can run analytical algorithms . Big data analytics on Hadoop can help your organization operate more efficiently, uncover new opportunities and derive next-level competitive advantage. The sandbox approach provides an opportunity to innovate with minimal investment. Hadoop is sub-project of Lucene (a collection of industrial-strength search tools), under the umbrella of the Apache Software Foundation.

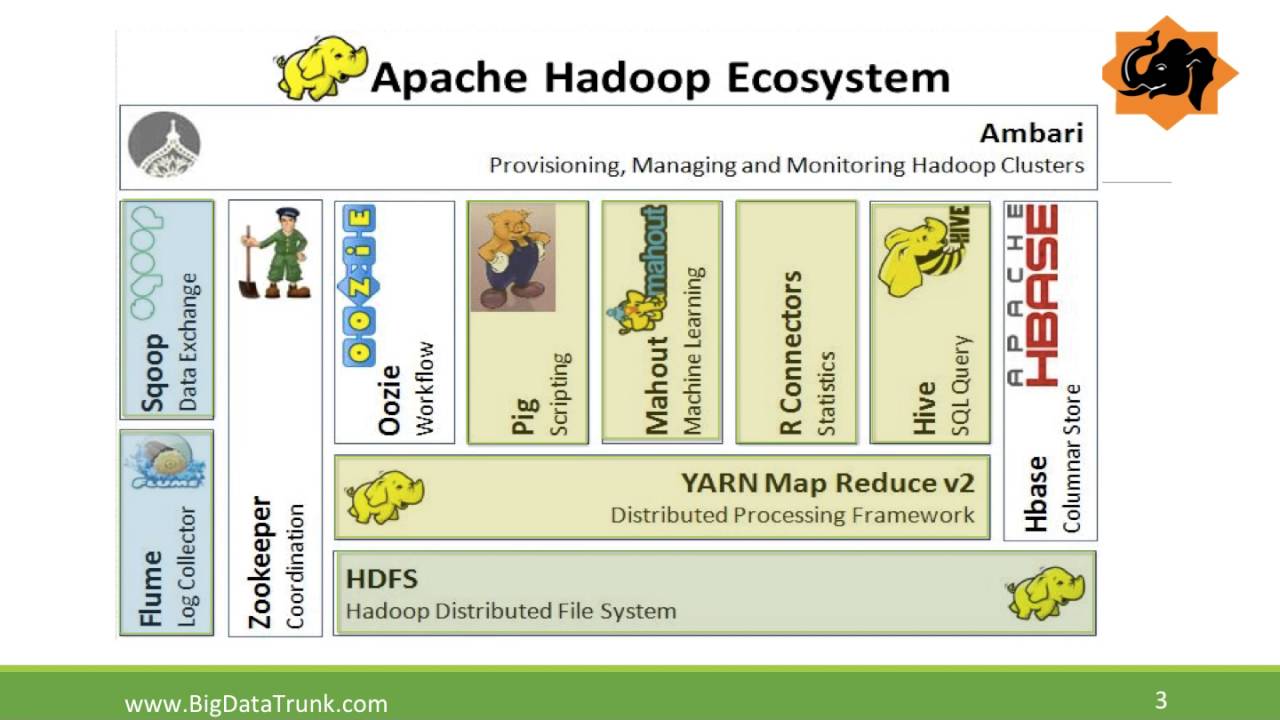


Figure 1.1. Apache Hadoop Ecosystem

2

Hadoop parallelizes data processing across many nodes (computers) in a compute cluster, speeding up large computations and hiding I/O latency through increased concurrency. Hadoop is especially well-suited to large data processing tasks (like searching and indexing) because it can leverage its distributed file system to cheaply and reliably replicate chunks of data to nodes in the cluster, making data available locally on the machine that is processing it. Hadoop is written in Java. Hadoop programs can be written using a small API in Java or Python. Hadoop can also run binaries and shell scripts on nodes in the cluster provided that they conform to a particular convention for string input/output. Hadoop provides to the application programmer the abstraction of map and reduce (which may be familiar to those with functional programming experience). Map and reduce are available in many languages, such as Lisp and Python.

1.3 **Applications of Apache Hadoop**

* Making Hadoop Applications More Widely Accessible

Apache Hadoop, the open source MapReduce framework, has dramatically lowered the cost barriers to processing and analyzing big data. Technical barriers remain, however, since Hadoop applications and technologies are highly complex and still foreign to most developers and data analysts. Talend, the open source integration company, makes the massive computing power of Hadoop truly accessible by making it easy to work with Hadoop applications and to incorporate Hadoop into enterprise data flows.

* A Graphical Abstraction Layer on Top of Hadoop Applications

In keeping with our history as an innovator and leader in open source data integration, Talend is the first provider to offer a pure open source solution to enable big data integration. Talend Open Studio for Big Data, by layering an easy to use graphical development environment on top of powerful Hadoop applications, makes big data management accessible to more companies and more developers than ever before.

With its Eclipse-based graphical workspace, Talend Open Studio for Big Data enables the developer and data scientist to leverage Hadoop loading and processing technologies like HDFS, HBase, Hive, and Pig without having to write Hadoop application code. By simply selecting graphical components from a palette, arranging and configuring them, you can create Hadoop jobs that, for example:

* + Load data into HDFS (Hadoop Distributed File System)
  + Use Hadoop Pig to transform data in HDFS
  + Load data into a Hadoop Hive based data warehouse

3

* + Perform ELT (extract, load, transform) aggregations in Hive
* Hadoop Applications, Seamlessly Integrated

For Hadoop applications to be truly accessible to your organization, they need to be smoothly integrated into your overall data flows. Talend Open Studio for Big Data is the ideal tool for integrating Hadoop applications into your broader data architecture. Talend provides more builtin connector components than any other data integration solution available, with more than 800 connectors that make it easy to read from or write to any major file format, database, or packaged enterprise application.

4

1. **Literature Survey**

**2.1 Review With My Perspective**

Since its inception, the MapReduce framework has been a subject of research and discussion among various experts, professionals and institutions. Many of these researchers have praised the subject. MapReduce is arguably the most successful parallelization framework especially for processing large data sets in datacenters comprising commodity computers (Zhiqiang & Lin, 2010). MapReduce has proven to be a useful abstraction and greatly simplifies large-scale computations (Prasad, 2009). The research community uses MapReduce and Hadoop to solve data-intensive problems in bioinformatics, computational finance, chemistry, and environmental science (Serge, 2013). In praise of Hadoop MapReduce, Andrew McAfee and Erik Brynjolfsson, 2011, stated: “The evidence is clear: Data-driven decisions tend to be better decisions. Leaders will either embrace this fact or be replaced by others who do” (McAfee & Brynjolfsson, 2012).

Despite its numerous benefits, a number of researchers have criticized MapReduce for different reasons. Even if high-level, declarative-style abstractions exist and have been widely adopted, Hadoop MapReduce is still far from offering interactive analysis capabilities (Vasiliki & Vladimir, 2014). The most popular of these criticism is found in David DeWitt and Michael Stonebreakers, 2008. There, MapReduce was described as “a giant step backwards”:

“As both educators and researchers, we are amazed at the hype that the MapReduce proponents have spread about how it represents a paradigm shift in the development of scalable, data-intensive applications. MapReduce may be a good idea for writing certain types of general-purpose computations, but to the database community, it is a giant step backward in the programming paradigm for large-scale data intensive applications”. (DeWitt & Stonebraker, 2008)

David DeWitt and Michael Stonebraker also stated that MapReduce is not novel at all – It represents a specific implementation of well-known techniques developed nearly 25 years ago, it is missing most of the features that are routinely included in current DBMS and it is incompatible with all of the tools DBMS users have come to depend on.

Is MapReduce truly missing most of the features that are routinely included in current database management systems? Another component of Hadoop (HBase) is responsible for

5

having those features included in current DBMS and as such, the burden should not be placed on MapReduce. It is correct to state that MapReduce is not the best solution there can be to solving the problem of Big Data, but indeed it is good enough and has room for improvement.

Hadoop is the popular open source implementation of MapReduce, a powerful tool designed for deep analysis and transformation of very large data sets**.** Hadoop enables you to explore complex data, using custom analyses tailored to your information and questions. Hadoop is the system that allows unstructured data to be distributed across hundreds or thousands of machines forming shared nothing clusters, and the execution of Map/Reduce routines to run on the data in that cluster.  Hadoop has its own filesystem which replicates data to multiple nodes to ensure  if one node holding data goes down, there are at least 2 other nodes from   
Hadoop has its origins in Apache Nutch, an open source web searchengine , itself a part of the Lucene project. Building a web search engine from scratch was an ambitious goal, for not only is the software required to crawl and index websites complex to write, but it is also a challenge to run without a dedicated operations team, since there are so many moving parts. It's expensive too: Mike Cafarella and Doug Cutting estimated a system supporting a 1-billion-page index would cost around half a million dollars in hardware, with a monthly running cost of $30,000

**2.2 Others Review**

* Word Count

Word Count is a simple application of MapReduce object oriented programming paradigm. It evaluates the number of times a key-value pairs which is available in data set. By importing packages such as io.\*, conf.\*, map.\*, util.\* from Hadoop API. The Mapper is executed by a map function, which is processed at a time. Input provided can be either a string or a dataset. It divides input file into splits or tokens that is done with help of “java.util.StingTokenizer” class. Output file is represented in the form of <<key>, value>. A distinct map initiates these key value pairs. An input file containing the following text like: VNR college workshop VNR Java Lab. The Reducer for these maps is executed by the Reduce function. It gathers input splits from the map and later it generates intermediate values. Thereby applying shuffling and sorting operation to overcome redundancy

* Low-cost storage and data archive

6

The modest cost of commodity hardware makes Hadoop useful for storing and combining

data such as transactional, social media, sensor, machine, scientific, click streams, etc. The low-cost storage lets you keep information that is not deemed currently critical but that you might want to analyze later.

* Sandbox for discovery and analysis

Because Hadoop was designed to deal with volumes of data in a variety of shapes and forms, it can run analytical algorithms. Big data analytics on Hadoop can help your organization operate more efficiently, uncover new opportunities and derive next-level competitive advantage. The sandbox approach provides an opportunity to innovate with minimal investment.

* Data lake

Data lakes support storing data in its original or exact format. The goal is to offer a raw or unrefined view of data to data scientists and analysts for discovery and analytics. It helps them ask new or difficult questions without constraints. Data lakes are not a replacement for data warehouses. In fact, how to secure and govern data lakes is a huge topic for IT. They may rely on data federation techniques to create a logical data structures.

* Complement your data warehouse

We're now seeing Hadoop beginning to sit beside data warehouse environments, as well as certain data sets being offloaded from the data warehouse into Hadoop or new types of data going directly to Hadoop. The end goal for every organization is to have a right platform for storing and processing data of different schema, formats, etc. to support different use cases that can be integrated at different levels.

* IoT and Hadoop

Things in the IoT need to know what to communicate and when to act. At the core of the IoT is a streaming, always on torrent of data. Hadoop is often used as the data store for millions or billions of transactions. Massive storage and processing capabilities also allow you to use Hadoop as a sandbox for discovery and definition of patterns to be monitored for prescriptive instruction. You can then continuously improve these instructions, because Hadoop is constantly being updated with new data that doesn’t match previously defined patterns.

7

Applications frequently require more resources than are available on an inexpensive machine. Many organizations find themselves with business processes that no longer fit on a single cost effective computer. A simple but expensive solution has been to buy specialty machines that have a lot of memory and many CPUs. This solution scales as far as what is supported by the fastest machines available, and usually the only limiting factor is the budget. An alternative solution is to build a high-availability cluster. Such a cluster typically attempts to look like a single machine, and typically requires very specialized installation and administration services. Many high-availability clusters are proprietary and expensive.

A more economical solution for acquiring the necessary computational resources is cloud computing. A common pattern is to have bulk data that needs to be transformed, where the processing of each data item is essentially independent of other data items; that is, using a single-instruction multiple-data (SIMD) algorithm. Hadoop provides an open source framework for cloud computing, as well as a distributed file system.

Hadoop supports the MapReduce model, which was introduced by Google as a method of solving a class of peta scale problems with large clusters of inexpensive machines. The model is based on two distinct steps for an application:

* Map: An initial ingestion and transformation step, in which individual input records can be processed in parallel.
* Reduce: An aggregation or summarization step, in which all associated records must be processed together by a single entity.

The core concept of MapReduce in Hadoop is that input may be split into logical chunks, and each chunk may be initially processed independently, by a map task. The results of these individual processing chunks can be physically partitioned into distinct sets, which are then sorted. Each sorted chunk is passed to a reduce task. The reduce task(s) with the map output.

8

1. **BRIF STUDY ON**

**3.1 Apache Hadoop And Its Architecture**

Hadoop framework includes following four modules:

* Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provides filesystem and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.
* Hadoop YARN: This is a framework for job scheduling and cluster resource management.
* Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.
* Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.

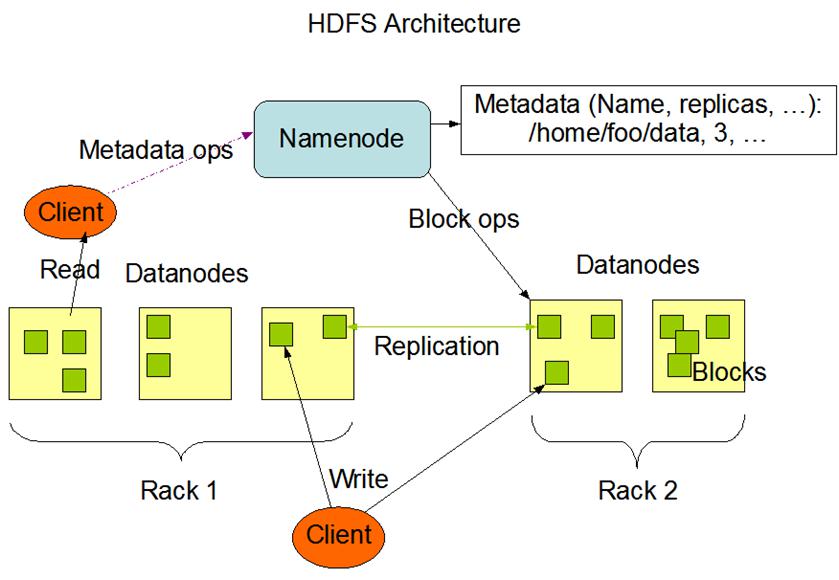


Figure 3.1. HDFS architecture

9

MapReduce

Hadoop MapReduce is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

The term MapReduce actually refers to the following two different tasks that Hadoop programs perform:

* The Map Task: This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).
* The Reduce Task: This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

Typically both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The MapReduce framework consists of a single master Job Tracker and one slave Task Tracker per cluster-node. The master is responsible for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves Task Tracker execute the tasks as directed by the master and provide task-status information to the master periodically.

The Job Tracker is a single point of failure for the Hadoop MapReduce service which means if Job Tracker goes down, all running jobs are halted.

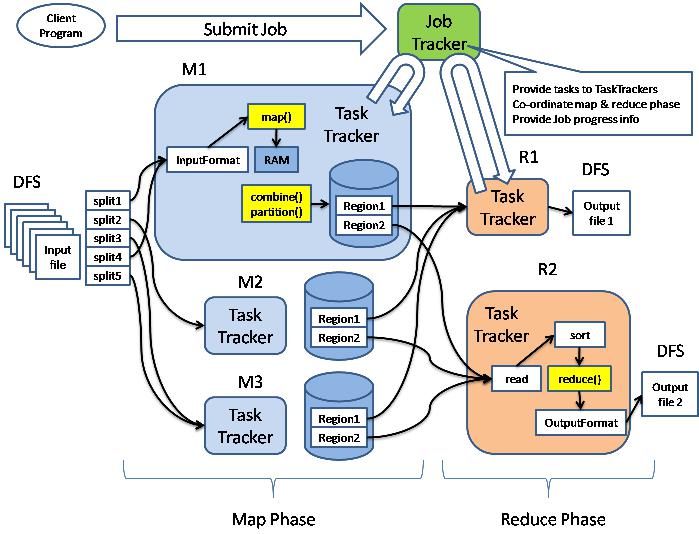


Figure 3.2. MapReduce procedure

10

**3.2 Why is Hadoop important?**

* Ability to store and process huge amounts of any kind of data, quickly. With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), that's a key consideration.
* Computing power. Hadoop's distributed computing model processes big data fast. The more computing nodes you use, the more processing power you have.
* Fault tolerance. Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. Multiple copies of all data are stored automatically.
* Flexibility. Unlike traditional relational databases, you don’t have to preprocess data before storing it. You can store as much data as you want and decide how to use it later. That includes unstructured data like text, images and videos.
* Low cost. The open-source framework is free and uses commodity hardware to store large quantities of data.
* Scalability. You can easily grow your system to handle more data simply by adding nodes. Little administration is required.

**3.3 How Does Hadoop Work?**

Stage 1

A user/application can submit a job to the Hadoop (a hadoop job client) for required process by specifying the following items:

1. The location of the input and output files in the distributed file system.

2. The java classes in the form of jar file containing the implementation of map and reduce functions.

3. The job configuration by setting different parameters specific to the job.

Stage 2

The Hadoop job client then submits the job (jar/executable etc) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

11

Stage 3

The Task Trackers on different nodes execute the task as per MapReduce implementation and output of the reduce function is stored into the output files on the file system.

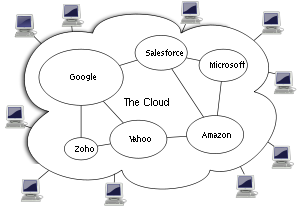


Figure 3.3 Cloud

12

**4 CONCLUSION**

**4.1 CONCLUSION**

The thesis has given a brief introduction to the core the technology of Hadoop but there are still many applications and projects developed on Hadoop. In conclusion, the Hadoop, which is based on the Hadoop HDFS and MapReduce has provided a distributed data processing platform. The high fault tolerance and high scalability allow its users to apply Hadoop on cheap hardware. The MapReduce distributed programming mode allows the users to develop their own applications without the users having to know the bottom layer of the MapReduce. Because of the advantages of Hadoop, the users can easily manage the computer resources and build their own distributed data processing platform.

Above all, it is obvious to notice the convenience that the Hadoop has brought in Big Data processing. It also should be pointed out that since Google published the first paper on the distributed file system till now, the history of Hadoop is only 10-year old. With the advancement of the computer science and the Internet technology, Hadoop has rapidly solved key problems and been widely used in real life. In spite of this, there are still some problems in facing the rapid changes and the ever-increasing demand of analysis. To solve these problems, Internet companies, such as Google also introduced the newer technologies. It is predictable that with the key problems being solved, Big Data processing based on Hadoop will have a wider application prospect.

13

**4.2 APPLICATIONS OF HADOOP**

Nowadays, with the rapid growth of the data volume, the storage and processing of Big Data has become the most pressing needs of the enterprises. Hadoop as the open source distributed computing platform has become a brilliant choice for the business. The users can develop their own distributed applications on Hadoop and processing Big Data even if they do not know the bottom-level details of the system. Due to the high performance of Hadoop, it has been widely used in many companies.

**4.2.1 Hadoop in Yahoo!**

Yahoo! is the leader in Hadoop technology research and applications. It applies Hadoop on various products, which include the data analysis, content optimization, anti-span email system, and advertising optimization. Hadoop has also been fully used in user interests’ prediction, searching ranking, and advertising location.

In the Yahoo! home page personalization, the real-time service system will read the data from the database to the interest mapping through the Apache. Every 5 minutes, the system will rearrange the contents based on Hadoop cluster and update the contents every 7 minutes.

**4.2.2 Hadoop in Facebook**

It is known that Facebook is the largest social network in the world. From 2004 to 2009, Facebook has over 800 million active users. The data created every day is huge. This means that Facebook is facing the problem with big data processing which contains content maintained, photos sharing, comments, and users access histories. These data are not easy to process so Facebook has adopted the Hadoop and Hbase to handle it.

14

**References**

[1] <https://core.ac.uk/download/pdf/38126642.pdf>

[2] <https://www.google.com/search?sxsrf=ACYBGNTCriL3K1-78hjlAtz6zcQ9Ls73wg%3A1571202249563&ei=yaSmXbODIoX39QO5_rOACQ&q=seminar+report+on+apache+hadoop&oq=seminar+report+on+apache+hadoop&gs_l=psy-ab.3..35i39.8688.12703..13702...0.2..0.298.3541.0j18j2......0....1..gws-wiz.......0i71j35i304i39.rB-9G3t61D4&ved=0ahUKEwjzi9K_gKDlAhWFe30KHTn_DJAQ4dUDCAs&uact=5>

[3] <https://intellipaat.com/blog/tutorial/big-data-and-hadoop-tutorial/overview-of-apache-hadoop/>

**ACKNOWLEDGEMENT**

I would like to place on record my deep sense of gratitude to **Prof.S.B. Kalyankar**, HOD-Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for his generous guidance, help and useful suggestions.

I express my sincere gratitude to **Prof.Amruta Joshi**, Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for her stimulating guidance, continuous encouragement and supervision throughout the course of present work.

I am extremely thankful to **Dr.Ulhas D. Shiurkar** Deogiri Institute of Engineering and management Studies Aurangabad, for providing me infrastructural facilities to work in, without which this work would not have been possible.

**Signature of Student**

Saurabh Narayan Pawar

(36062)