Chapter 1

INTRODUCTION

1.1 Overview

The explosive growth of web site over the 'Internet' (WWW) has raised numerous concerns towards organizations to evaluate their clients, and further enhance their advertising methods. As the vast majority of the organizations depend on the WWW to direct business, the traditional techniques and procedures for business examination needs to be revisited.

The information over the web is gathered as access logs that are created by the interaction of users with the web site. The web servers automatically store the interaction to website as server or access logs. The web log document is one line of record for every hit to the website. This contains information about the client's interaction with the website. Different business communities can utilize this data by analyzing for specific purposes. The analysis of the access logs can give them the best way to better the structure of the web site for effective use and benefit of the organization. For promoting organizations, such examination can help them in focusing on a particular customer group.

1.2 Motivation

With the improvement of Internet, e-trade websites now routinely need to work with log datasets which are up to a couple of terabytes in size. How to filter the messy data with minimal effort and discover useful information is a problem we are facing. The mining procedure includes a few stages from pre-processing the raw data to building the final models.

The issues like failure of hubs, communication among the hubs of the system, amble nature of the system are common. A software engineer does not need to worry about these sorts of framework related issues in the distributed system. Each of the developer needs to do is focus on actual requirement of the software. In most of the situation providing dependable system and a fast system cannot be achieved at the same time.

1.3 Why Log Analysis is performed?

The various reasons of log analysis are:

- For troubleshooting the applications
- To know traffic trends
- To find website access figures
- Security lookover

1.4 Hadoop for Log Analysis

As log records are persistently created, storing and producing this much information in a productive way is challenging. A moderate web server will create logs of size at any rate in GB's for a month period. We can't store this lot of information into an RDBMS as it is costly and less expensive options like MYSQL can't scale to the volume of information that is consistently being included.

A better option is to store all the log data in Hadoop Distributed File System which distributed the data across multiple commodity machines, so it will be practical to store terabytes or petabytes of log data. Hadoop also provides MapReduce system to process these log files in parallel.

1.5 Apache Hadoop vs. Traditional System

Hadoop is a modern system for distributed processing that is utilized for querying a substantial set of information and get the outcomes quicker using dependable and adaptable architecture.

In a conventional non distributed model, information put away in one server and any client system access this central data server to get the information. The non-distributed model has couple of major issues. In this model, you will generally scale vertically by including more CPU, including more storage and so forth. This model is also not reliable, as the main server fails, all data will be lost. From execution perspective, this model won't give the outcomes quicker when you are running a query against an enormous information set.

In a Hadoop distributed system, both information and processing are distributed over different servers. The key focuses about Hadoop system are:

- Each and every server offers local processing and storage i.e. when a query is run against an enormous data set, each server in this distributed system will execute the query on its native machine against the nearby data set. At last, the result set from all this nearby servers are consolidated.
- In basic terms, as opposed to running a query on a single machine, the query is run on a bigger dataset are returned quicker.
- You don't need a powerful server. Simply utilize a few less costly ware servers as Hadoop individual nodes.
- High adaption to non-critical failure: If any node fails in the Hadoop system, it still give back the final result properly as Hadoop deals automatically with node failures.
- A basic Hadoop usage can utilize only two servers. Yet, you can scale up to a few large number of servers with no extra effort.
- Hadoop is completely developed in Java. So, it can be installed on any operating system.
- Hadoop is not a substitution for RDBMS. You'll ordinarily utilize Hadoop for unstructured information.
- Originally Google began utilizing the conveyed model in view of GFS (Google File System) and MapReduce. Later Nutch (Open Source web look programming) was changed utilizing MapReduce. Hadoop was branched out of Nutch as a different project. Presently Hadoop is an open source Apache project that has expanded the popularity in recent years.

1.6 Problem Statement

Web log file is a document that is automatically created and it is maintained by the web server. All the user's interaction with the website are logged in this log file, including the view of web document, image, video, audio and other objects. One line of record will be generated for every action by the user with the website. Detailed examination of this web server logs is a big data problem. Based on the website, these logs contain information about customer behavior. For a website handling huge amount of users can generate

terabytes or even petabytes of log files per day. To analyze and extract valuable information from this large amount of data requires more sophisticated methods and techniques.

1.7 Objectives of the Project

The objectives are:

- To covert the semi-structured log data into structured data and storing it efficiently.
- To preprocess the stored access log datasets which includes removal of noisy data.
- Detailed analysis of web server logs to extract valuable information.
- To generate statistical reports using a report designer tool.

1.8 Scope of the Project

The scope is to exhibit

- Processing and analyzing web server access log data with less cost and response time.
- Creating statistical reports indicating the activities of the users.

With minor modifications the procedures and techniques used in this project can be applied to other types of logs such as Wi-Fi logs, firewall logs, e-mail logs etc.

Chapter 2

LITERATURE SURVEY

Various research works are done in web log analysis, Hadoop and some of them are assessed below.

Chen-Hau Wang, Ching-Tsorng Tsai, Chia-Chen Fan, Shyan-Ming Yuan [1] planned and implemented a weblog analysis system based on Hadoop system with HDFS and Hadoop Mapreduce programming system and Pig Latin dialect. The implementation demonstrates that MapReduce system can be successful solution for analyzing web logs in Hadoop environment.

Savitha K,Vijaya M S [2] uncovers the significance of one of the Big data technology Hadoop, where the system handles vast amount of information in a cluster for web log mining. Data cleaning, the main part of preprocessing is performed to remove the conflicting data. The preprocessed data is again manipulated utilizing session recognizable algorithm to discover the client session. Unique identification of fields is done to track the client conduct.

Siddharth Adhikari, Devesh Saraf,Mahesh Revanwar, Nikhil Ankam [3] depicts how a log file is processed utilizing MapReduce system. Hadoop system is used as it is helpful for parallel calculation of log files. Along with that it additionally advises about the brief prologue to pig tool which is used for taking an enormous information from different diverse sources and put it into HDFS for further processing.

Neha Goel, C.K.Jha [4] used Web Log Expert tool for analyzing the user behavior of an astrology website and generated different access reports and the results are incorporated into the website. They also performed a comparison study on different log analysis tools

Naga Lakshmi, Raja Sekhara Rao, Sai Satyanarayana Reddy [5] explained different types of web server log files and how server log files is preprocessed to perform web usage mining. They also explained different fields of the log file in detail.

According to **Harleen Puri [6]** and others web usage mining is the use of information mining systems to find web usage and better serve the needs of web based applications. Web usage mining has three stages, namely preprocessing, pattern discovery and pattern analysis. Using Apriori algorithm the server log records are grouped and the algorithm is executed on the server log files for Association Rule Mining.

Ramesh Rajamanickam and C.Kavitha [7] proposed user path extraction utilizing Euclidean Distance based calculation and demonstrated the proposed calculation can get high effectiveness.

Sayalee Narkhede [8] presented the Hadoop-MR log file examination tool that gives a measurable report on aggregate hits of a web site, client movement, and activity sources. This work was performed in two machines with three instances of Hadoop by distributing the log records evenly to all machines.

Parallelization of Genetic Algorithm (PGA) was recommended by **Kanchan Sharadchandra Rahate [9]**. PGA utilizes OlexGA bundle for ordering the document. The train model information is put in HDFS and the test model classifications the content model.

Milind Bhandare [10] put forth a generic log analyzer structure for various types of log files for example a database or file system. The work was executed as a distributed query handling to minimize the response time for the clients which can be extendable for other format of logs.

Wichian Premchaiswadi, Walisa Romsaiyud [11] examined the clients' behavior of Siam University's websites. They took one month, they gathered, separated, and reported total information about which pages clients visit, what request they make. They connected Hadoop MapReduce structure for enhancing the performance of response time as fast as possible. From the results they realized that registration page and web mail page are the most well-known pages amongst the students in Siam University.

Ravindra Gupta, Prateek Gupta [12] proposed an effective iterative FP Tree calculation for producing frequent access patterns of the web clients. The patterns are generated by

backward tree traversal. The operation will take less time to calculate the access pattern. They also showed a customed log preprocessing method.

M.Venkata Krishna, L.Raghavendra Raju [13] analyzed the web user profiles and designed a system that track the user profile. They preprocessed the data and identified distinct users and requests. Then clustering is performed to group similar requests and using this features new request are generated and incorporated in the web site.

Rahu Mishra, Abha Choubey [14] demonstrated that FP-growth algorithm can be utilized for discovering most frequest path accesed by the web user. Their test result showed that FP-growth system is proficient and adaptable for mining both long and short frequent access patterns.

L.K.Joshila Grace and others [15] discussed about various log files, log formats, log creation and procedures and algorithms for analysis of log files. They also gave an idea of generating extended log file format and learning behavior of users.

C.P.Sumathi, R.Padmaja Valli, T.Santhanam [16] demonstrated web log data preprocessing to remove messy data. Then identification of sessions and users are performed. Created user session files that can be helpful for different data mining tasks.

Murat Ali, Ismail Hakki Toroslu [17] proposed the smart miner system that extracts the user conduct from web logs. The system utilized the smart session construction to trace the frequent client access ways.

Bina Kotiyal, Ankit Kumar, Bhaskar Pant, RH Goudar [18] compared between RDBMS and Hadoop System. A trial work is demonstrated through Hadoop cluster for extracting significant information from the log files. Discussed about processing log data using mapreduce in java in less time.

Sayalee Narkhede, Trupti Baraskar [19] presented best fit Hadoop MapReduce model for studying web application log documents in cloud computing environment. In this process, data storage is done using HDFS and MapReduce model applied over log records

gives analyzed results in negligible response time. To get sorted consequences of investigation pig quires are composed over MapReduce result.

The semi structured log documents are huge datasets which are challenging to store, search, share, visualize and investigate. About 26% of web log types of information require big data knowledge to perform an analysis **[20].**

To enhance the use of a site and to track the client conduct, in web promoting and E-business the web log mining is performed utilizing Hadoop. The related works so far expressed above performs the work with great adaptability however neglects to test the time effectiveness between the distinctive methods of Hadoop and need of the versatility. The proposed work investigations the working of pseudo distributed Hadoop modes and the time productivity for semi structure log information, along which a factual report is made.

Chapter 3

SYSTEM REQUIREMENTS

3.1 Software Requirements

1. Operating System: Cent OS 6.6
2. Framework: Apache Hadoop 1.2.1
3. Data warehouse Tool: Apache Hive 0.13.1
4. Programming Language: HiveQL
5. Reporting Tool: JasperSoft's iReport 5.6.0

3.2 Hardware Requirements

1. Processor: Intel Core i3
2. Speed: 2.20 GHz
3. Hard Disk: 50GB
4. Monitor: 15 VGA Color
5. Mouse: Logitech
6. RAM: 4GB

Chapter 4

SYSTEM ANALYSIS

4.1 Existing System

Conventionally, essential objective was to increase processing power of one machine. At that time developed distributed system which allowed a single job running on multiple machines. From numerous years, High Performance Computing and Grid Computing are processing information utilizing Storage Area Network. The primary drawback of using Storage Area Network is single point of failure. Also at processing time, it copies information to compute node which is appropriate for small data. For huge information, moving information to processing node is not a worthy idea. Also traditional system tools works best in the presence of predefined schema. It require more time to move the data into system. Quires over this large data set will take hours and RDBMS systems are costly and less expensive systems like MySQL cannot scale to the volume of information that is continuously being added.

4.2 Proposed System

Hadoop framework gives data locality facility of moving processing to data rather moving data to processing which makes access to data quick. Hadoop also offers cost effective storing as it runs over the commodity machines. Hadoop can deal with terabytes or petabytes of information. Hadoop MapReduce is suitable for unstructured information, for example, content document and additionally for semi organized information on the grounds that it translate the information at processing time.

The proposed system has two stages

1. Log preprocessing
2. Log analysis

Preprocessing involves removal of noisy data and in the analysis phase using map reduce algorithm and execution of hive query to categorize the analyzed result. Fig. 4.1. Shows the flow chart of proposed methodology.

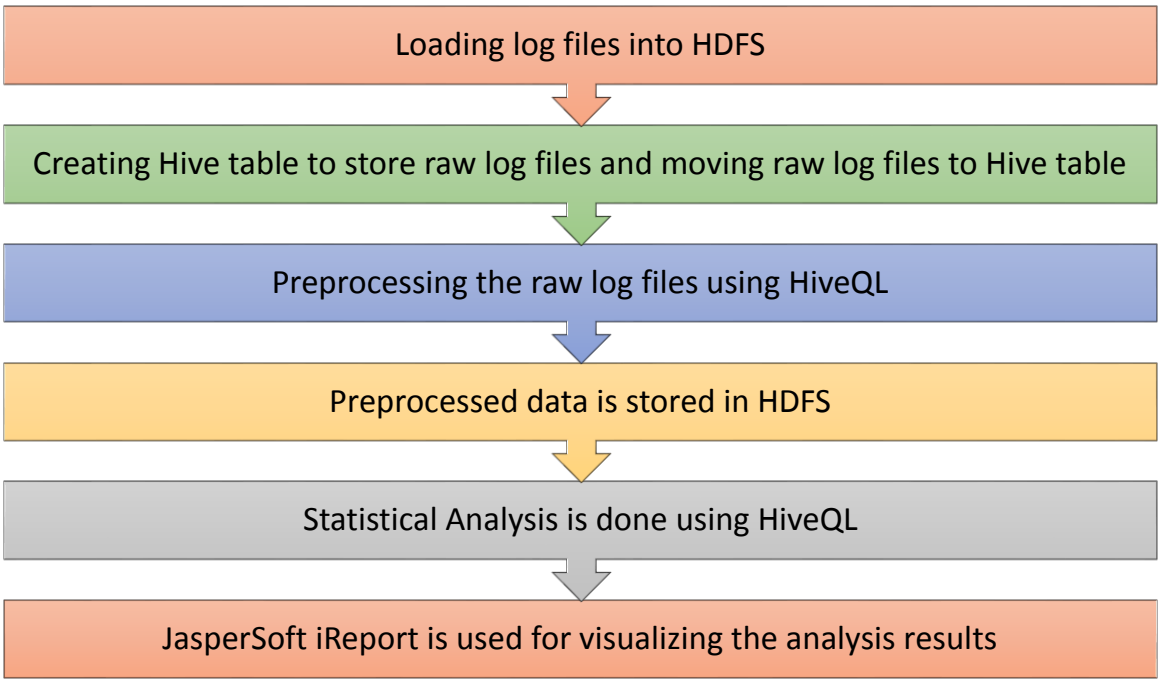

Fig 4.1 Flow chart describing the proposed methodology

Log files usually generated from the web server consist of large volume of data that cannot be handled by a traditional database or other programming languages for computation. The proposed work aims on preprocessing the log file using Hadoop as shown in Fig.4.1.The work is divided into phases, where the storage and processing is made in HDFS.

Web server log files are copied to Hadoop file system. The log file that resides in HDFS is loaded in to Hive table. Then data cleaning is done using Hive query Language. Data cleaning is the first phase carried out in the proposed work as a pre-processing step in web server log files. The web server log files contains a number of records that corresponds to automatic requests originated by web robots, that includes a large amount of erroneous, misleading, and incomplete information. In the proposed work the web log file containing request from robots, spider and web crawlers are removed. Request created by web robots are not considered as used data, it is filtered out from the log data.

In the preprocessing step the entries that have status of “error” or “failure” have been removed. Also some access records generated by automatic search engine agent is identified and removed from the access log. The important task carried out in data cleaning is the identification of status code. Only the log lines holding the status code value of “200”

is identified as correct log. So only the lines having value “200” in status code field are extracted and stored in a Hive table for further analysis.

Than the identification of unique user, unique fields of date, URL referred, and status code are identified. These unique values is retrieved and used for further analysis in order to find the total URL referred on a particular date or the maximum status code got successes on specific date. Finally the analysis result is created in the form of graphs and tables using JasperSoft iReport Designer tool.

Chapter 5

SYSTEM DESIGN

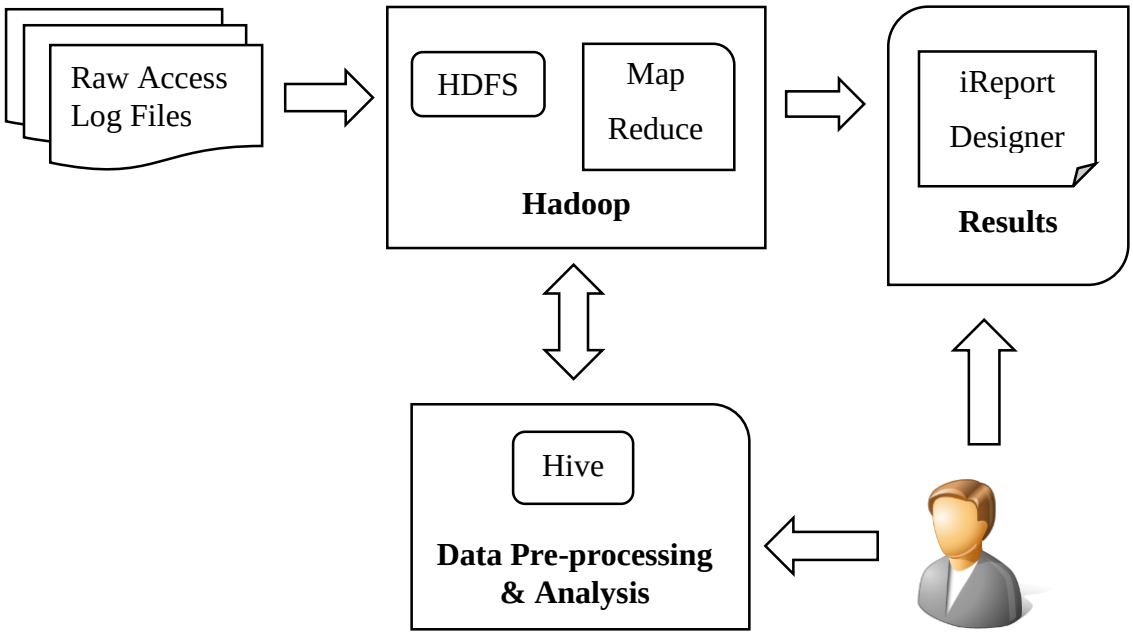

Fig. 5.1 System Architecture

The system architecture is shown in Fig. 5.1. Raw web log files are uploaded into HDFS specified directories through Hadoop command. Then Hive table is created to store the log files and log files are loaded into the hive table created. Then preprocessing task is applied to log files and cleaned log files are stored in HDFS. Finally different statistical reports are created using report designer tool.

5.1 Weblog Data

5.1.1 Types of Web Server Log files

1. **Access Logs**: All the requests made to the server are recorded in the access log file.

2. **Error Logs**: The failed request and the reason for failure are recorded in error log file.

The log files used in this experiment are Apache access log format. The most widely used log format is 'Common Log Format'. One line of Common Log format looks as shown in Fig. 5.2.

```
117.204.8.16 - - [31/Mar/2014:06:09:28 +0000] "GET /deptmechanical.php HTTP/1.1" 200 28647
```

Fig. 5.2 Common Log Format

Fig.5.2 indicates a remote host having IP address 117.204.8.16 requested for the page deptmechanical.php on 31st March 2014 at 06:09:28. The request was successful and 28647 bytes of data is transferred to the client browser.

In this research 'Combined Log Format' is used, a sample of this log file is shown in Fig. 5.3 it contains two more field referrer and user agent in addition to common log format fields.

```
117.204.8.16 - - [31/Mar/2014:06:09:22 +0000] "GET /depte&e.php HTTP/1.1" 200 27058 "http://www.ubdtce.org/" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.154 Safari/537.36"
117.204.8.16 - - [31/Mar/2014:06:09:25 +0000] "GET /staff/upload/ HTTP/1.1" 200 370 "http://www.ubdtce.org/depte&e.php" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.154 Safari/537.36"
117.204.8.16 - - [31/Mar/2014:06:09:28 +0000] "GET /deptmechanical.php HTTP/1.1" 200 28647 "http://www.ubdtce.org/depte&e.php" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.154 Safari/537.36"
117.204.8.16 - - [31/Mar/2014:06:09:30 +0000] "GET /staff/upload/ HTTP/1.1" 200 370 "http://www.ubdtce.org/deptmechanical.php" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.154 Safari/537.36"
117.204.8.16 - - [31/Mar/2014:06:09:30 +0000] "GET /images/mech3.jpg HTTP/1.1" 200 85522 "http://www.ubdtce.org/deptmechanical.php" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.154 Safari/537.36"
```

Fig. 5.3 Sample Raw Access Log File

Each line consists of host IP, time stamp, requested page, status code, bytes transferred, referrer and user agent. The fields machine identity and userid of the client information is not available. Therefore it is indicated as '-' in the log file.

The different fields of Combined Log Format is explained in the below table.

Table 5.1 Fields of combined log format

Fields	Example	Description
Host	117.204.8.16	This is the Internet Protocol address of the machine that made the HTTP request.
Identity	-	"-" means information is not available. (Identity of the host making request)
User	-	"-" means information is not available. (login name of the User)
Time Stamp	[31/Mar/2014:06:09:22 +0000]	Date & Time stamp of the request.
Request	GET /depte&e.php HTTP/1.1	This is the request method, web page to which request has been made and HTTP protocol used.
Status	200	Status code representing that the file sent is success or failure.
Bytes transferred	27058	Number of bytes transferred to the client browser.
Referrer	http://www.ubdtce.org/	URL that linked the user to this site.
User Agent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.154 Safari/537.36	This denotes the web browser and platform used by the client machine.

The different parts of combined log record is explained below in brief:

- **Host:** This is the IP address of the client machine that made the request. This IP address is not always a client machine's address, it may be a proxy server that exists between the server and the client machine.
- **Identity:** This information may not be available for some websites. This is the RFC 1413 identity of the client. This information is never expected. So a hyphen '-' is present in this field.
- **User:** This information is also sometime not available. This is the userid of the user requesting the web page. If the page requested by the client is not protected by a password, user information will not be present. If the information is not available then a hyphen is present in this field.
- **Time Stamp:** Date and time the request made to the web page.
- **Request:** This contains the information of the page to which request has been made. It also contains the HTTP method used by the client and the HTTP protocol used by the user to make the request.
- **Status code:** This is the code that server sends to the client machine indicating that the request is "success" or "failure". Some of the status code and their meaning is show in the table.

Table 5.2 Status Code and their Meaning

Status Code	Meaning
100	Server received the request
200	Request Successful
300	User should take additional action
400	Server not able to process the request
401	Requested page requires authentication
404	Request page is not found in the server
408	Server time-out

- **Bytes Transferred:** This information indicates number of bytes transferred to the client machine. If no content returned then the value would be '-'.
- **Referrer:** This field contains the URL from which the website is referred.
- **User Agent:** This field contain information about client machines browser and operating system used to make the request.

Table 5.3 User agent fragment and their meaning

User agent fragment	**Meaning**
Mozilla/5.0	This is the code name for Netscape navigator and is generally used by several browsers.
Windows NT 6.1	This token is used by windows 7 OS
WOW64	This means 32-bit windows OS is installed on a machine that has a 64-bit processor
AppleWebkit/537.36	This token is used by google chrome/27.0.1412.0
KHTML	This means webkit is built on KHTML engine developed by KDE.
Like Gecko	This means it works like a Gecko browser which provision open internet standards and is used by several browsers.
Chrome/39.0.2171.95	This means stable update of chrome
Safari/537.36	This token is used since chrome/27.0.1412.0

Chapter 6

TECHNOLOGIES USED

6.1 Apache Hadoop

Apache Hadoop is a technology for solving big data problem. It is a framework for storing and processing huge amount of data that is difficult to store and handle using traditional methods and software techniques [20].

The five Daemons of Apache Hadoop are:

1. **Namenode**: Holds metadata of the data stored in datanode.
2. **Secondary Namenode**: Keeps a replica of Namenode metadata.
3. **Jobtracker**: Coordinates and schedules MapReduce jobs.
4. **Datanode**: Stores the data blocks.
5. **Tasktracker**: Does the job assigned by jobtracker and monitors map and reduce tasks.

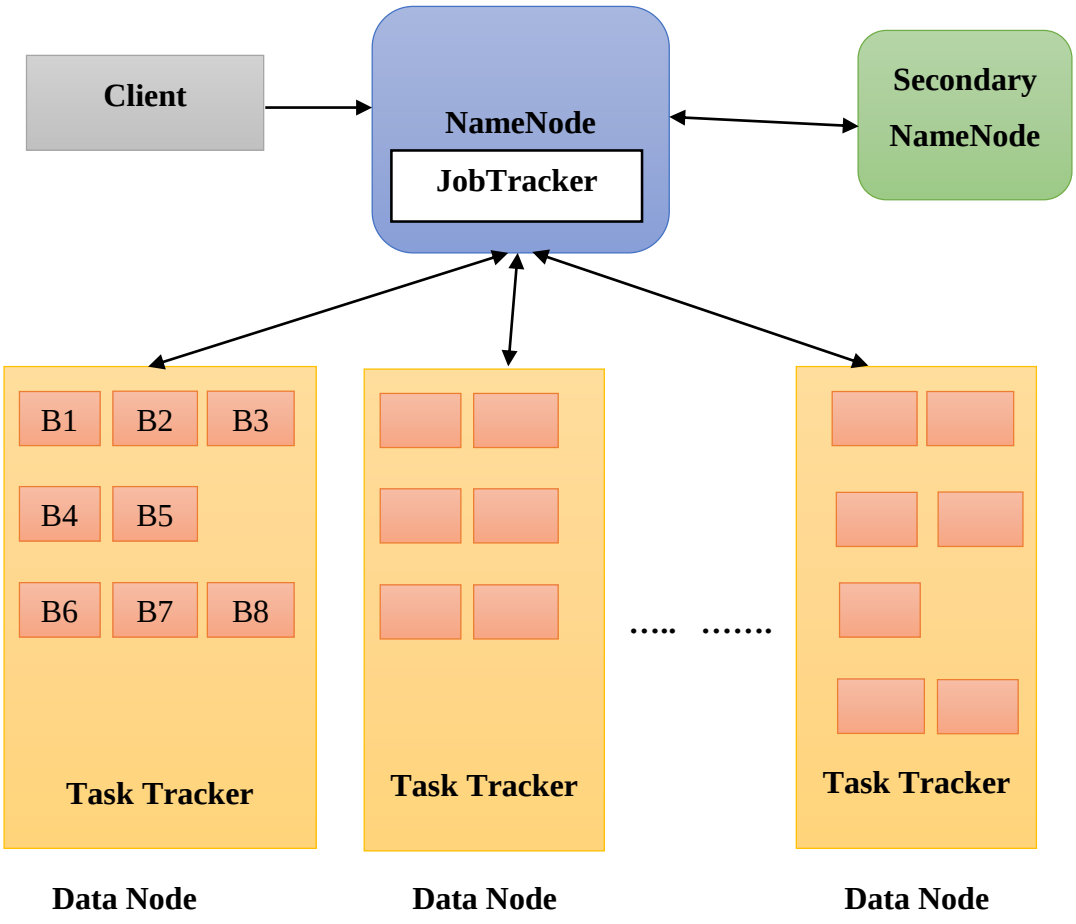

Fig. 6.1 High level architecture of Hadoop

The two main components of Hadoop are:

1. Hadoop Distributed File System for storage and
2. MapReduce for processing

6.1.1 Hadoop Distributed File System (HDFS)

HDFS is the file system used for storage in Hadoop framework. The figure below shows the high-level architecture of HDFS.

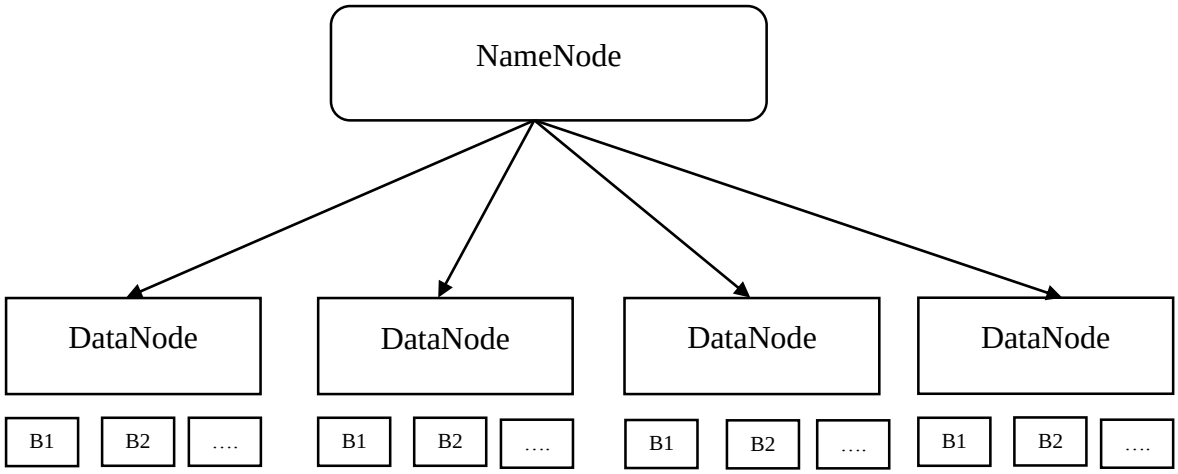

Fig 6.2: Components of HDFS

Some important points about HDFS are:

- In the figure there is one NameNode and multiple DataNodes. B1, B2, B3 represents data blocks.
- When a file is stored into HDFS, it divides the file into multiple blocks and stores on different nodes. HDFS makes copies of this blocks and stores the copies appropriately on different machines in a way that will be dependable and can be recovered quickly. The default HDFS block size is 68 MB.
- Hadoop system guarantees that any node failure will not result in data loss.
- In Hadoop version 1 there is only one NameNode which maintains the metadata of the file system.
- There will be several DataNodes that will stores the data blocks.
- When a query is executed by the user, the DataNode communicates with the NameNode to get the file metadata; once the location of the blocks are known the DataNode will retrieve the real data.
- Hadoop provides a command line interface to manage the data in HDFS.

- Hadoop also provides an in-built webserver for NameNode to browse the data stored in the filesystem and to view some cluster information.
- DataNode sends "heartbeat" and a block report containing the blocks stored in that DataNode every 3seconds to NameNode to say that it is alive.

6.1.2 MapReduce

MapReduce is a parallel programming model for processing the data stored in HDFS. The figure below shows the two main components of MapReduce.

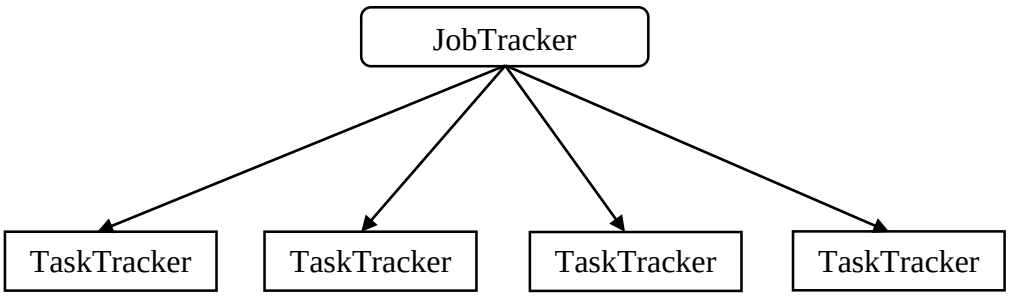

Fig 6.3: Components of MapReduce

Some important points about MapReduce are:

- MapReduce programming technique is used to retrieve and process the data stored in HDFS.
- In this model the programmer don't need to worry about lot of work, the model will take care of parallelizing the tasks, distributing data across the nodes, fault tolerance etc.,
- MapReduce divides the task into smaller parts and executes them on multiple machines in parallel. Thus the work is done very quickly.
- There are different phases in MapReduce programming which include:
- Map Phase
- Shuffling and Sorting Phase
- Reduce Phase
- The programmer have to write code for Map and Reduce phase only. Shuffling and Sorting is taken care by the framework automatically.
- The map function takes input as a list of key and value pair and produces intermediate results as a list of key and value pairs. This intermediate key and value pairs are shuffled and sorted by the framework and another list of key and value pairs are generated. This list of pairs are given as input to reduce function and final output will also generated as key and value pairs.

- JobTracker coordinates and manages all the MapReduce jobs running on multiple machine. It schedules the jobs and if any of the Map are Reduce jobs fails, it reallocates that task to another machine.
- TaskTracker does the tasks assigned by the JobTracker and it also sends heartbeats to jobTracker indicating that it is alive.

6.1.3 MapReduce Workflow

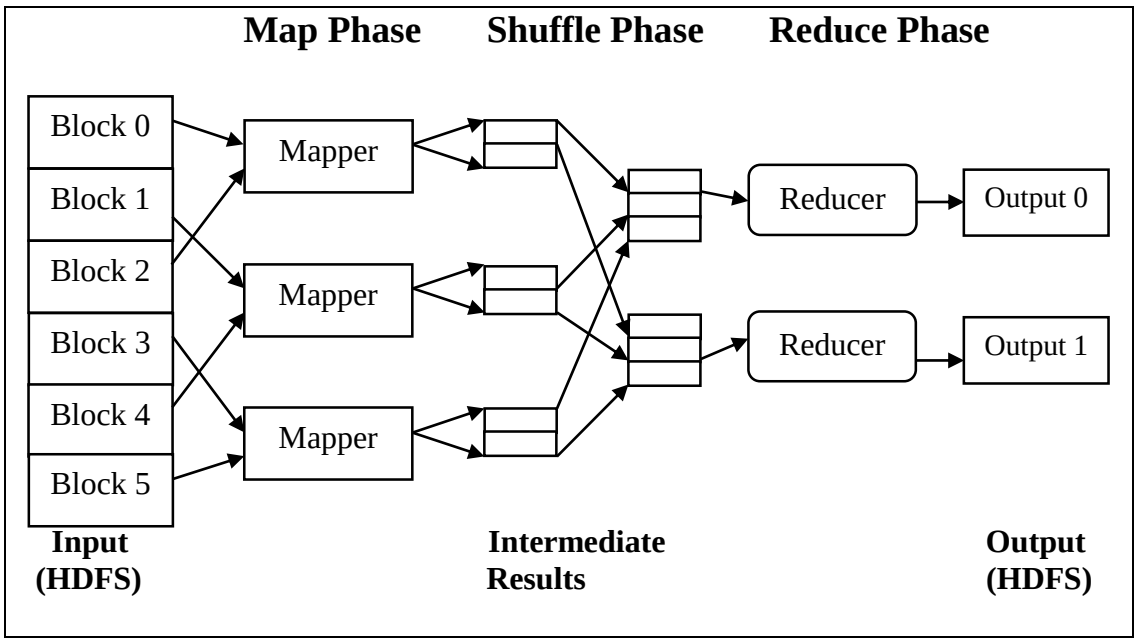

Fig. 6.4: MapReduce Framework

The three phases of MapReduce model is show in Fig. 6.4.

Map Phase:

In Map phase map function is applied to all the input splits. To do this mappers are sent to all the nodes in the cluster. The mappers process the blocks stored locally on each of the node. Processing takes place where data is present. This reduces a significant amount of network traffic. All the mappers execute in parallel and independently.

If a node in the cluster fails, then the work is assigned to another node in the cluster where a replica is present. The mapper takes input each block as a key and value pair and produces output as another key and value pair.

$$\textbf{Map (K1, V1)} \rightarrow \textbf{[(K2, V2)]}$$

In the above equation K1 and K2 are keys and V1 and V2 are values.

Shuffle Phase:

In shuffle phase the output key and value pairs from mapper are mixed and sorted. Then this key and value pairs are given as input to reducers based on their keys. The MapReduce framework assigns all the pairs having identical keys to the same reducer.

Reduce Phase:

The reducer function takes the sorted key and value pairs as input and gathers all the key and value pairs to produces a list of key and value pairs as output.

Reduce (K2, [V2]) → [(K3, V3)]

In the above equation K1 and K2 are keys and V1 and V2 are values.

6.2 Hadoop Ecosystem

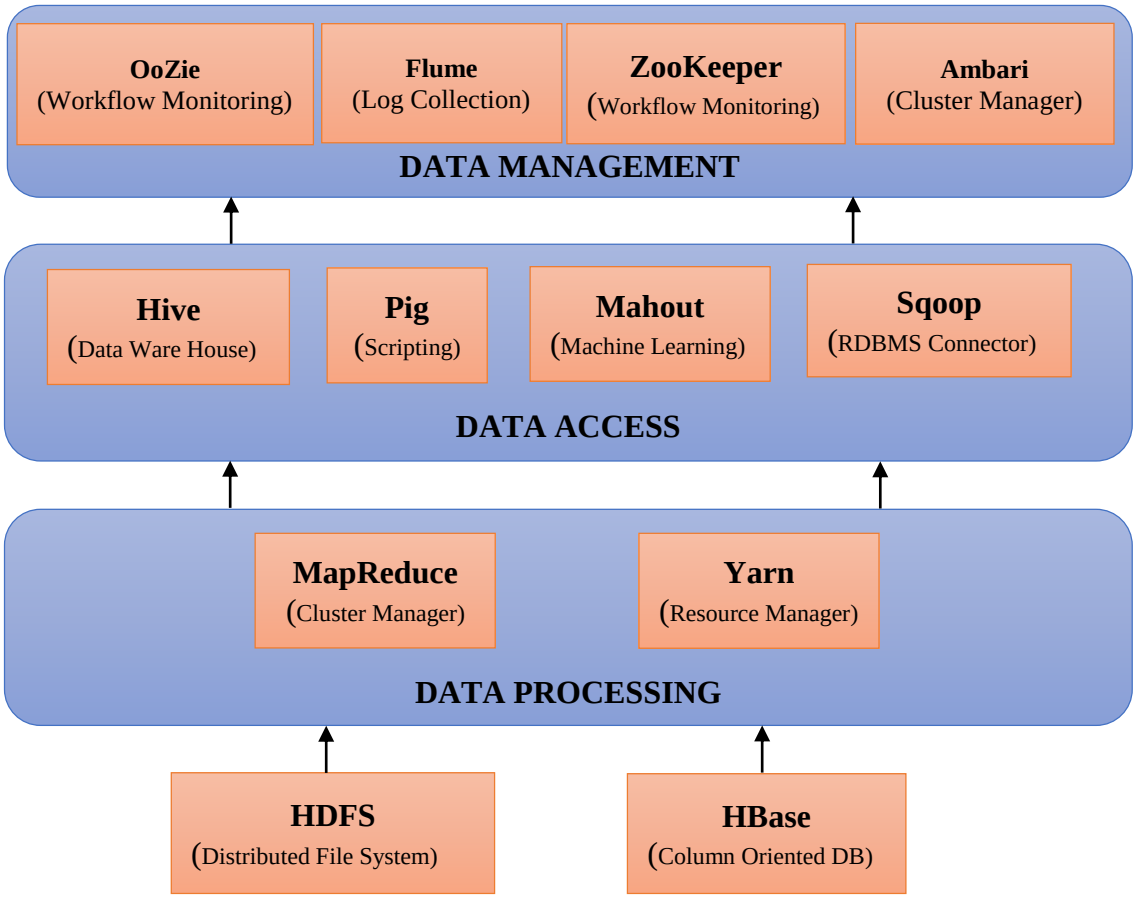

Fig. 6.5 Hadoop Ecosystem

Hadoop is not a single tool. It is a collection of various tools. Some of the important tools are show in Fig. 6.5 and explained in brief below:

- **HDFS**: HDFS is a fault tolerant, low cost file system for storing large datasets.
- **Ambari:** Ambari is a framework for managing and monitoring Hadoop cluster.
- **Apache Sqoop**: Sqoop is used for transferring large amount of data between HDFS and RDMS.
- **Apache Flume**: Flume is a tool for collecting and moving large amount of logs to HDFS.
- **Apache Oozie**: Oozie is used to coordinate and schedule Hadoop jobs.
- **Apache Pig:** Pig is a high level data flow language to handle large amount of data stored in HDFS. It provides a language called Pig Latin.
- **Apache Mahout:** Mahout is for implementing machine learning algorithms in Hadoop
- **Apache Hive**: Hive is a data ware house tool for querying huge amount of data in Hadoop. It provides a language called HiveQL.
- **Apache Hbase**: HBase is a column oriented database for Hadoop.
- **Apache Zookeeper:** Apache Zookeeper provides distributed configuration and synchronization services.
- **Yarn:** Yet Another Resource Negotiator (Yarn) manages resources in Hadoop Cluster.

6.3 Apache Hive

Apache Hive was developed by Facebook. It is a tool built on top of Hadoop to run SQL like queries on huge volumes of data stored in HDFS [21]. It is used by many organizations as data processing platform. Hive converts the SQL query into MapReduce jobs for execution on Hadoop cluster. In Hive data is organized in the form of tables. Metadata such as table schema are stored in a database. The language Hive provides is called HiveQL. Hive is schema on read i.e. schema is checked against the data at run time, it doesn't verify the data against schema when it is loaded. So loads will be faster.

Fig. 6.6 shows the architecture of Hive [22]. The key components of Hive are:

- **External Interfaces**: Hive offers Command Line Interface and Web Interface. It also provides Application Programming Interfaces (API) like ODBC and JDBC.
- **Thrift Server:** Hive server is based on Apache Thrift. Hive server permits a remote user to give quires to Hive, using different programming languages and retrieve results.

- **Metastore:** Metastore includes metadata about the data i.e., data about tables like table owner, column names, SerDe information, partition and bucketing information etc.
- **Driver:** Driver manages the HiveQL statements at the time of compilation and execution. The driver submits the mapreduce jobs to execution engine. Hadoop is the execution engine of Hive.
- **Compiler:** The compiler receives HiveQL statements from the driver and translates this statement into MapReduce jobs.

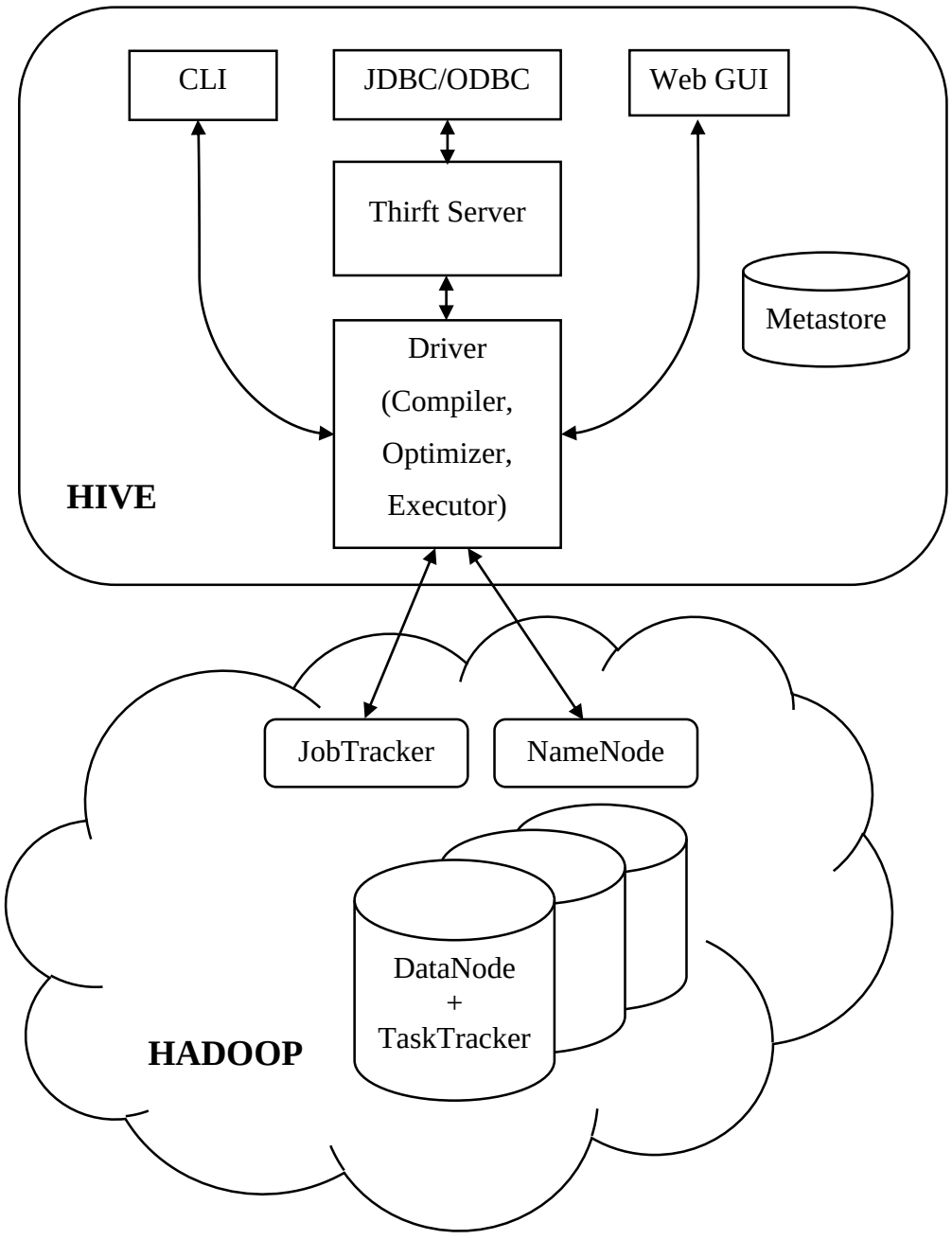

Fig. 6.6: Hive Architecture

6.3.1 HiveQL

HiveQL does not support all the SQL-92 specifications. HiveQL offers extensions to SQL to meet user needs. Further Hive have some features that are not in SQL like multitable inserts where users can perform different queries on the same data etc., and Hive provides only limited sub queries. HiveQL offers Data Definition (DDL) statements to create definite serialization formats. It also supports partitioning and bucketing columns. Data can be loaded into Hive table from an external source and query result can be inserted into Hive tables using 'load' and 'insert' Data Manipulation (DML) statements. Currently HiveQL does not offer updating or deleting the rows from Hive tables. HiveQL also offers User Defined Functions (UDF's) written in java. Users can also implement custom MapReduce scripts written in any language.

6.4 JasperSoft's iReport Designer

iReport Designer is an open source easy to use report designer tool. It creates charts, table, images, sub reports and many more. The data stored in HDFS can be accessed from iReport through JDBC. The reports created can be saved to various formats like PDF, XML, HTML, CSV etc, for later reference.

Chapter 7

SYSTEM IMPLEMENTATION

In this research Hadoop framework is used to compute the log processing in pseudo distributed mode of cluster. The web server logs of www.ubdtce.org website for a period of five months from December 2014 to March 2015 are used for processing in Hadoop environment. The log files are analyzed in Centos 6.6 OS with Apache Hadoop 1.1.2 and Apache Hive 0.10.0.

7.1 Pseudo Distributed Mode

Hadoop framework consist of five daemons namely NameNode, DataNode, JobTracker, TaskTracker, Secondary NameNode.

NameNode: Maintains the metadata of the filesystem.

DataNode: Stores the files as equal sized blocks.

JobTracker: Cordinates the MapReduce jobs.

TaskTracker: Does the job assigned by the jobTracker

In pseudo distributed mode all the daemons run on local machine simulating a cluster. Hadoop configuration files environmental parameters are explained in the below table.

Table 7.1 Hadoop configuration files

Files	Explanation
hadoop-env.sh	This file is used for setting environment variables used by Hadoop
core-site.xml	This file contains Hadoop core configuration, such as I/O settings of HDFS and MapReduce; IP location of master
hdfs-site.xml	This file contains background service settings of HDFS, Number of replications etc.,
mapred-site.xml	This file contains background service settings of MapReduce, jobtracker and tasktracker
Master	This file contains secondary namenode machine IP address
Slaves	This file contains list of DataNode machines IP address

Hadoop can be started after editing the above mentioned configuration files. Once Hadoop is started it will open NameNode, DataNode, JobTracker, TaskTracker in the process. To

know all daemons running issue the command jps which will show process id of each running deamons. The below figure shows the sample output of jps command.

Fig. 7.1 Output of jps command

The Fig. 7.2 shows the steps in implementation:

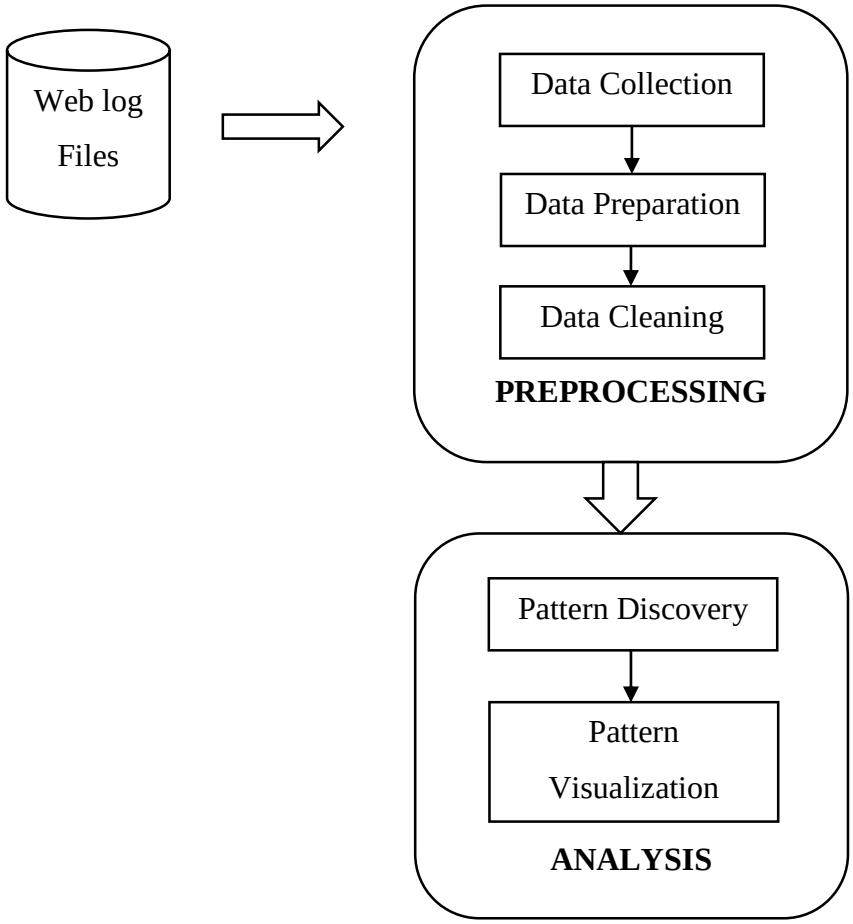

Fig. 7.2 Implementation Flow

7.2 Data collection

Data collection is the first step in log analysis process. It consists of gathering relevant web server access log files. In this experiment five months of access log files of www.ubdtce.org website are collected. Then the log files are moved to specified directory of Hadoop File system.

Hadoop command to load the log files from local file system to HDFS:

```
hadoop fs –copyFromLocal /home/hduser/Desktop/Datasets /user/inputlogs
```

The above command copies Datasets directory from the local system to /user/inputlogs Hadoop file system directory.

7.3 Data Preparation

In this phase an external Hive table is created and a regular expression SerDe is used to parse each field of the log line since the log is in a semi structured form. In Hive Table creation statement the path where log files are stored on the Hadoop File System is specified. Once the table is created, each log line is stored as rows and columns in a table.

HiveQL query Snippet #1:

For converting the semi structured log data into structured data.

```
CREATE EXTERNAL TABLE logdata (
  host STRING,
  identity STRING,
  user STRING,
  time STRING,
  method STRING,
  request STRING,
  protocol STRING,
  status STRING,
  size STRING,
  referrer STRING,
  agent STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
"input.regex" = "([^ ]*) ([^ ]*) ([^ ]*) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(\\w+) (.+?) (.+?)\" (-|[0-
9]*) (-|[0-9]*)(?: \"([^ \"]*|\"[^\"]*\")\" ([^ \"]*|\"[^\"]*\"))? (-|[0-9]*) (-|[0-9]*) ([^ ]*) (-|[0-
9]*)",
"output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s %10s %11s
%12s %13s %14s %15s"
)
STORED AS TEXTFILE
LOCATION '/user/demo/rawdata';
```

Hive provides "Serializer-Deserializer (SerDe)" to convert unstructured or semi structured data into regular table rows and columns. The log files are in a Hadoop directory The SerDe is used to parse the log lines using regular expression into table columns.

7.4 Data Cleaning

Data cleaning, a preprocessing method is applied in the proposed work to filter and minimize the original size of data. In the proposed work the web log file containing request from robots, spider and web crawlers are removed. The entries that have status of "error" or "failure" have been removed. Also some access records generated by automatic search engine agent is identified and removed from the access log. After applying data cleaning step, applicable resources are stored in the HDFS.

The Steps in Data cleaning are:

Step 1: Log lines holding 'robots.txt' in the request field and 'bot' string in the user agent field are filtered.

Step 2: some web bots fake their user agent. Therefore known web bots IP database is downloaded from the internet and compared with our Hive table and matching lines are removed.

After performing the above two steps, the Hive table contains only human hits. That is what we needed.

HiveQL Query Snippet #2:

To remove log lines holding 'robots.txt' in the request field and 'bot' string in agent field

```
CREATE TABLE hh_tmp
AS
SELECT *
FROM logdata
WHERE agent NOT REGEXP '.*(bot|spider|crawler|slurp|spam|fetcher).*' AND request
NOT REGEXP '.*robots.txt.*';
```

HiveQL Query Snippet #3:

Creating a table for storing ip block list database

```
CREATE EXTERNAL TABLE ip_block_list(host string)
ROW FORMAT DELIMITED
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/demo/ip_block_list';
```

HiveQL Query Snippet #4:

Join operation with ip block list table to filter the web bots records.

```
CREATE TABLE human_hits
AS
SELECT A.host,A.identity,A.user,A.time,A.request,A.status,A.size,A.referrer,A.agent
FROM hh_tmp A
LEFT OUTER JOIN
ip_block_list B
ON A.host=B.host
WHERE B.host IS NULL;
```

HiveQL Query Snippet #5:

Creating a table to store human visits.

```
CREATE TABLE human_visits
AS
SELECT *
FROM human_hits
WHERE request LIKE '%.php%';
```

7.5 Pattern Discover and Analysis

Only the lines holding the status code value of "200" is identified as correct log and this correct log records are extracted and stored in HDFS. The major advantage of this step is to eliminate an error that leads to accurate, error free log data, which produce a quality result and increase in efficiency. Than the identification of unique user, unique fields of date, URL referred, and status code are identified. These unique values is retrieved and used for further analysis in order to find the total URL referred on a particular date or the maximum status code got successes on specific date. Total number of visitors in a month, daily visitors are identified.

HiveQL Query Snippet #6:

To extract log lines holding status code "200".

```
CREATE TABLE success_human_visits
AS
SELECT *
FROM human_visits
WHERE status='200';
```

7.6 Pattern Visualization

The analysis results are presented in form for graphs and tables for better understanding of user behavior using a report designer tool. The tool used in this research is JasperSoft's iReport. First a JDBC connection is made to the Hive Thrift Server. Then queries are given to generate required graphs and tables. Finally the generated graphs can be saved as PDF for later references.

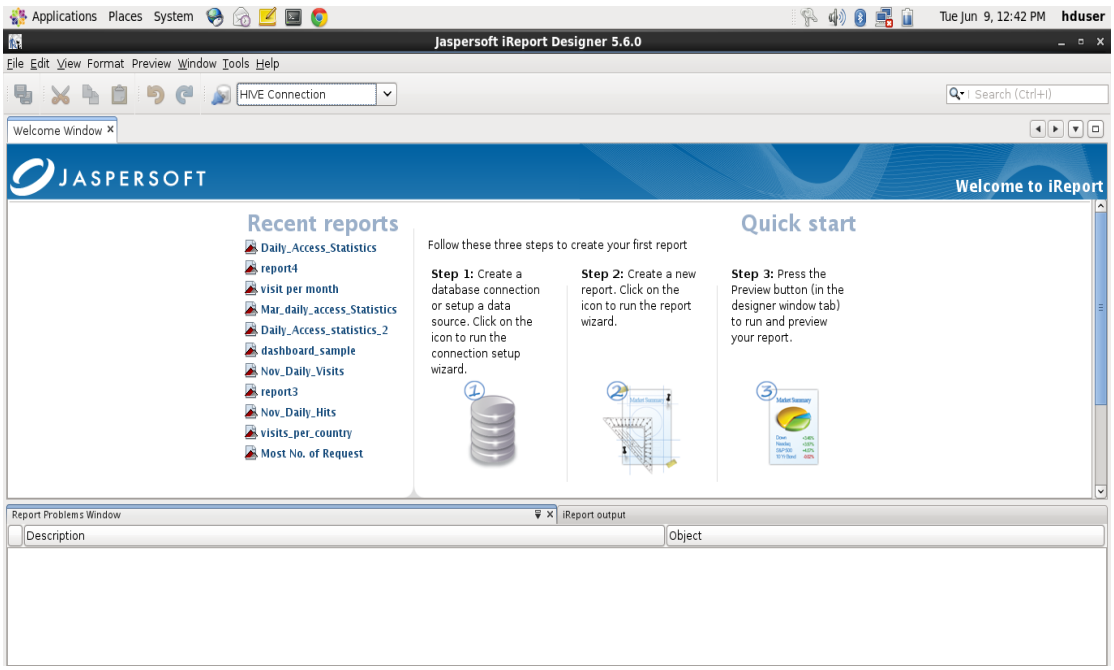

Fig.7.3 Welcome Screen of JasperSoft's iReport Tool

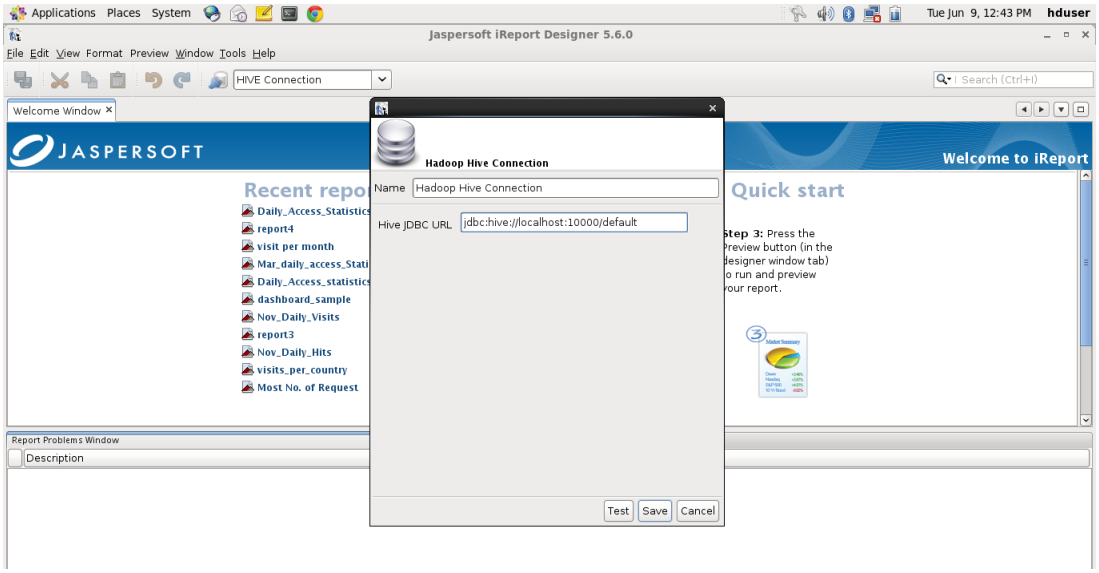

Fig 7.4 Making JDBC Connection to HIVE

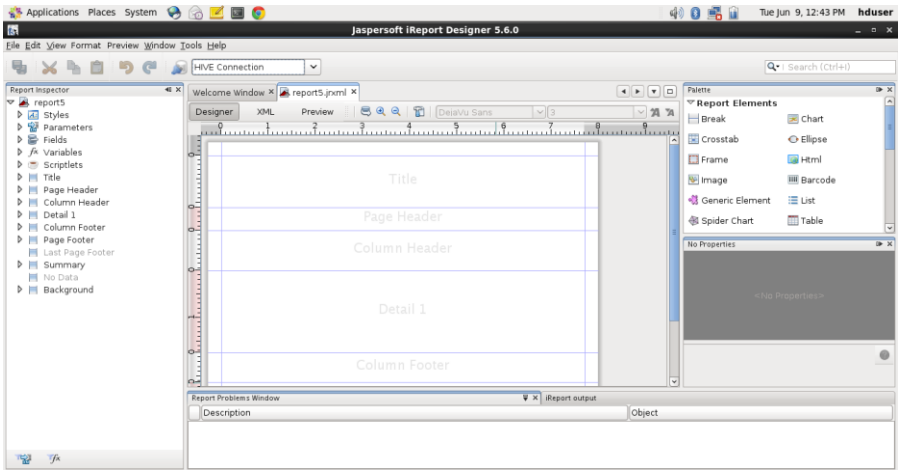

Fig.7.5 Report Creation Window

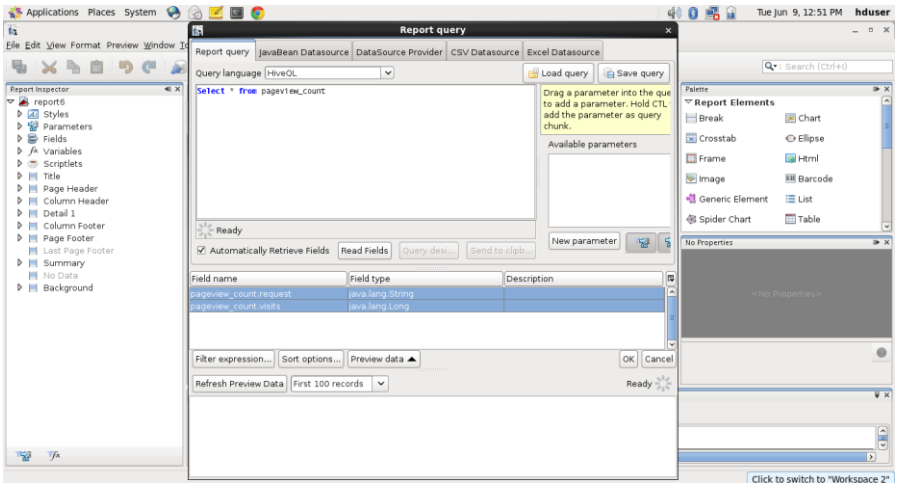

Fig.7.6 iReport Query Window

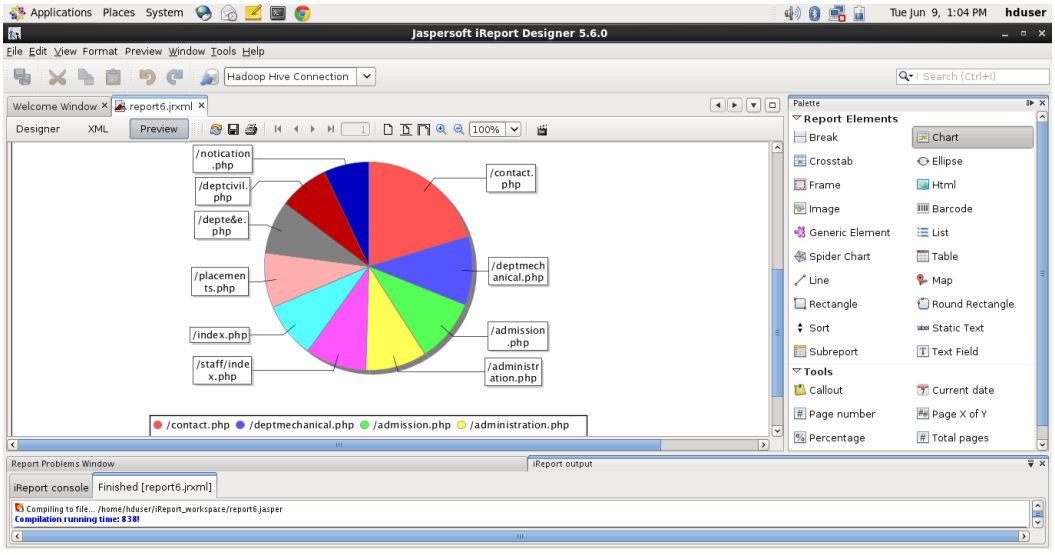

Fig. 7.7 Window Showing Sample Report Generated

Chapter 8

EXPERIMENTAL RESULTS

The experiment is carried out on pseudo-distributed mode of Hadoop. The major benefit of data cleaning is to deliver a quality result and increase in proficiency. After performing Pre-processing step results are shown in table 8.1. It indicates how much reduction happened in the size of data after pre-processing.

Table 8.1 Results Before and After Pre-processing

	Raw Data	**After Cleaning**
File Size	108.4 MB	9.3 MB
No. of Rows	4, 66,621	47, 039

In the current research web access logs were taken from www.ubdtce.org website for the time period 31/Oct/2014 to 31/Mar/2015and the following results were obtained:

8.1 General Statistics

In this case we get general information relating to the website like how frequently the website was hit, total visitors, transfer speed utilized and so forth. It obtains all the general information which one ought to know about a website. Table 8.2. Shows the hits, visits and bandwidth usage of ubdtce.org website for a period of five months.

Table 8.2 General Statistics obtained after analyzing web logs

Hits	
Total Hits	466621
Visitor Hits	422213
Visitors	
Total Visitors/ Unique IPs	4560
Total Page Views	47041
Bandwidth	
Total Bandwidth	8663.54 MB
Visitor Bandwidth	8184.62MB

8.2 Activity Statistics

It gives the measurement on every day and month to month basis. It gives on which days the site was visited maximum.

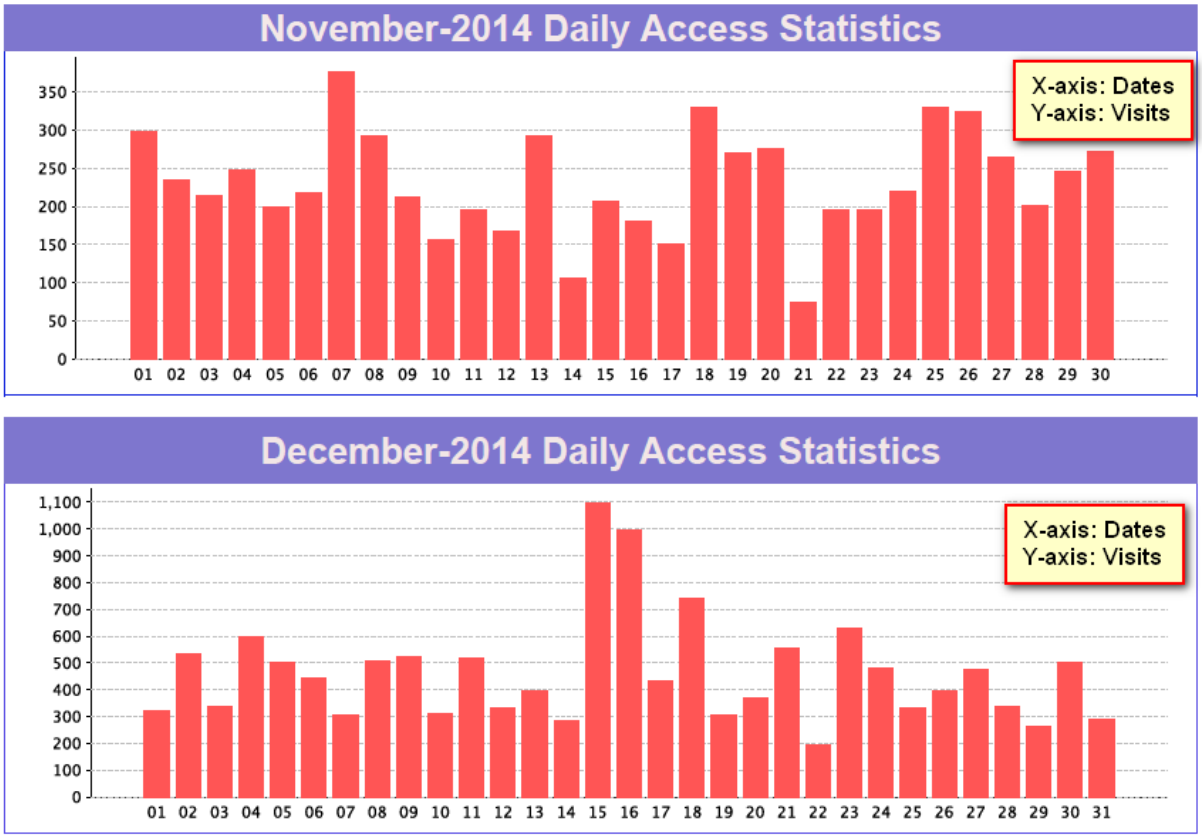

Fig 8.1 November and December daily access statistics

Figure 8.1 shows that more number of visits are on 7th, 13th, 18th, 25th, 26th of November and 15th, 16th, 18th December and very less visitors on 21st of November and 22nd of December. Fig. 8.1 also shows that more number of visitors are in the month of December and very less visitors in the month of October.

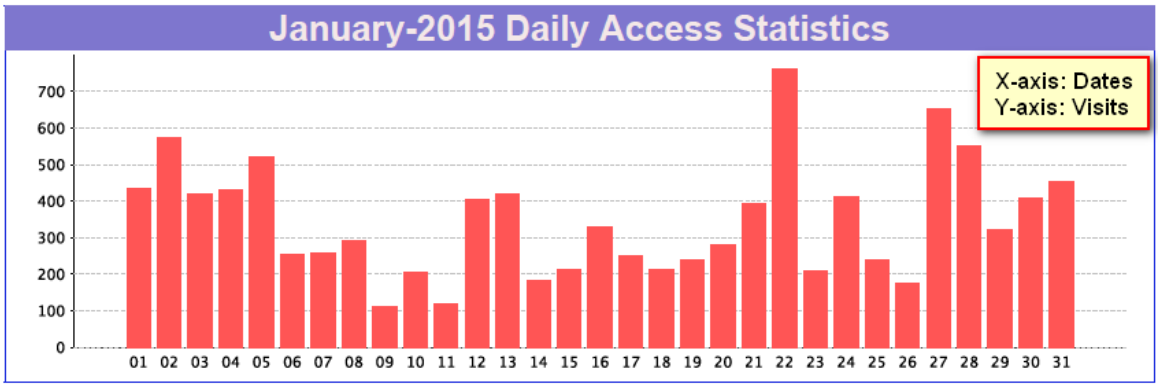

Fig. 8.2 January daily Access Statistics

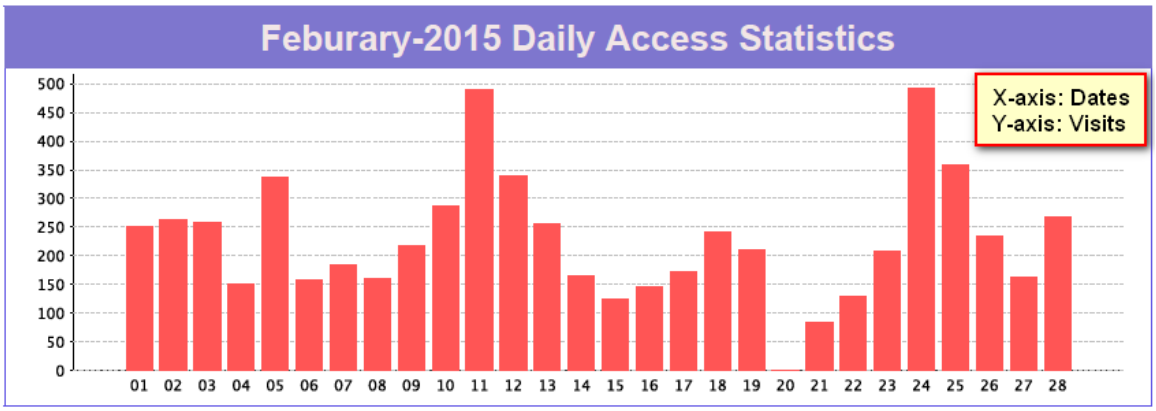

Fig 8.3: February daily Access Statistics

Fig. 8.2 and Fig. 8.3 demonstrate that more number of visits are on 22nd, 27th, 28th of January and 11th, 12th, 24th February and very less visitors on 9th, 11th of January and 20th of February.

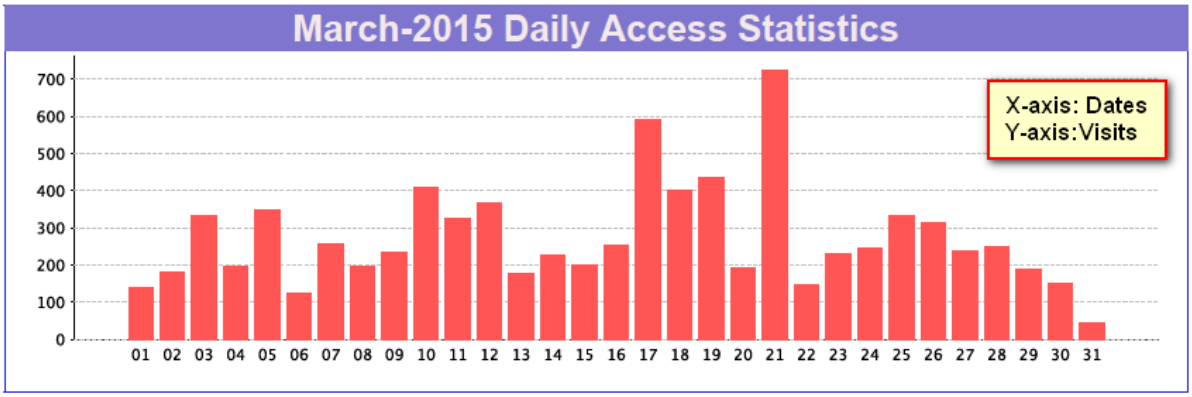

Fig 8.4 March daily Access Statistics

Fig. 8.4 shows that more number of visitors are in the month of December and very less visitors in the month of October.

Table 8.3 shows monthly visitors statistics to the website. More number of visitors are in December and very less in the month of October.

Table 8.3 Monthly visit statistics

Month	Visits
October-2014	206
November-2014	6946
December-2014	14322
January-2014	10742
Feburary-2014	6355
March-2014	8470

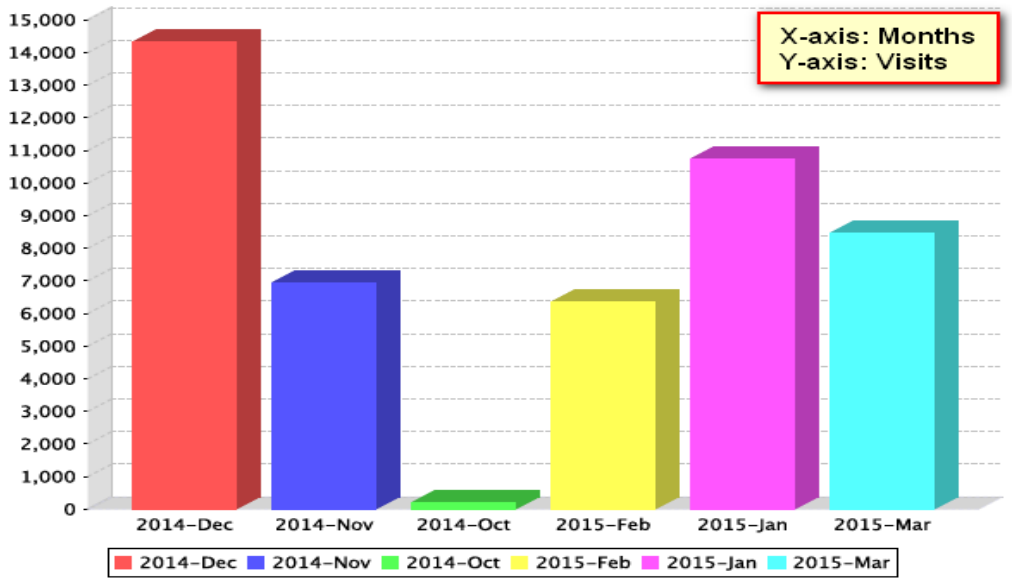

Fig 8.5 Monthly Access Statistics

8.3 Access Statistics

This part of the examination can be viewed as most essential as it gives which IP is generating more hits and more visits and which IP is utilizing high transmission. It helps in discovering that who all go to the website. The table 8.4 shows a list of IP addresses that hit the website alongside how frequently the website was visited by a specific client and the amount of data transmission utilized by every client.

Table 8.4 Access Statistics

Host	Hits	Visitors	Bandwidth (MB)
14.139.152.34	29772	4371	826
216.158.82.218	9391	9262	118
14.139.155.178	1805	143	34
71.198.24.238	1604	93	6
117.241.0.112	1165	214	12
14.141.216.130	1133	180	19
112.133.192.42	1029	150	27
117.240.86.5	811	101	15

Table 8.5 Top ten IP's with most Request

Remote Host	Visits
216.158.82.218	9262
14.139.152.34	4371
64.79.100.18	444
5.39.85.81	380
202.129.240.140	217
117.241.0.112	214
117.211.56.9	208
37.187.56.66	197
188.165.196.25	183
112.133.192.42	180

8.4 Visits-per-country

The table shows Number of visits to the website based on countries. The table 8.6 shows that more number of visits are from India and United States.

Table 8.6 Country code and Visits

Country Code	Visits
IN	25465
US	11099
FR	547
CN	297
UA	124
CA	115
IT	99
RU	89
TR	85

8.5 Error Statistics

The last feature is finding out what kind of errors people face when they access the website. The Fig.8.6 shows the errors users encountered when they accessed the website.

Table 8.7 Error Statistics

Status Code	Hits
200	366444
206	3862
301	41
302	993
304	18836
404	32211
406	152
413	3
508	49

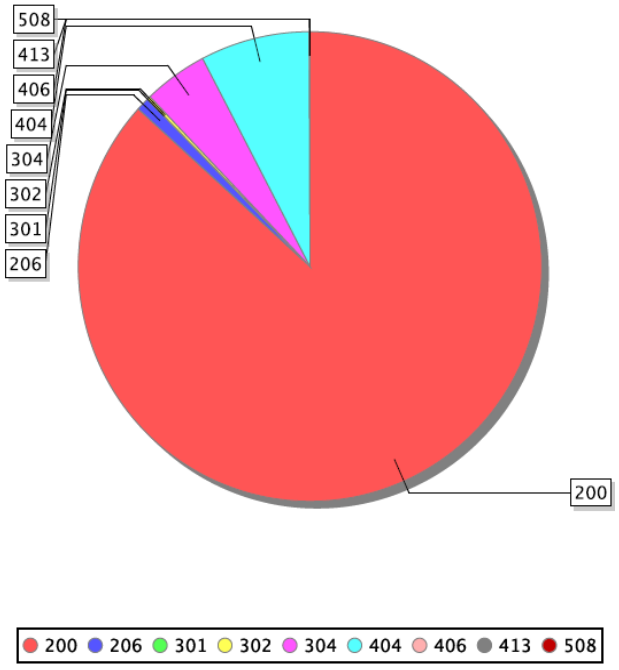

Fig 8.6 Pie chart showing the errors that occur frequently

8.6 Page View Statistics

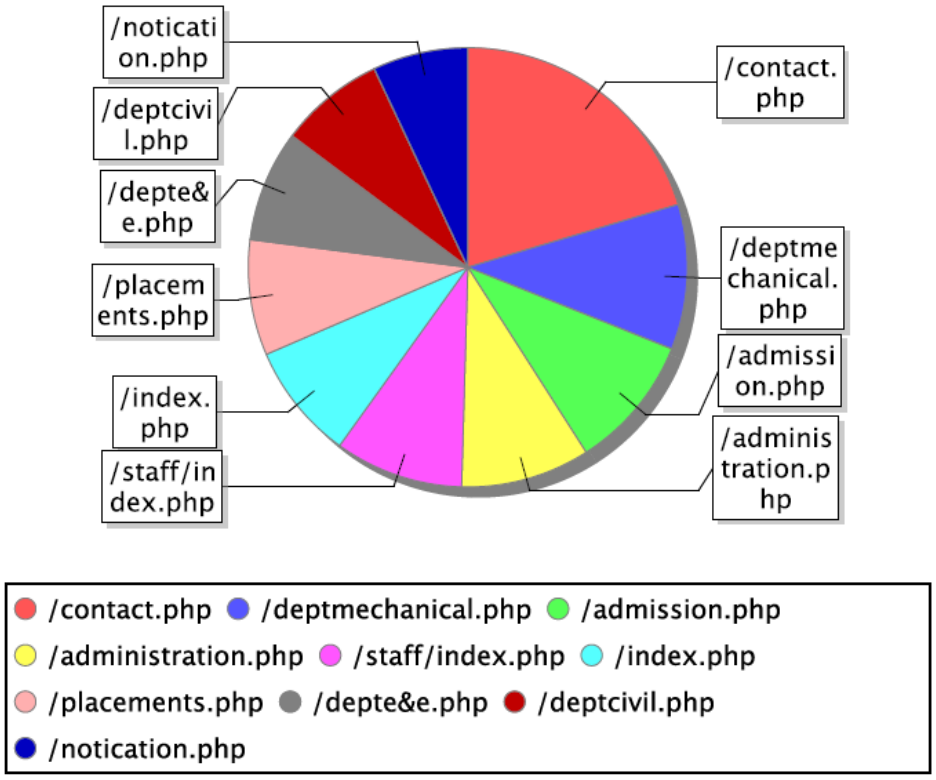

Fig 8.7 Pie chart showing views statistics of top 10 pages

Table 8.8 Views Statistics

Page	Views
/contact.php	2876
/deptmechanical.php	1507
/admission.php	1394
/administration.php	1343
/staff/index.php	1337
/index.php	1212
/placements.php	1200
/depte&e.php	1157
/deptcivil.php	1096

Table 8.8 shows the frequently visited pages in the website. Contact page is visited most number of times in the website. Mechanical department page, admission page and administration are other frequently visited pages in the web site.

Table 8.9 Top 10 Operating Systems used to make request

User Agent	Visits
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)	9262
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36	1416
Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36	1017
Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; MAARJS; rv:11.0) like Gecko	1009
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36	1000
Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36	917
Mozilla/5.0 (Windows NT 6.1; rv:35.0) Gecko/20100101 Firefox/35.0	739
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.71 Safari/537.36	638
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2	634
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.71 Safari/537.36	626

Chapter 9

CONCLUSION AND FUTURE ENHANCEMENTS

9.1 Conclusions

Web sites are one of the way for advertisements. So as to have outlined results for a specific web site, we have to do log examination that will help to enhance the business methodologies and in addition to produce measurable reports. In this project based on Hadoop framework web server log files are analyzed where data get stored on multiple nodes in a cluster so that access time required can be reduced and MapReduce works for large datasets giving efficient results.

Utilizing visualization tool for log analysis will give us graphical reports indicating hits for web pages, client's movement, in which part of the web site clients are interested. From this reports business groups can assess which parts of the site need to be enhanced, which are the potential clients, from which geographical region site is getting more hits, and so on., which will help in planning future marketing plans. Log analysis should be possible by different techniques however what is important is response time. Hadoop MapReduce model gives parallel distributed processing and reliable data storage for huge volumes of web log files. Here Hadoop's characteristic of moving processing to data rather than moving data to processing helps to enhance response time.

9.2 Future Enhancement

There are various issues in preprocessing of log information. Examining web client access log records helps to read the user behaviors in web structure to enhance the design of web site and web applications. It is important to remove messy data from log files. So cleaning is done to accelerate the examination process as it reduces the quantity and increases the quality of the results. Finding the user sessions are to be the most productive in the creation of effective web site. After creating sessions finding the path traversed by the user will be easier. More research is possible in processing stages to clean the log files, and to distinguish clients and to develop exact sessions.

BIBLIOGRAPHY

[1] Chen-Hau Wang, Ching-Tsorng Tsai, Chia-Chen Fan, Shyan-Ming Yuan, "A Hadoop Based Weblog Analysis System", 7th International Conference on Ubi-Media Computing and Workshops, IEEE 2014, pp.72-77.

[2] Savitha K, Vijaya M S, "Mining of Web Server Logs in a Distributed Cluster using Big Data Technologies", International Journal of Advanced Computer Science and Applications, Vol.5, NO.1, 2014, pp. 137-142.

[3] Siddharth Adhikari, Devesh Saraf, Mahesh Revanwar, Nikhil Ankam, "Analysis of Log Data and Statistical Report Generation Using Hadoop", International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 2, Issue 4, April 2014, pp. 4054-4058.

[4] Neha Goel and C.K.Jha, "Analyzing Users Bhavior from Web Access Logs using Automated Log Analyzer Tool", International Journal of Computer Applications, Vol. 62-No.2, January 2013, pp. 29-33.

[5] Naga Lakshmi, Raja Sekhara Rao, Sai Satyanarayana Reddy, "An Overview of Preprocessing on Web Log Data for Web Usage Analysis", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 2, Issue 4, March 2013, pp. 274-279.

[6] Harleen Puri, Arvind Selwal, Anuradha Sharma, "An Empirical Proposal Towards the Algorithmic Approach and Pattern in Web Mining for Assorted Applications", International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 1, Issue 2, April 2013, pp. 293-297.

[7] Ramesh Rajamanickam and C.Kavitha, "Fast Real Time Analysis of Web Server Massive Log Files Using an Improved Web Mining Architecture", Journal of Computer Science, Science publication, ISSN: 154-3636, June 2013, pp. 771-771.

[8] Sayalee Narkhede and Tripti Baraskar, "HMR Log Analyzer: Analyze Web Application Logs over Hadoop MapReduce," International Journal of UbiComp (IJU) vol.4, No.3, July 2013, pp.41-51.

[9] Kanchan Sharadchandra Rahate "A Novel Technique for Parallelization of Genetic Algorithm using Hadoop," International Journal of Engineering Trends and Technology (IJETT), vol.4, issue 8, August 2013, pp. 3328-331.

[10] Milind Bhandare, Vikas Nagare et al., "Generic Log Analyzer Using Hadoop Mapreduce Framework," International Journal of Emerging Technology and Advanced Engineering (IJETAE), vol.3, issue 9, September 2013, pp. 603-607.

[11] Wichian Premchaiswadi, Walisa Romsaiyud, "Extracting WebLog of Siam University for Learning User Behavior on MapReduce", 4th International Conference on Intelligent and Advanced Systems (ICIAS 2012), IEEE 2012, pp. 149-154.

[12] Ravindra Gupta, Prateek Gupta, "Application Oriented Web usage mining with customized web log preprocessing and frequent pattern tree", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 1, February 2012, pp. 596-598.

[13] M. Venkata Krishna, L. Raghavendra Raju, "Analysis of web mining and evolving of user profiles", International Journal of Engineering and Innovative Technology (IJEIT), Vol. 1, Issue 4, April 2012, pp. 206-208.

[14] Rahu Mishra, Abha Choubey, "Discovery of Frequent Patterns from Web Log Data by using FP-Growth algorithm for Web Usage Mining", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 9, September 2012, pp. 311-318.

[15] L. K. Joshila Grace and others, "Analysis of web logs and web user in web mining", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.1, January 2011, pp. 99-110.

[16] C.P.Sumathi, R.Padmaja Valli, T.Santhanam, " An overview of Preprocessing of Web Log Files for Web Usage Mining", Journal of Theoretical and Applied Information Technology, Vol. 34, Number 2, 31st December 2011, pp. 178-185.

[17] Murat Ali, Ismail Hakki Toroslu, "Smart Miner: A New Framework for mining Large Scale Web Usage Data," WWW 2009, April 20-24, 2009 Madrid, Spain, pp.161-170.

[18] Bina Kotiyal, Ankit Kumar, Bhaskar Pant, RH Goudar, "Big Data: Mining of Log File through Hadoop".

[19] Sayalee Narkhede, Trupti Baraskar, "Analyzing Web Application Log Files to Find Hit Count Through the Utilization of Hadoop MapReduce in Cloud Computing Environment".

[20] Tom White, "Hadoop: The Definitive Guide", O'Reilly , 2009

[21] Jason Rutherglen, Dean Wampler and Edward Capriolo, "Programming Hive", O'Reilly, 2012.

[22] Ashish Thusoo, Joydeep Sen Sarma and others, "Hive-A Warehouse Solution over a MapReduce Framework", VLDB'09, August 24-28, 2009.

Author Publication and Paper Presented

[1] Harish S and Kavitha G, "Statistical Analysis of Web Server Logs using Apache Hive in Hadoop Framework", at International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Volume 03 Issue 5, May 2015.

[2] Harish S, has presented the paper "Statistical Analysis of Web Server Logs using Apache Hive in Hadoop Framework" and won First Prize at State level technical paper presentation contest "TECHGYAAN-2K15", organized by department of computer applications, UBDTCE, Davanagere.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

ISSN (Online): 2320 – 9801 ISSN (Print): 2320 - 9798

PUBLICATION CERTIFICATE

This is to certify that

HARISH S

PG Student, Dept. of Studies in CSE, UBDT College of Engineering, Davangere, Karnataka, India

Published a research paper titled

"Statistical Analysis of Web Server Logs Using Apache Hive in Hadoop Framework"

in IJIRCCE, Volume 3, Issue 5, May 2015

Certificate No: V3I05C368
Date: 30th May 2015

IJIRCCE
Impact Factor: 5.618
www.ijircce.com

Editor-in-Chief
IJIRCCE

UNIVERSITY .B.D.T COLLEGE OF ENGINEERING, DAVANGERE

(A CONSTITUENT COLLEGE OF VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM)

Department Of Master of Computer Applications

TECHGYAAN-2K15

(STATE LEVEL TECHNICAL PAPER PRESENTATION CONTEST)

CERTIFICATE OF ACHIEVEMENT

This is to certify that Mr /~~Ms~~ /~~Mrs~~ Harish S

Of UBDTCE, Davangere has FIRST PRIZE in the event

TECHGYAAN-2K15 (STATE LEVEL TECHNICAL PAPER PRESENTATION CONTEST)

ORGANIZED BY DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS, UNIVERSITY .B.D.T COLLEGE OF ENGINEERING, DAVANGERE.

Dr.Mohamed Rafi
Professor and Chairman , DOS in CS&E

Mrs. Veena.B
Chief Co-ordinator, DEPT of MCA

Principal 2/5/15