Trend Analysis based on Access Pattern over Web Logs using Hadoop

Jalpa Mehta
M.Tech (Computer Science), Assistant Professor
Shah & Anchor Kutchhi Engineering College
Mumbai, India

Amir Ansari
B.E Scholar, Department of Information Technology
Shah & Anchor Kutchhi Engineering College
Mumbai, India

Aseem Girkar
B.E Scholar, Department of Information Technology
Shah & Anchor Kutchhi Engineering College
Mumbai, India

Ayesha Khanna
B.E Scholar, Department of Information Technology
Shah & Anchor Kutchhi Engineering College
Mumbai, India

Ankit Nagda
B.E Scholar, Department of Information Technology
Shah & Anchor Kutchhi Engineering College
Mumbai, India

ABSTRACT
There is an invariable progress and extension of the World Wide Web which has resulted into the generation of log files having enormous magnitude of data. Log files incorporate traits of user behavior, therefore it is essential to analyze log data and acquire knowledge from it. Web mining techniques primarily focuses on deciphering and scrutinizing the navigational behavior of user form various aspects and ascertaining the hidden knowledge from these web logs. As log files over the web are outsized, storage becomes a constraint wherein effective techniques such as virtual database prove to be ineffectual for the same. Conversely, Hadoop offers a large scale distributed batch processing infrastructure that provides adequate data storage, distributive and analogous processing, isolation of process and fault tolerant on occurrences of data loss. This paper characterizes on the dominant approach for managing the large chunk of web log data using Hadoop MapReduce which reduces the response time for throughput generation, loads the log data effectively and ensures reliability. The primary focus of the paper is to construct log analysis system which depicts trends based on the users browsing mode using Hadoop MapReduce which facilitates handling of heterogeneous query execution on log file.

Keywords
Cloudera, Hadoop, MapReduce, Log Files, Web Mining, MySql Database, Hadoop Distributed File System, Trend Analysis.

1. INTRODUCTION
A significant development in the field of technology in sectors such as business, public and private has been observed leading to accumulation of large data over the web. Information acquired from the web are used to describe the exponential growth and availability of data, both structured and unstructured. As data over the web is heterogeneous in nature, analyzing such data is necessary in order to gain acquaintance wherein log file analysis is an effective solution. Log files are the files that list the actions that have been occurred and reside in web server [5]. There prevails a need to process and store log files using traditional techniques however in the enterprise scenario the data from these log files is outsized due to which processing capacity of conventional approaches becomes incompetent for gaining information for processing.

1.1 Traditional Web Analysis
Web mining entails an extensive range of applications that primarily endeavors at analyzing and extracting concealed information from the data stored over the Web. An additional vital principle of Web mining is to endow with techniques that formulate the data access more proficiently and adequately. Web mining approaches are classified into three varied categories on the origin from which web data is obtained. These categories are:

(i) Web content mining ii) Web structure mining (iii) Web usage mining. Web content mining involves the determination of constructive information available on-line, depending upon different types of Web content which can provide beneficial information to users. Web structure mining is the course of action for ascertaining the composition of hyperlinks within the Web. It endeavors to distinguish the authoritative and the hub pages for a given subject. Web usage mining is an application of data mining that can be used to discover user access patterns from web log data [3]. Web usage mining emphasizes the task of ascertaining the activities of the users whilst they are browsing and navigating through the Web.

1.2 Web Log Analysis
Web server logs stores click stream data which can be beneficial for mining purpose. Web Log mining is the outcome of web usage mining which contains information of web access of diverse users. At this juncture, any kind of information access which is recorded by the web server into log file for corresponding data is accumulated. Analysis of log files provides the absolute facet of the access patterns of the users, for example varied outline of the user's behavior, operating system used, particular session of usage in the way of successful/ unsuccessful transactions etc., thus summarizing all these information in a predefined format.

1.3 Log Format

1.3.1 Common Log Format

An emblematic configuration for access log files is as follows:

Log Format *"%h %l %u %t \"%r\" %>s %b"*

The log file entries produced in CLF appears to be in the following format:

127.0.0.1 – Jack [31/Jan/2015:07:07:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2345

Each field specified in the above log file is as follows:

127.0.0.1 (%h) – This is the IP address of the client from which the request is made to the server. The IP address generated on server will not necessarily be the address of the client. Sometimes, there are existence of proxy server between the client and the server and therefore, the address of proxy server is generated, rather than actual user's address.

(%l) – The "hyphen" in the log output indicates that the requested piece of information is not available.

Jack (%u) –This is the user id of the client requesting the page as determined by HTTP authentication of the server.

[31/Jan/2015:07:07:36 -0700] (%t) –This indicates the date and time of the client which made request to the server. The format is as follows: [day/month/year:hour:minute +/- timezone]. Here, 31/Jan2015 is the date and 07:07:36 is the time at which the request was generated and -0700 is the time zone.

"GET /apache_pb.gif HTTP/1.0" (\"%r\") – The request from the client is generated in double quotes that includes three pieces of information. GET is the HTTP method used, /apache_pb.gif is the requested resource and HTTP/1.0 is the HTTP protocol used by the client with version (1.0).

200 (%>s) – This is the status code which indicates the success or failure of the HTTP request. The codes beginning with 2 indicate successful response, 3 indicate redirection, 4 indicate error caused by the client and 5 indicates the error caused by the server. .

2345 (%b) – This indicates the size of the object in the form of bytes of data transferred as part of the HTTP request, not including the HTTP headers.

1.3.2 Combined Log Format

Another log format called Combined log format is used which is the extension of Common log format. The Combined log format is as follows:

LogFormat *"%h %l %u %t \"%r\" %>s %b \"%{Referrer}i\" \"%{User-agent}i\""*

The log entries in combined log format will look like:

127.0.0.1 – Jack [31/Jan/2015:07:07:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2345 "http://www.google.com/" "Mozilla/4.05 [en] (WinNT; I)"

The following additional fields are described as follows:

"http://www.google.com/" (\"%{Referrer}i\")– This indicates the URL which links the user to the website.

"Mozilla/4.05 [en] (WinNT; I)" (\"%{User-agent}i\") – This is used to determine the web browser and the operating system used by the client.

2. LITERATURE SURVEY

2.1 Hadoop

As web generated data is enormous in nature, the users are highly inclined towards the web arena. This massive data accumulation over the web is in terms of terabytes or petabytes is stored in the form of a predefined format of log files comprising user behavior, IP address and other such web based attributes. As these datasets are huge in nature, scrutinizing such datasets requires functionalities like parallel processing and for the storage purpose a reliable storage system is essential. Conversely, virtual databases provide a constructive elucidation for amalgamation of data yet it becomes incompetent for outsized datasets. In order to provide the respective functionalities, there subsists a framework named Hadoop framework which aims at providing unswerving and reliable data storage with the assistance of Hadoop distributed file system (HDFS) and attains analogous processing system for colossal datasets with the help of another key component of the framework i.e. MapReduce. The foremost approach offered by means of Hadoop is to "Store first query later", as it loads the data to the Hadoop Distributed File System and then accomplishes and executes the respective queries.

2.2 Hadoop Distributed File System

Hadoop Distributed File System seizes an outsized log files in a superfluous manner across numerous machines in order to accomplish an elevation in terms of accessibility for parallel processing as well as resilience and recovery on occurrence failures. There is a provision of high throughput access to the log files. It is considered to be a block structured file system as it fragments the log files into small blocks which are of fixed size depending upon the percept of the user. Blocks are replicated by the replication factor over a period of time across multiple machines within the Hadoop cluster due to which on occurrences of failure the loss of data can be recuperated.

2.3 MapReduce

MapReduce is an effortless programming model which is essential for parallel processing of large dimension of data which can be of a structured or unstructured format list of data. Elementary conception of MapReduce is to renovate lists of effort data to lists of productive data. On certain occurrences when data is not in the best of its format and difficult to decipher an outline is required to make the change in the input data. This adaptation is done by MapReduce in order to make the input data comprehensible and this is done in two variant phases namely: Map and Reduce phase simply by segregating the entire work load into fragmented tasks and dispensing them over the numerous machines within the Hadoop cluster.

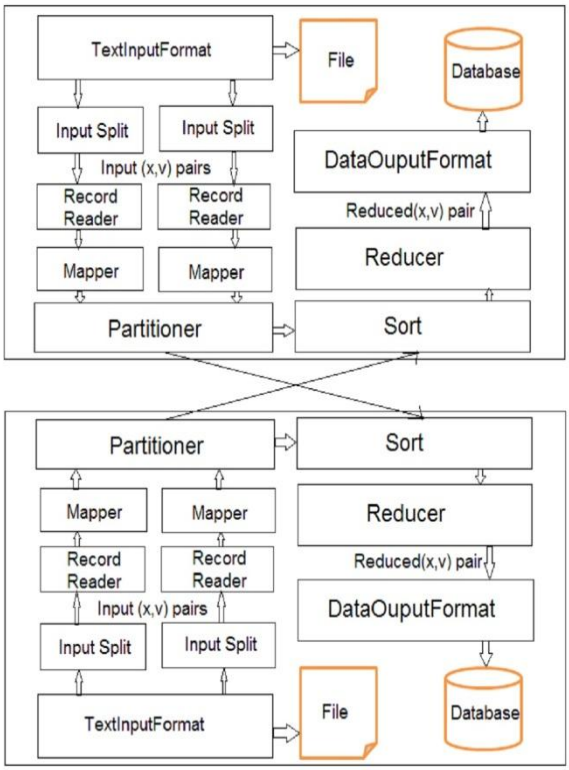

Figure 1. Workflow of MapReduce

2.3.1 Mapper Class

The Mapper class is the user-defined first phase of the MapReduce program. The Mapper class includes a key and value pair where the map() method emits (key, value) pair(s) which are forwarded to the Reducer class. A new instance of Mapper class is used in a separate Java process for each map. This process is known as Input Split that is used part of the total job input. However, the individual mappers are not provided with a mechanism to interact with one another in any way which enables the reliability of each map task to be governed exclusively by the consistency of the local machine. The Mapper class provides map() method with two parameters along with the key and the value: *OutputCollector* and *Reporter* object. The *OutputCollector* object comes with a method named collect() which will forward a (key, value) pair to the reduce phase of the job. The *Reporter* object processes information about allthe current task and also makes sure that the map task provides additional information about its progress to the rest of the system. Each mapper has the capability to increment the counters, and the JobTracker collects the increments made by the different processes and summative them for later retrieval when the job ends [7].

2.3.2 Reducer Class

Followed by partition, shuffle and sort, a Reducer instance is created for each reduce task. This instance is then used to perform the second important phase of job-specific work. Like Mapper, each key in the partition is assigned to a Reducer. Each time the key is assigned to the Reducer, the Reducer's reduce() method is called once. This receives a key along with an iterator over all the values coupled with the key. This values that are associated with a key are returned by the iterator in an undefined. The Reducer receives parameters as the OutputCollector and Reporter objects. These two objects are used in the same manner as used in the map() method.

3. PROPOSED SYSTEM

The information in the log files of the server principally encompass the actions of the users, cannot be used for mining purposes due to the pre-biased outline of the log files proves to be inappropriate for the desired mining methodology. Therefore the contents of the log file should be cleaned in the preprocessing step. The core concept at the rear of preprocessing is to eradicate superfluous data and to attain minimized file.

3.1 Pre-Processing

Data preprocessing is a preliminary data mining technique that entails transforming unprocessed data into a comprehensible format. Data cleaning is the primary phase carried out in the anticipated work as a preprocessing step in web server log files. The log file encloses a number of records that also incorporates a large amount of erroneous, ambiguous, and curtailed information. The preprocessed log value includes the timestamp, date, browser version and total bytes sent for a request. The session identification plays a vital utility in web log mining. This is done by using the time stamp details of the web pages. In addition to its session is the time duration spent in the web page thereby the overall instances used by every user of each web page is also a methodology by which session identification is possible. Another alternative for identifying session is by noting down the user id of those users who have visited the web page and have traversed through the links of the web page. The third step is to convert the data into the format considered necessary by the mining algorithms. If the sessions and the sequences are identified, this step can be accomplished more easily. The preprocessed log file is used to find the user identification, as MapReduce in general identifies the unique values based on key value pair. The log file consists of different fields and the user is identified based on IP address, which is considered as key and their corresponding count as value. This is conversion of the log file data into the format needed by the mining algorithms.

4. IMPLEMENTATION

Web log analysis for determining the trends is carried out using Hadoop approach. Hadoop makes use of Map and Reduce technique to fulfill the purpose with the help of java.util.regex package for pattern matching with regular expressions. The common log file or combined log format is expressed using regular expression, so as to obtain different fields from the log file and grouping them on the basis different categories such as IP address, username, request type, requested URL, status codes, size of page, referrer and user agent. The access log files which are available in the form of text file format are input files to MapReduce. The Splitting of input files is done based on regex expressions which treats each line as an input file and starts splitting the data into chunks on comparing with the defined regex format. Though MapReduce breaks the input into chunks, the regex expressions which are constructed based on the specialized log file sequence ensures an improved performance along with the parallelization. Each logs files are read and converted into (key, value) pairs using RecordReader. Mapper class uses Map() method which is called at each occurrence of RecordReader. Once the first Map task is completed, nodes start exchanging the intermediate output from the Map tasks to intend Reduce tasks which in turn known as shuffling. The Reducer calls Reduce() method at each key in the partition and then uses DBOutputWritable in Reduce class to process log data as per (key, value) pairs in

the database. The DBOutputFormat allows the output data to store in the database.

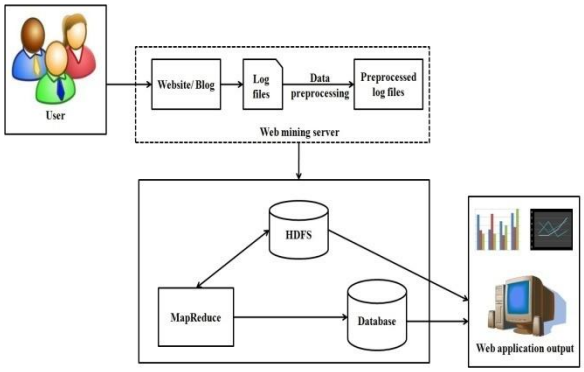

Figure 2. System Architecture

5. EXPERIMENTAL SETUP

For the trend analysis of a website/blog, an access log file of a blog with gigabytes of data has been acquired. These log files includes various information such as IP address, username, date, time, requested URL, referral URL, user agent information which enables to determine location, total hits and trending topics on a particular website/blog. The pre-processing of log files is handled by Hadoop setup of Cloudera Virtual machine. MySql has then installed on the machine for storage of log information in the database after its separation and for query processing while displaying the visual output. For processing of all log file information, create a separate web application which will allow users to interact with it and check for the statistics of their website/blog. The analyzed results are visualized in the form of pie-charts and bar graphs. The database output let us determine the webpage with most hits, most popular accessed keywords, total page views, unique visitors as per date, time, etc.

6. CONCLUSION

Trend analysis portrays the users browsing pattern and summarizes the outcome into a graphical report which depicts most visited web pages, browsing session and trending keywords. Hadoop MapReduce framework provides parallel distributed processing and reliable data storage for large volumes of log files. In order to manage such log files, Hadoop MapReduce plays a key role by proficient management of data and decreases the response time. The proposed system with the help of Hadoop MapReduce analyzes the log files and segregates the fields of the log files using regular expression mechanism. Regex not only reduces the code length but also reduces the overhead of usage of string functions. The segregated and structured fields are stored in the database in accordance with Hadoop thereby enabling ease of data retrieval.

7. REFERENCES

[1] 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

[2] Milind Bhandare, Prof. Kuntal Barua, Vikas Nagare, Dynaneshwar Ekhande, Rahul Pawar, " Generic Log Analyzer Using Hadoop Mapreduce Framework", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 9, September 2013

[3] Navin Kumar Tyagi, A. K. Solanki and Manoj Wadhwa, "Analysis of Server Log by Web Usage Mining for Website Improvement", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 8, July 2010

[4] L.K .Joshila Grace, V.Maheswari, Dhinaharan Nagamalai, "ANALYSIS OF WEB LOGS AND WEB USER IN WEB MINING", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.1, January 2011

[5] Praveen Kumar, Dr Vijay Singh Rathore, "Efficient Capabilities of Processing of Big Data using Hadoop Map Reduce", International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 6, June 2014

[6] Anuja Pandit, Amruta Deshpande, Prajakta Karmarkar, "Log Mining Based on Hadoop's Map and Reduce Technique", International Journal on Computer Science and Engineering (IJCSE) Vol. 5 No. 04 Apr 2013

[7] Yahoo Hadoop's tutorial, https://developer.yahoo.com/hadoop/tutorial/