A METHODOLOGY FOR WEBLOG DATA ANALYSIS USING HADOOP MAP REDUCE AND PIG

BY

DURGA PRASAD P. S. * T. VIVEKANANDAN ** A. SRINIVASAN ***

** PG Scholar, Department of Computer Science and Engineering, SITAMS, Chittor, Andhra Pradesh, India.*
- Associate Professor, Department of Computer Science and Engineering, SITAMS, Chittor, Andhra Pradesh, India.*

ABSTRACT

In the recent time, world is severely facing the problem related to the data storage and processing. Especially, the size of weblog data is exponentially increasing in terms of petabytes and zettabytes. The dependency of weblog data shows its importance on the users' actions on web. To solve and improve the business in all aspects, web data is prominent and hence it is vital. The traditional data management system is not adequate to handle the data in very large size. The Map Reduce programming approach is introduced to deal with the large data processing. In this paper, the authors have proposed a large scale data processing system for analysing web log data through MapReduce programming in Hadoop framework using Pig script. The experimental results show the processing time for classification of different status code in the web log data is efficient, than the traditional techniques.

Keywords: Hadoop, Embedded Pig, MapReduce, Web Log Data.

INTRODUCTION:

The advancements in various technologies enforce to generate a large amount of data in a rapid rate. The amount of data generated is growing rapidly, hence it becomes essential to measure, gather and analyse. World Wide Web (WWW) has been centralized for advancements in Information technology and is a primary medium for billions of people across the world.

Even though various algorithms and data replication [10], [17], were applied in the distributed computing environment, it is still inactive. To overcome the above issue, Hadoop MapReduce programming is used for distributed storage and processing for managing the large scale data. In order to analyse the user activities in webserver, log files are used.

In this paper, the authors present the analysis of web log files based on the status code. Here they have used the SASK web log data [8] for analysis. In recent time, the rapid development in cloud and web2.0 technologies have made various kinds of technologies to get evolved. The dynamic environment in web2.0 allows the user to access, share, deploy and collaborate with various web applications through online. The web apps running through online generates extremely large amount of data. According to [10] Solon Digital Sky surveyed Walmart handles nearly 1 million customer transactions every hour and estimated to generate more than 2.5 peta bytes of data.

The activities of users' information with web server is collected and maintained as web server log files. With the large amount of data available on web, it has enforced to identify effective retrieving mechanisms and user behaviour. In order to improve the web services, it becomes important to detect anomalies that exists in the web transactions. Web server handles Giga bytes to Tera bytes [11] of data every day, hence storing and processing huge volumes of data files with traditional computing programmes is inadequate. The HDFS (Hadoop Distributed File System) and MapReduce programming in Hadoop environment manages to handle the large volume of data in distributed and parallel manner. This paper presents an effective data analysis on web log files for improving web transactions.

1. Related Works

Siddharth Adhikari, Devesh Saraf, Mahesh Revanwar, and Nikhil Ankam [1] have worked on "Analysis of data log and statistics report generation using Hadoop" on web log data. They classified the data based on 6 error status code and generated the report using Tableau.

For effective analysis, pre-processing [2] of web logs files is an important phase in web usage mining. They proposed a pre-processing approach for the task of session identification from web log files. Further, they produced different statistical information such as total unique IPs, total unique pages, total sessions, session length and the frequency visited pages.

According to Natheer Khasawneh and Chein-Chung Chan's [3] opinion, the major role of processing the web logs is to find the most frequently visited user and session identification. And they suggested a trival algorithm and an active user based user identification algorithm to different sessions.

In general, log files are semi-structured. Murat Ali Bayir and Ismail Hakki Toroslu [4] proposed a framework, Smart-SRA to find user sessions and frequent navigation patterns. Smart-SRA uses Apriori algorithm to utilize the structure of web graph. The framework also uses the MapReduce to process server logs of multiple sites simultaneously.

P Srinivasa Rao, K Thammi Reddy and MHM Krishna Prasad, [5] presented the detection of anomalies in user access by extracting IP addresses and URL. They have implemented in a cluster setup with 4 machines to remove redundant IP addresses.

Sayalee Narkhede and Tripti Baraskar [6] explained the Hadoop MapReduce log analyser, which identifies the individual fields by pre-processing the web log files. Further, they presented the removal of redundant web log entries that evaluates the total hits from each city.

Ramesh Rajamanikkam and Kavitha [7] introduced an algorithm for path extraction using sequential pattern clustering method. The above algorithm extracts the data according to the knowledge by the web pages which are frequently used by the user. In addition, path extraction method is to extract the content path and transaction path of the user.

2. Preliminaries

2.1 Hadoop and HDFS

The massive amount of data generated globally needs a powerful computing environment to minimize the time and improve the performance of data execution. When the size of the data is very large, it becomes difficult to manage in a single storage. Hadoop framework [12] has been considered as a powerful tool to handle very large size of data and fault-tolerance. Hadoop supports [13] reliable, vast scale data processing and storage, using HDFS. The MapReduce programming model [9] enables to process vast amount of data in parallel using low end computer systems. MapReduce technique is composed of map and reduce procedures.

2.2 Pig

Pig is a scripting language officially called as PIG Latin [15]. It is scripted to generate the MapReduce job by simply writing with predefined procedures rather than writing the number of lines of code as in java [16]. It also promotes the research type of applications in single node machines to analyse the big data easily.

2.3 WebLog Files

Now-a-days usage of internet is growing in an exponential pattern in a rapid way. It becomes important to analyse the usage pattern of web and its status. Web log files are computer generated files maintained in a webserver [2,3,4]. Web log files consists of various information such as IP address, username, password, timestamp, file location, type of http request, status code, number of bytes downloaded and web browser version.

3. Proposed Model

With the development in cloud computing and Web2.0, the massive amount of data is generated exponentially. It becomes difficult to manage data in large size and also to manage with traditional data processing systems. In this work, the authors aim to develop a large scale processing system using Hadoop Map Reduce programming for handling massive volume of web log file. Web log data gathered from NASA [8] webserver is used for this work. The NASA Saskatchewan data contains

noisy and inconsistent values which may provide unreliable results. Pre-processing is carried out to remove the incomplete and noisy data. Further, weblog data is uploaded to HDFS and given to MapReduce job.

In Map Phase:

In this phase, status code is taken as key and the remaining as values, and then IP address as key and the rest will be values. They have undergone to the mapping function based on the specified keys and values.

In Reduce Phase:

Here, the values are reduced upon IP address ID and status, and then the results are produced by unique identifier values.

3.1 Architecture

In this system, the Large Scale data processing system has been introduced in which it enables the user to interact the system through GUI. It interacts with server and processes further as depicted in Figure 1.

3.2 Algorithm

For Classification of different status code.

1. START
2. INPUT: NASA's web log data.
3. OUTPUT: Analysis report on different status code in the web log file.
4. SET path← location of the input file in the local file system.
5. SET status_code←{401,404,203,500,.....}
6. call pig_servlet (PigServer ps, String input)
7. Alias1 = LOAD data as PigStorage(');
8. If($value = status_code)
   then
9. Alias2 = set_of_records;
   End if
10. alias3 = $IP_address;
11. Store ("alias","output_path);
12. End pig_servlet
13. End

4. Experimental Results

The authors experimented the above with hardware configuration of Intel i3 processor with 2.30 GHz, 4GB RAM, 64-bit OS with 64-bit processor. All the experiments were performed in Ubuntu 14.04 with Hadoop 1.2.1, JDK 1.8, pig-0.12.1, and tomcat server8.0. Log data from different webservers such as, Calgary-HTTP, Clark Net-HTTP, NASA-HTTP, and Saskatchewan-HTTP are used for the experiment. The dataset is processed using Pig Latin scripts and analysis about the log data is generated. Here, the user interface is designed to upload the data and the type of status

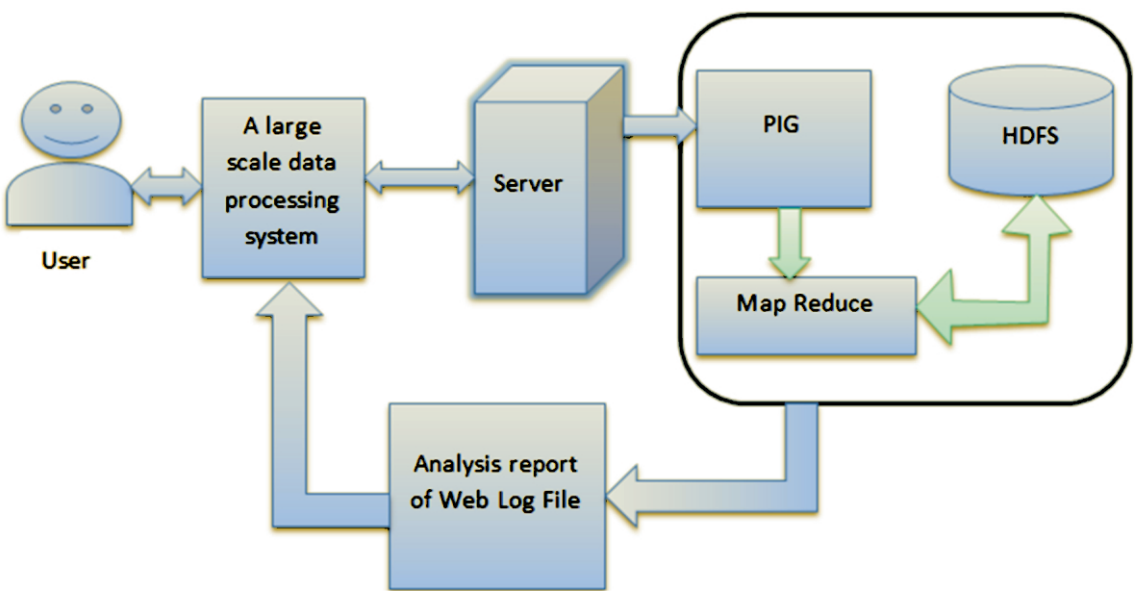

Figure 1. Architecture of the System

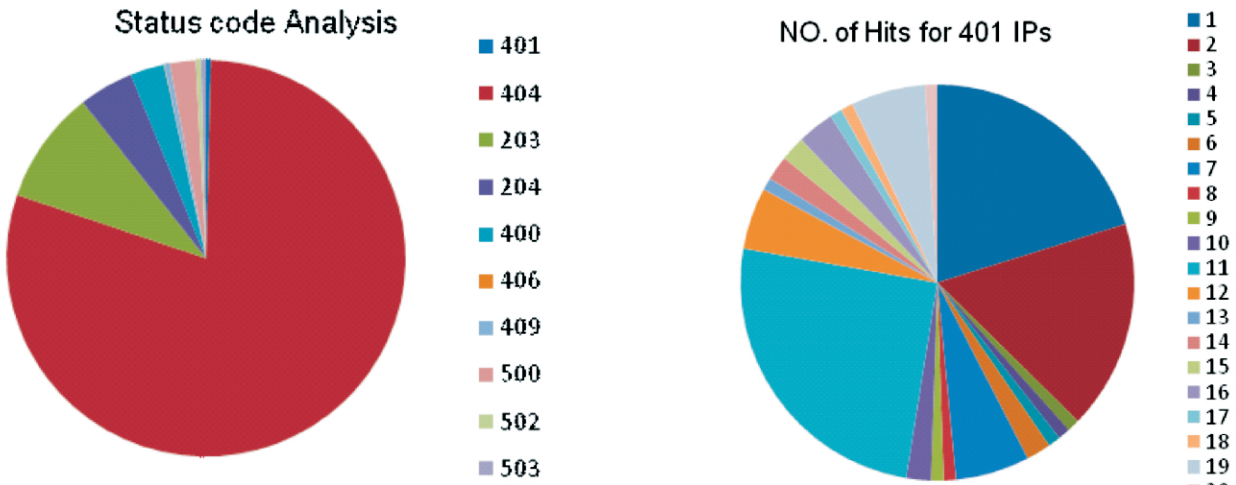

Figure 2 (a) Status Code Analysis on NASA Saskatchewan-HTTP Data, (b) No. of Hits for 401 Ips

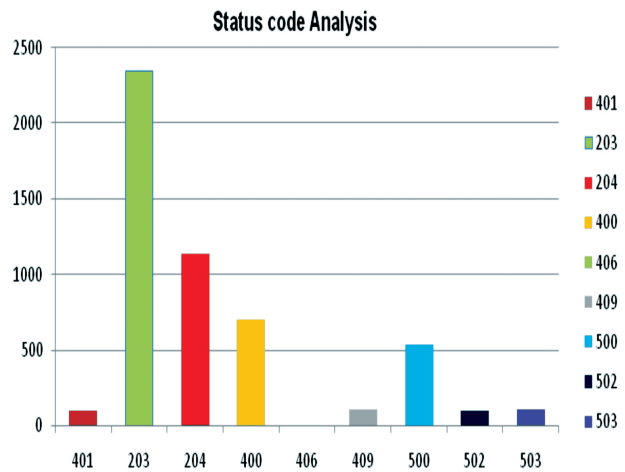

Figure 3. Bar Chart for Status Code Analysis

classification. The statistics of generated results are summarized in Figure 2.

In the pie charts, Figure 2 (a) shows the total status codes hit all IP addresses and Figure 2 (b) shows the single (401) status code classification with unique IPs count. The serial numbers specified in Figure 2 (b) is the unique IP address values in web log data. In the same way, Figure 3 depicts the result analysis of each status codes recorded in the web log data.

Conclusion

With the large amount of data generated in the web, an efficient system is required for processing and analysing the data. In this paper, the authors have designed a system for processing and analysing web log files using Hadoop MapReduce programming. From the results, they conclude that managing huge volumes of data using Distributed processing greatly reduces the execution time. It becomes important for managing and analysing the dynamic patterns of web log files. In their experiment, they analysed the frequency of different status codes. Further, they also analysed the number of occurrences of each status code in the web log file. In future, they have planned to experiment the large sized web log file in the Hadoop cluster environment.

References

[1]. Siddharth Adhikari, Devesh Saraf, Mahesh Revanwar, and Nikhil Ankam, (2014). "Analysis of Log Data and Statistics Report Generation using Hadoop". In *IJIRCCE,* Vol. 2, No. 4.

[2]. Thanakorn Pamutha, Siriporn Chimphlee and Chom Kimpan, (2012). "Data Pre-processing on Web Server Log Files for Mining Users Access Patterns". *International Journal of Research and Reviews in Wireless Communications,* Vol. 2, No. 2, ISSN: 2046-6447.

[3]. Natheer Khasawneh and Chien-Chung Chan, (2006). "Active User-Based and Ontology-Based Web Log Data Pre-processing for Web Usage Mining". *Proceedings of the IEEE International Conference on Web Intelligence*.

[4]. Murat Ali Bayir, and Ismail Hakki Toroslu, (2009). "Smart Miner: A New Framework for Mining Large Scale

Web Usage Data". *WWW 2009, ACM,* Madrid, Spain, 978-1-60558-487-4/09/04, April 20–24, 2009.

[5]. P. Srinivasa Rao, K. Thammi Reddy and MHM. Krishna Prasad, (2013). "A Novel and Efficient Method for Protecting Internet Usage from Unauthorized Access using MapReduce". *International Journal of Information Technology and Computer Science,* Vol. 3, pp. 49-55.

[6]. Sayalee Narkhede and Tripti Baraskar, (2013). "HMR Log Analyzer: Analyze Web Application Logs over Hadoop MapReduce". *International Journal of UbiComp (IJU),* Vol. 4, No. 3.

[7]. Ramesh Rajamanickam and C. Kavitha, (2013). "Fast Real Time Analysis of Web Server Massive Log Files using an Improved Web Mining Architecture". *Journal of Computer Science,* Vol. 9, No. 6, pp. 771-779, ISSN: 1549-3636.

[8]. NASA-HTTP, Web Logs Files. Retrieved from Http://ita.ee.lbl.gov/html/contrib/Saskatchawan-HTTP. html

[9]. Tom White, (2015). *Hadoop: The Definitive Guide,* Fourth Edition, ISBN: 978-1-449-31152-0 1327616795, 2015.

[10]. Naseera Shaik, T. Vivekanandan and K V Madhu Murthy, (2008). "Data Replication using Experience Based Trust in a Data Grid Environment". *Distributed Computing and Internet Technology, Springer,* Berlin, Heidelberg, Vol. 5375, pp. 39-50.

[11]. Economist, (2016). *Data, Data Everywhere.* Retrieved from http://www.economist.com /node, on 13th July 2016.

[12]. Hadoop, (2016). *Welcome to Apache™ Hadoop.* Retrieved from https://hadoop.apache.org.

[13]. Doug Cutting, "Hadoop Overview". Retrieved from http://research. yahoo.com/node/2116

[14]. "PIG", https://pig.apache.org.

[15]. Alan Gates, (2011). *Programming PIG,* O'reilly- First Edition.

[16]. Naseera Shaik, T. Vivekanandan and K V Madhu Murthy, (2008). "Trust Based Data Replication Strategy in a Data Grid Environment". In *Proceedings of International Conference on Information processing (ICIP),* Banglore.

ABOUT THE AUTHORS

P.S. Durga Prasad is a Postgraduation Scholar in the Department of Computer Science and Engineering at SITAMS, Chittoor, India. He is also certified for (CGI) Quality Theorem for Manual Testing. He has academic experiences and a life member in International Association of Engineers. His research interest includes Big Data Analytics, Network Security, and Data Mining.

T. Vivekanandan is currently working as an Associate Professor in the Department of Computer Science and Engineering at SITAMS, Chittoor, India. He has published papers in a reputed National and International Conferences and Journals. He is also a Life Member in the Indian Society for Technical Society (ISTE) and Computer Society of India (CSI). His research interest includes Big Data Analytics, High Performance Computing (HPC), & Cloud and Grid Computing.

A. Srinivasan is currently working as an Associate Professor in the Department of Computer Science and Engineering at SITAMS, Chittoor, India. He has more than 11 years of academic experience, besides he is a life member in the Indian Society for Technical Society (ISTE) and Computer Society of India (CSI). His research interest includes Network Security, Mobile Social Networks, etc.