Web Server Log Processing using Hadoop

[1]Pushkar Gavandi, [2]Bhavika Gori, [3]Smruti Ingawale, [4]Seema Yadav
[1,2,3]UG Student, [4]Asst. Professor
[1,2,3,4]K. J. Somaiya Institute of Engineering & IT, Sion, Mumbai, Maharashtra, India.
[1]pushkar.g@somaiya.edu, [2]bhavika.gori@somaiya.edu, [3]smruti.i@somaiya.edu,
[4]s.yadav@somaiya.edu

Abstract Big Data is an emerging growing dataset beyond the ability of a traditional database tool. Hadoop rides the big data where the massive quantity of information is processed using cluster of commodity hardware. A web server log file is a text file that is written as activity is generated by the web server. Log files collect a variety of data about information requests to your web server. Server logs act as a visitor sign-in sheet. Server log files can give information about what pages get the most and the least traffic? What sites refer visitors to your site? What pages that your visitors view and the browsers and operating systems used to access your site. The web server log processing has bright, vibrant scope in the field of information technology. The web server log processing can be so enhanced & expanded that it can be used in various spectra's & fields which are handling enormous amount of data on daily basis. It is reliable, fast and scalable approach for handling large numbers of logs and to transform log data into statistical data and generate reports accordingly.

Keywords — BigData, Cloud Computing, MapReduce, Hadoop, Log File, Ecosystem..

I. INTRODUCTION

Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. Due to the advancement of new technologies, devices, and communication means like social networking sites, the amount of data produced by mankind is growing rapidly every year. Website statistics are based on server logs.

IT organizations analyze server logs to answer questions about security and compliance. A server log is a simple text file which records activity on the server. Computer generated logs that capture data on the operations of a network. Useful for managing network operations, especially for security andregulatory compliance. There are several types of server log — website owners are especially interested in access logs which record hits and related information. These logs are in large amount thus resulting collection of large amount of data i.e Big Data. Big data means really a big data, it is a collection of large datasets that cannot be processed using traditional computing techniques. It includes huge volume, high velocity, and extensible variety of data. This data can be in structured, semi structured or in unstructured form.

In this paper, we will focus on a network security use case. Specifically, we will look at how Apache Hadoop can help the administrator of a large enterprise network diagnose and respond to a distributed denial-of-service attack.

II. EXISTING SYSTEM

The current processing of log files goes through ordinary sequential ways in order to perform preprocessing, session

identification and user identification. The non-Hadoop approach loads the log file dataset, to process each line one after another. The log field is then identified by splitting the data and by storing it in an array list. The preprocessed log field is stored in the form of hash table, with key and value pairs, where key is the month and value is the integer representing the month.

In existing system work is possible to run only on single computer with a single java virtual machine (JVM). A JVM has the ability to handle a dataset based on RAM i.e. if the RAM is of 2GB then a JVM can process dataset of only 1GB. Processing of log files greater than 1GB becomes hectic. The non-Hadoop approach is performed on java 1.6 with single JVM. Although batch processing can be found in these single-processor programs, there are problems in processing due to limited capabilities. Therefore, it is necessary to use parallel processing approach to work effectively on massive amount of large datasets.

III. PROPOSED SYSTEM

Proposed solution is to analyze web log generated by Apache Web Server. This is helpful for statistical analysis. The size of web log can range anywhere from a few KB to hundreds of GB. Proposed mechanism design solution that based on different dimensions such as timestamp, browser, and country. Based on these dimension, we can extract pattern and information out of these log and provides vital bits of information. The technologies used are Apache Hadoop framework, Apache flume etc.Use Hadoop Cluster (Gen1). Content will be created by multiple Web servers and logged in local hard discs. Proposed system uses four node environments where data is manually stored in local hard disk in local machine. This log data will then be transferred to HDFS using FLUME framework. FLUME has agents running on Web servers. This log data is processed by MapReduce to produce Comma Separated Values i.e. CSV.Find the areas where there exist errors or warnings in the server. Also find the spammer IPs in the web application.Then we use Excel or similar software to produce statistical information and generate reports.

Table 1: Comparison between existing system and proposed system

Feature	Existing System	Proposed System
Storage Capacity	Less	More
Processing Speed	Slow	Fast
Reliability	Less	More
Data Availability	Less	High
Data Location	Centralized	Physically highly Distributed
Data Structure	Pre-defined Structure	Structured, semi-structured or Unstructured

IV. ARCHITECTURE

Apache Web Server - Apache Hadoop is an excellent framework for processing, storing and analyzing large volumes of unstructured data - aka Big Data.

Server logs - Web server logs are semi structured files generated by the computer in large volume, usually of text file.log files contains or collect variety of information such as Date, Time, Client's IP address, Service name, Server IP, etc.

FLUME - Flume is a framework for populating Hadoop with data. Agents are populated throughout ones IT infrastructure – inside web servers, application servers and mobile devices, for example – to collect data and integrate it into Hadoop.

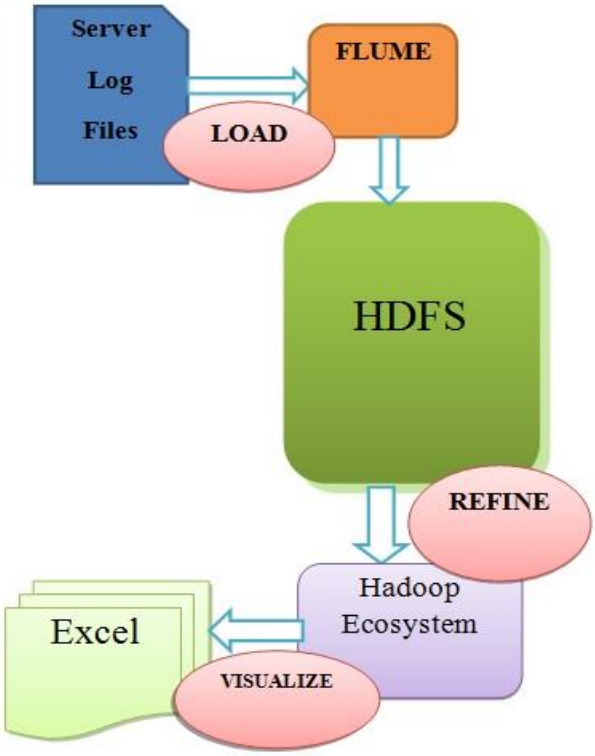

Fig.1 Proposed System Architecture

Hadoop Distributed File System (HDFS) - The storage layer of Hadoop is a distributed, scalable, Java-based file system adept at storing large volumes of unstructured data.

Hadoop Ecosystem–

i. **MapReduce -** MapReduce is a software framework that serves as the compute layer of Hadoop. MapReduce jobs are divided into two (obviously named) parts. The "Map" function divides a query into multiple parts and processes data at the node level. The "Reduce" function aggregates the results of the "Map" function to determine the "answer" to the query.

ii. **Pig** – It provides an engine for executing data flows in parallel on Hadoop. It includes a language, Pig Latin, for expressing these data flows. Pig Latin includes operators for many of the traditional data operations (join, sort, filter, etc.), as well as the ability for users to develop their own functions for reading, processing, and writing data. Pig runs on Hadoop. It makes use of both the Hadoop Distributed File System, HDFS, and Hadoop's processing system, MapReduce.

Microsoft Excel - Software that allows users to organize, format, and calculate data with formulas using a spreadsheet system broken up by rows and columns. It features the ability to perform basic calculations, use graphing tools, create pivot tables and create macro programming language. Excel has the same basic features as every spreadsheet, which use a collection of cells arranged into rows and columns to organize data manipulation. They also display data as charts, histograms and line graphs. Excel permits users to section data so as to view various factors from a different perspective.

V. CONCLUSION

In this study, we discuss the Web Server Log Processing that uses Hadoop for improving the performance of a database management system (DBMS)-based analysis service system that processes big data. Traditional log processing systems are not suitable for processing big data and providing service because of their disadvantage in consuming more time for processing and analyzing. We introduced a distributed parallel platform, Hadoop ecosystem, for improving the performance of the system by minimizing the processing time in analyzing big data.

This study explained the method of changing an existing log analysis service system to a distributed parallel-based environment system to address the problems encountered during the processing of big data. We optimized the system by using Hadoop ecosystem to improve the performance while processing big data.

Proposed system is useful for analyzing errors in sites, servers and finding spammer ip's. It is more reliable, fast and scalable approach for handling large numbers of logs. Processing of web server logs can help in analyzing traffic on various sites and help developers of website to make

changes accordingly as per results of analysis. It includes transformation of log data to statistical data and generation of reports accordingly.

Web Server Log Processing has bright, vibrant scope in the field of information technology. IT organizations analyze server logs to answer questions about security and compliance. Proposed system will focus on a network security use case. Specifically, we will look at how Apache Hadoop can help the administrator of a large enterprise network diagnose and respond to a distributed denial-of-service attack.

ACKNOWLEDGMENT

This project consumed huge amount of work, research and dedication. It would not have been possible if we did not have a support of many individuals and organizations. Therefore we would like to extend our sincere gratitude to all of them.

The authors are thankful to Principal Dr. Dilip R. Pangavhane and Head of Department of Information Technology, Prof. Uday U. Rote who has provided us with all the facilities to conduct our project work with immense co-operation & inspiration. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in this project.

In addition, we are highly indebted to Prof. SeemaYadav, Faculty of Information Technology, K. J. Somaiya Institute of Engineering and Information Technology, Sion for the guidance and constant supervision as well as for providing necessary information regarding the project and also for the support in completing the project.

REFERENCES

[1] Extracting WebLog of Siam University for Learning User Behavior onMapReduce -2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012) .

[2] Mining of Web Server Logs in a Distributed Cluster Using Big Data Technologies -(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 1, 2014.

[3] Tom White: Hadoop, "The Definitive Guide (1st edn.)", O'Reilly Media, Inc., United States of America, 2009.

[4]Hadoop MapReduce Change Log. Release0.22.1 – Unreleased. http://hadoop.apache.org/mapreduce/docs/r0.22.0/changes.html >, Accepted 02012012.

[5] Web Log Analysis for Security Compliance Using Big Data- Volume 5, Issue 3, March 2015 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[6]http://www.orzota.com/wp-content/uploads/2014/04loganalysis-paper.pdf

[7] Statistical Analysis of Web Server Logs Using Apache Hive in Hadoop Framework-International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 5, May 2015

[8] Arindam Banerjee and JoydeepGhosh, "Clickstream Clustering using Weighted Longest Common Subsequences", Int'l Conf the Web Mining Workshop at the 1st SIAM Conference on Data Mining, Chicago, 2001.

[9] J. Dean.S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Int'l Conf of Operating Systems Design and Implementation (OSID), San Francisco, CA, pp. 137-150, 2004.

[10] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar, "Reductions in streaming algorithms, with an application to counting triangles in graphs", Int'l Conf. 13th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 623–632, 2002.