Journal of Engineering and Applied Sciences
SINCE 2000
MEDWELL PUBLICATIONS

Generating Analytics from Web LOG

[1]Vempaty Prashanthi, [2]Srinivas Kanakala and [2]Subhash Parimalla
[1]*Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India*
[2]*VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India*

Key words: Big data, HDD'S, web log, MapReduce, Hadoop

Corresponding Author:
Vempaty Prashanthi
Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India

Page No.: 3503-3508
Volume: 15, Issue 20, 2020
ISSN: 1816-949x
Journal of Engineering and Applied Sciences

Abstract: Modern engineering incorporates clevertechnologies in all factors of our lives. Smart technologies are generating terra bytes of log messages every day to record their status. It is crucial to research these log messages and present usable records (e.g., patterns) to directors, so as to manipulate and reveal those technology. Patterns minimally represent large corporations of log messages and enable the administrators to do further analysis, along with anomaly detection and event prediction. Although, patterns exist typically in automatic log messages, spotting them in large set of log messages from heterogeneous resources without any prior information is a widespread undertaking. We propose a big data using Hadoop that extracts high pleasant styles for a given set of log messages. Our approach is fast, memory efficient, accurate and scalable. Hadoop is implemented in map-reduce framework for disbursed platforms to procedure hundreds of thousands of log messages in seconds. It is a robust approach that works for heterogeneous log messages generated in a wide style of systems. Our technique exploits algorithmic techniques to limit the computational over-head based totally on the truth that log messages are continually routinely generated. We examine the performance of Log-Mine on hugeunits of log messages generated in commercial applications. It has efficiently generated styles which might be as exact as the styles generated by genuine and un-scalable method whilst achieving a 500 per speedup.

INTRODUCTION

In the present world data analysis is a challenge in the era of varied inter disciplines though there is a specialization in respective disciplines. In other words effective data analytics helps for analyzing the data for any business system. But it is big data which helps and accelerates the process of analysis of data paving way for the success of any business intellectual system. With the expansion of the industry the data of the industry also expands. Then, it is increasingly difficult to handle the huge amount of data that's get generated no

matter what the business is like range of fields from social media to finance, flight data, environment and health.

The challenge of big data is how to use it to create something which is valuable to the user. It can be gathered, stored, processed and analyzed it to turn the raw data to support decision making. Big data is depicted in the form of a case study for analyzing web log data using Hadoop. These log files are very large and can have a complex structure. Although, the process of generating log files is straight forward but these log files are more error prone. This often leads to a situation when these log files generated continuously and occupy valuable storage on the storage devices but nobody uses them and utilizes enclosed information. This can analyze different kinds of log files such as E-mail logs, web logs, Firewalls log, server logs, call data logs etc. It can also be used in the concept of network coding[1, 2]. Clusters using network coding[3], to find energy efficient path in network[4].

Big data is a term that is used for storing and processing large volumes of data (structured and unstructured) which helps a business in a regular or daily basis. Big data mainly consists of 5 v's-volume, velocity, variety, veracity and value. Big data is not concerned with the amount of data but how efficiently it can processes that data and extract the required information. That extracted information or insights can help the organizations in better decision making and management.

When a web user surfs a specific web page or website, the server records the small amounts of it within the web access log format. In the web access request log you will see the types of files users measure accessing the situation from wherever the request has been created and alternative data like what browsers they are using and device access points. An access log could be a list of all the entries users have requested from an internet website. Such log files measure semi-structured data that is so hard to store, method and analyse visitors or accessed person's previous information from a warehouse system.

Literature review: By Keogh and Kasetty[5] and Ankerst *et al*.[6] the researchers have clearly reasoned why MapReduce is the choice for log processing rather than RDBMS. Researchers have showed various join processing techniques for log data in map-reduce framework. This research, along with Blanas *et al*.[7], greatly inspired us to attempt clustering on massive log data. In by Ding and Zhou[8] and Eltahir and Dafa[9] the authors describe a unified logging infrastructure for heterogeneous applications. Our framework is well suited to work on top of both of these infrastructures with minimal modification. In HPC (High Performance Computing), logs have been used to identify failures and troubleshoot the failures in large scale systems[10]. Such tools majorly focus on categorizing archived log messages into sequence of failure events and use the sequence to identify root cause of a problem.

An automated log analyzer must have one component to recognize patterns from log messages and another component to match these patterns with the inflow of log messages to identify events and anomalies[11, 12]. Such a log message analyzer must have the following desirable properties.

No-supervision: The pattern recognizer needs to be working from the scratch without any prior knowledge or human supervision. For a new log message format, the pattern recognizer should not require an input from the administrator.

Heterogeneity: There can be log messages generated from different applications and systems. Each system may generate log messages in multiple formats. An automated recognizer must find all formats of the log messages irrespective of their origins.

Efficiency: IoT-like systems generate millions of log messages every day. The log processing should be done so, efficiently that the processing rate is always faster than the log generation rate.

Scalability: Pattern recognizer must be able to process massive batches of log messages to maintain a cur- rent set of patterns without incurring CPU and memory bottlenecks (Fig. 1).

The existing system[13] uses "Relational Data Base Management System" (RDBMS). A RDBMS is a type of Database Management System (DBMS) that stores the data or tuples in the form of rows and columns. Relational databases are more powerful as they require less number of assumptions to know how the data can be drawn out from the specific database. As a result, the same database can be viewed in many different ways or in different perspectives.

The RDBMS had been the one of the best solution for all the things the database requires. RDBMS uses Structured Query Language (SQL) to store, query, update and delete the contents in that specific database. However, the volume and velocity of this raw data have changed drastically in the past few years. It's continuously increasing every minute by minute.

Limitations of using RDBMS for analysis: The size of data has been increased rapidly to the range of pet bytes where one pet byte = 1.024 terabytes in number. Here, the RDBMS cannot handle large amounts of data. To address this issue, RDBMS added more number of CPUs to the DBMS to increase its capability.

Another limitation is that the majority of the data that comes from social media, audio, video is in a semi-structured or unstructured format. However, the RDBMS cant process these unstructured data. To handle

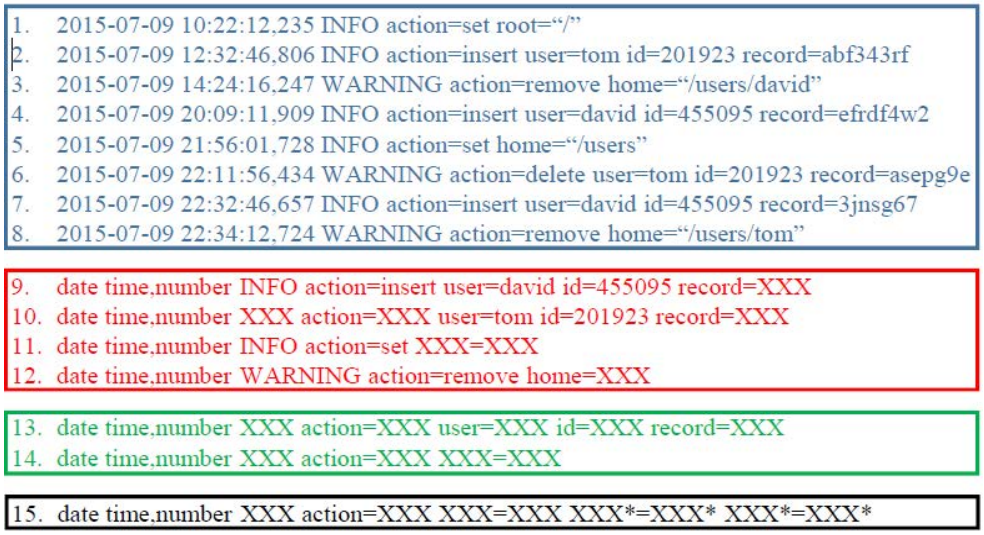

Fig. 1: Extracting log patterns for a given set of logs

such a huge amount of data high velocity is required. RDBMS doesn't support the high velocity data because it is designed not for the rapid growth but for the study data growth. Even if RDBMS tries to store and process these data, it may turn out to be much expensive.

Proposed system: The proposed gem is by using "HADOOP Ecosystems". Big data is a term used for large data or datasets that are so large that normal processing applications cannot handle it. Big data is a phrase used to mean a massive volume of all structured, semi structured and unstructured data. The data is generating at a rapid rate and in different number of formats that we cannot handle them. The social networking and mobile are the one contributing the highest amount of data generation. These above factors have progressed towards the term "big data". With the fast emergence of these data, traditional data processing techniques are unable to catch up with them. These factors have contributed for the adoption of big data. To know why big data is much better than RDBMS for data analytics, we have to know the advantages of big data for data analytics.

Advantages of using Hadoop for analytics:

- Identify the main causes of failure
- Processing large volumes of data and extracting insights
- Understanding the usage of insights developed
- Understanding the usage of marketing process through data driven process
- Offering discounts or offers to the customers based on their buying habits
- Improving the relationships between customer and vendor
- Re evaluating the risk associated with that business
- Enhancing the experience of customer
- Enhancing the interactions or values for both online or offline customers

System architecture: MapReduce has become the most frequently used framework for processing of huge amounts of structured or unstructured data stored in Hadoop cluster. It was designed by Google to provide the correspondence and reduce the fault tolerance of data. MapReduce processes the large data in the form of key value pair. We can choose the key value pair based on our choice. These key value pairs are used for MapReduce process as our system is not static. For static systems columns are used for analysing the data. MapReduce API will furnish the subsequent options like instruction execution, parallel processing of huge amounts of data and high availability. MapReduce work flow undergoes different stages which stores the output in HDFS with replications at the end. A Job tracker checks all the map reduce jobs which are working on Hadoop cluster. A Job tracker plays a crucial role in scheduling the jobs and keeps track of every map and reduces jobs. MapReduce contains two processing stages map stage and reduce stage. Between these two stages there is one more stage called intermediate stage which takes the input from the mapper perform shuffling. Sorting and combining (Fig. 2). Three phases exists in this system. Mapper phase 2. Intermediate phase 3. Reducer phase.

Mapper phase: Mapper phase gets the input values from the record reader. The record reader is responsible to send the key value pair to the mapper. The input received by

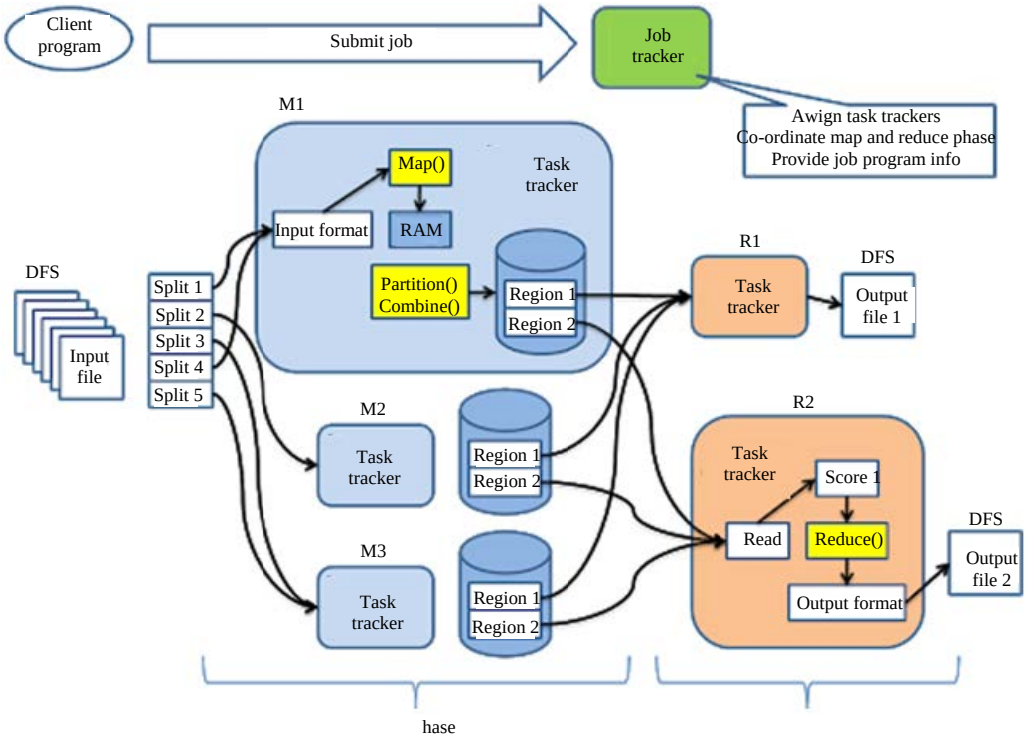

Fig. 2: MapReduce frame work

the mapper is spit into key value pairs. Based upon the keys and partition constraints input is distributed to the specified reducer. The output generated is also a set of key value pairs. This is termed as intermediate key value pair.

Intermediate phase: This phase comes in between the map and reduce phases. In this phase many operations are done based on the results required. In this phase, the same key values from different mappers will get into one mapper. Operations like shuffling, sorting and combing are done in this phase. It uses the Round Robin algorithm to write the intermediate key values pairs into the local disk.

Reducer phase: This is the second stage of the MapReduce data flow. In this phase, it receives the input from the practitioner and combiner. The reducer's logic will begin with the operations performed by the mapper. It produces the output files like part files which contains the actual output of the analysed data. Each time when the job is run reducer shows the number of reducers needed for the job for execution. As the reducer performs parallel processing and therefore, the performance and throughput of system is increased.

MATERIALS AND METHODS

Hadoop Distributed File System (HDFS) is used to store huge data sets or data and stream these informational indexes at a very high speed transfer to other applications. HDFS facilitates easy access of data. As a single machine cannot hold gigantic or large information, the records of this data are stored in different machines. This data is stored in a redundant style to safe guard the data for any attacks or occurrence of disappointments. HDFS likewise makes applications accessible to parallel processing (Fig. 3).

Implementation:

- Step 1: Create a webpage and host the webpage using bluehost hosting service
- Step 2: Extract the web log data from the webpage
- Step 3; Process the raw web log data to obtain a cleaned data set (CSV format)
- Step 4: Create a new directory with same name weblog analysis in the cluster
- Step 5: Write the MapReduce program in Eclipse
- Step 6: Create a jar file and copy the jar file to local edge node using WinScp
- Step 7: Login into the cluster using putty and copy the input file from local to cluster
- Step 8: Run the mapreduce program
- Step 9: Result is seen through command interface

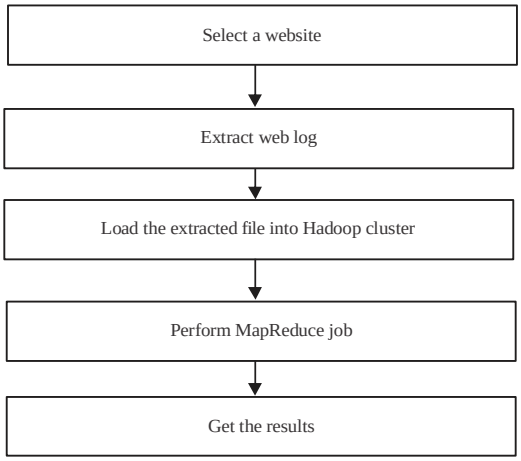

Fig. 3: Process flow

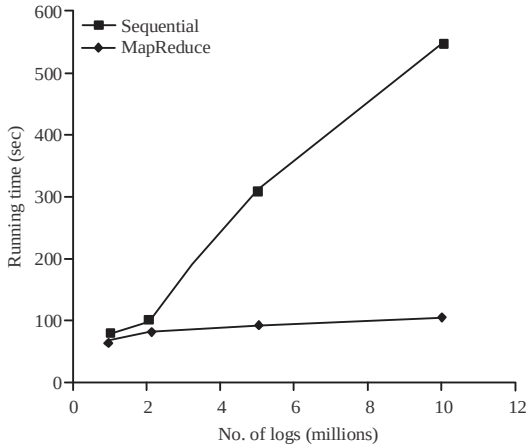

Fig. 4: Comparison of running times WRT No. of logs

RESULTS AND DISCUSSION

In this experiment, we compare sequential and map reduce technologies. We generate synthetic data by changing number of log entries (10 million default) and number of patterns (1500 default). We change the number of map-reduce workers (8 default) to understand scalability. Each worker has 1 GB of memory and a single-core CPU. As shown in Fig. 4, the execution time of the MapReduce implementation grows slowly compared to the growth of the sequential implementation. MapReduce implementation reaches up to 5 X speed-up by using 8 workers compared to the sequential implementation. Note that we have a fixed number of patterns in this experiment. Our MapReduce implementation can handle millions of logs in few minutes because the number of patterns does not grow at the same rate as the number of logs grows in real world applications. Figure 5 shows that with increasing number of patterns, the execution time of both sequential and

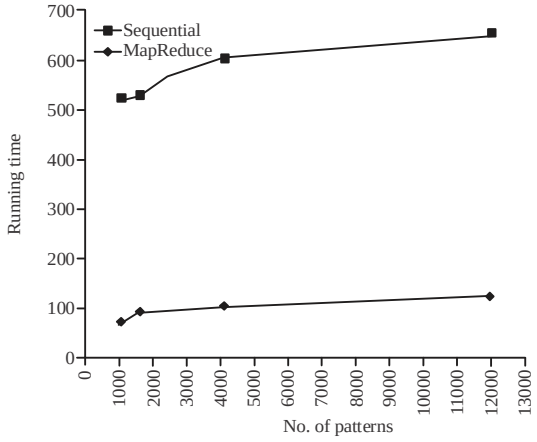

Fig. 5: Comparison of running times WRT No. of patterns

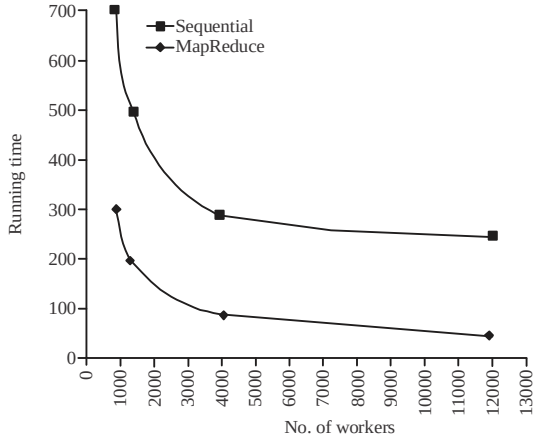

Fig. 6: Comparison of running times WRT No. of workers

MapReduce implementation consistently grows. In Fig. 6, we show that doubling the number of workers reduces the running time by 40%.

CONCLUSION

Nowaday's storing the information has turned into a major issue. Traditional database systems do not support large volumes of data. Therefore, a conventional database system called Hadoop is used for managing huge volumes of data. Number of elements and changed include in enormous information like web based life and cloud based life. These mechanical changes are putting weight on the appropriation of huge information. Big data is vastly improved than RDBMS for information investigation. From the outcomes, we can analyze diverse sorts of IP addresses used time stamps, number of references by every client to the site and locate the top N users. Based on number of bytes used information like most visited user can be identified. False snaps and unknown IP Addresses can be blocked giving a safe domain to clients.

REFERENCES

01. Prashanthi, V. and K. Srinivas, 2019. Identification of opportunities for coding in a network. Int. J. Recent Technol. Eng. (IJRTE.), 7: 140-144.
02. Prashanthi, V., D.S. Babu and C.V. Rao, 2018. Network coding aware routing for efficient communication in mobile ad-hoc networks. Int. J. Eng. Technol., 7: 1474-1481.
03. Kanakala, S., V.R. Ananthula and P. Vempaty, 2014. Energy-efficient cluster based routing protocol in mobile ad hoc networks using network coding. J. Comput. Networks Commun., Vol. 2014, 10.1155/2014/351020
04. Srinivas, K., A.V. Reddy and N. Autha, 2014. Connected dominating set-based broadcasting in mobile ad-hoc networks using network coding. Int. J. Applied Eng. Res., 9: 30279-30301.
05. Keogh, E. and S. Kasetty, 2003. On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining Knowl. Discovery, 7: 349-371.
06. Ankerst, M., M.M. Breunig, H.P. Kriegel and J. Sander, 1999. Optics: Ordering points to identify the clustering structure. ACM SIGMOD Rec., 28: 49-60.
07. Blanas, S., J.M. Patel, V. Ercegovac, J. Rao, E.J. Shekita and Y. Tian, 2010. A comparison of join algorithms for log processing in mapreduce. Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, June 6-11, 2010, ACM, Indianapolis, Indiana, pp: 975-986.
08. Ding, C. and J. Zhou, 2007. Log-based indexing to improve web site search. Proceedings of the 2007 ACM Symposium on Applied Computing, March 11-15, 2007, ACM, Seoul, Korea, pp: 829-833.
09. Eltahir, M.A. and A.A.F. Dafa, 2013. Extracting knowledge from web server logs using web usage mining. Proceedings of the 2013 International Conference on Computing, Electrical and Electronics Engineering (ICCEEE), August 26-28, 2013, IEEE, Khartoum, Sudan, ISBN: 978-1-4673-6231-3, pp: 413-417.
10. Faloutsos, C., M. Ranganathan and Y. Manolopoulos, 1994. Fast subsequence matching in time-series databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, May 24-27, 1994, Minneapolis, MN., USA., pp: 419-429.
11. Lee, G., J. Lin, C. Liu, A. Lorek and D. Ryaboy, 2012a. The united logging infrastructure for data analytics at Twitter. Proc. VLDB Endowment, 5: 1771-1780.
12. Lee, K.H., Y.J. Lee, H. Choi, Y.D. Chung and B. Moon, 2012b. Parallel data processing with MapReduce: A survey. ACM. SIGMOD Rec., 40: 11-20.
13. Rajachandrasekar, R., X. Besseron and D.K. Panda, 2012. Monitoring and predicting hardware failures in HPC clusters with FTB-IPMI. Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, May 21-25, 2012, IEEE, Shanghai, China, pp: 1136-1143.