Analysis of Web Log Files Using Map Reduce Algorithm

J. Ilanchezhian [#1], S. Dhilipkumar [*2], A. Saravanan [*3], P.Shanmugavel [*4],

[#]Assistant Professor, [*]student, Manakula Vinayagar Institute of Technology.
[1] ilanchezhian619@gmail.com,
[2] dhilipkumar1904@gmail.com,
[3] saravanancse565@gmail.com,
[4] shanmugavelbtech@gmail.com.

***Abstract*—Big Data is a collection of a huge and complex data that it becomes extremely tideous to capture, store, process, retrieve and analyze it with the help of on Relational database management tools or traditional data processing techniques. To store and process such a big data hadoop technology is used. Hadoop is an open source framework that allows to store and process big data in distributed environment across clusters of computers using simple programming models. It is intended to scale up from single server to thousands of machines, each offering local computations and storage. In this paper, we propose incremental MapReduce, the most used framework for mining bigdata. Incremental MapReduce 1) performs key-value pair level incremental processing, 2) supports iterative computation, which is widely used in data mining applications. That means incremental MapReduce processes big data in a less time and stores it in a more optimized form.**

***Keywords*— Big data, Hadoop, MapReduce, incremental processing.**

I. INTRODUCTION

Internet usage has increased tremendously and drastically. There are over billions, millions of Internet user at present. The reason for the success of Internet is its simplicity, availability and application can be accessible from anywhere just by using internet browser. The recent availability of huge amounts of data, along with advanced tools of exploratory data analysis, data mining/machine learning and data visualization, offers a whole new way of understanding the world. In order to exploit these huge volumes of data, new techniques and technologies are needed.

A new type of e-infrastructure, the Research Data Infrastructure, must be developed to harness the accumulation of data and knowledge produced by research communities, optimize the data movement across scientific disciplines, enable large increases in multi- and inter- disciplinary science while reducing duplication of effort and resources, and integrating research data with published literature.

Science is a global undertaking and research data are both national and global assets. A seamless infrastructure is needed to facilitate collaborative arrangements necessary for the intellectual and practical challenges the world faces. Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate. Challenges include analysis, capture, data duration, search, sharing, storage, transfer, visualization, and information privacy. The term often refers simply to the use of predictive analytics or other certain advanced methods to extract value from data, and seldom to a particular size of data set. Accuracy in big data may lead to more confident decision making. And better decisions can mean greater operational efficiency, cost reduction and reduced risk.

II. LITERATURE SURVEY

A. *Rise of big data on cloud computing*

To perform complex computations and massive scale operations cloud computing is the powerful technology. Because it eliminates the need to maintain expensive computing hardware, dedicated space and software. Big data generated through cloud computing has been observed.

Ibrahim Abaker Targio Hashem et al. (2015) has reviewed the rise of big data. The definition characteristics and classification of big data along with some discussions on cloud computing are introduced.

B. Incremental MapReduce for Mining Evolving Big Data

Despite the advances in hardware for hand-held mobile devices, resource-intensive applications (e.g., video and image storage and processing or map-reduce type) still remain off bounds since they require large computation and storage capabilities.

Recent research has attempted to address these issues by employing remote servers, such as clouds and peer mobile devices. For mobile devices deployed in dynamic networks (i.e., with frequent topology changes because of node failure/unavailability and mobility as in a mobile cloud), however, challenges of reliability and energy efficiency remain largely unaddressed.

Extensive simulations demonstrate the fault tolerance and energy efficiency performance of their frame work in large scale networks Chien-An chen (2014).

C. Cloud Computing and Big Data

Data analysis is an important functionality in cloud computing which allows a huge amount of data to be processed over very large clusters. MapReduce is recognized as a popular way to handle data in the cloud environment due to its excellent scalability and good fault tolerance.

However, compared to parallel databases, the performance of MapReduce is slower when it is adopted to perform complex data analysis tasks that require the joining of multiple data sets

in order to compute certain aggregates. A common concern is whether MapReduce can be improved to produce a system with both scalability and efficiency.

D. I^2 map reduce approach

Hadoop MapReduce is a leading open source framework that supports the realization of the Big Data revolution and serves as a pioneering platform in ultra large amount of information storing and processing. However, tuning a MapReduce system has become a difficult work because a large number of parameters restrict its performance, many of which are related with shuffle, a complicated phase between map and reduce functions, including sorting, grouping, and HTTP transferring.

During shuffle phase, a large amount of time is consumed on disk I/O with a low speed of data throughput. wang et al (2015) build a mathematical model to judge the computing complexities with the different operating orders within map-side shuffle, so that a faster execution can be achieved through reconfiguring the order of sorting and grouping.

E. Optimizing virtual machine

Big data is getting more attention in today's world. Although MapReduce is successful in processing big data, it has some performance bottlenecks when deployed in cloud. Data locality has an important role among them.

The focus of this paper is on improving data locality in MapReduce cloud by allocating adjacent VMs, for executing MapReduce jobs. Good data locality reduces cross network traffic and hence results in high performance. When a user requests for a set of virtual machines (VMs), VMs are chosen based on their physical distance between other VMs.

F. Big data analytics: a survey

The Goal of data mining is to discover hidden useful information in large databases. Mining frequent patterns from transaction databases is an important problem in data mining. As The database size increases, the computation time and required memory also increase. Base On this, we use the Map Reduce Programming mode which has parallel processing ability to analysis the large Scale network.

All The experiments were taken under hadoop, deployed on a cluster which consists of commodity servers Through Empirical evaluation in various simulation conditions, the proposed algorithms are shown to deliver excellent performance with respect to scalability and execution time Tsai et al. (2015).

G. Big Data Mining using Map Reduce

Big Data is a data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it. Hadoop is the core platform for structuring Big Data, and solves the problem of making it useful for analytics purposes. Hadoop is an open source software project that enables the distributed processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance. This paper presents the survey of big data processing in perspective of hadoop and map reduce suryawonshi and wadne (2014).

H. Scheduling in Hadoop

Hadoop is based on distributed computing having HDFS file system (Hadoop Distributed File System). Hadoop is highly fault tolerant and can be deployed on low cost hardware. Hadoop architecture is cluster based, which is consist of nodes (data note, name node), physically separate to each other, in ideal condition.

The performance of Hadoop can be increased by proper assignment of the tasks in the default scheduler. In Hadoop a program known as map reduce is used to collect data according to query. The research objective is to study and analyses various scheduling techniques, which are used to increase performance in Hadoop Arora and MadhuGoel (2014).

I. Market Basket Analysis Algorithm on Map/Reduce

As the web, social networking, and smartphone application have been popular, the data has grown drastically every day. Thus, such data is called Big Data. Google met Big Data earlier than others and recognized the importance of the storage and computation of Big Data. Thus, Google implemented its parallel computing platform with Map/Reduce approach on Google Distributed File Systems (GFS) in order to compute Big Data.

Map/Reduce motivates to redesign and convert the existing sequential algorithms to Map/Reduce algorithms for Big Data so that the paper presents Market Basket Analysis algorithm with Map/Reduce, one of popular data mining algorithms. It is believed that the operations of distributing, aggregating, and reducing data in the nodes of Map/Reduce should cause the bottleneck.

J. Map/Reduce Design and Implementation

Apriority is one of the key algorithms to generate frequent item sets. Analyzing frequent item set is a crucial step in analyzing structured data and in finding association relationship between items. Hence a distributed environment such as a clustered setup is employed for tackling such scenarios.

Apache Hadoop distribution is one ofthe cluster frameworks in distributed environment that helps by distributing voluminous data across a number of nodes in the framework. This paper focuses on map/reduce design and implementation of Apriori algorithm for structured data analysis Koundinya et al. (2012).

III.SYSTEM DESIGN AND METHODOLOGY

A. *PROPOSED SYSTEM*

As the log files are being continuously produced in various tiers with different types of information, the main challenge is to **store** and process this much data in an efficient manner to produce rich insights into the application and customer behavior. For example, a web server will generate logs of size at least in TB's for a month period.

- We cannot store this much of data into a relational database system. RDBMS systems can be very expensive and cheaper alternatives like MySQL cannot scale to the volume of data that is continuously being added.

- A better solution is to store all the log files in HDFS which stores data on commodity hardware, so it will be cost effective to store huge volumes (TBs or PBs) of log files in HDFS and Hadoop provides MapReduce framework for parallel processing of these files.

In order to solve the problem of existing system i.e. time consumption the proposed MapReduce approach is introduced.

Here the web log file data is taken for experiment and the entire log file is splatted into several blocks and each block are assigned to separate mappers. The mappers convert the input data into key, value (K, V) pairs.

The key(K) is time and the value (V) is the number of occurrence. The data given by the mapper is reduced by the reducer and its outcome is said to be the final outcome which is the number of visit per hour.

Furthermore, this is more time saving process when comparing with the existing Traditional approach. Because the data is spliced into several blocks and each blocks are processed simultaneously. The goal of analytics is to improve the business by gaining knowledge

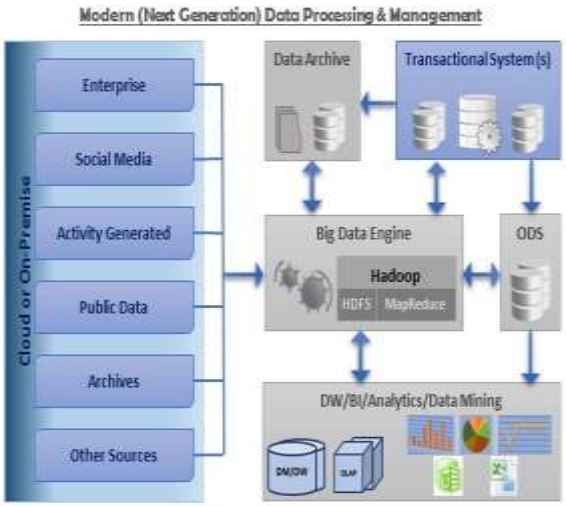


Figure 3.1 System Architecture

Figure 4.1 represents the System Architecture where the big data is split into several blocks in the HDFS. The map reduce operation is performed and will get the final output. The system schematic flow diagram is shown in figure 4.2 which starts from gathering weather sensor data, followed by the splitting of data into several blocks, and then map operation and ends with reduce operation. The final output we get is the average temperature for particular year.

B. *MODULES DESCRIPTION*

To implement this project, the work has been divided into the following modules.
Module 1: single node cluster
Module 2: move Bigdata to HDFS
Module 3: Processing BigData using MapReduce

MODULE 1: single node cluster

It can be useful to operate Application Center on a single server, without the context of a multi-member cluster. Application Center treats a single-node cluster, or stand-alone server, as a cluster of one member. Typically, the most common single-node cluster is a stager.

A stager is a server where content is placed on a "stage." On this stage, you can experiment with and fully test the quality and functionality of content from development and test environments before deploying the content to production environments.

In addition to stagers, other single-node clusters can benefit from Application Center without operating in a clustered environment. These servers can use Application Center for such tasks as: Deploying applications to or from other members or clusters. Viewing health and status alongside other clusters. Viewing performance data and event logs alongside other clusters.

MODULE 2: move Bigdata to HDFS

While moving big data to HDFS the following steps will be carried out.

- Make a HDFS directory on HADOOP Cluster by using - ***hadoop make file system />hadoopdfs -mkdir /input***
- Then Copy the sample voting input text file from local file System into this HDFS directory – ***hadoopdfs -copyFromLocal /Source /input***
- Change directory to run an example weblog program using jar file.
- Run the Hadoop map reduce code by using />Hadoop jar weblog.jar /input /output

MODULE 3: Processing BigData using MapReduce

DateWritable3A:

The DateWritable class is straightforward: It wraps a date, implements the readFields() method by reading the date in as a long, and writes the date out to the DataOutput by converting the date to a long. Finally, the comparison is delegated to the Date class's compareTo() method.

With this key in place, the next step is to build a Hadoop class that uses this key in a mapper, build a reducer, and assemble it into a workable application. Listing 2 shows the code for the LogCountsPerHour Hadoop application.

MODULE 3B: LogCountsPerHour

It defines a new mapper class called LogMapClass that emits DateWritable keys instead of Text keys.

- Its reducer is nearly identical to our previous reducer, but instead of emitting Text keys and a count, it emits DateWritable keys and a count.

- The run() method configures the class to run the appropriate mapper, reducer, and combiner as well as configures the output key (DateWritable) and output value (IntWritable).

The most interesting part of the LogCountsPerHour class is the mapper. In short, it parses an Apache Web Server log file line in the following format:

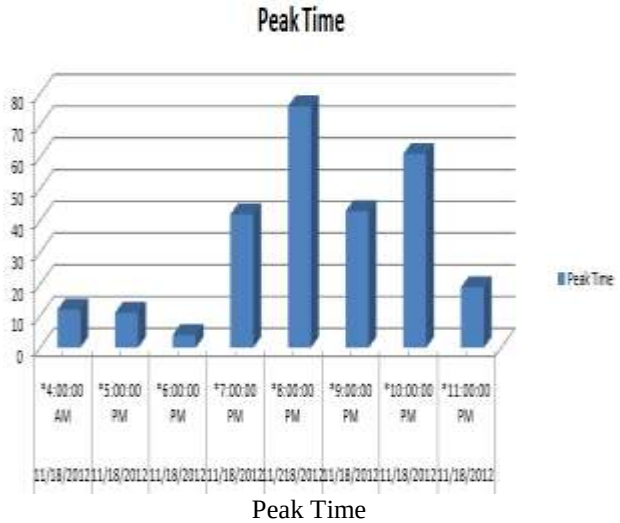
111.111.111.111 - - [18/May/2016:05:32:50 -0500] "GET / HTTP/1.1" 200 14791 "-" "Mozilla/5.0(compatible;Baiduspider/2.0;+http://www.baidu.com/search/spider.html)"

And from that it extracts the date:16/Dec/2012:05:32:50 -0500

And from that it extracts the day, month, year, and hour of the request. This means that all requests between 5:00 and 5:59:59 will be grouped together as a date object for the specified day at 5am.

This date will become the Key in our mapper, which means that when, for each record we output this hour and a count of 1, the combiners and reducers will ultimately compute the number of requests for that hour.

Figure 3.1



Peak Time

IV. CONCLUSIONS

This Project began by reviewing the domain of problems that MapReduce, and specifically Hadoop, is proficient at solving as well as the architecture that affords Hadoop its power. It presented the basics of building a MapReduce application and running it in Hadoop. It concluded with a real-world MapReduce application that analyzed a web server's log file and computed the number of page visits per hour. The proposed system performs more accurate mining of updated and new growing data. This is more efficient and more effective solution to process big data. Furthermore, this process will be fast when comparing with the existing approaches.

REFERENCES

[1] Yang Q., (2015), 'Introduction to the IEEE Transactions on Big Data', IEEE Transactions on Big data, Vol. 1, No. 1, pp. 2-14.

[2] Chen S., Wang Q., Yu G. and Zhang Y., (2015), 'I^2 MapReduce: Incremental MapReduce for Mining Evolving Big Data', IEEE transactions on Knowledge and Data Engineering, Vol. 27, No. 7, pp. 1906-1919.

[3] Gani A., Hashem, Mokhtar S., Khan S. and Yaqoo I. (2015), 'The rise of "big data" on cloudcomputing: Review and open research issues', Elsevier, no. 47, pp. 98-115.

[4] Chen C., Myounggyu W., Stoleru R. and Xie G. G. (2015), 'Energy-Efficient Fault-Tolerant Data Storage and Processing in Mobile Cloud', IEEE Transactions on cloud computing, pp. 28-41.

[5] Dahiphale D., Karve R., Liu H. and Vasilakos A. V. (2014),'An Advanced MapReduce: Cloud MapReduce, Enhancements and Applications', IEEE Transactions on Network and service Management, pp. 101-115.

[6] Madhu Kumar S. D. and Shabeera T. P. (2015) 'Optimising virtual machine allocation in MapReduce cloud for improved data locality', International Journal of Big Data Intelligence, Vol.2, No.1, pp.2 – 8.

[7] Han-Chieh Chao., Lai C., Tsai C., and Vasilakos V. (2015), 'Big data analytics: a survey', Journal of Big Data.

[8] Suryawanshi S., Wadne V. S. (2014), 'Big Data Mining using Map Reduce: A Survey Paper', IOSR Journal of Computer Engineering.

[9] Lian L. and Yang X. (2014), 'A New Data Mining Algorithm based on Map R Educe and Hadoop', international journal of signal processing, image processing and pattern recognition, Vol. 7, No. 2, pp. 131- 142.

[10] AslinJenil A.P.S. and Usha D. (2014) , 'A Survey of Big Data Processing in Perspective of Hadoop and Mapreduce', International Journal of Current Engineering and Technology, Vol. 4, No. 2, pp. 602-606.

[11] Arora S. and MadhuGoel, (2014), 'Survey Paper on Scheduling in Hadoop', International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, No. 5, pp. 812-815.

[12] Madhavi Vaidya, (2012), 'Parallel Processing of cluster by Map Reduce', International Journal of Distributed and Parallel Systems, Vol. 3, No. 1, pp. 167-179.

[13] Woo J. (2012), 'Market Basket Analysis Algorithm on Map/Reduce in AWS EC2', International Journal of Advanced Science and Technology, Vol. 46, pp. 25-38.

[14] Koundinya K., Kumar K., Madhu M. N., Srinath N., Sharma K. A. K., and Shanbag U. (2012), 'Map/Reduce Design And Implementation Of A Priori Algorithm For Handling Voluminous Data Sets', 'Advanced Computing: An International Journal, Vol. 3, No. 6, pp. 29-39.

[15] Rabi Prasad Padhy, 'Big Data Processing with Hadoop-MapReduce in Cloud Systems', International Journal of Cloud Computing and Services Science, Vol. 2, No. 1, pp. 16- 27.
[16] Japkowic N., Liu X., Matwin S. and Wang X. (2015), 'Meta-MapReduce for scalable data mining', Journal of Big Data.
[17] Branch R., Hurley R., McConnell S., Tjeerdsma H. and Wilson C. (2014), 'Cloud Computing and Big Data: A Review of Current Service Models and Hardware Perspectives', Journal of Software Engineering and Applications, Vol. 7, pp. 686-693.
[18] Agarwal J., Mishra N., Sharma S. and Shivhare H. (2013), 'Cloud Computing and Big Data', International Conference on Cloud, Big Data and Trust. 0

