ISSN 2395-1621

Map Reduced Based Log Analyzer

[#1]Sachin Ghisare, [#2]Ashish Jambale, [#3]Mohammad Rizwan,
[#4]Prof. Sonawane V. D., [#5]Prof. S.S.Patil

[1]ghisaresachin@gmail.com
[2]ashishjambale5005@gmail.com
[3]rizwan.tash@gmail.com

[#123]Department of Information Technology
[#45]Prof. Department of Information Technology

Al-ameen College of Engineering
Pune.

ABSTRACT

Log analysis is mainly used to extract the valuable information from the logs generated. Log analysis techniques and tools help in understanding patterns and trends within large data. When we use them for creating analytical models, they provide the best way for making decision. By analyzing all the data available, decision makers can better assess competitive threats, anticipate changes in customer behavior, strengthen supply chains, improve the effectiveness of marketing campaigns, and enhance business continuity. The size of logs generated is large in size; thus it's inefficient for common methods to analyze the logs on the single node. Therefore there is great demand to adopt a distributed method for large scale log analysis. The proposed system solves the problem of performance and scalability. In this project, we have developed a Hadoop cluster in which master node divides the task into subtasks using mapper and distribute them among several child nodes. Child node individually process the subtask and result produced at each node is sent to the master node using reducer which eventually sends results to oracle database. The user can see graphical representation of analysis reports through web application.

ARTICLE INFO

Article History

Received: 19th May 2016

Received in revised form :

20th May 2016

Accepted: 23rd May 2016

Published online :

25th May 2016

Keywords: **Hadoop, Map Reduce, Web log files, Log Analysis, HDFS, Distributed system.**

I. INTRODUCTION

Project Idea

In this project, we have developed a system to analyze the logs for behavior recognition using Hadoop framework an open source distributed file system and map-reduce implementation. We will be moving the large size log file onto Hadoop Distributed File System (HDFS). Map-Reduce algorithm will parse these log files. The analysis result generated is stored into the database. Moreover, we will implement a flexible and powerful way for displaying analysis results, in order to make the best use of this data.

Motivation of the Project

The main objective of log analysis system is to extract valuable information from logs generated from user transactions. When we use this information for creating analytical model provides best way to decision making. Also the advantages provided by the Hadoop technology with HDFS as distributed file system and Map reduce implementation for large-scale data processing in distributed cluster are the primary motivation for this project. By taking the advantages of these technologies we can design the system for large scale data analysis with enhanced performance and scalability.

II. LITERATURE SURVEY

Parallel Database is one of the traditional approaches to large scale data analysis . Parallel database systems stem from research performed in the late 1980s and most current systems are designed similarly to the early Gamma and Grace parallel DBMS research projects. Parallel databases gen- erally do not score well on the fault tolerance and ability to operate in heterogeneous environment properties . First, failures become increasingly common as one adds more nodes to a system, yet parallel databases tend to be designed with the assumption that failures are a rare event. Second, parallel databases generally assume a homogeneous array of machines, yet it is nearly impossible to achieve pure homogeneity at scale. Third, until recently, there have only been a handful of applications that required deployment on more than a few dozen nodes for reasonable performance, so parallel databases have

not been tested at larger scales, and unforeseen engineering hurdles await. Scalability problem can be solved by Hadoop Distributed framework. Hadoop provide framework to process large amount of data in very efficient way. Hadoop is reliable because it considers the fact that the data may be lost and wrong i.e. it has fault tolerance . So Hadoop keep several replications in the cluster in order to deal with the collapse of certain machine. Hadoop is efficient, because it can parallel calculate in different machines. Hadoop is flexible, and we can add a new machine into cluster easily in order to deal with more data. Ideally, the scalability advantages of Map-reduce could be combined with the performance and efficiency advantages of parallel databases to achieve a hybrid system that is well suited for the analytical DBMS market and can handle the future demands of data intensive applications . In our system, we describe our implementation of and experience with Hadoop DB, whose goal is to serve as exactly such a hybrid system. The basic idea behind Hadoop DB is to use Map-Reduce as the communication layer above multiple nodes running single-node DBMS instances. Queries are expressed in SQL, translated into Map-Reduce by extending existing tools, and as much work as possible is pushed into the higher performing single node databases. One of the advantages of Map-Reduce relative to parallel databases is cost. There exists an open source version of Map-Reduce (Hadoop) that can be obtained and used without cost. Yet all of the parallel databases mentioned above have a non- trivial cost, often coming with seven figure price tags for large installations. Since it is our goal to combine all of the advantages of both data analysis approaches in our hybrid system, we decided to build our prototype completely using open source components in order to achieve the cost advantage as well. Monitoring Sales Data plays a pivotal role within any organization. But, it is difficult to make strategic decisions based on this raw data alone. With the advent of new technology, the organization has the ability to analyze and use this data to improve retail productivity.

III. MATHEMATICAL MODEL

In this project we have performed experiments on comparing analysis time on different nodes num- ber to show advantages of MapReduce model. Here are the performance test results of analyzing system logs on implementation of our framework on different nodes number clusters respectively. We measure the processing time for all of them, the less time means the better performance.

Runtime= overhead + (time to process data)/(no of nodes)

Overhead refers to any constant time spent in the workflow. For example, The time to dis- tribute files using distributed cache counts as overhead. Any time spent that is independent of the amount of data you have will fall under this ”overhead” category. ”Time to process data” refers to the dynamic time in the workflow. This is the amount of time you spend processing the actual data, ignoring the constant time overhead. Number of nodes refers to the no of datanodes on which analysis is performed. Consider on a single node 5 MB of data is taking 15 seconds to process the data, if we add another node it will take less time say 9 seconds. This way on increasing the nodes we will able to decrease the processing time.

For conciseness, let’s rewrite the above equation using variables:

$$T = O + (P/N) \quad \ldots\ldots \quad (1)$$

Where, T= Runtime, O= Overhead, P= Time to process data, N= Number of Nodes.

Thus we conclude that, for the same size of source log, the more nodes analysis cluster has, the less the time it costs to do analysis. And we can see it more obvious as the data size becomes larger.

IV. PROPOSED SYSTEM

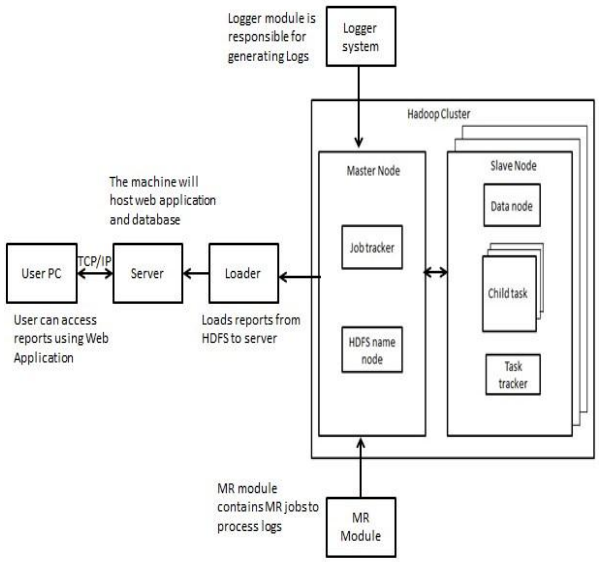

Fig 1. System architecture

The above Figure shows system architecture, it mainly consist of a Hadoop cluster in which one is Name Node and other is data node and a sever machine on which a oracle DBMS is installed and we are running a web application on the same machine for demonstration.

In the hadoop cluster the name node or say master node is the node which usually moves the log file into the cluster and also stores metadata, and a data node or say slave node which is storing this data in the form of blocks of data. The master node has a component job tracker which is responsible for running and tracking the job. The task tracker on slave machine will execute that task locally on the data which is stored in that slave node. The Logger module is responsible for generating a large size of logs.

The server machine in this architecture is the machine on which the oracle DBMS is installed and we are ruunung a web application on that machine which will get the result from databases and shows the result to user in graphical formats like graphs and charts.

Implementation

MR Module

This module will be responsible for storing the log files on HDFS. The modules also consists of MR jobs for processing the log and push the result on server machine. This module also executes on node.

UI Module

This module will consists of user interface through which data analyst can request the analysis report. This module is responsible for fetching reports from server machine.

Analysis Module

This module consists of sub modules which are responsible for analyzing data location wise, product wise, time wise, customer wise using map-reduce technique.

V. RESULT

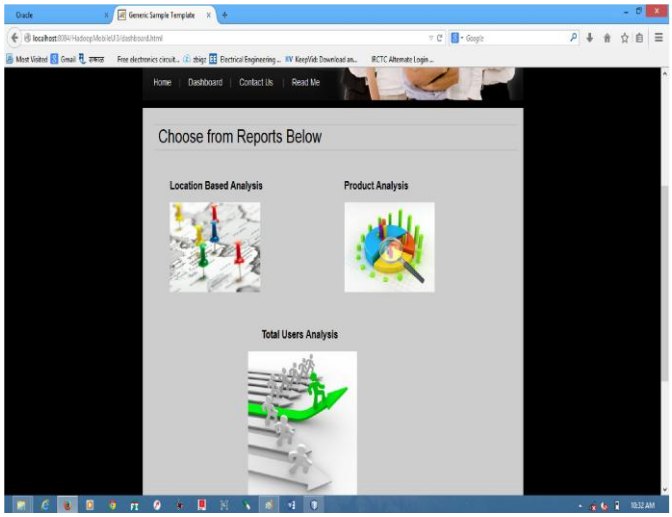

Fig 2. Log Analysis options

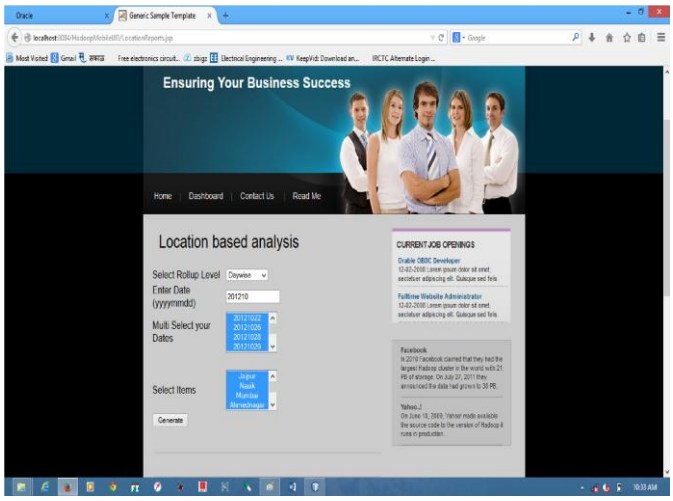

Fig 3. Location based analysis input

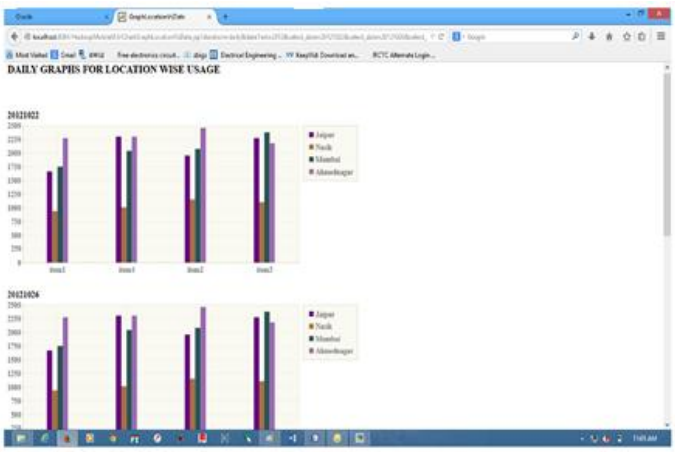

Fig 4. Location based analysis output

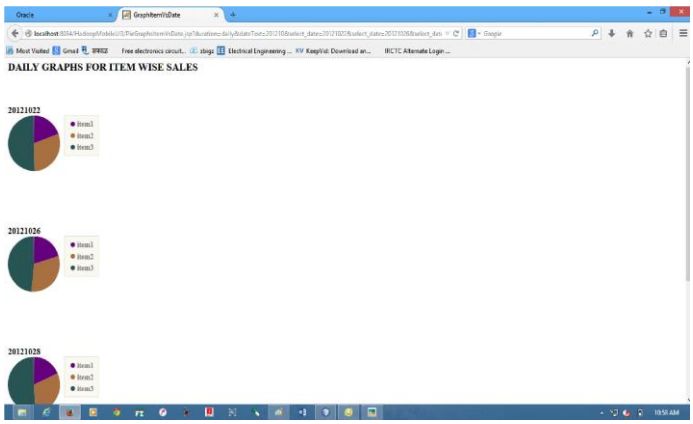

Fig 5.Product based analysis output

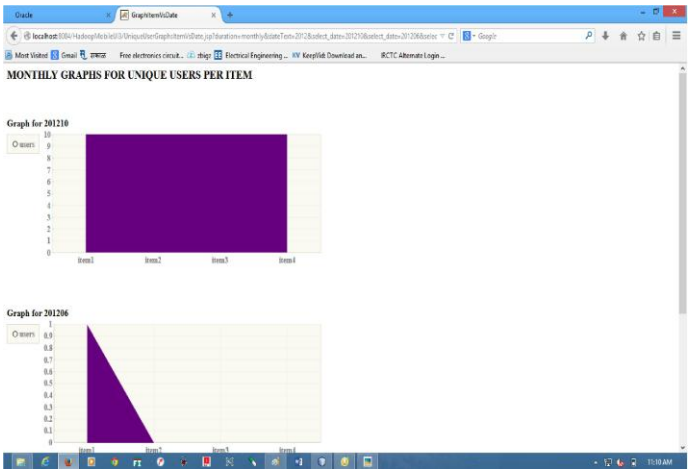

Fig 6. Total User analysis output

VI. CONCLUSION

In this project, we have implemented Map-Reduce Framework to analyse the logs of transactions done by users. In order to sustain in today's contemporary world organizations need better analysis system to understand market scenario and customer views. A better understanding of what and why customers are purchasing provides the basis for business planning from strategic to operational level. As there is a need of processing large log files in less amount of time the proposed system 'Log Analysis for Behavior Recognition Using Map-Reduce Technique' solves the problem of performance and scalability.

REFERENCES

[1] Ashra, M. Z., Taniar, D., Smith, K. (2004). ODAM: An optimized distributed association rule mining algorithm. Distributed Systems Online, IEEE, 5(3)

[2] Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel Abadi, Alexander Rasin, HadoopDB:An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical work-loads .

[3] Bringing the Power of SAS to Hadoop .Combine SAS World-Class Analytic Strengt with Hadoops Low-Cost, High-

Performance Data Storage and Processing to Get BetterAnswers, Faster.

[4] Deepika Fole And Chaitali Choudhary,Finding an Ecient Approach for Generating Frequent Patterns in Large Database February 2015.

[5] D. Christy Sujatha, D. Selvam, A. B. Karthick Anand Babu. Minimizing Time Span of Big Data Analytics using Hadoop -Map Reduce. International Journal of Engineering Research Technology (IJERT)ISSN: 22780181.

[6] K.Vanitha, R.Santhi : "Evaluating the performance of association rule mining algorithms", JGRCS, 2010

[7] K.V.Shvachko,The Hadoop Distributed File System Requirements, Hadoop Wiki, June 2006: http://wiki.apache.org/hadoop/DFS requirements.

[8] Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining,Addison-Wesley.

[9] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf.Very Large Data Bases, pages 487-499, Santiago, Chile, September 1994

[10] X. Jiong, Y. Shu, R. Xiaojun, D. Zhiyang, T. Yun, J. Majors, A. Manzanares,and q.xiao , improving Mapreduce performance of through data placement in heterogeneous hadoop clusters, april 2010.

