7th International Conference on Recent Trends in Engineering, Science & Management
Genba Sopanrao Moze College of Engineering, Balewadi-Baner, Pune (ICRTESM-17)
01st-2nd April 2017, www.conferenceworld.in ISBN: 978-93-86171-12-2

A NEW APPROACH FOR FREQUENT ITEMSET DATA MINING IN HADOOP ENVIRONMENT

Jadhav Kalyani B[1],Tamhane Manisha S[2], Surwase Sonali U[3],

Asst Prof.Pallavi Patil[4],

[1,2,3,] *Department of Computer Engineering, GSMCOE Balewadi, Pune (India)*

[4]*Prof.Department of Computer Engineering, GSMCOE Balewadi, Pune (India)*

ABSTRACT

Frequent pattern mining is an essential data mining task, with a goal of discovering knowledge in the form of repeated patterns. Many efficient pattern mining algorithms have been discovered in the last two decades, yet most do not scale to the type of data we are presented with today, the so-called "Big Data". Scalable parallel algorithms hold the key to solving the problem in this context. In this chapter, we review recent advances in parallel frequent pattern mining, analyzing them through the Big Data lens. We identify three areas as challenges to designing parallel frequent pattern mining algorithms: memory scalability, work partitioning, and load balancing. With these challenges as a frame of reference, we extract and describe key algorithmic design patterns from the wealth of research conducted in this domain.

Keywords- Frequent Itemset Mining, RHadoop, Map-reduce, HDFS

I. INTRODUCTION

This modern era, there is sudden increase in amount of data generated and ability to collect this large data has increased signicantly because of advances in hardware and software platforms. For example, Wal-Mart alone handles more than 1 million customer transactions each hour and imports those into databases estimated to contain more than 2.5 petabytes of data. Web log data sites such as Facebook and Twitter handles, stores and generated terabyte to petabyte of data each day alone and this number keeps on growing. Since the data is often so large that specialized methods are required for the mining process. With this extremely large data set it may be dicult or infeasible for single machine to process and nd association pattern rules between the data set. Because the traditional algorithm has issues of scalability, memory and computation cost, stability and low performances when it comes to deal with this huge data. Also the streaming and big-data architectures are slightly dierent and pose dierent challenges for the mining process. When processing this big data for the problem of frequent itemset there is need to consider a lots of challenges. A major problem arises when the data is large enough to be stored in a distributed way. Therefore, signicant costs are incurred in shuing the data or the intermediate results of the mining process across the distributed nodes. These costs are also referred to as data transfer costs. Therefore when handling large dataset, then the algorithms need to designed to take into account both the disk access constraint and the data transfer costs. In addition, many distributed frameworks such as

MapReduce require specialized algorithms for frequent pattern mining. The focus of big data framework is somewhat dierent from streams, in that it is closely related to the issue of shuing large amounts of data around for the mining process. Interestingly, it is sometimes easier to process the algorithms in a single pass in streaming fashion, than when they have already been stored in distributed frameworks where access costs become a major issue.Dealing with big datasets in the order of terabytes or even peta bytes is a challenging. Hence cloud computing provides an effective technique called Parallel programming which is becoming a necessity to deal with the massive amounts of data, which is produced and consumed more and more every day. Parallel programming architectures, and hence the algorithms, can be grouped into two major subcategories: shared memory and distributed (share nothing). On shared memory systems, all processing units can concurrently access a shared memory area. On the other hand, distributed systems are composed of processors that have their own internal memories and communicate with each other by passing messages. It is easier to adapt algorithms to shared memory parallelism in general, but they are typically not scalable enough.

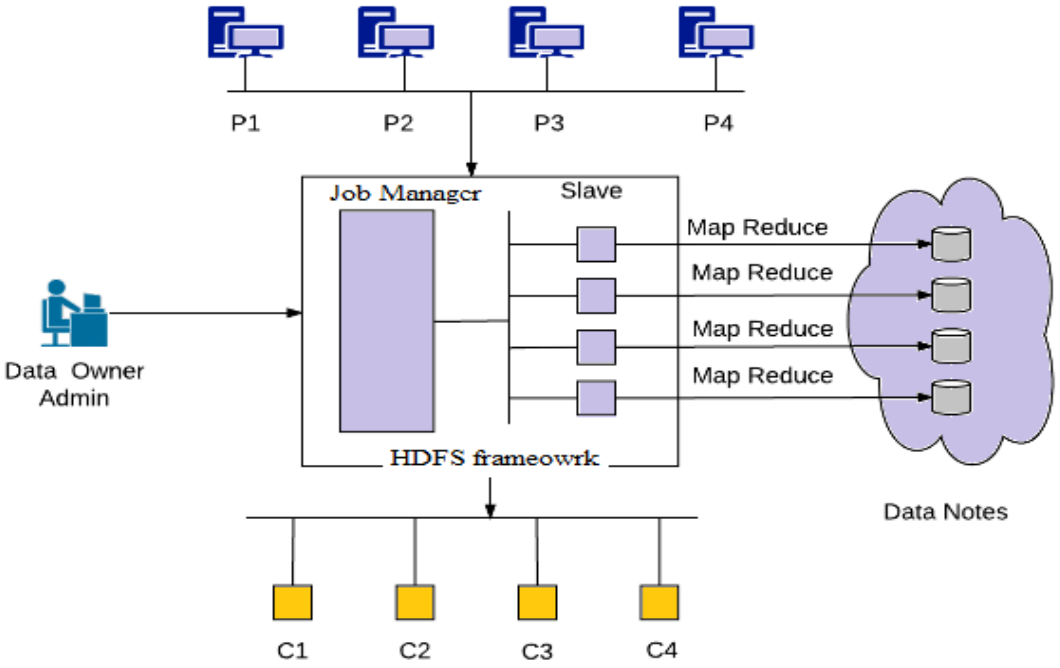
II. PROBLEM STATEMENT

The proposed work first investigate problem Frequent Itemset Ultra-metric Tree (FIUT) find the issues of existing system, System also focus on database security like SQL injection as well data collusion attacks etc and find effcient way for security as well execution in HDFS framework Implement FiDoop on our in-house Hadoop cluster. We show that FiDoop on the cluster is sensitive to data distribution and dimensions, because itemsets with different lengths have different decomposition and construction costs. To improve FiDoop's performance, we develop a workload balance metric to measure load balance across the cluster's computing nodes.

III.OBJECTIVES

i. To develop web page consisting of large databases which need to be considered for
ii. To check the databases present for any malicious content and if any found required attention to be given to that specific database.
iii. To develop DM system using hybrid algorithm which is based on Apriori and FP Growth for frequent itemset mining?
iv. To study working of DM system on conventional computer systems.
v. To embed DM system on hadoop framework and check its functionality.
vi. To compare working of DM system on conventional computer and in hadoop framework.
vii. To provide privacy and security to databases and DM system.
viii. To provide properly mined data to client in graphical representation.

IV. IMPLIMENTATION



V. EXISTING SYSTEM

Existing parallel mining algorithms for frequent itemsets lack a mechanism that enables automatic parallelization, load balancing, data distribution, and fault tolerance on large clusters. As a solution to this problem, we design a parallel frequent itemsets mining algorithm called FiDoop using the MapReduce programming model. To achieve compressed storage and avoid building conditional pattern bases, FiDoop incorporates the frequent items ultrametric tree, rather than conventional FP trees. In FiDoop, three MapReduce jobs are implemented to complete the mining task. In the crucial third MapReduce job, the mappers independently decompose itemsets, the reducers perform combination operations by constructing small ultrametric trees, and the actual mining of these trees separately. We implement FiDoop on our in-house Hadoop cluster. We show that FiDoop on the cluster is sensitive to data distribution and dimensions, because itemsets with different lengths have different decomposition and construction costs. To improve FiDoop's performance, we develop a workload balance metric to measure load balance across the cluster's computing nodes. We develop FiDoop-HD, an extension of FiDoop, to speed up the mining performance for high-dimensional data analysis. Extensive experiments using real-world celestial spectral data demonstrate that our proposed solution is efficient and scalable**.**

VI. PROPOSED SYSTEM

i. To develop web page consisting of large databases which need to be considered for data mining.
ii. To check the databases present for any malicious content and if any found required attention to be given to that specific database.

ISBN: 978-93-86171-12-2

iii. To develop DM system using hybrid algorithm which is based on Apriori for frequent itemset mining using privacy techniques on high dimensional data.
iv. This system also work with heterogeneous cluster node and we also provide energy consumption strategies base on ad-hoc usage of data node.
v. 5. We implement a thermal management approach on data node, like load balancing and choose specific data node which having minimum temperature.
vi. To compare working of DM system on conventional computer and in hadoop framework.
vii. To provide privacy and security to databases and DM system.
viii. To provide properly mined data to client in graphical representation.

VII. SYSTEM APPLICATIONS

Following are the applications of Home Automation System:

i. Patient disease recognition system.
ii. Health care recommendation hospitalized system.
iii. Queue recommendation base system.

VIII. HARDWARE & SOFTWARE COMPONENTS

8.1SOFTWARE REQUIREMENT:

Front End

i. Operating system: Windows XP/7 Higher
ii. Jdk 1.7.0
iii. Hadoop 1.2
iv. Internet Explorer 6.0/above Back-End
v. MongoDB
vi. Tools : Ecllipse, Heidi SQL, JDK 1.7 or Higher

8.2HARDWARE REQUIREMENT:

i. Processor:- Intel Pentium 4 or above
ii. Memory:- 512 MB or above
iii. Other peripheral:- Printer
iv. Hard Disk:- 10gb

IX. CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

This paper introduction on how the Hadoop framework can be used for large data storage and analytics purpose through this paper. Large amount of source data from social media, web logs or third party stores is stored on .Hadoop to enhance analytic models that drives research and discovery. Data can be stored on Hadoop clusters in cost effective manner and can be retrieved easily when needed. Operational cost of whole data analytics and

data processing can be lowered by use of Apache Hadoop. Its MapReduce on HDFS provides a scalable, fault tolerant platform for processing large amount of heterogeneous data.The paper summarizes the current issues in data mining algorithms migration towards Hadoop platform. We have identified the current gaps and open research areas. Our future research will focus on these open problems and propose effective solutions for the same.

9.2 FUTURE SCOPE

In this proposed research work to design and implement a system which provide the parallel processing patient request using HDFS framework which eliminate the queue base patient time prediction system. To predict the waiting time for each treatment task, we use the random forest algorithm to train the patient treatment time consumption based on both patient and time characteristics and then build the PTTP model. Because patient treatment time consumption is a continuous variable, a Classification and Regression Tree (CART) model is used as a meta-classifier in the RF algorithm. Because of the shortcomings of the original RF algorithm and the characteristics of the patient data, in this paper, the RF algorithm is improved in 4 aspects to obtain an effective result from large-scale, high dimensional, continuous, and noisy patient data. Compared with the original RF algorithm, our PTTP algorithm based on an improved RF algorithm has significant advantages in terms of accuracy and performance. Moreover, there is no existing research on hospital queuing management and recommendations. Therefore, we propose an HQR system based on the PTTP model. To the best of our knowledge, this paper is the first attempt to solve the problem of patient waiting time for hospital queuing service computing. A treatment queuing recommendation with an efficient and convenient treatment plan and the least waiting time is recommended for each patient.

.

X. ACKNOWLEDGEMENT

We are deeply indebted to our seminar guide, Asst.Prof.Pallavi Patil for his valuable guidance and support for completion of this seminar.

We are thankful to all our teachers and professors of our department for giving us their expertise in the related topic.

We would also like to thank our library staff, internet staff and laboratory assistants for providing us cordial support and necessary facilities which were of great help for preparing this report.

REFERENCES

[1]. Ninghui Li, WahbehQardaji, Dong Su, Jianneng Cao,"PrivBasis: Frequent Itemset Mining with Differential Privacy.", in VLDB, 2012.

[2]. Maurizio Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, "Anonymity preserving pattern discovery," VLDB Journal, 2008.

[3]. A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in KDD, 2002.

[4] W. K.Wong, D.W. Cheung, E. Hung, B. Kao, and N. Mamoulis, "An audit environment for outsourcing of frequent itemset mining," in VLDB, 2009.

[5]. Revealing Information while Preserving Privacy Irit Dinur Kobbi Nissim C. Dwork, “Differential privacy,” in ICALP, 2006

[6]. M. L. Gonzales, "Unearth BI in Real-time," vol. 2004: Teradata, 2004.

[8] B. Goethals, "Memory Issues in Frequent Pattern Mining," in Proceedings of SAC'04. Nicosia, Cyprus: ACM.