ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.5

A Survey on Efficient Approach of Finding Frequent Itemsets

Amuthabala.k[1], Sarumathi.D[2], Shruti[3], Supriya.M[4]
Assistant professor[1], Student[2, 3, 4]
Department of Information Science and Engineering
Atria Institute of Technology, Karnataka, India

Abstract:
Finding frequent itemsets is becoming popular in business management system, hence it is a survey about finding frequent itemsets using map reduce concept. To find frequent itemsets from the large transaction database we use frequent itemset mining algorithm. When we use large transaction datasets at a time in a single machine, the fore mentioned frequent itemset mining algorithm undergoes performance deterioration. And also the techniques of parallelism, data distribution, load balancing and fault tolerance had been disabled in existing mining technique system. Hence to address this issue, we came up with the concept of map reduce, a widely used programming model for processing big data.

Keywords: Minimum support count, frequent itemsets, FP-growth, FIUT, map reduce.

I. INTRODUCTION

The frequent itemset mining algorithms used to find frequent itemsets are FP-growth and FIUT tree. Here the inputs are data's and user specifying minimum threshold value. By using these two parameters we used frequent itemsets in the above mentioned frequent itemset algorithm. When we take datasets from data warehouse, it may contain noise. Hence to remove those noises the datasets undergoes preprocessing method, where it performs the operation of data cleaning, reduction, integration and wrangling. While retrieving datasets from data warehouse we would be directed to retrieve with different attributes, hence to remove this we perform preprocessing. FIUT mainly operated in Hadoop. Hadoop uses map reducing concept to find frequent itemsets. HDFS is the Hadoop distributed file system, which is provided by Hadoop itself to store large amount of datasets, we use HDFS to store large amount of itemsets. The efficiency and the time consumed to find frequent itemsets by both the algorithms are finally given in the form of graph, so that we could find the difference between them.

II. IMPORTANT SPECIFICATIONS

Parallelization: Designing the system or the program which processes data in parallel.

Data Distribution: Distributing the data to balance the workload and also they are graphical in nature to provide useful information.

Fault Tolerance: Even though failure happens for some of its component must possess the property to tolerate the failure.

III. ASSOCIATION RULE MINING

Association Rule Mining (ARM):

- To find frequent pattern, relations, associations, or casual structures from data sets found in various kinds of databases such as relational databases, transactional databases and other forms of data repositories.
- By extracting the most important frequent patterns ARM provides a strategic resource for decision support that simultaneously occurs in a large transaction database.
- To discover all rules that satisfy a user-specified minimum support and minimum confidence, is the ultimate objective of ARM.
- **There are 2 phases in ARM process:**

1) Identifying all frequent itemsets whose support is greater than the minimum support and,

2) Among the frequent itemsets, forming conditional implication rules.
Compared to second phase, first phase is more challenging and complicated.

IV. FIUT

- The FIUT algorithm consists of two key phases.

1) Two rounds of scanning a database is involved in first phase. Frequent one-itemsets is generated from first scan by computing the support of all items,

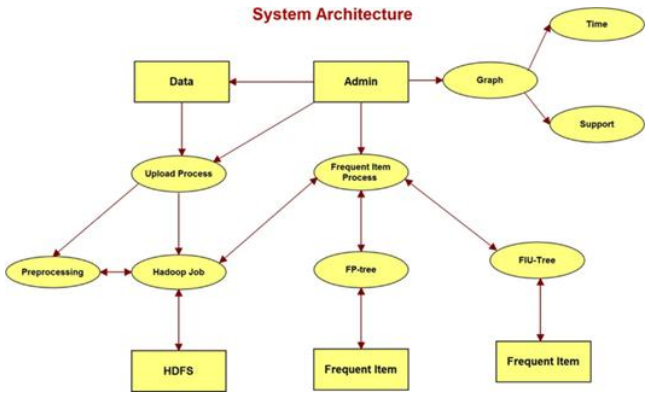
2) Whereas the second scan involves pruning all the infrequent items in each transaction record and results in k-itemsets. where, k denotes the number of frequent items in a transaction.

- Computing time and storage space by averting overhead of recursively searching and traversing conditional FP trees is significantly reduced by FIUT, compared to FP-Growth.

V. MAP REDUCE FRAMEWORK

Map reduce is a promising parallel and scalable programming model for data-intensive applications and scientific analysis. A Map reduce program expresses a large distributed computation as a sequence of parallel operations on datasets of key/value pairs.

VI. ARCHITECTURE DIAGRAM



ADMIN: The process of admin is to login, and should upload the data which are taken from UCI datasets. UCI is nothing but, for different data structure the data's will be uploaded in the user retrievable format. Where admin copies data's and stores it into the excel sheet, where it can be retrieved later. Once data is being stored it can be uploaded for the Hadoop job through upload process. Preprocessing: the data's which are available in the UCI are preprocessed datasets, but is we want to check for the noisy content in the dataset we check for preprocessing. In preprocessing, initially data is being sent, where it undergoes the process of data cleaning, data reduction and data integration, and finally the raw data is converted into the formattable or usable dataset.

HADOOP JOB: the job of Hadoop is, once the data is being received it undergoes process of storing the frequent itemsets, and finding frequent itemsets by satisfying minimum support count. Then finally the tree is drawn to show the result.

HDFS: Hadoop distributed file system which is the hardware provided by the Hadoop, where we can store large amount of data's thrice. If one data storage gets crashed, we can use with the other two storage of data. This is the main advantage of the HDFS. Once the data done with the Hadoop job it is sent to the frequent itemset process, where it is taken place twice to give the result for two graphs,1 FP- growth,2 FIUT. The efficiency and the time consumed for finding the frequent itemset in these 2 graphs is showed by giving graph, where we can find easily. Finally, the graph is sent to the admin.

EXISTING SYSTEM

- Apriori is an existing algorithm using the generate-and-test process that produces a large number of candidate itemsets; Apriori has to repeatedly scan an entire database. To reduce the time required for scanning databases, Han et al. Proposed a novel approach called FP-growth, which avoids generating candidate itemsets. Most previously developed parallel FIM algorithms were built upon the Apriori algorithm. Unfortunately, in Apriori-like parallel FIM algorithms, each processor has to scan a database multiple times and to exchange an excessive number of candidate itemsets with other processors.
- A major disadvantage of FP-growth like parallel algorithms, however, lies in the infeasibility to construct in-memory FP trees to accommodate large-scale databases.

PROPOSED SYSTEM

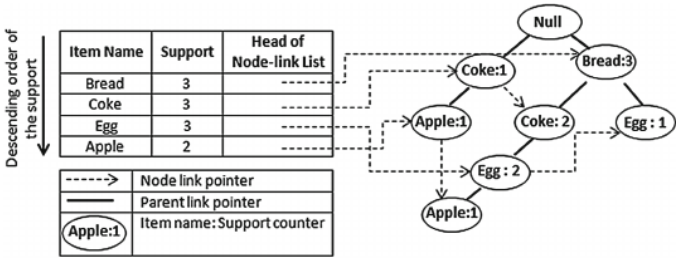
- In the proposed system a new data partitioning method to well balance computing load among the cluster nodes. We develop Fidoop-HD, an extension of fidoop, to meet the needs of high dimensional data processing.
- In light of the MapReduce programming model, we design a parallel frequent itemsets mining algorithm called FiDoop. The design goal of FiDoop is to build a mechanism that enables automatic parallelization, load balancing, and data distribution for parallel mining of frequent itemsets on large clusters.

VII. LITERATURE SURVEY

The concept is to propose pattern growth mining paradigm based FP-tax algorithm, which employs a tree structure to compress the database. Two methods to traverse the tree structure are examined: Bottom-Up and Top-Down. Experimental results show that both methods significantly outperform classic cumulate algorithm, in particular Top-Down FP-tax can achieve two order of magnitudes better performance than Cumulate. To mine association rules efficiently, we have developed a new parallel mining algorithm FPM on a distributed share-nothing parallel system in which data are partitioned across the processors. FPM is an enhancement of the FDM algorithm, which we previously proposed for distributed mining of association rules (Cheung et al., 1996). FPM requires fewer rounds of message exchanges than FDM and, hence, has a better response time in a parallel environment.

SPECIFICATIONS FOR REFERENCES SECTION

INPUT AND OUTPUT:



VIII. CONCLUSION:

Frequent itemset mining algorithms helps user to identify the frequent itemsets easily by using map reducing programming model. Here user will get to know which are the techniques used to find frequent itemsets. The output will be in the form of FP trees and FIUT, so that user can easily compare and check for the efficiency and time consumption.

IX. REFERENCES

[1]. M. J. Zaki, "Parallel and distributed association mining: A survey," IEEE Concurrency, vol. 7, no. 4, pp. 14–25, Oct./ Dec. 1999.

[2]. I. Pramudiono and M. Kitsuregawa, "FP-tax: Tree structure based generalized association rule mining," in Proc. 9th ACM SIGMOD Workshop Res. Issues Data Min. Knowl. Disc., Paris, France, 2004, pp. 60–63.

[3]. R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Rec., vol. 22, no. 2, pp. 207–216, 1993.

[4]. J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," Data Min. Knowl. Disc., vol. 8, no. 1, pp. 53–87, 2004.

[5]. S. Kotsiantis and D. Kanellopoulos, "Association rules mining: A recent overview," GESTS Int. Trans. Comput. Sci. Eng., vol. 32, no. 1, pp. 71–82, 2006.

[6]. R. Agrawal and J. C. Shafer, **"**Parallel mining of association rules," IEEE Trans. Knowl. Data Eng., vol. 8, no. 6, pp. 962–969, Dec. 1996.

[7]. A. Schuster and R. Wolff, "Communication-efficient distributed mining of association rules," Data Min. Knowl. Disc., vol. 8, no. 2, pp. 171–196, 2004.

[8]. M.-Y. Lin, P.-Y. Lee, and S.-C. Hsueh, "Apriori-based frequent itemset mining algorithms on MapReduce," in Proc. 6th Int. Conf. Ubiquit. Inf. Manage. Commun. (ICUIMC), Danang, Vietnam, 2012, pp. 76:1–76:8. [Online]. Available: http://doi.acm.org/10.1145/2184751.2184842

[9]. D. Chen et al., "Tree partition based parallel frequent pattern mining on shared memory systems," in Proc. 20th IEEE Int. Parallel Distrib. Process. Symp. (IPDPS), Rhodes Island, Greece, 2006, pp. 1–8.

[10] L. Liu, E. Li, Y. Zhang, and Z. Tang, "Optimization of frequent itemset mining on multiple-core processor," in Proc. 33rd Int. Conf. Very Large Data Bases, Vienna, Austria, 2007, pp. 1275–1285.

[11]. A. Javed and A. Khokhar, "Frequent pattern mining on message passing multiprocessor systems," Distrib. Parallel Databases, vol. 16, no. 3, pp. 321–334, 2004.

[12]. J. Neerbek, "Message-driven FP-growth," in Proc. WICSA/ECSA Compan. Vol., Helsinki, Finland, 2012, pp. 29–36.

[13]. Y.-J. Tsay, T.-J. Hsu, and J.-R. Yu, "FIUT: A new method for mining frequent itemsets," Inf. Sci., vol. 179, no. 11, pp. 1724–1737, 2009.

[14]. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[15]. J. Dean and S. Ghemawat, "MapReduce: A flexible data processing tool," Commun. ACM, vol. 53, no. 1, pp. 72–77, Jan. 2010.

[16]. W. Lu, Y. Shen, S. Chen, and B. C. Ooi, "Efficient processing of k nearest neighbor joins using MapReduce," Proc. VLDB Endow., vol. 5, no. 10, pp. 1016–1027, 2012.

[17]. D. W. Cheung, S. D. Lee, and Y. Xiao, "Effect of data skewness and workload balance in parallel data mining," IEEE Trans. Knowl. Data Eng., vol. 14, no. 3, pp. 498–514, May/Jun. 2002.

[18]. H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang, "PFP: Parallel FP-growth for query recommendation," in Proc. ACM Conf. Recommend. Syst., Lausanne, Switzerland, 2008, pp. 107–114.

[19]. L. Cristofor. (2001). Artool Project[J]. [Online]. Available: http://www.cs.umb.edu/laur/ARtool/, accessed Oct. 19, 2012.

[20] .J. S. Park, M.-S. Chen, and P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules," IEEE Trans. Knowl. Data Eng., vol. 9, no. 5, pp. 813–825, Sep. /Oct. 1997.

Authors

K. Amuthabala has obtained her B.E in Computer science Engineering at Avanishilingam University in Coimbatore, Tamilnadu, India in 2002 and her M.E degree in Software Engineering at Bangalore University in Bangalore, Karnataka, India in 2011.She is working as a Senior Lecturer in Information Science Department at Atria Institute of Technology, Bangalore, Karnataka. Her research areas of Interest include Data Mining Data warehousing and Cloud Computing.

Sarumathi .D Doing B.E in Information Science Engineering at Visvesvaraya Technology University, Atria Institute Of Technology, Bangalore.

Shruti Doing B.E in Information Science Engineering at Visvesvaraya Technology University, Atria Institute Of Technology, Bangalore.

Supriya .M Doing B.E in Information Science Engineering at Visvesvaraya Technology University, Atria Institute Of Technology, Bangalore.