Volume 7, Issue 4, October 2020 ISSN: 2348-8565 (Online)
International Journal of Modern Engineering and Research Technology
Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Advanced Algorithm for Association Rule Mining

Abhay Katiyar
Assistant Professor
Department of Computer Science and Engineering,
Jabalpur Engineering College,
Jabalpur (M.P.), India
E-mail: abhayittech@gmail.com

Rushikesh Tarachand Gaikwad
Research Scholar B.Tech
Indian Institute of Information Technology
Pune(M.H.), India
E-mail: grushikesh718@gmail.com

ABSTRACT

Association rule mining (ARM) is an important concept in Data Mining. Data mining is a technique for extracting useful information from large databases. ARM is a procedure which aims to find the frequent patterns, associations, correlations among sets of items in transaction databases. Many organizations such as industrial, commercial, or even scientific sites may generate large amount of transaction databases. Mining the rules effectively from such large dataset requires strong computing resources and much time. This paper describes a Parallel approach for association rule mining. In this approach we can distribute data mining tasks over several computing nodes to achieve parallel processing. The Apriori algorithm needs the frequent items to be short to perform well. This paper describes the various parallel association mining algorithm and their performance.

Keywords:— *CD – Count Distribution, DD- Data Distribution, IDD – Intelligent Data Distribution, HD- Hybrid Distribution. DMM –Distributed Memory Management, SMP – Shared Memory Processor, ARM – Association rule mining.*

I. INTRODUCTION

An association rule describes the association among items in which when some items are purchased in a transaction, the others are purchased too. Apriori algorithm does not need the transaction to be in main memory, but it the hash trees to be in main memory is needed. If the entire hash tree cannot fit in the main memory, the hash tree needs to be partitioned & multiple passes over the transaction database need to be performed (one for each partition of the hash tree), even with highly effective pruning method or apriori, the task of finding all association rules in many applications can require a lot of computation power that is available only in parallel computers.

The database to be mined are often very large measured in gigabytes & even terabytes, The need to handle large amount of data implies a lot of computational power, memory and disk I/O, which can only be provided by parallel computers.[1]

2. BASIC CONCEPTS

The task of finding all association rules in many applications can require a lot of computation power that is available only in parallel computers. Data is increasing in terms of both the dimensions (number of items) & size (number of transaction). One of the main attributes needed in an ARM algorithm is scalability. The ability to handle massive data stores. Sequential algorithm does not support scalability for

such large database. Therefore, we must depend on high performance parallel & distributed computing[2].

2.1 Parallel Association Rule Mining

Parallel association rule mining algorithms based on Apriori algorithms are count distribution (CD), data distribution (DD), intelligent data distribution (IDD) and Hybrid Distribution (HD)[3][4].

Achieving good performance on today's multiprocessor system is not trivial. The main challenges include synchronization and communication minimization, workload balancing, finding good data layout and data decomposition and disk I/O minimization (which is especially important for ARM). The parallel design space spans three main components, the hardware platform, the type of parallelism, and the load balancing strategy.

The association rule discovery is composed of two steps.

- The first step is to discover all the frequent item sets

  (Candidate set that has more support than time minimum support threshold specified).
- The second step is to generate association rules from the frequent item sets.

  The computation of finding the frequent item sets is much more expensive than finding the rules from these frequent item sets.

2.2 Classification of Parallel Association Mining

Parallelism can be classified based of memory system, parallelism type (data or task) and load balancing.

Distributed vs. shared memory systems.

Two dominant approaches for using multiple processors have emerged.

- Distributed Memory is where each processor has its own private memory.
- Shared Memory is where multiple processing elements share the same location in memory.

In distributed memory (DMM) architecture [3] each processor has its own private memory, which can be directly accessed only by that processor. For processor to access data in the private memory of another processor, message passing technique is used. In this technique one processor must send a copy of the desired data elements to the other processor.

In Shared memory (SMP) architecture [3] each processor has direct and equal access to the entire shared memory. Parallelism can be easily implemented on such system.

Data vs. Task parallelism.

In ARM there are two main paradigms for exploiting algorithm parallelism.

Task parallelism: It is the case where the processors independent to perform different computations. For example of counting a disjoint set of candidate which needs access to the entire database.

Data parallelism: It is the case where the database is partitioned among P processor for DMM [3][7].

In Hybrid parallelism both task and data parallelism is combined. It is also possible and desirable for exploiting all available parallelism in ARM methods.

Static vs. Dynamic load balancing.

Static load balancing initially uses heuristic cost function to distribute the work among the processors. There is not any subsequent data or computation movement is available for the connection of load imbalances resulting from ARM algorithms [3]. Dynamic load balancing is used to seek to address. The is work taken from heavily loaded processors and reassigned it to lightly loaded processors. Dynamic load balancing suffers from additional costs for work and data movement. It also needs additional cost for the mechanism which is used to detect whether there is an imbalance or not.

All parallel algorithms are based on sequential algorithm. Because of its success in the sequential setting many parallel algorithm uses Apriori method as their base method.

III. PARALLEL ASSOCIATION RULE MINING ALGORITHM

Most of the Parallel Association rule-mining algorithms are based on Apriori Algorithm. In which local and global support count should be counted to generate frequent item set.

3.1 Count Distribution (CD).

The focus of the count distribution algorithm is on minimizing communication. It does so even at the expense of carrying out redundant duplicate computation. The count distribution (CD)[4] is a simple parallelization of Apriori and achieves parallelism by partition data. The database D is partitioned into D1, D2, D3…Dn and distributed across n processors. In the first iteration of CD, every processor i scans its partition Di to compute the local supports of the entire size1 item sets. All processors are then engage in one round of support counts exchange. After that they independently find out global support count of all the items and then the large size1 item sets for the other iteration k, (k>1), each processor i runs the program fragment.

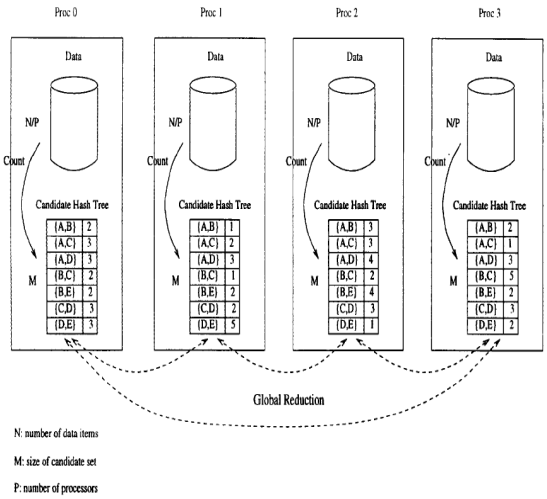

Figure 1: Count Distribution Algorithm.

3.2 Data Distribution (DD).

The data distribution algorithm attempts to utilize the aggregate main memory of the system more effectively. It is a communication happy algorithm that requires nodes to broadcast their local data to all other nodes. The DD algorithm[4] is designed to minimize computational redundancy and maximize the use of the total system memory by generating disjoin candidate sets on each processor, however each node must scan the entire database to examine its candidates.

- DD algorithm incurs from three types of inadequacy.
- The algorithm results in high communication overhead due to an incapable scheme used for data movement.
- Second the schedule for interactions among processor to idle.
- Each transaction has to be processed against multiple hash trees causing redundant computation.

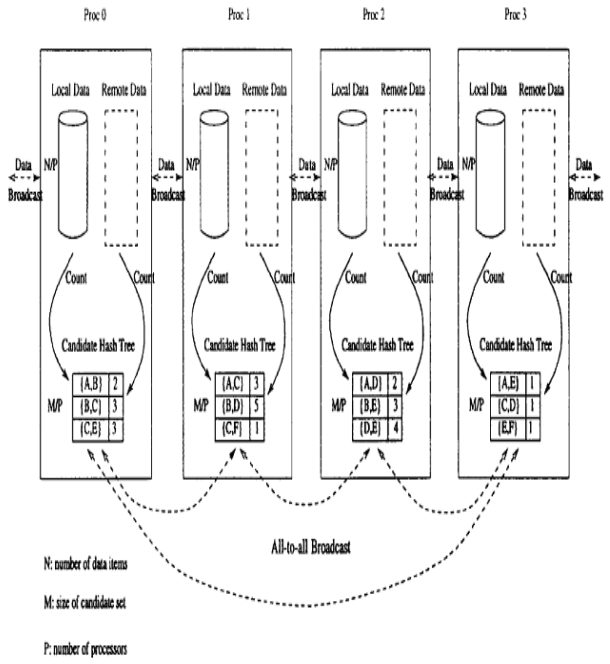

Figure 2: Data Distribution (DD) Algorithm

3.3 IDD (Intelligent data distribution)

IDD algorithm[4] is the improved version of DD algorithm. Here the communication overhead and processor idling time is minimized compared with DD and redundant computations are also eliminated. In IDD, Han and his colleague's use a liner-time, ring based, all to all broadcast for communication. Secondly as the candidate fit in the memory they switch to count distribution. Third, they perform a single item, prefix-based partitioning instead of a round-robin candidate partitioning. Before processing a transaction, they make sure that it contains the relevant prefixes. If not, the transaction can be discarded. Communication is still running in entire database, but a transaction might not process if it does not contain relevant items.

The IDD algorithm exploits the total system memory by partitioning the candidate set among all processors. The average number of candidates assigned to each processor is M/P, where M is the number of total candidates. As more processors are used, the number of candidates assigned to each processor decreases. This has two implications: first, with fewer numbers of candidates per processors, it is much more difficult to balance the work, second the smaller number of candidates gives a smaller hash tree and less computation work per transaction. Eventually, the amount of computation may become less than the communication involved. This would be more evident in the later passes of the algorithm as the hash tree size further decreases dramatically. This reduces overall efficiency of the parallel algorithm. This will be an even more serious problem in a system that cannot perform asynchronous communication.

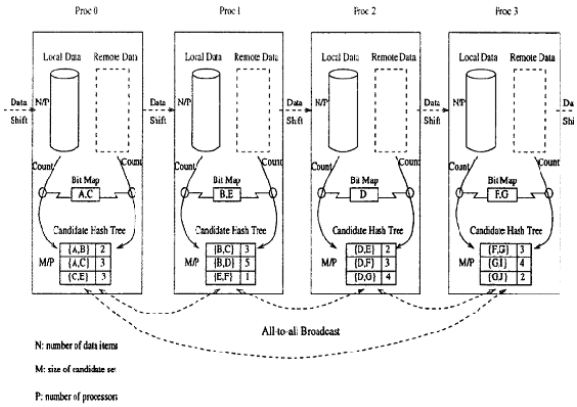

Figure 3: Intelligent Data Distribution (IDD) Algorithm

3.4 Hybrid Distribution (HD)

The Hybrid Distribution [4] is the combination of two algorithms, Count Distribution and Intelligent Data Distribution. It partitions the P number of processors into G equal-sized groups, where each group is considered as a super processor. Count Distribution is used among the G super processor, while the P/G processors in a group use intelligent data distribution. The database is partitioned horizontally among the G super processor, and the candidates are partitioned among the P/G processors in a group. In addition, for each pass the HD adjusts the number of

group dynamically. HD reduces database communication costs by 1/G and keep the processors busy during later iterations these are some advantages of Hybrid Distribution. The experiments done by Han and his colleague showed that whenever HD has the same performance as Count Distribution, it could handle much larger databases.

3.5 Prime Number based Parallel Association rule mining Approach

3.5.1 Buddy prima algorithm.

Buddy prima algorithm, a frequent item set mining algorithm, uses hybrid approach to mine frequent item set efficiently, In the first pass the algorithm scans the data set & computes the support count of all 1-item sets. The frequent 1 item sets are removed from further evaluation. Each time is represented by a unique prime no & each transaction is represented by the multiple of the equivalent prime no of the items in the item set. Parallel buddy prima algorithm [5-10] based on candidate distribution. It can be changed over count distribution, DD, IDD and HD algorithm.

IV. CONCLUSION

Parallel Association rule mining algorithms distribute the work load among several computing nodes for processing and hence these algorithms are required to improve the efficiency, processing ability i.e. overall performance of Association rule mining. Here we are using the large amount of data like scientific application, graphical application, Molecular biological Application and we distributed the work among the different processor so parallel approach provides better solution. In future we will more explore association rule mining and their applications for various new domains.

REFERENCES:

[1] Han, George Karypis, Vipin Kumar "Scalable Parallel Data Mining For Association Rules" IEEE Trans. Knowledge & Data Engg. 2000.

[2] David W. Cheung, Sau. D. Lee & Yongqiao. Xiao "Effect Of Data Skewness And Workload Balance In Parallel Data Mining". IEEE Trans. Knowledge & Data Engg.2002.

[3] Mohammed J.Zaki. "Parallel & Distributed Association Mining: A Survey." IEEE Trans. Knowledge & Data Engg. 1999.

[4] R. Agrawal, John C. Shafer "Parallel Mining Of Association Rules" IEEE Trans. Knowledge & Data Engg. 1996.

[5] Dr. S.N. Sivanandam, Dr. S. Sumathi, MS T. HarnsaPriya "Parallel Budy Prima – A Hybrid Parallel Frequent Item set Mining Algorithm For Very Large Databases" Academic Open Internet Journal, vol 13, 2004.

[6] T. Shintani and M. Kitsuregawa, "Hash Based Parallel Algorithms For Mining Association Rules." Proc. Conf. Parallel and Distributed Information Systems, 1996.

[7] D. Cheung, V. Ng, A Fu, and Y. Fu, "Efficient Mining of Association Rules in Distributed Databases" IEEE Trans. Knowledge and Data. Engg. Vol 8,no 6,1996.

[8] Feng, Feng, Junghoo Cho, Witold Pedrycz, Hamido Fujita, and Tutut Herawan. "Soft set based association rule mining." Knowledge-Based Systems 111 (2016): 268-282.

[9] Li, Kangping, Liming Liu, Fei Wang, Tieqiang Wang, Neven Duić, Miadreza Shafie-khah, and João PS

Catalão. "Impact factors analysis on the probability characterized effects of time of use demand response tariffs using association rule mining method." Energy Conversion and Management 197 (2019): 111891.

[10] Ceddia, Gaia, Liuba Nausicaa Martino, Alice Parodi, Piercesare Secchi, Stefano Campaner, and Marco Masseroli. "Association rule mining to identify transcription factor interactions in genomic regions." Bioinformatics 36, no. 4 (2020): 1007-1013.

* * * * *