International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-2 Issue-4, March 2013

Data Decomposition Technique Proposed for Candidate Itemsets Generation of Association Rule Mining Algorithms on Heterogeneous Cluster

Rakhi Garg, P. K. Mishra

Abstract*: Among various data mining task, association rule mining (ARM) is the major technique which is widely used in retail marketing, bioinformatics, website navigation analysis etc. It finds correlations among items in a given data sets and establishes an association between two non overlapping sets of frequently occurring values in a database. Various sequential and parallel ARM algorithms have been developed that differs in data layout, search technique, data structure, the number of database scans used and the system on which it is developed i.e. homogeneous or heterogeneous systems. This paper mainly put emphasis in the need of a candidate based ARM algorithm for heterogeneous PC cluster that reduces the database scans and time complexity. It also describe the design and functioning of the heterogeneous PC cluster i.e. MPICH2 and the data decomposition technique applied for candidate itemsets generation that plays important role in balancing workload as well as enhancing the performance of the algorithm on MPICH2 heterogeneous PC cluster.

Index Terms*: Association rule mining, candidate 1-itemsets, data mining, heterogeneous cluster.

I. INTRODUCTION

Due to the availability and the advent of high storage, speed and low price computer system, data mining makes its place in all walks of life right from day to day activity to administrative work. The huge amount of data is available and is stored in large databases. This data is of no use if it won't generate relevant, correct and useful information and knowledge. Data mining (DM) is the technique that finds hidden patterns in huge database which can be used in decision making. "It is an interdisciplinary field merging ideas from multiple disciplines such as database technology, statistics, machine learning, high performance computing, pattern recognition, neural networks, data visualization, information retrieval, image and signal processing, and spatial data analysis" [1]. An interesting, relevant, accurate and useful knowledge and information can be extracted from databases by applying data mining process on it. The knowledge thus discovered can be applied to decision making, process control, information management, and query processing [1]. Therefore, data mining is emerging as one of the most promising interdisciplinary advancement in the information industry [1]. Among various data mining task, association rule mining (ARM) is the major technique which is widely used in retail marketing, bioinformatics, website navigation analysis etc. It finds correlations among items in a given data sets and establishes an association between two non overlapping sets of frequently occurring values in a database. It helps retailers in planning marketing strategies, catalogue design and store layout by finding the association between the different items purchased by the customer. e.g. If retailer keeps bread with butter then the chances of sale will be increased because customer who buys bread is also interested in butter. ARM algorithm is basically two step process where first step finds the frequent itemsets and the second step generates rules that establish relation among these. Various sequential and parallel ARM algorithms have been developed that differs in data layout, search technique, data structure, the number of database scans used and the systems on which it is developed i.e. homogeneous or heterogeneous systems. It has been observed that most of the algorithms were implemented on dedicated homogeneous system and uses static load balancing technique based on the initial data decomposition for load assignment to the processors in the system which if used with heterogeneous system will lead to performance deterioration [12]. A typical parallel database server has multiple users, and has transient loads. This calls for an investigation of dynamic load balancing schemes. Dynamic load balancing is also crucial in a heterogeneous environment, which can be composed of meta and super-clusters, with machines ranging from ordinary workstations to supercomputers [12]. Very few algorithms for ARM have been designed for homogeneous as well as for heterogeneous PC clusters. Moreover, most of the researchers focuses on the first step of ARM i.e. finding of frequent itemsets as second one is straight forward. The frequent itemsets are generated from the candidate itemsets which are those itemsets that are expected to turn out to be large or frequent. Several sequential as well as parallel association rule mining algorithm have been developed to generate candidate and frequent itemsets. These algorithms differ from one another in the method of generating candidate sets, and counting of the support of a candidate, data structure employed and the number of database passes or scans takes place [3]. The main challenge is to develop one that generates candidate and frequent itemsets in least number of database scan or passes so as to reduce time complexity and space complexity.

Manuscript published on 30 March 2013.
*Correspondence Author(s)
Dr. Rakhi Garg, Computer Science, MMV, Banaras Hindu University, Varanasi, India.
Prof. P. K. Mishra, Department of Computer Science, Faculty of Science, Banaras Hindu University, Varanasi, India.

The above facts put an emphasis on more research work to be done to develop an algorithm based on candidate generation approach that can generate candidate itemsets with minimum number of database scans and reduces time as well space complexity. The proposed algorithm focuses on the data decomposition technique that when used in generating candidate 1-itemsets (C_1) reduces the database scan and time complexity and is discussed in section 4 of this paper. The algorithm is designed and executed on heterogeneous mpich2 cluster. MPICH2 cluster has already gained its popularity in the area of parallel computing. The paper is divided into five sections. In the first section i.e. introduction, the need of algorithm for heterogeneous PC cluster is discussed and in the second section the association rule mining and C_1 generation process is described. The third section focuses on the heterogeneous PC cluster and in the fourth section, the data decomposition technique that is proposed to generate C_1 on heterogeneous MPICH2 cluster have been discussed. Finally, conclusion is elaborated in the last section.

II. ASSOCIATION RULE MINING AND CANDIDATE ITEMSETS

In today's competitive and unpredictable business environment, only those organizations survive which have complete information and knowledge of customer buying habits and market strategy. ARM is one of the DM techniques that derive associations from data and has received a great deal of attention of scientist, researchers and industry. ARM was formulated by Agrawal et al. in 1993 and is often referred to as the market-basket problem [4]. In this problem, a set of items and a large collection of transactions, which are subsets of the items (baskets), are given and one has to find relationship between the various items within the baskets. The discovery of relationships among large volume of data is very useful in selective marketing, decision analysis, telecommunication networks, risk management and business management [1]. e.g. A retailer is interested in knowing the customers buying habit that helps him in developing marketing strategies, inventory management, sale promotions strategies, etc. The analysis of customers' buying habits can be done by finding associations between the different items that customers place in their shopping baskets. ARM is basically two step process [1]:-

(i) Finding frequent itemsets i.e. set of items satisfying a minimum support threshold.
(ii) Generation of rules that determines a strong association between set of frequent itemsets and satisfies a minimum confidence threshold.

The first step of the algorithm generates frequent itemsets from the candidate itemsets. The process of L_1 i.e. frequent 1-itemsets and C_1 generations are illustrated in fig. 1 [4] that was proposed by Agrawal Rakesh and Shafer John C [12]. The process of C_1 generation is the first step of any candidate based ARM algorithms. Let A, B be set of items J = $\{i_1,i_2,i_3,\ldots\ldots\}$ present in T transactions of database D. An association rule is an implication of the form $A \Rightarrow B$ holds, where $A \subset J$ and also $B \subset J$ and $A \cap B = \Phi$ [12]. The rule $A \Rightarrow B$ holds in transactions set D with support s, where s is the percentage of both A and B. The rule has confidence c in the transaction set D if c is the percentage of transactions in D containing A that also contains B [1]. Rule support (s) and confidence (c) are two measures of rule interestingness. ***Support*** refers to the % (percentage) of task relevant data tuples or transactions for which the pattern is true. And the ***confidence*** of a rule indicates the degree of correlation in the dataset between A and B. Simply one can say that the confidence measures a rule's strength. Thus, if we say that a rule $A \Rightarrow B$ has a confidence of 85%, it means that 85% of the records containing A also contain B [5].

Let us consider a database D with four transactions and five items i.e. a, b, c, d and e. It generates candidate C_1 and find L_1 frequent 1-itemsets after pruning of C_1. Let min_sup = 2.

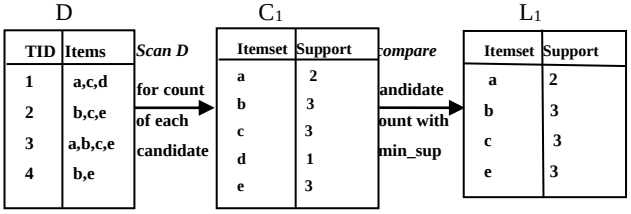

Figure 1: Generation of candidates (C_1) & frequent itemsets (L_1) by applying Apriori algorithm on D

III. HETEROGENEOUS PC CLUSTER

MPICH, a parallel computing system, is a strong and flexible implementation of the MPI. MPICH is a multi-platform, configurable system that can be used in development, execution, libraries, etc. for MPI, which is used in parallel or distributed computing projects. Parallelism can be achieved in MPICH either by using machines connected through network or multitasking on a single machine [6]. The goals of MPICH are seen as a research and a software project. As a research project, the goal of MPICH is to explore tradeoffs between performance and portability in the context of the MPI standard, study algorithms applicable to MPI implementation and investigate interfaces between MPI and tools. As a software project the goal of MPICH is to provide a portable, freely available MPI to everyone, give vendors and others a running start in the development of specialized MPI implementations and provide a testbed for other research groups working on particular aspects of message passing [7]. The MPICH is implemented as MPICH1 and MPICH2 that implements MPI-1.1 and MPI-2.2 standard respectively. MPICH2 is the latest and the most popular implementations of MPI [8]. MPICH2 is a high-performance and extensively convenient implementation of the MPI standard i.e. both MPI-1 and MPI-2. MPICH2 won the R&D 100 award in 2005 [9]. The goal of MPICH2 is to enable easy research in MPI through modular framework [10].

MPICH2 also support dynamic process management, remote-memory access, parallel I/O, C, C++, Fortran (77 & 90) language bindings, singleton init and thread safety at MPI_THREAD_MULTIPLE level [9]. MPICH2 is an open-source, freely available on internet. For experiment purpose it is downloaded form site http://www.mcs.anl.gov/mpi/mpich2 and runs as mpd. The heterogeneous cluster designed is a local cluster where personal computer represents the node of the cluster and is distributed memory system based. The model of the designed MPICH2 cluster is shown in fig. 2 below.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

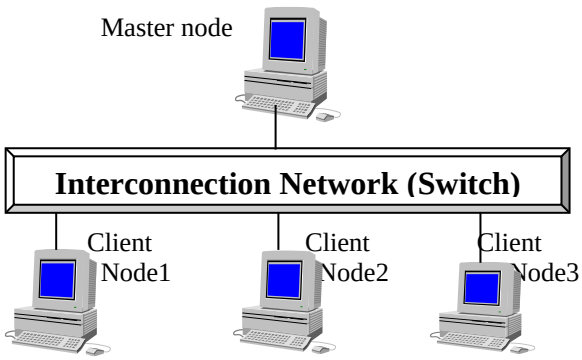

Figure 2: Cluster where node1, 2, 3 and 4 are PCs and are connected through switch

In this cluster, Ubuntu 10.04 operating system installed on three of the client nodes whereas Ubuntu 11.0 on a master node. Two nodes of a cluster are Pentium-4, core to duo out of which one is the master node , while other two nodes are Pentium-4 single processor. It means there are total four nodes and six processors in the cluster. This cluster is heterogeneous as the architecture of the nodes are different and the nodes are having different version of operating system Ubuntu. The configuration of the nodes of our mpich2 cluster is shown in Table I.

Table I: Hardware and Software specification of MPICH2 heterogeneous PC cluster

Hardware specification			
Number of nodes	1	1	2 (4 in total)
CPU	Pentium-4, core 2 Duo, 2.66GHz	Pentium-4, core 2 Duo, 2.66GHz	Pentium-4, 1.96GHz
Memory RAM	2GB DDR RAM	1.26GB DDR RAM	1 GB DDR
Network	nodes are connected through switch		
Disk	160GB HDD	160GB HDD	20GB HDD
Software specifications			
OS	Ubuntu 11.0(on master node) , Ubuntu 10.04 (on client nodes)		
Compiler	gcc/g++ 4.1.4		
Message Passing Library	MPICH2-1.3.2p1		
Tool	MPI		

IV. AN ALGORITHM PROPOSED FOR GENERATING C_1 CANDIDATE ON MPICH2 HETEROGENEOUS CLUSTER

The candidate generation based ARM algorithms generates candidates to obtain frequent itemsets i.e. C_{k-1} for the generation of frequent itemsets L_k where k is the k-itemste at every step of the algorithm till maximal frequent itemsets is obtained [11]. The C_1 generation is the first step involved in ARM algorithm based on candidate generations structure.

In the case of heterogeneous cluster, the nodes differs from each other in memory size, processing speed and operating system. In order to achieve performance on such cluster the main issuse that has to be emphasized is the dynamic load balancing so that none of them sits idle during the execution of the algorithm. The two data distribution strategies for achieving workload balance among the nodes of the cluster is disscussed in subsection 4.1. The algorithm for both the cases are described in subsection 4.2. These algorithms are implemented and discussed in the subsection 4.2 on heterogeneous MPICH2 cluster to compute C_1 as well as L_1 itemsets. The execution time of the processors as well as the total execution time of the cluster which is obtained experimentally in both the cases are shown in the subsection 4.3.

4.1 Data Distribution Strategy

It has been observed that ARM algorithms that employs vertical mining concepts takes less computation time to generate candidate and frequent itemsets [12]. Therefore, the vertical mining in the proposed approach to generate C_1 on MPICH2 heterogeneous cluster where horizontal database is firstly transformed into the vertical database is used. The vertical mining concept is used because with vertical tidsets the candidates as well as the frequent itemsets can be counted via tidset of transactional database intersection only and no complex structure is required. The real transactional database available for research purpose from http://www.cs.rpi.edu/~zaki/www-new/pmwiki.php/Software/Software is downloaded and converted into vertical database layout that contains vertical tidsets of transactions. In order to obtain equal workload balance among the nodes of our MPICH2 heterogeneous cluster so that none of them sits idle during the generation of C_1, the distribution of data present in the form of transactions among them is done in two ways. In the first case, the equal number of transactions among the nodes of the cluster is distributed whereas in the second case data as chunks of 500 transactions among the nodes is distributed. This is done to find out the best data decomposition method that can enhance the performance of ARM algorithm on heterogeneous cluster.

4.2 Algorithm

The algorithm for both of the data distribution strategies are shown in Table II and III respectively.

Table II: Algorithm for C_1 generation by equally distributing vertical tid-lists of itemsets among the nodes of the MPICH2 heterogeneous cluster

Begin

1. Read chess.ascii real transactional database in horizontal data layout at master node.
2. Convert it into vertical tidsets of transactions at the master node.
3. Store the number of items and the total number of transactions present in the given database at the master node.
4. For equal distribution of transactions among the client nodes of a cluster, compute the number of transactions that has to be distributed among the nodes which is simply obtained as (Total number of transactions/(Number of nodes in the cluster)) at the master node.
5. At the client nodes calculate the local support count of C_1 of each of the items of the database and send it to the master node and also the time employed in the computations.
6. At master node the global support count of C_1 is computed by simply adding the local support counts of C_1 of the items gathered from each client node and also the total processing time involved in it.

End

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table III: Algorithm for C_1 generation by distributing vertical tid-lists of itemsets as a chunk of 500 among the nodes of the MPICH2 heterogeneous cluster

Begin

1. Read chess.ascii real transactional database in horizontal data layout at master node.
2. Convert it into vertical tidsets of transactions at the master node.
3. Store the number of items and the total number of transactions present in the given database at the master node.
4. For distribution of transactions as chunks of 500 transactions among the client nodes of a cluster, at master node do:-

(i) Assign first 500 transactions to node1 and then
(ii) Compute the remaining number of transactions which is Rem_tran_1 = Total number of transactions-500.
(iii) Check if Rem_tran_1>0 then repeat the same procedure for node 2 as given in (ii), here we obtain Rem_tran_2 = Rem_tran_1-500.
(iv) Repeat the same procedure given in (iii) for the remaining nodes of the cluster.
(v) Once the allocation of transaction on all client nodes has been done, then go to step 5.

5. At the client nodes calculate the local support count of C_1 of each of the items of the database and send it to the master node and also the time required to do the computations.
6. At the master node the global support count of C_1 is computed by simply adding the local support counts of C_1 of the items gathered from each client node and also the total processing time involved in it.
7. Check if any of the transactions has not been assigned till now to any of the nodes. (i.e. Rem_trans_2 > 0) If it is so then repeat from step 4 to 6 till all have been assigned to the nodes for the C_1 generation otherwise stop.

End

4.3 Experiments and Results

The execution time of processors in both the cases are obtained and the total execution time is compared which is shown in fig. 3 and fig. 4. From these figures it's clear that althogh fault tolerance is strong in case of distributing data as chunks of five hundred transactions among the processors of a cluster but better result with respect to the execution time is obtained with the equal distribution of data among processors.

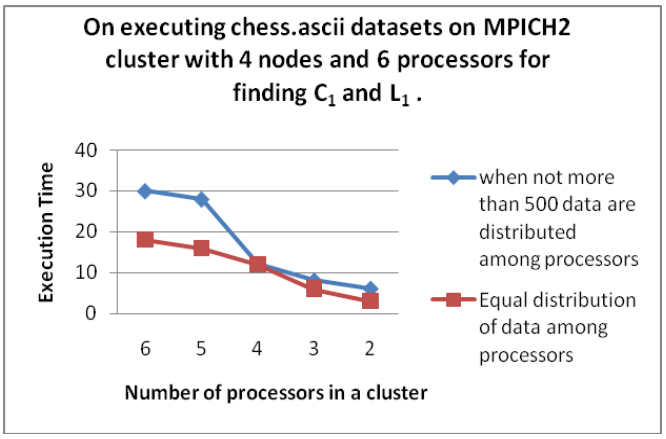

Figure 3: Shows the execution time (in sec.) of the number of processors in a cluster for C_1 and L_1 generation by applying two data decomposition strategies discussed in subsection 4.1

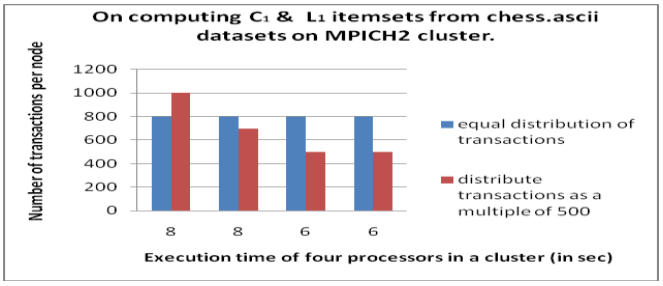

Figure 4: Shows the execution time and the number of transactions per node of the cluster while generating C_1 and L_1 on applying two data decomposition strategies discussed in subsection 4.1

V. CONCLUSION

From the experimental results obtained, it's very much obvious that equal distribution of tidlist (data sets) among the nodes of heterogeneous PC cluster reduces the time complexity of C_1 generation as compared to the one where the data sets are distributed as chunks of 500 transactions among the nodes of the cluster. Such data decomposition enhances the performance of any candidate based parallel ARM algorithms on heterogeneous cluster as candidate generation is the first step of such algorithms. So, we can say that this work has significant implications in the area of an attempt for designing parallel ARM algorithm based on candidate generation structure on heterogeneous cluster. It would provide the ground work for further exploration into areas said above. The data distribution strategies play an important role in load balancing on heterogeneous cluster. To achieve high performance on the heterogeneous PC cluster, an algorithm should be designed that minimizes the execution time, idle time, the communication cost between nodes of cluster and I/O network traffic.

REFERENCES

1. Han Jiwawei, and Kamber Micheline, "Data Mining: Concepts and Techniques", Book, Published by HarCourt India Pvt. Ltd., New Delhi, Academic Press, Morgan Kaufmann Publisher, 2001, pp. 5-7, 9, 15, 228-229, 246, 269.
2. Zaki M. J., "Parallel and Distributed Association Mining: A Survey, Concurrency", IEEE, Special Issue on Parallel Mechanisms for Data Mining, Volume 7, Issue 4, 1999, 14-25.
3. Ayan Necip. Fazil, "Updating large itemsets with early pruning", Thesis for Master of Science, submitted to the dept. of Computer Engineering & Information Science & The Institute of Engineering & Science of Bilkent University, 1999.
4. Pujari Arun K , "Data Mining Techniques", Book, Universities Press (India) Ltd., Hyderabad, First Edition, 2001, pp. 45, 47, 69,74.
5. Dunham Margaret H., Xiao Yongqiao, Gruenwald Le, and Hossain Zahid, "A Survey of association rules", available online at http://ww2.cs.uh.edu/~ceick/6340/grue_assoc.pdf, published in 2008.
6. Linuxmpich, available online at http://linux.about.com/cs/linux101 /g/mpich.htm; access date 02/05/2011.
7. Lusk Rusty, MPI and MPICH on clusters, available online at http://www.csm.ornl.gov/JPCA/rusty-jpc4.ppt; access date: 05/05/2011.
8. Wikipedia, Parallel computing, article published in Wikipedia, available online at http://en.wikipedia.org/wiki/Parallel_computing, access date: 05/01/2011.
9. MPICH2_flyer, available online at http://www.cels.anl.gov/events /conferences/SC07/presentations /mpich2-flyer.pdf, access date: 05/05/2011.
10. MPICH2, available online at http://www.mcs.anl.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

gov/research/projects/mpich2/index.php; access date: 04/05/2011.
11. Agrawal Rakesh and Shafer John C., "Parallel Mining of Association Rules", IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, December 1996, pp. 962-969.
12. Zaki M. J., "Parallel and Distributed Association Mining: A Survey, Concurrency", IEEE, Special Issue on Parallel Mechanisms for Data Mining, Volume 7, Issue 4, pp. 14-25, 1999.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)