Hardware Magnified Association Rule Mining for Hash Table Filter using MD5 Algorithm

M.Preethi[1], P.K.Mangaiyarkarasi[2]
[1]Research Scholar, Master of Philosophy in Computer Science,
[2]Associate Professor, Department of Computer Science,
Kongu Arts and Science College, Erode-638107.

Abstract: Data Mining is the extraction of hidden information from large databases called knowledge discovery in databases (KDD). Association Rule is used to detect the frequent item sets and also useful for discovering relationship among items from large databases. As the hardware's capacity is constant, the number of candidate itemsets is increased than the capacity of the hardware. It will create performance blockage. The hash-based and pipelined (HAPPI) architecture is to compare itemsets with the large databases and to decrease the number of candidate itemsets. HAPPI architecture includes 3methods such as Systolic array, Trimming filter and Hash table filter. Systolic array method is used to compare candidate itemsets with databases and then minimum support count value can be calculated. Trimming filter is used to decrease the items from each transaction. Hash table filter is used to decrease the itemset by finding or detecting duplicate records in the large databases. In hash table filter, MD5 algorithm is to be implemented. MD5 algorithm is mainly used to decrease the duplicate databases in the hardware. MD5 also proposed for other applications, where a large sized file must be compressed in a secure manner before being encrypted with a secret key under a public-key cryptosystem.

***Index Terms*: Association Rule, HAPPI Architecture, Systolic Array, Trimming Filter, Hash Table Filter.**

I. INTRODUCTION

Data mining is an extraction of information or patterns from data in large databases. It is related to the data analysis and the use of software techniques for identifying patterns and regularities in sets of data. Mining data is a component of a wider process called knowledge discovery from databases. Association rule mining finds interesting association and/or correlation relationships among large set of data items. Association rules show attributes value conditions that occur frequently together in a given data set.

Association Rule Mining was introduced because of its importance in data mining. The Association Rule Mining problem can be defined as the following. Let $A = \{a_1, a_2, \ldots\ldots, a_n\}$ be a set of items and $I = \{i_1, i_2, \ldots\ldots, i_m\}$ be a set of transactions, where each transactions $i_i \in I$ is a set of items that is $i_i \subseteq A$.

An Association Rule can be denoted as $A \Rightarrow B$. Many definitions have been introduced to describe the strength of the relationship between item sets A & B. There are different types of association rule levels. They are Support level, Confidence level and Interest level.

1) Support level is the percentage of transactions in the database that contain both *X* and *Y*.
2) Confidence level is the percentage of transactions containing *Y* in transactions those contain *X*.
3) Interest level represents a test of statistical independence.

Apriori algorithm is used to generate frequent item sets and confident association rule mining algorithm called AIS, which was given together with the introduction of this mining problem. The candidate generation process and pruning process are very important parts of this apriori algorithm uses at every iteration.

Apriori finds frequent item sets by scanning a whole database to check the frequencies of candidate item sets, which are originated by combining frequent item sets. However, Apriori-based algorithms have undergone some hindrance because they have too many candidate item sets.

Apriori-based hardware schemes require loading the candidate item sets and the database into the hardware. Capacity of the hardware is fixed, so the number of item in the database is larger than hardware separately. Therefore, the process of comparing candidate item sets with the database needs to be executed so many times. In addition, numerous candidate item sets and a huge database may cause a hindrance in the system.

In Fig.1, The HAPPI (Hash-based and Pipelined) architecture consists of three modules in this system. 1) **Systolic array** method is used to compare candidate item sets with databases and also finds the minimum support count value can be calculated. 2) **Trimming filter** is used to decrease the items from each transaction in a database simultaneously. 3) **Hash table filter** is used to decrease the item sets by detecting duplicate records in the large databases.

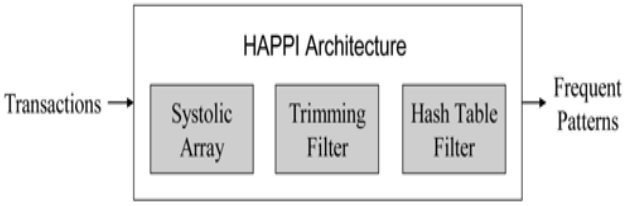

Fig. 1: HAPPI System Architecture.

II. RELATED WORKS

Ying-Hsiang Wen, Jen-Wei Huang, Ming-Syan Chen [1] in this paper, as the database is larger than hardware capacity and it uses HAPPI architecture to solve the bottleneck problem. HAPPI architecture uses pipelining methodology to compare itemsets with the database and collect useful information for reducing the number of candidate itemsets. Author of this paper implement Direct Comparison (DC) method and also implement a software algorithm DHP. The SPA (Systolic Process array) architecture is proposed to perform k-means clustering.

Mohammed J.Zaki, Karam Gouda [6] in previous paper, they used vertical mining algorithm for association mining which is very effective. They used transaction ids (Tids) and automatic pruning of irrelevant data. The main problem of tid is become too large for memory. So, they proposed Diffset novel vertical data representation.

Mamatha Nadikota, Satya P Kumar Somayajula, **Dr C.P V.N.J Mohan Rao [3]**, for association rules apriori algorithm is used to find the candidate itemset. Capacity of the hardware is fixed, so the number of item in the database is larger than hardware capacity. So the items are loaded in the hardware separately, due to this the time complexity is more to load candidate itemsets. So the HAPPI architecture with 3 modules has been proposed. They are systolic array, trimming filter and hash table filter. In this paper, two modules (i.e) systolic array and trimming filter is implemented.

S. Ayse Ozel and H. Altay Guvenir [2] In this paper, the hash table filter and trimming filter is used to decrease the size of the candidate item sets at each step of the transactions. PHP (Perfect Hashing and Pruning) is to form a hash table for the candidate item sets. PHP is a hash table which contains the actual counts of the candidate item sets. The DHP and PHP algorithm to solve the issue of mining association rules among items in a large database for transactions like sales and so on. The problem of discovering large item sets has been solved in the past by constructing a candidate set of item set. Then used for identifying the problem within the candidate set and those item set that meet the large item set requirements. The Hash-based algorithm is proposed for the candidate set generation.

Jong Soo Park, Ming –Syan Chan and Philip S.Yu [5] in this paper, author proposed to solve the issue of mining association rules among items in a large database of sales transactions. The problem of discovering large itemsets has been solved in the past by constructing a candidate set of itemset first and then, identifying with in this candidate set, those itemset that meet the large itemset requirements. In this paper, hash-based algorithm is proposed for the candidate set generation.

III. HAPPI ARCHITECTURE

In HAPPI architecture, the pipelined methodology is mainly using for comparing item sets with the database and also gathers useful information for decreasing the number of candidate item sets and items in the database at the same time. Therefore, we can decrease the incidence of loading the database into the hardware. The HAPPI architecture solves the hindrance problem using the Apriori-based database schemes. To solve this hindrance problem in association rule mining, here proposed the following five procedures in the HAPPI architecture: support/confidence counting, transaction trimming information, hash table filter building, candidate generation and candidate pruning.

There are many experiments conducted to evaluate the performance of HAPPI architecture. The experiment result itself shows that the HAPPI architecture outperforms the previous approaches on an execution time significantly and provides better accuracy level, particularly when the number of items in the database is bigger than the minimum support count value increases. The HAPPI architecture provides better performance than the previous approaches.

Apriori-based hardware schemes have to load the candidate item sets and the database into the hardware to execute the comparison process. Too many candidate item sets and a large database would definitely cause the performance bottleneck. To solve the above problem, the HAPPI architecture which contain of three modules that are systolic array, trimming information filter and hash table information filter have been proposed. The pipeline methodology is associated into the HAPPI architecture to perform the pattern matching in order to decrease the number of candidate item sets and items in the database at the same time. Hence, HAPPI architecture effectively solves the hindrance problem. Here, there are five procedures in the HAPPI architecture using the three hardware modules. The procedures are support counting, transaction trimming, hash table building, candidate generation and candidate pruning.

A) Pipeline Design

The transaction trimming and the hash table filter procedures are blocked by the support counting procedure. The transaction trimming procedure has been collected to evaluate the trimming information to execute the trimming process. This trimming evaluation process cannot be completed until the support count procedure compares all the transactions with all the candidate item sets in a database simultaneously. Once when the trimming filter transactions trimmed, the hash table filter procedure has to get the trimmed transactions from the trimming filter.

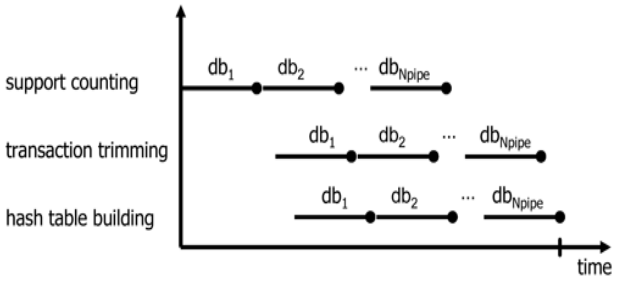

Fig. 2: The Pipeline procedures.

B) Transaction Trimming

The database in the transaction does not contain all the useful information. Because of the database size is too large it is difficult to extract the knowledge from the database. Likewise, each and every transaction in the database is not very helpful for getting frequent item sets. Hence the transaction trimming trims the items having less frequent count.

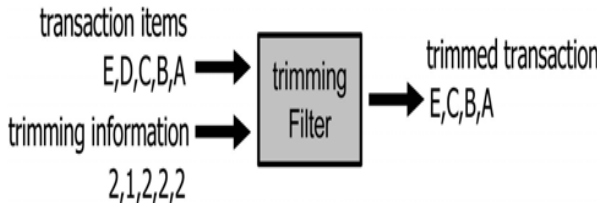

Fig. 3: The Trimming filter

C) Hash Table Filtering

Hash table filter is used to detect the duplicate records in a large databases or files.

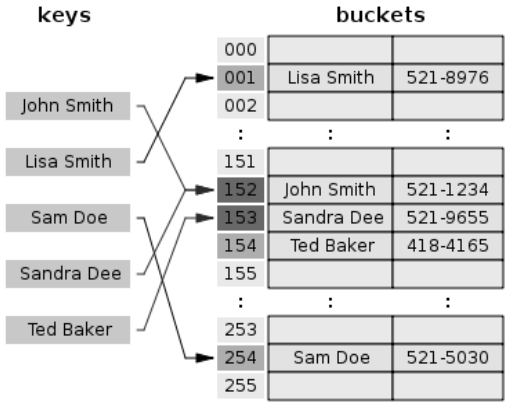

Fig. 4: Hash table building

The hash value generator and hash table updating module were used to build the hash table filter.

IV MD5 ALGORITHM

The MD5 algorithm is used to decrease the duplicate records in a large database in hash table filter. MD5 takes message of any random length as input and produces fixed length as output. MD5 algorithm is proposed for digital signature applications, where a large file must be compressed in a secure manner. Likewise, most of the hash table functions; MD5 is not both an encryption and encoding.

Security

The MD5 hash table function security is roughly compromised in initial stages. A collision attack can have the capacity to find collisions within seconds on a computer. This attack is to find the two inputs which can produce the ditto hash value is known as hash collision. There are two types in collision attacks 1) Classical collision attack 2) Chosen-Prefix collision attack.

In Classical collision attack, the attacker does not control the message, but they are promptly chosen by the algorithm. In Chosen-Prefix collision attack, the attacker can choose two promptly different documents, and then attach different calculated values that result in the various documents. Chosen-Prefix collision attack is more powerful than comparing to the Classical collision attack. These hash attacks and collision attacks have been determined in the public in different situations. MD5 algorithm has been used to save a hash password.

V EXPERIMENTAL RESULTS

In Hash table filter, the MD5 algorithm is used to detect the duplicate records and to reduce the size of the database, because the hardware capacity is fixed. The Hash Table Filter uses MD5 algorithm used to reduce the database capacity in hardware respectively.

To build a hardware hash table filter, we use a hash value generator in MD5 algorithm. This generating all the k-itemset combinations of the transactions and puts the k itemsets into the hash function to create the corresponding hash values. The hash value generator comprises a transaction memory, a state machine and index array. The transaction memory stores all the items of a transaction. The state machine is the controller that generates control signals. Then, the control signals are fed into the index array. By changing the values in the index array, the state machine can reduce the duplicate database in reduced time than HAPPI architecture.

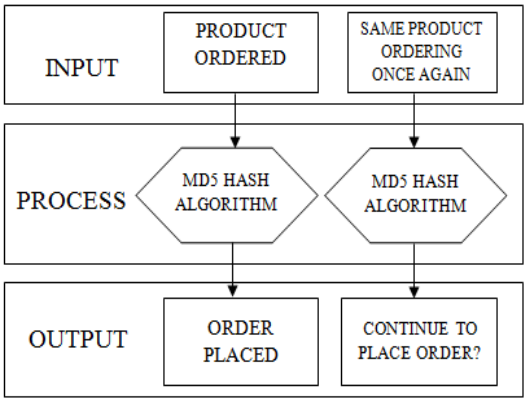

Fig. 5: MD5 in Hash Table Filter

MD5 Process

MD5 processes a variable-length message as input into a constant-length output. The input message is divided into small pieces of blocks. The padding process is as follows:

Step1: A single bit,"1" is added to the end of the message and then "0" bits are added so that the length in bits of the filled message becomes appended.

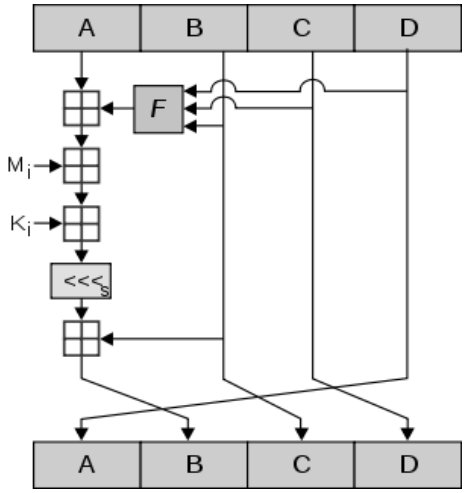

Fig. 6: MD5 operation.

Step2: After the padding with bits process completes, the resulting message has a length that is a definite multiple. The input message will have a length that is a definite multiple.

Step3: A temporary storage is used to count the message digest. Each of temporary storage is a known as register.

Step4: Process the message as word blocks.

This diagram represents that one operation within a round. There are four possible functions: each round used a different one. It denotes XOR, AND, OR and NOT operations.

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$
$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$
$$H(B,C,D) = B \oplus C \oplus D$$
$$I(B,C,D) = C \oplus (B \vee \neg D)$$

The following is the graph of comparison between the HAPPI architecture and MD5 algorithm using the dataset get from the experimental results:

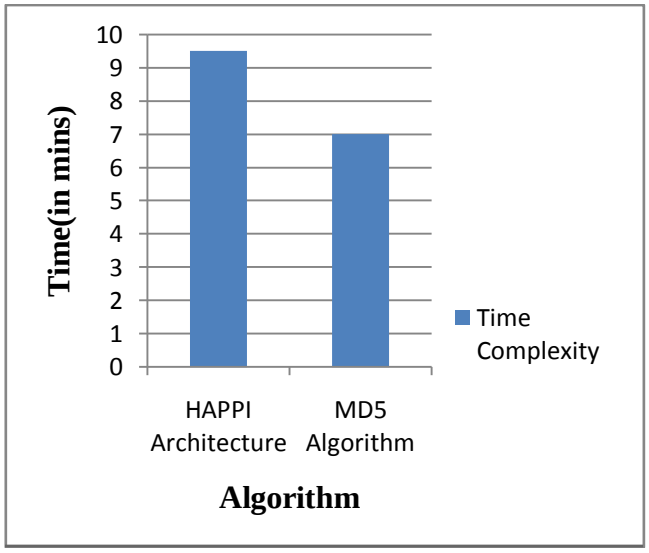

Fig. 7: Comparison of Time Complexity between HAPPI and MD5 algorithm applied on sample data

VI CONCLUSION AND FUTURE WORK

The hashing and pipelining technique for association rule mining had some flaws in it. The HAPPI architecture has three modules which are used to decrease the candidate itemsets. The existing method used only systolic array and trimming filter to decrease the candidate item sets in the databases. It has a performance blockage and time complexity in decreasing the item sets in the database and data in the database simultaneously. So, hash table filter is used to decrease the candidate item sets and detect the duplicate records in the large files.

Secure hash function (SHA) is a main hash function which is used to decrease the candidate item sets in future work. SHA has many versions such as SHA-1, SHA-2 and SHA-3. BLAKE and BLAKE2 are other algorithm is a family of hash functions which also can be used for future purposes.

REFERENCES

1) Ying-Hsiang Wen, Jen-Wei Huang, Ming-Syan Chen, "Hardware-Enhanced Association Rule Mining with Hashing and Pipelining", IEEE Transactions on Knowledge and Data Engineering, Vol.: 20 Issue: 6, June 2008.
2) S. Ayse Ozel and H. Altay Guvenir, "An Algorithm For Mining Association Rules Using Perfect Hashing And Database Pruning", Bilkent University, Department of Computer Engineering, Ankara, Turkey.
3) Mamatha Nadikota, Satya P Kumar Somayajula, Dr. C. P. V. N. J. Mohan Rao, "Hashing and Pipelining techniques for Association Rule Mining", International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011.
4) Manisha Bhargava, Arvind Selwal, "Association Rule Mining using Apriori Algorithm: A Review", International journal of Advanced Research in Computer Science, Review Article, Volume 4, No. 2, Feb 2013.
5) Jong Soo Park, Ming –Syan Chan and Philip S.Yu, "An Effective Hash-based algorithm for Mining Association Rules" IBM Thomas J. Watson Research Center. Yorktown Heights, New York 10598.
6) Phani Ratna Sri Redipalli, G. Srinivasa Rao, "Hardware Enhancement Association Rule with Privacy Preservation" International Journal for Computer Technology and Applications, Volume 2 (5).
7) Mohammed J.Zaki, Karam Gouda, "Fast Vertical Mining using Diffsets", 2003.
8) Mr.Praveen S Patil, "Hardware-Enhanced Association Rule Mining with Hashing and Pipelining", International Journal of Combined Research and Development, Volume:1 Issue:1.
9) H. Mannila, H. Toivonen, and A. I. Verkamo, "Efficient Algorithms for Discovering Association Rules", Proceedings of the AAAI Workshop on Knowledge Discovery in Databases, Usama M. Fayyad and Ramasamy Uthurusamy (Eds.), Washington, pp. 181-192, (July 1994).
10) R. Srikant and R. Agrawal, "Mining Generalized Association Rules", Proc. of the 21st VLDB Conference, Zurich, Switzerland, (1995).
11) R. Agrawal, and R. Srikant, "Fast Algorithms for Mining Association Rules", Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, (Sept. 1994).