ENHANCED RECONFIGURABLE WEIGHTED ASSOCIATION RULE MINING FOR FREQUENT PATTERNS OF WEB LOGS

SP. Malarvizhi [1), B. Sathiyabhama [2)

[1) AVS Engineering College, Salem, India, malarvizhisp@gmail.com,
[2) Sona College of Technology, Salem, India, sathiyabhama@sonatech.ac.in

Abstract: Systolic tree structure is a reconfigurable architecture in field-programmable gate arrays which provide performance advantages. It is used for frequent pattern mining operations. High throughput and cost effective performance are the highlights of the systolic tree based reconfigurable architecture. Frequent pattern mining algorithms are used to find frequently occurring item sets in databases. However, space and computational time requirements are very high in frequent pattern mining algorithms. In the proposed system, systolic tree based hardware mechanism is employed with Weighted Association Rule Mining (WARM) for frequent item set extraction process of the Web access logs. Weighted rule mining is to mine the items which are assigned with weights based on user's interest and the importance of the items. In the proposed system, weights are assigned automatically to Web pages that are visited by the users. Hence, systolic tree based rule mining scheme is enhanced for WARM process, which fetches the frequently accessed Web pages with weight values. The dynamic Web page weight assignment scheme uses the page request count and span time values. The proposed system improves the weight estimation process with span time, request count and access sequence details. The user interest based page weight is used to extract the frequent item sets. The proposed system will also improve the mining efficiency on sparse patterns. The goal is to drive the mining focus to those significant relationships involving items with significant weights. *Copyright © Research Institute for Intelligent Computer Systems, 2014. All rights reserved.*

Keywords: Frequent pattern mining, FPGA, reconfigurable architecture, systolic tree, Automatic weight estimation, WARM, Web logs.

1. INTRODUCTION

Items commonly appearing together in a Transactional database can be determined using frequent pattern mining. In the research literature to mine the frequent patterns, algorithms like Apriori, FP-growth, DynFP-growth, FP-Bonsai (The art of growing and pruning small FP trees) [15] and AFOPT [16] are used. The FP-growth algorithm stores all transactions in the database as a tree using two scans [1, 11]. It uses recursion to traverse the tree and mine patterns. It is difficult to implement recursive processing directly in hardware, as dynamic memory allocation typically requires some software management. For this reason, the dynamic data structures such as linked lists and trees are widely used in software implementations and these are very rarely used in a direct hardware implementation. Consequently, it would be very difficult to directly translate this FP-growth algorithm into hardware.

FPGAs (Field Programmable gate arrays) referred to as "programmable logic", are general-purpose hardware Silicon chips. In contrast to ASICs (application-specific integrated circuits), FPGAs have no pre-determined functionality. But they can be configured to implement arbitrary logic by combining gates, flip flops and memory banks. FPGAs can provide performance advantages with lower resource consumption (e.g., energy) and with low cost than conventional CPUs. Existing reconfigurable systolic tree architecture for frequent pattern mining describes a prototype using FPGA platform. Advantage of FPGAs over traditional computing platforms is the ability to parallelize algorithms at the operand-level granularity, as opposed to the module-level or higher [2]. Hence they minimize reliability concerns with true parallel execution and deterministic hardware dedicated to every task. FPGA-based approach can be several times faster than a software implementation of the FP-growth algorithm. The adoption of FPGA technology continues to increase as higher-level tools evolve to deliver the benefits of reprogrammable silicon to engineers and scientists at all levels of expertise.

A systolic tree is an arrangement of pipelined processing elements (PE) in a multidimensional tree pattern [3]. The significant features of systolic tree based reconfigurable architecture are high throughput and faster execution. The systolic tree is mapped in FPGA hardware which is similar to the FP-tree as used in software. The transaction items are updated into the systolic tree with candidate item matching and count update operations. Due to the limited size of the systolic tree, a transactional database must be projected into smaller ones each of which can be mined in hardware.

Association rule mining (ARM) is one of the most essential types of knowledge representation and pattern extraction which brings out the implicit relationships among the items present in large number of transactions. It finds out the association rules that satisfy the predefined minimum support and confidence from a given database.

Traditional model of ARM is adapted to handle WARM problems where each item is allowed to have a weight [4]. The goal of using weighted support is to make use of weights in the mining process. Introduction of weights during mining is to prioritize the selection of target itemsets according to their significance in the dataset, rather than measuring frequency (the rate of occurrence of item sets) alone.

The application of data mining techniques on the Web is now the focus of an increasing number of researchers. Web mining discovers information from Web documents and services. Web log mining is a promising tool to study user behaviors, which could further benefit web-site designers with better organization and services. Proposed system is designed to perform weighted rule mining for Web logs and automatic weight estimation scheme is used. Each Web page is assigned a weight value with reference to the request count and access sequence. Enhanced systolic tree based approach is used in the proposed system to mine frequent item sets in Web log data as shown in Fig. 1.
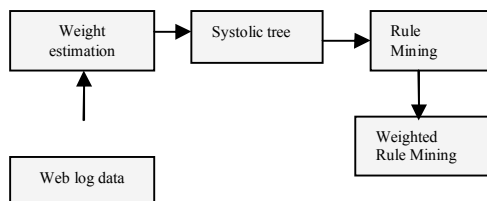

Fig. 1 – Enhanced systolic tree approach for Web log data.

Rest of the sections is organized as follows. Section 2 discusses related works of the systolic tree based rule mining systems. Web-log analysis and

weight estimation process is described in Section 3. The design and construction of a systolic tree, mining process and need for database projection is discussed in Section 4. In Section 5 is discussed the enhanced systolic tree based weighted rule mining approach. In Section 6, experimental evaluation is presented and the comparison for memory requirement and efficiency analysis between systolic tree based rule mining and systolic tree based weighted rule mining is given. Finally the conclusion and future enhancement is given in Section 7.

2. RELATED WORK

Many efforts have been made in hardware-based frequent pattern mining algorithms in recent days. A parallel implementation of the Apriori algorithm on FPGA [5] focuses on significantly reducing the processing time, through the use of a new extension to the systolic array architecture. Due to the processing time involved in reading the transactional database multiple times, the hardware implementation is 4 times faster than the fastest software implementation. The authors of parallel FPGA further explored the parallelism and developed a bitmapped CAM architecture which provides 24 times performance gain over the software version [6].

More number of candidate itemsets and a large database would create a performance bottleneck incurred in Apriori based hardware schemes. The Hash based and Pipelined (HAPPI) architecture proposed in [7, 10] for hardware enhanced rule mining uses the pipelining methodology which outperforms the Apriori architecture introduced by Z. K. Baker and V. K. Prasanna [5] when the number of different items and the minimum support values are increased.

Narayanan et al, portrayed hardware architecture for a Decision Tree Classification (DTC) algorithm, and showed that optimizing the Gini score computation significantly increases the overall performance [8]. Among the several solutions developed, DTC is a popular method that yields high accuracy while handling large datasets.

The classical models ignore the difference between the transactions, and the weighted association rule mining does not work on databases with only binary attributes. A new measure using link-based models called w-support takes the quality of transactions into consideration rather preassigned weights [9].

Web log files are analyzed to get the information that are useful for improving the services offered by Web portals and information access and retrieval tools, giving information on problems occurred to

the users. Mining frequent patterns from Web log data can help to optimize the structure of a Web site and improve the performance of Web servers. Web users can also benefit from these frequent patterns. Frequent pattern mining techniques are discussed for discovering different types of patterns in a Web log [13]. For obtaining frequent access patterns from the web log data and to provide valuable information about the user's interest, FP-growth algorithm is used [17]. A new algorithm called Combined Frequent Pattern Mining (CFPM) is proposed by Liping Sun and Xiuzhen Zhang [14] to cater for Web log data specifically.

An efficient mining methodology for WARM has been proposed [12] by Wei Wang et al. WARM assigns numerical attribute for every item. This judges the weight of the item in a particular weight domain. The issues of discovering significant binary relationships in transaction datasets in a weighted setting is addressed in [4] where each item is allowed to have a weight. This uses the idea of WARM [12] to bring out an improved one. This algorithm is both scalable and efficient in discovering significant relationships in weighted settings.

3. WEB-LOG ANALYSIS AND WEIGHT ESTIMATION

3.1. WEB-LOG ANALYSIS

Web mining focuses on automatically discovering information and knowledge through the analysis of Web contents, Web structure and Web usages. Since the Web is huge, heterogeneous and dynamic, automated Web information and knowledge discovery calls for novel technologies and tools, which may take advantage of the state-of-the-art technologies from various areas, including machine learning, data mining, information retrieval, database and natural language processing [13].

Web mining techniques are used to analyze the Web information resources. Web content mining and structure mining methods are used to analyze the Web page contents. The user access information is maintained under the Web logs in Web server environment. Usage mining techniques are used to analyze the Web logs to extract user access patterns. The Web content management and link connectivity are improved using the access patterns [14].

The Web page request details are maintained in the Web logs. Page URL, IP address, requested time, session ID and span time are updated for each page request in the log list.

The systolic tree based rule mining system uses the Oracle relational database to store the service and access log details. Each page request is updated as a separate entry with unique session ID for the user. Part of the Web log data from a host site is loaded into the database and this migration of data in the database is called as data populate process. This process takes a time slot based on the number of records to be populated in the database. The second step is the data cleaning process where noisy and redundant data are all removed. At the end of the cleaning process, session conversion which arranges all page requests of a single session ID into a single transaction is arrived. Access sequence details provide the sequence of pages requested in terms of page URL with respect to requested time for every individual session ID of an IP address. Table 1 shows a piece of such access sequence detail arrived for an institutional Web log data. Session details provide along with IP address the URL of each page visited and the number of pages visited for every individual session ID. The total number of sessions for a particular duration in the Web log and the respective total number of pages for all the sessions are also arrived in the session details. A piece of session detail is shown in Table 2. This session information is used in the rule mining process.

Table 1. Access sequence details (for one session ID of an IP address)

Sl.No	Requested Time	Page URL
1.	01/March/2011 01:43:23	/ece.html
2.	01/March/2011 01:49:36	/educontactus.html
3.	01/March/2011 01:51:09	/eee.html
4.	01/March/2011 01:54:09	/college%20brochure/brochure.html

Table 2. Session details

Sl. No.	Session ID	IP Addr.	Page1	Page2	--	Page n
1.	2257103	50.90.19.76	/civil.html			
2.	2257104	67.202.41.3	/rank.html	/edufeedba..		
3.	2257105	115.98.7.131	/rank.html	/chairmansmessa….	--	/toobeauti…
4.	2257106	117.193.226...	/eee.html			
5.	2257107	66.249.71.115	/admission.html	/gallery.html	----	

The weight estimation process for each page URL is carried out with span time, request count and access sequence details.

3.2. WEIGHT DEFINITIONS

Transaction is a set of weighted items, each of which may appear in multiple transactions with different weights [4].

3.2.1. WEIGHTED ATTRIBUTES

A ($a_1, a_2, .. , a_k$) are variables selected to calculate weights. Depending on the domain, there could be any variable ranging from item's price in a supermarket domain to visitor page dwelling time in a Web log mining domain. There are two types of weights namely the *item weight* and the *item set weight*.

3.2.2. ITEM WEIGHT

Item weight is a value attached to an item representing its significance and denoted as w*(i)*. For example in the Web log mining setting where each item is a page visited in a click-stream/transaction, the weight can be related to a users average dwelling time on that page. In other words, the item weight is a function of selected weighting attributes therefore denoted as *w(i) = f(a)*.

3.2.3. ITEMSET WEIGHT

Based on the item weight w(i), the weight of an itemset, denoted as w(is), can be derived from the weights of its enclosing items. One simple way is to calculate the average value of the item weights, denoted as:

$$w(is) = \frac{\sum_{k=1}^{|is|} w(i_k)}{|is|}, \qquad (1)$$

Item weight is a special itemset weight when the itemset has only one item.

3.2.4. TRANSACTION WEIGHT

Transaction weight is a type of itemset weight. It is a value attached to each of the transactions. Usually the higher the transaction weight, the more it contributes to the mining result.

In the proposed system, weights are assigned for the Web pages. Automatic weight estimation scheme is used in this system. Each Web page is assigned a weight value with reference to the request count and sequence. Access sequence based weight estimation model is used in the system. The weight values are used in the weighted rule mining process.

3.2.5. WEIGHTING SPACE

Weighting space WS is the context within which the weights are evaluated

1) *Inner-transaction* space *WSt:* this space refers to the host transaction that an item is weighted in.

2) *Item space WS_I*: this space refers to the space of the item collection that covers all the items appears in the transactions.

3) *Transaction* space *WS_T*: This space is defined for transactions rather than for items.

3.2.6. WEIGHTED SUPPORT

Weighted support *WSP* of an itemset. A set of transactions T respects a rule R in the form A➔B, where A and B are non-empty sub-itemsets of the item space I and they share no item in common. Its weighted support is the fraction of the weight of the transactions that contains both A and B relative to the weight of all transactions. This can be formulated as:

$$wsp(AB) = \frac{\sum_{K=1}^{|WS_T|\&(AB)\subseteq t_k} Weight(t_k)}{\sum_{K=1}^{|WS_T|} weight(t_k)}, \qquad (2)$$

The transaction weight (t_k) is derived from weights of the items presented in the transaction. It is the average weight of the items presented in the transaction. $WS_t(t_k)$ is the inner-transaction space for the k_{th} transaction in transaction space WS_T.

$$weight(t_k) = \frac{\sum_{i=1}^{|WS_t(t_k)|} weight(item(i))}{|WS_t(t_k)|}, \qquad (3)$$

This value is substituted in equation (2) to calculate the weighted support of a potentially significant itemset.

3.3. ACCESS LOG BASED WEIGHT ESTIMATION

The access log based weight is estimated using the Web page access information for a Web site and Weight definitions. The access log maintains the session id, IP address, page URL and requested time details. Single entry is maintained for each page hit. The Web server assigns the session at the time of user entry into a Web site. The unique session id is used to identify the pages that are accessed by the same user under the same session period. The access log details are converted into session list. The session list is prepared by grouping all the pages that are accessed in the same session. The session list maintains single entry for each session. The pages that are visited in a session are arranged in order of their requested sequence. The session sequence table is used to estimate the access log based weights for

the Web pages. Fig. 2 shows the pseudo code which describes the access log based weight estimation process.

```
Begin
  for i=1 to pc
  alwi=0.0;
  twi=0.0;
  for j=1 to 10
  wj=pscij/tpcj;
  twi=twi+wj;
  next
  alwi=twi/10;
  next
End
```

Fig. 2 – Weight Estimation algorithm.

The access log based page weight is estimated with the page sequence count and total page count information. In the above algorithm **alw** is access log weight, **psc** is the page sequence count, **tpc** is total page count for each request sequence and **tw** refers the sum of intermediate weights. "i" in the algorithm refers to rows i.e. each page URL and "j" refers to columns i.e. request sequence (for a fair count of 10). The algorithm may be extended for a maximum of n number of pages visited by an user. The access log weight is assigned for each page in the Web site.

4. SYSTOLIC TREE, PATTERN MINING AND DATABASE PROJECTION

4.1. DESIGN AND CREATION OF SYSTOLIC TREE

Systolic tree structure is used for frequent pattern mining. In VLSI terminology it is an arrangement of pipelined processing elements (PEs) in a multidimensional tree pattern. It is configured to store the support counts of the candidate patterns in a pipelined fashion. Given a transactional database, the relative positions of the elements in the systolic tree should be the same as in the FP-tree [3]. The transaction items i.e. the Web page request sequence is updated into the systolic tree with candidate item matching and count update operations.

A systolic tree structure consists of the following PEs:

1. **Control PE**. The root PE of the systolic tree does not contain any item. Any input/output data of the systolic tree must go through it first. One of its interfaces connects to its leftmost child.

2. **General PE**. All other PEs are general PEs. Each general PE has one bidirectional interface connecting to its parent. The general PE which has

children has one interface connecting to its leftmost child. The general PEs have siblings may have an interface connecting to its leftmost sibling. They may contain an item and the support count of the stored item.

Each PE has a level associated with it. The control PE is at level 0. The level of a general PE is equal to its distance to the control PE. The children of a PE have the same level.

Each general PE has only one parent which connects to its leftmost child directly. The other children connect to their parents indirectly through their left siblings. A systolic tree which has W levels with K children for each PE has $\sum_{i=0}^{w} K^i$ **PEs.**

A PE has three modes of operation: WRITE mode, SCAN mode and COUNT mode [3]. The systolic tree is built using the WRITE mode Algorithm. Input items are streamed from the root node in the direction set by the defined WRITE mode algorithm. The support count of a candidate itemset is extracted in both SCAN mode and COUNT mode. This process is called as candidate itemset matching.

4.2. PATTERN MINING USING SYSTOLIC TREES

To mine frequent patterns in the systolic tree, a collaborating hardware cum software platform is required. The software sends a candidate pattern to the systolic tree [2]. Once all items in a candidate itemset are sent to the systolic tree, a control signal signifying the COUNT mode is broadcasted to the whole systolic tree. After some clock cycles, the systolic tree sends the support count of the candidate pattern back to the software. The software compares the support count with the support threshold and decides whether the candidate pattern is frequent or not. After all candidate patterns are checked with the support threshold in software, the pattern mining is done. The approach to get the support count of a candidate pattern is called candidate itemset matching. The main principle of matching is that any path containing the queried candidate itemset will be reported to the control PE.

4.3. NEED FOR DATABASE PROJECTION

It is not always possible that a tree-based representation will fit in the available hardware resources (either memory or logic) for any given database. If the case is such that the memory or logic is not large enough to hold the whole tree, it is necessary to split the database into multiple smaller databases with fewer frequent items. If the technique of database projection is not used, then candidate itemset matching will take enormous time when the number of frequent patterns is large. Each of the

projected database has no more than N=min(K,W) frequent items and is guaranteed to fit into the FPGAs [3].

5. ENHANCED-SYSTOLIC TREE AND WARM

A number of data mining algorithms have been introduced to the community that perform summarization of the data, classification of data with respect to a target attribute, deviation detection and other forms of data characterization and interpretation.

Applications of association rule mining include cross marketing, attached mailing, catalog design and customer segmentation. An association rule discovery algorithm searches the space of all possible patterns for rules that meet the user-specified support and confidence thresholds. Looking back at Fig. 3, we have shown the process involved in constructing the Enhanced-Systolic tree (A tree which possesses the items with weights) and Weighted Association Rule Mining based on it, in the form of a Data Flow Diagram (DFD). For comparison and evaluation purpose, association rule mining is also performed from ordinary systolic tree constructed before weight assignment.
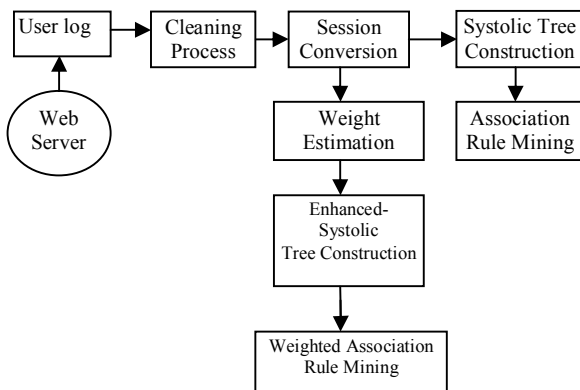

Fig. 3 – Enhanced-Systolic tree based WARM (DFD).

5.1. Weighted Association Rule Mining (WARM)

An itemset is denoted large if its support is above a predefined minimum support threshold. In the WARM context, we say an itemset is significant if its weighted support is above a pre-defined minimum weighted support threshold.

Support and confidence are used in association rule mining. Based on support threshold and confidence threshold, frequent patterns are mined. In WARM, itemsets are no longer simply counted as they appear in a transaction. This change of counting

mechanism has made it necessary to adapt traditional support to weighted support (use of equation 2). The goal of using weighted support is to make use of the weight in the mining process and prioritize the selection of target itemsets according to their significance in the dataset, rather than their frequency alone [4]. Hence, systolic tree based algorithm is enhanced with weighted rule mining concepts. The page weights (item weight) are used in the weighted rule mining process. Based on the estimated capacity of the systolic tree, the page request frequencies (count) and weight values are updated in the systolic tree for mining. The frequent patterns are extracted with the weight values. Hence, the weighted support and weighted confidence are estimated and the systolic tree is updated for mining weighted association rules.

The systolic tree based weighted rule mining for Web log involves the following steps.

- Step1: Populate user access log data into oracle database.
- Step2: Perform the data cleaning process with redundant data elimination.
- Step3: Prepare the session details from the cleaned log data.
- Step4: Estimate the page weights based on the page access sequence.
- Step5: Prepare the candidate sets using the attribute and their frequency values.
- Step6: Perform the capacity estimation process for the systolic tree.
- Step7: Update the itemsets and frequency values along with item (page) weights into the systolic tree.
- Step8: Estimate the weighted support and weighted confidence values.
- Step9: Fetch the rules using minimum support and minimum confidence.

6. EXPERIMENTAL EVALUATION

6.1. MATERIALS AND METHODS

The targeted device for simulation is a Xilinx Spartan-3E FPGA package interfaced with an Intel Core I5, 3.2 GHz processor machine with 2 GByte RAM and 500 GByte hard disk. Software requirements are Windows XP platform with front end C and back end Oracle 11g. Spartan 3E FPGA is a sequential and volatile device. Parallel FPGA implementation will be much faster than the sequential model used here which when combined with WARM gives much faster and accurate results. In parallel FPGA computer, thousands of operations can be performed in parallel during every clock cycle. The latest generation FPGA is a Xilinx Virtex7 FPGA, which is a parallel and nonvolatile

one that can perform thousands of tasks in parallel. It is highly performance oriented than the previous FPGA generations.

6.2. PERFORMANCE COMPARISON

Pages visited only once by only one visitor will have the least weightage and it is obviously not a frequent item. It will not be updated in the systolic tree for mining. This aids in reducing the memory space in additive and also improves the mining efficiency.

The performance analysis based on parameters such as memory requirement and runtime is carried out between the systolic tree based association rule mining (STARM) [3] and systolic Tree based WARM (STWARM) models. Evaluation on a Web log data of an educational institution which is considered for the data present in the Table 1 and Table 2 has given the results that the STWARM model reduces the memory requirement by 30 % than the STARM and the run time is reduced by 25 % in STWARM than the STARM. This is depicted in Fig. 4.
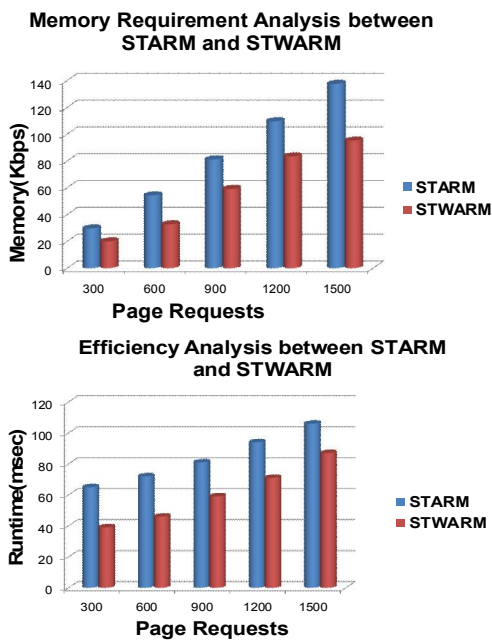

Fig. 4 – Performance analysis between STARM and STWARM for a Web log data.

STWARM makes use of the write, scan and count mode algorithms in STARM [3] for constructing systolic tree in hardware FPGA platform for mining the frequent patterns. STWARM also includes weights assigned to each page with the help of weight estimation algorithm (Fig. 2). Web log data is applied to STARM instead of a transactional database. The purpose of introducing the concept of weights in STWARM is

that it eliminates the irrelevant pages by mining frequent pages which fall above the weighted support threshold. In STARM, only support threshold is used to mine all the frequent pages reaching the threshold whereas STWARM mines only the relevant pages based on weights of the page.

Web usage mining carried out on Web log data is useful in many ways. The main purpose of this model is the usage in educational institutions to find out the unwanted pages visited by the students during a suspected duration which obviously spoils their progress. This means that, those few unwanted URLs which are mined with greater weighted support for that duration has the most weightage in spoiling the behavior of the students. The access of those URLs can be denied by the institution. This system may also mine the pages which are informative to the students, based on which future decisions can be made. Hence this model serves essential purpose for mining institutional Web logs, even though such a type of reconfigurable weighted rule mining can be implemented for Super market data as well for finding the most user preferable items, which have attracted them more. This helps in reordering the arrangement of items in shelves of super market based on weightage, so that they are well visible to the customers, which increases the sales. The Web logs in the servers of any industries or organization can also be mined for useful and most important information, based on which the organization can know about the behavioral pattern of their employees in order to keep them more involved and interactive.

7. CONCLUSION

The association rule mining techniques are used to extract frequent patterns. Systolic tree is used to organize candidate sets with frequency values. Due to the limited size of the systolic tree, a transactional database must be projected into smaller ones each of which can be mined in hardware efficiently. Systolic tree based rule mining scheme is enhanced for weighted rule mining process. The inability of ARM for treating units differently is solved by integrating weights in the mining process. The weighted rule mining techniques are used to fetch the frequently accessed Web pages with its weight values where page weights are used to denote the importance of the Web page. The proposed system improves the weight estimation process with span time, request count and access sequence details. The user interest based page weight is used to extract the frequent itemsets. Automatic weight estimation scheme is used in the system. Minimum weighted support and weighted confidence is used for frequent pattern

mining process to find the most interested pages in the Web site. WARM also scales up as the dataset size increases and it is used to improve the Web site contents.

Systolic tree based weighted rule mining system reduces the memory requirement by 30 % and run time by 25 % than systolic tree based rule mining system. The system proposed also improves the efficiency of sparse patterns. The system can be enhanced to mine frequent patterns on data streams, to perform parallel rule mining process, to support Web personalization and the usage mining process can be integrated with content mining model.

8. REFERENCES

[1] J. Han, J. Pei, Y. Yin and R. Mao, Mining frequent patterns without candidate generation: a frequent-pattern tree approach, *Data Mining and Knowledge Discovery*, (8) 1 (2004), pp. 53-87.

[2] S. Sun and J. Zambreno, Mining association rules with systolic trees, *Proceedings of the International Conference on Field-Programmable Logic and Applications (FPL'08)*, September 2008.

[3] S. Sun and J. Zambreno, Design and analysis of a reconfigurable platform for frequent pattern mining, *IEEE Transactions on Parallel and Distributed Systems*, (22) 9 (2011), pp. 1497-1505.

[4] F. Tao, F. Murtagh, M. Farid, Weighted association rule mining using weighted support and significance framework, *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'03*, Washington DC, USA, 24 - 27 Aug 2003, pp. 661-666.

[5] Z. K. Baker and V. K. Prasanna, Efficient hardware data mining with the apriori algorithm on FPGAs, Proceedings of the 13th Annual IEEE Symposium on Custom Field-Programmable Custom Computing Machines (FCCM'05), 2005, pp. 3-12.

[6] Z. K. Baker and V. K. Prasanna, An architecture for efficient hardware data mining using reconfigurable computing systems, Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'06), 2006, pp. 67-75.

[7] Y.-H. Wen, J.-W. Huang, and M.-S. Chen, Hardware-enhanced association rule mining with hashing and pipelining, *IEEE Transactions on Knowledge and Data Engineering*, (20) 6 (2008), pp. 784-795.

[8] R. Narayanan, D. Honbo, G. Memik, A. Choudhary, and J. Zambreno, An FPGA implementation of decision tree classification, Procideengs of the Conference on Design, Automation, and Test in Europe (DATE), April 2007, pp. 189-194.

[9] Ke Sun and Fengshan Bai, Mining weighted association rules without preassigned weights, *IEEE Transactions on Knowledge and Data Engineering*, (20) 4 (2008), pp. 489-495.

[10] Mamatha Nadikota, Satya P. Kumar Somayajula and C. P. V. N. J. Mohan Rao, Hashing and pipelining techniques for association rule mining, *International Journal of Computer Science and Information Technologies*, (2) 4 (2011), pp. 1448-1452.

[11] J. Han, J. Pei and Y. Yin, Mining frequent patterns without candidate generation, Proceeding Proceedings of the ACM SIGMOD International Conference on Management of Data SIGMOD'00, 2000, Paper ID: 196, pp. 1-12.

[12] W. Wang, J. Yang and P. Yu, Efficient mining of weighted association rules (WAR), Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2000, pp. 270-274.

[13] R. Iváncsy and I. Vajk, Frequent pattern mining in web log data, *Acta Polytechnica Hungarica*, (3) 1 (2006), pp. 77-90.

[14] L. Sun and X. Zhang, Efficient frequent pattern mining on web logs", J. X. Yu, X. Lin, H. Lu, and Y. Zhang (Eds.): APWeb 2004, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg, Vol. 3007, 2004, pp. 533-542, 2004.

[15] F. Bonchi and B. Goethals, FP-Bonsai, the art of growing and pruning small FP-trees, *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD'04*, Sydney, Australia, Springer-Verlag, Heidelberg, Germany, 2004, pp. 155-160.

[16] J. Gao, Realization of new association rule mining algorithm, *Proceedings of the IEEE International Conference on Computational Intelligence and Security CIS'07*, 15-19 December 2007, pp. 201-204.

[17] R. Mishra and A. Choubey, Discovery of frequent patterns from web log data by using FP-growth algorithm for web usage mining, *International Journal of Advanced Research in Computer Science and Software Engineering*, (2) 9 (2012), pp. 311-318.

SP.Malarvizhi *received the BE degree in Electrical and Electronics Engineering from Annamalai University in 1994 and the ME degree in Computer Science and Engineering from Anna University of Technology, Coimbatore in 2009. Currently she is Professor in the Department of Computer Science and Engineering at AVS Engineering College, Salem, India. She is a member of the ISEEE, ISTE India and NC member of Computer Society of India.*

B. Sathiyabhama *received her Ph.D degree in Computer Science and Engineering from National Institute of Technology, Tiruchirappalli in 2009. Currently she is a Professor and Head in the department of Computer Science and Engineering at Sona College of Technology, Salem, India where she has been since 1998. Her research interests include Bioinformatics, Compilers, Algorithm Analysis, Big Data Analytics and Data mining with a focus on optimization based clustering techniques and. She is a member of the IEEE, ISTE India and Computer Society of India.*