

5E: A Framework to Yield High Performance in Real-time Data Mining over the Internet

Allan K.Y. Wong and Richard S.L. Wu

Department of Computing, Hong Kong Polytechnic University

Hunghom, Kowloon, Hong Kong (SAR), P.R. China

Email: csalwong@comp.polyu.edu.hk Fax: (852) 27740842

Abstract

The aim of the 5E (5 essentials) framework proposed in this paper is to yield high performance in real-time data mining over the Internet. The essentials are: a) realize the Internet power with object-based parallelism; b) select an inter-object interaction pattern suitable for the problem; c) apply the correct programming model to cut communication overhead; d) equip program objects with mobility for better system performance; e) work with the correct hardware architecture for timeliness. It was verified by different experiments over the Internet that the 5E is indeed effective for mining association rules from very large databases. These experiments were performed in a stable environment involving a) the Java-based Aglets platform for mobile agents, b) the IBM synthetic data package to generate the necessary large databases, c) the Apriori algorithm for the mining process, and d) the SPDM method for program and data parallelizations. The test results however do not constitute a validation of the proposed 5E framework, but an indication that the research is in the right direction and more investigations would be worthwhile.

Keywords: five essentials, real-time data mining, mobile agents, SPDM, inter-object interaction, overlay elimination

1. Introduction

This paper presents the research findings of how the 5E (5 essentials) framework can yield high performance in real-time data mining over the Internet. The verification tests for the 5E were based mainly on the mining of association rules from very large databases. The central idea of the 5E is to combine the following five essential elements to work naturally together:

- a) Object-based parallelism: The goal is to realize the Internet power with object-based parallelism, which is the natural outcome when encapsulated objects are running in parallel [1].
- b) Inter-object interaction pattern: There are three basic patterns [2] for cooperating objects to interact, namely, *master and slaves*, *iterative process*, and

network processes. The interaction pattern for a solution must fit the nature of the problem, and a wrong choice would impede both the correctness and the performance of the final system.

- c) Programming model: There are three basic programming models, namely, *sequential*, *logically non-distributed* where program objects communicate only by shared memory (e.g. Linda's Tuple Space [3]), and *logically distributed* where objects interact by message passing. Choosing a wrong programming model for a solution would definitely lead to poor system performance as demonstrated in [4].
- d) Mobility: It applies to both program and data objects. It is vital in three aspects: i) for fault tolerance because objects can migrate to safer locations, ii) for load balancing because objects can migrate to less busy nodes, and iii) for distributed data mining when there are more data objects (databases) in different nodes than the data-mining program objects. The third aspect is actually a form of agenda parallelism [5].
- e) Hardware architecture: There are two basic types of architectures, namely, *physically distributed* where every node in the cluster has its own disjoint primary memory, and *physically non-distributed* where all nodes communicate by shared memory. Running *logically distributed software* on *physically non-distributed hardware* would result in poor performance because of the mismatch [4].

Many data mining systems are real-time phenomena because they have to meet the following two requirements:

- a) the mining results must be correct and reliable, and
- b) the results must be produced within a sufficiently small interval. For example, data mining of financial data to assist instant on-line investment decisions must be completed before the stringent deadlines. It means that the data mining process must possess the characteristics of a hard real-time system [6] because missing any deadline may lead to catastrophic consequences. Data mining is a knowledge discovery process in which the targeted knowledge may be represented in different forms, for example, association rules [7], and multimedia data [8]. We restricted our experiments and tests, which were performed on the Internet, to mining association rules

from large databases [9] because for this case stable tools and relevant credible experience are available in-house.

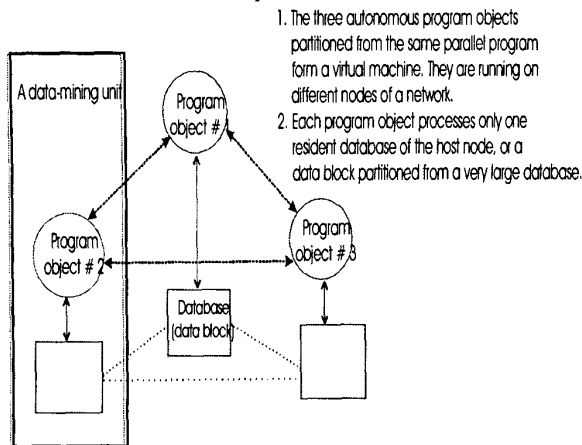


Figure 1. The passive form of object-based data mining

Object-based data mining can appear in two different forms: *passive* and *active*. In the passive mining form (Figure 1), a unit consists of a program object and a data object, which may be a partitioned data block of a large database or a resident database in the host node. If the components in the unit migrate, then it becomes the active

form of data mining (Figure 2). The migration of the data object usually applies to situations when the object is actually a data block partitioned from a very large database. The active form is especially suitable for successive mining of the resident databases on different hosts. When a mobile mining program object or “*miner*” has finished with the current host it will migrate to mine another node. A typical example of active data mining is to mine specific information from different Internet web sites with the help of a few intelligent mobile agents [10]. The active form needs strong object mobility support, which is an important open issue with no consensus solution and is being actively addressed by many researchers [10]. At present, there are some experimental platforms, which can support object mobility to different extents, and these platforms are generally known as *mobile agents systems* that include the relatively well-known examples such as Ara [11], Aglet [12], and Telescript [13]. Among them the IBM Aglets, which is aimed at supporting Java-based mobile agents on the Internet, is more stable because of its longer history and richer user experience. In fact, distributed programming based on mobile agents is new paradigm known as *remote programming* [14], which promises better performance for distributed systems by minimizing the overhead incurred by inter-object communications.

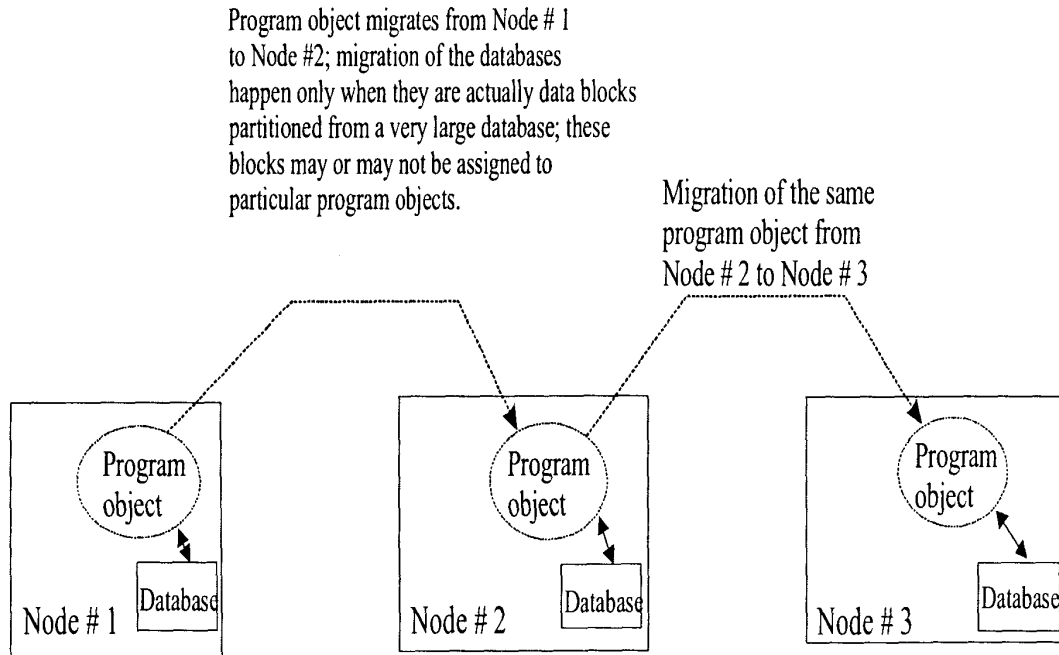


Figure 2. The active form of data mining

2. Motivation and Related Work

The motivation of this research is to find a way to support efficient real-time data mining of very large databases over the Internet. The methodology is to seamlessly combine the five essential elements to work naturally together: object-based parallelism, inter-object interaction pattern, programming model, object and data mobility, and hardware architecture. The three objectives to be achieved in the project are as follows:

- Finding the correct pattern for inter-object communications among the collaborating mobile objects in data mining.
- Identifying the parallelization method that effectively supports both the active and passive forms of object-based data mining.
- Identifying the strength and limitations of the 5E framework.

Despite the importance of parallel data mining, there is a lack of publications that directly address the issue. Recent and relevant publications include [15,16,17,18], which however concentrate mainly on theoretical discussions with little empirical proof. The literature for the five essential elements are however abundant because of their relative maturity. The lack of a comprehensive methodology to exploit the synergy by the five elements over the Internet for real-time data mining of very large databases has motivated the present work.

3. Details of Our Research

The 5E research is divided into five phases as follows:

- Identify the suitable model for inter-object communications in data mining.
- Identify a suitable parallelization method to support the two forms (active and passive) of object-based data mining.
- Find a stable tool to generate the very large databases necessary for the mining of association rules in the verification tests and experiments.
- Choose a stable distributed programming system to generate the logically-distributed object-based test programs.
- Choose a stable platform that can support object mobility over physically-distributed hardware.

Since the distributed computing paradigm is shifting from cluster based to Internet (web) based, we followed the trend by choosing an Internet/Java based distributed platform for the 5E tests and experiments.

In our previous work [2] three basic communication models for inter-object interactions were identified: *master and slaves*, *iterative stages*, and *process network*. We had also proved that a) each of these models is in fact natural

to a specific class of problems, and b) choosing the right inter-object communication model is the first imperative step in the formulation of correct parallel data-mining algorithms.

3.1 An Effective Parallelization Method

The objective of a parallelization method is to eliminate the I/O time needed for managing the overlay structure when a large database is too large to be completely encoded into the primary memory in a standalone system [19]. The elimination is achieved by partitioning a large database into smaller blocks so that each of them can be encoded completely in the primary memories of different machines. It is clear from this perspective that the parallelization process for distributed data mining is inherently data oriented and the SPMD (*single program multiple data* [20]) method (Figure 3) is a natural choice. In the SPMD approach, the data mining program object (A in Figure 3) is duplicated (A's), and the very large database (C) is partitioned into smaller blocks (C's) for distribution to different nodes for encoding into the local primary memories (B's). Matching the SPMD method with a correct inter-object interaction model is of paramount importance for yielding high performance in parallel data mining

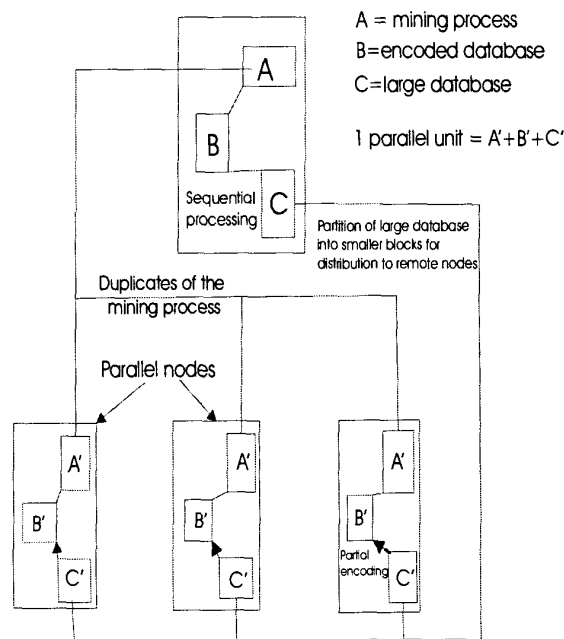


Figure 3. Parallelization by SPMD – database partitioned and distributed; but the “miner” (Apriori in this case) are duplicated and distributed

3.2 A Stable Environment for Verification

The environment for the tests and data collection consists of the following stable elements:

- Apriori parallelization: The stable Apriori algorithm [9] was parallelized for the verification tests involving mining association rules from large databases. In this case, the parallelization is achieved by duplicating the Apriori algorithm. The duplicates are then allocated to different nodes for parallel execution.
- Aglets platform: The Java-based IBM Aglets platform [12] was adopted for the parallel data mining tests over the Internet because it supports mobile agents with known stability.
- IBM synthetic data generation code [21]: The large data bases necessary for the mining experiments were generated automatically by this stable package.

In these experiments, a large database was partitioned into smaller blocks by the SPDM concept so that each data block could be allocated to an Internet node for independent mining by the Apriori duplicate. Since every participating node had an Apriori duplicate and an allocated partitioned data block, the number of cooperating nodes was equal to the number of partitioned data blocks. One of the participating nodes acted as the master and coordinated the slave nodes by barrier synchronization, which is natural for level-based mining strategies such as the Apriori. The Apriori duplicates were implemented as mobile agents (*aglets* in this case) to run on the Aglets platform. These mobile agents differ from *other units of parallelism* in the traditional distributed programming environments (e.g. PVM and p4). The traditional units of parallelism must be preinstalled in the predetermined nodes and locations before they can be invoked by various means that can incur high communication costs (e.g. RPC (remote procedure call), TCP/IP programming, or Java RMI (remote invocation)). The use of mobile agents, however, obviates the need of preinstallation and the result is that inter-object communications can be optimized. The behaviour by the aglets includes *creation*, *cloning*, *dispatching*, *retraction*, *deactivation*, *disposal*, and *message retransmission*.

3.3 Establishing the Reference for Comparison

In order to study and to empirically evaluate the performance of the 5E framework for data mining over the Internet, we needed to establish a reference so that comparisons can be made between this reference and the test results from the 5E verification exercises. The reference establishment task involved:

- Identifying three different decoding methods for the bit-encoding scheme [19].

- Incorporating these methods separately into original Apriori to create three different modified versions of the Apriori.
- Running these three different sequential versions and compare their results to identify the fastest version among them.
- Adopting the fastest version identified in step c) above as the basis for the SPDM parallelization and also as the reference for comparison with the 5E test data.

The test results from steps: c) and d) would be compared so that the efficacy of the 5E framework can be verified.

The modification of the sequential Apriori is achieved by embedding the chosen decoding mechanism, shown as *highlighted program statements* in Figure 4. The ultimate aim of encoding is to represent the whole database in the primary memory so that in the subsequent data mining operations no accesses to the original database would be necessary. In reality, it would be impossible to encode a very large database into the primary memory without an overlay structure. It means that occasional accesses are still necessary no matter how effective the encoding method is and how efficient the access method would be. Parallelization is a solution for the elimination of such overlay structures because a large database is partitioned

- $L_1 = [\text{frequent-1 itemsets}] \ \& \ \text{encode}(\text{whole_db});$
- For($k=2; L_{k-1} \neq \{\}; K++$) {
- $C_k = \text{apriori_gen}(L_{k-1});$
- For($m=0; m < \text{no_trans_in_encoded_db}; m++$) {
- $\text{decode_trans_m_for_level_k}(\text{encoded_db});$**
- $\text{count_itemsets_for_}C_k(\text{trans_m});$**
- $L_k = [\text{itemsets_for_}C_k \geq \text{Minsup}];$ }
- Find_association_rules_for_large_itemsets(L_k);

Figure 4. Modified sequential Apriori algorithm (modifications highlighted)

Bit power	2^0	2^1	2^2	2^3	
X(position) →	1	2	4	8	$V_t = \sum 2^X \downarrow$
Transaction ↓	CAT	DOG	BIRD	FISH	
1	0	1	1	1	14
2	0	0	1	0	4
3	1	0	1	0	5
...

Encoded values($V_t = \sum 2^X$)

Figure 5. Every item in a transaction is represented by its 2^X value (X is a bit's position)

into blocks that are small enough to be encoded completely in the receivers' primary memories for parallel accesses.

3.3.1 Encoding and Decoding. In the bit-encoding scheme all the items in a transaction are represented by their unique bit positions in a vector. That is, the whole database will be encoded into a two dimensional matrix illustrated in Figure 5. Every vector in the matrix is uniquely represented by its encoded value $V_i = \sum 2^X$, where X is the unique bit position of a specific item. For example, the 1st transaction in Figure 5 is represented by its encoded value of 14; that is, $2^1 + 2^2 + 2^3 = 14$.

For the above bit-encoding scheme, we have chosen three possible decoding schemes as follows:

- a) *Basic decoding:* For this method the following logic is executed repeatedly until i , which initialized to the encoded value V_i , is reduced 0:

```
if ( $i \geq 2^X$ ) then {
   $i = i - 2^X$ ;
   $X^{th\_item(position)\_in\_transaction} = TRUE$ ;
   $X = X - 1$ 
}
else  $X = X - 1$ ;
```

In this case the encoded value of 14 in Figure 5 would represent the "TRUE states" for items in the $2^{X=1}$, $2^{X=2}$, and $2^{X=3}$ positions of the first transaction. The encoded transaction would be scanned bit by bit starting from the most significant bit.

- b) *Binary decoding:* This method is based on the *basic decoding* approach above except that the first step is to find "the most significant TRUE position for X " by binary search. The aim is to slash the scanning time of the *basic decoding* approach by half.

- c) *Logarithmic decoding:* In this method, which saves decoding time by eliminating the entire data scanning process, the following logic is executed repeatedly until i (initialized to the encoded value V_i) is reduced to 0:

```
 $i = i - 2^{\lceil \log_2(i) \rceil}$ ;
 $\lceil \log_2(i) \rceil^{th\_item\_in\_transaction} = TRUE$ 
```

The $\lceil \log_2(i) \rceil$ operation converts $\log_2(i)$ into an integer by truncating whatever after the decimal point. For example, it yields the integer values of 3, 2, and 1 from the encoded value of $V_i = 14$ successively.

The main difference among these three methods is the time required for data scanning in the decoding process. The logarithmic decoding method requires no data scanning at all, and therefore should be the most efficient.

The database for the encoding/decoding experiments was generated by the IBM synthetic data generation package with the following specifications: a) *number of transactions* = 100,000 (100K), b) *average number of items per transaction* = 10, c) *total number of possible items* = 1000 (N1000), and d) *number of possible*

transaction patterns = 1000. In the test results illustrated in Figure 6, 1st marks the basic decoding method; 2nd marks the binary decoding method; 3rd marks the logarithmic method, and Apriori marks the original Apriori. By comparing the four curves, it is obvious that logarithmic decoding has yielded the best performance in

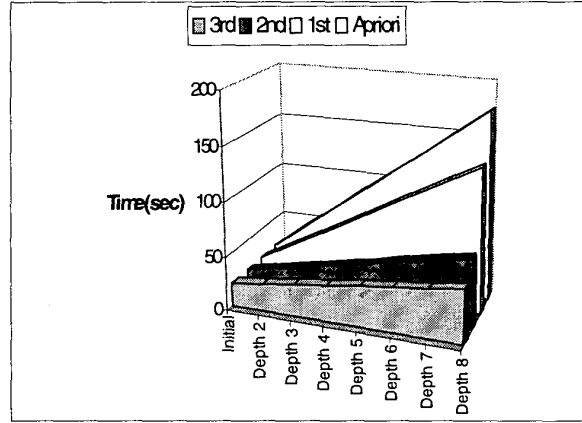


Figure 6. Performance difference among four decoding methods (D100K.N200)

mining the association rules from the same large database.

3.4 Parallelization with Logarithmic Decoding

The fastest logarithmic decoding method was adopted for the 5E verification tests, based on mining of association rules from very large databases. That is, every mobile agent is a copy of the modified Apriori shown in Figure 4, but with bit-encoding and logarithmic decoding. With the same database (D100K.N200) the test results

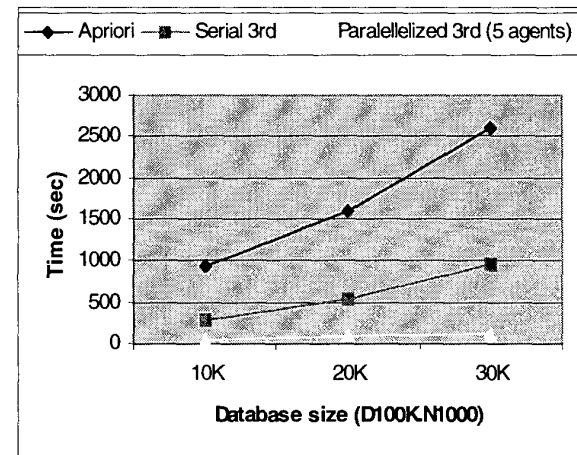


Figure 7. Performance difference between sequential and parallel data mining

(Figure 7) show that the parallelized Apriori (marked Parallelized 3rd on the graph) performed much better than the original Apriori (with byte encoding; marked Apriori) and the sequential version before parallelization (marked Serial 3rd).

5. Conclusion

In this paper, the 5E framework is proposed for the purpose of yielding high performance in real-time data mining over the Internet. The five essential elements in this framework are, namely, object-based parallelism, mobility, inter-object interaction pattern, programming model, and hardware architecture. The central idea is to combine the five essentials to work naturally together to achieve the required timeliness. The framework was tested over the Internet with stable tools and elements including the Aglets, the IBM synthetic data generation package, the Apriori algorithm (modified), and the SPDM parallelization method. The verification tests for 5E were based mainly on mining of the association rules from large databases. Although the preliminary test results have confirmed that the 5E framework can indeed support efficient data mining over the Internet, they do not constitute a validation of the framework. But, they do indicate that the proposed 5E approach is a correct direction for achieving high performance for real-time data mining over the Internet, and therefore further work along the same line would be meaningful. The immediate future work will include the following:

- a) Validate the 5E framework by more rigorous tests based on mining of association rules from very large databases over the Internet.
- b) Test the 5E framework with other types of data mining operation, for example, multimedia data so that the usefulness of the 5E approach can be generalized.

6. Acknowledgement

The research is supported by the both the GY843 and GV894 central funds from the Hong Kong Polytechnic University, Hong Kong (SAR), P.R. China.

7. References

- [1] J.A. Stankovic et al, **Strategic Directions in Real-Time and Embedded Systems**, *ACM Comp. Surveys*, 28(4) 1996, 751-763
- [2] D.S. Yeung and A.K.Y. Wong, **An Evaluation Identifying Features of Unified Distributed Programming**, *The International Journal of Computer Systems, Science & Engineering*, 13(5) 1998, 311-322
- [3] D. Gelernter and N. Carriero, **Coordination Languages and Their Significance**, *Communications of the ACM*, 35(2) 1992, 96-107
- [4] C.H. Cap and V. Strumpen, **THE PARFORM - A Performance Platform for Parallel Computing in a Distributed Workstation Environment**, Technical Report, Institute for Informatic, U. of Zurich, Switzerland, June 1992
- [5] N. Carriero and D. Gelernter, **How to Write Parallel Programs: A Guide to the Perplexed?** *ACM Computing Surveys*, 21(3) 1989, 323-357
- [6] A. Burns and A. Wellings, **Real-time Systems and Their Programming Languages**, *Addison Wesley*, 1990, 1-9
- [7] D. Tsur, J.D. Ullman, S. Abiteboul, C. Clifton, R. Motwani, S. Nestorov and A. Rosenthal, **Query Flocks: A Generalization of Association-Rule Mining**, *Proceedings of ACM SIGMOD Conference on Management of Data*, 1998, 343-354
- [8] O.R. Zaiane, J. Han, Z. Li, S. Chee and J. Chiang, **MultiMediaMiner: A System Prototype for MultiMedia Data Mining**, *Proceedings of ACM SIGMOD, International Conference on Management of Data*, 1998, 581-583
- [9] R. Agrawal, T. Imielinski, and A. Swami, **Mining Associations Between Sets of Items in Massive Databases**, *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1993, 207-216
- [10] R. Khosla, and T.S. Dillon, **Engineering Intelligent Hybrid Multi-Agent Systems**, *Kluwer Academic Publishers*, Massachusetts, USA, 1997
- [11] Ara, www.uni-kl.de/AG-Nehmer/Ara
- [12] O. Mitsuru and K. Guenter, **Aglets Specification**, <http://aglets.tri.ibm.co.jp/spec10.html>
- [13] D.B. Lange, **Mobile Objects and Mobile Agents: The Future of Distributed Computing?** *Prentice Hall*, 1998
- [14] W.T. Cockayne and M. Zyda, **Mobile Agents**, *Manning Publication*, 1997
- [15] R.A. Agrawal and J.C. Shafer, **Parallel Mining of Association Rules**, *IEEE Transactions on Knowledge and Data Engineering*, 8(6) 1996
- [16] R. Afrawal and J.C. Shafer, **Parallel Mining of Association Rules: Design, Implementation and Experience**, *IBM Research Report RJ 10004*, 1996
- [17] E.H. Han, G. Karypis, and V. Kumar, **Scalable Parallel Data Mining for Association Rules**, *Proc. of ACM SIGMOD International Conf.4 on Management of Data*, 1997, 277-288
- [18] D.W. Cheung, J. Han, V.T. Ng, A.W. Fu and Y. Fu, **A Fast Distributed Algorithm for Mining Association Rules**, *Proceedings of the International Conference on Parallel and Distributed Information Systems*, 1996, 31-34
- [19] Allan K.Y. Wong, S.L. Wu and L. Feng, **An Efficient Algorithm for Mining Association Rules for Large Itemsets in Large Centralized Databases**, *Proceedings of the IEEE SMC'99*, Tokyo, Japan, 905-910
- [20] T. Lewis and H. El-Rewini, **Introduction to Parallel Computing**, *Prentice Hall*, 1992, 46-51
- [21] IBM Almaden Research Center, **Synthetic Data Generation Code for Association and Sequential Patterns**, <http://www.almaden.ibm.com/almaden/projects.html>, 1998