THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

A Framework for Scalable (Mobile Agent Based) Distributed Mining of Association Rules over the Internet

By Sui-Lun Wu

A Thesis Submitted in Partial Fulfillment

of the Requirement for

the Degree of Master of Philosophy,

Department of Computing

Hong Kong Polytechnic University

9 March 2002

Abstract

The aim of this thesis is to propose a framework for scalable (Mobile Agent Based) distributed mining of association rules. The scalability is the means to maintain an optimal computation-to-communication (CTC) ratio for high mining performance. The objectives, which have been achieved successfully, include the following:

a) To evaluate the impact of asynchronous agent collaboration (through asynchronous message passing) on mining performance in terms of turnaround time.
b) To study the relationship between asynchronism and the CTC ratio.
c) To propose a *Scalable Split & Aggregate Framework* (S^2AF) to maintain an optimal CTC ratio.
d) To propose at least two algorithms to study the feasibility of the proposed S^2AF under different conditions.
e) To choose a stable platform for testing S^2AF so that credible test results can be collected for analysis.

Mobile agents are software entities or program objects that work independently to discharge their duties similar in fashion to human agents. Meanwhile these entities can move from node to node for various reasons. Since agents can replicate or terminate themselves, by nature a mobile agent program is scalable.

This research covers different relevant areas of investigations, which require both

backtracking and cross-referencing operations. To make the work more effective, a methodology to "*investigate & experiment & proceed with possible backtracking, cross referencing and looping* (IEP)" is devised and adopted. It is implemented as a research roadmap.

All the experiments were carried out over the chosen stable Java-based Aglets mobile agent platform running over part of the PolyU Intranet in our laboratory. Most of the tests in the early investigations were based on the data generated by the IBM data generated package, which is widely used by other researchers in the area of data mining of association rules. In addition, tests were carried out with real multimedia data (mainly image) in later investigations the aim being to demonstrate that the proposed S^2AF is indeed applicable to real-life problems. The analysis of these test results led to the proposal of another efficient algorithm, namely, the *OWLBA (Optimized Weighted Load Balancing Algorithm*). The PolyU Intranet is intentionally used to gain insight into scaling the S^2AF for real-life Internet based applications in the area of scalable distributed data mining, especially when mobile agents are involved.

Acknowledgement

I sincerely thank my supervisor Dr. Allan Wong and co-supervisor Professor Tharam Dillon for their support and guidance throughout the thesis. I would also like to thank those who helped me do better research in the experimental investigations, including Dr. Ling Feng (information on how to alter the IBM data generation package for better test data); Dr. Vincent Cao (valuable criticism on the limitation of the IBM package); and Mr. Sam Ho (his experience on mining "*longitudinally partitioned*" data blocks for association rules). Last but not least, I thank Dr. W.N. Leung and Dr. K.W. Hung for their positive criticism on some of the intermediary outcomes.

List of Figures

List of Tables

Table of Contents

Chapter 1

Introduction

1.1 Knowledge Discovery in Databases (KDD) and Data Mining

Organizations have realized how important relevant past experience is in making sound business decisions and planning. Finding useful experience in a timely manner from a huge amount of data efficiently and effectively is a data mining process. For example, a financial analyst makes forecasts from reports and records made public by different companies. The fact is that these companies now produce so much data daily that it may require a lifetime for an analyst to read it all under normal conditions. This phenomenon is particularly true with data available on the World Wide Web (WWW). In order to make use of the WWW information, there is a need to develop efficacious methods, namely, data mining algorithms, for filtering, selecting, and interpreting data. In this light, there is a rise of interest in the new field of 'data mining' or KDD (knowledge discovery in databases) [11],[24],[38],[44],[57],[61]. In a general sense, mining is a process to remove useless "debris" to find the treasure.

The terms of 'data mining' and 'KDD' were defined in the first international KDD conference in Montreal, Canada in 1995. It was proposed that the term KDD be used to describe the whole process of extraction of knowledge from data. In this context, knowledge means relationship and/or patterns among data items. It was also proposed that the term 'data mining' should be used exclusively for the discovery

stage in the KDD process. In principle, the knowledge discovery process consists of six possible phases/stages as shown in Figure 1, namely:

a. Selection (Select the data that will be useful for data mining.)
b. Cleaning (Pre-processing)
c. Enrichment (Pre-processing)
d. Transformation (Encoding – encode the data into the format that can be used for data mining.)
e. Data mining (Mine the useful information from the data selected.)
f. Evaluation (Evaluate the useful information from the mining process and generate the report.)

Among these stages data mining is the key element of KDD.

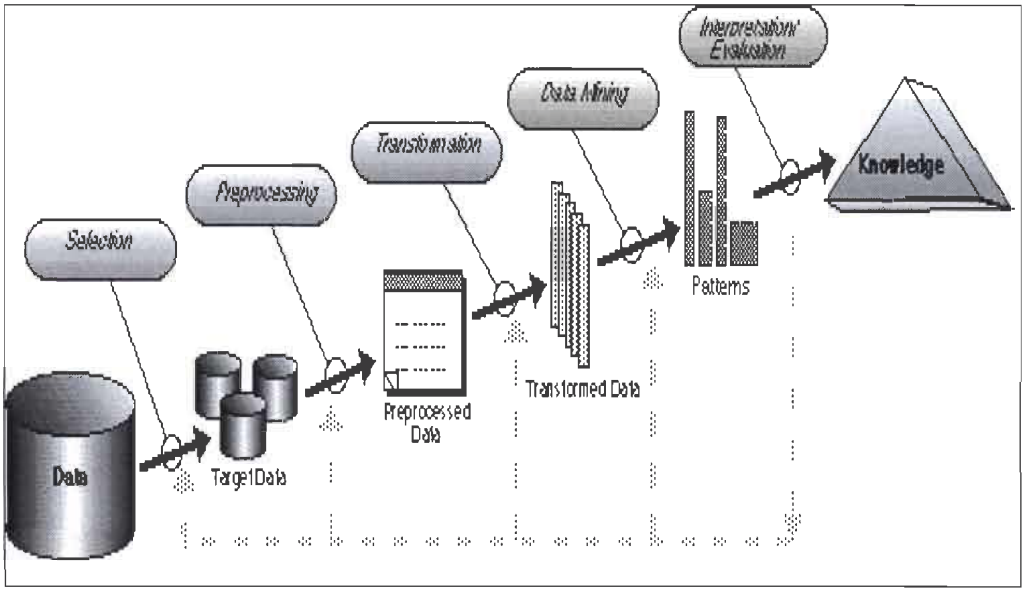

FIGURE 1. AN OVERVIEW OF THE STEPS COMPRISING THE KDD

One may wonder what is the main difference between data mining and the traditional query tools. Firstly, data mining does not replace the query tools, but gives us another way to retrieve information, which would appear differently than

that gathered by traditional query applications. Query tools, such as the SQL, retrieve straightforward answers; for example, 'which client has responded to the credit card advertisement letter?' or 'what is the average sales rate for this month?' It is however difficult to use a traditional query tool to find out 'what is the optimized segmentation of our clients?' or 'is there any relationship between a particular product and its clients?' One may use the trial-and-error approach to find the answers, but that could take a long time. The alternative is to shorten the time to find the knowledge through machine learning or association rules.

Data Mining is actually a multi-disciplinary field comprising different techniques, including machine learning [10],[43],[48], pattern recognition, statistics, databases, and visualization. The motivation is to discover hidden knowledge, unexpected patterns and new rules from large databases. Data mining is a young but important area of research because of its usefulness in determining future action based on historical data. Its usefulness is natural to stock trading because investors could predict the trend from recent local/international market movements. Another example is that the manager of a department store could make use of past transactions to decide how commodities should be put alongside each other to increase sales.

In fact, as mentioned before, data mining is not a single technique, but rather an '*anything goes*' affair. That is, any technique that helps extract useful data can be included in the KDD model. In a general sense, data mining can be applied in different areas such as marketing, medicine, stock price predictions...etc. In reality, data mining is already utilized in organizations such as American Express and

AT&T because KDD has become a means of analyzing client files [2]. The foci of different research projects in the light of KDD can be divided loosely into the following categories:

- *Query tools*
- *Association rules [4],[62]*
- *Genetic algorithms [1],[8],[53]*
- *Online analytical processing (OLAP) [36],[37]*
- *Decision trees [19],[54],[58]*
- *Neural networks [13],[34],[35]*
- *Statistical techniques*
- *Visualization*
- *Case-based learning (k-nearest neighbor)*

The width of the domain of data mining is well manifested by the availability of different techniques in different problem domains [2],[6],[7],[9],[22],[23],[25],[26],[27],[52],[65],[71] that include the mining of multimedia data [30],[39],[70],[72],[71],[73],[77]. Although data mining is synonymous with Knowledge Discovery in Databases (KDD), *this research, however, would focus on mining of association rules, in a scalable manner, with mobile agents [20],[55],[60],[75] over sizeable networks exemplified by the Internet.*

Data mining over a sizeable network such as the Internet is usually based on either one of the following two models [64], namely, *passive* (Figure 2) and *active* (Figure 3). In the passive mining form an operation unit consists of a static set of

components, namely, "*a static program object or agent + a data object*". The data object may be data block partitioned/split from a large database or an original resident database in the host node of the agent. If any of the components in the unit migrate, this is the active form of data mining. The active form is especially suitable for successive mining of the resident databases on different hosts. When a mobile agent or "*miner*" has finished with the current host it will migrate to mine another node. A typical example of active data mining is to search/discover/mine specific target information from different Web sites, with agents that possess intelligence [20],[42],[60]. This project deals with the active form of data mining over the Internet because it involves migration of components in the operation unit.

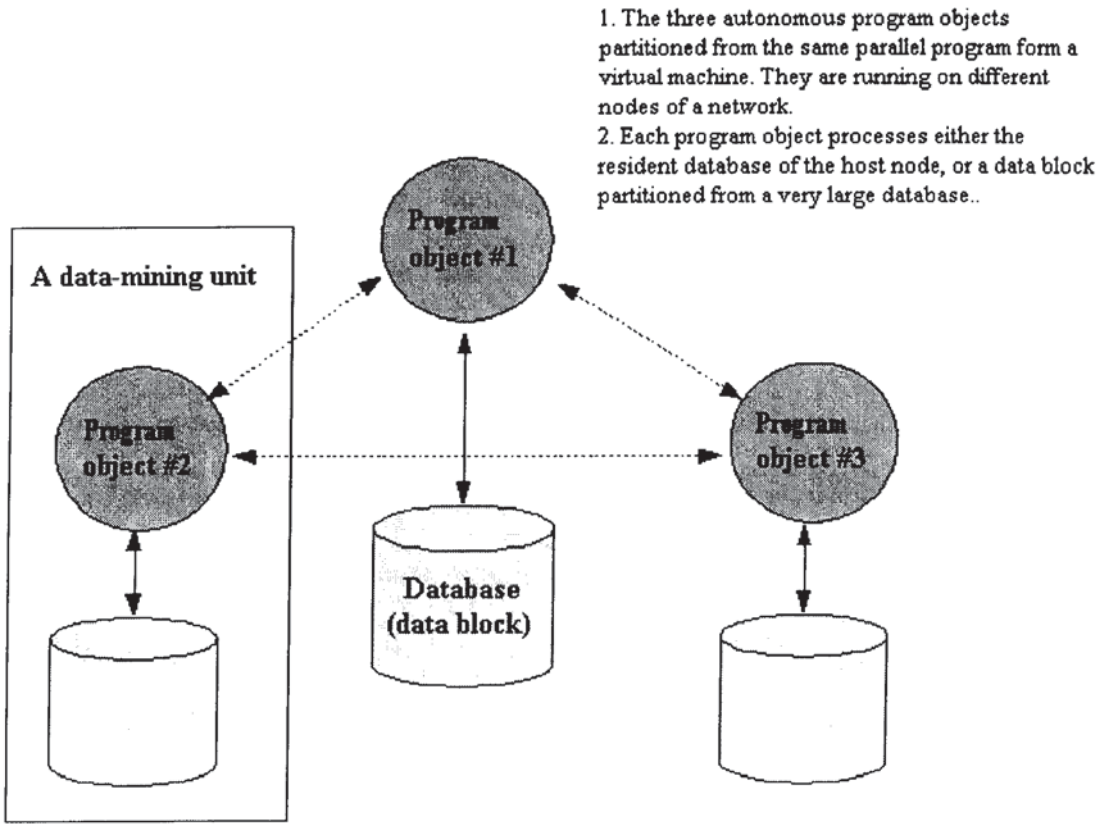

FIGURE 2. THE PASSIVE DATA MINING MODEL

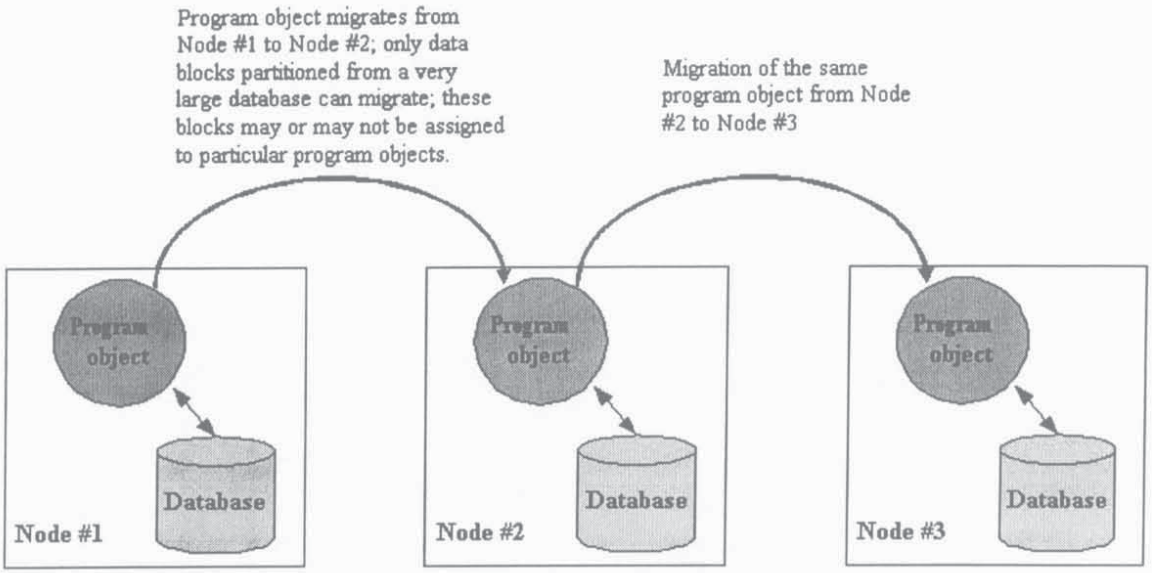

FIGURE 3. THE ACTIVE FORM OF DATA MINING

To test the proposed framework, mobile agent platforms were studied so that a stable platform could be chosen to support the S^2AF (Scalable Split & Aggregate Framework) experiments and tests. There are many experimental mobile agent platforms available in the field but some of them may disappear overnight. The more frequently seen examples from a literature search include Concordia, Ara and Aglets. After careful investigation, the IBM Aglets [75] was chosen because of its stability and rich user experience [60]. Our experience so far with the Aglets [64],[65],[66] is positive. In order to generate trustworthy data for tests in investigation in the early stages of the project, the IBM data generation package was chosen for simulation experiments. The data generated by this package is relational data and the user can choose the data size, the number of data items and the average length of the transactions in the database to be generated automatically [76]. Some of our publications (e.g. [3],[5]) are based on the results from such simulation experiments.

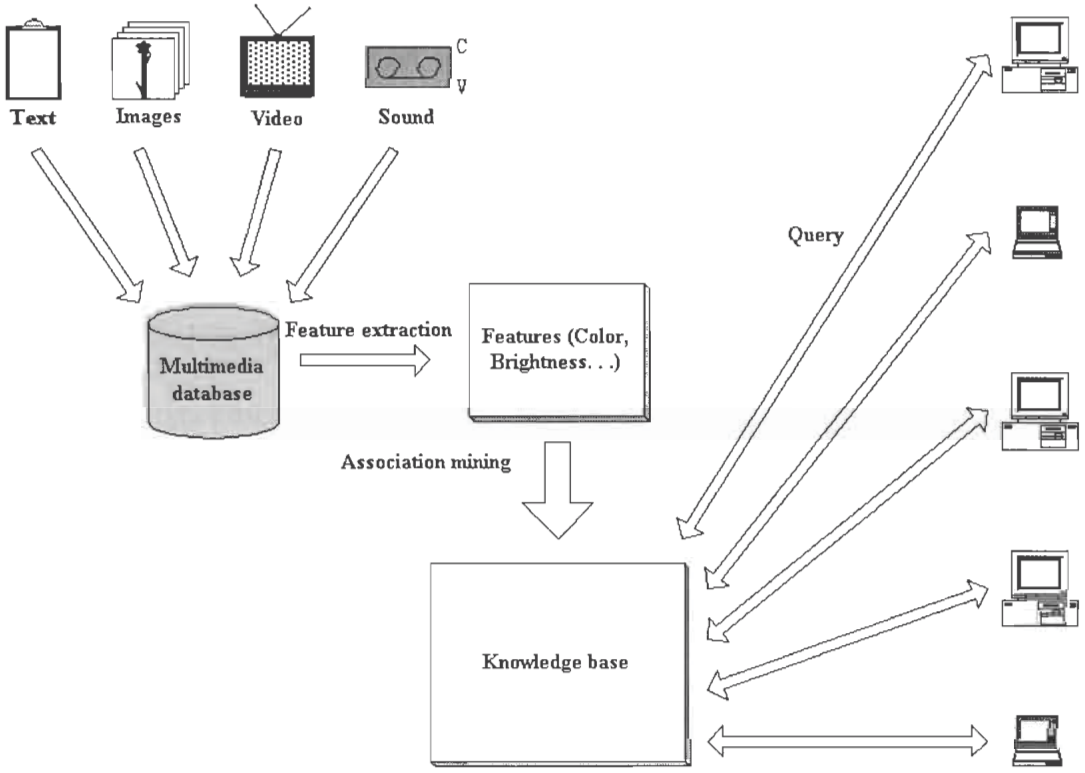

FIGURE 4. THE TEST ENVIRONMENT FOR MINING MULTIMEDIA DATA WITH THE PROTOTYPE

In order to demonstrate that the proposed framework can indeed support scalable distributed data mining of association rules in general, multimedia data will be employed to demonstrate this possibility [77]. Figure 4 provides the basic idea of how the proposed S^2AF framework can be applied to the multimedia data mining over the Internet.

1.2 Problem and Motivation

The aim of the thesis is to propose a framework *for scalable (Mobile Agent Based) distributed mining of association rules* over the Internet. Scalability is defined as the ability of a framework to operate with similar efficiency over both small networks (e.g. Local Area Network (LAN) or a controlled environment such as an Intranet) or large ones such as the Internet. This framework should enable the constituent agents

of a mobile-agent based mining program to collaborate effectively to maintain high mining performance. Performance here is expressed in terms of shorter mining time to obtain the target information or mining speedup.

In general, scalability can be achieved through many means that include:

a) Agent/object mobility that usually involves object migration,
b) Partitioning or splitting of a large data block into smaller modules so that additional agents created by replication or cloning can handle these modules in a parallel fashion, and
c) Minimizing communication overhead in the inter-object collaboration process.

An agent [28],[29],[45] is a program that assists people and has different responses according to the specific environment. It is autonomous and controls its own actions adaptively. If an agent is capable of mobility, then it is known as a mobile agent [12],[33],[41],[46],[47]. Mobility allows the agent to travel from one node to another and makes use of the local resources. That is, when an agent arrives at a machine, it runs on it and accesses the local resources. After finishing with the present machine and returning the result, it moves to another to continue the mining work according to the pre-installed schedule (itinerary). It is common that those machines that can support mobile agents do not require the pre-installation of any agent program.

The mobile agent approach provides many advantages such as follows:

1. It reduces the network traffic because it moves to where the data is and returns the result only after it has finished. In this process, the need to transfer large volumes of data is obviated.

2. It provides more fault tolerance because the agent can handle critical real-time jobs even if the network connection is temporarily not available between the client and the server.
3. The mobile agent can react autonomously to the changes of the environment.
4. It provides safety because an agent can be dispatched to another node for preservation before the current host shuts down.
5. No pre-installation is needed.
6. It provides scalability because an agent can be replicated for more parallelism.

The property of scalability is important for the algorithms proposed in this research for successful distributed mining of association rules over the Internet. Scalability in the context of this research is that the workability and performance of the proposed algorithms would not depend on the size of the underlying network (i.e. the number of nodes). Rather, the number of agents can be increased by replication or decreased by aggregation.

Object (or agent) migration can provide the necessary load balancing to gain speedup, and object cloning can also provide more intrinsic parallelism for higher speedup. There are many ways that one can minimize communication overhead but the focus here is to reduce it by maintaining an optimal *computation-to-communication* (CTC) ratio. The rationale is that with an optimal CTC ratio the system would spend more time in useful computation (for mining) rather than as overhead for handling inter-object communication [21].

1.3 Objectives

At this point, it is worthwhile to point out that the impact of the CTC ratio on distributed data mining is not well understood. Therefore, it is an important issue to resolve before the proposed S^2AF can work efficiently and effectively. Naturally the objectives include the following:

a) To evaluate the impact of asynchronous agent collaboration (through asynchronous message passing) on mining performance in terms of turnaround time.

b) To study the relationship between asynchronism and the CTC ratio.

c) To propose a *scalable split & aggregate framework* (S^2AF) to maintain an optimal CTC.

d) To propose at least two algorithms to study the feasibility of the proposed S^2AF under different conditions.

e) To choose a stable platform for testing S^2AF so that credible test data can be obtained.

Asynchronous mobile agents collaborate in a client/server relationship. The client could ask the server for a service anytime and then proceed with other tasks before coming back to collect the service result. Whether the client/server relationship is blocking or non-blocking depends on the nature of the task. In the synchronous mode, when a client requests service from several servers simultaneously, the service results may not come back at the same time. This aspect of asynchromism is reflected in the β factor of the formula (2.2.1), namely, $TimeDelay = ServiceRTT * N^{e^{-\beta}}$. In the S^2AF, split means partitioning a database or a large data block into smaller modules so that these modules can be handled in

parallel by additional agents that are created by replication (cloning). And aggregate means merging smaller data modules into a large one to be handled by a single agent and those excess agents would be purged. A stable platform would enable us to create different controlled network environments for different testing purposes. Credible experience and results would provide insight into how the S^2AF can be generalized for different Internet applications, as well as shed light on what direction should be adopted in the future S^2AF enhancement.

The acceptance criteria for the proposed framework include:

1st: The framework should be scalable so that an optimal CTC ratio can be maintained.

2nd: The framework should be applicable for Internet based distributed data mining.

3rd: The framework should support real-life applications.

The contribution of the proposed framework, namely, S^2AF, is that it can maintain an optimal CTC ratio through scalability attained by the *split & aggregate* (S&A) strategy. The CTC ratio is usually lowered by the sequential property in interleaved asynchronism as indicated by equation (2.2.1), namely, $TimeDelay = ServiceRTT * N^{e^{-\beta}}$. To rectify this situation, measures must be formulated so that an optimal CTC ratio can be adaptively maintained or even enhanced. In fact, the preliminary investigation in this direction led to several publications, and the overall conclusion is that the S&A strategy approach is indeed an effective solution for the stated purpose of optimal CTC ratio maintenance.

Experiments with the S^2AF prototype would be carried out in a controlled Internet

environment over a stable Java-based *mobile agent platform* (MAP). The main reasons to choose a stable MAP are:

a) A properly selected Java-based MAP would enhance the continuity of the project because its growth with the Internet means continuous support by the vendor(s).

b) The MAP stability would enhance the credibility of the test data.

c) The mobility needed to support scalability is inherent and this helps generalize the S^2AF for Internet applications.

1.4 Outline of the Thesis

In this thesis, an improved Apriori algorithm and four load balancing algorithms of the S^2AF (Scalable Split & Aggregate Framework) are proposed. The thesis is organized into seven chapters. Chapter 2 introduces different kinds of sequential and distributed association mining algorithms. And it also introduces the impact of the CTC ration on the performance of parallelization. The improved Apriori algorithm by encoding and decoding is described in chapter 3. In chapter 4, the theoretical foundation of the proposed framework is described. The S^2AF is described in chapter 5. The simulation results and evaluations are reported in chapter 6. The final chapter concludes the thesis.

Chapter 2

Sequential and Distributed Association Rules Mining

Data mining of association rules has already gone through several stages as shown in Figure 5, namely, 1st stage - sequential approach (SA), 2nd stage - parallel approach (PA), and 3rd stage – distributed approach (DA). This evolution pairs up with that of the hardware speed in MIPS (million instructions per second) and network capacity in MegaKIPs (million of bits per second over 1 kilometer) [49]. The three stages evolve progressively; that is, techniques in the previous stage usually form the basis for the next. The main drive is to derive from parallelism the necessary speedup, without which data mining would be unsuccessful due to the large volumes of data involved.

To summarize, data mining of association rules has covered different areas of continuous work by different researchers. Some of these areas precede others; for example, sequential data mining of association rules precedes the parallel approaches. These areas can be identified as follows:

a) *Sequential mining*: The work concentrates on how to reduce the data size as the mining process is progressing. It includes the important issue of representation of the data to be mined in the main memory so that I/O overhead can be economized. The typical example is the Apriori and the HybridApriori algorithms [3],[4],[6],[25],[27], [30],[40],[56],[65],[72].

b) *Parallel mining*: The drive is to speed up the mining process by parallelism,

either in a multiprocessor environment or a distributed one [5],[14],[16],[17],[59],[64],[66].

c) *Intermediary algorithm improvement*: The aim is to make the mining algorithms more efficient. For example, at the early stage of the thesis work, it was found that the logarithmic decoding algorithm together with binary data representation would perform better than the traditional sequential Apriori [32],[40],[50],[51],[56],[62],[65],[74].

d) *Scalability*: The aim is to look at how mining of association rules can be achieved in a scalable manner over a network platform. The intention of scalability is to reduce mining overhead in an adaptive manner. Among the first to investigate this issue is Han et al [31] and later, others including [15], [67], tried to resolve the scalability issue in a distributed environment by aggregation.

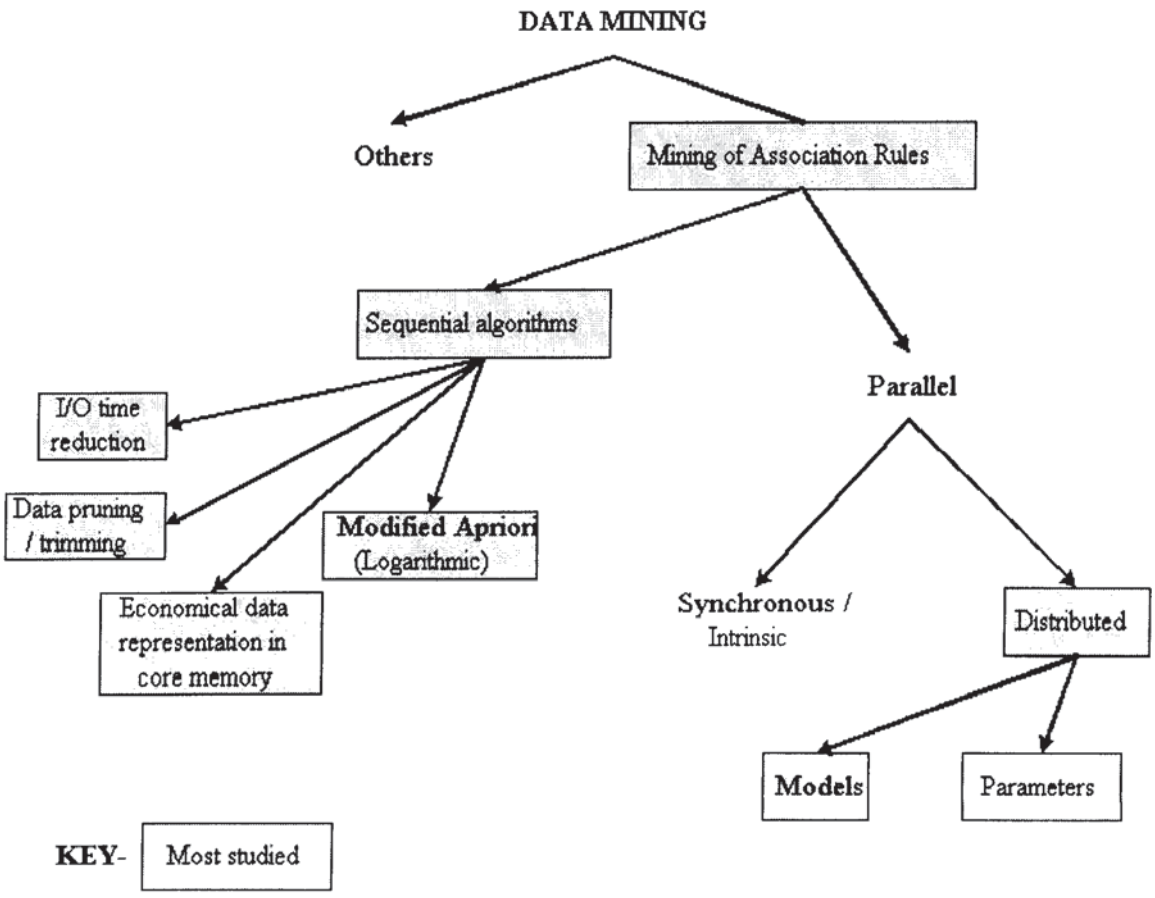

FIGURE 5. THE ROADMAP FOR REVIEWING RELATED WORK

2.1 Sequential Mining of Association Rules

In a glimpse, SA is represented by the sequential Apriori work in [6]. Agrawal et al [3] first introduced the concept of association rules in 1993 and proposed the sequential Apriori algorithm [6] for mining them. An association rule describes the relationship between the items in the database. If one assumes the following: I is a set of items and database D is a collection of transactions, where each transaction T is a set of items such that $T \subseteq I$. Then, one says that the association rule $X \Rightarrow Y$ holds when the following two conditions are true: a) X and Y are both large itemsets because they have at least s% support (count) in D, and b) c% of the transactions in D that contain X also contain Y. Given a set of transactions D, the problem of mining association rules is to generate all association rules to relate large itemsets that have minimum confidence support. For example, a rule: *"80% of customers who purchase a pencil box also purchase a school bag"* is an association rule. The Apriori approach to find association rules consists of the following steps:

a) Generate all combinations of items from the database and then count the transactions to find out whether large itemsets do exist (≥ support count, s%).
b) Generate the association rules for large itemsets, which should also have enough confidence percentage (c%).

Mining for association rules from large database is usually time consuming and therefore sequential approaches such as the Apriori would take a long time and become impractical for solving real-life problems. It is generally understood that most of the mining time is consumed by the I/O operations and data swapping because of the following two issues:

a) For every deeper level of mining, the database has to be reread into the main

memory.

b) The limited capacity of the main memory means that frequent I/O operations are needed to bring data yet to be mined into the memory.

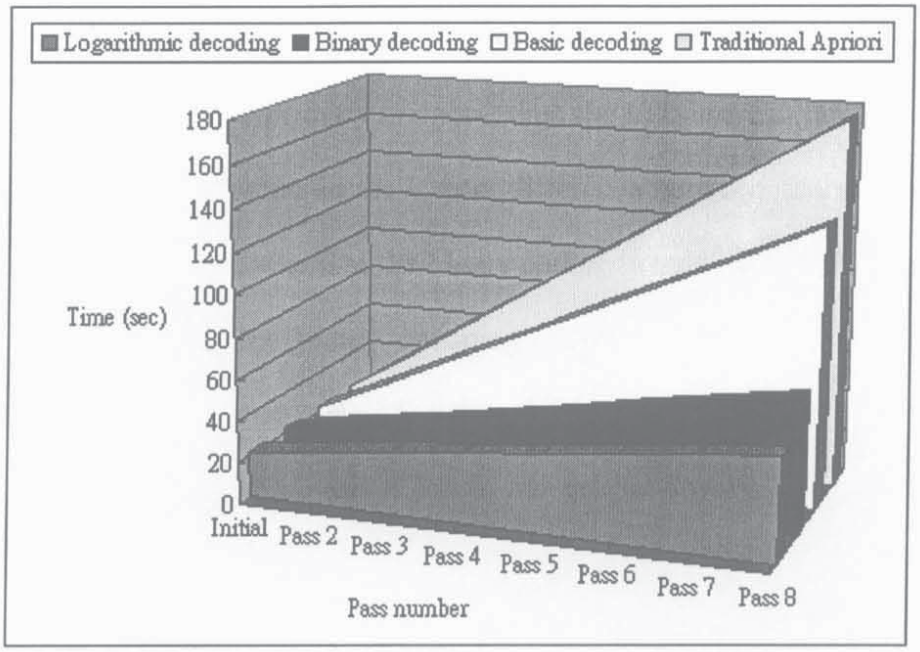

FIGURE 6. COMPARISON OF FOUR ALGORITHMS (T10.I4.D100K.N200)

The mining time for a sequential mining methodology, however, can be reduced by applying the following three approaches or a combination of them:

a) Optimization of the database representation in the primary memory to lessen the I/O time in the iterative mining processes. The argument is that less I/O would mean faster mining because the CPU capacity would be utilized more for useful mining work. For example, the work on AprioriHybrid is trying to achieve such an optimization goal [6] and the others [65].

b) Prune the database of those transactions not required for mining the deeper levels [6],[40].

c) Organize the data to reduce the number of search cycles (e.g. [56]).

In fact, a data coding (data representation) format/method can have serious impact on mining performance, as has been found in the first phase of this research [65]. First, the data format will affect the I/O time in every round of data search; that is, more data encoded into the memory would decrease the requirement of such I/O time. Ideally, if the whole database at any stage in the mining process can be encoded/represented in the primary memory, then the I/O time can be eliminated. In our previous work, we compared the impact difference between character encoding and binary encoding and found that the binary coding is much more efficient [65]. In this case, not only can the binary encoding reduce I/O time, but also efficient methods devised for such encoding can further reduce the mining time synergistically. To demonstrate this point, in our previous work, the three decoding methods proposed by us were tried and compared with the traditional Apriori algorithm. The findings indicate that the decoding method can indeed affect the system performance significantly.

The three decoding methods, which were proposed in the first phase for some of the project's intermediary experiments are as follows: basic decoding, binary decoding and logarithmic decoding. When one of these methods is included into the traditional Apriori algorithm [3], a *modified Apriori algorithm (MAA)* is formed. The test results show that MAA could improve the mining performance of the traditional Apriori by up to 500%, provided that the logarithmic algorithm works together with binary encoding. The performance difference is shown in Figure 6.

Actually the originators had also proposed an approach to improve the speed of the traditional Apriori, and their proposed solution is the AprioriHybrid approach [6]

with the AprioriTid as the intermediate step. The mining process should start with the traditional Apriori and then the mining should switch to the AprioriTid approach at some stage when the candidate set is small enough to fit into the main memory. The ApriroiTid stores the candidate itemsets of each uniquely identified transaction in a table and this makes the counting process much faster. AprioriTid does not use the database at all for finding the support count for the candidate itemsets after the first pass. The candidate itemsets in the previous pass are encoded and stored into the memory to save the reading effort. There are also other methods proposed by different projects to solve the above two issues with varying degrees of success (e.g.[32],[50],[51]). The weakness of the AprioriTid is that when the number of passes/levels in the mining process is small the table for the candidate sets for the identified transactions can be larger than the database itself. This defies the original principle of maximizing the usage of the main memory and possibly leads to memory overflow as pointed out in [6]. The major weakness of the AprioriHybrid approach is that one cannot decide when this approach should be invoked. For example, if it is invoked at the last pass/level of mining, then the approach becomes meaningless. Meanwhile the costs incurred in the switch from one approach to another are wasted.

If the database is extremely large, then measures used in sequential mining to optimize memory usage and I/O operations would not be enough to yield the speedup necessary for the expected mining performance. The alternative is parallel data mining.

2.2 Distributed Mining of Association Rules

Distributed mining of association rules [5],[56],[59],[74], is a form of parallel data mining, over a sizeable network. The parallel approach or PA in data mining work is mainly concerned with making use of the *traditional high performance computing* (HPC) approach, in which multiprocessor-based parallelism and the shared-memory approach such as the SIMD architecture [68] were the norm. Certainly, the norm has changed because contemporary HPC also includes distributed architectures; for example, the MPP is a vendor-supported distributed architecture marketed as a HPC system [21]. A typical example of previous PA work is described in the IBM Research Report on the Parallel Mining of Association Rules [5], which identifies the three basic approaches for parallel computing, namely, *Count Distribution*, *Candidate Distribution* and *Data Distribution*.

The principle of the Count Distribution is to parallelize the mining process by partitioning the database into smaller blocks for different processors to mine the large itemsets. In each mining pass, the processors exchange their counts for the blocks for which they are responsible. The problem with this approach is synchronization because each processor should wait for all the processors to finish before going to the next mining pass. Furthermore, when the size of the candidate itemsets in the hash tree is too large, there is a danger of memory overflow.

For solving the memory overflow problem in Count Distribution, the Data Distribution approach is proposed. The idea is to parallel the mining process by distributing the candidate itemsets to different processors and each processor mines the large itemsets from the database. Yet, this would give rise to the problem of large

data transfer rates. Not only does each processor need a data block of roughly the same size, each processor also needs to transfer all its local data to every other processor. This would consume the data communication bandwidth and lead to poor performance. The Candidate Distribution approach tries to resolve the problem of synchronization at each pass of the mining process. The idea is to remove the dependence between the processors so that they can process independently. For example in the pass *l*, the algorithm divides the large itemsets among the collaborating processors, but each processor mines independently and prunes the candidate itemsets when information is available from other processors.

There is a considerable amount of findings in the area of parallel data mining. For example, the Adaptive Algorithm (Count Distribution) developed by David W. Cheung [15] makes use of the advantage provided by shared-memory multiprocessor (SMP). The goal is to build an adaptive asynchronous parallel algorithm for solving the problem of synchronization in each pass in mining association rules. There are also other researchers who are trying to improve the Count Distribution algorithm [14],[16],[17],[59]. Since controlling the data transfer rate in parallel data mining is an important key to better performance, some research projects concentrate on how this can be achieved effectively. A comprehensive example in this aspect is the Scalable Parallel Data Mining Algorithm proposed by Han et al [31]. The idea is to use the Torus topology for building an intelligent data distribution algorithm to optimize data transfer rates in the Data Distribution approach.

This work, however, is of very preliminary nature because the concept includes only

the topological issues such as how the database should be partitioned and how the mining processes should collaborate, but without looking at what kind of delays or problems could make the collaboration clumsy or even impossible.

Recently the issue of data mining from distributed databases, or simply DA, has attracted more international attention simply because of the need to utilize the Internet for some E-commerce applications. Data mining over a sizeable network is not easy because of asynchronous communications inherent in the network for inter-object collaboration. As discovered in some experiments in the first phase of this research [64], asynchronism causes a time delay that can be characterized by the following equation:

$$TimeDelay = ServiceRTT * N^{e^{-\beta}} \quad \text{.....(2.2.1)}$$

The parameter β is known as the *degree of overlapped parallelism*; a higher value means the N distributed process (in our case distributed mobile agents) are operating in a higher degree of parallelism and less time delay. For example, 100% parallelism would lead to $TimeDelay = MiningTime * N^0$, or all the N processes would finish together within *MiningTime*. That is, the parameter β is sensitive to the time differentials among all the collaborating objects, as illustrated in Figure 6. If the data exchanges between two collaborating distributed processes are interleaved as illustrated in Figure 7, then the overall time delay due to data exchange depends on β, which can be calibrated for a system [63],[64].

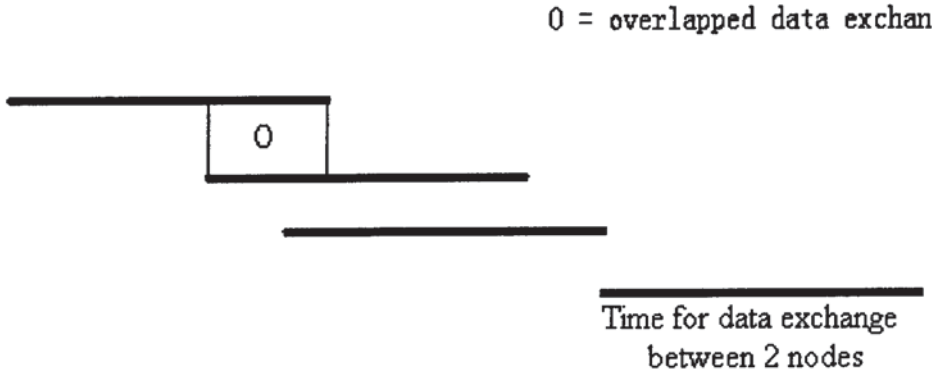

FIGURE 7. POSSIBLE OVERLAPS (4 PROCESSES) IN EVERY MINING CYCLE

At present there is little activity to investigate and improve β and the early work in this project has contributed some useful ideas to equation (2.2.1), in light of web-based applications [11],[64]. Meanwhile, in the second phase of this research, an aggregation approach was proposed for maintaining a high computation-to-communication (CTC) ratio [66] for improving β. This aggregation approach, which has paved the way for the final S&A approach that forms the basis of the S^2AF, is one of the two aggregation approaches that can be identified from literature so far; the other one was described by Han et al [31]. The term S&A is a generic description of the ability to split and aggregate, but S^2AF is more specific in the sense that the S&A actions are independent of the number of agents or the size of the network.

2.3 CTC Ratio

In the early investigation, the impact of the CTC ratio on performance is investigated. In fact, a detailed study of the relationship between CTC and β is skipped in this thesis because it is the scope of another ongoing project [18], in which the preliminary results concluded that the relationship for a particular setup

can indeed be calibrated.

The study of the CTC impact in the intermediary work was carried out mainly by the empirical approach and the programming paradigm is the master-slaves over the stable Aglets mobile agent platform in our laboratory. A set of experiments was performed and in these experiments every slave implements the MAA (Modified Apriori Algorithm – Figure 12). The aim is to find out at what point that additional processors would not yield any better performance. This performance stagnation is caused by the fact that the computation gain from parallelism is offset by the necessary communication overhead. The set of results presented in Figure 8 is obtained from a small database characterized by (T10.D20K.N200); average transaction length is 10 items in a database of 20K and the transactions are created out of the possible 200 data items. This graph shows that when more than three agents are used the gain increases, but the CTC ratio rapidly decreases. After more than three processors are employed the speedup gain begins to slow down. This is caused by the fact that the size of the database has shrunk so much that the mining time (computation) has dropped dramatically. This drop leads to shorter computation time and relatively longer communication time. Such a drop is advantageous only up to the optimal region (Region 2 in Figure 9), and further drop however would mean poor performance that requires correction (Region 3 in Figure 9) to reverse the problematic trend.

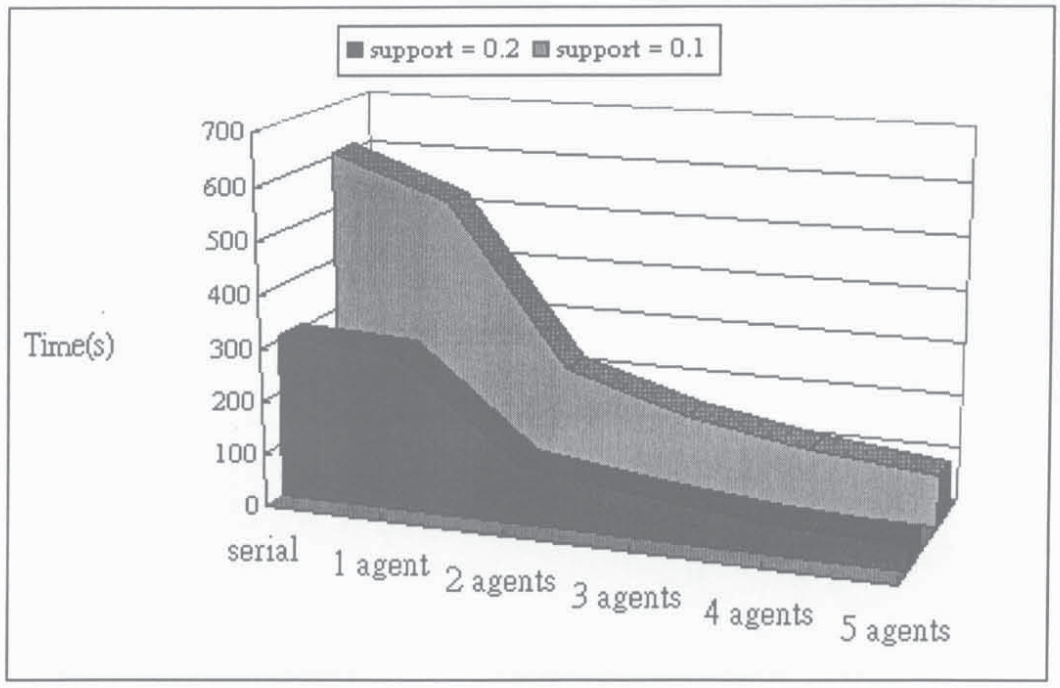

FIGURE 8. MINING TIME OF PARALLELIZED MAA VERSUS NUMBER OF AGENTS ON DIFFERENT MACHINES (T10.D20K.N200)

Since every slave in the MS mining model may have very different CTC ratios at any of the mining passes, those with low CTC ratios would drag the good performers behind. This kind of drag makes the overall distributed mining performance slower than it should be as illustrated in Figure 8, where addition of extra agents do not make the mining process any faster due to too low a CTC ratio. In fact, the aggregation approach was proposed to resolve this problem in a scalable manner (Region 3 of Figure 15).

The principal idea of aggregation is to combine encoded structures that yield low CTC ratios to get rid of the extra slave agents. In this process the overall *ComT* would also be reduced due to less number of interacting agents. Since aggregation would incur significant communication costs for data exchange and transfer in the communication process, it must be steered in a scalable fashion with care.

Scalability in the context of this thesis means satisfying the following general basic criteria, namely:

a) There should be no artificial limitation to the number aggregations (or split when it is involved).
b) The aggregation operation is reversible if it pairs with the split operation.
c) The aggregation (and/or split) should not be hindered by the infrastructure of the underlying network, except its boundary.

There are many parameters that can be chosen for steering mining scalability. In the investigation described in (see page 76, [p1]), the aggregation algorithm was based on a *computation time ratio* (CTR) between two successive mining passes. In fact, the CTR can be defined as one of the following:

a) 1st: "*the T_{MT} (actual mining time) for the current mining pass (T_c) over that for the previous pass (T_p)*".
b) 2nd: "*the CT (computation time) for the current mining pass (CT_c) over that for the previous pass (CT_p)". The CT by the server would include all the timing elements (e.g. queuing) other than the communication ones, the latter are generalized as the communication time (ComT).*
c) 3rd: "*the RTT (roundtrip time) for the current mining pass (RTT_c) over that for the previous pass (RTT_p)". The service RTT is the time taken from the point of request to the point when service result is returned and received.*

In the [p1] investigation the first (1st) definition was adopted (the FAA in Figure 16). When the slaves pass their partial counts for the potential large itemsets back to the master, they also indicate their actual data mining time T_{MT}, simply called T_c or T_p

here, to the master, which records them for scalability steering purposes. In this case, the master combines the partial counts to determine the actual large itemsets, from which the large candidate itemsets C_k for the next mining pass will be generated. Before passing the new C_k to the slaves for the next round, the master assesses whether aggregation for the under-performing slaves should be performed. To summarize, the assessment is based on the following proposed algorithm:

```
If ((T_c/T_p) ≤ 0.5) then {
    Aggregate ();              /*aggregation needed for the 100% mining time delay*/
    Pass_on_to_current_slaves (C_k);   /*for the next round of mining */
}  Else
    pass_on_to_slaves (C_k);           /*new candidate large itemsets to old slaves*/
```

At the time of this early investigation, simple rules were tried out and they consist of the following:

a) Two under-performing encoded structures are aggregated into one and one of the two slaves will be chosen randomly to continue the mining process; the other is deleted.

b) A lone under-performing encoded structure is aggregated with a randomly chosen candidate that continues with the mining; the under-performing slave is deleted.

The aggregation of the encoded structure for the logarithmic decoding is a simple merging operation. Since every transaction is an encoded value defined by $V_t = \sum 2^x$,

where x marks the ordinal position for an item, aggregation is accomplished simply by appending one encoded structure to another. Except for the aggregation algorithm, the rest of the testing environment is the same as for Figure 9. The test results in Table 1, with the database characterized by (T10.D20K.N200), indicate that the proposed aggregation approach can indeed enhance the mining performance by maintaining a reasonable CTC ratio. The performance improvement is consistent with aggregation occurring at different passes [p1]. The experiments in this case were performed in a controlled and dedicated Intranet environment so that the communication overhead and workload would remain relatively constant. This is important because we then know that the change in the CTC ratio is mainly due to the data size.

TABLE 1. RESULTS FOR SCALABILITY TESTS FOR DATABASE characterized (T10.D20K.N200)

Number of agents→	**3 agents (total mining time units**	**4 agents (total mining time units**	**5 agents (total mining time units**
No aggregation	98 s	62 s	50 s
Aggregation applied	78 (20.4% performance improvement)	47 (20.2% performance improvement)	35 (30% performance improvement)
Number of aggregations	1 that happened on the 3rd pass (L_3)	2 that happened both on the 3rd pass (L_3)	1st on the 3rd pass (L_3) and another on the 4th pass (L_4)

In fact, many experiments were performed to investigate the impact of the CTC and Figure 9 shows the result from another experiment with a smaller database [p4] and a CTR of T_c/T_p as well. In this case the mining process augmented by aggregation always performs better than the one without. In this investigation it was found the CTC would have serious impact on system performance and this means using more

agents may not necessarily generate more performance. The CTC can be improved by aggregation and this finding has paved the way for the proposal of the S^2AF concept, which combines data splitting and aggregation in a dynamic manner, as the situation requires.

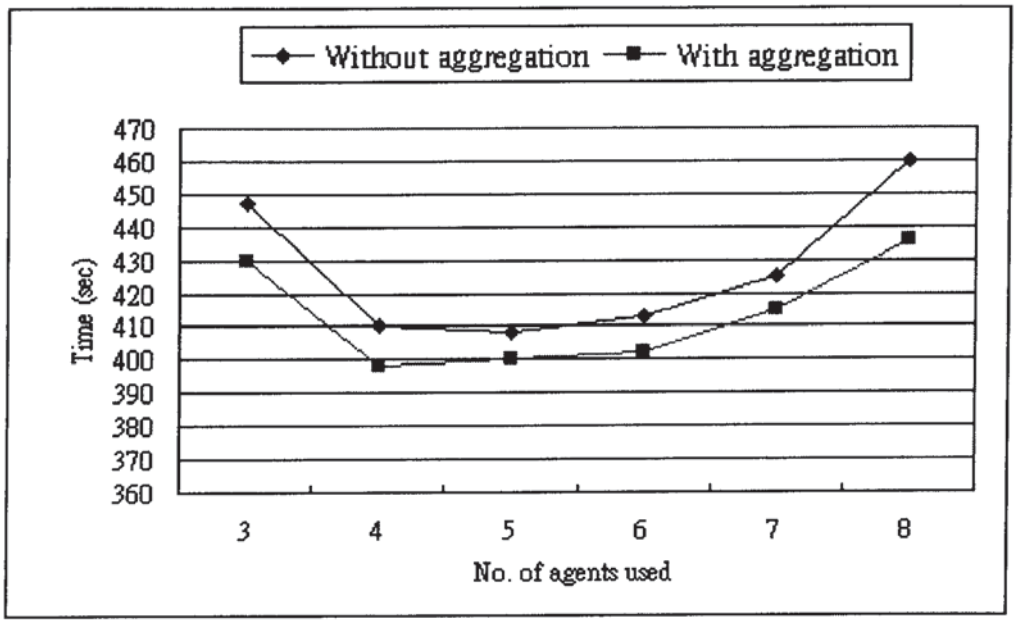

FIGURE 9. MINING TIME OF PARALLELIZED MAA WITH AND WITHOUT AGGREGATION (T10.D10K, N100)

Chapter 3

Speeding Up Apriori

This chapter studies the impact of encoding and decoding on performance with the proposed modified Apriori (see page.77, [p7]) for sequential execution. The objective is to explore how sequential data mining algorithms may be improved.

3.1 Binary Encoding and Decoding

Binary encoding means representing the data items with binary bits so that a transaction can appear as a binary string. Binary decoding means that a decoding method deciphers information encoded in the form of binary strings.

3.1.1 Impact of Encoding and Decoding Scheme

The traditional approach for mining association rules from very large databases consists of the following sequential steps, as demonstrated by the traditional Apriori algorithm (Figure 10):

a) Read a large chunk of data into the primary memory,

b) Perform the sequential search,

c) Read more data from the secondary memory if the mining process for the current level is not finished and repeat this I/O operation until no more level to mind, and

d) Repeat step c) for every level.

From the above steps, it is clear that the I/O cost can be reduced provided the encoding system could place more data into the primary memory so that the subsequent number of I/O accesses for each level of mining can be minimized. The concept for the sequential mining is illustrated in Figure 11, which also differentiates the traditional Apriori (Figure 10) and the MAA (Modified Apriori Algorithm) proposed in the intermediary work (Figure 12). The MAA works with binary coding and either one of the three proposed binary-based decoding approaches [65],[66] is used.

```
1. L1=[frequent-1 itemsets]          /* Initialization: read whole database to identify L1 large itemsets */
2. For(k=2; Lk-1!={};k++){           /* Loop until no more large itemset */
3.   Ck=apriori_gen(Lk-1);           /* Generate candidate itemsets CK from Lk-1 for level k*/
4.   For(m=0; m<number_of_transactions_in_database; m++){
5.     count_itemsets_for_Ck [Read_database(transaction_m)];}  /*Finding Lk 's large itemsets*/
6.   If(any_itemsets_for_Ck ≥Minsup) Lk=TRUE ; }   /*large itemsets for level k (Lk) exists*/
7.   Find_association_rules_for_large_itemsets_at_all_levels;   /* For all levels */
```

FIGURE 10. THE TRADITIONAL APRIORI ALGORITHM

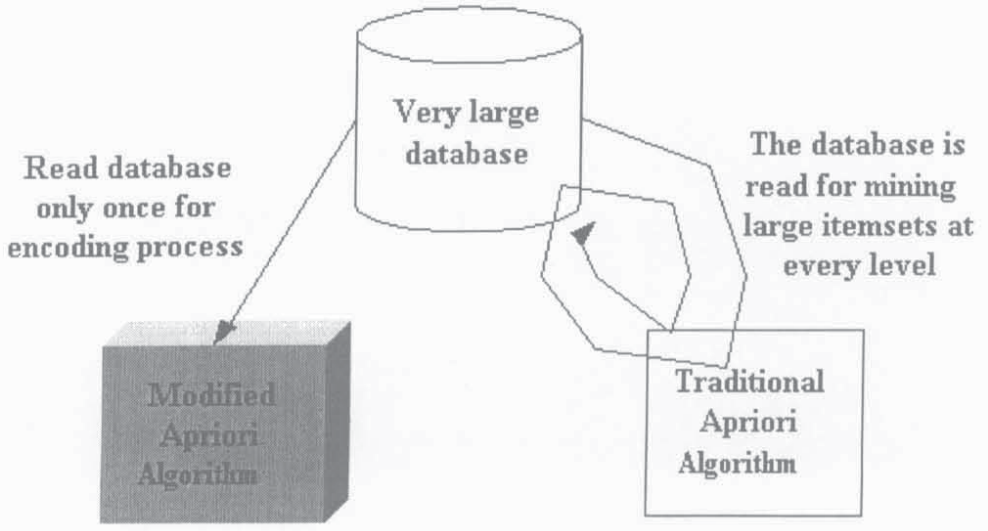

FIGURE 11. THE DIFFERENCE BETWEEN TRADITIONAL APRIORI ALGORITHM AND MAA

```
1. L1=[frequent-1 itemsets] &        /* Read the whole database to identify large itemsets L1
3.  encode (whole_database);         & encode every transaction by the rule Vt=Σ2^X ; X=TRUE*/
3.  For(k=2; Lk-1!={};K++){                /* Loop until no more large itemset */
4.    Ck=apriori_gen(Lk-1);                /*Generate itemsets CK from Lk-1 for level k*/
5.    For(m=0; m<no_transactions_in_encoded_database; m++){
6.      decode_transaction_m_for_level_k (encoded_database);      /*Work on Vt value*/
7.      count_itemsets_for_Ck (transaction_m);                    /*For finding Lk 's large itemsets*/
8.    }
9.    Lk=[itemsets_for_Ck ≥Minsup];      /*Identify large itemsets (≥Minsup) for Level k→Lk)*/
10. }
11.Find_association_rules_for_large_itemsets (Lk);   /* From large itemsets at all levels */
```

FIGURE 12. MAA WITH BINARY ENCODING AND DECODING SCHEMES

3.1.2 Detail of Binary Encoding

There are two mandatory requirements for the bit-encoding method: (a) the database should be read only once within the whole life cycle of data mining, and (b) memory utilization should be maximized. In the proposed encoding method, as part of the intermediary work, every item in a transaction is represented by a 2^X value, where X marks the item's physical ordinal position in the transaction. The whole transaction is then represented by its unique encoded value. The proposed bit-encoding concept that transforms a large database into its miniaturized and manageable form is exemplified by Table 2, where the encoded value for the first transaction is equal to $2^1 + 2^2 + 2^3 = 14$.

TABLE 2. EVERY ITEM IN A TRANSACTION IS REPRESENTED BY ITS 2^X VALUE (X IS A BIT'S PHYSICAL POSITION)

	2^0	2^1	2^2	2^3	
X →	1	2	4	8	
	CAT	DOG	RABBIT	CHICKEN	
1	0	1	1	1	14
2	0	0	1	0	4
3	1	0	1	0	5
4	0	1	0	1	10
5	1	0	0	1	9
	...	...	...	...	

Encoded variable ($V_t=\Sigma 2^X$)

3.1.3 Details of Three Decoding Methods

The decoding method can seriously affect the performance of the data mining process. For demonstration of this point, the test results from three different methods will be compared. The three methods are as follows:

a) *Basic decoding*:

This algorithm of the following logic is executed repeatedly until i, which initialized to the encoded value V_t, is reduced to 0:

```
For (X= number_of_items; X>0; X--) {
    if (i ≥ 2^X) then {
        i=i–2^X;
        X^th_item_in_transaction = TRUE;
    }
}
```

In this approach the encoded value of 14 in Table 2 would represent the "*TRUE states*" for those items in the $2^{X=1}$, $2^{X=2}$, and $2^{X=3}$ positions of the first transaction.

During the decoding operation the encoded transaction is scanned bit by bit, starting from the most significant bit first.

b) *Binary decoding*:

This approach is based on the *basic decoding* approach 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.

```
Binary_search_for_start_position(Number_of_items) {
    Top = Number_of_items;
    While (Top >= Bottom) {
        Middle = (Top + Bottom) / 2;
        if (i >= 2^Middle )AND (i < 2^Middle+1 ) then {
            return Middle;
        } else if ( i < 2^Middle )
            Top = Middle – 1;
        Else { Bottom = Middle + 1; }
    }
    return – 1;
}
```

c) *Logarithmic decoding*:

In this approach, which saves decoding time by eliminating the entire scanning process, the following logic is executed repeatedly until i (initialized to the encoded value V_t) is reduced to 0:

```
i = i – 2 ^integer(log2(i));
[integer(log2(i))]th_item = TRUE;
```

The *integer(log_2(i))* 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_t = 14 successively.

3.2 Test Results for Impact of Binary Encoding/Decoding

These results were collected from experiments performed in the UNIX/Java environment. The Java test programs written for the MAA were executed sequentially in a SUN station.

The binary encoding method maximizes memory utilization in a predictable manner [p7]. This is achieved for the following reasons: (a) memory usage is economized by encoding items in bits instead of bytes, and (b) bit representation is linear. On the contrary, if traditional Apriori reads and decodes items that are encoded in a predefined number of bytes, then the I/O cost would increase with the number of items per transaction. For verifying the argument that memory usage is linear and economical for binary encoding, many experiments were performed with different database sizes generated by the public IBM package. Figure 13 demonstrates the trend pinpointed by the results from these experiments. For the presented case the database had 100,000 transactions (D100K) constructed out of 1000 (N1000) possible items. The three aforementioned decoding methods were evaluated in different experiments. It was found that the *logarithmic approach* was consistently the most efficient among the three. The logarithmic approach forms the basis for the distributed test programs written for the mobile agent environment.

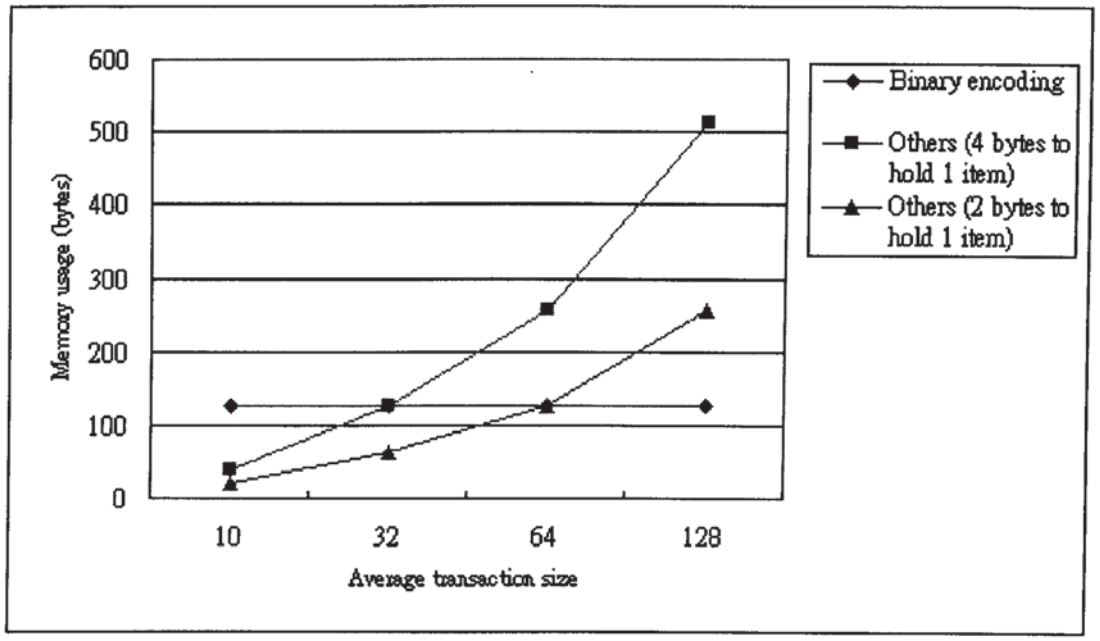

FIGURE 13. LINEAR MEMORY USAGE WITH BINARY ENCODING FOR EACH TRANSACTION (1000 DIFFERENT ITEMS)

Each test case for studying the impact of binary encoding/decoding involves a specific combination of database size and number of items. Figure 14 demonstrates the difference in performance among the four algorithms, namely, the traditional Apriori (4 bytes are used in this case for encoding a data item), and three variants of the MAA algorithm. Each MAA variant embeds either the basic, the binary, or the logarithmic decoding method. The database for producing the test data for the comparison was generated by the public IBM package [76] with the following statistics:

a) Total number of transactions in the database (D) is 1 00,000 (100K).

b) Average number of items per transaction (T) is 10.

c) Total number of items in the database (N) for forming transactions is 200.

d) For large itemsets, the number of transaction patterns is 1000; the average number of items in a transaction (I) is 4.

In the data mining experiments, large itemsets were tallied up to the 8th level ($L_{k=8}$).

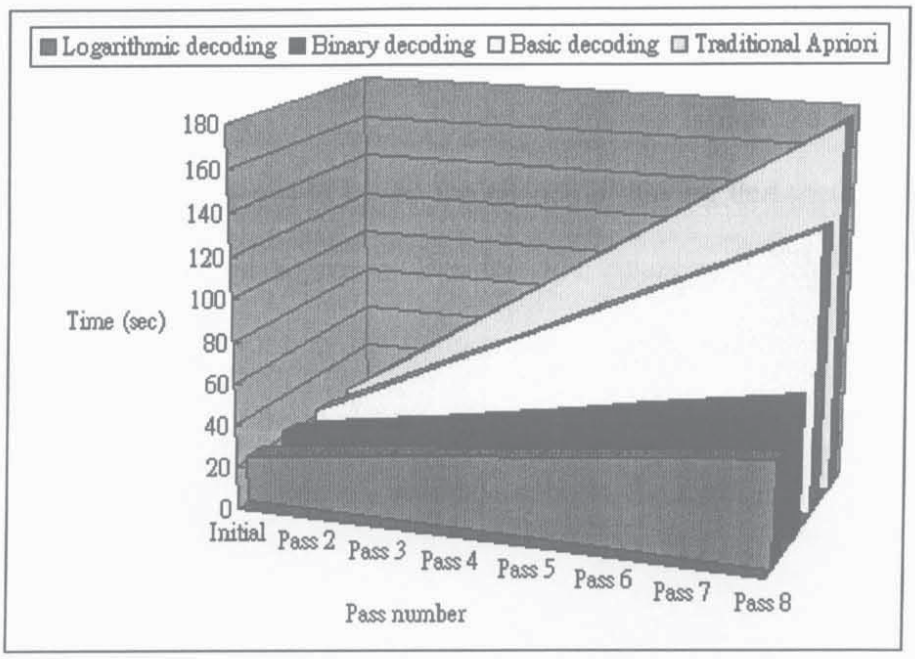

FIGURE 14. (REPLICA OF FIGURE 6) COMPARISON OF FOUR ALGORITHMS (T10.I4.D100K.N200)

3.3 Connective Summary

The aim of in this investigation and experiments in the UNIX environment is to study how the performance of the traditional Apriori algorithm can be improved by binary encoding/decoding. The approach is to modify the Apriori by incorporating into the former with the proposed encoding and decoding mechanisms. In order to demonstrate the importance of efficient decoding to high data mining performance, three methods, namely, basic, binary, and logarithmic were evaluated. These three decoding methods were devised with respect to the bit-based encoding approach that maximizes memory utilization in a predictable manner. The findings from different experiments with sequential programs have confirmed that the logarithmic decoding method is the most efficient among the three. It can speed up the data mining

process significantly as demonstrated in the performance comparison. In the latter work the modified Apriori algorithm (MAA) would always carry the logarithmic decoding approach by default, in the distributed mobile agent based environment, namely, the chosen Aglets [75]. One important finding is that the linear memory usage by binary decoding would lessen the chance of having that memory overflow problem in the AprioriTid approach. Besides, the binary approach would put the simplified data representation in the main memory at the start, and therefore it is more efficient than the AprioriHybrid approach, which puts the simplified data representation, namely the table of candidate sets, in the main memory only at the later stage.

Chapter 4

Theoretical Foundation for the Proposed Framework

4.1 Introduction

The theoretical foundation or rationale for the S^2A (*scalable split/aggregate*) mechanism is illustrated in Figure 15 under the following assumptions:

a) Agents collaborate in a client/server relationship.

b) The computation time (*CT*) by the server would include all the timing elements (e.g. queuing) other than the communication ones, the latter are generalized as the communication time (*ComT*).

c) The service *RTT* (roundtrip time) is the time taken from the point of request to the point when service result is returned and received.

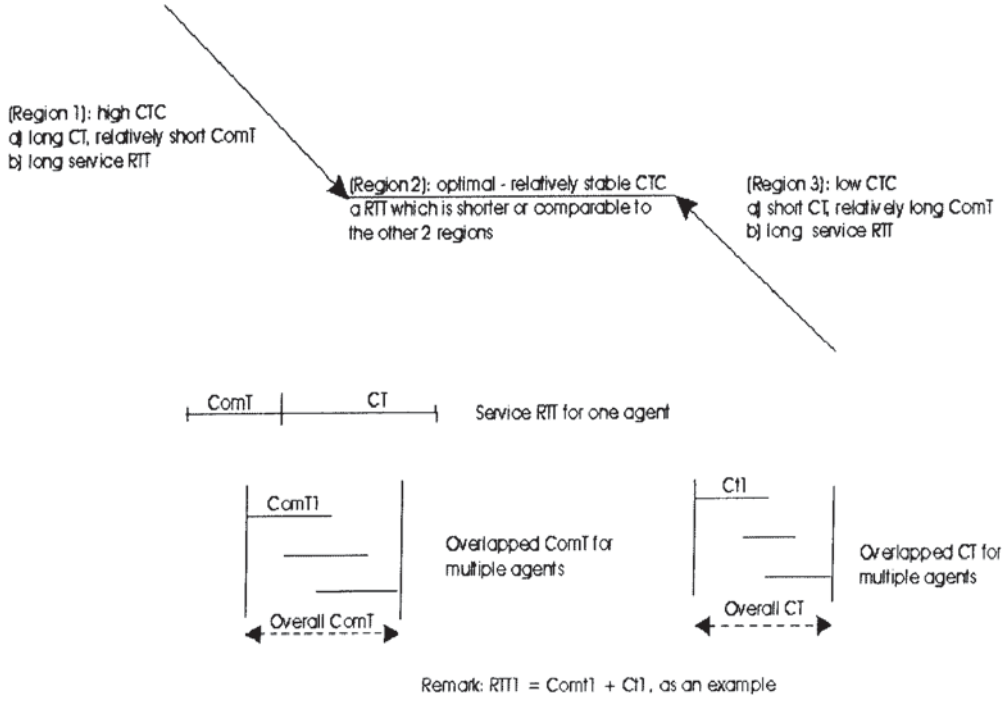

FIGURE 15. THE RATIONALE FOR THE S^2A OPERATION

From the above assumption, the service *RTT* can be represented as $RTT = CT + ComT$. In a general sense, a high CTC ratio implies that the system is spending its precious time usefully for computation rather than wasted as communication overhead.

Figure 15 has three distinctive conceptual regions as follows:

a) *Region 1*: This region is characterized by high CTC ratio, which means a lower *ComT* than *CT*, and a typical long *RTT*. This phenomenon exists because the data blocks for mining agents are huge and/or complex, requiring a long *CT*. The solution to lower the *CTC* ratio is to raise mining efficiency (speedup) by agent replication accompanied by splitting large data blocks. That is, use computation parallelism to bring down the *CT*.

b) *Region 2*: In this optimal region, it is expected that at the steady state the *CTC* would stay reasonably high and relatively constant. The *RTT* at this stage should be either shorter (especially Region 3) or comparable to the other two regions. The main task of an S^2AF algorithm is to upkeep this state for good performance. There is however a special case in which the S^2AF approach may not be effective; that is, when both *CT* and *ComT* are long. But the detailed exploration of this problem is left as an item for the future because of time constraints.

c) *Region 3*: It is characterized by a low CTC but relatively long *ComT* and long *RTT*. The low *CTC* is due to the fact that the size of the data to be mined in the subsequent passes/levels (i.e. computation time requirement) is becoming progressively smaller. The proposed solution is to lift the CTC ratio by aggregation, which would increase *CT* and shorten the overall *ComT* and *RTT* due to less interacting agents. The basic idea is to bring it back to Region 2.

The foundation in Figure 15 is the basic conceptual framework for the *scalable split/aggregate (*or simply S^2A*)* approach proposed in this thesis for mobile agent based distributed data mining over the Internet. The idea is to maintain the Region 2 of operation by:

a) Creating multiple mobile agents: Each of these agents would mine a block of data partitioned from the huge target database. An agent would replicate (*split)* itself, and each of the two replicas or clones would mine a smaller data block partitioned from the original database that was assigned to the "parent" agent that invoked the cloning.
b) Merging of data blocks: Two data blocks will be merged if the CTC ratio is lower than the predefined threshold, and this means that the extra (excess) agent would be eliminated.

The *split and aggregate* mechanisms in general depend on carefully predefined threshold values. For example, the findings in this research (see page 76, [p1, p2, p5, p6]) indicate how the *CT*, the actual mining time (T_{MT}) (sometimes simply referred to as T_c or T_p in the algorithms), and/or the *RTT* can be deployed to gauge the fluctuation in the CTC ratio. The gauging is more accurate provided that the *ComT* remains relatively constant. Sometimes a specially designed protocol [p1] may be needed for the agents to return the current values of the chosen parameters for gauging the CTC ratio. The aggregation approach, namely, the First Aggregation Algorithm (FAA), proposed in [p1] for adjusting mining performance, is shown in Figure 16. Table 3 shows the test results from this approach, with the given database (T10.D20K.N200) generated by the chosen IBM package. The parameters for the aggregation approach are as follows:

a) T_c is the actual mining time for the current mining pass.

b) T_p is the actual mining time for the previous pass.

c) C_k is the large candidate itemset for the next mining pass.

d) *T10* means 10 items per transaction on average out of the possible 200 items (N200) in a database of 20K transactions (D20K).

In the *First Aggregation Algorithm*, the actual mining time (T_c or T_p) does not include the overhead for context switching and queuing, but simply the time to find C_k.

```
If ((Tc/Tp)≤0.5) then {
    aggregate ( );                          /* 100% mining time decay          */
    pass_on_to_current_slaves (Ck);         /* Ck for the new round of mining  */
} Else
    pass_on_to_slaves (Ck)                  /* new candidate large itemsets to old slaves*/
```

FIGURE 16. FAA AS AN AGGREGATION ALGORITHM EXAMPLE

TABLE 3. (REPLICA OF TABLE 1) RESULTS FOR SCALABILITY TESTS FOR DATABASE CHARACTERIZED (T10.D20K.N200)

Number of agents→	**3 agents (total mining time units)**	**4 agents (total mining time units)**	**5 agents (total mining time units)**
No aggregation	98 s	62 s	50 s
Aggregation applied	78 (20.4% performance improvement)	47 (24.2% performance improvement)	35 (30% performance improvement)
Number of aggregations	1 that happened on the 3rd pass (L_3)	2 that happened both on the 3rd pass (L_3)	1 on the 3rd pass and another on the 4th pass (L_4)

4.2 The 5E Investigation

Despite the importance of parallel data mining, there is a lack of publications that address the issue directly and in sufficient detail. Recent and relevant publications concentrate mainly on theoretical discussions, with little empirical proof. The literature that describes the five essential (5E) elements (see page 77, [p5]) that can yield high performance in real-time data mining over the Internet is abundant because of their relative maturity, even though the discussions of the elements are somewhat scattered. The lack of a comprehensive methodology to exploit the synergy by the five elements over the Internet for distributed data mining of very large databases has motivated some of our investigations.

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

a) Identify the suitable model for inter-object communications in data mining.
b) Identify a suitable parallelization method to support the two forms (active and passive) of object-based data mining.
c) Find a stable tool to generate the very large databases necessary for the mining of association rules in the verification tests and experiments.
d) Choose a stable distributed programming system to generate the logically-distributed object-based test programs.
e) 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. As identified in another previous work

[69], there are three communication models for inter-object interactions, namely, *master and slaves*, *iterative stages*, and *process network*. It was 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.

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 could be made between this reference and the test results from the 5E verification exercise. The reference establishment task involved:

a.) Identifying three different decoding methods for the bit-encoding scheme.

b.) Incorporating these methods separately into traditional Apriori to create three different sequential versions of modified Apriori or MAA.

c.) Running these three different MAA versions and compare their results to identify the fastest version among them.

d.) Adopting the fastest version identified in step c) above as the basis for the SPDM parallelization, and also using it as the reference for comparing with the 5E test data.

The test results from steps: c) and d) are compared to verify the efficacy of the 5E framework. The modification of the traditional Apriori is achieved by embedding the chosen decoding mechanism, shown as *highlighted program statements* in Figure 12. The ultimate aim of encoding is to represent the whole database in the main 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 eliminating the overlay structure because a large database is partitioned for parallel operations.

The databases for the encoding/decoding experiments were 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.* From the comparison of the four curves in Figure 14, it is obvious that logarithmic decoding has the best performance with the same large database.

4.3 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 [52]. 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 [50]*) method (Figure 17) is a natural choice. In the SPMD approach, the data mining program object (A in Figure 17) 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 main 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.

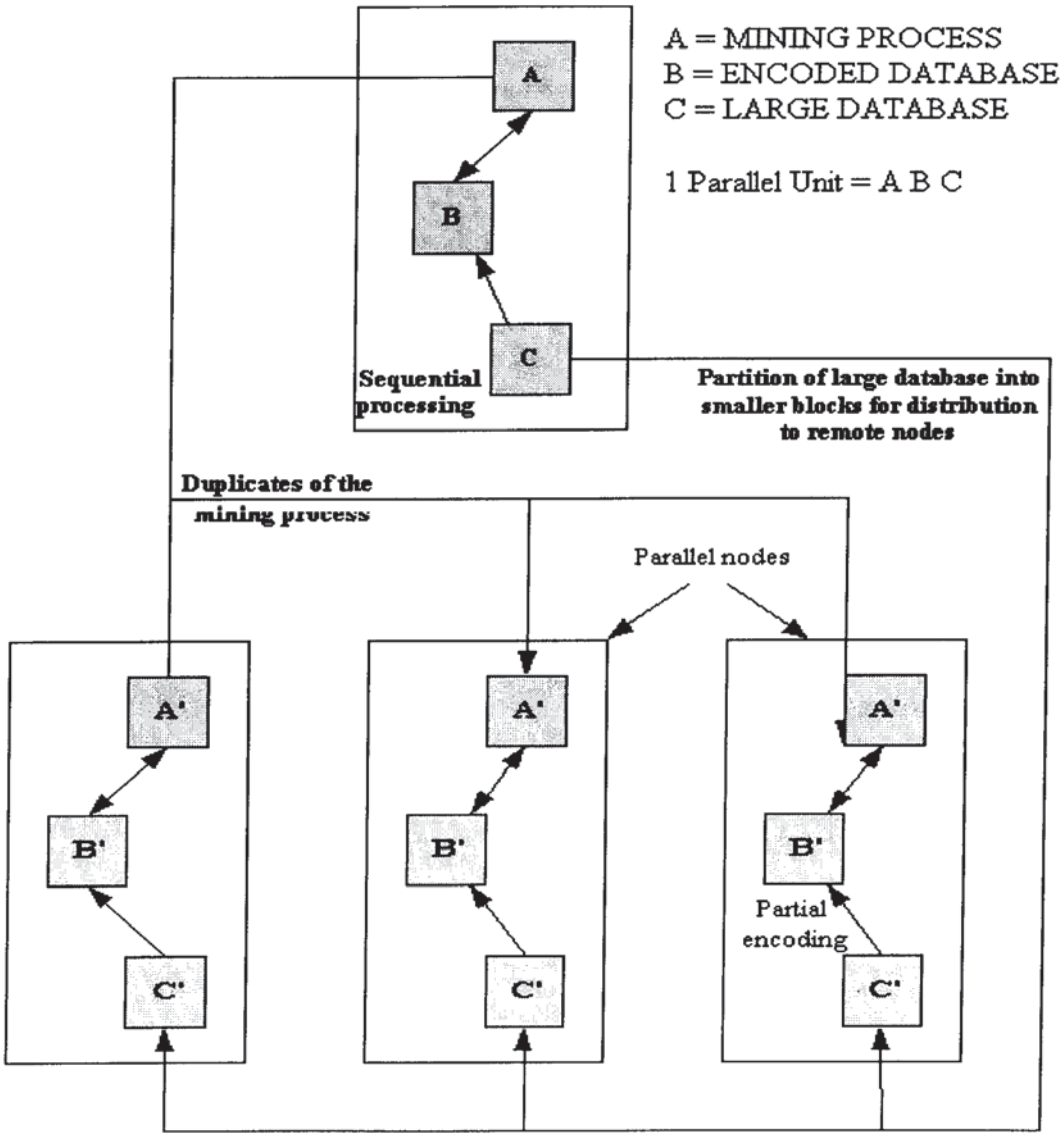

FIGURE 17. PARALLELIZATION BY SPMD – DATABASE PARTITIONED AND DISTRIBUTED; THE "MINER" (APRIORI IN THIS CASE) IS DUPLICATED AND DISTRIBUTED

4.4 Parallelizing the Logarithmic Decoding Approach

The platform for testing is the stable Aglets mobile agent platform [75] running over the network in our laboratory and the data for the tests was generated by the IBM

data generation package [76]. The fastest logarithmic decoding method forms the basis for the MAA in the 5E verification tests. Each test is mining of association rules from a very large database. That is, every mobile agent is a copy of the MAA shown in Figure 12, but with bit-encoding and logarithmic decoding. With the same database (D100K.N200) the test results (Figure 18) show that the parallelized MAA (marked Parallelized MAA on the graph) performed much better than the traditional Apriori algorithm (with byte encoding; marked Traditional Apriori Algorithm) and the sequential version of MAA before parallelization (marked MAA).

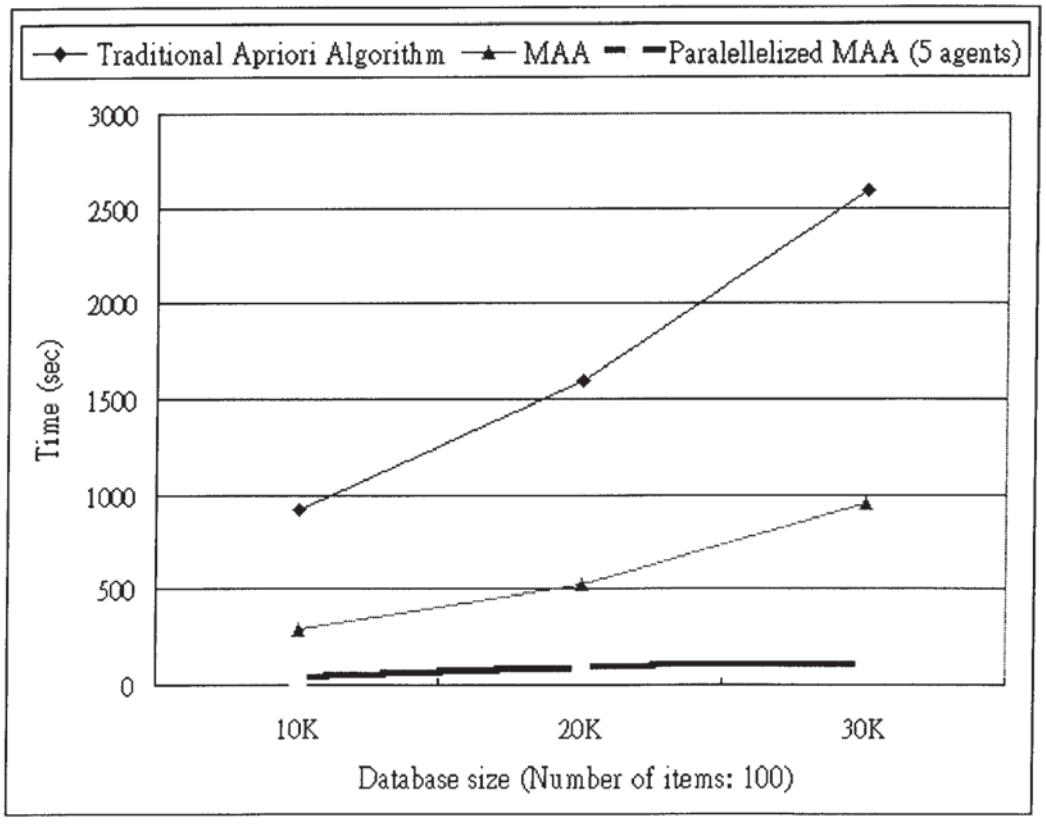

FIGURE 18. PERFORMANCE DIFFERENCE BETWEEN SEQUENTIAL AND PARALLEL DATA MINING IN THE AGLETS ENVIRONMENT (EVERY PROGRAM IS AN AGLET (AGILE APPLET))

4.5 Degree of Overlapped Parallelism

In this investigation, the impact by the degree of overlapped parallelism among collaborating objects or agents is investigated. The assumption is that these objects would migrate for various reasons such as for better performance and reliability. Conceptually, if one observes the object behaviour with time grids (Figure 19), then the following can be defined [63]:

a) C_j as the probability for j object migrations to occur within a time grid,

b) ρ is the probability for an object to migrate,

c) O as the overhead incurred by a distributed program due to object migrations,

d) T_o as the overhead for an object migration,

e) M as the total number of successive time grids in life span of the distributed software, and

f) N is the total number of mobile objects in the software.

β_1 indicates the degree of overlapped parallelism among the migrating objects.

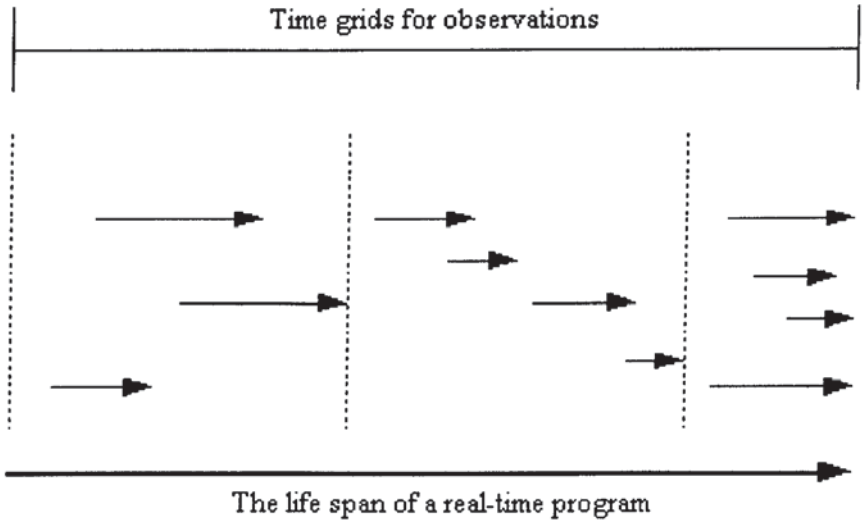

FIGURE 19. OBSERVATIONS MADE IN TIME GRIDS

$$C_j = \binom{N}{j}\rho^j(1-\rho)^{N-j}; O = T_o M[\sum_{j=1}^{N} jC_j]^{e^{-\beta_1}} \quad \text{.....(4.5.1)}$$

If all the object migrations were started and finished at the same time within every time grid, the 100% overlap or total parallelism would mean a large β_1 that implies $O = T_o M$. A simpler scenario is that there is total parallel program execution for the MS (master-slave) paradigm over a large network. If we assume the following:

a) T_{actual} is the total execution time for the MS distributed program,

b) S is the number of passes in the barrier synchronization before the program completes execution,

c) β_2 is the degree of overlapped parallelism among the slaves in each barrier synchronization pass,

d) N is the number of slaves, and

e) T_m is the average service time by the slave (excluding the communication time),

then T_{actual} can be defined by equation (8.2), which indicates that a very large β_2 would yield N^0 (total parallelism) .

$$T_{actual} = T_m * S * N^{-\beta_2} \quad \text{.....(4.5.2)}$$

In fact, the preliminary empirical tests of equation (4.5.2) in this thesis and the research work by others show that equation (4.5.2) can indeed be used to predict the performance of an object-based distributed program as long as β_2 can be measured [p2].

4.6 Connective Summary

In this study, the 5E framework is investigated for the purpose of yielding high performance in distributed 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 modified Apriori algorithm (with logarithmic decoding), and the SPDM parallelization method. The 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 distributed data mining over the Internet, and therefore further work along the same line would be meaningful.

The investigation of the impact by overlapped parallelism on object-based distributed programs has revealed that the program performance either in terms of migration overhead or the total program execution time (T_{actual}) is related to the degree of overlapped parallelism generalized here as β. In fact, in the subsequent investigations, the empirical data show that the effect of β has substantial correlation with the computation-to-communication or CTC ratio. More detailed investigations and calibration of β is outside the present scope of this thesis; such in-depth investigations would require a long time as indicated by the preliminary experience in [18].

Chapter 5

The Scalable Split and Aggregate Framework (S^2AF) for Distributed Mining

5.1 Introduction

The rationale for the S^2AF is illustrated in Figure 20, where the CTC ratio is divided into three regions. For the first region (Region 1) the solution is to maintain a reasonable CTC through parallelism and bring the operation to the optimal region (Region 2). In Region 3 the CTC decreases due to a drop in data size and this should be corrected by aggregation. In fact, an effective S^2A algorithm should be able to monitor the rate of CTC decay, and when the decaying rate has deteriorated to beyond a chosen threshold, then it should be the time to aggregate to improve the CTC ratio and bring the operation back to Region 2.

With the help of the roadmap (Figure 32) in the section of "Choice of research Methodology" in the Appendix, different investigations were carried out to explore and understand the importance of the five elements for achieving good distributed computing for which mobile agent based data mining is one form. The investigations include the following: the impact by the method for representing or encoding data in the main memory, the impact of the degree of overlapped parallelism β, which is directly related to *ComT*, and the effect by an aggregation approach. The experience from these investigations has provided the insight into

how to maintain a high CTC ratio dynamically and led to the proposal of the novel scalable split & aggregate framework or simply S^2AF.

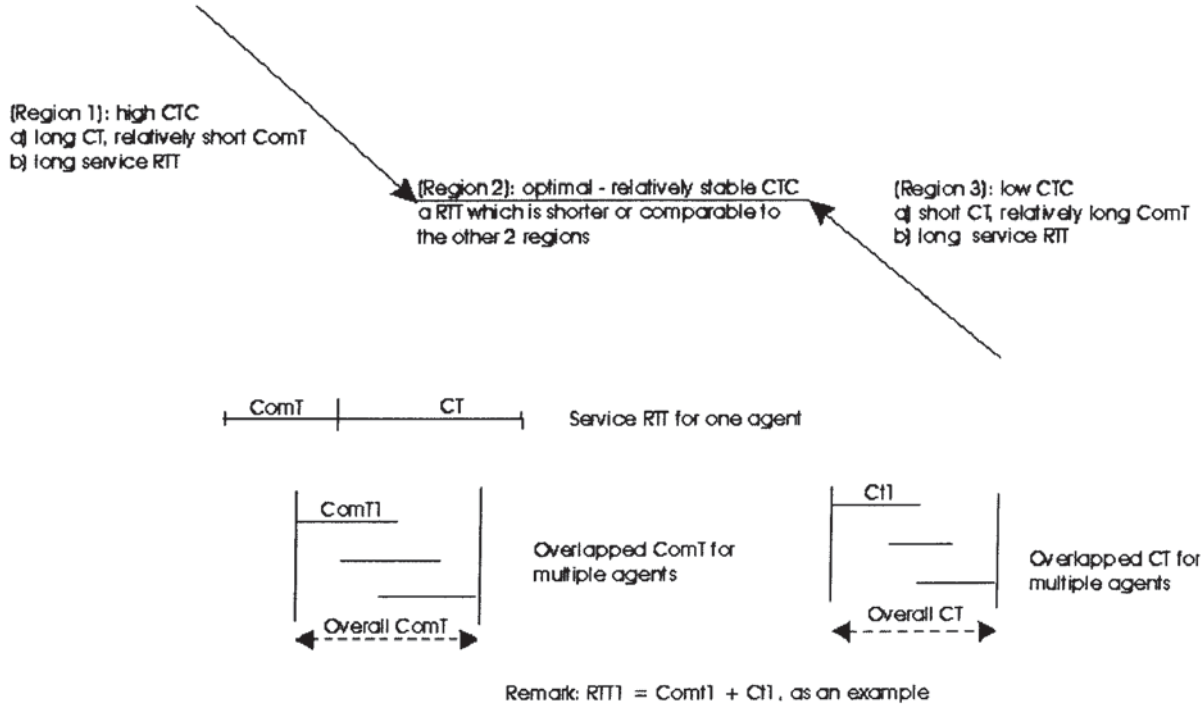

FIGURE 20. (REPLICA OF FIGURE 15) THE RATIONALE FOR THE S&A OPERATION

Figure 21 shows the block diagram of the proposed S^2AF. The SPDM paradigm forms the backbone of parallelization in the novel S^2AF concept, in which a single mining algorithm is cloned into multiple "*miners*" that each of them would mine a data block. The data block is either a partition from a target database (passive model) or a distributed database in a distant host (active model). In the mining process the following would happen dynamically:

a) A data block/base is split and an additional miner cloned in case the mining time is too long due to the sequential search.

b) Two data blocks are combined into one and the excessive miner purged. This happens when the CTC ratio is lower than the threshold, due to a low *CT*.

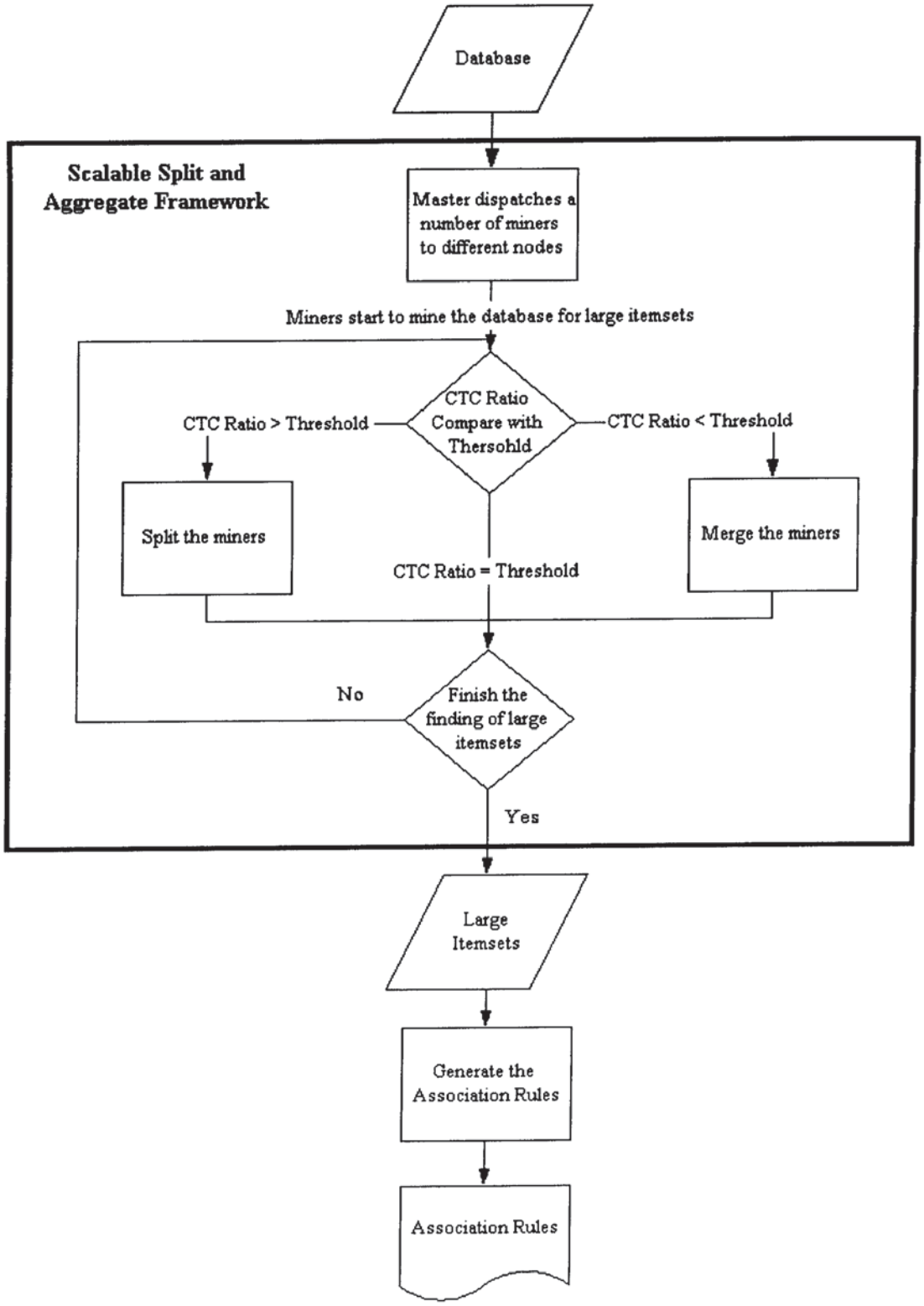

FIGURE 21 THE BLOCK DIAGRAM OF THE S^2AF

The split and aggregate capability depends on the following:

a) The miner or the system must be able to sense the operation conditions and

determine whether it is time to split or aggregate. In fact, split and aggregate can happen in parallel for different constituent agents (in the context of this thesis, the mobile agents) that make up the "mining software".

b) Thresholds must be identified to serve as the points when split or aggregate should happen.

At this stage of the research the following are applied by default:

a) The modified Apriori (with logarithmic decoding) is the "miner" algorithm.
b) The Java-based Aglets is the testing platform.
c) The SPDM is the parallelization paradigm that forms the basis for the split and aggregate operations.
d) The IBM data generation package is the tool for generating the necessary artificial test data.
e) The master-slave or MS paradigm is the basis for the miners to collaborate.

For the tests, it is assumed that the master would determine when a split or aggregation should occur. In the aggregation case, the master would determine which agents should be aggregated, and likewise the master would determine which agent should be cloned so that its present data block would be split.

In the aggregation process, a number of agents will transfer their database to other agents and then they will dispose of themselves. It will reduce the communication overhead due to the number of agents reduced.

In the split process, the master will create a number of new agents (according to the

parameters of the algorithms) and transfer them to some new nodes. And then the master will ask the old agents to divide their databases into a number of partitions (according to the parameters of the algorithms) and transfer these partitions to the new agents for the load balancing process. It will reduce the computation overhead.

The logic to determine when and how the scalable split and aggregate occurs can take many forms. Here, four different S^2AF algorithms [p3] are proposed, namely, the *load balancing algorithm* (LBA), *weighted load balancing algorithm* (WLBA), *naïve load balancing algorithm* (NLBA), *and optimal weighted load balancing algorithm* (OWLBA) to demonstrate how the conceptual S^2AF can be realized. These algorithms would work better in environments where the *ComT* is relatively stable because this means that the long *RTT* is mainly due to the actual mining time T_{MT}, which depends on the size of the data block being mined. In the implementation and tests of the new algorithms T_{MT} is a main parameter. Since the testing environment is dedicated and controlled the overhead for context switching (T_{CS}) and queuing (T_{WT}) are insignificant when compared with the actual mining time (T_{MT}).

The four new S^2AF algorithms, namely, LBA, WLBA, NLBA and OWLBA have not addressed the issue of when the split action that reduces the CTC should stop. There lacks a mechanism to detect whether the optimal region (Region 2) has been reached. Rather, the present algorithms rely on the fact that once Region 3 is reached, the aggregation process would bring it back to Region 2. This means that such back-and-forth region shifts might bring about out undesirable operational

oscillations. How to include suitable detection mechanisms to eliminate these oscillations is left for detailed exploration in the future.

5.2 The Load Balancing Algorithm (LBA)

For the Load Balancing Algorithm (LBA) agents replicate themselves when the database or data blocks need to be split scalably for better performance. The aim is to counteract the effect of long *RTT* due to unreasonably high CTC ratio (Region 1) because of massive data size and/or data complexity. The performance of this algorithm is more predictable when *ComT* is relatively stable. The split begins when the *CT* of the current cycle, namely, CT_c, is either larger than or equal to that of the previous one, namely, CT_p. In contrast, the aggregation process will start once CT_c is smaller than CT_p. The logic of the LBA is depicted as follows:

```
If (CT_c ≥ CT_p) then {
    Clone_agents&Split_data_block( );      /* Replicate agents */
} Else if (CT_c < CT_p)
Merge_data_blocks&Purge_excess_agents()   /* Merge the data blocks together */
```

5.3 The Weighted Load Balancing Algorithm (WLBA)

In this algorithm, split begins when CT_c is larger than or equal to CT_p by a predefined number of times to ensure that the condition $CT_c \geq CT_p$ is indeed a true one. The aggregation process starts once the current *CT* is much smaller than the previous *CT* also by a specified number of times. For the verification experiments, the number chosen by empirical experience was five, and logically the algorithm is depicted as follows:

```
If (CT_c ≥ CT_p * Fold) then { /* In our experiments, Fold is set to 5 */
    Clone_agents&Split_data_block( );     /*Replicate the agents */
} Else if (CT_c < CT_p / Fold )
Merge_data_blocks&Purge_excess_agents()   /* Merge the data blocks together */
```

5.4 The Naive Load Balancing Algorithm (NLBA)

In the Naive Load Balancing Algorithm (NLBA), the total number of mining passes is estimated according to the average transaction length first. Our experience indicates that the number of passes varies with the average transaction length, independent of the changes in other parameters. From the estimated number of mining passes and the rough distribution shape of the mining cycle times, the NLBA determines the number of agents to be spawned initially for the mining process. The logic of the NLBA is depicted as follows:

```
estimate_pass_number = transaction_length / 2;
middle_pass_number = estimate_pass_number / 2;
If (current_pass_number >= (middle_pass_number / 2) &&
   (current_pass_number <= middle_pass)) then {
   Clone_agents&Split_data_block( );     /*Replicate the agents */
} Else if (current_pass_number >= (middle_pass_number) )
Merge_data_blocks&Purge_excess_agents()   /* Merge the data blocks together */
```

5.5 The Optimized Weighted Load Balancing Algorithm (OWLBA)

The experience gained from early experiments indicates that although the three new S^2AF algorithms can find the large itemsets efficiently they all have a shortcoming. That is, these algorithms involve only splitting an agent into two or merging two

agents into one. They cannot determine how many agents should be cloned from the splitting process or how many agents should be merged into one. This shortcoming has motivated the proposal of one more S^2AF algorithm, namely, the *Optimized Weighted Load Balancing Algorithm* (OWLBA).

In this newest algorithm, the number of the replica or clones is determined by the ratio of the mining time (T_c) of the current cycle over that of the previous cycle's (T_p). A larger ratio means more replicas should be produced and the reverse is true for the merging process. The logic of the OWLBA is depicted in the following pseudo-program:

```
Split_check(Tc , Tp) {
        if (Tc > Tp*8) {
                Clone_agents&Split_data_block(8);          /* each agent split into 8 agents */
        } else if (Tc > Tp *4) {
                Clone_agents&Split_data_block(4);          /* each agent split into 4 agents */
        } else if (Tc > Tp 2)
                Clone_agents&Split_data_block(2);          /* each agent split into 2 agents */
        }
}

Merge_check(Tc , Tp) {
        if (Tc*8 > Tp) {
                Merge_data_blocks&Purge_excess_agents(8);  /* every 8 agents merge into 1 agent */
        } else if (Tc*4 > Tp){
                Merge_data_blocks&Purge_excess_agents(4);  /* every 4 agents merge into 1 agent */
        } else if ((Tc*2 > Tp) {
                Merge_data_blocks&Purge_excess_agents(2);  /* every 2 agents merge into 1 agent */
        }
}
```

Chapter 6

Simulation Results

6.1 Experiments using IBM Data Generation Package

In this investigation the effect of overlapped parallelism in object-based distributed computing was examined. There are totally nine Sun Microsystems Ultra-5 workstations in our laboratory are used for the experiments. The environment for the experiments consists of the stable Aglets mobile agent platform running over the LAN in our laboratory. This LAN is part of the PolyU's Intranet. Some of results from these experiments were published in the paper [p5] and the rest of experience became the contribution to another paper [p6].

In the investigation, the motivation is to find a way to support efficient distributed 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 investigations are as follows:

a) Finding the correct pattern for inter-object communications among the collaborating mobile objects in data mining.
b) Identifying the parallelization method that effectively supports both the active and passive forms of object-based data mining.
c) Identifying the strength and limitations of the 5E framework [64]; 5E stands for

five essential elements, namely, object-based parallelism, inter-object interaction pattern, programming model, mobility and hardware architecture.

The tests of the three new algorithms were conducted over the Algets in a less controlled Intranet environment in our laboratory. A very large database for this experiment was synthesized by using the IBM data generation package, and the attributes for this database are summarized in Table 4.

TABLE 4. SUMMARY OF THE DATABASE USED IN THE EXPERIMENT

Average size of the transactions	15 items
Average size of potential large itemsets	4
Correlation between patterns	0.25
Number of patterns	5000
Number of possible items	100

All the tests started with only one slave, which was replicated or disappeared according to conditions specified in the algorithms. Figure 22 compares the mining times for each pass between the pure MAA (distributed but without split and aggregation) and the LBA. The LBA was consistently observed to be more efficient than MAA when the same experiment was repeatedly performed.

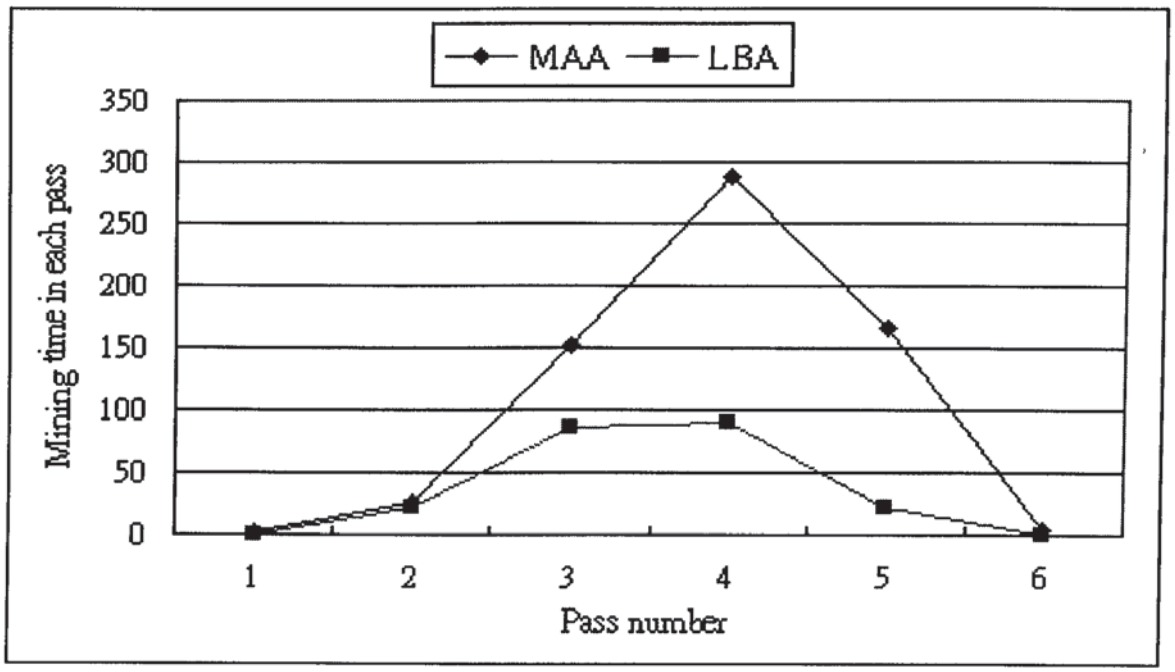

FIGURE 22. MINING TIME FOR EACH PASS OF MODIFIED APRIORI ALGORITHM (MAA) AND LOAD BALANCING ALGORITHM (LBA) WITH SUPPORT COUNT OF 0.02 (T15.D1K, N100)

Figure 23 compares the mining times (each pass) between the distributed pure MAA and the three new algorithms. It is observed that the new algorithms indeed perform better than MAA. Figure 24 shows the number of agents used in every pass in each of the three new algorithms (LBA, WLBA and NLBA) [p3], as well as the FAA (First Aggregation Algorithm [p1]) that was developed in the earlier investigations. From the figure, it is noticed that the number of agents for all these algorithms for the same conditions can be very different.

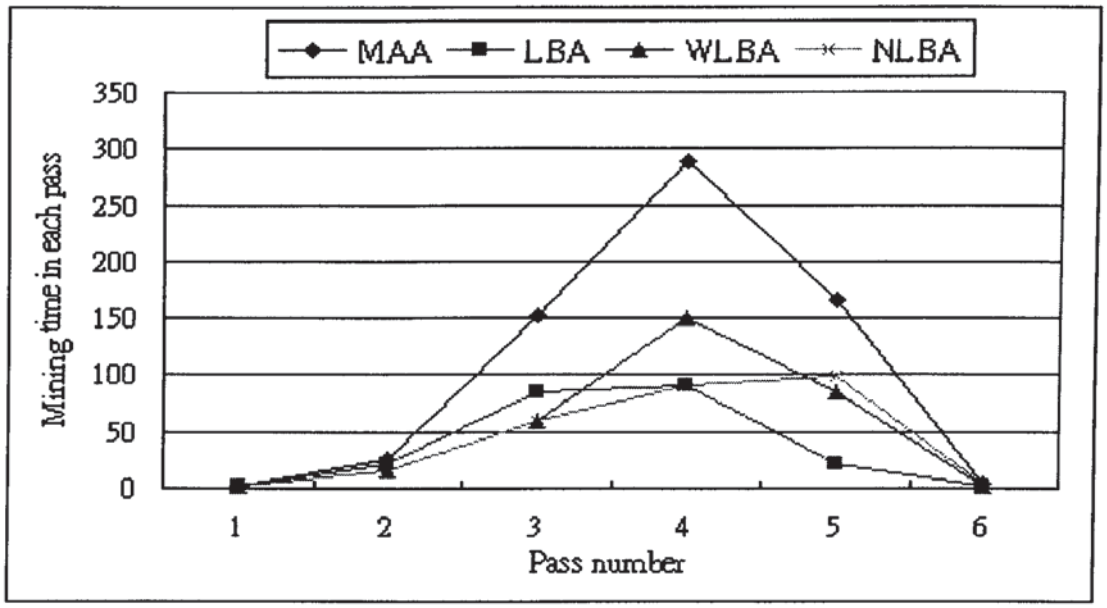

FIGURE 23. MINING TIME OF EACH PASS OF MODIFIED APRIORI ALGORITHM (MAA) AND THREE PROPOSED ALGORITHMS WITH SUPPORT 0.02 (T15.D1K, N100)

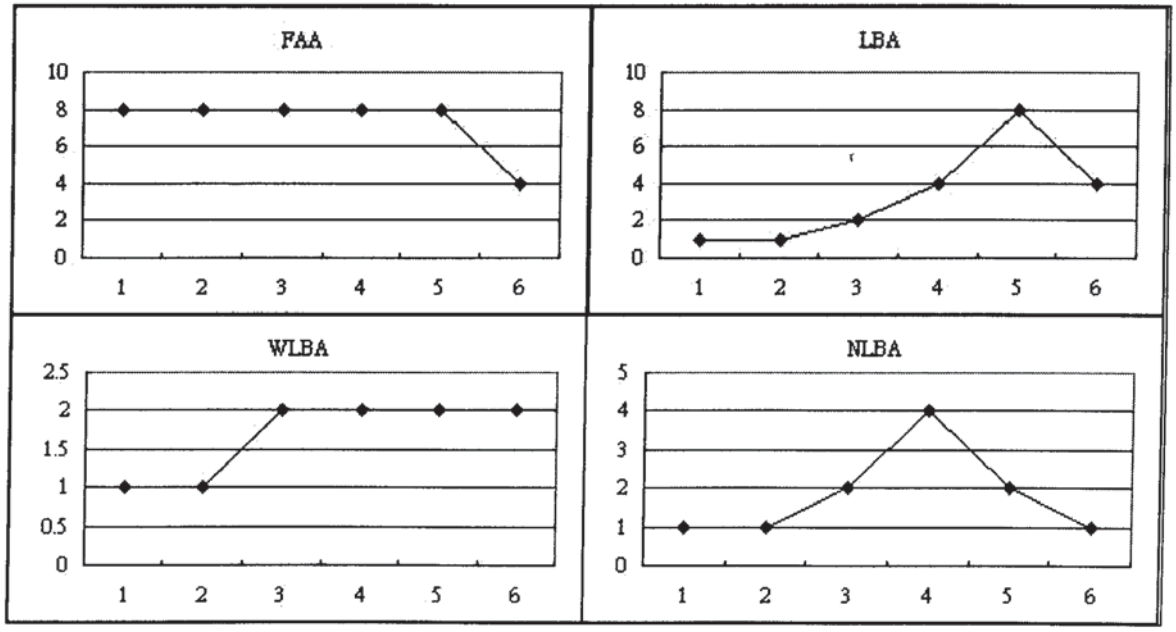

FIGURE 24. NUMBER OF MINING AGENTS NEEDED IN EACH PASS OF THE THREE PROPOSED ALGORITHMS WITH SUPPORT COUNT 0.02 AND A DATABASE OF (T15.D1K, N100)

Figure 25 compares the mining times for the four algorithms (as shown in Figure 23), which by nature balance the load of the agents effectively. That is why these algorithms sometimes were referred to as load balancing algorithms in some of the

published papers [p1-p7]. Databases of different sizes and complexities were used in the different tests under the same operating environment, which is the Aglets mobile agent platform running over the in-house Intranet. For these tests the support count was 0.02, and it was found that the mining pattern depends on the average size of the transactions. The main observations from Figure 25 are as follows:

a) WLBA is faster than the LBA and the NLBA when the data size is smaller than 1K, even though the WLBA uses less number of agents. This is due to the higher communication cost for the other two algorithms.

b) When the data size is 1K or more, the LBA is more efficient. Although the LBA uses more agents (maximum 8), the benefit from the computation effectively offsets the communication overhead. Yet, the WLBA is sometimes faster than the NLBA because the computation gain from parallelism by the latter (maximum 4 agents) is not enough to offset the communication overhead.

c) When the data size is between 3K and 4K, LBA is faster than WLBA and NLBA, but NLBA is faster than WLBA, and these phenomena can be attributed to the high CTC ratios.

d) The three new algorithms perform better than the FAA until the database has reached a size of 10K or more. After that, the FAA always performs better.

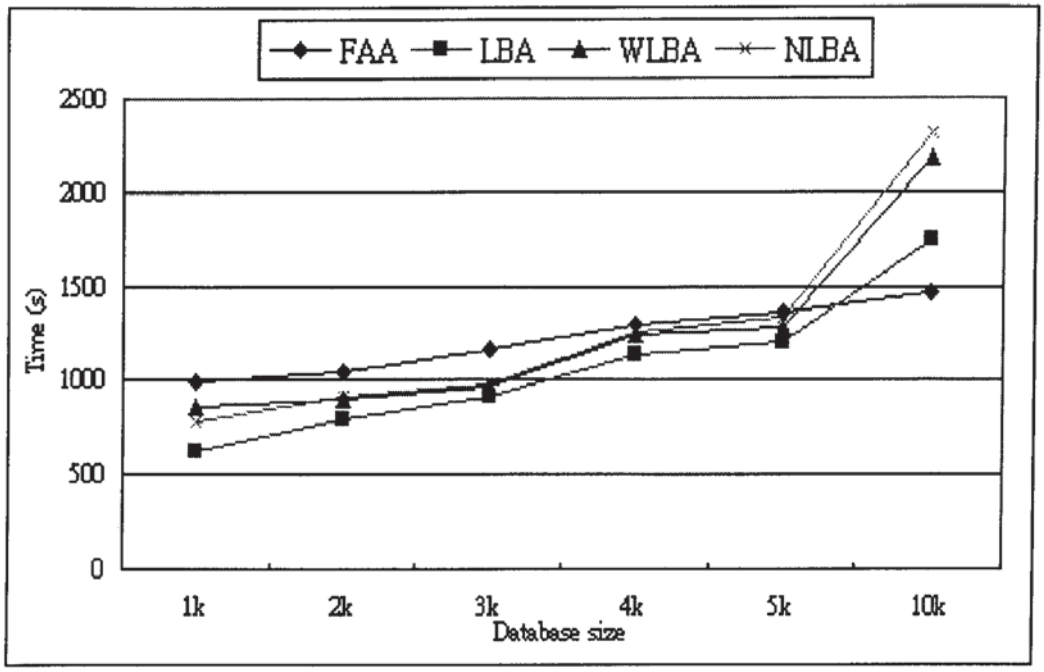

FIGURE 25. COMPARISON OF THE MINING TIMES: FOUR ALGORITHMS WITH DATABASES OF DIFFERENT SIZES AND A SUPPORT COUNT 0.02

Our preliminary analysis reveals that the phenomenon in d) above is due to the operation procedures in the experiments; that is, how agents are spawned. For example, in the experiments the three new algorithms, namely, LBA, WLBA and NLBA started with a single agent. More agents were spawned in a scalable and adaptive manner to maintain a reasonable CTC ratio through the processes of split and aggregate. If one starts these experiments with more than one agent, then the outcome could be different depending on the network traffic pattern at the time. The outcome, however, suggests that it may be a good strategy to invoke the three different algorithms one at a time in an adaptive manner as the instantaneous condition requires. Such adaptive invocations could yield more optimal CTC ratios and lead to better performance, and the investigation of the pros and cons of this approach is planned for the future work.

At this stage of the thesis work, the conceptual S^2AF is proposed from the insight gained from previous investigations. Three new algorithms have been proposed, namely, LBA, WLBA and NLBA to demonstrate how the S^2AF for mining association rules over a network can be realized. The essence of scalability includes two adaptive mechanisms, namely, a) replicating/purging mobile agents, and b) splitting/aggregating database modules. The overall approach can be simply referred to as the scalable S&A strategy. The three new algorithms depend on different parameters to achieve the S^2AF goal effectively. These algorithms were tested thoroughly over the in-house Aglets mobile agent platform, with databases artificially generated by the IBM data generation package. From the test results, it can be concluded that the new algorithms are indeed effective in producing speedup for distributed mining, under different operating conditions involving computation complexity, data complexity, and communication delay. The preliminary experience indicates that S^2AF concept is a right direction for controlling and maintaining reasonable CTC to gain high performance for mobile agent based distributed data mining over a sizeable network such as the Internet. The next immediate step in the research is to apply the new algorithms to some real problems to demonstrate that the S^2AF concept can indeed deal with real situations effectively.

6.2 Experiments using Multimedia Databases

The aim of the demonstration of the multimedia databases is to verify that S^2AF concept is indeed applicable to real-life problems. For this purpose the four algorithms, namely, LBA, WLBA, NLBA and OWLBA were used to mine a multimedia database, with the focus being image data. The features from each image, in this case, the tongue, are extracted and represented in a descriptor, which has a

format similar to a transaction. That is, each feature is considered an "item" in a "transaction". The basic setup for the life application is similar to the previous verification exercise, except real data are used instead of the artificial data generated by the IBM data generation package [76].

A very large tongue image database is used for the experiment. The seventy-five features of the images are extracted to generate a table of descriptors (known as the multimedia database in the research). The mined association rules would relate these features in a predefined fashion, namely, a) shape of the tongue, b) spots on the tongue, c) texture and its position on the tongue, and d) color of the tongue. Part of the table that forms the image database is shown in table 5:

TABLE 5. THE IMAGES' FEATURES ARE EXTRACTED INTO A TABLE FOR MINING

	Shape 1	**Shape 2**	**...**	**Color 1-10**	**Color 11-20**	**...**	**Texture 1**	**Texture 2**	**...**
Image 1	1	0	...	1	0	...	1	1	...
Image 2	0	1	...	1	0	...	1	0	...
Image 3	1	0	...	0	1	...	0	0	...
...									

The feature extraction mechanism, which involves Sobel filters and Hough Transform, is not within the scope of this thesis. The feature extractor prototype is already made available by another project. In this thesis the prototype extractor is used to extract the features, which are then built into descriptors as part of the thesis work. Therefore, the weaving of features into multimedia database of descriptors or transactions, as illustrated by Table 5, is a contribution of this thesis.

The performance comparison of the three algorithms with multimedia databases of

different sizes is shown in Figure 26. This result shows that for the real-life multimedia data mining, the WLBA has the best performance.

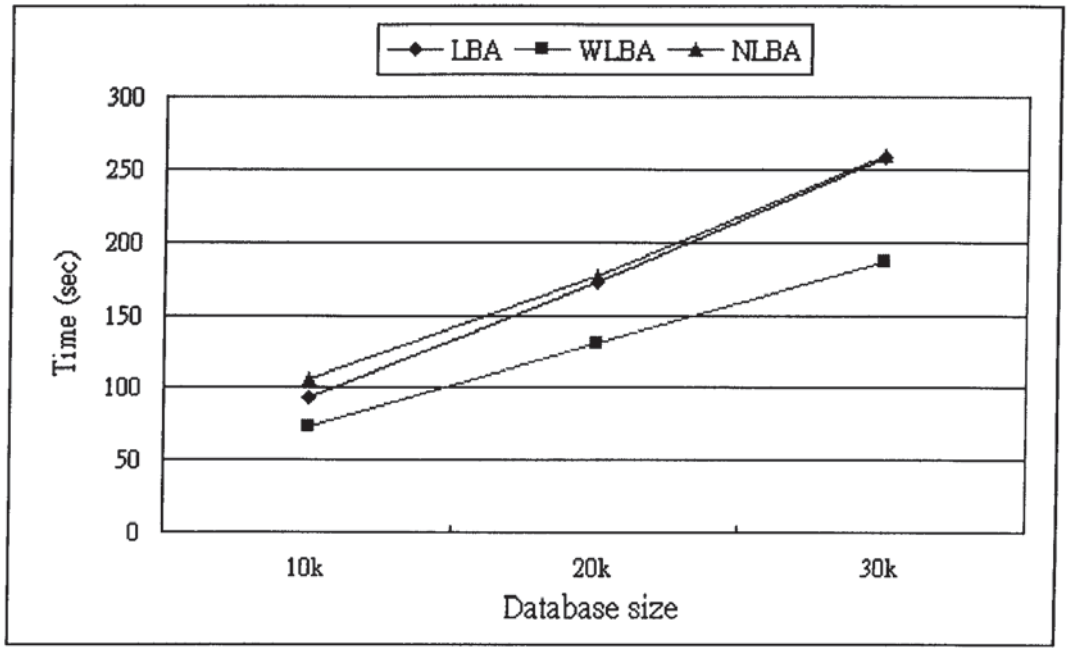

FIGURE 26. COMPARISON OF THE MINING TIMES: THREE ALGORITHMS WITH DATABASES OF DIFFERENT SIZES AND A SUPPORT COUNT 0.1

For the new OWLBA, more experiments were carried out and one set of results is presented in Figure 27, which shows that the OWLBA is indeed more effective than the other three algorithms for mining the provided multimedia databases. In these experiments, the following are imposed for simplicity: a) cloning can produce only 2, 4 or 8 agents, and b) the merging process should also involve only with 2, 4 or 8 agents.

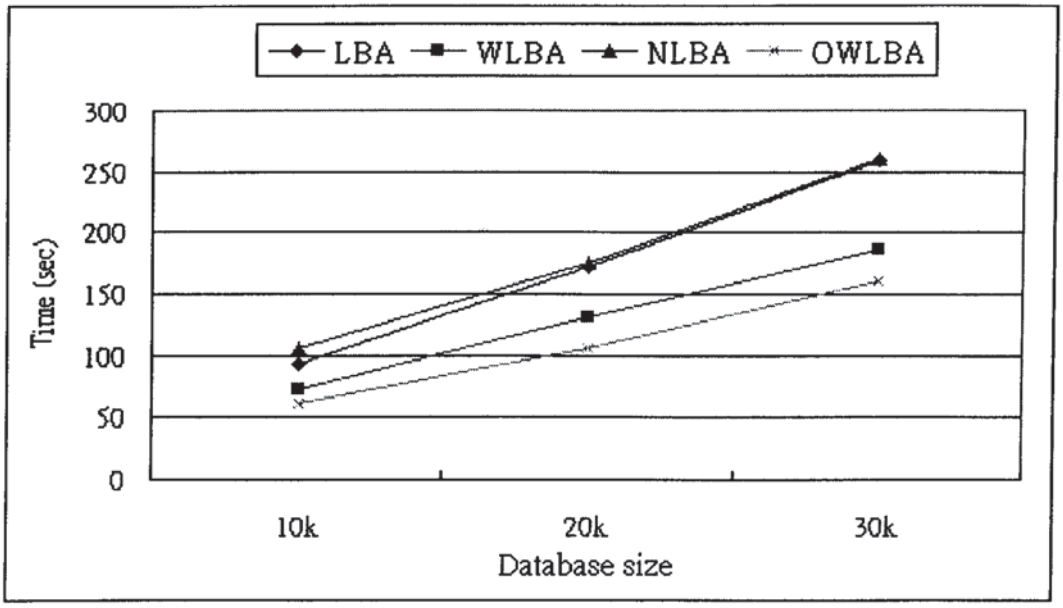

FIGURE 27. COMPARISON OF THE MINING TIMES: FOUR ALGORITHMS WITH DATABASES OF DIFFERENT SIZES AND A SUPPORT COUNT 0.1

6.3 Comparsion with the Count Distribution Algorithm

In addition, more experiments were carried out in order to compare the efficiently of the S^2AF framework and the other algorithms. A well-known algorithm, namely, Count Distribution algorithm is chosen for comparison. Both the database that was generated by the IBM artificial data generation package and the image database are used for the experiments. The experiment results are shown in the following figures. In Figure 28, experiments are carried out with different sizes of the multimedia database. Figure 29 shows the ratios of the response time of the same experiments. From the figure, it shows that the performance of the OWLBA is much better than the Count Distribution Algorithm.

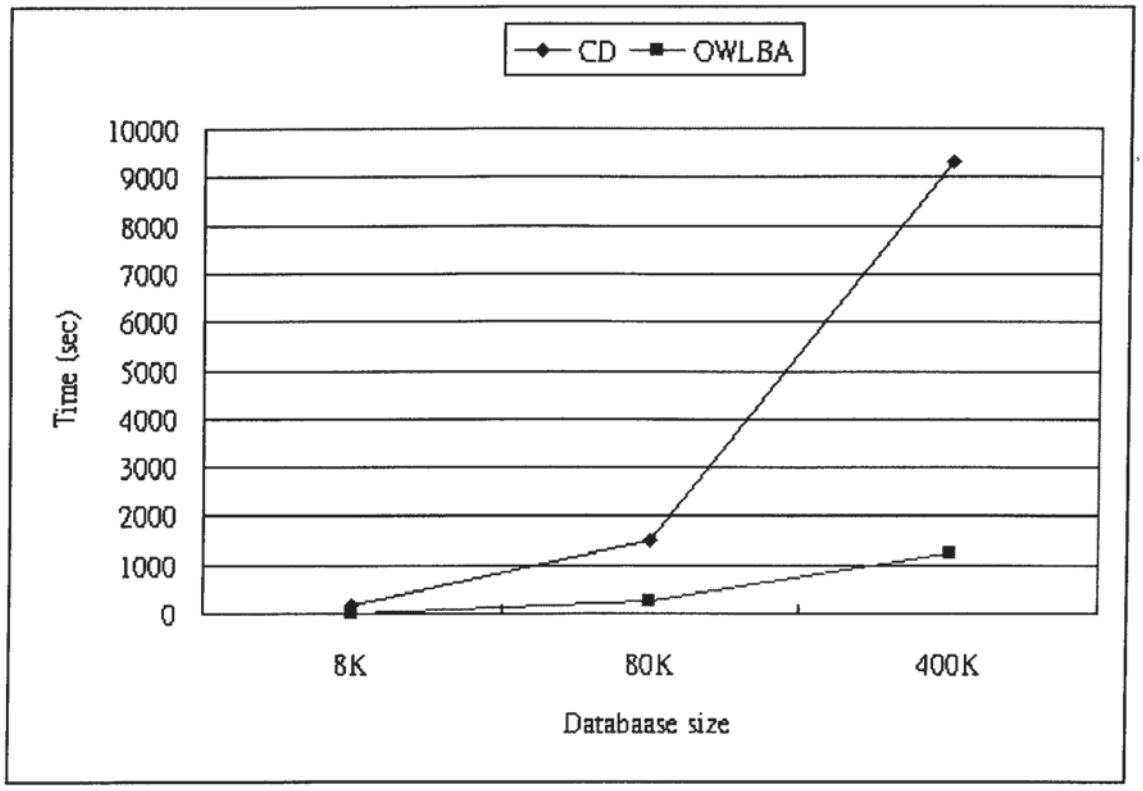

FIGURE 28. COMPARISON OF THE MINING TIMES WITH DIFFERENT DATABASE SIZES (IMAGE DATABASES): CD (COUNT DISTRIBUTION), OWLBA WITH SUPPORT COUNT 0.1 (T8.N75)

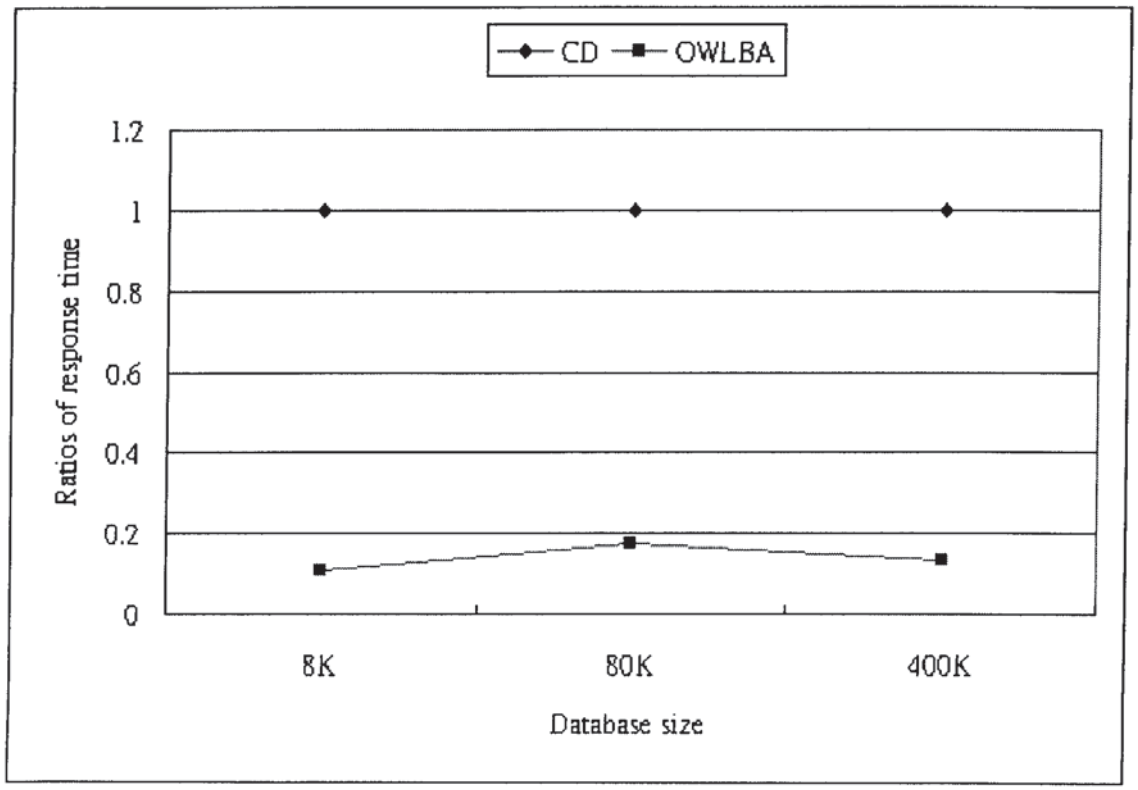

FIGURE 29. COMPARISON OF THE RATIOS OF RESPONSE TIME WITH DIFFERENT DATABASE SIZES (IMAGE DATABASES): CD, OWLBA WITH SUPPORT COUNT 0.1 (T8.N75)

In Figure 30 and Figure 31, experiments are carried out with databases of different transaction length and databases of different number of items. The aim is to study the effect of different parameters on the performance of the algorithms. However, due to the need to generate of databases of different parameters, only the IBM artificial data generation package is used. According to the results, the performance of the OWLBA is still better than the Count Distribution algorithm in the mining of databases of different parameters.

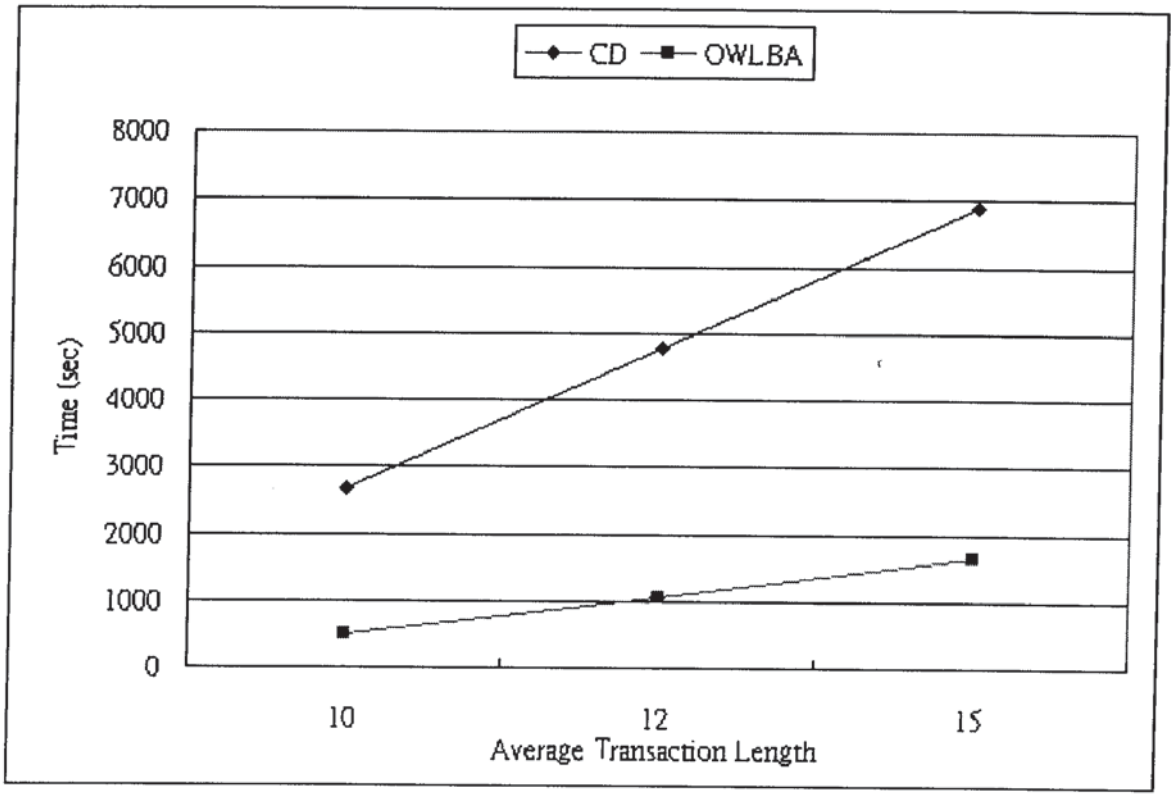

FIGURE 30. COMPARISON OF THE MINING TIMES WITH DIFFERENT AVERAGE TRANSACTION LENGTH (IBM DATA GENERATION PACKAGE): CD, OWLBA WITH SUPPORT COUNT 0.02 D80K.N100)

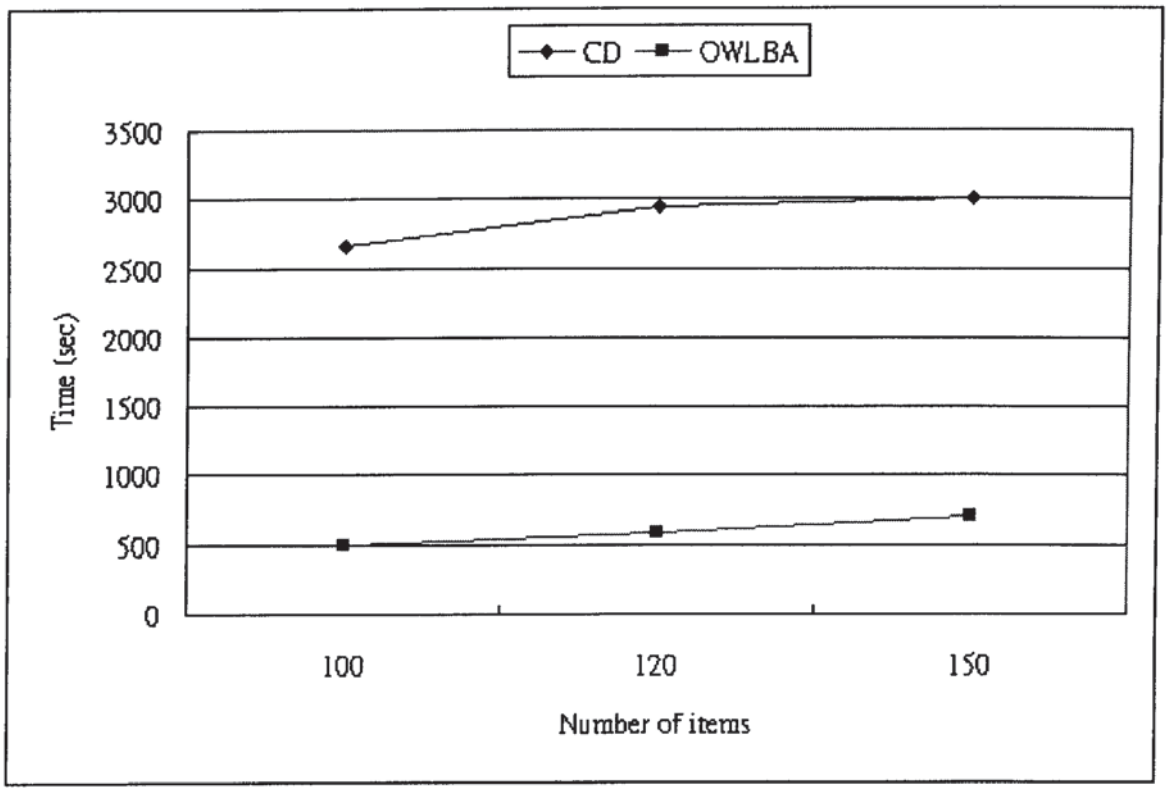

FIGURE 31. COMPARISON OF THE MINING TIMES WITH DIFFERENT NUMBER OF ITEMS (IBM DATA GENERATION PACKAGE): CD, OWLBA WITH SUPPORT COUNT 0.02 D80K.N100)

6.4 Connective Discussion

In mobile agent based distributed data mining of association rules, many factors would affect the performance in a random manner. If one looks at the master/slave or MS model as an example, the service roundtrip time or *RTT* for each round of barrier synchronization is made up of two parts. The first part is the *ComT* (the time that represents the communication cost, excluding the queuing and service processing time in the local host) and the second part is the *CT* (the processing time needed by the slave, including queuing/waiting time and the service execution time). The actual length of *ComT* depends on how many retransmissions are needed to get a client's service request across the communication channel successfully. The number of retransmissions, however, depends on the collective channel error probability δ, which varies independently. The *CT* depends on the seasonal

workload of the host where the slave is located. The random nature of *ComT* and *CT* would affect β (degree of overlapped parallelism among the cognate collaborating mobile agents) and the CTC ratio in a combinatorial manner. This research does not address the exact or formalized relationship between β and the CTC ratio. In fact, the formalization of this relationship is the scope of another on-going work of another project in the department [18]. The empirical experience in this thesis shows that whenever β changes the CTC would change as well. Our preliminary analysis indicates that this is inevitable because β affects *ComT*, which affects the CTC ratio. The *CT* is construed as consisting of three parts, namely, T_{WT} (queuing/waiting time), T_{MT} (actual mining/service time, which is also known as T_c or T_p in different S^2AF algorithms for simplicity), and T_{CS} (context switching time). In this thesis, the CTC ratio is defined as $CT/ComT$. If T_{CS} is very high, it may be better off to move (migrate) the slave and its data block (to be mined) to a less busy host, provided that the performance (speedup) benefit from the migration is more than the cost to do it. The cost functions for such migrations are outside the scope of the present work. If T_{MT} is too high, then it may be worthwhile to improve the mining speedup through parallelism, and this may be achieved by replicating the miner and splitting the original data block into smaller modules. The *RTT*, which is *CT* + *ComT*, is always affected by the dynamic variations in the T_{WT}, T_{MT}, and T_{CS} parameters. Since it is not easy and also impractical to measure the individual values for these dynamic parameters, a practical approach may be to estimate the overall dynamic changes by making use of the *RTT*, which includes the effect of *CT* and *ComT*. In fact, measuring the *RTT* is a commonly adopted approach in different performance measurement domains, for example in the area of Internet End-to-End Performance Measurement, because some stable tools for accurate measurement of *RTT* are

available. In the future work, how these tools can be adopted will be investigated.

Although replication of the miner and partitioning of the original data block would bring shorter service *RTT*, it would not go on forever. Once the operation is outside the optimal region (Region 2 into Region 3), the CTC ratio would first decrease gradually and then wane, provided that *ComT* has remained relatively stable. The same applies to the shortening of the service *RTT*; the asymptotic decrease in *CT* is the reason. At this stage parallelization would decrease the CTC even further because more agents means more communication overhead. The solution is to reverse the trend of low CTC by aggregation. The physical meaning for Region 3 is that *ComT* is now the dominant factor. Reducing an excessively long T_{MT} by replicating a miner and partitioning its original database or data block is fundamental to S&A operations and the core of the S^2AF concept. Any effective S^2AF algorithm, however, should be able to detect and operate within the optimal region for the CTC ratio. In reality, the optimal region may shift and a successful S^2AF algorithm would know when to split further or aggregate to keep its operation within the region. From the different experiments and tests, it can be concluded that the predicate and the thresholds that determine when to split or aggregate would decide the efficacy of the algorithm. The data from these experiments though has not taken the actual effect of *ComT* into serious consideration because it is relatively stable in the dedicated testing environment. In order to explore this, four new S^2AF algorithms were proposed, namely, LBA, WLBA, NLBA and OWLBA. For the large multimedia data blocks, our empirical results indicate that the OWLBA is always the most efficient.

The contribution by the timing elements, namely, *ComT*, T_{WT}, T_{CS} and T_{MT} to a S^2AF algorithm is a random phenomenon. Therefore the determination of the optimal CTC ratio should be an on-the-fly operation. The principle is that if the S^2AF algorithm finds that a further split would yield little decrease in the service *RTT*, the split operation should stop. If the CTC ratio keeps on decreasing, but the service *RTT* remains consistently constant, then it may be the appropriate time to start the aggregation process to bring the operation back to Region 2 (Figure 15 and Figure 20). The reason for a relatively constant service *RTT* is that the dominance of *ComT* renders *CT* relatively insignificant because of the small data size. In our experiments, the workload of all the dedicated hosts in the controlled Intranet environment is light (very low T_{CS} and T_{WT}), with little fluctuation. Besides, the number of retransmissions is so low that it can be considered negligible. Therefore, the service *RTT*, which is more sensitive to the *CT* in this case, is a good reflection of the CTC ratio. In the uncontrolled and more open Internet environment, the service *RTT* could be seriously affected by different dynamic factors. The effect of such network openness will be left for more in-depth investigations in the future.

Chapter 7

Conclusion and Future Work

The aim of this project is to propose a scalable split & aggregate framework, namely, the S^2AF, for efficient mobile agent based mining of association rules over the Internet. With the help of the IEP methodology, all the objectives for the project have been achieved satisfactorily and the findings are published in seven papers, including one refereed journal and six refereed conference papers. The findings from the early investigations have provided the basis for the formulation and consolidation of the S^2AF concept. The earlier experiments were performed on the stable Aglets mobile agent platform running in the controlled Intranet environment in our laboratory. The data for all these experiments and tests were synthesized by the IBM data generation package, which is popular among many data-mining research groups. The previous experience in another research in the department has also concluded that this data generation package is stable and credible. The four S^2AF algorithms proposed in the project, namely, LBA, WLBA, NLBA and OWLBA worked effectively in the controlled environment. In the tests, the effect by the T_{CS} and T_{WT} is minimal because all the hosts to support the MS mining paradigm were dedicated machines. This makes the service *RTT* in every round of barrier synchronization reflective of the CTC ratio. The preliminary empirical results indicate that OWLBA has the best performance when the database is large and complex. In the later part of the research the S^2AF verified with artificial data was actually applied to a real-life problem, namely, distributed mining of associations in

the multimedia data, which consists only of images. The goal is to demonstrate that the S^2AF would work equally well for real-life problems over the Internet.

Although the findings from the project have concluded that the S^2AF is a sound approach for mining associations over a sizeable network with mobile agents, before the framework can be used to resolve real-life problems effectively and efficiently in a scalable manner the following have to be investigated:

a) *The effect of T_{CS} and T_{WT}*: The effect of these elements must be understood in order to design high-performing S^2AF algorithms because this effect can change dramatically in a sizeable uncontrolled environment such as the Internet.

b) *Vigorous tests in an uncontrolled environment*: The S^2AF must be tested and validated in uncontrolled environments, ideally different clusters "annexed" from the Internet. Only the volatility of these environments can unveil the limitation of this framework.

c) *Test with different real data*: It is a well-known fact that the performance of parallel operation depends on both the complexity of the algorithm and the complexity of the data as well. While the complexity of an algorithm can usually be defined with relative ease, the complexity of data in a database can be hard to determine. By testing out the S^2AF with different data types, the adaptiveness of an S^2AF algorithm can be identified. For example, with these tests, one may be able to answer questions such as "*Is it effective to change the thresholds adaptively with respect to the data complexity in an on-the-fly manner?*"

d) The four new S^2AF algorithms, namely, LBA, WLBA, NLBA and OWLBA, have not addressed the issue of when the split action that reduces the CTC ratio should stop. There is no mechanism to detect whether the optimal region

(Region 2) has been reached. Rather, these algorithms rely on the fact the once Region 3 is reached, the aggregation process would bring it back to Region 2. This means that such back-and-forth region shifts might bring out some undesirable operational oscillations. It is important to include suitable detection mechanisms to eliminate these oscillations for better and smoother performance.

e) The experiments with the new algorithms are based on the *CT* only, and this means that the effect of *ComT* has not been considered. It is necessary in the future to investigate this aspect because the dynamic *ComT* can seriously affect performance, especially when the parallel responses from collaborating agents do not overlap well (Figure 9, Figure 21).

One important contribution by the S^2AF is flexibility because the framework not only allows but also requires the development of different S^2AF algorithms for different problems and application domains. Therefore, the future work should involve formulation and development of different efficacious S^2AF algorithms.

Appendix A

Achievement and Publication

All the objectives for this research, which are stated in the Problem Statement section, have been achieved successfully. The results from the research have been published in one refereed journal paper and six refereed conference papers. The list of these papers is as follows:

1 Refereed Journal (published):

[p1] Allan K.Y. Wong, S.L. Wu and Ling Feng, **An Efficient Algorithm for Mining Association Rules for Large Itemsets in Large Databases: From Sequential to Parallel**, International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, Special Issue: Data Mining, 8(2), June 2000, 109-118

6 Refereed Conference (published):

[p2] S.L. Wu, Allan K.Y. Wong, K.W. Hung, W.N. Leung, and Sam K.F. Ho, **MEIN: A Model for Effective Mining of Cross-Object Relationships in Distributed Databases over a Large Network Exemplified by the Internet**, *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2001)*, vol. 2, Las Vegas, USA, June 2001, 577-582

[p3] S. L. Wu, V. Cho, Allan K. Y. Wong and Tharam S. Dillon, **Three Load Balancing Algorithms for Improving Performance of Mining Association Rules**

from Very Large Databases with Agents over the Internet, *Proc. of the International Conference on Internet Computing (IC2001)*, vol. 2, Las Vegas, USA, June 2001, 723-729

[p4] S.L. Wu, Allan K.Y. Wong and T.S. Dillon, **An Aggregation Algorithm for Mining Association Rules from Very Large Databases with Mobile Agents**, *Proc. of the International ICS Symposium on Multi-Agents and Mobile Agents in Virtual Organization and E-Commerce (MAMA2000),* Wollongong, Australia, December 2000, 589-594

[p5] Allan K.Y. Wong and Richard S.L. Wu, **5E: A Framework to Yield High Performance in Real-time Data Mining over the Internet,** *Proc. HPCAsia2000 Conference,* Beijing, P.R. China, May 2000, 708-713

[p6] Allan K.Y. Wong, Anthony K.M. Lam and Richard S.L. Wu, **The M^2D^2 Framework: A Combination of Multi-language Programming, Mobile Objects, Data Interoperability and Dynamic Compilation for Effective Real-time Computing on the Internet,** *Proc. HPCAsia2000 Conference,* Beijing, P.R. China, May 2000, 451-456

[p7] Allan K.Y. Wong, S.L. Wu and L. Feng, **An Efficient Algorithm for Mining Association Rules for Large Itemsets in Large Centralized Databases**, *Proc. of the IEEE SMC'99*, Tokyo, Japan, 905-910

The papers, namely, p5, p6 and p7 were experience from the first phase (Refer to the "Choice of Research Methodology" in the next section) of the research, and p1, p2, p3 and p4 represent the experience in the second phase. In particular p2 and p3 contain the research results obtained in the final stage of the research.

Appendix B

Choice of Research Methodology

There are many methodologies for research in general and they cover different problem domains. In the area of computing research, three basic types of research can be identified, namely:

a) *Exploratory research*: This type tackles a little known problem or topic and it is typical that the research idea cannot be formulated very well at the beginning. The result would push out the knowledge frontiers or lead to discovery of new knowledge.

b) *Testing-out research*: This type is to find the limits of previous generalizations.

c) *Problem-solving research*: This type usually starts with a specific real-world problem and then brings all the available intellectual resources together for its solution.

It is important to pick the right methodology for a particular type of research to facilitate success. For example, one may adopt one or the following basic methodologies:

a) *Top Down*: First the objectives are defined and then they are realized step by step. The typical examples include: 1) the Waterfall model in software engineering work, which does not strongly encourage user intervention, and 2) the Fast Prototyping approach that encourages repetitive user input until the system is finally accepted.

b) *Bottom Up*: Usually a coordination model is proposed so that what is available (commodities and/or intellectual resources) can be interconnected into a single system. The typical example is Linda, which is a well-known coordination language.

The Top Down approach is seemingly suitable for the testing-out type of research, and likewise the Bottom Up approach is more natural for the problem-solving type. By nature this research is exploratory even though it would produce a prototype for testing and supporting further research at the end. It is somewhat top-down because the course of research would include literature search, problem statement, proposed solutions, and data collection. It is however difficult to apply the Top Down approach in a strict sense because many early exploratory investigations are necessary and these investigations would involve repetitive backtracking and cross-referencing to gain the necessary insight for the next step. Therefore, there is a need to devise a more original methodology to meet the research needs, namely, the "*investigate & experiment & proceed with possible backtracking, cross referencing and looping* (IEP)" approach. Since data mining in a distributed environment is a relatively new area, previous techniques are usually experimental and limited in scope. In order to gain insight into what is suitable for supporting the S^2AF, intermediary tests and experiments have to be carried out to shed light on the pros and cons of different methods, for possible adoption, modification and comparison. The outcome may also include proposals for new methods.

Research Road Map

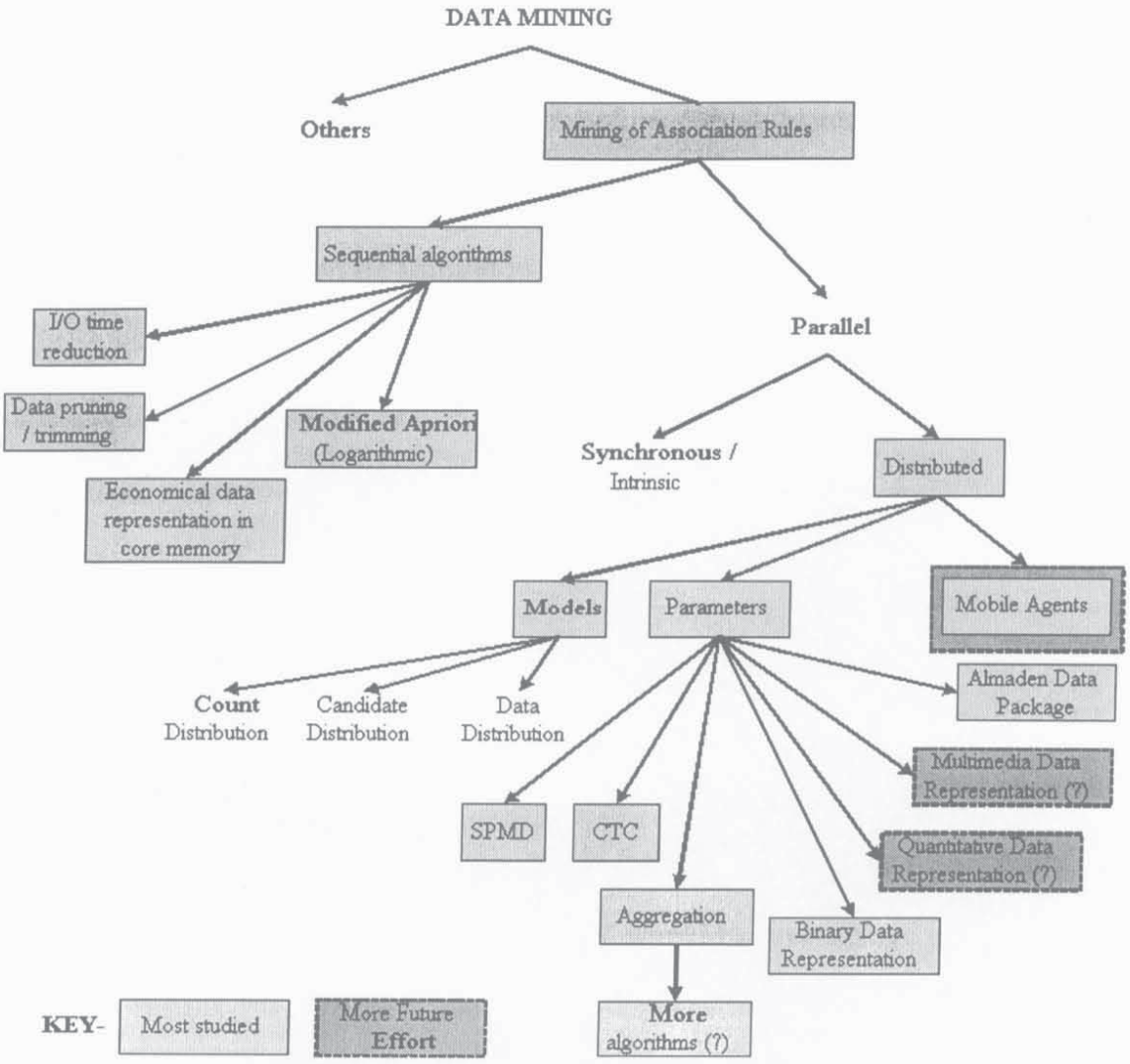

FIGURE 32. THE ROADMAP FOR PROJECT MANAGEMENT

After careful evaluation of the empirical experience gained from the different intermediary tests and experiments, sometimes carried out in a repeated manner, it can be concluded that IEP is indeed more suitable for the present project. A typical top-down approach consists mainly of sequential steps to be carried out in a straight-down hierarchical fashion. For example, it may consist of the following steps: literature search, proposed framework, experiments and data collection (including data analysis), and then conclusion. The main difference between the IEP and a strict top-down approach is that in the IEP approach explorations and

investigations start from the root through the branches to a leaf and then back again. The traversals up and down the branches and leaves represent a heuristic process, and these traversals may repeat many times before enough material, data and insight can be consolidated for the next stage of the S^2AF research. In fact, the high-level view of IEP approach can be illustrated by the roadmap shown in Figure 32. This roadmap identifies what should be explored and achieved for completing the thesis. An example of IEP traversal may be the following path: *Data Mining* → *Mining of association rules* → *Sequential algorithms*→ *I/O reduction*→ *Sequential algorithms*→ *Mining of association rules*→ *Parallel*→ *Distributed*→ *Parameters*→ *CTC*→ *Parameters*→ *Aggregation*. This path represents one of the many possible "*operation*" paths in the course of the project because traversals back and forth are necessary for cross-reference, data refinement and/or comparison. In the research plan those items that should be investigated in the first phase of the research are in "*solid-line boxes*" and those "*dotted-line boxes*" would be investigated in the second phase. The research is separated into two phases because it is anticipated that the results from the first phase would shape the direction in the second phase.

In the final stage of the thesis work, trials would be carried out to apply the S^2AF concept to actual mining of multimedia data (mainly images) in a controlled Intranet environment. The goal is to show that the S^2AF can indeed work with real applications and certainly this is only a preliminary demonstration. In fact, more serious applications of the S^2AF to distributed mining of large volumes of media data would necessitate more detailed investigations, and due to time constraints, this is planned for the future work.

The possible roadmap traversals include the following sequences (with backtracking) and purposes, namely:

a) Understanding the rationale of mining of association rules in general,
b) Studying some sequential mining approaches and algorithms,
c) Studying different distributed mining approaches and algorithms (comparing them to sequential approaches as well),
d) Gathering enough information for consolidating the S^2AF concept,
e) Looking for a stable mobile-agent platform for testing purposes,
f) Refining the S^2AF prototype for better data collection and analysis, and
g) Demonstrating how the S^2AF can be applied to some real problems in a small scale.

Bibliography

[1] P.W. Adriaans, Predicting Pilot Bid Behavior with Genetic Algorithms. *Proc. of the Sixth International Conference on Human-Computer Interactions, Symbiosis of Human and Artifact*, (Y. Anzai, K. Ogaw and H. Mori, eds), Tokyo, 1995

[2] Pieter Adriaans and Dolf Zantinge, *Data Mining*, Addison-Wesley, July 1996, 8–9

[3] R. Agrawal, T. Imielinski, and A. Swami, Mining Associations between Sets of Items in Massive Databases, *Proc. of the ACM SIGMOD Int'l Conference on Management of Data*, Washington D.C., May 1993, 207-216.

[4] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. Verkamo, Fast discovery of association rules. *In Advances in Knowledge Discovery and Data Mining,* Cambridge MA: AAAI Press/MIT Press, 1996

[5] R. Agrawal and J.C. Shafer, Parallel Mining of Association Rules, *IEEE Transactions on Knowledge and Data Engineering*, 8(6), December 1996

[6] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules, *Proc. of the 20th Conference on Very Large Databases*, Santiago, Chile, September 1994

[7] I. Bhandari et al, Advanced Scout: Data Mining and Knowledge Discovery in NBA Data, *Data Mining and Knowledge Discovery*, 1(1), 1997, 121-125

[8] L.B. Booker, D.E. Goldberg and J.H. Holland, *Classifier Systems and Genetic Algorithms*, 1989

[9] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, Dynamic Itemset Counting and Implication Rules for Market Basket Data, *Proc. of ACM SIGMOD International*

Conference on Management of Data, 1997, 283-264

[10] J. Carbonell, *Machine Learning, Paradigms and Methods*, Cambridge MA: MIT Press, 1990

[11] M.S. Chen, S.P. Jong and P.S. Yu, Data Mining for Path Traversal Patterns in a Web Environment, *Proc. of the 16th International Conference on Distributed Computing Systems*, May 1996, Hong Kong, 385-392

[12] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris, and G. Tsudik, Itinerant Agents for Mobile Computing, IEEE Personal Communications Magazine, 2(5), October 1995, 34-49

[13] M. Chester, *Neural Networks: A Tutorial*, Englewood Cliffs NJ: Prentice-Hall, 1993

[14] D.W. Cheung, J. Han, V. Ng, A.W. Fu and Y. Fu, A Fast Distributed Algorithm for Mining Association Rules. *Proc. Fourth International Conference on Parallel and Distributed Information System (PDIS-96)*, Miami, Florida, December, 1996, 31-43

[15] D.W. Cheung, K. Hu and S. Xia, An Adaptive Algorithm for Mining Association Rules on Shared-memory Multi-processors Parallel Machine, *Distributed and Parallel Databases*, Kluwer Academic Publishers, March 2001, 99-132

[16] D.W. Cheung, K. Hu and S. Xia, Asynchronous Parallel Algorithm for Mining Association Rules on a Shared-memory Multi-processors, *Proc. of the Tenth Annual ACM Symposium on Parallel Algorithms And Architectures (SPAA-98)*, Puerto Vallarta, Mexico, June 1998

[17] D.W. Cheung, V.T. Ng, A.W. Fu, Yongjian Fu, Efficient Mining of Association Rules in Distributed Databases, *IEEE Transactions on Knowledge and Data*

Engineering, 8(6), 1996, 911 - 922

[18] C.W. Chiu, Utilizing Average Roundtrip Time (RTT) and the Local Workload ρ for Efficient Object-Oriented Real-time Computing over the Internet, *MScST Thesis, Department of Computing, Hong Kong PolyU*, 2000/2001

[19] P. Clark and T. Niblett, The CN2 induction algorithm, *Machine Learning*, 3, 1996, 261-83

[20] W.T. Cockayne and M. Zyda, *Mobile Agents*, Manning Publication, 1997

[21] M. Crovella, R. Bianchini, T. LeBlanc and E. Markatos, Using Communication-to-Computation Ratio in Parallel Program Design and Performance Prediction, *Proc. of the 4th IEEE Symposium on Parallel Distributed Processing*, 1992, 238 – 245

[22] A. Czyzewski, Mining Knowledge in Noisy Audio Data, *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining*, 1996, 220-225

[23] T.S. Dillon, T. Hossain, W. Bloomer, and M Witten, Improvements in Supervised BRAINNE: A Method for Symbolic Data Mining Using Neural Networks. In S. Spaccapietra and F. Maryanski, eds. *Data Mining and Reverse Engineering, IFIP 7th Conference on Database Semantics (DS-7),* 7-10 Oct. 1997, Leysin, Switzerland, London, UK, Chapman & Hall, 1998, 67-88.

[24] U.M. Fayyad, G. Piatesky-Shapiro, P. Smyth and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, Cambridge MA: AAAI Press/MIT Press, 1996

[25] Jianlin Feng and Yucai Feng, Binary Partition Based Algorithms for Mining Association Rules, *ADL 98. Proc. IEEE International Forum on Research and Technology Advances in Digital Libraries,* 1998, 30 –34

[26] L. Feng, H. Lu and Allan K.Y. Wong, A Study of Database Buffer Management

Approaches: Toward the Development of a Data Mining Based Strategy, *IEEE SMC'98 Proc.*, San Diego, California, USA, 1998, 2715-2719

[27] L. Feng, H. Lu, J. Yu, and J. Han, Exploiting Templates to Make Multi-Dimensional Inter-Transaction Association Rules Mining Practical, *Proc. 1999 Int. Conf. on Information and Knowledge Management (CIKM'99)*, Kansas City, Missouri, Nov. 1999

[28] S. Franklin and A. Graesser, "Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents." *Proc. of 3rd International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996

[29] M.R. Genesereth and S. P. Ketchpel, Software Agents, *Communications of the ACM*, 37(7), July 1994 48-53

[30] W.I. Grosky and Yi Tao, Multimedia Data Mining and Its Implications for Query Processing. *IEEE 9th International Workshop on Database and Expert Systems Applications, Vienna*, Austria, August 26-28, 1998, 95-100

[31] E.H. Han, G. Karypis and V. Kumar, Scalable Parallel Data Mining for Association Rules, *Proc. of ACM SIGMOD International Conference on Management of data*, 1997, 277-288

[32] Jiawei Han, Jian Pei and Yiwen Yin, Mining Frequent Patterns Without Candidate Generation, *Proc. of ACM SIGMOD on Management of data*, 2000, 1–12

[33] C.G. Harrison, D.M. Chess, and A. Kershenbaum, Mobile Agents: Are They a Good Idea?, *IBM Research Report*, RC 19887, 1994

[34] R. Hecht-Nielsen, *Neurocomputing*, Reading MA: Addison-Wesley, 1991

[35] J. Hertz, *Introduction to the Theory of Neural Computing*, Reading MA: Addison-Wesley, 1991

[36] C.T. Ho, R. Agrawal, N. Megiddo, R. Srikant, Range Queries in OLAP Data

Cubes, *Proc. of the ACM SIGMOD Conference on Management of Data*, Arizona, May 1997

[37] C.T. Ho, R. Agrawal, N. Megiddo, J.J. Tsay, Techniques for Speeding up Range-Max Queries in OLAP Data Cubes, *IBM Research Report*, April 1997

[38] M. Holsheimer and M. Kersten, Architectural Support for Data Mining. *Technical report CWI*, 1994

[39] P. Hui, Intelligent Collaborative Data Mining (I-CAN) with ANN-based Intelligent Agents over the Internet, technical report, Department of Computing, Hong Kong Polytechnic University, May 2000

[40] S.P. Jong, Using a Hash-Based Method with Transaction Trimming for Mining Association Rules, *IEEE Transaction on Knowledge and Data Engineering*, vol. 9, 1998.

[41] G. Karjoth, D. B. Lange, and M. Oshima, A Security Model for Aglets, *IEEE Internet Computing*, 1(4), 1997, 68-77

[42] R. Khosla and T.S. Dillon, *Engineering Intelligent Hybrid Multi-Agent Systems*, Kluwer Academic Publishers, Massachusetts, USA, 1997

[43] Y. Kodratoff, *Introduction to Machine Learning*, London: Pitman, 1988

[44] Y. Kodratoff, G. Nakhaeizadeh and C. Taylor, Statistics, Machine Learning and Knowledge Discovery in Databases, *Mlnet Familiarization Workshops,* Heraklion, Greece, 1995

[45] Y. Labrou and T. Finin, A Proposal for a New KQML Specification, *Technical Report CS-97-03*, Computer Science and Electrical Engineering Department, University of Maryland Baltimore Country, Baltimore, 1997

[46] D.B. Lange and Mitsuru Oshima, Mobile Agents with Java: The Aglet API, *World Wide Web Journal*, Baltzer Science Publishers, Bussum, The Netherlands,

1998s

[47] D.B. Lange, M. Oshima, G. Karjoth, and K. Kosaka, Aglets: Programming Mobile Agents in Java, *Proc. of Worldwide Computing and Its Applications (WWWCA'97), Lecture Notes in Computer Science*, Vol.1274, Springer Verlag, New York, 1997

[48] N. Lavrac and S.K. Wrobel, Ed, *Machine Learning: ECML-95*, New York: Springer-Verlag, 1995

[49] T. Lewis, The Next $10{,}000_2$ Years: Part 2, *IEEE Computer*, May 1996, 78-85

[50] S. Li, H. Shen and L. Cheng, New Algorithms for Efficient Mining of Association Rules, *Frontiers of Massively Parallel Computation*, 1999. Frontiers '99, 1999, 234 –241

[51] Jun-Lin Lin and M.H. Dunham, Mining Association Rules: Anti-Skew Algorithms, *Proc. of the 14th International Conference on Data Engineering*, 1998, 486 –493

[52] H. Mannila and K.-J. Raiha. Dependency Inference, *Proc. of the VLDB Conference*, Brighton, England, 1987, 155-158

[53] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, New York: Springer-Verlag, 1994

[54] R. Michalski, On the Quasi-Minimal Solution of the General Covering Problem, *Proc. of the Fifth International Symposium on Information Processing*, 1969, 125-8

[55] C. Panayiotou, G. Samaras, E. Pitoura and P. Evripidou, Parallel Computing Using Java Mobile Agents, Proc. of 25th EUROMICRO Conference, 2, 1999, 430-437

[56] Jong Soo Park, Ming-Syan Chen and P.S. Yu, Using a Hash-Based Method with Transaction Trimming for Mining Association Rules, *IEEE Transactions on*

Knowledge and Data Engineering, 9(5), Sept.-Oct. 1997, 813 –825

[57] G. Piatesky-Shapiro and W. Frawley, *Knowledge Discovery in Databases*, Cambridge MA: AAAI Press/MIT Press, 1991

[58] J. Quinlan, *C4.5: Programs for Machine Learning*, Redwood City CA: Morgan Kaufmann, 1988

[59] T. Shintani and M. Kitsuregawa, Hash Based Parallel Algorithms for Mining Association Rules, *The 4th International Conference on Parallel and Distributed Information Systems*, 1996, 19 –30

[60] L.M. Silva, G. Soares, P. Martins, V. Batista and L. Santos, The Performance of Mobile Agent Platforms, Agent Systems and Applications, *Proc. of the 3rd International Symposium on Mobile Agents*, 1999, 270 -271

[61] E. Simoudis, B. Livezey and R.. Kerber, Using Recon for Data Cleaning, *Proc. of the First International Conference on Knowledge Discovery and Data Mining,* New York: AAAI Press, 1995

[62] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen and H. Mannila, Pruning and Grouping Discovered Association Rules, *Proc. of the Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases*, 1995

[63] Allan K.Y. Wong, Anthony K.M. Lam and Richard S.L. Wu, The M^2D^2 Framework: A Combination of Multi-language Programming, Mobile Objects, Data Interoperability and Dynamic Compilation for Effective Real-time Computing on the Internet, *Proc. of HPCAsia2000 Conference, Beijing*, P.R. China, May 2000, 451-456

[64] Allan K.Y. Wong and Richard S.L. Wu, 5E: A Framework to Yield High Performance in Real-time Data Mining over the Internet, *Proc. HPCAsia2000 Conference, Beijing*, P.R. China, May 2000, 708-713

[65] Allan K.Y. Wong, S.L. Wu and L. Feng, An Efficient Algorithm for Mining Association Rules for Large Itemsets in Large Centralized Databases, *IEEE SMC'99 Proc.,* Tokyo, Japan, 1999, 905-910

[66] Allan K.Y. Wong, S.L. Wu and Ling Feng, An Efficient Algorithm for Mining Association Rules for Large Itemsets in Large Databases: From Sequential to Parallel, *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, Special Issue: Data Mining*, 8(2), June 2000, 109-118

[67] S.L. Wu, Allan K.Y. Wong and T.S. Dillon, An Aggregation Algorithm for Mining Association Rules from Very Large Databases with Mobile Agents, *Proc. of the International ICS Symposium on Multi-Agents and Mobile Agents in Virtual Organization and E-Commerce (MAMA2000),* Wollongong, Australia, December 2000, 589-594

[68] Z. Xu and K. Hwang, MPPs and Clusters for Scalable Computing, *Proc. of I-SPAN'96,* Beijing, P.R. China, June 1996, 117-123

[69] 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), September 1998, 311-322

[70] J. You and P. Bhattacharya, A Wavelet-Based Coarse-to-Fine Image Matching Scheme in a Parallel Virtual Machine Environment, *IEEE Transactions on Image Processing*, 9(9), Sept. 2000, 1547-1559

[71] O.R. Zaiane, J. Han, Z. Li, S. Chee and J. Chiang, MultiMediaMiner: A System Prototype for MultiMedia Data Mining, *Proc. of ACM SIGMOD, International Conference on Management of Data*, 1998, 581-583

[72] O.R. Zaiane, Jiawei Han, Ze-Nian Li, J. Hou, Mining Multimedia Data, *Proc.*

of CASCON'98: Meeting of Minds, Toronto, Canada, November 1998, 83-96

[73] O.R. Zaiane, J. Han, and H. Zhu, Mining recurrent items in multimedia with progressive resolution refinement, *Proc. of the 16th International Conference on Data Engineering*, 2000, 461–470

[74] Mohammed J. Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara and Wei Li, New Algorithms for Fast Discovery of Association Rules, *3rd International Conference on Knowledge Discovery and Data Mining (KDD)*, Newport, California, August, 1997, 283-286

[75] http://www.trl.ibm.co.jp/aglets/

[76] IBM Almaden Research Center, Synthetic Data Generation Code for Association and Sequential Patterns, *http://www.almaden.ibm .com/*, 1998

[77] *Workshop on Multimedia Data Mining*, The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, August, 2000