World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:6, 2008

Parallel and Distributed Mining of Association Rule on Knowledge Grid

U. Sakthi, R. Hemalatha, and R. S. Bhuvaneswaran

***Abstract*—**In Virtual organization, Knowledge Discovery (KD) service contains distributed data resources and computing grid nodes. Computational grid is integrated with data grid to form Knowledge Grid, which implements Apriori algorithm for mining association rule on grid network. This paper describes development of parallel and distributed version of Apriori algorithm on Globus Toolkit using Message Passing Interface extended with Grid Services (MPICH-G2). The creation of Knowledge Grid on top of data and computational grid is to support decision making in real time applications. In this paper, the case study describes design and implementation of local and global mining of frequent item sets. The experiments were conducted on different configurations of grid network and computation time was recorded for each operation. We analyzed our result with various grid configurations and it shows speedup of computation time is almost superlinear.

***Keywords*—**Association rule, Grid computing, Knowledge grid, Mobility prediction.

I. Introduction

MANY organizations have multiple databases distributed over different branches. One of the main issue to be faced by next generation internet is how to manage and analysis those data. The knowledge gird environment provides effective way to store, acquire, represent exchange of information and converts them into useful knowledge through data mining. Distributed data mining is widely used to analyze large data sets maintained over geographically distributed sites. Globus toolkit is a set of software components for building distributed systems.

An association rule is a rule which implies certain association between set of objects (such as “occur together” or “one implies the other”) in a database. Mining association rule in database generates useful patterns for decision support system, knowledge discovery, mobility prediction and many other real time applications. Apriori is the most frequently used algorithm for mining association rules. Database or data warehouse stores large amount of transaction records that are located in different locations. The overhead in integrating the data sources will be too high. Mining association rule in a single large database require more processing power [2]. Due to property of distributed environments, conventional technology used for centralized data mining is no longer suitable for new systems. Grid service implements apriori algorithm in parallel and distributed manner.

Apriori algorithm is executed on multiple grid nodes in parallel. Distributed data mining algorithms optimize the exchange of data needed to develop global knowledge models based on concurrent mining of remote data sets. Fast Distributed Mining (FDM) Algorithm is implemented on grid network. In each grid node, FDM finds the local support counts and prunes all infrequent item sets. After completing local pruning, each grid node broadcasts messages containing all the remaining candidate sets to all other grid nodes to request their support counts. It then decides whether large itemsets are globally frequent and generates the candidate itemsets from those globally frequent itemsets. The FDM communication complexity is O(|Cp| * n), where |Cp| and n are large itemsets and the number of sites.

MPICH-G2 is a grid enabled message passing interface that allows grid node to communicate candidate sets to all other grid nodes. In distributed design , clusters of workstations are interconnected across a WAN [1]. All the nodes in distributed systems are connected as a computational grid. MPICH-G2 automatically converts data in messages sent between machines of different architectures and supports multiprotocol communication by automatically selecting TCP for intermachine messaging and (where available) vendor-supplied MPI for intramachine messaging.

II. Problem Definition

Personal Communication Systems (PCSs) will support a huge user population and offer services that will allow the users to access various types of data such as video, voice and images. A PCS allows dynamic relocation of mobile users since these systems are based on the notion of wireless access. Mobility management in mobile computing environments covers the methods for storing and updating the location information of mobile users who are served by the system. A hot topic in mobility management is mobility prediction. Mobility prediction can be defined as the prediction of mobile user’s next movement where the mobile user is traveling between the cells of a PCS or GSM network. An association rule like L1 → L2 is discovered in the user’s moving behavior. The predicted movement can be used to increase the efficiency of PCSs.

U. Sakthi is doing Ph.D in Computer Science and Engineering, Anna University, Chennai, India (e-mail: sakthi.ulaganathan@gmail.com).

R. Hemalatha is with Computer Science and Engineering, St.Joseph’s College of Engineering, Jeppiaar Educational trust, Old mahapalipuram Road, Chennai, India.

R. S. Bhuvaneswaran is with the Ramanujan Computing Centre, Anna University, Chennai, India.

In the first phase of our three phase algorithm, movement data of mobile users is mined for discovering regularities in inter-cell movements. These regularities are called mobility patterns. Mobility rules are extracted from the mobility patterns in second phase of our algorithm. In the third phase, the mobility rules, which match the current trajectory of a mobile user, are used for the prediction of the user's next movement. The movement of a mobile user from his current cell to another cell will be recorded in a database which is called Home Location Register (HLR). The HLR stores the permanent subscriber information in a mobile network. Every base station keeps a database in which the profiles of the users located in this cell are recorded which is called Visiting Location Register (VLR). The VLR maintains temporary user information like current location for managing requests from subscriber who are out of the home area. The movement history of a mobile user is extracted from the logs on its home location register.

Let $I = \{i1,i2,…..,in\}$ be a set of items. Let DB be a database of transactions, where each transaction T consists of items such that $T \subseteq I$. Given itemsets $X \subseteq I$, a transaction T contains X if and only it $X \subseteq T$. An association rule is an implication of the form $X \rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \Phi$. The association rule $X \rightarrow Y$ holds in DB with confidence c if the probability of a transaction in DB which contains X also contains Y is c.

Frequent patterns are patterns that appear in a data set frequently. For example, a set of items, such as milk and bread, which appear frequently in a transaction data set is a frequent itemset. A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history of data base, is a frequent sequential pattern. In general, association rule mining can be viewed as a two step process: 1. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, min_sup. 2. Association rules are generated from the frequent itemset which satisfies minimum support and minimum confidence.

Distributed Association Rule Mining (DARM) discovers rules from various geographically distributed datasets. Fast Distributed Mining (FDM) of association rules generates a smaller number of candidate sets and reduces the number of messages to be passed at mining rules. The essential task of a distributed association rule mining algorithm is to find the globally large itemsets L. Every site computes the local support counts of all these candidate sets and broadcasts them to all other sites. Subsequently, all the sites can find the globally large itemsets for that iteration, and then proceed to the next iteration.

MPICH - G2 is a grid enabled implementation of Message Passing Interface (MPI) that allows a user to run MPI programs across multiple computers at the same on different sites. This library also use the services provided by the Globus Toolkit for authentication, authorization, resource allocation, executable staging and I/O, as well as for process creation for monitoring and control. The MPICH-G2 implementation of the message Passing Interface uses Grid Resource Allocation and Management (GRAM) to coschedule subtasks across multiple computers. GRAM provides a web service interface for initiating, monitoring, and managing the execution of arbitrary computations on remote computers.

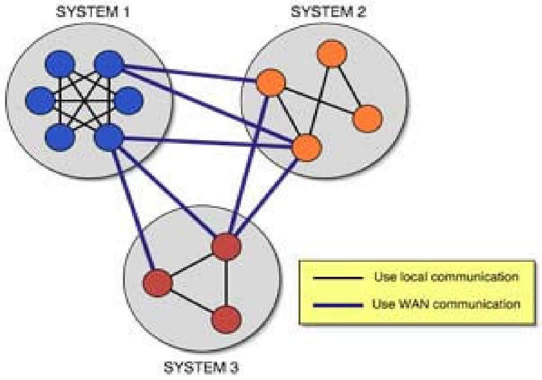

Fig. 1 Grid Configuration for PDKD

III. Related Work

Java Agents developed by Stolfo groups for meta-learning is an agent-based distributed data mining system that has been developed to mine data stored in different sites for building meta-models. In JAM agent is developed using java applets for supporting movements to sites. Kensington Enterprise parallel and distributed data mining is based on three tier architecture and implemented in java and uses the Enterprise JavaBeans component architecture. Bio-diversity database platform is another agent-based distributed data mining system implemented in java. There are several groups working in knowledge grid to develop distributed data mining system. Knowledge grid supports implementation of parallel apriori algorithm on basic grid services. This method is used in several applications such as to generate mobile user's location patterns in GSM architecture to predict the next location of mobile users. Every grid node is deployed with Globus tool kit and apriori grid service to generate frequent mobility patterns. The parallel communication of frequent pattern of different length is transferred between the application program is achieved by MPICH-G2.

IV. Introduction to OGSA

The Service Oriented Architecture (SOA) is essentially a programming model for building flexible, modular, and interoperable software applications [9]. The Grid community adopted the Open Grid Service Architecture (OGSA) as an implementation of the SOA model within the Grid context [3]. OGSA provides a well-defined set of basic interfaces for the development of in interoperable Grid systems and applications. OGSA adopts Web Services as basic technology. Web services define techniques for describing software components to be accessed, methods for accessing these components, and discovery mechanisms that enable the identification of relevant service providers.

In OGSA both the physical resource including computer, storage, program, and the logical resource such as knowledge, algorithm, are represented as grid service. OGSA defines standard mechanisms for creating, naming, and discovering transient Grid service instances.

V. Parallel and Distributed Mining on Knowledge Grid

The KNOWLEDGE GRID is a parallel and distributed software architecture that integrates data mining techniques and grid technologies. The KNOWLEDGE GRID can be exploited to perform data mining on very large data sets available over grids, to make scientific discoveries, improve industrial processes and organization models, and uncover business valuable information [6,8]. Data grid middleware is central for management of data movement on grids. The KNOWLEDGE GRID architecture is designed on top of mechanisms provided by grid environments such as the globus toolkit. It uses the basic Grid services such as communication, authentication, information, and resource management to build more specific parallel and distributed knowledge discovery (PDKD) services [5].

The KNOWLEDGE GRID services are organized into two layers: the Core KGrid layer, which is built on top of generic Grid services, and the High level K-Grid layer, which is implemented over the Core layer. Fig. 1 shows the process of parallel mining of mobility patterns in GSM architecture. Mined data can be used by experts to provide different services to the mobile users. Grid will be used as a platform for implementing and deploying geographically distributed knowledge discovery and knowledge management services and applications. Apriori service is implemented on high level grid K-Grid layer. Data mining algorithms and knowledge discovery process are both compute and data intensive, therefore the Grid offers a computing and data management infrastructure for supporting decentralized and parallel data analysis. Grid-based data mining would allow corporate companies to distribute compute-intensive data analysis among a large number of remote resources.

VI. Implementation

Initially, the process of data preparation was performed to collect, clean the datsets from home location register located in every base station of PCS. In this paper we proposed the task of parallel mining of association rule in geographically distributed database with the globus toolkit. The Globus Toolkit (GT) supports the development of service oriented distributed computing applications and grid infrastructures. In the OGSA context, the parallel association rule mining process is implemented as a grid service. Data grid is integrated with computational grid resource to perform distributed data mining to extract knowledge [4]. The mining service has several components specific to a grid service: service data access, service data element, and service implementation. The association rules discovery service is interacting with the rest of the grid services: service registry, service creation, authorization, manageability and concurrency.

A. Mining Mobility Patterns

Mobility Prediction can be defined as the prediction of a mobile user's next movement where the mobile user is traveling between the cells of a GSM network. The globus toolkit is an open architecture, open source for building grids. Virtual organization infrastructure is developed using grid technologies. Distributed data mining is accomplished using data mining methods and grid technology. Apriori grid service is implemented on each grid node. Local apriori scans local database to generate candidate sets of frequent-1 itemsets from the local database Di which satisfies local support count. Local frequent-1 itemset is send to other grid nodes using MPICH-G2 to find the global frequent itemsets. Every grid node receives frequent-1 itemsets from all other grid nodes and finally updates its local database with global frequent-1 itemsets. This algorithm terminates with set of large mobility patterns stored in all grid nodes. This algorithm returns global frequent User Mobility Patterns (UMP) of different length.

```
UMPMining()

Input:  All the UAPs in the database
        Minimum value for support, supp_min
        Coverage region graph, G

Output: User mobility patterns (UMPs), L

$C_1$ = the patterns of length one
K = 1
L = null
while $C_k$ is not empty
  foreach UAP a € D {
    S = { s | s € $C_k$ and s is subsequence of a }
      foreach s € S {
        s.count=s.count+s.suppInc
      }
  }
$L_k$ = { s | s € $C_k$ , s.count >= supp_min }
L = L U $L_k$
return L
```

Mobility Pattern mining algorithm

B. Mobility Association Rule Generation

In knowledge Grid framework, the information about the knowledge discovered after a Parallel and Distributed Knowledge Discovery (PDKD) is stored in Knowledge Base Repository (KBR). After mining, mobility patterns of mobile users are stored in Knowledge Base. It can be used to generate mobility rules. Assume that we have a User Mobility Path P = $\{l_1,l_2,...,ln\}$, where l represents location of mobile user and n>1. All the mobility rules which can be derived from such a pattern are:

$\{l_1\} \rightarrow \{l_2,...,ln\}$
$\{l_1,l_2\} \rightarrow \{l_3,...,ln\}$
....
$\{l_1,l_2,...,l_{n-1}\} \rightarrow \{ln\}$

In the above mobility rule, the part of the rule before the arrow is the head of the rule and the part after the arrow is the tail of the rule. When mobility rules are generated, a confidence values are calculated for each rule. For a mobility rule R: $\{l_1,l_2,...,l_{i-1}\} \rightarrow \{li,l_{i+1},....,l_n\}$, the confidence is determined using the formula:

$$\text{Confidence (R)} = \frac{(l_1,l_2,...,ln)}{(l_1,l_2,...,l_{i-1})} \text{ X } 100$$

All possible mobility rules are generated and their confidence values are calculated from the mined mobility patterns. Then the rules which have a confidence higher than a predefined confidence threshold ($conf_{min}$) are selected. These rules can be used in next phase for mobility prediction. The mined mobility rule is compared with the current location of mobile user to predict the next possible locations.

C. Mobility Prediction

The moving behavior of each mobile uses is mined from long-term collection of the user's moving logs. Assume the mobile user has followed a path P = $\{l_1,l_2,...,l_{i-1}\}$ up to now. Our method finds out the rules whose head part contained in path P, and also the last location in their head is l_{i-1}. These rules are called as matching rules. We store the first location of the tail of each matching rule along with a value which is calculated by summing up the confidence and the support values of the rule in an array of such tuples. For example mobile user has followed a path P = {1,3,4,6} up to now and he is currently in location 6.

We found the rules {6}→ {2},{6}→ {5}and {6}→ {3} as the matching rule. The stored array will be [(2, 87.56), (3, 56.78), (then, both location 2 and 5 are predicted location for the next movement. We define the parameter, m, which is the maximum number of predictions that can be made each time the user moves. For prediction we select first m predictions from the sorted tuple array. Then the cells of these tuples are our predictions for the next movement of the mobile user. It means that we use the first m matching rules that have the highest confidence plus support values for predicting the user's next movement.

VII. Experiment and Performance Analysis

To measure the efficiency of parallel apriori algorithm we have set up grid with different configurations. Test were conducted on stand alone PC and two clusters of three nodes each and three clusters of three nodes each. Nodes with in the cluster were connected by LAN link and clusters are connected by WAN link. Each node was installed with the Globus 3 toolkit and deployed with the apriori grid service. Mobile users logs are stored in three different database systems: Oracle 10g, PostGreSQL and MySQL. A series of test were run. First, parallel apriori algorithm is called by apriori grid service. Machine hardware design was P4 2.4 MHz, 1 Gb RAM and operated with WindowsXP and Linux operating system.

Table I shows computation time and transfer time for each configuration. When more than one node is used, computation and transfer is performed in parallel. The total time is calculated by compression/decompression and data transfer. However total speedup factor of about 3 has been achieved employing six nodes and a speedup of about 8 has been achieved by using nine nodes.

Figs. 3 and 4 show, respectively, execution times and speedup achieved under different configurations shown in Table I. It should be observed that, with nine nodes the speedup of the computational is slightly super linear. These results show how the use of the Knowledge Grid may bring several benefits to the implementation of distributed knowledge discovery application both in terms of data analysis distribution and scalability results.

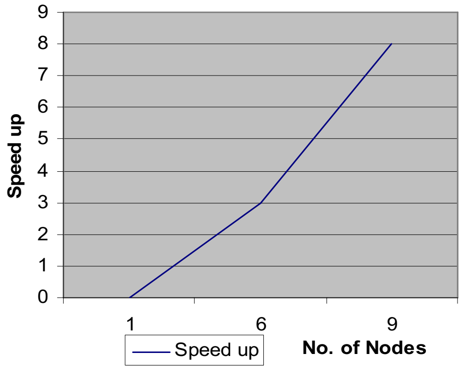

Fig. 3 Computation time

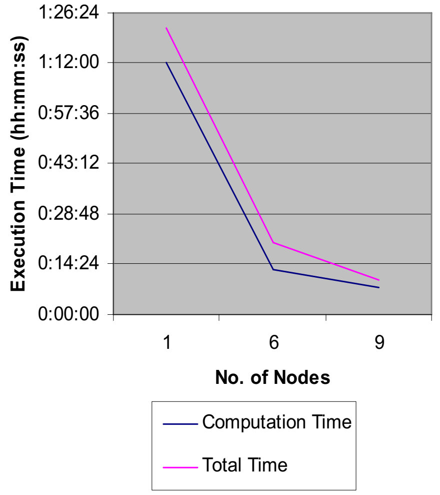

Fig. 4 Speed up

VIII. Conclusion

We built grid system on a cluster of workstation using open-source globus toolkit. In this paper we proposed parallel association rule mining algorithm using grid technology. Both grid technology and parallel mining algorithm reduce the computational time and increase the speed of the application. In all iteration, the Message Passing Interface extended with Grid services (MPICH-G2) supports communication of frequent mobility patterns between the grid nodes in different clusters. The experimental result shows that parallel and

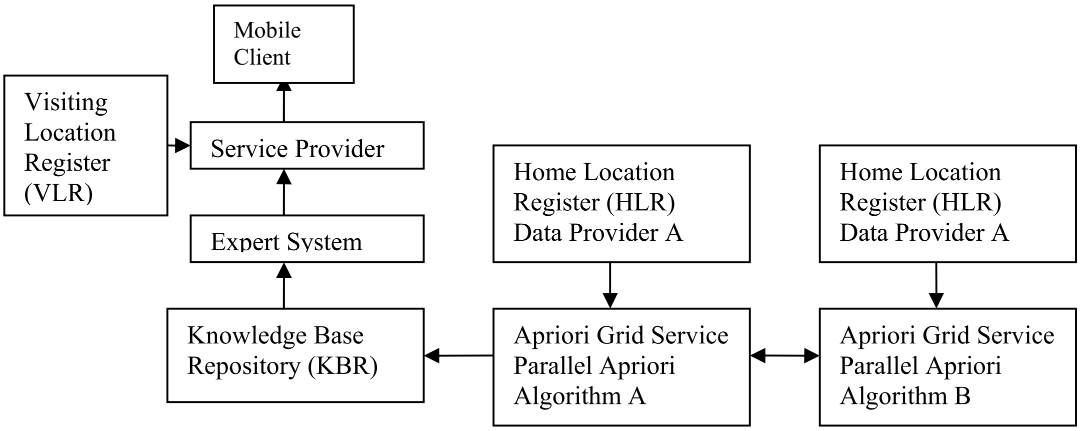
Mobile Client
Visiting Location Register (VLR)
Service Provider
Expert System
Knowledge Base Repository (KBR)
Home Location Register (HLR) Data Provider A
Home Location Register (HLR) Data Provider A
Apriori Grid Service Parallel Apriori Algorithm A
Apriori Grid Service Parallel Apriori Algorithm B

Fig. 2 Parallel and Distributed Association Rule Mining

TABLE I
EXECUTION TIMES ON DIFFERENT GRID CONFIGURATIONS

No of Nodes	Data Size per Node	Execution Time (hh:mm:ss)				
		Data Transfer	Compression	Decompression	Computation	Total
1	540 Mb	0	0	0	1:12:12	01:22:12
6	90 Mb	00:2:56	00:3:28	00:01:12	00:12:56	00:20:32
9	60 Mb	00:00:55	00:00:45	00:00:23	00:07:45	00:09:48

distributed version of Apriori algorithm is optimal than distributed algorithm. Its performance is scalable in terms of the database size and the number of nodes. Grid technology is used as a platform for implementing and deploying geographically distributed knowledge and knowledge management services and applications. The discovered knowledge can be used by the experts to provide various services to the mobile user in web environment. In our problem this method is used to predict the next movement of mobile user while moving among different locations. The Knowledge Grid (KG) is integrated with grid services system to support distributed data analysis, knowledge discovery and knowledge management services.

REFERENCES

[1] M.Perez, A. Sanchez, V. Robels, J.P.na Design and implementation of a data mining grid-aware architecture, Future Genaration computer systems 23, pp 42-47, 2007.

[2] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proc. ACM SIGMOD Intl. Conf Management Data, 1993.

[3] I. Foster, C.Kesselman and S. Tuecke, The anatomy of the grid: enabling scalable virtual organizations, Int'l J. High-perform Comput Appl 15 (2001).

[4] Mario C., Domenico T., Paolo T., "Distributed Data Mining on the Grid", Future Generation Computer Systems, 2002.

[5] Rakesh Agrawal, John C. Shafer, "Parallel Mining of Association Rules", IEEE Transactions on knowledge and Data Engineering, December 1996.

[6] R. Nararajan, R.Sion, T.Phan, A Grid based approach for enterprise scale mining, Future generation computer systems 23, (2007) 48-54.

[7] Sotomayor B, Chilgders L. Globus Toolkit 4: Programming Java Services. Morgan Kaufmann, 2006.

[8] H. Karguta and C. Kamath and P.Chan, Distributed and Parallel Data Mining: Emergence, Growth, and Future Directions, In: Advances in Distributed and Parallel Knowledge Discovery, AAAI/MIT Press, pp.409-416, (2000).

[9] W. Cheung,X.-F.Xhang,Z.-Luo,F.Tong Service-oriented distributed mining, IEEE internet computing 10, (2006) 44-54.

U. Sakthi received B.E in Computer Science and Engineering from Madras University, Chennai, India, in 2001 and M.E in Computer Science and Engineering from Anna University, Chennai, India, in 2005. She is currently doing Ph.D at Anna University, Chennai, India. Her research area interests include grid computing, data mining and distributed systems.

R. Hemalatha received B.E in Computer Science and Engineering from Bharathiyar University and M.E in Computer Science and Engineering from Annamalai University, Chidambaram, India. Her research area interests include grid computing, Ad-HOC networks, software agents.

R. S. Bhuvaneswaran received Master of Technology in Computer Science and Engineering from Pondicherry University, India, in 1996 and Ph.D in Computer Science and Engineering from Anna University, India, in 2003. He received JSPS Post Doctorate Fellowship (2004-2007) in grid computing at Nagaya Institute of Technology, Japan. Presently, he is with Anna University as Assistant Professor. His research interests include distributed systems, wireless networks and fault tolerant systems.