Mobility Prediction of Mobile Users in Mobile Environment Using Knowledge Grid

U.Sakthi† and R.S.Bhuvaneswaran††,

†Department of Computer Science, Research Scholar, Anna University, Chennai 600 025, India
††Ramanujan Computing Center, Asst.Professor, Anna University, Chennai 600 025, India

Summary:
In this paper, we propose a new distributed algorithm named KMPM (Knowledge Grid Based Mobility Pattern Mining) for mining next location of a mobile user in a Personal Communication Systems (PCS) network. The moving logs of mobile users in mobile computing environment are stored in the grid node located in different locations. The generated moving logs are used for mining mobility patterns in a mobile computing system. The discovered location patterns can be used to provide various location based services to the mobile user by the application server in mobile computing environment. Data grid provides geographically distributed database for Computational Grid which implements (Knowledge Grid based Mobility Pattern Mining) KMPM algorithm. We built data grid system on a cluster of workstation using open source Globus Toolkit4.0 and Message Passing Interface extended with Grid Services (MPICH-G2). The experiments were conducted on different configurations and the computation time was recorded for each operation. We compared our result with various grid configurations and it shows a very good speedup.

Key words:
Data Mining, Knowledge Grid, Mobility Rules, Mobility Prediction

1. Introduction

A Personal Communication System (PCS) allows mobile users to move from one location to another location since these systems are based on the notion of wireless access. In mobile system each mobile user is associated with Home Location Register (HLR) which stores up-to-date location of the mobile users. These logs accumulate as large database, in which data mining technique is applied to find the frequently followed location. Each Base station (BS) in PCS is connected with Separate Home Location Register led to the geographically distributed data grid node. Grid network was built with cluster of grid node contains moving logs of mobile users. Existing research work applied data mining technique on mobile data for path mining in a single database server [3]. If the size of the moving logs is very large, the overhead in integrating the data source will be too high. To overcome this problem data mining algorithm is executed in distributed environment. The most prominent example of distributed environment is grid, where a large number of computing and storage units are interconnected over a high speed network.

Data grid is designed to allow large moving logs to be stored in repositories. In business area it is necessary to develop environment for analysis, inference and discovery over the data grid. Therefore, the evolution of the data grid is represented by knowledge grid offering high level services for distributed mining and extraction of knowledge from data repositories available on data grid [4]. The knowledge Grid (KG) is a parallel and distributed architecture that integrates data mining techniques and grid technologies. The Knowledge Grid is exploited to perform distributed data mining on very large data sets available over grids to find hidden valuable information, process models to make decisions and results to make business decisions. In our work knowledge grid is developed to predict the next location of mobile user in mobile environment. By using the predicted location, the system effectively allocate resources to mobile users in the neighbor location and it is possible to answer the queries that refer to the future position of mobile uses.
The outline of this paper is as follows. In Section 2, the process of mining mobility pattern (MP) of mobile users in grid environment is described. Section 3 describes the related work in distributed data mining and Section 4 describes the services of knowledge grid. Section 5 describes how the inter-process communication is performed in MPICH-G2. Section 6 explains the distributed algorithm for mining mobility patterns in grid environment. Section 7and Section 8 explains the process of generating mobility rules and mobility prediction. The performance analysis of the proposed method is described in Section 9. The conclusion is given Section 10.

2. Problem Definition

The mobile users move from one location to another location in a wireless PCS network. The coverage area of the network is divided into number of location areas. Each mobile device is linked with the Base Station (BS). Each Base station contains Home Location Register (HLR) which stores permanent details of the mobile users and

Manuscript received January 5, 2009
Manuscript revised January 20, 2009

Visiting Location Register (VLR) which stores temporary details of the mobile users. These register includes attributes like user ID, user location, call time, call duration, etc. User ID acts as a key for the mobile user records. The movement history of a mobile user is extracted from log files and it is stored in grid node for mining mobility patterns. The movement of mobile user is called as User Actual Path (UAP) which have the form UAP = $\langle l_1, l_2, \ldots, l_n \rangle$, where n is the number of locations followed by the mobile user and l_k represents kth location in the movement path. In our grid network, mobile user movement history is geographically distributed in different data grid nodes. Knowledge grid based mobility pattern mining algorithm is executed on computational grid over the data grid to generate the trajectories that are frequently used by mobile users. The frequently followed trajectory is named as User Mobility Pattern (UMP) which is used to generate mobility rules. The communication of candidate itemsets between grid nodes is achieved by MPICH-G2 and reduces the computation overhead. Our proposed algorithm is executed on knowledge grid to generate mobility patterns from the database distributed on the data grid node. The execution time of our parallel and distributed algorithm is relatively small when compared to sequential algorithm.

Let UAP = $\langle l_1, l_2, \ldots, l_n \rangle$ be the set of locations. A database DB is a set of mobile user records, where each record contains set of locations. Let DB = {DB1, DB^2,...,DB^m} where m is number of data grid nodes in grid network and DB^k is the database stored in kth data grid node. Local frequent locations are mined and it is communicated to all other grid node to find the global frequent locations called as mobility patterns. Mobility rules are generated from the mobility patterns which is in the form of A $\rightarrow$ B having global confidence c if c% of records in DB that contain A followed by B. Mobility rule A$\rightarrow$ B represents the mobile users current location is A then he/she will move to location B. The mobility rule A –> B has global support s if s% of records contain A $\cup$ B. In general, mobility prediction is performed in two steps. 1. Find all mobility patterns: By definition, each of these locationsets will occur at least as frequently as a predetermined global minimum support count, min_sup. 2. Mobility rules are generated from the mobility patterns which satisfy global minimum support and minimum confidence.

3. Related Work

The centralized mobility pattern algorithm was discussed in [4]. In mobile environment, moving log size is very large. It will increase the overhead to integrate all the moving logs into one database server. The algorithm proposed in [4] cannot be efficient for large data size. Many parallel and distributed variants of sequential apriori algorithm have been discussed in other resources [6] and [7]. In [5], grid implementation of frequent itemsets in a grid environment deals with sales transaction of a company. These algorithms [5], [6] and [7] cannot be used directly in our domain, because this algorithm does not take into account the network topology while generating the candidate patterns. The generation of candidate pattern in [5] is not same as the candidate pattern in mobile environment. In PCS, only the sequence of neighboring location of the network can be considered as the mobility pattern. We propose parallel and distributed knowledge grid mining approach based on Apriori algorithm for mining mobility patterns in mobile environment.

4. Knowledge Grid

Grid computing is the most emerged area for high performance distributed applications like knowledge discovery process. Knowledge Gird (KG) is an integration of basic grid services, data grid services and computational grid services for distributed data mining and knowledge discovery. We built Knowledge Grid using the open-source Globus Toolkit [9].

Globus Toolkit Services

The basic grid components provided by the globus toolkit are:

1. Grid Security Infrastructure (GSI): Provides authentication (identity of the users and services) based on certificate produced by certificate authority and standard X.509.The mutual authentication is provided by Secure Socket Layer (SSL) protocol.

2. Monitoring and Data Service (MDS): MDS is an information service component provides information about available grid resources and periodically collects their status. It provides static information like hostname, operating system, etc., and dynamic information like CPU workload, memory status.

3. Globus Resource Allocation and Management (GRAM): responsible for resource allocation, job creation, monitoring management, job control and provides interface for job submission on remote machine.

4. GridFTP: It is an extension of FTP for parallel data transfer, file transfer based on GSI authentication mechanisms.

5. HeartBeat Monitor (HBM): Responsible for identifying globus process failures and application process failures and immediately recovery action can be taken.

6. Global Access to Secondary Storage (GASS):

7. Dynamically-Updated Resource Online Co-allocator (DUROC): It acts as a coordinator between subjobs running on different computational grid.

Knowledge Grid Services

The knowledge grid contains Core K-Grid layer and High level K-Grid layer [11]. The Core K-Grid layer of knowledge grid performs two main services. 1. Knowledge Directory Service (KDS) manages metadata about data source, algorithm used for data mining, mining results and visualization tool. All metadata are represented in eXtensible Markup Language (XML) documents stored in Knowledge Metadata Repository (KMR). The discovered mobility patterns after the execution of distributed mining process is stored in Knowledge Base Repository (KBR) 2. The Resource Allocation and Execution Management Service (RAEMS) generates datamining process execution plan and it will be stored in a Knowledge Execution Plan Repository (KEPR). The execution plan will generate resources request expressed using the Resource Specification Language (RSL) for GRAM. The high level K-Grid layer supports the following services. 1. Data Access Service (DAS) is responsible for accessing data for data mining. 2. Tools and algorithm access Service (TAAS) is responsible for loading tools and algorithm defined in KDS. 3. Execution Plan management Service (EPMS) enables users to create execution plan by assigning programs to data resources. On multiple execution of program, different execution plans are created. 3. Result Presentation Service (RPS) is responsible for presenting mobility patterns to the users stored in Knowledge Base Repository (KBR).

5. InterProcess Communication using MPICH-G2

MPICH-G2 is a grid enabled open source library for implementing Message Passing Interface (MPI). It supports parallel and distributed data mining MPI application to run on cluster of machines of different architecture. MPICH-G2 uses TCP for inter-machine communication and vendor API for intra-machine communication. Our distributed KMPM algorithm is executed on several computational grids using MPICH-G2 component. MPICH-G2 uses RSL script for subjob execution. The design of our parallel and distributed mining of mobility pattern application on knowledge grid is shown in Figure 1. Initially Grid Security Infrastructure (GSI) generates certificates for the user authentication to sign on other site. The user can use Monitoring and Discovery Service (MDS) to select grid node based on memory, CPU load and network topology. The globus-run command is executed to submit job on multiple machines by creating MPI computation. MPICH-G2 uses RSL to specify the URL of the computational and grid resources. The RSL script defines the job in the following way:

```
+
(&(resourceManagerContact = "kalannia/jobmanager-pbs")
(count = 10)
(label = "subjob 0")
(environment= GLOBUS_DUROC_SUBJOB
_INDEX 0)
(directory = "/home/sakthi/gridmining")
(executable = "/home/sakthi/gridmining/KMPM")
)
```

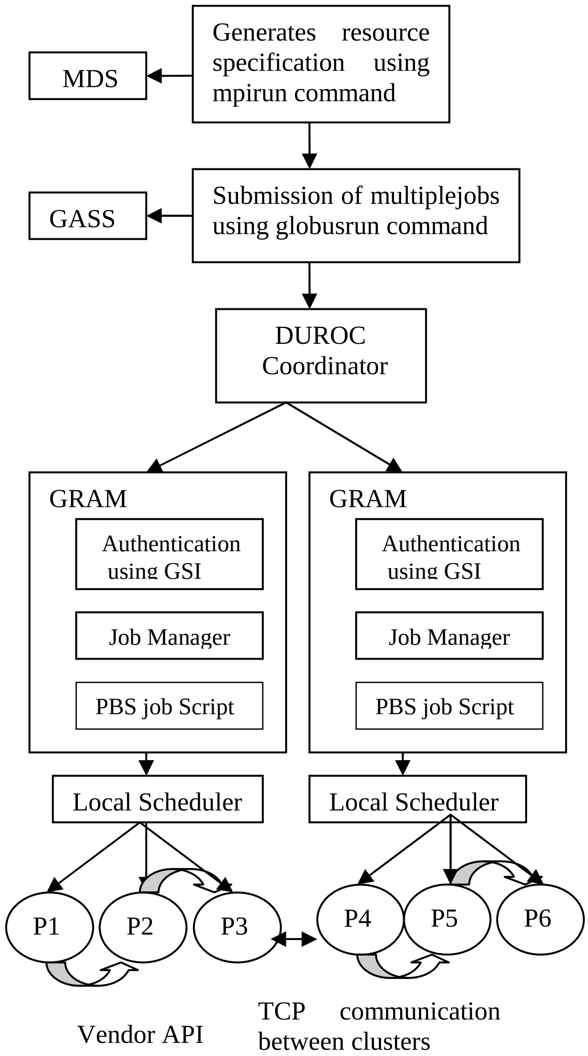

Figure 1: Interprocess communication

The parameter "resourceManagerContact" specifies the URL of the cluster resource and the corresponding jobmanager-pbs. The parameter "count" specifies the number of nodes required for computation, and the "label" represents name of the sub-job. The parameter "environment" specifies the directory in which the globus is installed. The parameter "directory" specifies the working directory, and the "executable" parameter specifies the location of the executable. The mining job is started on the computational resource by using GRAM running on each server. The MPICH-G2 calls Dynamically-Updated Request Online Coallocator (DUROC) library file to start the mining on multiple computational resources specified by the RSL script.

6. KMPM algorithm

Let us assume there are n processing nodes P1, P2,…,Pn in our distributed system. The data base contains mobile user actual path (UAP) is distributed over n processing nodes. In our work, modified form of Count Distribution (CD) algorithm is used to mine User Mobility Pattern (UMP) from User Actual Path (UAP). The locations which are frequently followed by mobile user are called Mobility Patterns. Count Distribution algorithm is previously used in various domains. In our work, CD algorithm is used with new method for calculating support count of subsequence in UAP. Let X.sup and X.sup$_i$ be the global and local support count of subsequence X at a process P_i. X is globally large if X.sup > S and locally large if X.sup$_i$ > S_i , where S is Global minimum support and S_i is local minimum support at P_i. Our KMPM algorithm is executed on n processing nodes in parallel. Intermediate results are transferred to other nodes using send and receive commands of MPICH-G2. The pseudocode for KMPM algorithm is given below:

```
// pseudocode for local mining at a process Pi
  LSk=null //k is the length of subsequence
  for each UAP a ∈ Di
  {
     find the subsequence of UAP and put it in S
     for each subsequence s ∈ S
        {
       //calculate the support count and store it in LSk
       s.count=s.count+s.suppInc
        }
   }
  // pseudocode for global mining at kth pass
  k=1
  while (GSk ≠ null)
  {
    for ( i=1; i<n; i++)
       {
         node Pi exchange and merge local support
         counts of LSk with all n nodes and
         find the global support count of all
         subsequence and store it in GSk
       }
    for each subsequence in GSk
       {
       if the global support count of subsequence is above
       the minimum global support count then
       put the subsequence in  global mobility pattern
       GMPk
       }
     k++
  }
```

The above algorithm is an adaptation apriori algorithm in a distributed environment. Every process generates subsequence of length k called as Local Subsequence (LS_k) and then calculates local support count for each LS_k. These subsequence local support count is exchanged with all other process to generate Global Subsequence (GS_k) and then calculates global support count for each GS_k. The subsequences which have a support count greater than the threshold global support count are selected as Global Mobility Pattern (GMP_k). For instance, consider UAPs $\langle 4, 6, 8, 0, 5 \rangle$, $\langle 2, 4, 8, 0, 6 \rangle$ and $\langle 1, 2, 4, 6 \rangle$ where the number 4 represents location of mobile user. The support count of the subsequence $\langle 4, 6 \rangle$ can be calculated as follows. s.count=s.count+suppInc and suppInc= $\frac{1}{1+totdis}$ where totdis is the number of location between 4 and 6. s.count value is 2 because it appears in 1st and 3rd UAP. In 2nd UAP there are two locations between 4 and 6. Therefore the support value for 4 and 6 is $\langle 4,6 \rangle$.count = $2+\frac{1}{1+2}$=2.33. It will increase the accuracy of the support counting. This algorithm will generate more accurate Global Mobility Patterns (GMP).

7. Mobility Rule generation

In our Knowledge grid, after the execution of parallel and distributed mining algorithm, Mobility Patterns (MP) are stored in Knowledge Base Repository (KBR). It can be used to generate mobility rules. For example, Mobility pattern is $\langle 4, 6, 8 \rangle$.

Mobility rules are as follows:

$\langle 4 \rangle \rightarrow \langle 6, 8 \rangle$

$\langle 4, 6 \rangle \rightarrow \langle 8 \rangle$

From the UMPs, all possible mobility rules are generated and their confidence value is calculated. In general, mobility rule R is represented as $\langle t_1, t_2,..t_i \rangle \rightarrow \langle t_{i+1},t_{i+2},\ldots,t_p \rangle$.

Confidence value for the rule R is calculated using the following formula:

$$\text{Confidence (R)} = \frac{\langle t_1, t_2, ..., t_i \rangle .count}{\langle t_{i+1}, t_{i+2}, ...t_p \rangle .count}$$

Then the mobility rules which have a confidence higher than a predefined confidence threshold ($conf_{min}$) are selected. These mobility 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.

8. Mobility Prediction

In mobile web environment, next location of mobile users is predicted using mobility rule and current location of the mobile user. Mobility rule contains two parts namely; head - the part before the arrow and tail- the part after the arrow. Our process generates set of rules whose head matches with the current location of the mobile user. These rules are called as matching rules. The first location in the tail of the matching rule and match value is stored in the resultant array. Match value is calculated by summing up the support value of the UMP and confidence of the rule. The matching rules in the array are sorted in descending order with respect to match value.

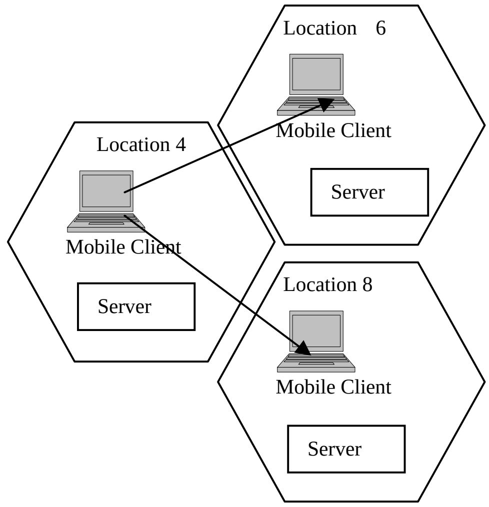

Figure 2: movement of mobile user in GSM network

This process generates most confident and frequent rules. The parameter m defines number of predictions required. It selects only first m locations from the resultant array. In figure 2, there are three locations 4, 6 and 8. For example, currently the mobile user is in location 4. Our algorithm generates matching rules $\langle 4 \rangle \rightarrow \langle 6 \rangle$ and $\langle 4 \rangle \rightarrow \langle 8 \rangle$. The match value is calculated for each predicted location and stored in resultant array. Resultant array contains two values [(6, 78.56), (8, 68.56)]. If m=1, then the location 6 will be predicted as next location. If m=2, then the locations 6 and 8 are the predicted as next locations.

Table1: Execution on time different configurations

No of Nodes	Data Size per Node	Execution Time (seconds)				
		Message Exchange	Compression	Decompression	Computation	Total
1	800 Mb	T0	0	0	6132	6132
8	100 Mb	656	178	72	2976	3882
15	60 Mb	965	45	23	265	1298

The predicted location can be used by expert system to provide location based service to the mobile user. Our KMPM algorithm takes less time to predict the next location.

9. Performance Analysis

To measure the efficiency of KMPM algorithm on grid environment, we have set up grid network with different configurations. The experiment was conducted on single node and two clusters of four nodes and three clusters of five nodes. Each node contains data set with collection of UAPs. Nodes with in the cluster were connected by LAN link and clusters are connected by WAN link. Each node was installed with the Globus Toolkit and deployed with the KMPM algorithm. Data set is stored in different data base systems: Oracle 10g, PostGreSQL and MySQL.

Table.1 shows the message exchange time and computation time for each configuration. When the number of node is increased by 8, the execution time of KMPM algorithm is reduced by 45%. In general, if the number of node is increased, then the computation time of mining algorithm is reduced. It shows a very good speedup. During global mining phase, subsequence support count is exchanged between nodes. Each node performs n send operations and n receive operations to generate GS_1. When the number of node is increased, communication overhead also increased. Compared to sequential algorithm our KMPM algorithm requires less computation time. Figure 4 shows speed up in various grid configurations. If we increased node to 8, the speed will be increased 3 times compared to sequential algorithm. If we increased node to 15, the speed will be increased 5 times compared to sequential algorithm. Our KMPM algorithm provides better performance to mine frequent mobility patterns of mobile users.

Table 2: Parameters used in experiment

Parameter	Definition	Default value
m	Maximum number of predictions	2
S	Global support count	2
Si	Local support count at node Pi	1.55
L	Average length of UAPs	8
$conf_{min}$	Minimum confidence percentage	75%

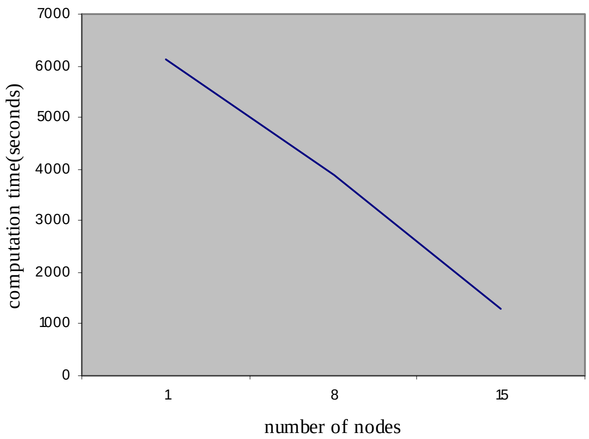

Figure 3: Computation time

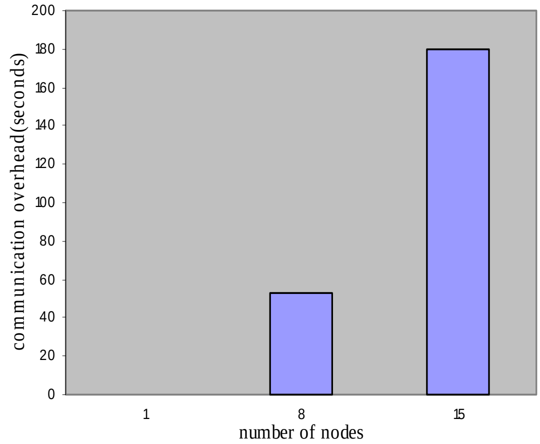

Figure 5: Communication overhead

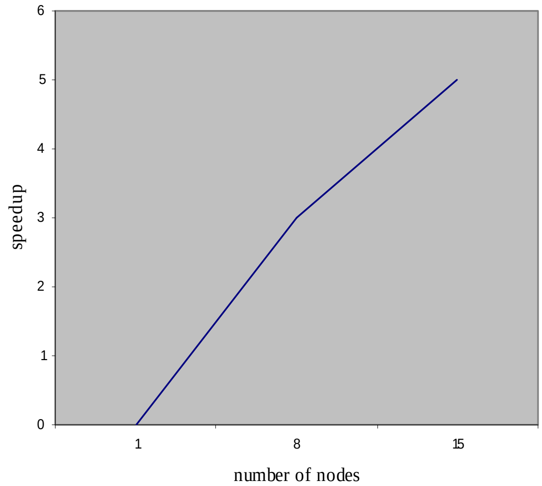

Figure 4: Speedup

10. Conclusion

In this paper, we have proposed parallel and distributed algorithm implemented on Knowledge grid to predict the next location of mobile user in a mobile web computing system. In the first step of mining algorithm, mobility patterns are minined from the User Access Path (UAP) and mobility rules are generated using mobility patterns. Finally current location of mobile user is compared with the mobility rule to predict the location of mobile user. By using the predicted movement, the system can effectively allocate resources and provide location based services to the mobile users. Knowledge Grid based Mobility Pattern Mining (KMPM) algorithm for mobility prediction needs less computation time compared to sequential mobility prediction algorithm and it supports scalability. The proposed approach shows how the Knowledge Grid system is used for distributed data analysis. Also compared to other distributed system, grid reduces the message communication overhead using MPICH-G2 technology. The subsequence exchange between processes is effectively achieved by using MPICH-G2.

References

[1] Congnan Luo, Anil L. Pereira and Soon M.Chung. Distributed Mining of Maximal Frequent Itemsets on a Data Grid System. The Journal of Super Computing, 37, 71-90, 2006.
[2] Lamine M. Aouad, Nhien-An Le-Khac and Tahar M. Kechadi. Distributed Frequent Itemsets Mining in Heterogeneous platforms. Journal of Engineering, Computing and Architecture. Volume 1, Issue 2, 2007.
[3] Gokhan Yavas, dimitrios Katsaros. Ozgur Ulssoy and Yannis manolopoulos. A data mining approach for location prediction in mobile web environments. Data and Knowledge Engineering, 54, 121-146, 2005.
[4] F.Benman. Private Communication. Nov, 2001.
[5] Cristian Aflori and Mitica Craus. Grid implementation of Apriori algorithm. Advances in engineering software. Volume 38, Issue 5, 295-300, 2007.
[6] R. Agrawal and J. Shafer. Parallel mining of association rules. IEEE Transaction on knowledge and Data Engineering, 8(6): 962-969, 1996.
[7] Assaf Schuster, Ran Wolff and Dan Trock. A High-Performance Distributed algorithm for mining association rules. Knowledge And Information Systems (KAIS) Journal, Volume 7, Number 4, May 2005.
[8] M. Cannataro and D. Talia. The Knowledge Grid. Commun. ACM, volume 46, no. 1, 89-93, 2003.
[9] http://www.globus.org
[10] I. Foster and C. Kesselman. The Anatomy of the grid: Enabling scalable virtual organizations. Int. Journal of super computing applications. Volume 15, no. 3, 2001.
[11] Mario Cannataro, Antonio Congiusta, Andrea Pugliese, Talia, Paolo Trunfio. Distributed data mining on grids: Services, Tools and applications. IEEE Transaction on Systems Man and Cybernetics. Vol. 34, No.6, 2004.

Biographical Notes

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. Currently she is doing Ph.D at Anna University, Chennai, India. Her research area interest includes grid computing, knowledge grid, data mining and distributed systems.

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 interest includes distributed systems, wireless networks and fault tolerant systems.