ISBN 978-93-84422-22-6
Proceedings of 2015 International Conference on Image Processing, Production and Computer Science (ICIPCS'2015)
Istanbul (Turkey), June 3-4, 2015 pp. 14-22

Hybrid Distribution for Association Rules Extraction on Grid Computing

Mohammed REBBAH[1], Mohammed El Amine YEMRES, Miloud KHALDI, Mohammed DEBAKLA[1]

University of Mustapha Stambouli Mascara, Algeria
[1] LRSBG Laboratoire, TVIM Team
rebbah_med@yahoo.fr, yermes.amine@yahoo.fr, khaldi_univ@yahoo.fr debakla_med@yahoo.fr

Abstract: *The extraction of association rules in distributed systems is capable of greatly reducing the time of extraction and exploitation of large data sets, however, these benefits are forced to emerging issues related on the one hand the nature of distributed systems and the sharp increase in volumes of data and its geographic dispersion. The trend towards grid computing that enables applications to handle distributed heterogeneous computing resources as a single virtual machine is justified by these last two points. Globus_HyDAR, our service developed under Globus grid, falls within the framework of grids Data mining; has proposed a hybrid intelligent distribution (horizontal and vertical) data. The results clearly show gains execution time compared to the horizontal and vertical distributions.*

Keywords: *Distributed Data mining, Association Rules, Grid Computing, Distribution of data, Apriori.*

1. Introduction

Data mining can be viewed as the formulation, analysis, and implementation of an induction process (proceeding from specific data to general patterns) that facilitates the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. Data mining techniques are widely used today for the analysis of large corporate and scientific data stored in databases. However, industry, science, and commerce fields often need to analyze very large datasets maintained over geographically distributed sites by using the computational power of distributed systems [1].

Data mining algorithms in particular and knowledge discovery techniques in general are both computational and data-intensive. Because of the complexity of the Data mining algorithms that involves an iterative process slow and cumbersome in space, the existing algorithms are face to a technical challenge that is far from being solved by traditional architectures (Client-Server, P2P ...) [2]. The Grid promises to provide means for utilizing geographically distributed resources as a single meta-system .The term "computing grid" designates a group of connected computers that combine resources to work together, appearing to users as a single virtual computer. This architecture can play a significant role in providing an effective computational infrastructure support for Data mining.

The usage of grid resources by knowledge discovery systems has the advantage that on the one hand more complex and computational-intensive knowledge discovery algorithms can be applied and on the other hand all the data resources available via grid technologies can be perceived by grid-enabled knowledge discovery applications as accessible data repositories. The key to the success of grid computing is the middleware, software that organizes and integrates disparate computing capabilities available to the grid; essentially all major Grid projects are currently built on protocols and services provided by the Globus Toolkit developed under a project by the consortium Globus Alliance.

This paper deals with high performance search of association rules. It proposes to built an ”intelligent” database fragmentation by hybridation of horizontal and vertical distribution; this new distribution is exploited by new algorithm that we called HV-Distrib. This algorithm (HV-Distrib) is executed through a grid service **Globus-HyDAR** (**Hy**brid **D**istribution for **A**ssociation **R**ules Extraction on **Globus** Toolkit) conforms to the standard of OGSA and OGSI exploitable on Middleware Globus Toolkit 4.

The paper is organized as follows. In Section 2, we briefly describe GRID environment, OGSA and OGSI platform and HV-Distrib algorithm and his implementation as a grid service on Globus Toolkit 4 are described in section 3. The databases and experiments are described, followed by the results of HV-Distrib executed in GT4 in Section 4 and the section 5 concludes the paper.

2. Related Work

There are several systems proposed in the field of the high-performance Data mining. Most of them do not use computational grid infrastructure for the implementation of basic services such as authentication, data access, communication and security. These systems operate on clusters of computers or over the Internet. The best known systems for distributed Data mining are presented below: Knowledge Grid [3], Grid Miner [4], Discovery Net [5], TeraGrid [6], ADaM (Algorithm Development and Mining) [7] on NASA’s Information Power Grid, and the Data Cutter project [8] have focused on the creation of middleware / systems for mining on top of the data grid. None of these projects consider “distributed” Data mining algorithms.

2.1 Distributed Data Mining (DDM)

A primary motivation for DDM discussed in literature, is that a lot of data is inherently distributed. Merging of remote data at a central site to perform Data mining will result in unnecessary communication overhead and algorithmic complexities. Simply put, DDM is Data mining where the data and computation are spread over many independent sites. Typically, in a DDM environment, each site has its own data source and Data mining algorithms operate on it producing local models. Each local models represents knowledge learned from the local data source, but could lack globally meaningful knowledge. Thus the sites need to communicate by message passing over a network, in order to keep track of the global information. DDM is thus not just about “distributing” the centralized Data mining task on several compute nodes[9].

2.2 Data mining on the Grid

So far, little attention has been devoted to knowledge discovery on the Grid. An attempt to design architecture for performing Data mining on the Grid was presented in [10]. The authors present the design of a Knowledge Grid architecture based on the non-OGSA-based version of the Globus Toolkit, and do not consider any concrete application domain. This architecture extends the basic grid services with services of knowledge discovery on geographically distributed infrastructures. DataCutter [11] is a middleware framework for distributed Data mining computations in a Grid environment. It targets distributed, heterogeneous environments by allowing decomposition of application-specific data processing operations into a set of interacting processes. In DataCutter, data intensive applications are represented as a set of filters. A filter is a user-defined object with methods to carry out application-specific processing on data.

3. Globus_HyDAR

As previously cited, this article proposes a grid service (Globus_HyDAR) that combines association rules discovery Data mining task with Grid technologies. This service is particularly useful for large organizations, environments and enterprises that manage and analyze data that are geographically distributed in different data repositories or warehouses. The proposed service also deals with a technical challenge, which is distribution of data.

3.1 Grid Environment and OGSA

Grid computing is a form of distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations. The

Open Grid Services Architecture (OGSA) is an SOA (Service Oriented Architecture) for the Grid. It is a non-proprietary effort by Argonne National Laboratory, IBM, the University of Chicago and other institutions, that combines Grid computing with Web services. The goal of this architecture is to enable the integration of geographically and organizationally distributed heterogeneous components to form virtual computing systems that are sufficiently integrated to deliver desired QoS [12, 3]. OGSA defines the mechanisms for creating, managing, and exchanging information among entities, called Grid Services. The Globus Toolkit 4.0 is an open source project that can be downloaded from the Globus Alliance Web site. It provides a set of OGSA capabilities based on WSRF (WS-Resource Framework). The WSRF is a set of Web Service specifications being developed by the OASIS (Open Access Same-Time Information System) organization that, taken together and with the WS-Notification (WSN) specification, describe how to implement OGSA capabilities using Web services.

As stated previously, OGSA represents everything as a Grid Service. Grid Services are stateful transient Web Service instances that are discovered and created dynamically to form larger systems [13]. Transience is what allows for the dynamic creation and destruction of services and has significant implications for how services are managed, named, discovered, and used-that is what makes a Grid Service different from a Web Service. A Grid Service conforms to a set of conventions, expressed as WSDL (Web Service Description Language) interfaces, extensions, and behaviors, for such purposes as:

- Discovery: mechanisms for discovering available services and for determining the characteristics of those services so that they can be invoked appropriately
- Dynamic service creation: mechanisms for dynamically creating and managing new service instances
- Lifetime management: mechanisms for reclaiming services and state in the case of failed operations
- Notification: mechanisms for asynchronously notifying Grid Service clients of changes in state

3.2 HV-Distrib

In a parallel and distributed context, such as a grid or a cluster, constraints over the execution platform must be taken into account: the nonexistence of a common memory imposes to distribute the database in fragments; the high cost of communications suggests that parallel treatments must be as independent as possible. Since the problem of association rules needs to compare all data together, it is necessary to find an intelligent data fragmentation to distribute the computation (independent fragments).

3.2.1 Hybrid Distribution of Data

Though the simplicity of horizontal partitioning that it gives a partial view of the different parts of the table, remains the problem of balancing load between sites still exists. Similarly, although the vertical distribution exceeds the inconvenience of waiting and that each client has a global view on a subset of attributes, it is much more complex than the partitioning horizontal, this is due mainly to the large number of possible alternatives to the more requires additional joints where an application accesses multiple partitions. The following sections introduce the research contributions of this dissertation.

The principle is to realize a hybrid distribution by an algorithm that takes partitioning of the data set so that each customer has a vertical partition with another horizontal to keep the active role of the customer and reduce the waiting time.

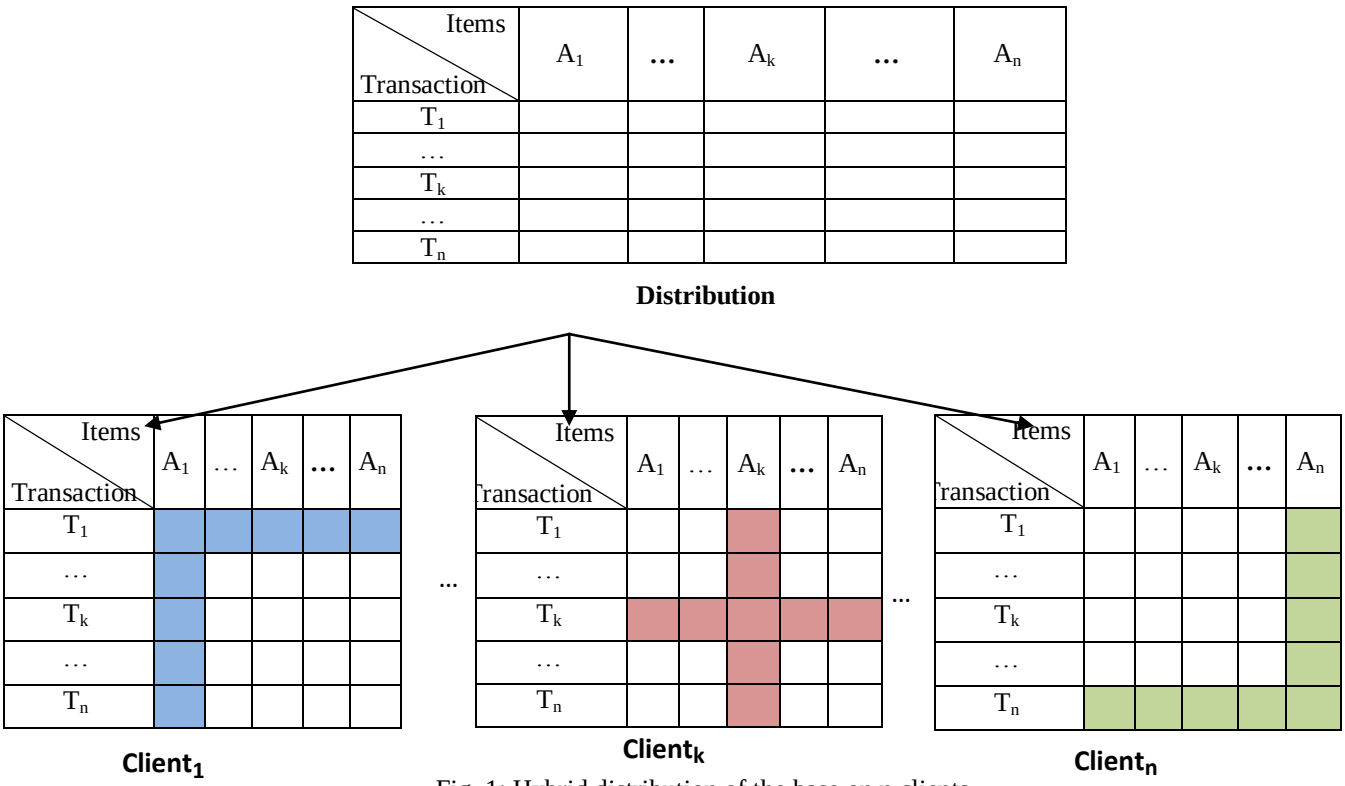

Fig. 1: Hybrid distribution of the base on n clients.

Vertical partitioning is to define for each customer a projection of the table where the fragments have the same number of attributes. This projection allows customers to be more independent and eliminate the transfer. In the meantime, and to avoid waiting of the customers we have assigned a horizontal fragment to the costumers to alternate between the two types of partition.

3.2.2 Functioning

At the launch of the calculation by the user, the server will trigger a series of generation of configuration files and send the result of each client. These files contain a full description of the partitions on local treatment to each client, i.e. the horizontal and vertical partitions.

After receiving the configuration files, two types of treatment will be made at the client, the first vertical partition on which to calculate the frequent itemsets and send to the server. When the server receives frequent itemsets, it generates global candidates and send them to customers, to start the second treatment on the horizontal partition on the calculation of frequencies from the list of candidates received and predefined by the server and sent to new to him.

The server side collects its frequencies, frequent extract candidates and added to frequent itemsets resulting from local customers in order to generate new candidates.

This process is iterative, each client uses the waiting time for generation of candidates at the server level by the calculation of frequent itemsets of vertical partition, and upon receipt of the list of candidates for the next level, it calculates the frequency of appearance of this candidates on the horizontal partition and communicate the result to the server, in turn, it aggregates the results of all customers to generate the list of candidates for the next level.

3.2.3 Algorithm

HV-Distrib (*Horizontal Vertical Distribution*) algorithm is developed to realize the partitioning and hybrid distribution proposed.

The main steps of the algorithm at the server can be summarized as follows:

Step 2 ... 13: For each client send the vertical and horizontal fragment specifying the recording start and end (tran-debut, tran-end) and the attribute start and end (item-begin, item-end).

Step 14 ... 18: The server brings together in a list (list-$freq_{level}$) all frequent itemsets from lists of frequent itemets calculated on vertical partition and sent by customers.

Step 19 ... 20: The server generates a list of new candidates of the second level.

Step 21: While the list of candidates is not empty do (Step 22 ... 34).

Step 22 ... 24: For each client, the server sends the list of candidates generated.

Step 25 ... 27: The server brings again lists of frequents (calculated from vertical partition) of the current level determined by customers.

Step 28 ... 30: The server brings together in a list (CountGlobal) frequencies of candidates with lists of frequencies (list-count) calculated from the horizontal partition by customers, each in its horizontal part.

Step 31: The server calculates the supports of candidates from the list (CountGlobal).

Step 32: The server includes a list of global frequent itemsets (list-$freq_{level}$ ∪ $freq_{Hlevel}$).

Step 33 ... 34: The server generates a list of new candidates.

1: SHV-Distrib(dataset, *nbclient*, *nbtransaction*, *nbitem*)
2: **For** *i*=1 **to** *nbclient*
3: *item-debut* = 1 ;
4: *item-fin* = *nbitem* / *nbclient* ;
5: *tran-debut* = 1 ;
6: *tran-fin* = *nbtransaction* / *nbclient* ;
7: P_{vi} = **Select*** $A_{item\text{-}debut},\ldots,A_{item\text{-}fin}$ **from** dataset ;
8: P_{hi} = **Select* from** dataset **where** *num-tran*≥*tran-debut* **and** *num-tran*≤*tran-fin*;
9: Send ($client_i$, P_{vi}, P_{hi}) ;
10: *item-debut* = *item-fin* + 1 ;
11: *item-fin* = *item-fin**(*i*+1) +1 ;
12: *tran-fin* = *tran-fin**(*i*+1) +1 ;
13: **End For**
14: *level* = 1 ;
15: $liste\text{-}freq_{level}$ = Ø ;
16: **For** *i*=1 **to** *nbclient*
17: $liste\text{-}freq_{level}$ = ∪ frequent(P_{vi}, *level*, $client_i$ **)** ;
18: **End For**
19: $liste\text{-}cand_{level+1}$ = generer-cand($liste\text{-}freq_{level}$) ;
20: *level*++ ;
21**: While** (*liste-cand* ≠ Ø)
22: **For** *i*=1 **to** *nbclient*
23: Send ($liste\text{-}cand_{level}$, $client_i$);
24: **End For**
25**:** **For** *i*=1 **to** *nbclient*
26**:** $liste\text{-}freq_{level}$ = ∪ frequent (P_{vi}, *level*, $client_i$);
27**:** **End For**
28**:** **For** *i*=1 **to** *nbclient*
29**:** *CountGlobal***=** ∪ (*list-count*, $client_i$);
30**:** **End For**
31**:** $freq_{Hlevel}$ = Calc-freq (*CountGlobal*);
32: $liste\text{-}freq_{level}$ = $liste\text{-}freq_{level}$ ∪ $freq_{Hlevel;}$
33: $liste\text{-}cand_{level+1}$ = generer-cand ($liste\text{-}freq_{level}$) ;

34: *level*++;
35: **End While**

Algorithm 1 : HV-Distrib algorithm at the client server.

The principal steps of the algorithm at the client are as follows:

Step 1 ... 4: The client generates a list of frequent itemsets from the vertical partition of the first level *(nl)* and sends it to the server.

Step 5 ... 7: The client generates a list of frequent itemsets from the vertical partition of the second level and sends it to the server.

Step 8 ... 9: The client receives a list of global candidates of the level 2 sent by the server *(list-cand$_{ng}$)*.

Step 10: While the list of applications is not empty do (step 11 ... 18).

Step 11: increment the local level.

Step 12 ... 13: Calculate the frequency of each itemset of the list of candidates (sent by the server) on the horizontal fragment and send it to the server.

Step 14 ... 15: generate the list of frequent itemsets from the vertical partition of the current local level and send it to the server.

Step 16: increment the global level *ng*.

Step 17: receive the list of candidates of the global level sent by the server.

1: CHV-Distrib(P_{vi}, P_{hi})
2: *nl* : local level ;
3: *ng* : global level;
4: *nl* = 1 ;
5: *list-freq$_V$* = Cal-freq$_V$(P_{vi}, *nl*) ;
6: Send-serveur(*list-freq$_V$*) ;
7: *nl* ++;
8: *liste-freq$_V$* = Cal-freq$_V$(P_{vi}, *nl*) ;
9: Send-serveur(*list-freq$_V$*) ;
10: *ng* = 2 ;
11: *liste-cand$_{ng}$* = Rec (*liste-cand$_{ng}$*) ;
12: **While** (*liste-cand* ≠ Ø)
13: *nl*++ ;
14: *liste-count* = count(P_{hi}, *liste-cand$_{ng}$*) ;
15: Send-serveur(*liste-count*) ;
16: *liste-freq$_V$* = Cal-freq$_V$(P_{vi}, *nl*) ;
17: Send-serveur(*liste-freq$_V$*) ;
18: *ng*++ ;
19: *list-cand$_{ng}$* = Rec (*liste-cand$_{ng}$*);
20: **End While**

Algorithm 2: HV-Distrib algorithm at the client.

4. Experimentation

4.1 Globus Toolkit 4.0.1

To deploy the Data mining system as a grid service, we are using the Globus Toolkit 4.0.1 (GT4) to build a distributed environment over a computational grid. The GT4, developed by the Globus Project, a grid computing research organization, provides technology guidelines to deliver tighter integration between grid

computing networks and Web service technologies, improvements to key grid protocols, database support, integration with J2EE, and an array of other grid-oriented features, according to Globus.

Globus provides the features and basic services necessary for building grids. Thus we find services and mechanisms such as security, location, resource management and communication ... It is composed of a set of modules, each with an interface that the higher level can use to invoke its mechanisms (see table 1.1) [14].

The following table outlines these basic services.

TABLE I: Various basic services of Globus.

Service	Name	Description
Resource management	GRAM	Resource allocation and management processes.
Communications	Nexus	Communication Services unicast and multicast.
Security	GSI	Authentication and authorization.
Information	MDS	Information on the structure and stat of the grid.

4.2 Tests

To evaluate performance of Globus_HyDAR, we explore sets of binary data (0/1) in order to test the functioning of our service and validation of its results. A main feature of these data sets is the important dimension of these; all parameters are listed in following table:

TABLE II: Characteristics of data sets.

Name of Data sets	Size (Ko)	# transactions	# items
Zoo	60	2 000	15
Flag	60	192	21
Transa	167	10 000	8
Mushroom	1 905	8 124	119
Chess	475	3 196	75

These datasets are explored through the grid with a number of Calculation nodes fixed at 4. Computers that we used were single CPU 2.80 GHz Pentium IV's, with 256 MB of Central Memory (RAM). The computers were on 100 MB network.

The table below shows the different results from different data sets by taking several values minconf and minsup.

TABLE III: Results of tests.

Name of Data sets	Minsup	Minconf	Level	Nb-rules	Extraction Time
Zoo	**50**	**50**	**5**	**109**	**00 :01 :04**
	70	**50**	**3**	**6**	**00 :00 :40**
	80	**50**	**2**	**1**	**00 :00 :30**
Flag	**20**	**20**	**4**	**61**	**00 :01 :30**
	30	**30**	**3**	**12**	**00 :01 :00**
	50	**50**	**2**	**1**	**00 :00 :30**
Transa	**20**	**20**	**4**	**26**	**00 :00 :58**
	30	**30**	**4**	**13**	**00 :00 :50**
	50	**50**	**1**	**0**	**00 :00 :33**
Mushroom	**30**	**30**	**9**	**2707**	**00 :10 :45**
	40	**50**	**7**	**526**	**00 :02 :20**
	50	**50**	**5**	**140**	**00 :01 :31**
Chess	**75**	**50**	**11**	**20970**	**00 :17 :20**
	80	**80**	**10**	**8208**	**00 :15 :32**
	85	**50**	**8**	**2653**	**00 :05 :08**

The results obtained in the phase of the test confirm the interest of the emergence of Data mining in grid computing. And to better demonstrate this interest we have compared our results with others obtained with vertical and horizontal distributions implemented on client-server architecture (see Figures 2 and 3).

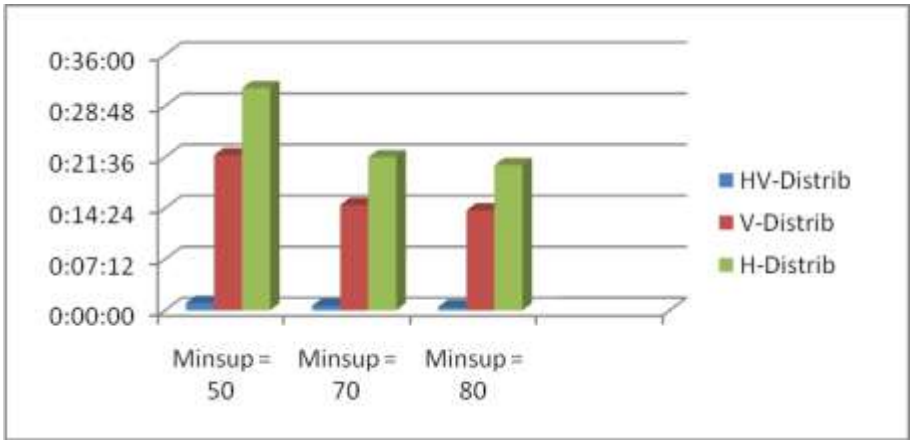

Fig. 2: Comparison with the horizontal and vertical strategies using the Zoo dataset.

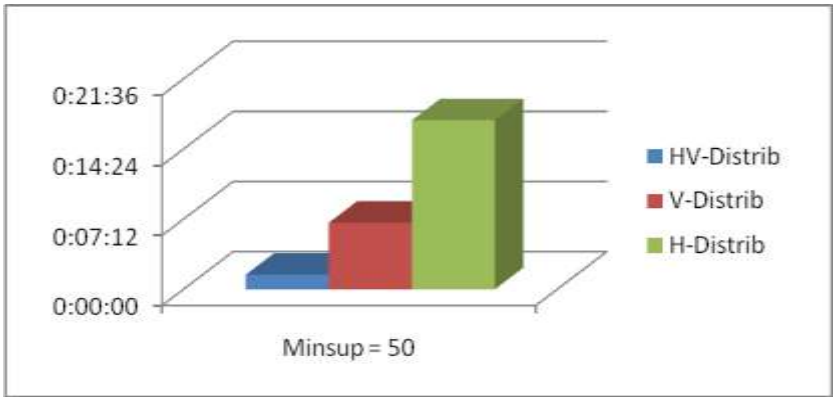

Fig. 3 : Comparison with the horizontal and vertical strategies using the Flag dataset.

5. Conclusion and open Issues

Computing grids bring many advantages to Data mining. This is because Data mining applications are particularly resource hungry; the amount of data available (and so the computational power required to process it) continues to grow exponentially with a geographical distribution. In an attempt to improve exactitude, the algorithms become more complex and require even more computational power. And grid computing provides the computational power required. Because many Data mining tasks can be effectively parallelized, the grid is their natural execution environment.

Our proposed approach offers an effective grid service Globus_HyDAR deployed on Globus Toolkit 4. Globus_HyDAR implements a new algorithm of extraction of association rules that we called HV-Distrib (Horizontal Vertical Distribution) represent an extension of Apriori algorithm; based on hybrid distribution of data. In the future, we will increase the number of calculation nodes. Besides, further experiments should be done to evaluate the affection of data distribution and network bandwidth on the overall performance of the system.

6. References

[1] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, Ramasamy Uthurusamy Advances in Knowledge Discovery and Data Mining (01 February 1996)

[2] Zeng, Li, et al. "Distributed data mining: a survey." Information Technology and Management 13.4 (2012): 403-409. http://dx.doi.org/10.1007/s10799-012-0124-y

[3] Mario Cannataro and Domenico Talia. The knowledge grid. Communications of the ACM, 46(1):89–93, 2003.
http://dx.doi.org/10.1145/602421.602425

[4] Peter Brezany, Juergen Hofer, A Min Tjoa, Alexander Woehrer.GridMiner: An Infrastructure for Data Mining on Computational Grids. Accepted for the APAC Conference and Exhibition on Advanced Computing, Grid Applications and eResearch, Queensland Australia, 2003.

[5] V. Curcin, M. Ghanem, Y. Guo, M. Kohler, A. Rowe, J. Syed, P. Wendel. Discovery Net: Towards a Grid of Knowledge Discovery. In Proc of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2002, Edmonton, Alberta, Canada, July 23 - 26 2002.

[6] TeraGrid, http://www.teragrid.org/about/

[7] J. Rushing, R. Ramachandran, U. Nair, S. Graves, R. Welch, and H. Lin. ADaM: a data mining toolkit for scientists and engineers. Computers and Geosciences, 31:607–618, jun 2005.
http://dx.doi.org/10.1016/j.cageo.2004.11.009

[8] Michael Beynon and Renato Ferreira and Tahsin M. Kurc and Alan Sussman and Joel H. Saltz. Datacutter: Middleware for filtering very large scientific datasets on archival storage systems. In IEEE Symposium on Mass Storage Systems, pages 119–134, 2000.

[9] Maria S. Pérez, Alberto Sànchez, Victor Robles, Pilar Herrero, José M. Peña: Design and Implementation of a Data Mining Grid-aware Architecture. Universidad Polit´ecnica de Madrid, Madrid, Spain.

[10] Mario Cannataro and Domenico Talia. The knowledge grid. Commun. ACM, 46(1):89–93, 2003.
http://dx.doi.org/10.1145/602421.602425

[11] W. Du and G. Agrawal. Developing Distributed Data Mining implementations for a Grid environment. In CCGRID'02, 2002.

[12] Global Grid Forum. The Open Grid Services Architecture, Version 1.0. 2005.

[13] Foster, I.; Kesselman, C.; Nick, J.; & Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, . 2002.

[14] I. Foster et C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit Intl J. Supercomputer Applications Page 13, 1997.