Data Mining Partition in Grid Computing

S.Murali[1], K. Raj Kumar[2], D. Renuga Devi[3], G. Jeya Sudha[4]

[1]Lecturer
Velammal College of Engineering and Technology, Madurai
muralicse2008@gmail.com

[2]Assistant Professor
National Engineering College, Kovilpatti
rajkumark.twins@gmail.com,

[3]Assistant Professor
Renganayagi Varatharaj College of Engineering, Salvarpatti
renuka1011@gmail.com,

[4]Assistant Professor
P.S.R. Rengasamy College of Engineering for Women, Sevalpatti
sudhakrishnan89@gmail.com

Abstract: ***Grid computing is nothing but the computing environment in which the resources are shared by multiple systems to obtain a goal. In day today life performance analysis of very large data sets in the computing environment is necessary in many applications. The distributed grids are formed from the computing resources of multiple individuals or multiple administrative domains. This can make easier to perform commercial transactions. This paper is an introduction to the Grid infrastructure and the potential for machine learning tasks.***

Keywords: Partitioned Data Mining, Grid Computing, Knowledge Grid

1. Introduction

1.1. Grid Computing

Grid computing is defined as the sharing of resources of many computers or system in a network to a single problem at the same time - usually to a scientific problem that needs a great number of computer processing cycles in order to access large amounts of data. Grid computing has the software which divides and farms out the pieces of a program into as many as several thousand computers. Grid computing also called as distributed and large-scale cluster computing and it is in the form of network-distributed parallel processing. Grid computing is also called partitioned computing that gives us another way of sharing the resources of the computer and yields the maximum benefit in the efficient time and speed. The partitioned computing enables multiple applications to share the computing infrastructure which results in much greater flexibility, cost, power efficiency, performance, scalability and availability. Recently, grid computing is emerging as an effective paradigm for coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations operating in the industry and business arena [4].

1.2. Grid Computing System

A Data Grid can include and provide transparent access to semantically related data resources that are different managed by different software systems and are accessible through different protocols and interfaces. A data grid computing system deals with the sharing of the controlled data and the management of large amount of partitioned data.

1.3. Data Mining Partition

Partitioned or shared data mining is defined as the extraction or analysis of data dealing with the nodes of partitioned computing environment.

2. Partitioned Data Mining and Grids

Today many organizations, companies, and scientific centers produce and manage large amounts of complex data and information. Climate data, astronomic data and company transaction data are just some examples of massive amounts of digital data repositories that today must be stored and analyzed to find useful knowledge in them. This data and information patrimony can be effectively exploited if it is used as a source to produce knowledge necessary to support decision making. This process is both computationally intensive and collaborative and partitioned in nature. In the latest years, through the Open Grid Services Architecture (OGSA), the Grid community defined Grid services as an extension of Web services for providing a standard model for using the Grid resources and composing partitioned applications as composed of several Grid services. OGSA provides an extensible set of services that virtual organizations can aggregate in various ways defines uniform exposed-service semantics, the so-called Grid service, based on concepts and technologies from both the Grid computing and Web services communities. Recently the Web Service Resource Framework (WSRF) was defined as a standard specification of Grid services for providing interoperability with standard Web services so building a bridge between the Grid and the Web. Unfortunately, high level products to support the knowledge discovery and

management in partitioned environments are lacking. This is particularly true in Grid-based knowledge discovery [4], although some research and development projects and activities in this area are going to be activated mainly in Europe and USA, such as the Knowledge Grid, the Discovery Net, and the AdAM project. Workflows are mapped on a Grid, assigning its nodes to the Grid hosts and using interconnections for communication among the workflow components (nodes). Research projects such as the TeraGrid project and the Grid Data Mining project aim at developing data mining services on Grids, whereas systems like the Knowledge Grid, Discovery Net, and Grid- Miner developed KDD systems for designing complete distributed knowledge discovery processes on grids.

3. Services of Grid

The Service Oriented Architecture (SOA) is essentially a programming model for building flexible, modular, and interoperable software applications. SOA enables the assembly of applications through parts regardless of their implementation details, deployment location, and initial objective of their development. Another principle of service oriented architectures is, in fact the reuse of software within different applications and processes. In OGSA every resource (e.g., computer, storage, and program) is represented as a Grid Service: a Web Service that conforms to a set of conventions and supports standard interfaces. OGSA defines standard mechanisms for creating, naming, and discovering transient Grid Service instances; OGSA also defines mechanisms required for creating and composing sophisticated partitioned systems, including lifetime management, change management, and notification. The WS-Resource Framework (WSRF) was recently proposed as a refactoring and evolution of Grid Services aimed at exploiting new Web Services standards, and at evolving OGSI based on early implementation and application experiences. WSRF provides the means to express state as stateful resources and codifies the relationship between Web Services and stateful resources in terms of the implied resource pattern, which is a set of conventions on Web Services technologies, in particular XML, WSDL, and WS-Addressing. OGSA provides a well-defined set of basic interfaces for the development of interoperable Grid systems and applications [5]. OGSA adopts Web Services as basic technology. Web Services are an important paradigm focusing on simple,Internet-based standards, such as the Simple ObjectAccess Protocol (SOAP) and the Web Services DescriptionLanguage (WSDL), to address heterogeneous partitioned computing. 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. A stateful resource that participates in the implied resource pattern is termed as WS- Resource. The framework describes the WS-Resource definition and association with the description of a Web Service interface, and describes how to make the properties of a WS-Resource accessible through a Web Service interface. Through WSRF is possible to define basic services for supporting partitioned data mining tasks in Grids. Those services can address all the aspects that must be considered in data mining and in knowledge discovery processes from data selection and transport to data analysis, knowledge models representation and visualization. To do this it is necessary to define services corresponding to single steps that compose a KDD process such as preprocessing, filtering, and visualization;[2] single data mining tasks such as classification, clustering, and rule discovery;[2] partitioned data mining patterns such as collective learning, parallel classification and meta-learning models.At the same time, those services should exploit other basic Grid services for data transfer and management such as Reliable File Transfer (RFT), Replica Location Service (RLS), Data Access and Integration (OGSA-DAI) and Distributed Query processing (OGSA-DQP). Finally, Grid basic mechanisms for handling security, monitoring, and scheduling distributed tasks can be used to provide efficient implementation of high-performance partitioned data analysis. In the following we described two systems that have been developed according this service-based approach to develop partitioned data mining in grids.

The Knowledge Directory Service (KDS) manages the metadata of the knowledge Grid resources and also performs the operations to search the metadata resources. The metadata managed by the KDS regard the following kind of objects:

• Repositories of data to be mined, such as databases, plain files, eXtensible Markup Language (XML) documents and other structured or unstructured data (data sources).
• Tools and algorithms used to extract filter and manipulate data (data management tools).
• Tools and algorithms used to analyze (mine) data that is data analysis tools available on the grid.
• Tools and algorithms used to visualize, store and manipulate mining results i.e. data visualization tools.
• Knowledge obtained as a result of the mining process, i.e. learned models and discovered patterns.

3.1. Grid framework

The Knowledge Grid framework is a system implemented to support the development of distributed KDD processes in a Grid [2]. It uses basic Grid mechanisms to build specific knowledge discovery services. The Knowledge Grid provides users with high-level abstractions and a set of services by which is possible to integrate Grid resources to support all the phases of the knowledge discovery process, as well as basic, related tasks like data management, data mining, and knowledge representation. In this implementation, each Knowledge Grid service (K-Grid service) is exposed as a Web Service that exports one or more operations (OPs), by using the WSRF conventions and mechanisms. The operations exported by high-level K-Grid services (data access services (DAS), tools and algorithms access services (TAAS), execution plan management services (EPMS), and result presentation services (RPS)) are designed to be invoked by user-level applications, whereas operations provided by core K-Grid services (knowledge directory services (KDS) and resource access and execution services (RAEMS)) are thought to be invoked by high-level and

core K-Grid services.

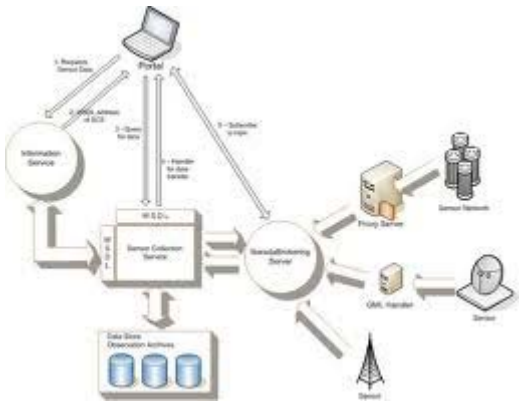

Figure 1: Client interactions in Grid environment

In the WSRF-based implementation of the Knowledge Grid each service is exposed as a Web Service that exports one or more operations (OPs), by using the WSRF conventions and mechanisms. The operations exported by High-level K-Grid services are designed to be invoked by user-level applications only, whereas the operations provided by Core K-Grid services are thought to be invoked by High-level as well as Core K-Grid services. Users can access the Knowledge Grid functionalities by using a client interface located on their machine. The client interface can be an integrated visual environment that allows for performing basic tasks (e.g., searching of data and software, data transfers, simple job executions), as well as for composing partitioned data mining applications described by arbitrarily complex execution plans.

The Knowledge Grid is a higher-level framework which is used to provide various services on Grid-based knowledge discovery tools. Those services allow the users to create and manage the complex knowledge discovery applications that in turn used to integrate the data sources and data mining tools providing the distributed services on the Grid. All of these services are currently re-programmed and re-implemented as WSRF-compliant Grid Services.

3.2. Weka4WS – A Web Service Resource Framework enabled toolkit

Weka4WS is a web service resource framework enabled toolkit supporting the partitioned or distributed data mining on the Grid environments. It provides much number of machines learning algorithms in Java for pre-processing of data, clustering, association rules, and visualization, and invoked through a common graphical user interface. In Weka, the algorithms can be executed only locally and hence the overall data mining process takes place on a single machine. The primary goal of Weka4WS is to extend Weka which supports the remote execution of data mining algorithms through WSRF Services. Distributed data mining tasks can be concurrently executed on the Grid nodes by improving the application performance and by exploiting the data distribution. In Weka4WS, the data mining algorithms for classification, clustering and association rules can be also executed on remote Grid resources. To enable remote invocation, all the data mining algorithms provided by the Weka library are exposed as a Web Service, which can be easily deployed on the available Grid nodes. Thus, Weka4WS also extends the Weka GUI to enable the invocation of the data mining algorithms that are exposed as Web Services on remote Grid nodes. .The Weka4WS software prototype has been developed by using the Java WSRF library provided by Globus Toolkit (GT4). The Weka4WS user interface is a modified Weka Explorer environment which is used to support the execution of both local and remote data mining tasks. On every computing node, a WSRF-compliant Web Service uses all the data mining algorithms provided by the Weka library. All involved Grid nodes in Weka4WS applications use the GT4 services for standard Grid fu and so on. We distinguish those nodes in two categories on the basis of the available Weka4WS components: user nodes that are the local machines providing the Weka4WS client software; and computing nodes that provide the Weka4WS Web Services allowing for the execution of remote data mining tasks. Data can be located on computing nodes, user nodes, or third-party nodes (e.g., shared data repositories). If the dataset to be mined is not available on a computing node, it can be uploaded by means of the GT4 data management services.

Figure 2 shows the software components of user nodes and computing nodes in the Weka4WS framework. User nodes include three components: Graphical User Interface (GUI), Client Module (CM), and Weka Library (WL). The GUI is an extended Weka Explorer environment that supports the execution of both local and remote data mining tasks. Local tasks are executed by directly invoking the local WL, whereas remote tasks are executed through the CM, which operates as an intermediary between the GUI and Web Services on remote computing nodes.

Figure 2: Grid Computing - User and Computing nodes components

The primary goal of the Weka4WS suite is to support the remote execution of data mining algorithms by extending the Weka through WSRF web services. In the same way, by exploiting the data distribution and by improving the application performance the tasks of the distributed data mining can be concurrently executed on decentralized Grid nodes. Through the GUI a user can either: i) start the execution locally by using the Local pane; ii) start the execution remotely by using the Remote pane. Each task

in the GUI is managed by an independent thread. Therefore, a user can start multiple distributed data mining tasks in parallel on different Web Services, this way taking full advantage of the partitioned Grid environment. Whenever the output of a data mining task has been received from a remote computing node, it is visualized in the standard Output pane. A recent paper [7] presents the architecture, details of user interface, and performance analysis of Weka4WS in executing a distributed data mining task in different network scenarios. The experimental results demonstrate the low overhead of the WSRF Web service invocation mechanisms with respect to the execution time of data mining algorithms on large data sets and the efficiency of the WSRF framework as a eans for executing data mining tasks on remote resources. Weka library as a web service enables the remote invocation and also deploys the Grid nodes. Weka GUI enables the data mining algorithms on the remote machines. The integration and the interoperability in the Grid environment is achieved by re-designing and re-developing the Weka4WS by the WSRF.

The Web services resource framework (WSRF) is acting as a standard for implementing Grid services and its applications. The framework can be used for developing high-level services for partitioned data mining applications. This paper describes Weka4WS, a framework that extends the Weka toolkit to support distributed or shared data mining on WSRF-enabled Grids. Weka4WS adopts the WSRF technology runs the remote data mining algorithms and manages the computations in distributed computing.

4. Conclusion

Data mining partitioned in the grid computing environment helps to distribute intensive analytic processing among various resources. The possibility of utilizing the grid based data mining applications attracts the organizations those analyzes the data partitioned across geographically dispersed heterogeneous platforms. The mining in the grid environment also leads to new integration and automated analysis techniques which allow the organizations to mine the data set. This overcomes the current technology of extracting and moving data into a centralized location for mining processes which is more difficult to conduct due to the fact that data is geographically dispersed.

References

[1] M. Cannataro, D. Talia, Semantics and Knowledge Grids: Building the Next Generation Grid, IEEE Intelligent Systems, 19(1), (2004), pp. 56–63.
[2] M. Cannataro, D. Talia, the Knowledge Grid, Communications of the ACM, 46(1), (2003), pp. 89–93.
[3] H. Kargupta 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).
[4] F. Berman. From TeraGrid to Knowledge Grid, Communications of the ACM, 44(11), pp. 27–28, 2001.
[5] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, the Physiology of the Grid, In: F. Berman, G. Fox, and A. Hey (eds.), Grid Computing: Making the Global Infrastructure a Reality, Wiley, pp. 217–249, (2003).
[6] M. Cannataro, A. Congiusta, C. Mastroianni, A.Pugliese, D. Talia, P. Trunfio, Grid-Based Data Mining and Knowledge Discovery, In: Intelligent Technologies for Information Analysis, N. Zhong and J. Liu (eds.), Springer-Verlag, chapt. 2 (2004), pp. 19–45.
[7] D. Talia, P. Trunfio, O. Verta. Weka4WS: a WSRFenabled Weka Toolkit for Distributed Data Mining on Grids. Proc. PKDD 2005), Porto, Portugal, October 2005, LNAI vol. 3721, pp. 309–320, Springer-Verlag, 2005.
[8] H. Witten and E. Frank. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann