Research Paper

Volume : 2 | Issue : 3 | Mar 2013 • ISSN No 2277 - 8179

Data Mining Patters in Grid Computing

Engineering

KEYWORDS: Distributed Data Mining, Grid Computing

Mr. S. Murali	Lecturer, Velammal College of Engineering & Technology, Viraganoor, Madurai – 625 009
Mrs. C. B. Selvalakshmi	Assistant Professor, Velammal College of Engineering & Technology, Viraganoor, Madurai – 625 009
Mrs. S. Padmadevi	Assistant Professor, Velammal College of Engineering & Technology, Viraganoor, Madurai – 625 009
Mr. P. N. Karthikayan	Assistant Professor, Velammal College of Engineering & Technology, Viraganoor, Madurai – 625 009

ABSTRACT

Grid computing is a way of partitioned computing focused on high-performance orientation and resource sharing. In day today computing environment, performing analysis of very large data sets is necessary in many applications. The data are geographically partitioned and the complexity of the data is increasing and in those cases the grid technologies is used to provide an effective computational support for knowledge discovery. This paper is an introduction to the Grid computing environment.

I. INTRODUCTION

A. Grid Computing

A multiple or simultaneous processing architecture that shares the CPU resources across a network, and all the systems in the network functions as a large supercomputer allows the unused CPU capacity for all participating machines to be allocated that is extremely computation intensive and programmed for 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].

B. Data Grid Computing System

A data grid computing system deals with sharing and managing the large amount of partitioned data. This system provides a transparent access to all shared data resources that are managed by different software systems and this system can access by using different protocols and interfaces.

C. Partitioned Data Mining

Partitioned or shared data mining deals with the problem of data analysis in environments of partitioned computing nodes.

II. 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. 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. 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. 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.

III. PARTITIONED DATA MINING GRID SERVICES

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. 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 Object Access Protocol (SOAP) and the Web Services Description Language (WSDL), to address heterogeneous partitioned computing. In OGSA every resource (e.g., computer, storage, 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. Through WSRF is possible to define basic services for supporting partitioned data mining tasks in Grids. 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.

A. The framework for the Knowledge Grid

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. 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.

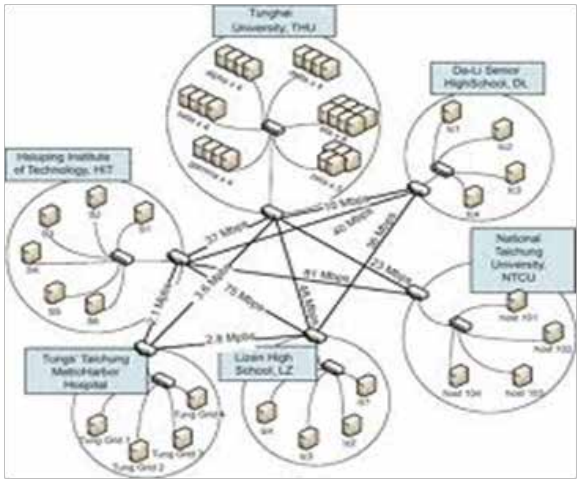

Figure 1: Client and Grid environment interactions

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. The client interface performs its tasks by invoking the appropriate operations provided by the different High-level K-Grid services. Those services may be in general executed on a different Grid node; therefore the interactions between the client interface and High-level K-Grid services are possibly remote.

B. Weka4WS

Weka4WS is a framework that extends the widely used open source Weka toolkit for supporting distributed data mining on WSRF-enabled Grids. Weka4WS adopts the WSRF technology for running remote data mining algorithms and managing partitioned computations. The Weka4WS user interface supports the execution of both local and remote data mining tasks. On every computing node, a WSRF compliant 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. Data can be located on computing nodes, user nodes, or third-party nodes (e.g., shared data repositories).

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.

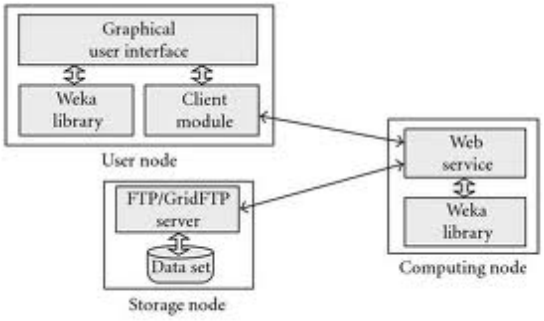

Figure 2: Components of user and Computing nodes in Grid Computing

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 means for executing data mining tasks on remote resources.

IV. CONCLUSION

Data mining patterns shared in the grid computing environment is highly useful for the companies to distribute the data and also for the effective analysis among different resources. In this world the data are geographically dispersed in various places and hence this shared data mining patterns overcomes the data extraction and also it avoids the moving of data into a centralized location for mining processes. The shared data mining patterns in the grid environment leads to a way of new integration and automated analysis techniques.

REFERENCE

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] Witten and E. Frank. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann