Distributed Data Mining in the Grid Environment

C. B. SelvaLakshmi, S. Murali, P. Chanthiya and P. N. Karthikayan

Abstract Grid computing has emerged as an important new branch of distributed computing focused on large-scale resource sharing and high-performance orientation. In many applications, it is necessary to perform the analysis of very large data sets. The data are often large, geographically distributed and its complexity is increasing. In these areas, grid technologies provide effective computational support for applications such as knowledge discovery. This paper is an introduction to grid infrastructure and its potential for machine learning tasks.

Keywords Grid computing · Knowledge grid · Data mining · Distributed data mining

Introduction

Grid Computing

A parallel processing architecture in which CPU resources are shared across a network, and all machines function as one large supercomputer, it allows unused CPU capacity in all participating machines to be allocated to one application that is

C. B. SelvaLakshmi (✉) · S. Murali · P. Chanthiya · P. N. Karthikayan
Velammal College of Engineering and Technology, Madurai, India
e-mail: cbselak08@gmail.com

S. Murali
e-mail: muralicse2008@gmail.com

P. Chanthiya
e-mail: chanthiyapuhal@gmail.com

P. N. Karthikayan
e-mail: karthikayan.it@gmail.com

S. Sathiakumar et al. (eds.), *Proceedings of International Conference on Internet Computing and Information Communications*, Advances in Intelligent Systems and Computing 216,
DOI: 10.1007/978-81-322-1299-7_15,

extremely computation intensive and programmed for parallel processing. Grid computing is also called peer to peer computing and distributed computing. The grid computing gives us yet another way of sharing the computer resource and yields us the maximum benefit at the time and speed efficiency. Grid computing enables multiple applications to share computing infrastructure, resulting in much greater flexibility,cost, power efficiency, performance, scalability and availability at the same time.

Data Grid

A data grid is a grid computing system that deals with the data controlled sharing and management of large amount of distributed data. 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.

Distributed Data Mining

Distributed data mining deals with the problem of data analysis in environments of distributed computing nodes.

Distributed 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 is particularly true in grid-based knowledge discovery [1], 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 (K-Grid), the Discovery Net, and the AdAM project. In particular, the K-Grid [2] that we shortly discuss in the next section provides a middleware for knowledge discovery services for a wide range of high performance distributed applications. Examples of large and distributed data sets available today include gene and protein databases, network access and intrusion data, drug features and effects data repositories, astronomy data files, and data about web usage, content, and structure. Workflows are mapped on a grid, assigning its nodes to the grid hosts and using interconnections for communication among the workflow components (nodes). 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 distributed 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.

Grid Services for Distributed Data Mining

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 SOAs is, in fact the reuse of software within different applications and processes. The grid community adopted the OGSA as an implementation of the SOA model within the grid context. OGSA provides a well-defined set of basic interfaces for the development of interoperable grid systems and applications [3]. 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 distributed computing. Web service defines techniques for describing software components to be accessed, methods for accessing these components, and discovery mechanisms that enable the identification of relevant service providers. OGSA defines standard mechanisms for creating, naming, and discovering transient grid service instances. 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 [4]. 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. 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 distributed 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] distributed 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). Moreover, distributed data mining algorithms can optimize the exchange of data needed to develop global knowledge models based on concurrent mining of remote data sets. This approach also preserves privacy and prevents disclosure of data beyond the original sources. Finally, grid basic mechanisms for handling security, monitoring, and scheduling distributed tasks can be used to provide efficient implementation of high-performance distributed data analysis.

The K-Grid Framework

The K-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. These services can be developed in different ways using the available grid environments. The K-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 K-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. Fig. 1.

In the WSRF-based implementation of the K-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 K-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 distributed data mining applications described by arbitrarily complex execution plans. The client interface performs its tasks by invoking the appropriate

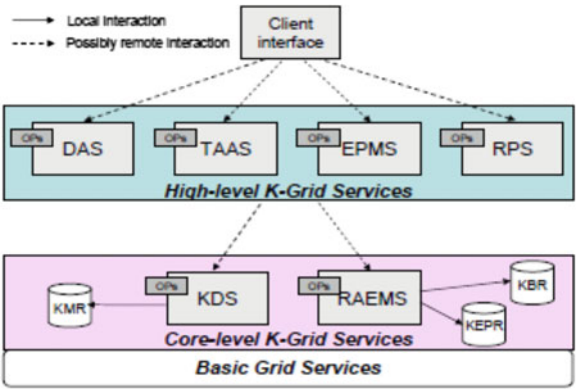

Fig. 1 Interactions between a client and the knowledge grid environment

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.

Weka4WS

Weka4WS is a framework that extends the widely used open source Weka toolkit [5] for supporting distributed data mining on WSRF-enabled grids. Weka4WS adopts the WSRF technology for running remote data mining algorithms and managing distributed computations. The Weka4WS user interface supports the execution of both local and remote data mining tasks. On every computing node, a WSRF compliant Web Service is used to expose all the data mining algorithms provided by the Weka library. The Weka4WS software prototype has been developed by using the Java WSRF library provided by Globus Toolkit (GT4). 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.

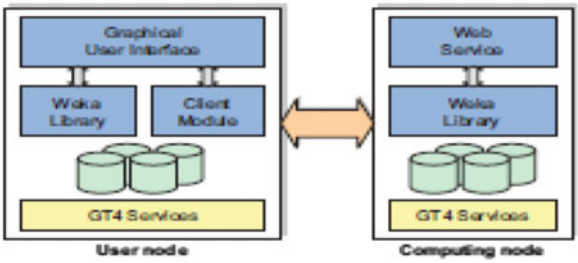

Fig. 2 Software components of user nodes and computing nodes

Through the GUI a user can either: (1) start the execution locally by using the Local pane; (2) 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 distributed 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 [6] 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. By exploiting such mechanisms, Weka4WS provides an effective way to perform compute-intensive distributed data analysis on large-scale grid environments. Weka4WS can be downloaded from http://grid.deis.unical.it/weka4ws.

Conclusion

The development of practical grid computing techniques will have a profound impact on the way data is analyzed. In particular, the possibility of utilizing grid-based data mining applications is very appealing to organizations wanting to analyze data distributed across geographically dispersed heterogeneous platforms. Grid-based data mining would allow companies to distribute compute intensive analytic processing among different resources. Moreover, it might eventually lead to new integration and automated analysis techniques that would allow companies to mine data where it resides. This is in contrast to the current practice of having to extract and move data into a centralized location for mining processes that are becoming more difficult to conduct due to the fact that data is becoming increasingly geographically dispersed, and because of security and privacy considerations.

References

1. Berman, F.: From Teragrid to Knowledge Grid. Commun. ACM. **44**(11), 27–28 (2001)
2. Cannataro, M., Talia, D.: The knowledge grid. Commun. ACM. **46**(1), 89–93 (2003)
3. Foster, I., Kesselman, C., Nick, J., Tuecke, S. The physiology of the grid. In: Berman, F., Fox, G., Hey, A. (eds.) Grid Computing: Making the Global Infrastructure a Reality, pp. 217–249. Wiley (2003)
4. Czajkowski, K. et al. The WS-Resource Framework Version 1.0. http://www-106.ibm.com/developerworks/library/ws-resource/wswsrf.pdf
5. Witten, H., Frank, E. Data Mining: Practical Machine Learning Tools with Java Implementations. Morgan Kaufmann
6. Talia, D., Trunfio, P., Verta, O. Weka4WS: a WSRFenabled Weka toolkit for distributed data mining on grids. In Proceedings of PKDD 2005, LNAI vol. 3721, pp. 309–320. Springer-Verlag, Porto, Portugal, October (2005)
7. Cannataro, M., Congiusta, A., Mastroianni, C., Pugliese, A., Talia, D., Trunfio, P.: Grid-based data mining and knowledge discovery. In: Zhong, N., Liu, J. (eds.) Intelligent Technologies for Information Analysis, pp. 19–45. Springer-Verlag, (2004)
8. Cannataro, M., Talia, D.: Semantics and knowledge grids: building the next generation grid. IEEE Intell. Syst. **19**(1), 56–63 (2004)
9. Kargupta, H., Kamath, C., Chan, P. Distributed and parallel data mining: emergence, growth, and future directions, In: Advances in Distributed and Parallel Knowledge Discovery, pp. 409–416. AAAI/MIT Press (2000)