International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-1 Number-1 Issue-1 September 2011

Exploiting need of Service-Oriented Framework for Executing Data Mining Services

Sonali Jain[1], Niket Bhargava[2]
M.Tech Scholar, BIST Bhopal, India[1]
Assistant Professor, BIST Bhopal, India[2]

Abstract

Weka4WS adopts the emerging Web Services Resource Framework (WSRF) for accessing remote data mining algorithms and managing distributed computations. The Weka4WS user interface is a modified Weka Explorer environment that supports the execution of both local and remote data mining tasks. Workflow environments are widely used in data mining systems to manage data and execution flows associated to complex applications. Weka, one of the most used open-source data mining systems, includes the Knowledge-Flow environment which provides a drag-and-drop inter-face to compose and execute data mining workflows. It allows users to execute a whole workflow only on a single compute on the basis of simplicity. We analyzes several aspects of distributed workflow execution in Weka4WS, a framework that extends Weka and its Knowledge Flow environment to exploit distributed resources available in a Grid using Web Service technologies and also some other workflows and design which is better in efficiency and work. We also discuss several architecture prospective for betterment.

Keywords

Data Mining, Weka, Grid, Workflow

1. Introduction

Modern scientific collaborations require large-scale data mining and integration (DMI) processes [1]. Their investigations involve multi-disciplinary expertise and large-scale computational experiments on top of large amounts of data that are located in distributed data repositories running various software systems, and managed by different organizations [2].Data mining technology can analyze massive data. Although it plays vital role in many domains, if it is used improperly it can also cause some new problem of information security. There are some new problems in the application of data mining recently.
Over the past years the Grid has attracted great attention due to its ability to pool heterogeneous, distributed resources, not necessarily designed to work together, into an integrated environment offering a wide set of services and capabilities. Grids are successfully used in, e.g., distributed collaborative researches and a large enterprise with complex computational needs.The community is experiencing an even more in-depth discovery of new research areas, applications and challenges. In several cases the Grid has shown that it is not always feasible to understand some needs, identify an already existing non-Grid solution and simply apply it to the grid context.

Situations within certain composite service applications often invoke high numbers of requests due to heightened interest from various users. In a recent, real-world example of this so called query-intensive phenomenon, the catastrophic earthquake in Haiti generated massive amounts of concern and activity from the general public. This abrupt rise in interest prompted the development of several Web services in response, offering on demand retagged maps of the disaster area to help guide relief efforts. Similarly, efforts were initiated to collect real-time images of the area, which are then composed together piecemeal by services in order to capture more holistic views. But due to their popularity, the availability of such services becomes an issue during this critical time.

The Weka Knowledge Flow allows users to execute a complete workflow only on a single machine. On the other hand, most knowledge flows include several independent branches that could be run in parallel on a set of distributed machines to reduce the overall execution time. The Grid facilities [3] are exploited by Weka4WS because it provides a set of services to access distributed computing nodes, which can be effectively used to run complex and resource-demanding data mining applications. In particular, Weka4WS adopts a service-oriented architecture in which Grid nodes expose a wide set of data mining algorithms as Web Services, and client applications can invoke them to run distributed data mining applications defined as workflows.

Process view is also important in terms of executing data mining services on grids. Process views have several purposes. One purpose is information filtering. Particular artifacts, activities, or whole structures in a process are not essential during particular tasks related to process management. They can therefore be neglected in those situations. For example, activities in a process which run fully

automated can be faded out during the performance of staff related tasks. Filtering information reduces the overall complexity of a process. Another purpose of process viewing is information summarization. A filter removes information. In contrast to that, a summarization makes it more compact by aggregating structures. Besides, process views can also support the translation of information.

We provide here an overview of executing data mining services on grid. The rest of this paper is arranged as follows: Section 2 introduces Grid Services; Section 3 describes about Weka4WS; Section 4 shows the evolution and recent scenario; Section 5 describes the challenges. Section 6 describes conclusion and outlook.

2. Grid Services

The third and latest version of the Globus Toolkit is based on something called Grid Services. Before defining Grid Services, we're going to see how Grid Services are related to a lot of acronyms you've probably heard (OGSA, OGSI ...), but aren't quite sure what they mean exactly. The following diagram summarizes the major players in the Grid Services world: The working process is shown in fig1.

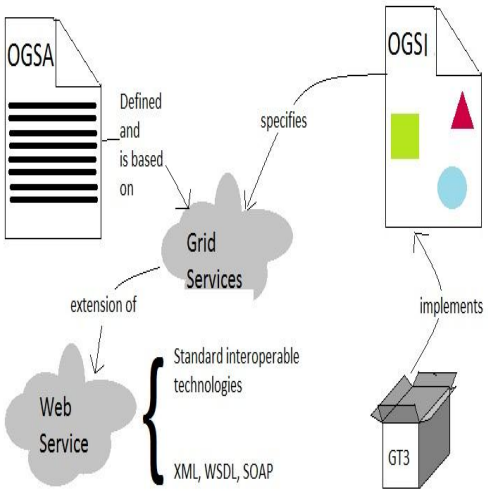

Fig 1 Working Process of Grid Services

The Open Grid Services Architecture (OGSA), developed by The Global Grid Forum, aims to define a common, standard, and open architecture for grid-based applications. The goal of OGSA is to standardize practically all the services one finds in a grid application (job management services, resource management services, security services, etc.) by specifying a set of standard interfaces for these services.

However, when the powers-that-be undertook the task of creating this new architecture, they realized they needed to choose some sort of distributed middleware on which to base the architecture. In other words, if OGSA (for example) defines that the Job Submission Interface has a submit Job method, there has to be a common and standard way to invoke that method if we want the architecture to be adopted as an industry-wide standard. This base for the architecture could, in theory, be any distributed middleware (CORBA, RMI, or even traditional RPC). For reasons that will be explained further on, Web Services were chosen as the underlying technology.

However, although the Web Services Architecture was certainly the best option, it still had several shortcomings which made it inadequate for OGSA's needs. OGSA overcame this obstacle by defining an extended type of Web Service called Grid Service (as shown in the diagram: Grid Services are defined by OGSA). A Grid Service is simply a Web Service with a lot of extensions that make it adequate for a grid-based application (and, in particular, for OGSA). In the diagram: Grid Services are an extension of Web Services. Finally, since Grid Services are going to be the distributed technology underlying OGSA, it is also correct to say that OGSA is based on Grid Services.

The Globus Toolkit is a software toolkit, developed by The Globus Alliance, which we can use to program grid-based applications. The third version of the toolkit (GT3) includes a complete implementation of OGSI (in the diagram GT3 implements OGSI). However, it's very important to understand that GT3 isn't only an OGSI implementation. It includes a whole lot of other services, programs, utilities, etc. Some of them are built on top of OGSI and are called the WS (Web Services) components, while other are not built on top of OGSI and are called the pre-WS components.

3. Weka4WS

Weka4WS is a framework developed at the University of Calabria to extend the widely used Weka toolkit for supporting distributed data mining on Grid environments. Weka provides a large collection of machine learning algorithms written in Java for data pre-processing, classification, clustering, association rules, and visualization, which can be invoked through a common graphical user interface. In Weka, the overall data mining process takes place on a single machine, since the algorithms can be executed only locally.

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.

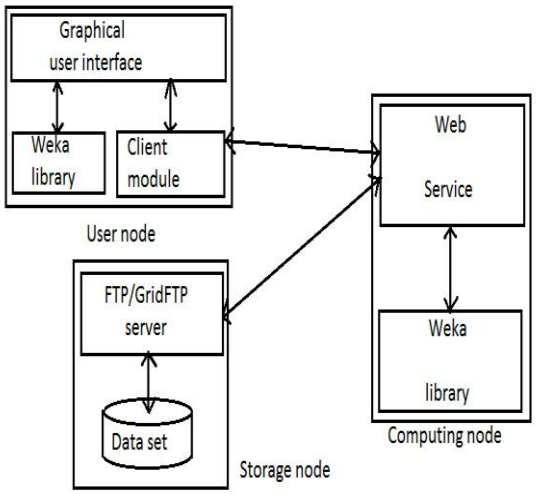

Fig 2 Weka4ws

Through the GUI a user can both:

 start the execution locally by using the standard Local pane.
 start the execution remotely by using the Remote pane.
 each task in the GUI is managed by an independent thread in an asynchronous way.

A user can start multiple 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.

4. Evolution and Recent Scenario

In 2006, C. Pautasso et al. [4] proposed about parallelism that could be effectively exploited in data mining workflows is data parallelism , where a large data set is split into smaller chunks, each chunk is processed in parallel, and the results of each processing are then combined to produce a single result.

In 2007,D Talia et al. [5] proposed about The Knowledge Grid which is another service-oriented system supporting distributed data mining workflow execution.LikeWeka4WS, it uses WSRF as enabling technology.UnlikeWeka4WS, which extends an already existing workflow system the Knowledge Grid defines its own workflow formalism and provides a set of services to support the workflow execution.
A service-oriented approach similar to that ofWeka4WS is adopted by FAEHIM [6], which exposes a set of data mining algorithms as Web Services. However, differently from Weka4WS, FAEHIM does not provide a work flow system of its own, but relies on Triana [7] for composing data mining services as workflows. Moreover, the FAEHIM services are not based on the WSRF technology.

Integrating agents and service-oriented architectures has been attempted before. Moreau [8] has given detailed comparisons between these two approaches. The research focuses on agents that are able to describe their operations as services and to search and adopt other services by using mappings between agent and service concepts. Such approaches are often based on a proxy that bridges one set of concepts to another.

In 2009, Bin Cao et al. [9] proposed about Karma which is a tool that collects and manages provenance data. Karma has a modular architecture that supports multiple types of data sources for provenance data. Karma can listen to notifications on a messenger bus or receive messages synchronously and process the notifications to determine provenance information.

Workflow engines [10] are used for representing task dependencies and controlling execution. Generic Application Factory (GFac)[11] and Opal toolkit provide tools to wrap legacy scientific application codes as web services. The wrapper handles grid security and interaction with other grid services for file transfer and job submission. However the execution logic state for each application has to be managed individually and there is no easy way to abstract out, customize and reuse policies or code (e.g., provenance instrumentation) across implementations. This is very fruitful in terms of accuracy and efficiency in terms of traditional approaches.

GridSim [12] and CloudSim [13] provide a simulator framework of grid and cloud resources enabling modeling of large grid and cloud resources. Simgrid [14] is a simulation toolkit that enables the study of scheduling algorithms for distributed applications. Mumak is a Hadoop based simulator that can be used with the real job and task trackers to simulate execution on thousands of nodes for testing and debugging. These simulators represent and however

these tools do not reflect application level execution intricacies that require extensive testing.

In 2010, David Schumm et al. [15] proposed about process views which is technology independent and can be applied to any process language which can be represented by a process graph, such as the Business Process Modeling Notation (BPMN) and Event-driven Process Chains (EPC).

In 2010, Tobias Pontz et al. [16] proposed about an IT infrastructure based on service and grid computing technology. Additionally, a virtual value creation chain has been introduced to integrate virtual prototyping methodologies. The current contribution elaborates the importance of differentiating, defining and managing both value and knowledge flows in such a virtual value creation chain. Consequently, a service-oriented knowledge management system is envisaged by describing tasks of a knowledge manager and deducing a solution concept.

In 2010, Alexander Wöhrer et al. [17] proposed about rationale, theory, design and application of logical optimization of data flows for data mining and integration processes. A dataflow model is defined and several optimization algorithms, namely dead elements elimination, process re-ordering, parallelization, and data by-passing are developed. The first research prototype of the framework has been implemented in the context of the ADMIRE Data Mining and Integration Process Designer for logical optimization of specifications expressed in the DISPEL language developed in the ADMIRE project.

In 2010, David Chiu et al. [18] proposed an approach to accelerate service processing in a Cloud setting. We have developed a cooperative scheme for caching data output from services for reuse. They propose an algorithm for scaling our cache system up during peak querying times, and back down to save costs. Using the Amazon EC2 public Cloud, a detailed evaluation of our system has been performed, considering speed up and elastic scalability in terms resource allocation and relaxation.

5. Challenges

Evaluating the execution times of the different steps needed to perform a typical data mining task in different network scenarios

- Evaluating the efficiency of the WSRF mechanisms and Weka4WS as methods to execute distributed data mining services.
- We used 10 datasets extracted from the census dataset available at the UCI repository:
- Number of instances: from 1700 to 17000
- Dataset size: from 0.5 to 5 MB
- Weka4WS has been used to perform a clustering analysis on each of these datasets:
- Algorithm used: Expectation Maximization (EM)
- Number of clusters to be identified: 10

6. Conclusion and Outlook

Production planning is an important process in customer supplier interaction and can be supported by sophisticated and knowledge-intensive virtual prototyping methodologies arranged in a virtual value creation chain. Apart from original results (i.e., value flow), specific knowledge has to be determined, managed, and exchanged along the execution of this chain.

In addition, we also concentrate on a SOA based workflow for an intelligent multi-agent system can work seamlessly together despite being functionally independent of each other.

Cloud providers have begun offering users at-cost access to on demand computing infrastructures. We also discuss about a Cloud-based cooperative cache system for reducing execution times of data-intensive processes. The resource allocation algorithm presented herein is cost-conscious as not to over-provision Cloud resources. We have evaluated our system extensively, showing that, among other things, our system is scalable to varying high workloads.

References

[1] T. Hey and A. Trefethen, “Cyberinfrastructure for e-science,” Science Magazine, vol. 308, no. 5723, pp. 817–821, 2005.
[2] J. Gray, D. T. Liu, M. Nieto-Santisteban, A. Szalay, D. J. DeWitt, and G. Heber, “Scientific data management in the coming decade,” SIGMOD Rec., vol. 34, no. 4, pp. 34–41, 2005.
[3] I. Foster, C. Kesselman, J. Nick, S. Tuecke. The Physiology of the Grid. In: F. Berman, G. Fox, A. Hey (Eds.) Grid Computing: Making the Global Infrastructure a Reality, Wiley: 217-249, 2003.
[4] C. Pautasso, G. Alonso, Parallel Computing Patterns for Grid Workflows, Workshop on Workflows in Support of Large-Scale Science, 2006.
[5] A. Congiusta, D. Talia, P. Trunfio. Distributed data mining services leveraging WSRF. Future Generation Computer Systems, 23(1):34-41, 2007.
[6] A. Ali Shaikh, O. F. Rana, I. J. Taylor. Web Services Composition for Distributed Data Mining. Workshop on Web and Grid Services for Scientific Data Analysis, 2005.

[7] I. Taylor, M. Shields, I. Wang, A. Harrison. The Tri-ana Workflow Environment: Architecture and Applications Workflows for e-Science, Springer: 320-339, 2007.

[8] I.J. Taylor et al., eds. Workflows for e-Science: Scientific Workflows for Grids, Springer-Verlag, 2006.

[9] Bin Cao, Beth Plale, Girish Subramanian, Ed Robertson, Yogesh Simmhan, "Provenance Information Model of Karma Version 3,"Services, IEEE Congress on, pp. 348-351, 2009 Congress on Services - I, 2009.

[10] D. Leake and K.-M. Joseph, Towards Case-Based Support for e-Science Workflow Generation by Mining Provenance, in Proc of the 9th European conference on Advances in Case-Based Reasoning. 2008.

[11] S. Krishnan et al. Design and Evaluation of Opal2: A Toolkit for Scientific Software as a Service. IEEE Congress on Services (SERVICES-1 2009), July, 2009.

[12] R. Buyya and M. Murshed, GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, The J. of Concurrency and Computation: Practice and Experience, Nov.-Dec., 2002.

[13] R. N. Calheiros, R. Ranjan, C. A. F. De Rose,and R. Buyya, CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, Australia, March, 2009.

[14] H. Casanova, Simgrid: A toolkit for the simulation of application scheduling.IEEE/ACM International Symposium on Cluster Computing and the Grid May, 2001.

[15] David Schumm, Tobias Anstett, Frank Leymann, Daniel Schleicher, 14th IEEE International Enterprise Distributed Object Computing Conference Workshops, IEEE 2010.

[16] Tobias Pontz, Manfred Grauer, Daniel Metz, Sachin Karadgi, 3rd International Conference on Information Management, Innovation Management and Industrial Engineering, 2010, IEEE.

[17] Alexander Wöhrer, Eduard Mehofer and Peter Brezany, 2010 Sixth IEEE International Conference on e–Science Workshops.

[18] David Chiu, Apeksha Shetty and Gagan Agrawal, SC10 November 2010, New Orleans, Louisiana, IEEE.