Volume 3, Issue 3, March 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

# Analysis and Design of Service-Oriented Framework for Executing Data Mining Services on Grids

**Pooja Sharma**
Dept. of Computer Science & Engg
SVITS, Indore, India

**Anand Rajavat**
Dept. of Computer Science & Engg
SVITS, Indore, India

***Abstract—Data mining services on grids is the need of today's era. 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 computer on the basis of simplicity. There are several workflows in today's scene. Most data mining workflows include several independent branches that could be run in parallel on a set of distributed machines to reduce the overall execution time. 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 architectures prospective for betterment.***

***Keywords – Data Mining, Weka, Grid, Workflow***

## I. 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 large enterprises 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.

## II. Grid Services

Grid computing concerns the application of the resources of many computers in a network to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. Grid computing requires the use of software that can divide and farm out pieces of a program to as many as several thousand computers. Grid computing can be thought of as distributed and large-scale cluster computing and as a form of network-distributed parallel processing. It can be confined to the network of computer workstations within a corporation or it can be a public collaboration (in which case it is also sometimes known as a form of peer-to-peer computing). A number of corporations, professional groups, university consortiums, and other groups have developed or are developing frameworks and software for managing grid computing projects. The European Community (EU) is sponsoring a project for a grid for high-energy physics, earth observation, and biology applications. In the United States, the National Technology Grid is prototyping a computational grid for infrastructure and an access grid for people. Sun Microsystems offers Grid Engine software. Described as a distributed resource management (DRM) tool, Grid Engine allows engineers to pool the computer cycles on up to hundreds of workstations at a time. (At this scale, grid computing can be seen as a more extreme case of load balancing.) Grid computing appears to be a promising trend for three reasons: (1) its ability to make more cost-effective use of a given amount of computer resources, (2) as a way to solve problems that can't be approached without an enormous amount of computing power, and (3) because it suggests that the resources of many computers can be cooperatively and perhaps synergistically harnessed and managed as a collaboration toward a common objective. In some grid computing systems, the computers may collaborate rather than being directed by one managing computer. One likely area for the use of grid computing will be pervasive computing applications - those in which computers pervade our environment without our necessary awareness. In 2010 Ashutosh Dubey et al. [4] concerns about the main issues handled in Grid computing systems is resource in dissimilar fashion and model to incorporate them. There are a number of challenging issues when taking mobile environment into account, such as intermittent connectivity, device heterogeneity, Weak access security, Computation behavior, large data size, and range and device mobility.

The working process is shown in fig1.

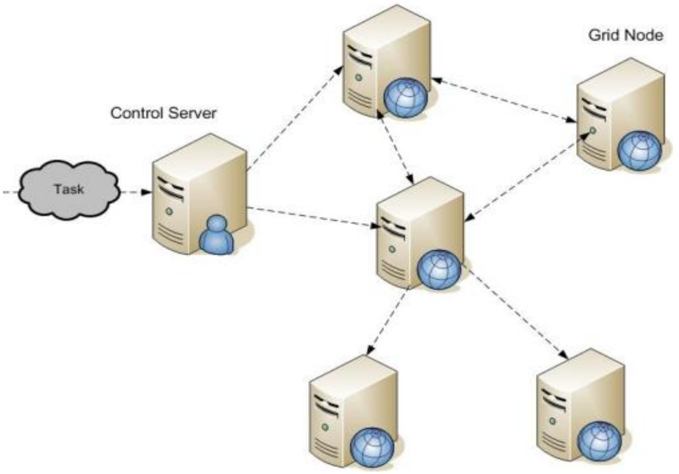

Fig 1 Working Process of Grid Services

## III. 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. The goal of Weka4WS is to extend Weka to support remote execution of the data mining algorithms through WSRF Web Services. In such a way, distributed data mining tasks can be concurrently executed on decentralized Grid nodes by exploiting data distribution and improving application performance. 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. To achieve integration and interoperability with standard Grid environments, Weka4WS has been designed by using the Web Services Resource Framework (WSRF) as enabling technology. In particular, Weka4WS has been developed by using the WSRF Java library provided by Globus Toolkit 4.0.x (GT4).

In the Weka4WS framework all nodes use the GT4 services for standard Grid functionalities, such as security and data management. Those nodes can be distinguished in two categories:

1. User nodes, which are the local machines of the users providing the Weka4WS client software;
2. Computing nodes, which provide the Weka4WS Web Services allowing the execution of remote data mining tasks.

Weka4WS is therefore distributed in two separated packages:

1. Weka4WS-client, which contains the client software (including the extended Weka GUI) to be installed on the user nodes;
2. Weka4WS-service, which contains the WSRF-compliant Web Services to be installed on the computing nodes.

**Computing nodes:**

As 'root' user, perform the following step:

1. add the following line to the file /etc/sudoers:
   ```
   globus ALL= NOPASSWD: /bin/ls, /bin/cp, /bin/mkdir, /bin/chown, /bin/gzip
   ```

As 'globus' user (or alternatively as user which runs the globus container), download the Weka4WS-service package in a directory (for example in its home directory), and perform the following steps:

1. Extract the Weka4WS-service package:
   ```
   tar xzvf weka4ws-service-2.1.tgz
   ```
2. Enter the just created directory:
   ```
   cd ./weka4ws-service-2.1
   ```
3. Generate the Weka4WS GAR file running the command:
   ```
   ./build.sh
   ```
4. Deploy the Weka4WS service running the command:
   ```
   ./deploy.sh
   ```

## IV. Evolution and Recent Scenario

The state of the data repository is the union of the states of its constituent data products. It helps to consider workflows as acting on a state machine, taking the data products from one state to another. The data valets need to know the status of the repository at any point in time, to see if the pipeline is healthy and all data products are being surfaced to the users.

In 1993, M. Hsu,et al.[6] proposed about Modeling the results of the workflows as nodes of a state machine enables simple discovery of that status. The nodes of this state machine refer to the status of the data being manipulated and the transition edges are workflows or activities that operate on the data to take them to one of the subsequent states[5,7].

In 1998, Bardram et al.[8] proposed about Knowledge sharing which is just one kind of cooperative activity the MK platform was demanded to support. The other categories, according to Bardram classification, are: organization of work, planning and scheduling, and communication, with the general objective of creating new knowledge. In the first phase of the project we explicitly detailed these activities, with particular emphasis on the identification of different roles for human actors .Role-Based Collaboration (RBC) theory is a natural approach to integrate the theory of roles into the CSCW systems [9,10].

In 2006, C. Pautasso et al. [11] 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. [12] 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 Weka KnowledgeFlow), 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 [13], which exposes a set of data mining algorithms as Web Services. However, differently from Weka4WS, FAEHIM does not provide a workflowsystem of its own, but relies on Triana [14] 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[15] 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. [16] 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 [17] are used for representing task dependencies and controlling execution. Generic Application Factory (GFac)[18] 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 (e.g.,resource selection) or code (e.g., provenance instrumentation) across implementations.This is very fruitful in terms of accuracy and efficiency in terms of traditional approaches.

GridSim [19] and CloudSim [20] provide a simulator framework of grid and cloud resources enabling modeling of large grid and cloud resources.Simgrid [21] 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.[22] 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. [23] 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.[24] proposed about rationale, theory, design and application of logical optimization of dataflows 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. [25] 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.

## V. CHALLENGES

First of all, when ArguBroker sends an abstract workflow to GOLEM, housekeeping information such as how the workflow is displayed/presented to user in KDE and how the monitoring of workflow execution is carried out, are removed. Secondly and also the most importantly, even with the above information filled in, the concrete workflows returned by GOLEM that contain real services execution information and the information of temporal time orders between them, in many cases are still not directly executable due to the missing shimming operations. A shimming operation is an intermediate operation that bridges gaps between two services.

The workflow emulator supports a trace mode where information about start times can be initialized and used to generate appropriate workflow completion time. As soon as a service provider or service integrator accepts a required service, he is able to request specific knowledge for performing his offered service via a knowledge support system as part of the knowledge management system. Knowledge requests are stored as knowledge requirements in a knowledge repository. They are reused by modeling the knowledge flows with KMDL.

## VI. 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] Ashutosh Dubey and Shishir Shandilya, 2010 5th International Conference on Industrial and Information Systems, ICIIS, India, IEEE 2010.

[5] P. Yang, “Formal Modeling and Analysis of Scientific Workflows Using Hierarchical State Machines,” Sci. Work. and Bus. Work. Stds. in e-Sci. Workshop (SWBES), 2007

[6] M. Hsu, R. Obermarck, R. Vuurboom, “Workflow Model and Execution,” Data Engg. Bulletin, vol. 16(2), 1993.

[7] R. Duan, R. Prodan, T. Fahringer, “DEE: A Distributed Fault Tolerant Workflow Enactment Engine,” Lect. Notes in Comp. Sci. (LNCS), vol 3726, 2005.

[8] Bardram, J. E.: Collaboration, Coordination, and Computer Support, An Activity Theoretical Approach to the Design of Computer Supported Cooperative Work. PhD Thesis, University of Aarhus, Denmark (1998).

[9] Edwards, W.K.: Policies and Roles in Collaborative Applications. In: ACM Conference on Computer-Supported Cooperative Work Cambridge, USA (1996).

[10] Guzdial, M., Rick, J., and Kerimbaev, B.: Recognizing and Supporting Roles in CSCW. In: ACM Conference on Computer-Supported Cooperative Work (CSCW’00), pp.261--268. Philadelphia, Pennsylvania, USA (2000).

[11] C. Pautasso, G. Alonso, Parallel Computing Patterns for Grid Workflows, Workshop on Workflows in Support of Large-Scale Science, 2006.

[12] A. Congiusta, D. Talia, P. Trunfio. Distributed data mining services leveraging WSRF. Future Generation Computer Systems, 23(1):34-41, 2007.

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

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

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

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

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