Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Open Service Framework Mining (OSFM) for Executing Data Mining Tasks

Pooja Sharma, Anand Rajavat
Department of Computer Science and Engineering,
SVITS, Indore, M.P. 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 compute 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. In this paper we proposed a novel Open Service Framework Mining (OSFM) for executing data mining tasks. Our algorithm contains five phases 1) Authentication 2) Reading Database3) Define the minimum support 4) Subset Find 5) Prune phase. Finally our algorithm shows better performance showing the simulation result.

Keywords – Data Mining, OFSM, Prune, Open Framework

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 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 to 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 several data mining services. The rest of this paper is arranged as follows:

Section 2 introduces Open Service Framework; Section 3 describes about recent scenario; Section 4 shows the Proposed Algorithm; Section 5 shows the simulation result. Section 6 describes Conclusion and outlook.

2. Open service Framework

Existing storage systems are usually custom-built and custom-tuned to offer both scalability and good performance at high cost. Such systems are usually extended with information and data lifecycle management tools that strive to manage data continuously and based on their use. For instance, hierarchical storage management systems, move data between first and second (or third) tier storage, in an effort to optimize capacity, performance, and cost.

Although such (commercial and open) tools are continuously being extended, they target existing storage systems and architectures, and mainly high-end storage systems. The work in this task focuses on how future storage architectures can both rely on commodity components as well as embed in the architecture mechanisms that will assist with data managements, rather than providing management as an add-on feature of the system. In some sense, existing solutions provide (and bind) mechanisms and policies in 3rd-party solutions, whereas research aims at separating mechanisms from policies and making mechanisms part of the system itself.

Building large installations from low-cost commodity components would make data Grids less expensive and enable them to closely track the latest technological advancements of inexpensive mass-produced storage devices. Recent advancements in commodity interconnection network technologies and the continuing reduction in disk prices present new opportunities to address these issues. Various projects [46] currently strive to address the following issues: ² Build scalable storage systems that can hold peta bytes of storage in a cost-effective manner ² Make the storage infrastructure location-independent and client-agnostic in an efficient manner. ² Provide benchmarking methodologies. Building large-scale distributed data storage systems faces the problem of storage resource virtualization incompatibility. The incompatibility results from the different levels of abstraction in the resources virtualization and the lack of a single, common framework for describing the storage services offered by the virtualized resources. Such a unified method of describing the services would make it possible to interface the different types of storage services in an effective way and would offer the starting point for building large-scale storage infrastructures. Many initiatives develop techniques for virtualizing the resources. For instance, initiatives such as Lustre, GPFS, Frangipani and Petal etc. provide a virtual volume abstraction by distributing blocks to many storage nodes. Projects, such as Storage Resource Broker aim at the virtualization of files, file repositories and similar structures. P2P systems such as Gnutella, Kazaa etc. use yet another level of abstraction of the virtualized data storage resources, providing access to files on the basis of metadata e.g. the song title or artist name. Unfortunately, these approaches remain incompatible and different scientific data Grids remain isolated. While the methodologies for building the local, metropolitan, country-wide or continent-wide storage installations are quite well-developed, connecting them with other installations remains in the area of pioneer works.

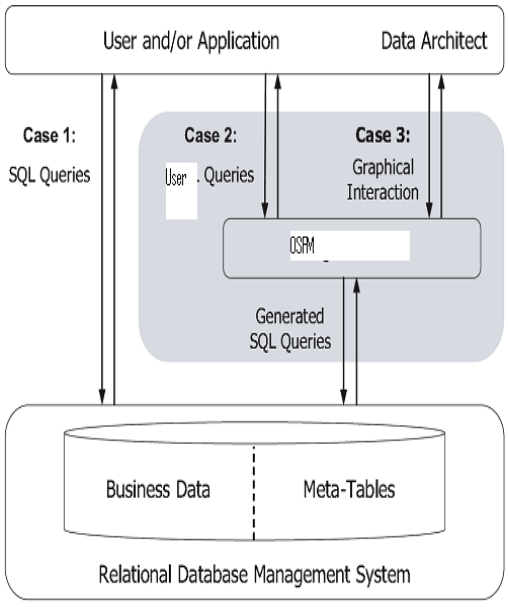

Fig1. Open Service Framework

3. Recent Scenario

In 2009, Bin Cao et al. [4] 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 [5] are used for representing task dependencies and controlling execution. Generic Application Factory (GFac)[6] 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 [7] and CloudSim [8] provide a simulator framework of grid and cloud resources enabling modeling of large grid and cloud resources.Simgrid [9] 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.[10] 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. [11] 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.[12] 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. [13] 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 algorithms 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.

In 2011, Ashutosh Dubey et al. [14] proposed a novel algorithm named Wireless Heterogeneous Data Mining (WHDM). The entire system architecture consists of three phases: 1) Reading the Database. 2) Stores the value in Tbuf with different patterns. 3) Add the superset in the list and remove the related subset from the list. Finally we find the frequent pattern patterns or knowledge from huge amount of data. This technique is efficient for mobile devices.

4. Proposed Algorithm

Our algorithm open service framework mining (OSFM) consists of five phases.

Phase I- in first phase we check the authentication of the user only authorized user can use our data mining framework.

Phase II- In second phase after authentication we read the data from the database.

Phase III-In this phase user enters the minimum support count by which we will find the frequent pattern.

Phase IV- In this phase we find the frequent pattern.

Phase V- In this phase we find the final data after pruning.

Assumptions:

DS- Data Set

MIN_SUPPORT= minimum support count

TEMP- Temporary Buffer

EXIT- java method for exit from the whole program

```
OSFM (DS)

STEP 1:  [AUTHENTICATION]
          IF (TRUE)
             PRINT ("WELCOME IN THE OPEN FRAMEWORK");
          ELSE
              EXIT (0);

STEP 2: [READING DATABASE]
         [READ FROM THE FILE]

STEP 3:  [DEFINE THE MINIMUM SUPPORT]
          MIN_SUPPORT = VALUE [DEFINE BY THE OWNER]

STEP 4: SUBSET (DS, MIN_SUPPORT)
        4.1: STORES THE VALUE IN TEMP
               FOR  i  = 0  to  n-1
                 IF (FREQ >= MIN_SUPPORT)
```



```
                TEMP = DS[ i ];
    4.2:  [FIND THE SUPERSET]
           PEBUFFER = [ALL THE VALUE > = MIN_SUPPORT]
    4.3:  [DELETE THE REPEATED VALUE]
    4.4:   PRUNE (PEBUFFER);

STEP 5: PRUNE (PEBUFFER)
          FOR i =  1  to   n-1
            IF P (FREQUENT)
             PRBUF = APPEND. FREQUENT (DS);
            ELSE

           DELETE
          PRINT (PRBUF);
```

Simulation Result

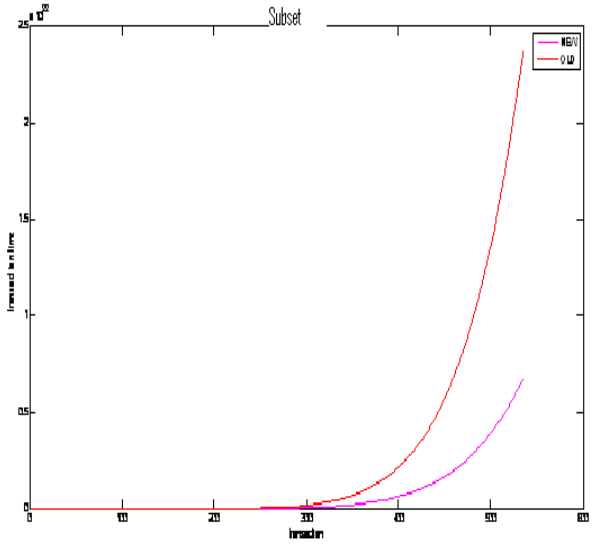

Fig 2 Subset Comparison Graph

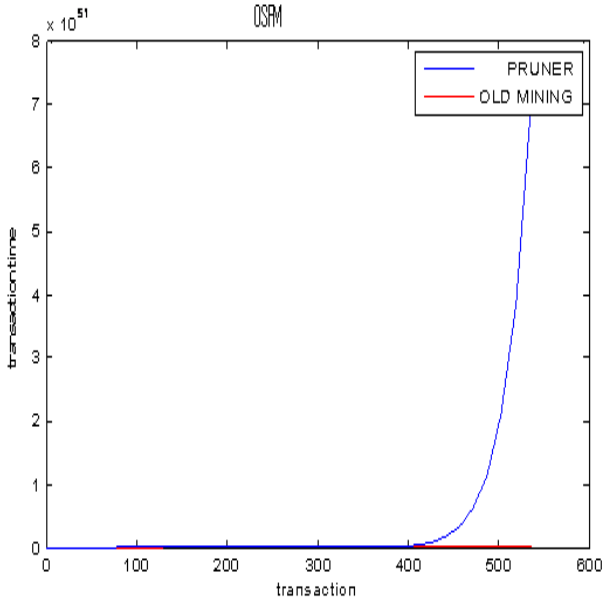

Fig 3. Prune Comparison Graph

The new method shows in the graph that the time is less in comparison of old methods like spade. So it is more efficient. We taking Spade algorithm and our techniques to analyze several aspects like speed and computation time.
The second graph shows that the efficient working of SB-Pruner in comparison of old data mining techniques like

SPADE.The result shows that the novel method OSFM is more efficient than conventional data mining techniques.

5. 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 a intelligent multi-agent system can work seamlessly together despite being functionally independent of each other.

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 Phys- iology of the Grid. In: F. Berman, G. Fox, A. Hey (Eds.) Grid Computing: Making the Global Infrastructure a Reality, Wiley: 217-249, 2003.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] H. Casanova, Simgrid: A toolkit for the simulation of application scheduling.IEEE/ACM International Symposium on Cluster Computing and the Grid May, 2001.
[10] David Schumm, Tobias Anstett, Frank Leymann, Daniel Schleicher ,14th IEEE International Enterprise Distributed Object Computing Conference Workshops,IEEE 2010.
[11] Tobias Pontz, Manfred Grauer, Daniel Metz, Sachin Karadgi, 3rd International Conference on Information Management, Innovation Management and Industrial Engineering,2010,IEEE.
[12] Alexander Wöhrer, Eduard Mehofer and Peter Brezany, 2010 Sixth IEEE International Conference on e–Science Workshops.
[13] David Chiu, Apeksha Shetty and Gagan Agrawal, SC10 November 2010, New Orleans, Louisiana,IEEE.
[14] Ashutosh Kumar Dubey, Ms. Smriti Pandey Prof. Nitesh Gupta ," A Novel Wireless Heterogeneous Data Mining (WHDM) Environment Based on Mobile Computing Environments ",CSNT 2011,IEEE.

Authers

Pooja Sharma

