ARCHITECTURE-CENTRIC DATA MINING MIDDLEWARE SUPPORTING MULTIPLE DATA SOURCES AND MINING TECHNIQUES

Sai Peck Lee and Lai Ee Hen
Department of Software Engineering, Faculty of Computer Science & Information Technology, Universiti Malaya
50603 Kuala Lumpur, Malaysia

Keywords: Knowledge Discovery, Data Mining, Middleware, Data Mining Middleware.

Abstract: In today's market place, information stored in a consumer database is the most valuable asset of an organization. It houses important hidden information that can be extracted to solve real-world problems in engineering, science, and business. The possibility to extract hidden information to solve real-world problems has led to increasing application of knowledge discovery in databases, and hence the emergence of a variety of data mining tools in the market. These tools offer different strengths and capabilities, helping decision makers to improve business decisions. In this paper, we provide a high-level overview of a proposed data mining middleware whose architecture provides great flexibility for a wide spectrum of data mining techniques to support decision makers in generating useful knowledge to help in decision making. We describe features that we consider important to be supported by the middleware such as providing a wide spectrum of data mining algorithms and reports through plugins. We also briefly explain both the high-level architecture of the middleware and technologies that will be used to develop it.

1 INTRODUCTION

In today's information age, the increasing volume of data due to the capability of technologies in the generation and collection of data (Cheng, 2000) has led to the needs of turning these data into useful information for decision making. Knowledge Discovery in Databases (KDD) comes into the image where low-level data are turned into high-level knowledge for decision support. According to Fayyad et al, "Knowledge discovery in databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad, Piatetsky-Shapiro, Smyth, 1996). Understanding the implicit information in the data is important for strategic decision support. However, the data is often scattered throughout the corporation and integration of data before analysis is necessary (Michael & Gruenwald, 1999). Hence, the ability for data mining tools to access different data sources is essential. In addition, one of the challenges in data mining is the ability to mine diverse knowledge in databases (Jiawei & Micheline, 2006). Different users may have different interests of knowledge. As such, a well designed data mining tool should provide a wide spectrum of data mining techniques to solve different business problems. As such, this research is to propose an architecture for a data mining middleware to support a variety of data mining functionalities, where the architecture provides great flexibility for a wide spectrum of data mining techniques from multiple data sources. The intention of the proposed architecture is to design a platform- and language-independent middleware that allows organizations to mine data through a wide range of data sources such as relational databases, multidimensional databases, flat files, hierarchical databases, object-oriented databases, XML files and others, to solve real-world business problems.

2 RELATED WORK

There are various data mining tools available in the market such as IBM Intelligent Miner, SPSS Clementine, SAS Institute Enterprise Miner, Oracle Data Miner, and Microsoft Business Intelligence

Peck Lee S. and Ee Hen L. (2007).
ARCHITECTURE-CENTRIC DATA MINING MIDDLEWARE SUPPORTING MULTIPLE DATA SOURCES AND MINING TECHNIQUES.
In *Proceedings of the Second International Conference on Software and Data Technologies - Volume ISDM/WsEHST/DC*, pages 224-227
DOI: 10.5220/0001326102240227

Development Studio. Majority of the studies conducted on those tools tend to primarily focus on the functions of the tools rather than the performance of the tools. Our study mainly focuses on the performance of the tools based on five attributes: memory shortages, excess paging with a disk bottleneck, paging file fragmentation, memory leaks, and cache manager efficiency.

Based on our study, data mining tools such as Oracle Data Miner and Microsoft Business Intelligence Development Studio cache a certain percentage of both unmined and mined data in the application tier. Such a strategy off-loads computing cycles from the backend systems (for example, Microsoft Business Intelligence Development Studio, and Oracle Data Miner). However, both the unmined and mined data are not fully persisted or cached in the backend systems. As such, we might have cases whereby two users might be mining the same data set and this causes redundancy in terms of work performed.

Our study also reveals that data mining at the memory level will lead to better performance. For example, IBM Intelligent Miner consumes only 15% of Physical Disk\Disk Time and 100MB of Memory\Available Bytes. This explains that there is a trade-off between memory and disk. If we spend more time at the memory level, then we should spend less time on disk activity (also referred to as Disk I/O). Disk I/O is often a major bottleneck to data mining performance.

To improve the performance of data mining, our study reveals that major data mining activities should be performed in-memory at the server level. Therefore the proposed memory repository of the middleware will be adopted from SQL Server Analysis Services. In the transition of 32-bit computing to 64-bit computing, we believe the proposed middleware will be able to leverage at the memory level. In the near future, we believe major data mining tools like Microsoft Business Intelligence Development Studio and Oracle Data Miner, which are almost vendor dependent, will leverage at the memory level.

At the time of our study, tools like Microsoft Business Intelligence Development Studio, Oracle Data Miner, and SAS Institute Enterprise Miner only support a predefined set of data sources. For example, Microsoft Business Intelligence Development Studio only supports ODBC, OLEDB and other types of predefined data sources. Oracle Data Miner, on the other hand, only supports JDBC compliant driver such as OCI-based drivers. Implementing new data sources into such tools are difficult and often require understanding of the specified data source API specification. For example, in the case of Oracle Data Miner, implementers need to understand the JDBC API specification.

A data mining tool might face the constraint of platform dependent (Sanjiv, 2006). Tools such as Microsoft Business Intelligence Development Studio and SPSS Clementine are not platform independent. Microsoft Business Intelligence Development Studio depends on .NET Framework which currently only supports the Windows platform. In order to support other platforms such as Linux, tedious customizations are needed. SPSS Clementine, on the other hand, releases different binaries on different platforms. Oracle Data Miner uses the same binaries on different platforms, and as such, is platform independent.

3 PROPOSED DATA MINING MIDDLEWARE

This paper discusses our proposed architecture for a data mining middleware to be developed which employs the strengths and eliminates the weaknesses of other data mining tools available in the market. We will refer this middleware as Java-Based Data Mining Middleware (JDMM). This proposed architecture is a server centric middleware that provides the flexibility in which data mining techniques are unlimited. New data mining techniques are allowed to be plugged into the middleware. In addition, JDMM will be a platform-, data source-, and data mining technique-independent middleware which is accessible from front-, back- and web-office environments. JDMM is designed to minimize the level of disk activity (Disk I/O) over time during data mining by introducing the concept of memory-optimized repository and other technology. Disk I/O is an important performance metric during data mining as disks are often a major bottleneck attribute to data mining performance. Performance of applications with any I/O will be limited, further CPU performance improvements will be wasted (Peter & David, 1993). This is particularly true for a database driven data mining. Hence, JDMM architecture needs to be designed to address the issue of I/O throughput of disks to enable a highly scalable and an almost instantly responsive server-centric data mining middleware.

3.1 Overview of Proposed Middleware

Figure 1 portrays the proposed high-level system architecture for JDMM. The middleware is proposed with three possible roles of users: Administrator, Implementor, and Business Analyst. Administrators will administer and ensure the uptime of JDMM. Implementors are technical users who are able to plug new adapters through the JDMM Web Configurator. Lastly, Business Analysts are non-technical users who are responsible on business decision-making by accessing Web JDMM to solve real-world business problems. The Enterprise Java Bean (EJB) server acts as a retrieval engine and consists of different adapters to interconnect different data sources with JDMM.

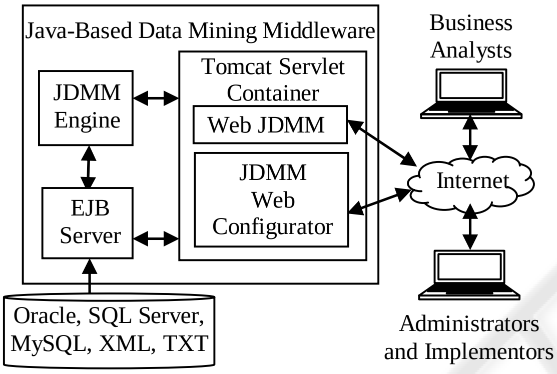
Java-Based Data Mining Middleware
Tomcat Servlet Container
JDMM Engine
Web JDMM
JDMM Web Configurator
EJB Server
Business Analysts
Internet
Oracle, SQL Server, MySQL, XML, TXT
Administrators and Implementors

Figure 1: JDMM Architecture Information Flow.

After data retrieval, JDMM organizes the data to create a data mining model using a specific data mining technique. Each technique is governed by adapters which are pluggable rule adapters. At this stage, the result can be stored into a data mining repository or directly to a persistent data store at a specific point of time interval. The primary objective of the repository is to cache results so that computed results are not computed again. The result is a XML file that will then be delivered to the client in any proprietary format incorporated in JDMM.

We believe that the architecture is able to address real-world business scenarios in business areas such as human resource, business management and project management, IT operations, financial, marketing and so forth. For example, in a typical bug tracking application, JDMM can be employed to analyse project related metrics such as issues per state, priority, severity, category and resolution. These metrics are useful to measure the success of future projects. If the number of projects to be measured is large, the memory repository of JDMM is able to reduce the time required during data mining. On the other hand, if the project related data are stored in different sources, the JDMM adapters can be configured accordingly. These data sources can be managed collectively within JDMM.

3.2 JDMM Detailed Architecture

The internal architecture of the proposed middleware is divided into two threads namely Inbound Threads for managing incoming uninterpreted operational data (raw data) and Outbound Threads for managing all outgoing interpreted data (mined data) shown in Figure 2.

Both the Send Adapter and Receive Adapter are part of a framework known as the Adapter Framework. Through the JDMM Web Configurator, implementors of JDMM are able to plugin different adapter components to connect to different data sources. Each adapter is configurable and each configurable parameter is stored in a XML file.

Java-Based Data Miner (JDM) will be a pure Java API for developing data mining applications. The idea is to have a common API for data mining that can be used by clients without users being aware of or affected by the actual vendor implementations for data mining.

JDM Extension will be an extension to JDM that includes additional data mining models, data scoring and data transformations. JDM Extension will be designed to be a highly-generalized, object-oriented, data mining conceptual model using Data Mining Group’s Predictive Model Markup Language (PMML) data mining standard. PMML is an XML markup language to describe statistical and data mining models ("Predictive Model Markup Language", 2005).

4 PROPOSED LOGICAL COMPONENTS OF JDMM

JDMM Web Configurator and Web JDMM are situated in the application system layer which is built on top of the business specific component systems, Adapter Framework and JDM. Adapter Framework is an extensible framework that allows different multiple adapters connecting to different data sources to be added. This framework will be the component system to enable JDMM users to establish connections to a wide variety of data sources. JDM will be the engine to perform any data mining process.

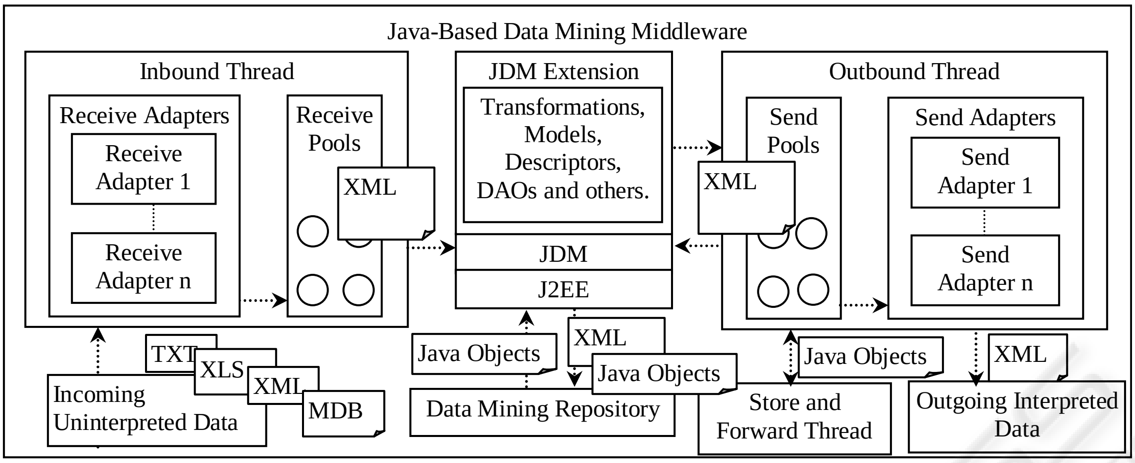

Figure 2: Detailed software architecture of JDMM.

JDMM Web Configurator and Web JDMM communicate with each other during the data mining process. If a decision maker wishes to mine a different data source, a different adapter corresponding to the data source will need to be configured and deployed into the system through the JDMM Web Configurator. From Web JDMM, the decision-maker actor will be able to perform the data mining process from the adapter that is deployed.

5 FUTURE CONSIDERATION

An implementation of the JDMM architecture is in progress. At the current phase of our work, we believe conceptually that JDMM is able to provide an analytical platform that is configurable and extendable throughout the business enterprise. A known limitation of JDMM is that it relies heavily on memory. Careful implementation of JDMM is required to ensure that unused objects are to be efficiently and effectively garbage collected. Otherwise, memory will be a potential bottleneck. In the near future, JDMM will be leveraged to enable massive data sets to be analyzed. As data sets increase in size, traditional data mining tools become less efficient. Our approach to analytical scalability can be addressed through grid computing and 64-bit computing. “Grid” computing has emerged as an important new field, distinguished from the conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and in some cases, high-performance orientation (Foster, Kesselman, & Tuecke, 2001). We foresee that grid computing will empower data mining with instant responsiveness and very high throughput in terms of analyzing mission-critical data sets in real-time enterprises and industries. With grid computing, mined data can be accessed, captured, or updated many times faster, giving business analysts a fast response to business-critical decisions. With 64-bit computing, the memory limitation of the proposed architecture issue will be solved.

REFERENCES

Cheng Soon Ong, 2000. *Knowledge Discovery In Databases: An Information Retrieval Perspective*, Malaysian Journal of Computer Science. Vol. 13 No. 2. pp. 54-63

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P., 1996. *From Data Mining to Knowledge Discovery: An Overview*. Advances in Knowledge Discovery and Data Mining. MIT Press. 37-54.

Ian Foster, Carl Kesselman, and Steven Tuecke, 2001. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. 1-25.

Jiawei Han and Micheline Kamber, 2006. *Data Mining: Concepts and Techniques*. Second Edition. Elsevier. p5 – 45

Michael Goebel, and Le Gruenwald, 1999. *A Survey Of Data Mining And Knowledge Discovery Software Tools, Sigkdd Explorations*. ACM SIGKDD. Volume 1, Issue 1 – 20 - 33

Peter M. Chen and David A, 1993. *Storage Performance—Metrics and Benchmarks*. Patterson. Volume 81. 1-33.

Predictive Model Markup Language (PMML). 2005 Technology Reports. Cover Pages. http://xml.coverpages.org/pmml.html. Retrieved August 8, 2005

Sanjiv Purba, 2006. *Handbook of Data Management*. Viva Books Private Limited.