GENERIC DATA ACCESS AND INTEGRATION SERVICE FOR DISTRIBUTED COMPUTING ENVIRONMENT

Hemant Mehta[1], Priyesh Kanungo[2] and Manohar Chandwani[3]

[1]School of Computer Science, Devi Ahilya University, Indore, India
mehtahk@yahoo.com
[2]Patel College of Science & Technology, Indore, India
priyeshkanungo@hotmail.com
[3]Department of Computer Engineering, Institute of Engineering and Technology, Devi Ahilya University, Indore, India
chandwanim1@rediffmail.com

ABSTRACT

This paper presents a Generic Data Access and Integration Service GDAIS for various distributed computing environments. The service can be deployed within Peer-to-Peer, Cluster, Grid and Cloud computing environment. Grid service environment has been utilized to develop GDAIS and it provides the generic data access facility to the clients. The clients can manipulate data stored in various data stores in transparent manner. The service supports simplified access to a variety of sources like the files on file system, web services and various databases like XML databases, relation databases, object oriented databases and object relational databases. A uniform API to manipulate data from disparate and heterogeneous data sources is developed. Various operations that can be performed are 'read', 'select', 'insert', 'update',' delete' and 'bulk insert'. These APIs are developed as web services.

KEYWORDS

Cloud computing, Cluster computing, Data Access and Integration, database, Grid computing.

1. INTRODUCTION

Distributed computing systems have been extensively used for large scale scientific and commercial applications for a long time. These applications generate large amount of data that entail extensive computations. To meet the processing needs of these applications, various types of distributed systems have evolved over a period of time. Cluster computing is used if the processing can be done within the organization itself. Grid computing shares the capability of geographically distributed resources. P2P computing shares the data directly between peer nodes in decentralized manner. Grid and clusters can be used for market oriented computing where consumers pay to the resource providers for the resources and services being offered. Service oriented computing focuses on providing business processes in the form of web services. The concepts of Grid computing, market oriented computing and service oriented computing evolved further and combined together to develop Cloud computing. Cloud computing is a collection of dynamically scalable and virtualized resources, which are provided as a service over the Internet. These services can be Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

The web services are software systems designed to support the interaction amongst different machines connected through a network. The service requesters can further be software that requires information provided by the web service. Simple Object Access Protocol (SOAP) is

DOI : 10.5121/ijgca.2010.1102

used for preparation of the request to and response from the web services. Web services publish their details to a registry called Universal Description, Discovery and Integration (UDDI). UDDI is a platform independent registry for web services [14]. UDDI uses Web Service Definition Language (WSDL) to describe the details of web services. Using WSDL, the detailed functionality of web services can be specified. These details are accessed by the clients to invoke the functionality of web services. Web services use the existing internet infrastructure and HTTP for the interaction with its requesters. Extensible Markup Language (XML) is a standard to specify the data about an application. SOAP messages, UDDI and WSDL use XML for specifying their details.

In the last few years, users have started harnessing the power of web services for Grid computing called service oriented Grid computing. The largest Grid users' community called, open grid forum (OGF), has described architecture for service oriented Grid computing environment for commercial and scientific applications. This architecture is known as Open Grid Services Architecture (OGSA). OGSA is a web service based distributed computing and interaction architecture. OGSA takes care of heterogeneity of the resources resulting in proper communication and sharing of the information amongst them.

Research in the Grid applications mainly focuses on the processing data stored in files. However, several applications require accessing the data stored in sources other than the files. Providing a single and uniform interface to these different sources is called as data integration. The data integration is a challenging task and is still a persistent problem for the database community. The problem becomes more complex when the data sources are diverse and heterogeneous. Generally, the applications involving the huge amount of data are deployed over the Cluster, Grid or Cloud computing environment. All such applications require the data from the diverse sources, demanding the effective data integration technique, preferably in the form of web services due to the diversity in their resources.

This paper is organized as follows: Section 2 examines related work. Section 3 provides detailed model of GDAIS. The sample client is described in Section 4. Section 5 presents the details on interoperability of the GDAIS. Finally, Section 6 concludes the paper.

2. RELATED WORK

Literature in the field of data access and data integration services contains some recent research projects including OGSA-DAI & GDIS, SDO, Oracle Data Service Integrator and Microsoft SQL Azure Database Service.

The Global Grid Forum's (GGF) Data Access & Integration Services Working Group (DAIS-WG) proposed the specification for Grid data services based interface to various XML and relational databases. The Open Grid Services Architecture Data Access Integration (OGSA-DAI) provides an implementation to the DAIS-WS specification [11]. It allows exposing various data sources to the web, Grid or Cloud. OGSA-DAI provides the facility to query and manipulate the data using web services. However, it doesn't support the integration of data from heterogeneous data sources.

Comito *et al.* proposed a service based data integration architecture called Grid Data Integration System (GDIS). It is a decentralized architecture build on top of OGSA-DAI and Globus Toolkit [12]. It provides the integration of data from various data sources [5]. Gounaris *et al.* have presented a service based system for data integration on data grids. They developed this system based on the OGSA Distributed Query Processor and XMAP [21] query reformulation algorithm [21]. They used a wrapper-mediator approach for distributed query

processing across independent databases on various grid resources. This technique is specially developed for data grids.

Gallo et al. have presented architecture for data access and integration called Grid Integration Service (GISE). They tested this design on epidemic monitoring and simulation system. It uses mediators, wrappers and canonical data models to perform query on distributed heterogeneous datasources [6]. This service is built using Globus toolkit.

Service Data Objects (SDO) is a specification for data programming architecture and an implementation API. The key concepts of the SDO are data objects, data graphs and data access services (DAS). The data objects are the way to represent the business data in the form of properties. Data graph contains the data objects and act as a transportation unit between various components. Data graphs manage the data objects by keeping track of insert, delete and update operations performed on the data objects [4]. Oracle Data Services Integrator (ODSI) is an implementation of SDO [1]. ODIS provides read and write access to various data sources like relational databases, XML databases, files, web services etc.

SQL Azure Database is a Cloud-based database service build on SQL server technology. It supports various data manipulation operations to be performed on the databases. Its main limitation is that it supports only one relation database called Microsoft SQL server [2, 3].

Paton et al. have proposed a data access and integration service on the grid. This is not a concrete implementation of these services it is just a proposal of accessing the databases using the service based infrastructure [20]. In this research the Paton et al. have proposed various services and the operations supported by the services. The various services proposed are basic data service (for basic database operations e.g. Select, update etc.), transactional database service (for transaction related operations e.g. commit, rollback etc.) and coordinated database service (for distributed transactions).

All the existing data access and integration services require various toolkits and technologies. Moreover, they are either specific to particular databases or to a particular architecture. There are several similar research contributions which are following a particular architecture that is, centralized, decentralized or mediator based architecture. These projects are specially designed for large scale commercial applications and hence they are very complex and heavy load projects. Moreover, some of the projects are proprietary projects. However, our proposed GDAIS is simple and generic in terms of both the databases and architecture. GDAIS support all the kinds of data storage and supports the centralized as well as decentralized architectures. Moreover, GDAIS is built using simple and basic technologies so that is can be used specially with scientific applications. GDAIS has been developed using Java based open source tools, namely Java, Java Enterprise Edition, Apache Tomcat, Apache Axis and Eclipse IDE [7, 8, 10, 13, 16].

3. SYSTEM MODEL

A system model of GDAIS is shown in Figure 1. The lowest layer depicts various operating systems and platforms hosting the data stores. The data stores may be a relational database, object oriented database, object relational database indicated by (RBD…ODB), XML database, web service, flat files etc. Interface is provided for various popular databases available like Oracle, DB2 [17], MS SQL Server, MySQL [18] etc. To store the text data in flat files both text files and binary files are used. The objects are stored in the binary file, while the text information can be written in text files using ASCII or UTF code. However, objects are stored in the binary streams using the concept of object serialization [9]. Several Java-based tools are

used to develop GDAIS. Function of GDAIS can be accessed by any client connect from the clusters, the Grid, P2P network or the Cloud.

Figure 2 contains class level details of the GDAIS. Currently, GDAIS is composed of total four classes and three interfaces. The interfaces are the specification of the functionality of the various classes of GDAIS. The DAOInterface represents the interface having the methods to be implemented in the class used for data access. Classes implementing the data access and integration facility should extend this interface. It provides methods to insert update or delete the data stored in data stores. Another interface is DataSourceInterface that defines the methods that should be implemented by the classes accessing data from different data sources. We provided few sample concrete classes to access data from Oracle, MySQL, object serialization etc. The last interface is DataServiceInterface that defines facility provided by the GDAIS. The class ObjectFactory is defined to implement the factory design pattern [15]. Another class is a concrete class defined for web service implementation of the GDAIS. A web service requires two more classes one for stubs (DataServiceStub) and the other is the callback handler. These two classes are generated using Apache Axis.

The Apache Axis is used to expose the Java class as a web service. To create a web service, it creates two classes called stub and callback handler. The stub class is created to convert the methods in the java class into the methods of web service. The callback handler class can be used to filter or modify the data returned by the methods of the web service. The callback handler intercepts the data or error message returned by the web service. These data or error can be processed further by the corresponding methods of the callback handler class. From Figure 2, we observe that for each method in the web service, callback handler class creates two methods. First method is created to access the data returned by the actual method while the other method accesses the error message returned by the actual method.

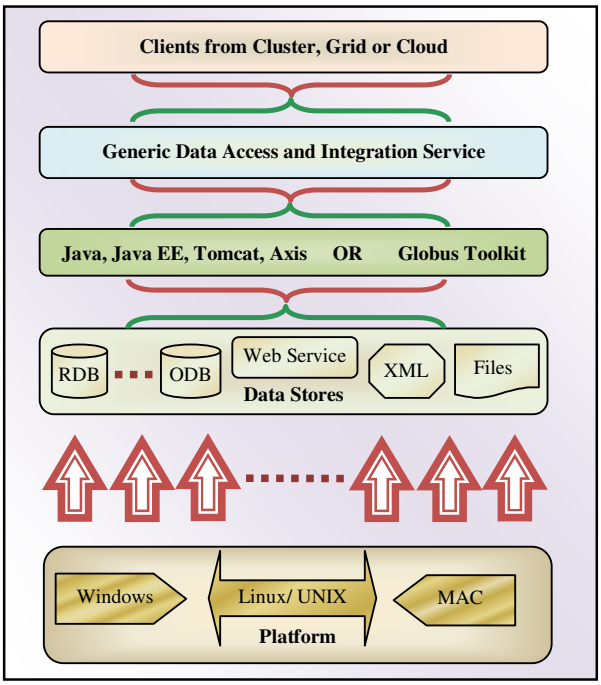

Figure 1. A System Model of GDAIS

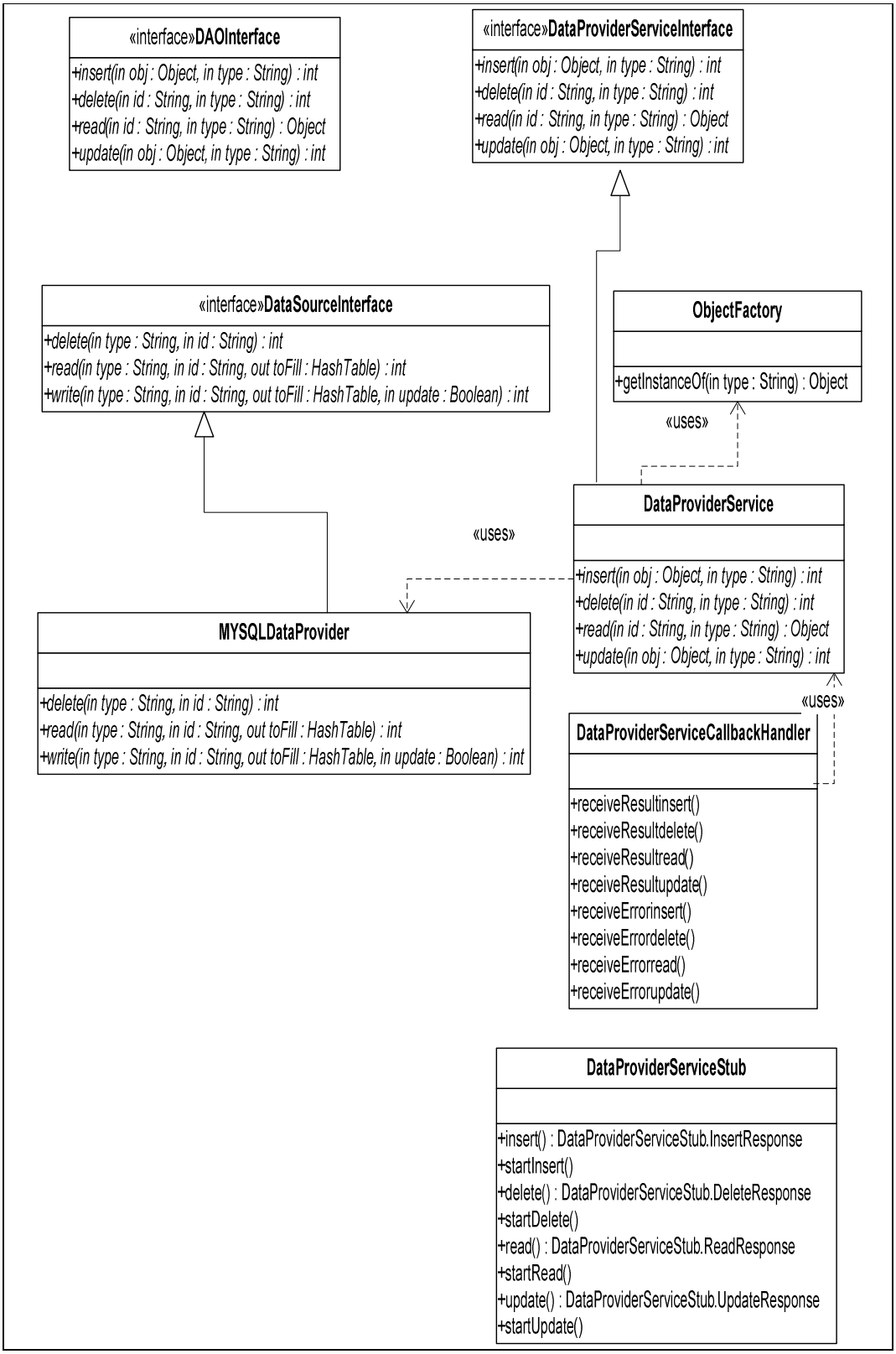

Figure 2. Class Diagram of GDAIS

4. SAMPLE CLIENT

The following code fragment describes the sample client to access the web service. Similarly, we can create client programs for accessing remaining operations supported by GDAIS. The client uses web service stub to access the functionality of the service. This sample client is selecting data from a data store and just printing the details. The code fragment assumes that the data returned is an object having getDetail method. Initially, the process starts with setting the details of the data source in line 2. The read method performs select operation on data.

```
1. DataServiceStub stub = new DataServiceStub("URL of the service");
2. DataServiceStub.setDriver("Data Source Name");
3. DataServiceStub.Read readObj = new DataServiceStub.Read();
4. DataServiceStub.ReadResponse readRes = stub.read(rd);
5. print("The details from the object read are", res.get_return().getDetail());
```

5. INTEROPERABILITY OF GDAIS

GDAIS is a standard SOAP service and easily interoperable from any web service client application. The UDDI hosts the online registry of the web services in the network. The client application fetches information regarding the web services from UDDI and using SOAP messages they can interact with the web services. SOAP, WSDL and UDDI protocols provide a self-describing way to discover and use the methods provided by the web services regardless of its platform. Figure 3 depicts the sample communication between a web service and web service client. Every web service registers their details at the UDDI registry. Perspective clients can find the desired web service from a UDDI registry.

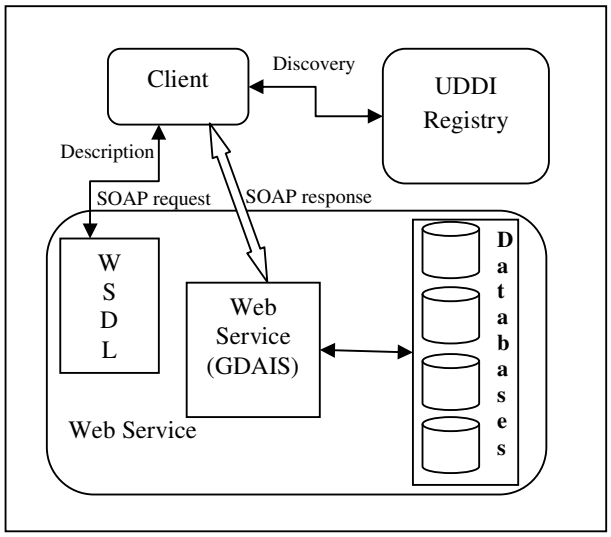

Figure 3. Interaction between the web service, UDDI registry and client

UDDI registry keeps the details regarding the location and address of the web service. Subsequently the client fetches the information regarding the web service functionality from the WSDL. Now clients prepare the call to web service, in turn, the service container fetches the data from various data stores. The client program uses jUDDI API to access the UDDI registry information [19]. Three kind of query about web service can be requested to the UDDI registry viz. white pages, yellow pagese and green pages. White pages query returns the address and contact information about the web service. Yellow pages query retrieves the category and

taxonomy information about the web service, while the green pages query determines the functional and technical details regarding the web service. As the Globus toolkit is widely used for Grid computing environment, we also integrated the GDAIS with the Globus toolkit.

6. CONCLUSION

In this paper a web service based API for data access and integration called GDAIS is presented. This service provides facility to access and manipulate data from various data store, namely, file system, XML databases, relational databases, object oriented database and object relational databases. The API is simple, generic and easy to use as compared to other similar research work. It is generic in terms of databases as well as the implementation architecture. It provides simple interface with less resource consumption for the users of data stores. We feel that it can be very useful for the data intensive scientific applications deployed over Cluster computing, Grid computing or Cloud computing environment.

7. REFERENCES

[1] Oracle Data Service Integrator (ODSI), Available Online at: http://download.oracle.com/docs/cd/E13162_01/odsi/docs10gr3/pdf/appdev.pdf

[2] Microsoft SQL AZURE, Available Online at: http://www.microsoft.com/windowsazure/whitepapers/

[3] Microsoft SQL Server, Available Online at: http://www.microsoft.com/SQL/default.mspx

[4] Service Data Objects Specifications, Available Online at: http://www.osoa.org/display/Main/Service+Data+Objects+specifications/

[5] C. Comito, D. Talia, T. Paolo, "Grid Services: Principles, Implementations and Use", *International Journal of Web and Grid Services,* Inderscience Publishers, Vol. 1, Issue 1. Aug 2005, p. 48-68.

[6] E. Gallo, H. F. Gagliardi, F. A. B. d. Silva, V. C. Neto, D. Alves, "GISE-A Data Access and Integration Service of Epidemiological Data for a Grid-Based Monitoring and Simulation System", 40th Annual Simulation Symposium (ANSS'07), pp. 267-274, Mar 2007.

[7] The Java Language Platform, Available Online at: http://java.sun.com/javase

[8] The Java Enterprise Edition, Available Online at: http://java.sun.com/javaee/

[9] Object Serialization in Java, Available Online at: http://java.sun.com/j2se/1.5.0/docs/guide/serialization/

[10] The Apache Tomcat, Available Online at: http://tomcat.apache.org/

[11] Open Grid Services Architecture Data Access and Integration, Available Online at http://www.ogsadai.org.uk

[12] The Globus Toolkit, Available Online at: http://www.globus.org/toolkit/

[13] The Apaches Axis Project, Available Online at: http://ws.apache.org/axis

[14] The UDDI Specification version 3.0, Available Online at: http://www.uddi.org/pubs/uddi_v3.htm

[15] The factory method pattern, Available Online at: http://en.wikipedia.org/wiki/Factory_method_pattern

[16] Eclipse Integrated Development Environment, Available Online at: http://www.eclipse.org

[17] IBM DB2 Database Available Online at: http://www.ibm.com/db2

[18] MySQL Open Source Database, Available Online at: http://www.mysql.com/

[19] The Apache jUDDI, Available Online at: http://ws.apache.org/juddi

[20] N.W. Paton, V. Dialani, T. Storey, M. P Atkinson, D. Pearson, P. Watson, "Database Access and Integration Services on the Grid," *4th Global Grid Forum (GGF 4) Databases and the Grid BOF*, Feb 2002, pp. 1-18.

[21] C. Comito, D. Talia, "XML Data Integration in OGSA Grids", *1st International Workshop on Data Management in Grids (DMG),* Sep 2005, pp. 4–15.

[22] A. Gounaris, C. Comito, R. Sakellariou, D. Talia, "A Service-Oriented System to Support Data Integration on Data Grids", *7th IEEE International Symposium on Cluster Computing and the Grid(CCGrid'07),* May 2007 pp. 627-635.

Authors

Mr. Hemant Mehta is Assistant Professor of Computer Science at School of Computer Science, Devi Ahilya University, Indore, MP, India. He received B.Sc. (Computer Science) Hons., MCA (Master of Computer Applications) from Devi Ahilya University, Indore, India in 1998 and 2001 respectively. He has published one book and seven research papers. His area of research is resource management under distributed computing (Cluster, Grid and Cloud computing). He is having 10 years of experience in the research, academics and software development.

Dr Priyesh Kanungo is working as a Principal in Patel College of Science and Technology, Ralamandal, Indore, MP, India. He received ME, M Phil, Ph D. in Computer Engineering. He completed Ph D in the supervision of Dr Manohar Chandwani from IET-DAVV, Indore. His area of Research is Distributed Scheduling (Dynamic Load Balancing). He is having 22 years of teaching experience in M Tech, MCA, MBA, BE etc in Devi Ahilya Vishwavidyalaya, Indore. Presently he is Guiding 5 Ph Ds in the area of Distributed Scheduling and has published around 20 research papers.

Dr Manohar Chandwani is Professor & Director, Institute of Engineering & Technology, Devi Ahilya University, Indore MP India. His research interest is in the areas like NLP, Information Security and Distributed Computing. He has guided seven PhDs and has published 25 Journal Papers & 75 Conference Papers at national and international level. He is author of three books in computer science. He is Fellow of CSI, IETE, IE (India), Senior Member of IEEE and member of ACM. He has been the chief editor of the Journal of CSI during 1999-2007 and he shall be the web publication committee, chair of CSI. He has been involved in the development of Institute of Engineering & Technology since its inception.