

Research of hybrid database middleware architecture

Hui-Ru Zhang, Xiao-Peng Ji and Yong-Quan Yang

Ocean University of China, Qingdao, Shandong, China

Email: zhanghuiru1018@163.com, jxiaopeng@ouc.edu.cn, i@yangyongquan.com

With the rapid development of techniques, more and more enterprise applications need to integrate the different information from diverse database sources. In order to resolve the problem of data sharing and collaboration among heterogeneous databases in the enterprise network, we propose hybrid database middleware architecture based on XML and CORBA technology in this paper. The frame uses XML as a medium for data format change, applying the CORBA theory and JDBC technology to carry out the operation, and it makes the user transparently share. The middleware architecture shields the heterogeneity of operations, which can provide users a unified access interface and realize the characters of transportability, extensibility, transparency.

Keywords: Heterogeneous; Middleware; CORBA; XML.

1. Introduction

For the last few years, along with the development of the network techniques and the database techniques, the new software and hardware platform emerge in endlessly, many companies and enterprises have realized the internal information management. An important foundation of these enterprise applications is the database system. However, due to the different departments within the enterprise business, a lot of independent information services and management systems are established. Enterprises, simultaneously, have accumulated a lot of data which is memorized with different forms and depended on different Database Management Systems. As a result, a complex and heterogeneous database environment is constituted, which forms an impediment to realize the resource sharing. How to integrate the different information system and provide uniform interfaces becomes an urgent problem, and the hybrid database middleware is an ideal solution to realize heterogeneous data interaction.

Middleware is a kind of absolute system software or service program, is a sort of software platform, which lies between the distributed application system and the operating system to provide global, mature and stable services to

application. It aims to help applications communicate with the underlying operating system, shield the complexity and heterogeneous of the distributed environment, so as to realize the resource sharing [1]. The middleware provides flexible and agile environment for the development and operation of applications, reduces the complexity of program design, and also significantly reduces the burden of programmers. However, the database middleware is the most sophisticated of all kinds.

Common Object Request Broker Architecture (CORBA) is an object oriented distributed calculating platform raised by OMG, and is one of main distributed computing technologies. The ORB is the kernel component of OMA Reference Model (issued by OMG), and it guarantees the realization of interoperation under the isomeric network environment. CORBA is the functional description and agreement of ORB, is an object-oriented distributed computing platform, which enables transparent access between the shielding hybrid databases. Because of CORBA platform is programming language-independent and platform-independent, it has been recognized by people among the life and gradually become the mainstream of middleware technology platform.

XML (Extensible Markup Language) is a new markup language. With the development of the related techniques and applications, it quickly became an actual standard for information exchange and data sharing. It is an open way to describe the data structure, and it has deeply influenced the unity of the different information formats. At the same time, XML has the characteristics of cross-platform and strong adaptability, and can be exchanged the structured data and unstructured documents between different system platform to achieve rapid packaging and integration of resources. Therefore, as a medium of data exchange, XML brings an important shortcut for the hybrid database middleware architecture [2].

Forasmuch as the significant advantages of CORBA and XML, this paper discusses how to play the best of both to concisely solve the heterogeneous database transparent access problem, and proposes an architecture model which based on the two technologies.

2. Related Work

Hybrid database middleware integration is not a new field of research, along with the increased amount of data in the enterprises and institutions. The requirement of data processing is more and more complex, and the demand has emerged on the surface of transparent sharing data. Under this circumstance, scholars at home and abroad have done some related research.

The US government began to support the research of YAT system through the OPAL plan in 1998, the system adopts the Mediator/Wrapper middleware framework, which aims to implement the relational schema [3]. In 2000, the system study received the grant again by the OPAL plan and AQUARELLE. The study mainly did the research of information integration which based on XML technology, CHRISTOPHIDES V, CLUET S, SIMEON J et al. focused on the research of XML integration view and query to the packing of the heterogeneous database, and the relevant algebraic system structure was put forward.

Star Bus is a CORBA middleware product which supported by the 863's program and designed by the School of Computer Science in National University of Defense Technology. It contains soft bus, system services, Web connectors and many other related technologies to communicate application system integration. For now, it has been successfully applied to the telecommunication, finance, and military command automation, and significant benefits have been achieved.

Infor Link is a database middleware which developed by the CVCISE [4]. It provides a standard JDBC and ODBC interface for users to meet the demands of extensibility and openness. Meanwhile, it can afford a unified access interface and database connection pool mechanism, support for the reuse of the database connection; improve the efficiency of database connection resources.

Versatile is developed by Southeast University, and is a scalable heterogeneous data source integration system prototype based on CORBA. The model on IONA's products can package and integrate the data of SQL Server, Versant, file system and hypertext. The system can not only integrate the data source, and can also integrate data from new data source anytime.

In summary, along with the existing heterogeneous data integration methods and the application of hybrid database sharing platform. This thesis absorbs the advantages of the current performance, advances a new kind of hybrid database middleware architecture which combined XML and CORBA technology.

3. Our Work

In order to achieve the transparency and consistency of heterogeneous database access, we propose a model based on XML and CORBA technology to construct a database access middleware. The model does not need to change the original database storage and management mode of database; it provides a unified logical view in the intermediate to hide the details of the underlying data, so that the users can integrate various data sources as a consolidated whole.

3.1. Key Technology

CORBA is a kind of middleware, and it provides a multifunctional consistent object model. The standard is mainly divided into three levels: object request broker, common object services and public facilities [5]. The bottom is Object Request Broker (ORB), which encapsulates the operations and attributes of various objects indifferent interfaces. At the same time, it stipulates the definition and mapping of the distributional object, blends the distributed processing and object-oriented method. It provides a way to achieve transparent communication through the ORB to call the operation of interface [6]. Based on CORBA, we propose the new hybrid database middleware architecture in this paper; it can share information and resources in a distributed heterogeneous environment to solve the interoperability issues between different hardware and software [7].

XML is a kind of common language specification of the year 1998 after it was endorsed by the World Wide Web Consortium (W3C), it is a simplified subset of SGML with specially designed for Web applications. As an extensible markup language, XML's own descriptive makes it ideal for data exchange between different applications, and this exchange is not in a predetermined set of data structure. The biggest advantage is the ability of data description and transmission, so that it possesses an enormous openness [8]. By providing an XML interface for the different existing database systems, the outside world can implement access to any existing platform, and send the results as XML to other platforms, so as to realize the information exchange between heterogeneous databases. The key to map the database mode is DTD and Schema, with it, the data source type can be mapped to the supported XML data, thus integrating the heterogeneous data types [9]. All in all, XML is easy to understand, easy to learn, and has strong adaptability, so that data resources can be quickly packaged and integrated with it. Therefore, this paper introduces the technology of XML to make the hybrid database middleware better able to adapt to an open environment, and better interact with heterogeneous data sources.

JDBC is an application programming interface that Java applications use to access relational databases, which consists of a set of Java classes and interfaces. It provides a unified interface that allows communication between heterogeneous databases in different levels of database function model, and it can also offer consistent support for relational database. Furthermore, JDBC can be used to send SQL statements easily to various relational databases, there's no need to write different applications for each database, and the program can run on different platforms [10].

3.2. Design Concept

The ultimate goal of design the middleware architecture is to accomplish the users' transparent sharing of heterogeneous database. In view of that XML has obvious superiority condition in the field of data storage and data exchange, CORBA middleware can effectively avoid the problems which brought by the underlying platform inconsistent, JDBC can provide heterogeneous communication with a convenient method. The method of the middleware architecture design idea is as follows:

First of all, all requests take XML as its medium of resource transformation; users do not need to know the specific structure of the data source, and issue the requests directly. Secondly, the hybrid database middleware sends the user's requests to query, integration and transformation, distributes it to CORBA for further processing. And then, JDBC provides opportunities to realize the underlying database access and manipulation. What's more, the databases get the request and process the data in a certain way. After having completed the entire rendering process, the results of database operation will be returned to the middleware; the middleware integrates and processes it further and ultimately in a unified form for feedback to the requestor. The details of all the process is transparent to users, and provides a unified interface for them.

3.3. Overall Architecture

The entire database middleware architecture consists of three layers, including application, middleware and database layers. Heterogeneous database access middleware layer sits between the application layer and the database layer; it aims to integrate information of heterogeneous data sources, shield heterogeneity problems between the databases, and provide support for transparent access to the application layer. It is mainly responsible to pass the requests of application layer to the database layer according to their own logic, receive the results set returned by the database layer, and return a final result to the application layer. The overall architecture diagram is shown in Fig. 1 Overall design:

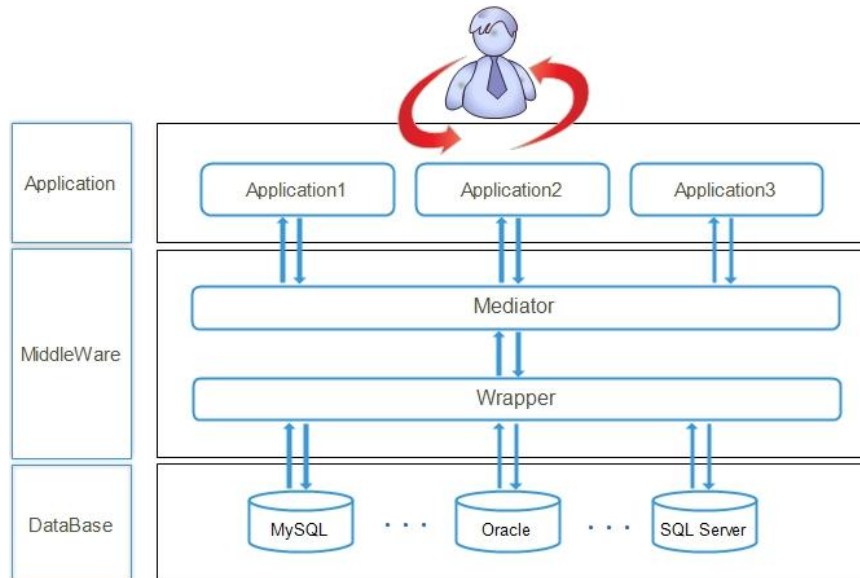


Fig. 1 Overall design

- The application layer is primarily responsible for working with the interaction between the user and middleware, and there is no need to know the specific structure of each data source. After obtaining the user's requirements, it sends the requests to the middleware layer directly in the form of different packaging, and feedback the results to the user.

- The middleware layer is the core of the architecture; it is upon the database, which can make transparent accessing heterogeneous databases. The main subject of this layer is to process data from the application layer and interact with the underlying database, lastly, it handles the message that is returned and presents to the application layer in a fixed unified format.

- The database layer is at the bottom of the architecture. Due to the database system is different, there's a variety of data storage forms, namely heterogeneous data sources. It is mainly charged with receiving data access request from middleware layer, making the appropriate treatment according to the request and sending the results back to the middleware.

3.4. Detail design

The middleware layer of the hybrid database is mainly composed of two parts: the mediator and the wrapper, they complete the major function of the middleware. Wherein the mediator is a kind of software structure that does not store any actual data, a query can be transform into another or more than one

efficient evaluating query with it after the user submits a query, and then, according to the integrated treatment of the data source, the mediator creates a summary and displays it back to the user. However, the wrapper is the core of database integration middleware and the key to achieve, which ultimately responsible for interacting with heterogeneous data, packaging the heterogeneous data sources, so as to achieve the consistency of the operation, the transparency of the data access and location transparency.

3.4.1. Mediator

Mediator is mainly responsible for data query, integration and processing, according to a certain standard, which maps the request data and converts it into a unified XML format. Next, the data will be transmitted to the wrapper for packaging and interact with the underlying database.

In this paper, the mediator is divided into data query, data integration and data conversion, in order to facilitate future maintenance and the wrapper's access. Its internal structures are shown in Fig. 2. Mediator architecture:

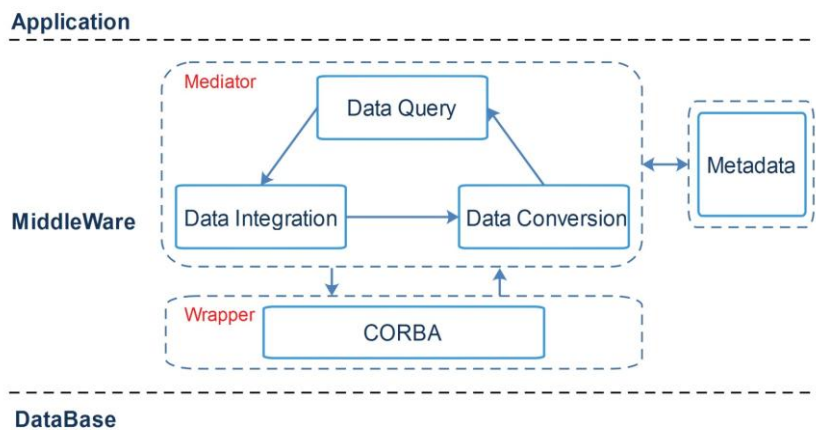


Fig. 2. Mediator architecture

- The essential definition of Meta Data is information about the data in a database, is a description of data, information and knowledge [11]. In this article, the metadata stores the corresponding relationship and transformation rules between the data sources and data integration, saves the data mapping relationship from heterogeneous data sources, which can form a global data dictionary. After the request information is obtained, it's a good idea to traverse

the data dictionary, fetch the corresponding data and convert into a unified format for further processing.

- The mediator receives the data from application layer, interacts with metadata to review whether there are data type to meet the user’s request. And then, the mediator continues to execute a set of operations, converts the user’s query into a sequence of heterogeneous database access, and eventually sends it back to the wrapper in a uniform format.

3.4.2. Wrapper

The wrapper is primarily interacted with database by CORBA; the basic principles of CORBA are intercepting the request data, finding out service object to realize the request, passing parameters and methods to service object, and then returning the results.

This section is divided into CORBA client and CORBA server. The CORBA client is mainly responsible for defining the data interface and calls to CORBA service object. However, the CORBA server aims to process the interaction and access to heterogeneous database with the client. This paper cites the JDBC technology to realize the connection and operation between CORBA and database. The main architecture of wrapper is shown in Fig. 1 Wrapper architecture:

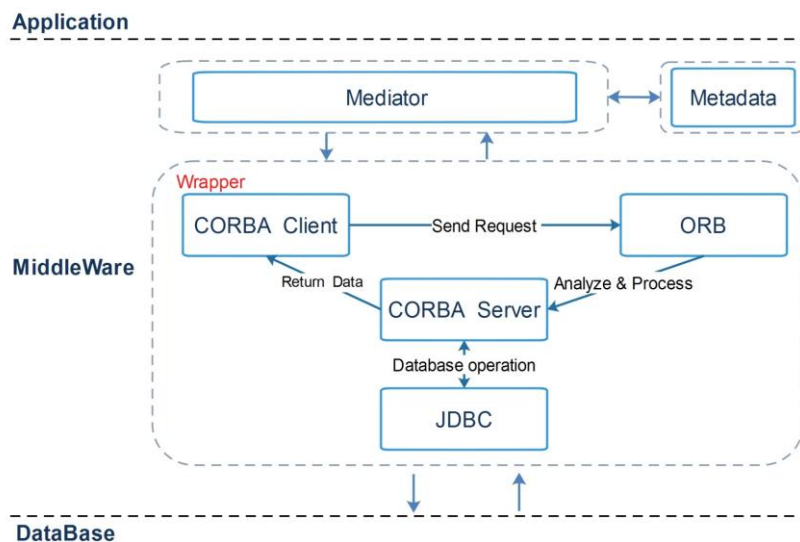


Fig. 1 Wrapper architecture

CORBA client mainly includes the definition of the interface and how to make calls to CORBA service object. An object's interface is defined through the CORBA Interface Definition Language (IDL) that hides the underlying object implementation, and ORB is the basis and core of CORBA. The CORBA client receives the service request sent by the mediator, analyzes the request to make a CORBA calls, and sends it to CORBA server through ORB, so as to complete the mapping on the server of the request. However, CORBA server uses Java programming, JDBC technology and standard SQL language to access heterogeneous database.

The whole process of wrapper is as follows: Firstly, the client receives the message which delivered from the mediator, and then sends access request to the ORB after parsing. Secondly, the ORB receives and analyzes requests and sends the information to CORBA server so that it can do further legitimacy verification and resolution. Thirdly, after the corresponding heterogeneous database through JDBC, the database passes the results of the query processing to CORBA client and finally encapsulated into a unified format to send back to the mediator, thereby completing the entire treatment process.

4. Conclusion

In this paper, as a practical matter, in order to solve the problem of heterogeneous database resource sharing, a hybrid database middleware architecture schema based on CORBA and XML is given. The schema shields the differences between the different data storage systems and environment, provides users a unified view. The users do not need to know about the knowledge of various database systems; as long as you submit your requirements without data exchange and data transmission, you can get information they want from heterogeneous data sources, so as to make transparent accessing heterogeneous databases become possible.

However, our research work is just a preliminary architecture of hybrid database middleware. Some achievements have been made in the process; however, there is much to be improved. Since some relating researches are on the first step, there is a series of further work to be done more carefully in the future.

Acknowledgements

This work is supported by the People's Livelihood Science and Technology Plan of Qingdao (14-2-3-62-nsh); Science and Technology Development Plan of Shandong Province (2014GGX101005); Self Innovation and Achievements

Transformation of Shandong Province (No.2014CGZH0708); Major Projects of Shandong Province (No.2015ZDZX05002).

References

1. YE WEI, LE JIAJIN. Query Optimization in Distributed Database Middleware [D]. Donghua University, January, 2016.(In Chinese).
2. WANG Rui, CHEN Qing-kui. Research and implementation for hereogeneous database integration[A]. Computer Engineering and Design, Nov, 2008.
3. Wei Ye, Mei Wang and Jiajin Le. Query Execution Optimization Based on Incremental Update in Database Distributed Middleware. ICA3PP 2015.[C] Swizerland: Springer International Publishing, pp.257-270, 2015.
4. MISHIMA T, NAKAMURA H. A proposal of new dependable database middleware with consistency and concurrency control[C]. 13th IEEE International Symposium on Pacific Rim Dependable Computing, 2007:334-337.
5. ZHANG YANG, ZHU DUHUI, LI JUN. Design and Implementation of Middleware based on CORBA and XML. Microcomputer Information[D], 2007, 23(11) : 243-245.
6. C.R.OZANSOY, A.ZAYEGH, A.KALAM. Interoperable CORBA Middleware Design for Substation Communication Systems[C]. Eight IEEE International conference, 2004.
7. A.Quamar, K.A.Kumar, and A.Deshpande. Sword: Scalable workload-aware data placement for transactional workloads [J]. In Proc. Of the 16th Intl. Conf. on Extending Database Technology, EDBT'13, pages 430-441, New York, NY, USA, 2013.ACM.
8. Banerjee S, Krishnamurthy V, Krishnaprasad M. Oracle8i- the XML enabled data management system[A]. 16th Data Engineering International Conference[C], 2000:561- 568.
9. ZHANG CHI. A Study of Database Exchange Technology in Isomeric DataWare Based on XML. Computer Knowledge And Technology. 2008, pp.1637-1639.
10. S. White, M. Fisher, R.Catel. JDBC API Tutorial and Reference, Second Edition[J]. Addison-Wesley, November, 2001.
11. Guo Chao, Prof. Ning Hong. Research and Exploiture of the Integrated Query Tool for Distributed Heterogeneous Database Based on Metadata. National University of Defense Technology[D]. April, 2009.
12. PI Yingying, Prof.CHEN ChunLing. Design and Implementation of DataBase Access Middleware on Massive Data Management[D]. Nanjing

- University OF Posts And Telecommunications. February, 2013. (In Chinese).
13. J.Shute, R.Vingralek, B.Samwel, B.Handy, C.Whipkey, E.Rollins, M.Oancea, K.Littlefield, D.Menestrina, S.Ellner, J.Cieslewicz, I.Rae, T.Stancescu, and H.Apte. Fl: A distributed sql database that scales. Proc. VLDB Endow. ,6(11):1068-1079,Aug.2013.