

## Dynamic Integration System for Heterogeneous Database Based on S2SH

Zhenyou Zhang

Collage of Information Engineering  
Hebei United University  
Tangshan, China  
e-mail: youzhenadd@163.com

Ziqi Zhao

Collage of Information Engineering  
Hebei United University  
GangYao Sub-branch  
Tangshan commercial Banks  
Tangshan, China  
e-mail: zzqbf@163.com

Zhi Cao

Collage of Information Engineering  
Hebei United University  
Tangshan, China  
e-mail: 526818718@qq.com

**Abstract**—Between each node database there must be different in the distributed heterogeneous database system, so how to resolve these different (It's said that realize the mutual transformation between heterogeneous databases) was an basic problem in the heterogeneous database system. This paper analyzes the system structure and the conversion mechanism for the distributed heterogeneous database and the realization of the conversion module theoretical including the conversion information extraction module, structural conversion module, structure formation module and put forward a concrete realization solutions.

**Keywords**- system struct; S2SH;dynamic integration; conversion mechinism;middle type

### I. INTRODUCTION (HEADING 1)

In the current enterprise mergers and takeovers and rapid development process, database system of once more integration had important meaning. Heterogeneous database was a collection of databases, it distribute in different nodes and had non homogeneous. Both for the heterogeneous database query, synchronization, or other operations, it must toke these databases integrated into a system in the middle, and the integrated does not change the original system of independence. And the dynamic integration of heterogeneous database system was an important method to solve the integration of once more database system.

The integrated system designed based on browser application form, the system expansion and operation has a good advantage. S2SH (Structs2, Spring, Hibernate) and.net is the excellent development method, and use S2SH could better adapt to the application of multiple platform.

### II. SYSTEM STRUCT

#### A. The System Struct

The system used B/S structure, show in figure 1, and the browser/application server/database server. The browser used to show the information to user, and the application server used to realize the service of user(include Web service, bossiness service etc...), and database server used to save data(include oracle,SQLServer,DB2,Mysql etc...).

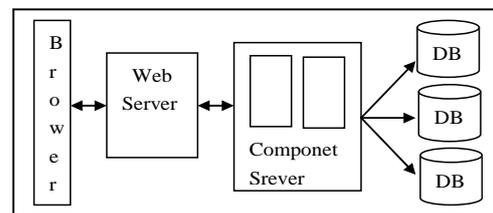


Figure1. System structure diagram

#### 1) Data layer (Database Server)

It could be more relational data source, also could be the other data sources, and was used to save data.

#### 2) Business logic layer (Application server)

The multiple machines constitutes virtual host after affairs manager by physical distribution, and virtual host distribution with multiple servers, and each server consist of multiple service composition. Service is the embodiment of business logic, it also accept the user's the basic unit service request. Business logic layer to realize the connection to the data layer, and response express logic layer client connection requirements at the same time, and to interact with express logic layer.

#### 3) Express logic layer (Brower)

Express logic layer was to interact with the system of the place for user, and was also responsible for data display form, and receiving data. It could call the core service processing in the business logic layer, and show out the treatment results.

#### B. S2SH framework realization principle

S2SH framework was the open source J2EE framework. According to the function of application logic would be divided into three layer structure: representation layer was Struct2 framework, business logic layer was Spring framework, and data persistence layer was Hibernate framework, between each layer seamless link, and support the corresponding business component.

#### 1) Struts2.

System uses Struts2 realize MVC design pattern. Struts2 will a Web system program was divided into three parts, namely: model, view and controller. Model by the EJB components, Java Bean accomplish specific business composition; View by the JSP file was composed; Controller to realize by Action.

Struts2 frame to realize the principle of MVC design pattern as shown in figure 2:

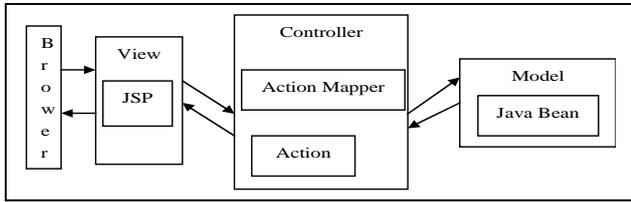


Figure.2 Schematic of truts2 achieving to MVC

Struts2 framework consists of three parts, the three parts were: core controller(FilterDispatcher), business controller and the user to achieve the business logic of the component. In this three parts when, only Struts2 framework provided the core controller FilterDispatcher, and the rest of the business controller and business logic components need the user to achieve.

Struts2 the general framework of process was as follows:

- 1) Browser to send request.
- 2) Core controller FilterDispatcher according to request decided to call the right Action.
- 3) The Web Work interceptor chain automatic to request application general function.
- 4) Callback the execute Action method, this execute method to retrieve the user request parameter, and then perform a database operation, and it can be the data to save to the database, also can search information from the database. In fact, the Action was just a controller; it would call business logic components to handle the user's request.
- 5) The execute Action method to deal with the information; it will be output to the browser.

2) Spring.

The Spring act as the management container role in S2SH framework; Spring can greatly reduce the coupling. Struts were responsible for the control of Service (business logic processing class), so as to control the Service life cycle, so that depends on the strong couple between layer and layer. The Spring framework serves as a control object of Action (the Struts) and Service type of function, the relationship between the two was loose, and it used the Ioc spring mechanism (control inversion and rely on injection) .

Control reversal: it used a container control the relationship in program, and the non-traditional realized by program itself.

Injection of dependencies: the dependencies between components decided by container in operating period, that the container would dynamic for dependency relationship among components.

Transaction processing: JDBCTemplate submitted the affairs of success; exception handling was through the Try/Catch in the past, but in the Spring. The container integrated the TransactionTemplate, it wrap the all of the transaction function, including abnormal affairs rolled back, the operation successful data submitted to the complex business functions. These were management by the Spring container , it also have very good management control to business.

The Hibernate was also to business management, Hibernate in business management was through the SessionFactory to create and maintain Session. And Spring were integrated the SessionFactory configuration, it does not need to hibernate through. CFG. XML to set SessionaFactory. So it could use Spring to manage the business. It avoids the data operation every time when the business would get Session examples to start affairs/submit/rollback transaction and the Try/Catch operation. This was also the AOP mechanism in very good application. On the one hand the development business logic more clearly, professional division of labor was easier to manage. On the other hand, the application Spring AOP isolation reduces the program's coupling and it may be improve the code reuse degree in different application.

3) Hibernate.

Hibernate was an open source object relation mapping framework, it was very lightweight object encapsulation to the JDBC, and make Java programmers can follow one's inclinations of that the use object of the programming thinking to manipulate database. Hibernate could be applied in any use JDBC occasions, and could be use in Java client program, also can be use in Servlet/JSP Web applications. The most revolutionary significance, Hibernate could instead of CMP in application of EJB in J2EE to complete the burden of the persistent data.

III. THE INTEGRATION OF SYSTEM REALIZATION

Distributed heterogeneous database integrate system have three modules: structure information collection module, structure conversion module and data display module. The function in information collection module was to extract the node database table structure information. The function in structure conversion module was to some type of table structure information into the same type structure information database table. The function in data display module was shows the system of data table to the user.

A. Struct information acquisition module

This module mainly realizes extract structure information of user data table from each node database, which was the each field type information in user table. Because of the heterogeneity of the heterogeneous databases, it has differently method from the different database. Extraction of flow chart as figure 3.

Here in SQLServer2000 and Mysql5.0 as an example to illustrate the key SQL statements to extract table structure information. Mysql and oracle three database field type as is shown in table I .

TABLE I. DATA TYPE

	Database Type			
	Mysql	SQLServer	oracle	Middle type
1	integer	int	number	int
2	float	float	number	int
3	double	double orecision	number	int

	Database Type			
	Mysql	SQLServer	oracle	Middle type
4	char(n)	char(n)	char(n)	char(n)
5	varchar(n)	varchar(n)	varchar(n)	varchar(n)
6	datetime	datetime	date	datetime

For SQLServer2000 relational database, the user data table structure information stored syscolumns system table and systypes system table in the user database. Specific SQL statements for: "select syscolumns.name, systypes.name, syscolumns.isnullable, syscolumns.length from syscolumns, systypes where syscolumns.xusertype = systypes.xusertype and syscolumns.id = object\_id('"+users table name+"')".

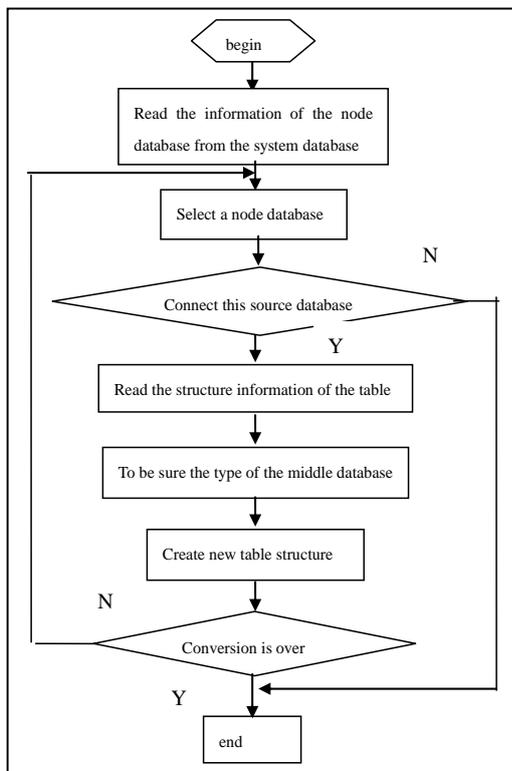


Figure. 3 extract the structure information from user table

For Mysql5.0 database, its user database table structure information stored in columns system tables in the database information\_schema. First information\_schema database must be connected, and then read the table structure information. Specific SQL statements: "select column\_name, data\_type, character\_maximum\_length, is\_nullable from columns where table\_schema="+database name+" and table\_name="+users table name+"".

Because the expression list structure information different, these information need corresponding conversion for show the users.

B. Struct conversion module

Heterogeneous database essence is the transformation of a node database user data in the table structure and data, in other heterogeneous database on reconstruction.

In different manufacturers development database system, used to define field type instructions are different. The user table information in Mysql and SQLServer shown in table II.

TABLE II. TABLE STRUCT INFORMATION

	User Table Information		
	description	SQLserver	Mysql
1	Database	Current database	information_schema database
2	Field name	syscolumns.name,	column_name
3	Data type	systypes.name	data_type
4	Is null	syscolumns.isnullable	is_nullable
5	Field length	syscolumns.length	character_maximum_length

The basement of heterogeneous database conversion mechanism was collecting the information of the conversion. In order to acquisition to the structure information from user data table in node database, this study adopts metadata programming approach, that was, according to the different characteristics of the heterogeneous database, read the user data table structure information. This was the premise of the data conversion.

Realizing the transformation between heterogeneous databases, first acquisition to convert database user data tables of information, such as table structure, table structure of storage location and other information. And then conversion a database structure to another heterogeneous database, and last it's replication the user data.

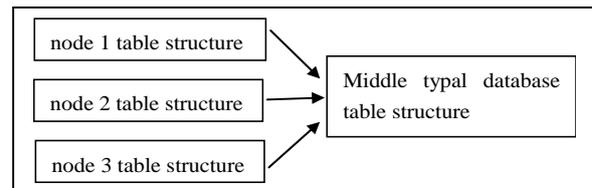


Figure.4 Structure conversion diagram

The key of the transform mechanism of heterogeneous database table was the structure conversion between heterogeneous databases. It was well known that the table structure column type of presentation between heterogeneous databases was not consistent different. Because of the type of heterogeneous database was varied, how to transformation in different between heterogeneous databases conversion was a complicated problem. Usually there were three conversion ways, one was opposite each other, and this way was the most complex conversion. Another way to convert structure to the SQL statement for undertake unity, as a result of each database SQL language has manufacturer to expand, making the changeover job more difficult. Another way is converted into an intermediate database undertake unity, although look

more complex, but is a practical method. The conversion diagram as shown in figure 4 shows.

*C. Implementation of the structure conversion*

This module mainly realizes structure conversion of the user table. In extracting the original structure information of the user table from the database users table ,it has to conversion the field data information structure the in user table to middle type data types, and database structure for some without the corresponding data types require special treatment. Specific conversion process figure 5.

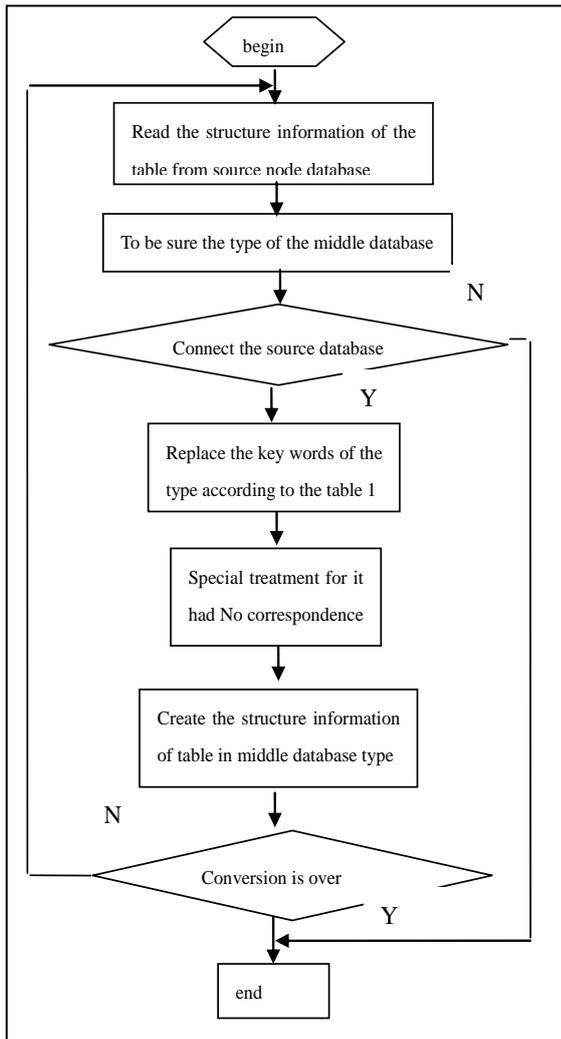


Figure.5 conversion the structure information

IV. SUMMARY

Distributed heterogeneous database integration was an important problem in the management of information; the traditional database reconstruction has many disadvantages. In this paper, heterogeneous database integration mechanism, resolves the database system dynamic change, and choose some application database as a conversion among media, improves the system practicability. This study hopes to heterogeneous database data integration study provides some new ideas.

REFERENCES

- [1] Y.Zhou, W.Xu: heterogeneous database system structure. Beijing: science press, 2007.
- [2] P.Y. Shao: Distributed database systems and its application. Beijing: science press, 2008.
- [3] Bernstein.P, Shipman D and Rothnie J. "Concurrency Control in a System for Distributed Database". ACM Trans on Database System, Apr, 2007.
- [4] Ceri, S. "Distributed Databases-Principle and System". McGraw-Hill, Aug, 2006.
- [5] Raiger, I.L. "Transactions and Consistency in Distributed Databases". IEEE Transactions on Software Engineering, June, 2006.
- [6] S. Wilbur, B. Bacar. "Building distributed system with remote procedure call". Software Engineer Journal. Proceeding of SPIE, Bellingham, Washington, 2008.