第 34 卷 第 4 期 计 算 机 工 程 2008 年 2 月
Vol.34 No.4 Computer Engineering February 2008
·软件技术与数据库· 文章编号：1000—3428(2008)04—0057—03 文献标识码：A 中图分类号：TP392

分布式异构数据库迁移系统的设计与实现

熊 辉[1,2]，刘彦峰[1,2]，郭大庆[1,2]
(1. 中国科学院新疆理化技术研究所，乌鲁木齐 830011；2. 中国科学院研究生院，北京 100039)

摘 要：介绍分布式异构数据库的相关概念和特点，基于 B/S 模式设计分布式异构数据库迁移系统，将 JDBC SQL 数据类型作为各种异构数据库数据类型的中间模型，实现整体和部分迁移。说明了数据转换规则、数据迁移实现方法、转义处理、大型数据字段处理、迁移过程、断点续传。项目实例表明该系统具有较好的可行性和实用价值。

关键词：分布式异构数据库；数据库管理系统；数据转换；转义处理；迁移过程

Design and Implementation of Distributed Heterogeneous Database Migration System

XIONG Hui[1,2], LIU Yan-feng[1,2], GUO Da-qing[1,2]
(1. Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi 830011;
2. Graduate School, Chinese Academy of Sciences, Beijing 100039)

【Abstract】This paper introduces the conception and features of Distributed Heterogeneous DataBase(DHDB), designs distributed heterogeneous database migration system based B/S model, which adopts JDBC SQL data type as the middle one of various data types, implements holistic and partial migration, and introduces type conversion, data migration method, transferred meaning and large data field management, migration process, as well as netants. The feasibility and practicality are proved in project cases.

【Key words】Distributed Heterogeneous DataBase(DHDB); database management system; type conversion; transferred meaning management; migration process

1 分布式异构数据库

分布式异构数据库是数据库技术与网络技术相结合的产物，即多个数据库系统的集合，可以实现数据的共享和透明访问，每个数据库系统在加入分布式异构数据库系统之前已经存在，拥有自己的数据库管理系统DBMS，具有物理分布性、场地自治性以及场地之间协作性[1]。分布式异构数据库的各个组成部分具有自治性，实现数据共享的同时，每个数据库系统仍保有自己的应用特性、完整性和安全性控制。异构性主要体现在：计算机体系结构的异构，基础操作系统的异构，DBMS本身的异构。分布式异构数据库系统的目标在于实现不同数据库之间的数据信息资源、硬件设备资源和人力资源的合并和共享[2]。

分布式异构数据库系统具有分布数据库和异构数据库的双重特点，实现了数据分布性和逻辑整体性，优点主要有：灵活的体系结构，适应分布式的管理和控制机构，经济性能优越，可靠性高，可用性好，局部应用的响应速度快，可扩展性好，易于集成等[1]。由于数据库系统的存储模式，类型定义等不尽相同，因此集成应用时就必须消除各数据库的异构模式，方便数据的统一管理。

2 分布式异构数据库迁移系统的设计与实现

系统基于B/S模式设计，采用Java与数据库接口规范(Java Database Connectivity, JDBC)。JDBC API定义了若干Java中的类，表示数据库连接、SQL指令、结果集、数据库元数据等。它允许Java程序员发送SQL指令并处理结果，通过驱动程序管理器，JDBC API可利用不同的驱动程序连接不同的数据库系统[3]。

2.1 系统总体结构

分布式异构数据库迁移是自下而上的数据集成方法，采用异步分工合作的 4 层协议，如图 1 所示。

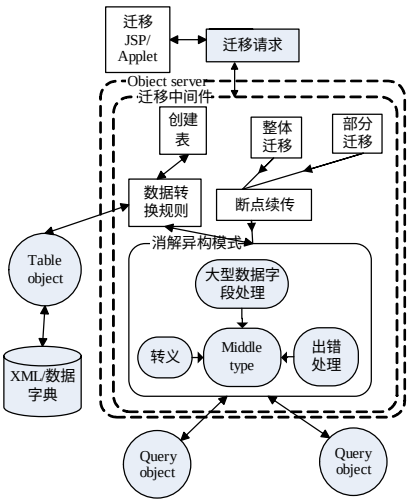

图 1 系统结构

基金项目：国家科技型中小企业技术创新基金资助项目(03c26216501196)

作者简介：熊 辉(1981－)，女，硕士研究生，主研方向：数据库技术，电子政务；刘彦峰、郭大庆，硕士研究生

收稿日期：2007-03-25 **E-mail：**xionghui04@mails.gucas.ac.cn

第 1 层有底层独立自治的数据库的访问对象 Query object 和各关系型数据库。Query object 载有各个数据库商家开发的针对各自数据库的 JDBC 驱动，实现数据库的连接和数据访问。各 DBMS 运行于各自的数据库应用系统中，是产生异构数据库的原因；第 2 层消除异构模式，是异构数据库得以实现的基础，包括：消除命名冲突，格式冲突，结构冲突，数据冲突，保留关键字冲突，大字段冲突等，有转义处理、大型数据字段处理和出错处理功能模块；第 3 层是数据库转换规则和断点续传，定制了各个数据库之间数据类型的映射关系，是异构数据库得以运行的依赖；第 4 层是异构数据库的功能层，主要实现创建表、整体迁移和部分迁移，在第 2 层和第 3 层的支持下实现数据的共享和透明访问。

另外，Table Object负责读取、组织数据字典信息，系统将数据字典纪录组织在XML文件里，Object Server通过 Table Object的界面读取连接数据库所需的信息，组织信息成为动态数据字典。Query Object的工作就是利用JDBC Driver连接数据库、查询数据库，将查询结果返回给Object Server[4]。

2.2 数据转换规则

数据类型转换规则是数据迁移的首要问题。每个 DBMS 都定义了一套自身的数据类型，随着数据库系统的发展和版本的升级，数据类型的种类也不断增多，但不论数据类型如何变化，其功能都是满足用户的数据处理基本要求，这些具有共性的方面，给系统间的数据转换带来了可能和方便。不同数据库的数据类型存在差异，其自身定义和扩充之间的区别，给系统间的数据转换带来困难。例如，DBMS 返回的日期和时间数据格式在各个 DBMS 中有很大的不同，有些系统以 8 字节整数格式返回日期和时间，另外一些以浮点数格式返回。所以，异构数据库数据类型转换的关键是找出其中的对应关系。

数据类型转换有 2 种方法：(1)定制类型映射表，设计多个相应的双向数据转换程序，解决不同的数据类型匹配问题；(2)建立全局统一的公共数据模式，利用 JDBC SQL 数据类型定制一一映射关系表。系统采用第(2)种方法，通过 JDBC SQL 数据类型作为各种数据库数据类型的中间模型，完成数据转换和迁移。由于 XML 具有规范化无规则数据特性，因此数据类型映射关系表由 XML 支持完成转换。对于数据库迁移来说，定制转换规则很关键，系统主要是用 XML 结合 JDBC 的技术手段解决不同种数据库之间的迁移。

存储在数据源中的数据所属类型为数据源数据类型或 SQL 数据类型。驱动程序在 JDBC SQL 语法及驱动程序数据类型中也定义了一套数据类型，即 JDBC SQL 数据类型(以 SQL 前缀开头的数据类型)。

每个驱动程序负责映射特定数据来源的SQL数据类型到JDBC SQL数据类型标识符。因此，不同的数据源在进行数据转换的过程中可以通过JDBC SQL数据类型标识符作为基准来得到数据类型的缺省映射关系。驱动程序通过函数java.sql. Database Meta Data.get TypeInfo方法返回数据源的SQL数据类型和JDBC SQL数据类型的映射关系。在得到各个数据库的映射关系表后可根据转换原则：数据源的SQL数据类型到JDBC SQL数据类型最后到数据目标的SQL数据类型，完成类型转换。以SQLServer 和Oracle数据库为例，表 1 显示了部分SQLServer数据类型[5]、JDBC数据类型值、Oracle数据类型 3 者之间的对应关系。

表 1　数据类型的对应关系

SQLServer数据类型	JDBC 数据类型值	Oracle数据类型
uniqueidentifier /Nchar/char	1	CHAR
Intidentity 或　int	4	NUMBER
float	6	FLOAT
tnntext /text	-1	NCLOB/ CLOB/LONG
varbinary	-3	(BLOB)/RAW
bit	-7	NUMBER
…	…	…

2.3 数据迁移实现方法

数据迁移是指在不同数据库之间将源数据库中指定表的数据迁移到目标数据库的指定表中。系统利用动态 SQL 语句将数据记录逐条取出后赋值到指定的数据库字段中，比较灵活和稳定。动态 SQL 语句能对所有关系数据进行处理，可利用它分条读取源数据库中的数据，将这些数据以字符串形式存入变量中，然后插入目标数据库的表字段中实现数据迁移。在处理过程中关键是字段值字符串的处理。

2.3.1 转义处理

迁移过程中的关键是处理字段值字符串，字符串中转义字符的处理尤其重要，它直接影响记录能否被执行。例如在 SQL 语句中“ ’ ”用来间隔字段，但如果在源数据库中读出有“ ’ ”的字符串，就不需转义处理。如对于 SQL 语句：insert table (column1,column2column3)Values ('asdfasdf','asdfasdf'', '''asdf''asdf'''),必须辨别出哪些“ ’ ”是字段间隔，哪些是字符，在经过转义处理后得到 insert table(column1, column2column3) Values('asdfasdf','''asdfasdf''','''''asdf'''''asdf''''')。系统中由 ErrorDispose 类完成。其中，Exminer()成员将“ ’ ”替代成 “ ’’ ”，过程如下：

```
public String Exminer(String S) {
    String s = new String();
  if (S != null) { s = S.replaceAll("'","''");   }
    else { s = null;}
  return s;}
```

2.3.2 大型数据字段的处理

在迁移过程中，必须对大型数据字段进行特殊处理，如 SQL Server 中的数据类型 tnntext 和 text, Oracle 中的 NCLOB 和 CLOB 等。以 CLOB, BLOB 为例，系统中碰到 CLOB, BLOB，先在 CLOB, BLOB 字段写入空值，然后在数据库中读出包含这个大字段的记录，再修改字段的内容来存取 CLOB, BLOB.类 handle_field_type 中提供了所有大型数据字段的处理，包括 insert_blob(), insert_clob(), read_blob(), read_clob()等方法。

2.4 迁移过程

迁移之前，要创建目标数据库、数据库表、数据库的表间关系及存储过程等。系统利用 JDBC 的 DatabaseMetaData 接口从源与目标 DBMS 中读取 DBMS 所支持的数据类型，找出源与目标数据类型的对应关系，将该对应关系存入指定的 XML 文档，定制类型转换规则，其目标是解决异构数据库的数据类型转换，且可以一次定制，多次使用。

迁移分为整体迁移和部分迁移。整体迁移将整个数据库复制到其他数据库中；部分迁移将一个数据库表中的一些字段复制到其他数据库表中。部分迁移可分为单表到单表的迁移，即单个表中的一些字段的所有记录复制到另外数据库表中；单表到多表的迁移，即将一个表中的任意个字段分别复制到其他数据库表中去。在迁移过程中如果遇到数据类型不匹配，则要根据转换规则进行类型转换；如果没有现成的类

型转换规则，则要定制类型转换规则。当源数据库结构及数据迁移到目标数据库后，根据源数据库中的表间关系，建立目标数据库表间关系。

数据迁移过程以从 Oracle 迁移到 SQLServer 为例：(1)创建表模块，通过调用数据转换规则库，创建与源数据类型相兼容的目标数据库，包括系统表、表间关系和数据表。(2)加载 Oracle JDBC 驱动的 Query object 对象，通过 SQL 语句 SELECT * FROM SCHEMANAME.TABLE 读取数据库结果集。(3)取出一条记录，将结果集中每条记录由 RESULTSET.GET××()得到非字符串字段值 *x*，再将这个值转换成 JDBC 类型字符串，由××.toString()完成，得到字符串 *y*。(4)将 y 经过转义模块过滤后，加到 INSER SQL 语句中，形成新的 SQL 语句 INSERT INTO SCHEMANAME.TABLE VALUES(*,*,*)。(5)由已加载 SQLServer JDBC 驱动的 Query object 存储到目标数据库中去。

实现整体迁移，必须在目标数据库中建立与源数据库中表结构相同的表，才能将源表中的数据迁移到目标表中[2]。部分迁移则需要选择数据库，系统访问数据库读取元数据信息，如读取表名、读取字段名等，表和字段等元信息发送到用户界面由用户选择，根据选择的表和字段，经过数据转换规则的有效性检测，就可以用整体迁移类似的方法进行。

2.5 断点续传

在异构数据库迁移过程中，难免出现死机、停电、网络中断等内在和外在的原因，因而需要断点续传性能支持。数据库迁移的断点续传不同于文件传输的断点续传。文件传输以字节为元操作，它的续传可以从每个字节重新传输，而异构数据库迁移以记录为元操作，它的续传是从每个记录开始的，断开也是以记录为单位，实现代码如下：

```
if (record_counts != 0) {
        for (int i = 1; i <= record_counts; i++) {
          source_rs.next(); }}
```

3 应用实例

在测试数据库迁移时，为了充分体现测试数据的异构性，选用医院管理系统应用的 HISSYS6.0 数据库中的数据。该数据库的数据量大，数据类型丰富，如果能将 HISSYS6.0 数据库迁移到另一种 DBMS 中，就能证明系统是成功的。HISSYS6.0 系统的 DBMS 是 SQL Server 2000，即源数据库，ORACLE 与 SQL Server 2000 能够充分体现不同 DBMS 的异构性，因此，选择 ORACLE8i 作为目标数据库。以部分迁移作为应用实例：用户在操作界面上根据迁移之前创建的数据库、数据库表、数据库的表间关系及存储过程等，选择源和目标数据库，然后根据用户需求选择相应的数据库表以及所选表中的数据字段，提交以后就可以实现数据的部分迁移，源表中的数据成功迁移到目标数据表中。

通过项目实例，说明分布式异构数据库迁移系统具有较好的可行性和效率，系统的功能在应用中能够很好的实现，具有很高的实用价值。

4 结束语

分布式异构数据库迁移系统基于 B/S 模式，可利用网络中的任何终端对网络中的任何数据库进行操作，具有很高的实用价值。整个系统的功能相对强大和完善，能够满足应用需求。该系统已经应用于多个相关项目中，取得了很好的应用效果和经济效益。随着数据库技术、XML 技术、网络的不断发展，该系统也将逐步趋于完善。

参考文献

[1] 分布式数据库概述[EB/OL]. (2005-08-03). http://fineboy,cnblogs.com/archive/ 2005/08/03/206395.html.

[2] 孔祥疆. 软件开发方法与建立异构数据库使用平台模型[D]. 乌鲁木齐: 中国科学院新疆理化技术研究所, 2005.

[3] 梁陈剑, 张 威. JDBC3.0 数据库开发与设计[M]. 北京：北京希望电子出版社, 2003.

[4] 罗林球, 孔祥疆, 李 晓. 基于 CORBA/数据字典/JDBC 的异构数据库检索系统实现[J]. 计算机应用, 2006, 26(6): 91-94.

[5] Snakebin. SQL 数据类型详解[EB/OL]. (2005-07- 09). http://snakebin.bokee.com/ 2217273.html.

(上接第 53 页)

4 结束语

从实验结果可以看出，该算法具有较好的负载平衡能力，而且通过选择临近节点进行负载移动，大大降低了系统的响应时间。从第 2 节的描述中还可以看出，该算法具有较好的扩展性，并且能够适应于不同处理能力计算机组成分布式数据流系统的环境。因此，该算法能适应实际应用中建立在 DHT 为基础重叠网络上的分布式数据流系统对扩展性的要求，已在某研究所数字化工程项目支撑环境中得到初步应用。

参考文献

[1] Balazinska M, Balakrishnan H, Stonebraker M. Contract-based Load Management in Federated Distributed Systems[C]//Proc. of NSDI'04. San Francisco: [s. n.], 2004: 197-210.

[2] Ying Xing, Zdonik S, Hwang J H. Dynamic Load Distribution in the Borealis Stream Processor[C]//Proc. of ICDE'05. Tokyo, Japan: [s. n.], 2005: 791-802.

[3] Cherniack M, Balakrishna H, Balazinska M, et al. Scalable Distributed Stream Processing[C]//Proc. of CIDR'03. Asilomar, CA: [s. n.], 2003: 257-268.

[4] Stoica I, Morris R, Liben-Nowell D. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications[J]. IEEE/ACM Transactions on Networking, 2003, 11(1): 17-32.

[5] Zhang Zheng, Shi Shuming, Zhu Jing. SOMO: Self-organized Metadata Overlay for Resource Management in P2P DHT[C]//Proc. of IPTPS'03. Berkeley, CA, USA: [s. n.], 2003: 170-182.

[6] Ratnasamy S, Handley M, Karp R M, et al. Topologically-aware Overlay Construction and Server Selection[C]//Proc. of IEEE INFOCOM'02. [S. l.]: IEEE Press, 2002: 1190-1199.

[7] Gotsman C, Lindenbaum M. On Themetric Properties of Discrete Space-filling Curves[J]. IEEE Trans. on Image Processing, 1996, 5(4): 794-797.

[8] Harchol-Balter M, Crovella M E, Murta C D. On Choosing a Task Assignment Policy for a Distributed Server System[J]. Journal of Parallel and Distributed Computing, 1999, 59(2): 204-228.