

基于JDBC的异构分布式数据库访问

张少中, 王秀坤, 张志勇

(大连理工大学计算机系, 大连 116024)

摘要: 实现异构分布式数据库共享所要解决的主要问题是数据库转换和数据库的透明访问, 目前流行的实现异构数据库互联访问的两种方式是采用公共程序编程接口和专业数据库网关产品。基于JDBC的异构分布式数据库访问方法是以程序编程接口为思想的实现方案。该文详细讨论了JDBC的工作原理和基于JDBC的异构分布式数据库访问的实现方法。

关键词: 异构分布式数据库; 数据共享; JDBC; API

Research on JDBC-based Accessing Heterogeneous Distributed Database

ZHANG Shaozhong, WANG Xiukun, ZHANG Zhiyong

(Department of Computer Science, Dalian University of Technology, Dalian 116024)

【Abstract】 The most important problems of implementation of database sharing in heterogeneous distributed database system are database conversion and database accessing. There are two common methods to implement database accessing. One is common application program interface, and another is professional database gateway. Using JDBC accessing heterogeneous distributed database system is based on API. The paper bats around the principium of JDBC. It also achieves the accessing of heterogeneous distributed database system based on JDBC.

【Key words】 Heterogeneous distributed database; Data share; JDBC; API

计算机技术进入社会生活以来, 数据处理作为计算机的主要工作, 经历了几个不同的发展时期: 非文件系统阶段、文件系统阶段、集中式数据库阶段、分布式数据库系统阶段、异构分布式数据库系统阶段。在计算机网络技术不断发展的今天, 分布式数据库系统作为主流的数据系统, 正在演化为异构分布式数据库系统。

异构数据库系统是相关的多个数据库系统的集合, 可以实现数据库的共享和透明访问, 每个数据库系统加入异构分布式数据库系统之前本身就已经存在, 异构分布式数据库的多个组成部分具有自身的自治性, 实现数据共享的同时, 每个数据库系统仍保持自己的应用特性、完整性控制和安全性控制。

一般而言, 分布式数据库系统的异构性主要表现在以下几个方面:

- (1) 数据库概念模式: 数据库概念模式分层次、网状、关系和面向对象4种;
- (2) 数据库的物理模式: 不同厂商的数据库管理系统拥有不同的物理模式;
- (3) 计算机体系结构: 宿主机硬件的区别;
- (4) 基础操作系统: 不同的宿主机操作系统;
- (5) 网络连接: 不同种类的网络拓扑结构。

在这种环境下, 如何最大限度地保护用户资源, 透明地访问异构网络、计算机平台, 实现异构分布式数据库之间的连接、数据交换和数据共享, 无缝地联合使用异构信息源中的大量信息, 已经成为数据库系统研究的关键。

1 实现异构分布式数据库系统需要解决的问题

对于异构分布式数据库系统实现数据共享应当达到两点: 一是实现数据库转换; 二是实现数据库的透明访问。

1.1 数据库转换

数据库转换就是将一种数据库系统中定义的模式动态或静态转换为另一种数据库中的数据模型, 然后根据需要再读

入数据, 从而用户就可以利用自己熟悉的数据库系统和熟悉的查询语言实现数据库共享的目标。

在数据库模型转换过程中需要解决以下几个问题:

- (1) 全局数据模式。建立能够描述各LDB间的数据类型及计量单位间的对应关系, 以便向用户提供统一的数据视图(数据库、表及属性)。
- (2) 命名服务。为异构分布式数据库各成员结点提供统一的名字注册、地址查询、结点位置与配置信息等服务。这些信息都存放在一个公共服务器的名字数据库中。
- (3) 全局数据字典。描述异构数据库系统的核心, 涉及联邦查询语言定义、查询分解、查询翻译、查询优化、调度执行、查询结果集成等。
- (4) 安全控制。当多个异构的LDB集成至异构分布式数据库系统后, 应当保证LDB中的保密数据不被未授权的用户访问或进行授权以外的操作。

1.2 数据的透明访问

在异构分布式数据库系统中实现了数据的透明访问, 用户就可以将异构分布式数据库系统看成普通的分布式数据库系统, 用自己熟悉的数据处理语言去访问数据库。

实现数据库的透明访问, 首先要建立规范的应用程序接口(API), 这种可用在相关或不相关的数据库管理系统中存取数据的标准应用程序接口(API)为应用层开发者提供了一组规范化的访问异构分布式数据库的功能调用接口, 其中的关键在于: 以现有的SQL标准为基础, 定义支持异构分布式数据库访问的联邦结构化查询语言。

2 实现异构分布式数据库互联的基本方法

2.1 公共编程接口(API)

公共编程接口有两个: 第一个是客户应用编程接口

作者简介: 张少中(1969~), 男, 博士生、副教授, 主研方向: 数据库系统、决策支持系统; 王秀坤, 教授、博导; 张志勇, 硕士生
收稿日期: 2001-12-25

(CAPI), 它是一组过程库, 通常以TSR方式或DLL方式驻留在客户工作站上, 一个CAPI可装入后端专用的驱动程序以访问不同的数据源; 另一个是服务器应用编程接口(SAPI), 它提供控制服务器与客户应用请求和目标数据库之间的交互。Microsoft的ODBC、Oracle的SQL*NET、Sybase的Open Client/Open Server就是这种公共编程接口。

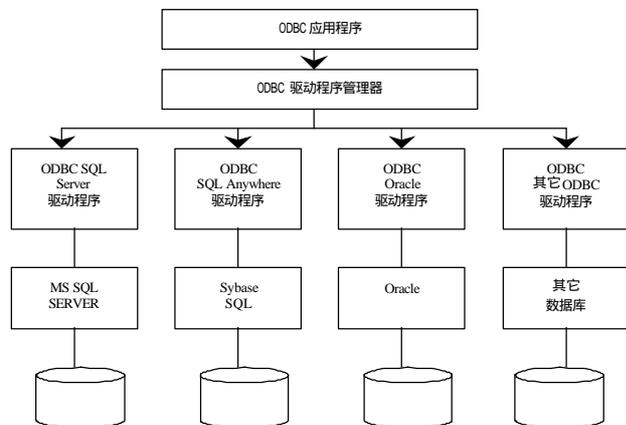


图1 ODBC功能结构图

ODBC允许一个应用程序访问ODBC支持下的不同数据源, ODBC数据源由数据及其相应的DBMS或文件管理器、操作系统和网络软件组成, 它负责为应用程序存储和管理数据。同时ODBC还为Windows开发者提供了SQL数据库访问函数调用, 屏蔽了底层数据库系统的不同, 从而简化了对数据库的访问。客户端应用程序通过ODBC与不同的数据库连接, 再通过嵌入的SQL语句就可以实现异构分布式数据库数据的转换和透明访问。ODBC的结构如图1所示。

2.2 专业数据库网关(Gateway)

专业数据库网关是一个转换器, 客户可以通过它访问网关支持的异种数据库。Oracle、Sybase等大型数据库都有自己的网关产品。

Sybase的OmniCONNECT是第一个支持全局透明访问的新一代数据库网关, 提供了在整个企业范围内不同数据库管理系统之间完全透明的数据集成, 实现在不同的SQL语句、不同厂商的数据库和数据存储位置之间的透明访问。

同时, Sybase的DirectCONNECT用于与非Sybase数据源建立联系的访问服务器, 它允许用户将其习惯使用的应用系统同关键的企业数据源集成起来, 并保证整个企业信息系统的完整。DirectCONNECT为各类数据源提供基本的Sybase Transact-SQL透明性, 支持以BD-Library及ODBC应用程序接口(API)编写的客户应用。DirectCONNECT可以独立工作, 也可以与其他中间件产品配合使用, 以实现分布数据的存取、复制、拷贝等管理功能。如图2所示。

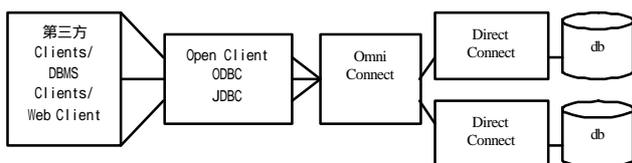


图2 Sybase数据库网关结构图

3 基于JDBC的异构分布式数据库访问

Java是Client/Server或Browser/Server等多层系统体系结构重要工具之一, 其优势主要体现在跨平台、可移植性, 非常适合异构型、分布式计算环境的要求。

JDBC(Java Data Base Connection)是一种用Java实现的数据库接口技术, 是开放数据库互联(ODBC)的Java实现。JDBC保持了ODBC独立于特定数据库的基本特性, 使用相同源代码的应用程序通过动态加载不同的JDBC驱动程序访问不同的DBMS。JDBC支持在应用程序中同时建立多个数据库连接, 各个DBMS间通过不同的URL进行标志。同时, JDBC更具有对硬件平台、操作系统异构性的支持。这样, 采用JDBC可很容易地使用SQL语句实现异构分布式数据库的访问, 为异构分布式数据库之间的互操作奠定了基础。

3.1 JDBC的功能结构

JDBC的API中设计了专门的驱动程序管理工具, 这种驱动程序管理器将JDBC API划分为两个不同的层次, 分别为: 驱动程序层(JDBC Driver API)和应用程序层(JDBC API)。JDBC Driver API是由数据库厂商或专门的驱动程序生产厂商开发的, 在使用JDBC API前应保证正确安装了JDBC Driver API; JDBC API是面向程序开发人员的一系列抽象的接口, 它使得应用程序员可以进行数据库连接, 执行SQL语句, 并且得到返回结果。JDBC功能结构如图3所示。

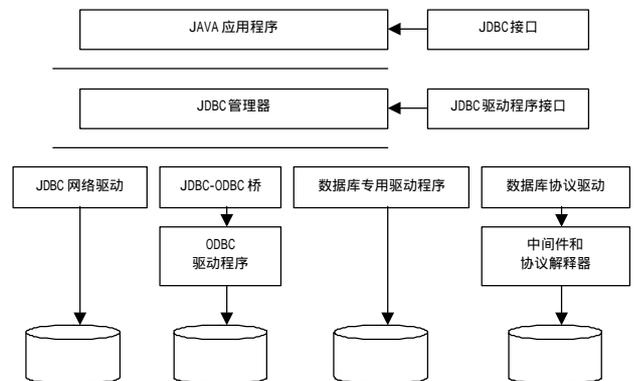


图3 JDBC功能结构

JDBC Driver API可以分成4类: (1)本地数据库专用驱动程序(Native API partly-Java Driver), 该驱动方式将用户应用程序的JDBC调用转换成对Oracle、Sybase、Informix、DB2等DBMS系统的调用, 这种驱动程序要求在客户端预先安装; (2)JDBC-ODBC桥接驱动程序(JDBC-ODBC Bridge), 该方式通过JDBC-ODBC Bridge使用ODBC来访问数据库, 因此, 所有支持ODBC的数据库都可以不加任何修改而直接与JDBC协同工作, 该方式要求在客户端安装与服务器相同的ODBC和JDBC-ODBC Bridge本地库; (3)数据库协议驱动程序(JDBC-NET pure Java Driver), 该方式将JDBC与一种通用数据库协议驱动程序相连, 利用中间件和协议解释器将该协议驱动程序与某种具体的数据库系统相连, 该方法具有良好的跨平台、连接异构数据库的特点, 但运行该程序需要第三方厂商开发的中间件和协议解释器的支持; (4)JDBC网络驱动(Native-protocol pure Java Driver), 该方式将JDBC调用直接转换成数据库使用的网络协议, 用户可以直接访问数据库系统, 该方法可以不预先进行客户端安装, 可以随Java Applet从服务器端下载。

JDBC API主要包括: Java.sql.DriverManager, 用于处理装载驱动程序并为创建新的数据库连接提供支持; Java.sql.Connection, 完成某一指定的数据库的连接; Java.sql.ResultSet, 定义指定SQL语句执行的原始结果集, 并提供对执行SQL语句后产生的结果的访问; Java.sql.Statement, 在一个给定的连接中作为SQL执行的容器。

3.2 使用JDBC实现异构分布式数据库访问的步骤

不同的JDBC驱动程序可以有不同的访问异构分布式数据库系统的方法,而其步骤则大同小异,下面以数据库专用驱动程序方法加以说明:

- (1)安装合适的驱动程序,包括一组动态连接库(DLL)文件和有关系统类库,这些文件可从SUN公司或有关数据库厂商处得到,这些文件必须安装在运行Java Application的本地客户端上。
- (2)编写程序,加载后台数据库专用JDBC驱动程序。
- (3)创建连接对象(Connection对象),完成与远端数据库的连接,连接数据库时应提供数据库主机的IP地址、连接端口、用户名及口令等信息。
- (4)创建Statement对象,包括欲执行的SQL语句,若SQL语句较长,则可以使用Prepared Statement类的对象。
- (5)执行SQL语句,将返回结果放入新建ResultSet类的对象。
- (6)处理ResultSet对象中的数据。

3.3 程序实现

在该程序中,后台数据库服务器为SUN5500,操作系统为Solaris,数据库为Oracle for Unix。

```
import java.sql.*;
class UseJDBC
{
    public static void main(string args[]) throws SQLException, class
    NotFoudbException
    {
        //加载Oracle数据库专用JDBC驱动程序
        class.forName("oracle.jdbc.driver.OracleDriver");
        //连接数据库,注明数据库的主机IP、端口号、数据库别名、
        //数据库用户名和口令
        Connection Conn=
        DriverManager.getConnection("jdbc:oracle:thin:@201.118.254.
        019:1521:oracle80","water","hlj2001");
        //创建一个SQL语句
        Statement stmt=Conn.createStatement();
        //运行从数据库rain表中查询降雨量raincub的语句并将查询结
        //果放入结果集对象
        ResultSet.rset=Stmt.execute Query("select raincub from student
        ");
        //逐个获取结果集中的数据域,输出
        While (rset.next())
```

(上接第256页)

记,位于同一节点的下一个元素指针及上一个元素指针。元素存放指针即该元素在该类元素磁盘文件存放的起始地址。

7 结束语



图3 采油厂S 2+3油层小层平面图全图

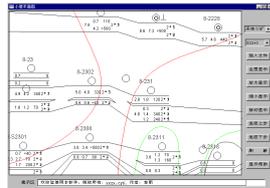


图4 采油厂S 2+3油层小层平面图放大图

该技术研究成功后,利用该技术研制了油层小层平面图在WWW上的查询,其应用效果很好:(1)该成果实现了在网上查询大型地质图件功能,克服了数据量大、处理复杂难度

```
System.out.println(rset.getString(1));
} }
```

该程序rain表中所有raincub的内容查询出来,传送到运行Java程序的本地客户端并输出显示。程序首先将合适的数据库JDBC驱动程序加载到本地客户端,然后利用DriverManager类的方法getConnection()完成与远端数据库的连接,连接时须提供欲连接的数据库主机IP和端口号,完成物理寻址,还要提供合法的用户名和口令,用以进行用户身份确定,连接成功后,使用getConnection()语句返回一个Connection类的对象,利用这个对象再进行相关的SQL操作。

4 小结

随着网络应用的不断普及,基于网络的异构分布式信息管理系统正在迅速发展。在企业级分布式数据库管理系统中对各种异构分布式数据库实现透明操作是一项重要的工作。JDBC是Java与数据库的接口规范,JDBC定义了一个支持标准SQL功能的通用低层的应用程序编程接口,这些由Java语言编写的类和接口利用不同的驱动程序连接不同的数据库系统。JDBC为访问异构分布式数据库提供了统一的方式,也为各异构成员之间的协作和多个系统之上的操作奠定了基础。利用Java的平台无关性,JDBC自然地实现了跨平台的特性,适合于异构分布式数据库应用。

未来企业信息系统的的发展将始终保持跨系统、跨平台的局面,能够提供一个独立于特定数据库管理系统的统一编程界面和一个基于SQL的通用数据库访问方法,将是未来实现基于异构系统平台的数据库应用的目标,因此,JDBC必将在Internet上的异构分布式数据库访问中发挥重要的作用。

参考文献

- 1 Shaw R.Integrating Databases with Java via JDBC.JavaWorld,1996
- 2 印 旻.Java与面向对象程序设计教程北京高等教育出版社,1999
- 3 Quan Xia,Ling Feng,Lun Hongjun.Supporting Web-based Database Application Development Database Systems for Advanced Applications.Proceedings of 6th International Conference on,1999:17-24
- 4 梁 鹰罗伟其异构数据库的数据转换在大型信息系统中的实现.计算机工程与应用,2000,36(9):103-105
- 5 张心耕张京生郑晓军.异构数据库互联的设计方法计算机工程与应用,1998,34(9)

高等技术难点,使以往只能通过绘图仪成图的大型图件可以在网上直接生成,并通过统一的浏览器查看,具有通用性;(2)该成果绘图速度快、界面友好、操作方便,并提供了放大、缩小、平移等功能;(3)通过使用该成果,提高了油田开发人员的工作效率,节约了出版和绘图等费用。成果附图见图3和图4。

参考文献

- 1 Wesley A.The Java Tutorial Object-oriented Programming for the Internet.McGraw-HillCompanies Inc.,1996
- 2 易文韬,陈颖平.Java手册北京科学出版社,1997
- 3 刘 岳梁启章专题地图制图自动化武汉:测绘出版社,1993
- 4 李鸿吉,张菊明.电子计算机制图方法及其应用.北京:地质出版社,1995
- 5 千志军编石油数学地质济南华东石油学院出版社,1992