

文章编号: 1007-2985(2009)03-0040-04

基于 JDBC 构造中间件的工作方式与实现^{*}

江凤莲, 郭华龙, 龚长斌

(龙岩学院 数学与计算机科学学院, 福建 龙岩 364000)

摘 要: 就高校教务管理系统中异构数据源之间的传输问题, 提出了基于 JDBC 构造中间件的工作方式及其实现方法。结合高校教务管理系统的实际业务情况构造中间件, 引入数据库连接池技术, 很好地解决了该系统在大量数据同时访问时出现工作效率低的问题, 提高了整个系统的数据访问效率, 具有高独立性、移植性与复用性。

关键词: 异构数据源; 中间件; JVM; JDBC

中图分类号: TP311.138

文献标识码: A

1 中间件

目前解决异构数据源之间的传输问题的普遍方法就是引入中间件技术。中间件^[1]泛指能屏蔽操作系统和网络协议差异, 为异构系统之间提供通讯服务的软件。它为平台和应用软件之间的数据交换提供高效率的数据通道和功能调用。作为平台与应用软件的中间层, 中间件一般应该具有以下的一些通用的特点: 支持分布计算, 提供跨网络、硬件和 OS 平台的透明性, 支持标准的接口, 提供接管和恢复功能, 保证事务及关键性业务不被丢失, 提供一个公共的应用通信与数据接口层^[2], 使应用程序可以很容易定位和共享应用逻辑与数据。

2 JDBC 接口技术

JDBC 是一种可用于执行 SQL 语句的 Java API (Application Programming Interface, 应用程序设计接口), 图 1 是 JDBC 的体系结构图^[3], 由一些 Java 语言编写的类、界面组成^[4]。JDBC 给数据库应用开发人员、数据库前台工具开发人员提供了一种标准的应用程序设计接口, 使开发人员可以用纯 Java 语言编写完整的数据库应用程序。通过使用 JDBC, 开发人员可以很方便地将 SQL 语句传送给几乎任何一种数据库。用 JDBC 编写的程序能够自动地将 SQL 语句传送给相应的数据库管理系统 (DBMS)。同时, 使用 Java 编写的应用程序可以在任何支持 Java 的平台上运行, 不必在不同的平台上编写不同的应用程序。其主要是由于 Java 虚拟机 JVM^[5] (JVM 的体系结构如图 2 所示) 的存在很好地实现了 Java 程序的可移植性。

2.1 JDBC 的任务

简单地说, JDBC 能完成下列 3 件事: 一是与一个数据库建立连接; 二是向数据库发送 SQL 语句; 三是处理数据库返回的结果。

2.2 JDBC 是一种底层的 API

JDBC 是一种底层 API, 它将直接调用 SQL 命令。SQL 语言嵌入 Java 的预处理器, 实现从关系数据库

* 收稿日期: 2009-02-09

基金项目: 福建省教育厅科技项目 (JA08228)

作者简介: 江凤莲 (1970-), 女, 福建连城人, 福建龙岩学院数学与计算机科学学院讲师, 在读硕士生, 主要从事计算机应用与数据库研究。

到 Java 类的直接映射. 总之, JDBC API 是能体现 SQL 最基本抽象概念的、最直接的 Java 接口.

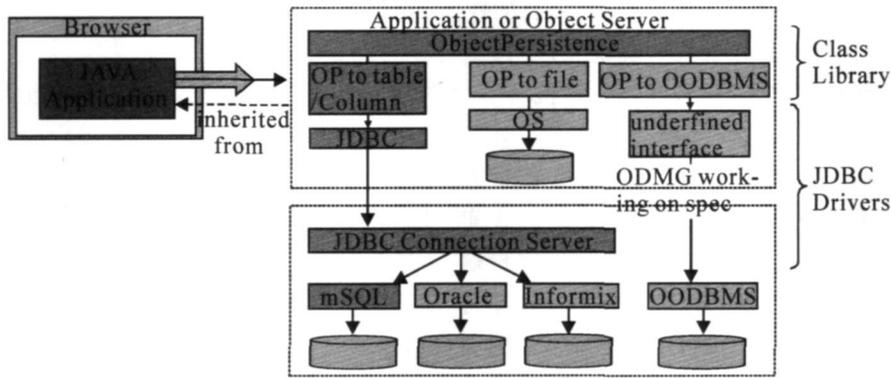


图 1 JDBC 体系结构图

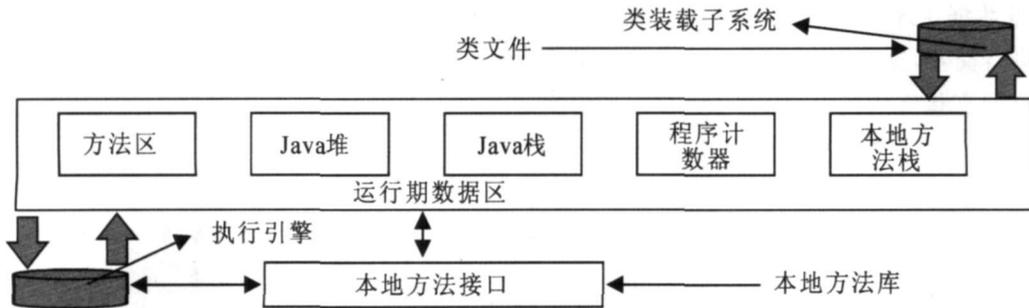


图 2 JVM 体系结构图

2.3 JDBC 技术的优势

一是通过 JDBC 技术, 用户不必锁定在特定的技术架构就能继续使用他们现有的数据库并方便地访问信息, 即使这些数据存储在不同的数据库管理系统中. 二是简化系统的应用开发. JDBC 隐藏了许多数据访问任务的复杂性, 将许多繁重的工作隐藏在一个简单的表象内, 使其易于发布和简单维护. 三是对于网络计算机的零配置. 对于 JDBC API, 不需要在客户端进行配置; 对于用纯 Java 写的驱动程序, 建立连接所需要的信息都可以通过 JDBC URL 定义, 或通过注册到 JNDI 命名服务的 DataSource 对象而得到.

由于该教务系统所涉及的 3 个系分属不同的数据库存储数据, 因此采用 JDBC 技术.

3 数据访问中间件框架设计

3.1 JDBC 驱动程序

采用 JDBC 很容易用 SQL 语句访问异构数据库, 通过 JDBC API, 只需编写一个 JDBC API, 就可以将 SQL 查询语句送往 Sybase、Oracle 或 Informix 数据库, 同时采用 Java 语言编写应用程序, 具有平台无关性, 不需要为不同的平台编写不同的应用程序. 在数据库访问中间件框架设计之前, 必须分析 JDBC 驱动程序, 它是利用 JDBC 标准建立起 Java 程序和数据源之间的桥梁.

首先, JDBC 提供了完成下列基本任务的方法: 以 URL 或注册到 JNDI 名称服务的 DataSource 对象为基础创建和管理数据源连接, 因此客户端不必进行复杂的配置; 构造 SQL 命令, 向数据源发送 SQL 命令; 提取和处理返回给 Java 应用或 Applet 的结果集.

目前 JDBC 驱动方式共有 4 类: 第 1 类是 JDBC-ODBC 桥, 再加上 ODBC 驱动程序; 第 2 类是本机 API, 部分是 Java 的驱动程序; 第 3 类是面向数据库中间件的纯 Java 驱动程序; 第 4 类是直接面向数据库的纯 Java 驱动程序.

第 1 类驱动程序只用于原型开发, 而不要用于正式的运行环境. 第 2 类 JDBC 驱动程序是本机 API 的部分 Java 代码的驱动程序, 用于将 JDBC 调用转换成主流数据库 API 的本机调用. 第 3 类 JDBC 驱动程序是面向数据库中间件的纯 Java 驱动程序, JDBC 调用被转换成一种中间件厂商的协议, 中间件再将这些调用转换到数据库 API. 第 4 类 JDBC 驱动程序是直接面向数据库的纯 Java 驱动程序, 即所谓的瘦 (thin) 驱动程序, 它将 JDBC 调用转换成某种直接可被 DBMS 使用的网络协议, 这样客户机和应用服务器可以直

接调用 DBMS 服务器。

由上述分析可知第 4 类驱动程序具有较高的性能,能够直接访问 DBMS。

3.2 JDBC 构造中间件的工作方式和实现

高校教务系统(以龙岩学院数学与计算机科学学院为例)采用第 4 类驱动方式来实现 JDBC 中间件的构造。

由于数学与计算机科学学院下属的 3 个系之间的数据库各不相同,因此采用第 4 类驱动方式实现 JDBC 构造中间件(见图 3)的工作方式为:客户端首先访问 Web 服务器,下载 Java 字节码文件,并将 Applet 相关类的字节码文件和 JDBC 接口字节码文件下载到客户端,然后与 Web 服务器脱离,接着 Applet 根据数据库地址,端口号和帐号与数据库服务器连接进行交互,这样用户与数据库服务器的交互是由浏览器直接完成的。由于 JDBC 技术的可操作性、可维护性和安全性及事务处理能力、使用效率都比较高,因此具有很大的优势。

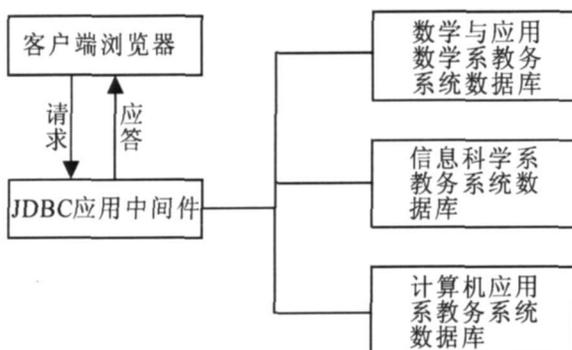


图 3 基于 JDBC 中间件的教务管理系统模型

该中间件除了采用第 4 类驱动方式外,还应包括应用程序层。应用程序层包括 4 个主要接口,即 Connection、Statement、ResultSet 和 DatabaseMetaData 接口。

为了处理大量数据访问所造成的访问效率问题,可采用连接池的方法。

当程序中需要建立数据库连接时,只需从内存中取一个来用而不用新建。同样,使用完毕后只需放回内存即可。而连接的建立、断开都由连接池自身来管理。同时,还可以通过设置连接池的参数来控制连接池中的连接数、每个连接的最大使用次数等等。使用连接池将大大提高程序效率,同时可以通过其自身的管理机制来监视数据库连接的数量、使用情况等。现以一个名为 ConnectionPool 的连接池来实现连接池的连接。ConnectionPool 的基本属性:

```

m_ConnectionPoolSize: 连接池中连接数量下限
m_ConnectionPoolMax: 连接池中连接数量上限
m_ConnectionUseCount: 一个连接的最大使用次数
m_ConnectionTimeout: 一个连接的最长空闲时间
m_MaxConnections = - 1: 同一时间的最大连接数
m_timer: 定时器
  
```

这些属性定义了连接池与其中的每个连接的有效状态值。连接池的自我管理,实际上就是通过定时的对每个连接的状态及数量进行判断而进行相应操作。

可以定义出 ConnectionPool 要完成管理所需要的基本接口:

```

public class ConnectionPool implements TimerListener{
public boolean initialize() // 连接池初始化
public void destroy() // 连接池的销毁
public synchronized java.sql.Connection getConnection() // 取一个连接
public synchronized void close() // 关闭一个连接
private synchronized void removeFromPool() // 将一个连接从连接池中删除
private synchronized void fillPool() // 维护连接池大小
public synchronized void TimerEvent() // 定时器事件处理函数
}
  
```

通过这几个接口,已经可以完成连接池的基本管理。在 TimerEvent() 函数中完成连接池的状态检验工作,fillPool() 时连接池至少保持最小连接数。因为要保存每一个连接的状态,所以还需要一个数据库连接对象:

```

class ConnectionObject{
  
```

```

public java.sql.Connection con; public boolean inUse; // 是否被使用标志
public long lastAccess; // 最近 1 次开始使用时间
public int useCount; // 被使用次数
}

```

加入了 ConnectionObject 对象后,在 ConnectionPool 中操作的应该只是 ConnectionObject,而其他进程需要的只是 ConnectionObject 的 con 属性,因此再加入一个类,作为其他进程获得与返回连接的接口:

```

class Conn{
GetConnection(); // 从连接池中取出一个有效连接
CloseConnection(); // 返回连接,此时并没有关闭连接,只是放回了连接池
DestroyPool(); // 销毁连接池
}

```

4 结语

JDBC 构造中间件的工作方式和实现是研究的重点技术.笔者通过对 JDBC 的 4 类驱动方式进行比较,得知第 4 类 JDBC 驱动方式是直接面向数据库的纯 Java 驱动程序,具有较高的性能,能够直接访问 DBMS,所以采用第 4 类驱动方式;接着根据该教务系统的实际情况构造了一个基于 JDBC 的中间件,同时由于该教务系统存在大量数据同时访问时工作效率低的客观问题,因此采用数据库连接池的技术加以解决,从而提高整个系统的数据访问效率.

参考文献:

- [1] 罗莉娟,刘旗超,薛伟林.用设计模式构建基于 XML 的双层数据访问中间件[D].四川大学,2004:36-42.
- [2] 张剑.XML 技术系列讲座[EB/OL].<http://q.yesky.com/group/review-8451650-1.html>,2006-12-25.
- [3] 周浩,周华.丰富您的 Web 功能[EB/OL].http://www.pcoworld.com.cn/2001/back_issues/2118/1833a.asp,2007-10-29.
- [4] 阎宏.Java 与模式[M].北京:电子工业出版社,2002:308-316.
- [5] 刘宝林.Java 程序设计与案例[M].北京:高等教育出版社,2006:1-10.

Working Mode and Implementation of Constructing Middle Ware Based on JDBC

JIANG Feng-lian, GUO Hua-long, GONG Chang-bin

(College of Mathematics and Computer Science, Longyan College, Longyan 364000, Fujian China)

Abstract: Aiming at heterogeneous data transmission in university educational management systems, the authors put forward a working mode and its implementation of constructing middle wave based on JDBC: The middleware is constructed according to the actual business of educational management system, and the technology of database connection pool is introduced. This scheme deals with the inefficiency which results from a large amount of data access at the same time; thus, it improves the data access efficiency of the entire system, and has high independence, portability and reusability.

Key words: heterogeneous data sources; middleware; JVM; JDBC