

J2EE 平台上异构数据源互操作的实现方法*

朱 骐

(盐城工学院 计算中心,江苏 盐城 224003)

摘 要 基于 J2EE 企业级平台,讨论了可用于异构数据源互操作的各项关键的使能技术及服务,提出了互操作实现框架,给出了一个验证实例。

关键词 J2EE;互操作;使能;服务;EJB

中图分类号 TP311 **文献标识码** A **文章编号** 1671-532X(2004)01-0037-04

数据源异构是信息集成面临的最主要的异构,面对各信息系统异构的数据层,信息集成必须去实现异构数据源的互操作及数据的互用^[1]。

异构数据源的物理结构与逻辑结构存在许多差异,要实现异构数据源的互操作,首先必须获得与每种数据源的连接,在网络环境下,连接方案主要有 3 种:①数据库网关②基于 TCP/IP、Http 进行套接字编程③数据源连接 APIs,其中以第 3 种方案最为简单可行。

Java JDBC 是 Java 连接数据源的 API 集合,它为 Java 应用系统中数据源的互操作提供了基础。J2EE 作为 Java 企业级平台,其数据使能技术把 JDBC 资源与命名服务结合起来,再与分布式通信使能技术 Java RMI 集成,就能从远程管理客户所获得的对数据源的连接,而这一切都由 J2EE 应用使能的关键技术 EJB 通过 Entity Bean 组件方便实现。EJB 技术还提供 Session Bean 组件来实现对应用的各种业务的处理。

本文主要讨论在 J2EE 企业级平台上通过上述技术实现异构数据源互操作的思路与方法。

1 J2EE 企业级平台

J2EE(JavaTM 2 Platform, Enterprise Edition)是 Java 构建企业级应用的标准平台,它由一整套服务(Services)、应用程序接口(APIs)和协议构成,对开发面向 Web 的、或面向桌面的多层应用提供功能支持,使开发过程大为简化。

1.1 J2EE 企业级应用模型

J2EE 企业级应用模型是多层的体系结构(如图 1),对于面向 Web 的应用,其层次有:客户层、Web 逻辑层、业务逻辑层和数据层;对于面向桌面的应用,则 Web 逻辑层省去。多层方式使企业级应用具有很强的伸缩性,并允许每层专注于特定的角色,这一特点也使得 J2EE 的诸种技术可作为信息集成的易用技术。

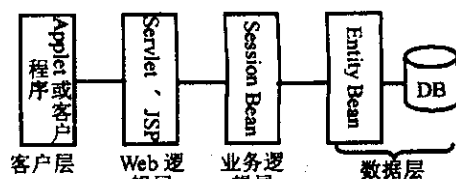


图 1 J2EE 多层应用模型

Fig. 1 J2EE's Application Model with Layers

每层都由企业组件构成,企业组件采用容器开发方法。所谓容器就是运行组件的软件包,它为企业组件提供部署服务和高效的、可扩展的、相互依赖的计算环境,J2EE 规范定义了 4 种企业组件/容器:

- (1)Applet 容器,运行 applet;
- (2)客户程序容器,运行桌面客户程序;
- (3)Web 容器,运行 Web 组件,如 Servlet、JSP 等;
- (4)EJB 容器,运行 EJB(Enterprise JavaBean)组件,如 Entity Bean、Session Bean 等。

* 收稿日期 2003-09-15

作者简介:朱 骐(1965-),男,江苏南京人,盐城工学院讲师,研究方向为信息系统与集成技术。

4 种容器中的企业组件在多层应用模型中被部署在不同层次、分担不同的角色。前两者位于客户层中,负责与用户的交互,提供用户界面;后两者是服务器方组件,具体地,Web 组件位于 Web 逻辑层中,充当 Web 服务器方组件,负责提供动态页面;而 EJB 组件分成两类:①Session Bean 组件处于业务逻辑层中,负责处理应用中的大多数业务,如统计、汇总、分析、功能集成、打印等等;②Entity Bean 组件位于数据层中,充当数据源的逻辑代理,提供数据服务。

每个企业组件都被部署为相应层次中的一个模块^[2]。

1.2 互操作涉及的技术及服务

J2EE 平台上,用于异构数据源互操作的企业使能技术及服务有下列 4 个方面:

(1)数据使能技术:JDBC 是 Java 数据使能的基本机制。传统的 JDBC,通过在驱动程序管理器中注册的驱动程序实现与数据源连接,数据源可以是关系型数据库、CSV 文件、Excel 文件等等。

基于 J2EE 的 JDBC 可以把数据源对象与名字绑定,通过使用 J2EE JNDI(Java Naming and Directory Interface)服务建立与数据源的连接,它与 Java RMI 一起内嵌在 Entity Bean 中,因而可以建立中间层组件(在客户程序与数据源之间)来实现对数据源对象的注册,这有两点优势:①数据源由中间层使用 JNDI 来注册,对于分布式客户程序,可以由远程来管理它所获得的连接,减轻客户方负担;②客户程序不受数据源 URL 的变更影响,只要修改中间层组件即可。因此该方法特别适用于电子商务瘦客户机的要求。

(2)分布式通信使能技术:适应网络分布计算环境,传统的分布计算方法 RPC(Remote Procedure Call)与对象技术结合便产生了分布对象计算技术。RMI(Remote Method Invocation)是纯 Java 的分布式对象通信模型,基于 RMI 的客户可以通过远程接口,与在不同进程中的对象进行通信,使用对象方法提供的服务;在 J2EE 应用的多层体系结构中,Web 逻辑层与业务逻辑层、业务逻辑层与数据层总是呈现远程 C/S 关系,RMI 是 J2EE 构建这些关系的基础。另外 J2EE 也可以通过 CORBA 获得跨平台的分布对象计算支持。

(3)Web 使能技术:它是 J2EE 创建 Web 服务器方组件的技术,其核心是 Java Servlet 和 JavaServer Pages。Java Servlet 负责处理 Web 逻辑层的重

要商务逻辑,用面向对象的方法构造,在收到客户方浏览器的请求后,Java Servlet 组件作出响应,必要时会从业务逻辑层或数据层调用数据服务。JSP(JavaServer Page)是 J2EE Web 服务器方脚本技术,主要负责 Web 页面的表示逻辑,在被调用时由编译工具(如 JSPServlet)编译成 Servlet 实现类。

(4)应用使能技术:EJB 是 J2EE 用于创建服务器方逻辑的基本机制。EJB 组件都含有对 JNDI 和 RMI 的支持,因而能够为远程客户提供服务。

Entity Bean 用来对企业数据进行封装,并向客户提供对企业数据的基本操作,它是数据源的 Java 逻辑映射,从功能上看,它与 Java 独立(Standalone)应用程序中的“数据源连接及操作类(class)”没有区别。Session Bean 用来在企业系统中执行某些操作且能将结果返回给客户,它与 Java 独立应用程序中的一般“业务处理类”的作用是一致的。

远程客户通过 EJB 远程接口(继承自 javax.ejb.EJBObject 接口)实现与 EJB 对象的连接,EJB 对象由本地接口(继承自 javax.ejb.EJBHome 接口)为客户创建。EJB 向开发人员提供的好处是快速、简单地构建可伸缩的业务逻辑层和数据层。

2 异构数据源互操作原理

2.1 异构数据源互操作概念构成

每一数据源都有自己的一组基本操作,基本操作集 BO 形式化描述为:

$$BO = \{\text{查询, 修改, 添加, 删除, ...}\}$$

若存在两个异构数据源 DB_1 、 DB_2 , DB_1 的基本操作集为 BO_1 , DB_2 的基本操作集为 BO_2 , 则 DB_1 与 DB_2 能实现的互操作集 IO 为:

$$IO = BO_1 \cdot BO_2$$

在应用系统中实际的互操作集 RIO 是 IO 的一个子集^[3]。

2.2 基于 J2EE 的异构数据源互操作框架

基于 J2EE 平台,可构造一个异构数据源互操作的实现框架(如图 2)。

各 Entity Bean 模块是不同数据源的逻辑映射,并提供对数据源的基本操作集。尽管数据源之间在数据格式、数据精度、数据模型和语义等逻辑结构方面存在差异,但在 Entity Bean 层次上,数据的逻辑结构能达到同一的 Java 规范,这为异构数据源间的数据互用提供了基础。这类模块位于数据源一侧,与数据源共同组成数据层。

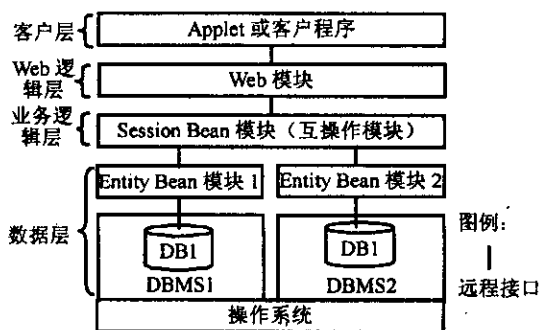


图 2 异构数据源互操作的总体方案

Fig.2 Interoperability Scheme Between Heter-structure Datasources

Session Bean 模块是业务处理模块,包括本文重点讨论的互操作业务的处理,它向上为 Web 逻辑层(面向 Web 的应用)或直接为客户层(面向桌面的应用)提供应用的各种业务功能;在两个 Entity Bean 模块提供的基本操作集 BO 基础上,该模块实现了应用系统实际互操作集 RIO。

3 验证实例

3.1 系统配置及实现目标

笔者根据上述框架设计了一个验证实例。系统配置为:操作系统 Windows2000,两个 DBMS 分别是 J2EE1.3.1 自带的 cloudscape4.06 和 SQL server2000。在 cloudscape4.06 中建有 ExampleDB 数据库,该库的 EMPLOYEE 表中存有雇员一般信息,字段集 = {EMPLOYEEID, NAME ...},在 SQL server2000 建有 ExampleDB1 数据库,该库的 HEALTH 表中存有雇员健康状况信息,字段集 = {EMPLOYEEID, STATUS ...}。实现目标是:在页面上输入雇员编号,提交后便显示 EMPLOYEE 表中 NAME(姓名)字段、HEALTH 表中 STATUS(身体状况)字段的信息,.....。

下面给出与互操作相关模块的实现。

3.2 数据层模块实现

两个数据层 Entity Bean 模块分别是 Embase 和 Hebase,Embase 包括 EmployeeBean.class、EmployeeHome.class、EmployeeInterface.class 3 个类,是 EMPLOYEE 表的逻辑映射;Hebase 包括 HealthBean.class、HealthHome.class、HealthInterface.class 3 个类,是 HEALTH 表的逻辑映射。采用 J2EE1.3.1 部署工具对 2 组类代码分别进行部署,部署方式是 CMP1.0,Primary Key 设为 EMPLOYEEID,jdbc 通过名字绑定各自的数据库对象以实现和数据源

的连接,并通过配置 SQL 语句确定对数据的基本操作,最后将两模块的 JNDI Name 分别设为 ejb/EmployeeEntityBean 和 ejb/HealthEntityBean,以便能被业务逻辑层模块识别调用。

3.3 业务逻辑层模块实现

业务逻辑层 Session Bean 模块为 Sessmid,该模块由 EmheBean.class、EmheHome.class、EmheInterface.class 3 个类组成,在 EmheBean.class 中通过 JNDI 远程调用数据层模块 Embase 和 Hebase 中有关操作数据的基本方法,并在此进行功能集成。部署时该模块状态为 Stateless,并将 JNDI Name 设为 ejb/EmheSessionBean,以便 Web 逻辑层或客户层调用。部分代码如下:

```
...//定义包和导入包
//在 EmheBean 类中实现互操作集 IO
public class EmheBean implements ejb.Session-
Bean
{
    //定义变量
    private String id = "";
    private String name = "";
    private String status = "";
    ...//其它变量
    EmployeeHome employeeHome = null;
    HealthHome healthHome = null;
    public EmheBean() throws SQLException{
        try{
            Context initial = new InitialContext();
            //生成并远程查找名为 ejb/EmployeeEntityBean 的对象
            Object objref1 = initial.lookup("ejb/EmployeeEntityBean");
            employeeHome = (EmployeeHome)PortableRemoteObject.narrow(objref1, EmployeeHome.class);
            //生成并远程查找名为 ejb/EntityHealthBean 的对象
            Object objref2 = initial.lookup("ejb/HealthEntityBean");
            healthHome = (HealthHome)PortableRemoteObject.narrow(objref2, HealthHome.class);
        }
        ...//捕获例外
    }
    //调用数据层各模块的方法(基本操作集
```

BO 中的操作)

```

public String getName( String id ) throws Ex-
ception { //查询姓名
    try{
        EmployeeInterface employee = employee-
Home.findByPrimaryKey( id );
        name = employee.getName( );
        return name ;
    }
    ...//捕获例外
}

public String getStatus( String id ) throws Ex-
ception { //查询身体状况
    try{
        HealthInterface health = healthHome.
findByPrimaryKey( id );
        status = health.getStatus( );
        return status ;
    }
    ...//捕获例外
}
...//其它调用
//实际互操作集 RIO 中的操作
public String getNS( String id ) throws Excep-

```

tion { //同时取得姓名和身体状况

```

    try{
        String s = getName( id )+ getStatus( id );
        return s ;
    }
    ...//捕获例外
}
...//实际互操作集 RHO 中的其它互操作
...// SessionBean 模块的其它方法
}

```

3.4 性能分析

因为 J2EE 为实例中各 EJB 模块提供了优秀的计算环境,在性能方面,比 Java 独立应用程序有很大提高,对于大系统,由于模块环节增多,要提高集成系统总体性能,方法来自两个方面:①在业务逻辑层减少中间环节;②在数据层充分利用可用的索引、加大分配给过程缓存的百分比、使用存储过程等。

4 结束语

J2EE 平台构建可伸缩性应用的特点,为异构数据源在 Java 上的互操作提供了可行性和易用性,但是要达到与不同语种实现的应用系统的集成,还必须使用支持 Java 的 CORBA 去实现^[4]。

参考文献:

- [1] 万麟瑞,李绪蓉.系统集成方法学研究[J].计算机学报,1999,22(10):1025-1031.
- [2] Paul J Perrone,张志伟,谭郁松,张明杰译.J2EE 构建企业系统[M].北京:清华大学出版社,2001.
- [3] 万麟瑞,徐峰.敏捷制造系统集成框架研究[J].计算机工程与应用,2000(10):62-65.
- [4] 王继英,顾冠群.异构分布对象系统集成模型研究[J].计算机工程与应用,2001(3):21-25.

Interoperability Method Between Hetero-structure Datasources Based on J2EE

ZHU Qi

(Computer Center of Yancheng Institute of Technology Jiangsu Yancheng 224003, China)

Abstract Based on J2EE platform, the enablings and services which can achieve the aim of Interoperability between hetero-structure datasources are discussed in this paper. The general framework with the method of Interoperability are put forward. The validity of the framework has verified by applications.

Keywords J2EE; Interoperability; Enabling; Services; EJB