

DAO 模式在 EJB 中的应用

Application of DAO Pattern In EJB

张文东 侯 飞 (东营 石油大学(华东) 计算机与通信工程学院 257061)

潘 琪 周 宏 (东营 胜利石油管理局 257061)

摘要:本文介绍了 EJB 技术、DAO 模式和 Fast - LaneReader 模式,同时结合事例给出一种在 EJB 体系结构中采用经过 Fast - LaneReader 模式优化的 DAO 模式进行图像数据存取的方法。

关键词:EJB DAO 模式 Fast - LaneReader 模式

1 引言

随着计算机技术的发展和人们对信息服务需求的不断增长,Internet 上所应用的数据存储类型已经由开始的简单结构化数据扩展到复杂非结构化数据,尤其对于图像这类二进制字节流的非结构化的数据,并没有一种通用的处理规范,各个数据库厂商对于自己的产品分别采用了各自不同的处理方法,彼此之间并不兼容。例如,在 Oracle 中常采用 BLOB 数据类型,Sybase 中采用 IMAGE 数据类型,并相应提供了 API。数据库环境发生变化导致重新修改程序代码。本文介绍了一种在 EJB 体系结构中采用经过 Fast - LaneReader 模式优化的 DAO 模式存取图像数据的方法来解决此不足。

2 EJB 与设计模式

2.1 EJB

EJB 是 Sun 公司 J2EE 平台的核心技术之一,是一个用于开发和部署多层结构的、分布式的、面向对象的 Java 应用系统的跨平台的组件体系结构。通过 EJB 应用程序开发者可以专注于企业应用所需的商业逻辑,而不必担心周围框架的实现问题。

EJB 组件是 EJB 组件体系结构的核心。它分为 Session (会话) Beans 和 Entity (实体) Beans 以及消息驱动组件三种类型。Session Beans 代表了与业务处理相关逻辑的实现,是一种作为单个的 Client 执行的

对象。Entity Beans 代表了与数据逻辑相关的实现,通常用来对数据库中数据提供一种对象的视图。Entity Beans 按应用服务器管理持久性的方式来划分,可分为 CMP (Container 管理持久性) 和 BMP (Bean 管理持久性) 两种。CMP 通过 JDBC 自动生成标准的数据库访问代码,并将组件的实例映射到数据库中,从而能够自动管理插入、更新、删除数据库中与实体有关的数据。BMP 则要求程序员自行编写代码来操作后台数据库,因此它比 CMP 独立性差,但它能更好地处理复杂的或不规则的数据。

2.2 设计模式

(1) 设计模式概念。设计模式 (Design Pattern) 简单说就是一个常用的方案。在我们的开发过程中经常会遇到一些相同或相近的问题,如果每次都寻找一个相应的解决办法,那么就会增加开发时间,降低开发效率。为了节省时间,提高开发效率,就需要提供一些解决类似问题的、在应用中被证明可行的方案——设计模式。所以一个设计模式就是描述了一个被证明可行的方案,这些方案可以重用,有良好的伸缩性。

(2) DAO 模式。在基于 J2EE 平台的应用程序模式中,DAO (Data Access Object ,数据访问对象) 模式是一个位于业务层之上的集成层的模式,它实现了操作对数据源的访问机制。DAO 的作用主要是抽象数据源,提供对数据源的透明访问。DAO 模式为 BMP 提供了业务层和资源层之间的松散耦合。由于数据访问对象会截取和服务所有对数据存储层的访问,就使得数

据存储层的实现细节对客户透明。通过使用该模式,可以创建更加轻便灵活、低耦合度的应用程序。

(3) Fast - LaneReader 模式。Fast - LaneReader 模式不通过 EJB 组件,而是更直接地通过 DAO 来读取数据,从而消除 EJB 的经常开支(例如远程方法调用、事务管理和数据序列化等)。与 DAO 模式不同,Fast - LaneReader 模式是一种优化模式,但他不是要替代原有的访问机制,而是作为补充使其完备。需要注意的是,Fast - LaneReader 模式可以使数据的提取更加快速和有效,同时又存在因为绕开 EJB 组件而带来的过时数据的风险。如果数据经常改变,不宜使用这样的模式,而是使用 EJB 组件来保证修改的正确性和查询到最新的数据。

3 在 EJB 中利用 DAO 模式存取图像数据

3.1 实体 Bean 直接访问后台数据库的不足

实体组件通过 JDBC 访问后台数据库,然而对于图像数据 JDBC 并未提供相应的 API,因此持久性逻辑必须由开发者自己编写。但是这样导致 EJB 组件过分依赖于固定的数据库资源,产生以下不足:

(1) 商务逻辑与数据访问逻辑耦合紧密,改变存储机制时要修改大量程序代码;

(2) 绑定了后台数据库资源,降低了组件的可复用性;

(3) 如果采用 CMP 方式,不仅减少了大量的编码量,而且可由 Container 自动进行许多性能优化。但由于图像数据对象的参与,组件不得不采用 BMP 方式,使得性能大大降低;

(4) 在网络负载较大的情况下,从客户端读取用户信息十分缓慢,尤其是图像数据对象,每次从客户端读取数据往往会给网络带来巨大压力,同时也消耗大量的系统资源。

3.2 利用经 Fast - LaneReader 模式优化的 DAO 模式存取图像数据的方法

(1) 采用 BMP 与 CMP 相结合,使 BMP 继承 CMP。这样不会打乱 CMP 潜在优化机制,一旦将来 JDBC API 升级,对图像数据对象有了更完善的支持,我们只需独立部署 CMP。

(2) 在业务层与数据源之间引入 DAO 模式,使商务逻辑与数据访问逻辑解耦合。当业务层发起的对数据源的访问时,首先发送给 DAO 组件,然后由 DAO 组件实施具体的对数据源的存取操作。这样一来,从业务层的角度来看,数据源是不可见的,即业务层只需要关心如何访问 DAO 组件提供的接口,而不必考虑如何获取数据源的连接、操作数据源等细节问题,数据源发生变化时只要保证 DAO 组件提供的对外接口不发生变化,业务层 EJB 的实现就无需改动,此时针对数据源的变化,只需改动 DAO 组件的细节即可。

(3) 使用 Fast - LaneReader 模式进行优化。当客户端频繁地对图像数据进行只读查询时,通过 Fast - LaneReader 模式直接到达 DAO 组件,从而对数据源进行访问,这样做节省了系统资源,提高了效率。

4 应用事例

在 EJB 体系结构中,用户是一个基本的实体,因此下面以用户实体为例设计一个名为 Student 的 Entity Bean,实现对图形数据的存取。Entity bean 存储在数据库的表 Student 中(见表 1),表中的各个字段是描述 Student 的属性,其中 photo 字段代表学生照片,是图像数据类型。

表 1 Student 表结构

字段	name	userid	gengder	speciality	address	photo
含义	姓名	学号	性别	专业	地址	照片

4.1 框架设计(如图 1)

从结构图中我们可以看到,当对学生属性进行编辑时,通过 EJB 对后台数据源进行操作;当对学生属性进行频繁只读查询时,直接通过 StudentDAO 访问后台数据库。

4.2 实现细节

在具体实现时,DAO 模式结构包含了 Bridge 模式和 Adapter 模式。Bridge 模式将数据访问逻辑的抽象部分(StudentBMP)与它的实现部分(StudentDAO)分离,使它们都可以独立地变化;Adapter 模式将 3 个互不兼容的具体实现类(StudentDAOSQLServer, Student-

DAOOracle, StudentDAOSybase), 转换为另一个公共接口 (StudentDAO), 这样能够同时支持 SQLServer、Oracle 和 Sybase 三种数据库。

4.2.3 数据源

定义对已经存在的数据库驱动接口, 通过 Adapter 模式对该接口进行适配。

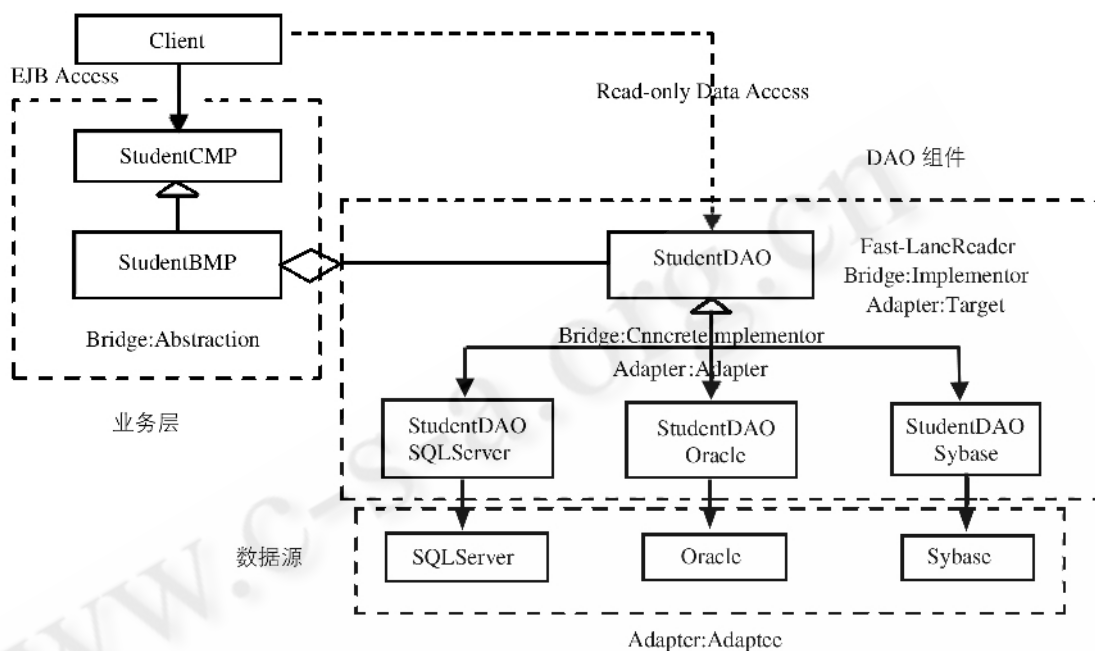


图 1 框架结构图

4.2.1 业务层

(1) StudentCMP, 实现商务逻辑, 如 getName()、setAddress()、getPhoto() 等。

(2) StudentBMP, 继承 StudentCMP, 实现一些标准数据对象的存取逻辑, 如 ejbFindByPrimaryKey() 等, 当涉及到操作图像数据对象的方法时, 调用 DAO 实现类型对象, 这样就把 StudentBMP 和后台资源类型完全屏蔽。

4.2.2 DAO 组件

(1) StudentDAO, 定义实现类的接口, 包括 ejbCreate(), ejbLoad(), ejbStore() 等, 提供持久性相关的 API, 供 Student BMP 访问和管理后台资源。

(2) DAO 实现类 (StudentDAO SQLServer, StudentDAO Oracle, StudentDAOSybase), 针对各种具体类型的后台数据库资源, 相应实现了 DAO 接口中定义的方法, 对 Adaptee 接口 (数据库资源) 与 Target 接口 (StudentDAO) 进行适配;

5 结束语

EJB 体系结构中运用经过 Fast-LaneReader 模式优化的 DAO 模式进行图像数据存取, 能够有效的解决由于后台数据库资源变化导致重新编译应用程序的不足, 并能够提高图像读取速度, 增强应用程序的伸缩性和可移植性。

参考文献

- 1 阎宏, Java 与模式[M], 电子工业出版社, 2002。
- 2 Alur P. J2EE 核心模式[M], 牛志奇等译, 机械工业出版社, 2002。
- 3 设计模式: 可复用面向对象软件的基础/伽玛等著 [M], 李英军等译, 机械工业出版社, 2000。
- 4 板桥里人 <http://www.jdon.com> 2002/05/06