维普资讯 http://www.cqvip.com

Hibernate技术的研究

林寒超[1],张南平[2]
(1.武汉理工大学 计算机科学与技术学院,湖北 武汉 430070;
2.武汉菲旺软件技术有限公司,湖北 武汉 430070)

摘 要:文中旨在介绍一种持久层技术——Hibernate。通过对数据库访问技术发展的探讨,引出了现在使用最广泛的Hibernate。接着对持久层和对象-关系映射的概念加以阐释,说明Hibernate技术出现的环境需要。介绍了Hibernate技术的原理和特点,以及与其它同类中间件如Entity Bean等进行比较,突出了Hibernate技术的优势。作为一种数据库访问中间件,Hibernate以它的优越性成为了众多程序员不可替代的选择。

关键词:Hibernate技术;持久层;对象-关系映射
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2006)11-0112-02

The Research of Hibernate Technology

LIN Han-chao[1], ZHANG Nan-ping[2]
(1. College of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China;
2. Wuhan Philwong Software Technology Co., Ltd, Wuhan 430070, China)

Abstract: Presents the Hibernate, a persistence layer technology. Firstly, discuss the development of the data access technology, the frequently used Hibernate is brought forward. And then explains the concepts of the persistence layer and ORM, and illuminates the need of the Hibernate's appearance. A big segment about the principle and the characteristic of the Hibernate is written, and compares with other similar technologies such as Entity Bean, the superiority of the Hibernate is pointed out. With its superiority, the Hibernate is inevitable to become the choice of programmers.

Key words: Hibernate technology; persistence layer; ORM

0 引 言

自从Java问世以来,针对Java开发的各种技术层出不穷。这些技术的出现在方便了项目开发的同时,也促进了Java的发展。但在数据库访问层,直接使用JDBC的方式进行数据库操作,一方面工程浩大,代码冗长,另一方面不方便移置和扩展,稳定性也存在极大问题。

于是,在对象关系数据库映射(Object Relational Mapping, ORM)设计思想下出现许多优秀的API和框架。由最初的直接使用JDBC到后期的DAO(Data Access Object),再到现在流行的持久层框架,如JDO,Entity Bean和Hibernate等,当然Hibernate就是其中现在最优秀、使用最为广泛的中间件。

1 持久层

"持久",英文即Persistence,简单来讲,就是把数据保存到可掉电式存储设备以供之后所用[1]。在大多数情况下,特别是企业级应用,数据持久化往往也就意味着将内存中的数据保存到磁盘上加以"固化",而持久化的实现过程则大多通过各种关系型数据库来完成。持久层也就是在系统逻辑层面上,专注于实现数据持久化的一个相对独立的领域(Domain)。

持久层是业务逻辑与关系数据库进行数据交互的中间层,该层的引入有效地分割了业务逻辑和关系数据库,使程序开发人员可以专注于业务逻辑功能的完善上,减少了开发的重复代码。相对于系统其它层面而言,持久层拥有一个较为清晰和严格的边界。图1是一个简单的持久层逻辑边界图。

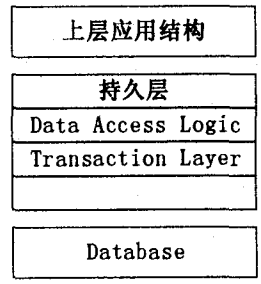

图1 持久层逻辑边界图

一般而言,"持久层"判定标准可以从如下3方面来界定。

收稿日期:2006-01-16
作者简介:林寒超(1980-),男,湖北汉川人,硕士研究生,研究方向为网络数据;张南平,博士,教授,研究方向为计算机网络。

维普资讯 http://www.cqvip.com

(1)如果表示层发生变化,需要从 JSP 迁移到 Java WebStart Client,数据持久化代码是否需要重新编译。

(2)如果业务逻辑层发生了变化,那么数据持久化逻辑代码是否需要重新编译。

(3)如果底层数据持久化机制发生了改变(如更换数据库类型),那么,系统中的非数据持久化部分代码(包括表示层、业务逻辑层)是否需要重新编译。

2 ORM 概述

ORM 是 Object - Relational Mapping 的简称,中文含义是"对象—关系映射"。"对象"即是 Object Oriented(面向对象技术)中的对象,而"关系"则是关系数据库中的关系[2]。

计算机语言发展是飞速的,从起初的面向过程语言发展到今天的面向对象语言可以说一个飞跃,现今在实际项目开发中的语言基本都是面向对象语言,如 Java。而现在技术最为成熟、使用最多的数据库是 RDB(关系型数据库)。这样一方面在业务逻辑层和表示层将系统中的各参与实体进行面向对象的封装,并籍此将现实世界中的逻辑进行高度抽象后,以计算机语言实现;而另一方面,在数据持久层,迫于目前数据库技术发展的现实,必须在现有的关系型数据库库模型上进行持久化实现。ORM 框架正是解决对象与关系相互转换的问题。

3 Hibernate 介绍

3.1 Hibernate 原理

作为一个 ORM 工具,Hibernate 比较彻底地映射 Java 对象,支持所有的使用各种 Java 思想[3]。它可以直接映射大部分的 JavaBean 而不需要对它们作任何修改,即使修改最多也就是在 Bean 里面加上一些私有访问方法;可以将一个用户定义的多个类的实例映射到一张表的同一行;还可以利用代理模式来简化载入类的过程。这将大大减少从数据库提取数据的代码的编写量,从而节约开发时间和开发成本。Hibernate 利用 reflection 机制,在系统启动时生成 SQL 语句,进行对象的持久管理。Hibernate 对每一种数据库都有对应的 Dialect 进行操作优化,从而提高它在各种情况下的效率[4]。

图 2 显示了 Hibernate 的工作原理,它是利用数据库以及其它一些配置文件如 hibernate.properties,XML Mapping 等来为应用程序提供持久服务的[5]。它的 API 中包括以下一些主要类。

(1)SessionFactory(net.sf.hibernate.SessionFactory)。

它包含已经编译的映射(mappings),是制造 session 的工厂,可能含有一些在各个事务(transaction)间之间共享的数据。

(2)Session(net.sf.hibernate.Session)。

单线程的、短寿命的对象,代表了一次会话的过程。实际上是把一个 JDBC 连接打包了,它可以包含一些持久话对象的缓存。

(3)Persistent Objects and Collections。

单线程的、短寿命的对象,包含了各种属性字段以及一些商业逻辑,它们可能仅仅是一些普通的 javabeans,但是它必然在某个时刻只能和一个 session 相关联。

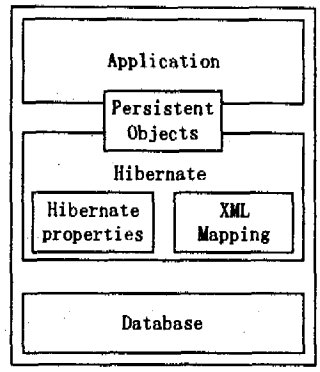

图 2　Hibernate 工作原理

3.2 Hibernate 特点

与其它 ORM 工具相比,Hibernate 拥有自身的优势和特点。

(1)Hibernate 是 JDBC 的轻量级的对象封装,它是一个独立的对象持久层框架。而 JDO 是 JDBC 的一个重量级封装。Hibernate 的特点主要是简单、易用、强大、灵活且速度快。Hibernate 可以代替 JDBC 的编程应用的场合,例如 Java 应用程序中访问数据库部分的代码,DAO 模型中接入数据库时的访问代码,甚至可以是 BMP 中访问数据库的代码。Hibernate 有自身的功能强大的 HQL,HQL 与 SQL 非常相似,并且它是完全面向对象的。

(2)Hibernate 不同于 Entity Bean。Entity Bean 是由容器处理大部分的数据完整性、资源管理和并发功能,因此开发人员主要关注业务逻辑和数据处理。Entity Bean 有两种持久性管理:BMP(Bean Managed Persistence)和 CMP(Container Managed Persistence)。BMP 需要由开发人员编写持久化代码;CMP 由容器自动生成持久化代码并管理持久性逻辑。从软件整体框架中来看,Hibernate 不能用于替代 Entity Bean。尽管 Hibernate 和 Entity Bean 都为程序员实现了透明的持久性,使程序员不需要关心太多的细节。但是,Hibernate 另外提供了更灵活的事务管理机制,Hibernate 使用 Session 来管理事务,而 Entity Bean 将事务管理派给了 EJB 容器。Hibernate 具有自身的事务管理,但是 Hibernate 的事务管理实际是 JDBC Transaction 的封装,或者是 JTA Transaction 的封装,因为 Hibernate 是对 JDBC 轻量级对象封装,Hibernate 的事务管理可以在 JDBC Transaction 和 JTA Transaction 之间选择,默认情况下将使用 JDBC Transaction。

(3)Hibernate 是一个跟 JDBC 密切关联的框架,Hibernate 的兼容性与 JDBC 驱动和数据库都有一定的关系,与 Java 程序或者 App Server 没有任何关系,不存在兼容问题。

(下转第 116 页)

维普资讯 http://www.cqvip.com

它们被放在由 vfsmntlist 指向的队列链表中。另一个指针——vfsmntail 指向表中的最后一项，nmru_vfsmnt 指针指向最近被使用的文件系统。每个 vfsmount 结构包括记录该文件系统的块设备的设备号、文件系统的装配目录和文件系统装配时分配的 VFS 超级块的指针。每个 VFS 超级块除了包含指向其对应文件系统的根 inode 节点的指针外，还指向它对应文件系统的 file_system_type 数据结构。每个文件系统的 inode 节点在本文件系统加载后一直驻留在 VFS inode 节点的缓存中。

3.3 在虚拟文件系统中查找文件

为了在 VFS 文件系统中查找某个文件的 VFS inode 节点，VFS 文件系统必须以每次一个目录的方式来解析文件名，查找代表文件名中间目录的 VFS inode 节点。每个目录查找过程都包含对代表它父目录的 VFS inode 节点记录的文件系统专有的查找过程的调用。由于总是能通过该文件系统的超级块找到该文件系统的根 VFS inode 节点，所以上面的办法是可行的。每当真实文件系统查找一个 inode 节点时，系统都会先检查目录缓存。如果目录缓存中没有该目录项，那么真实文件系统可以通过下层的文件系统或者在 inode 缓存中找到要找的 VFS inode 节点。

3.4 卸载文件系统

如果系统的某些进程正在使用该文件系统的某个文件的话，该文件系统不能卸载。如果有某些进程正在使用要卸载的文件系统的话，在 VFS inode 节点的缓存中就会有来自该文件系统的 VFS inode 节点。检查程序通过在 inode 节点表中查找来自于该文件系统所在设备的 inode 节点可以检测出这个问题。如果装配的文件系统的 VFS 超级块被标记为"脏"，这说明该超级块被改动过，所以文件系统要把它写回到硬盘上的文件系统中。一旦超级块被回写到硬盘上，由 VFS 超级块占用的存贮区就被归还给内核的自由存贮区池。最后为装配操作建立的 vfsmount 数据结构与 vfsmntlist 解除链接并释放掉[5]。

4 VFS 的系统调用

在 Linux 的 VFS 中提供了具体文件系统统一的操作界面，依赖于 VFS 的多种操作函数的数据结构。在这些结构体中，文件系统共用的多个操作函数的函数指针组成结构体。属性的赋值由各个文件系统根据需要进行，以保证不同的文件系统都可以利用该数据结构实现具体的操作[3]。文件操作数据结构是实现对文件的搜索、读写、更新等操作。节点操作数据结构是实现节点的创建、查找、删除等操作。超级块操作函数实现的是超级块的对节点的读取、写回、释放、删除等操作和超级块的释放、写回、获取文件系统状态、再次安装等。目录项操作是目录缓冲区常用的操作函数。Linux 文件系统的系统调用实现了用户进程访问文件系统。VFS 提供给用户大量的系统调用，在 VFS 的源程序中，系统调用占据了大量的编码。Linux 内核源程序提供的系统调用统一的格式为：

```
asm linkge retm_type sys_name(argument)
{
……
}
```

5 结束语

Linux 文件系统是一个优秀的文件系统。它的文件系统与 UNIX 操作系统有很多共同点，也有自身的特点。该文详细分析了 Linux 文件系统的建立、撤销过程，包括了文件系统的注册和注销，文件系统的安装和卸载过程。在整个文件系统的建立过程中，利用了文件系统类型链表和文件系统安装链表两个链表型数据结构。对文件系统的建立和撤消过程的研究，对于理解 Linux 文件系统的工作流程和实现方法起着关键的作用。

参考文献：

[1] 毛德操，胡希明. Linux 内核源代码情景分析(上册)[M]. 杭州：浙江大学出版社，2001.

[2] 胡晓明，王 强. Linux 核心源代码分析[M]. 北京：人民邮电出版社，2000：50－100.

[3] 陈莉君. Linux 操作系统内核分析[M]. 北京：人民邮电出版社，2000.

[4] 韩德志，钟 铭. Windows NT 文件系统驱动程序原理及设计方法[J]. 微机发展，2001，11(4)：63－65.

[5] 顾喜梅，顾宝根. Linux 虚拟文件系统实现机制研究[J]. 微机发展，2002，12(1)：60－62.

(上接第 113 页)

4 结束语

通过在原理和特点上与其它同类技术进行比较，得出在实际开发中 Hibernate 的优势。Hibernate 框架技术将开发人员从繁琐的代码中解放出来，使软件项目的开发效率大大提高。相信随着技术的不断改进，Hibernate 必将有更大的发展。

参考文献：

[1] 王 鑫. 基于 Hibernate 的 O/R 映射数据持久化的研究[J]. 中南民族大学学报：自然科学版，2005(3)：85－88.

[2] 梁添才，皮佑国. J2EE 数据持久层的 ORM 设计模式[J]. 深圳信息职业技术学院学报，2005(Z1)：22－26.

[3] 方 巍，孙 涌，张书奎. 整合 Struts 和 Hibernate 的 Web 系统应用[J]. 计算机与现代化，2005(12)：43－45.

[4] 田 珂，谢世波，方 马. J2EE 数据持久层的解决方案[J]. 计算机工程，2003(22)：94－96.

[5] Ebersole S. Hibernate Core for Java[EB/OL]. 2006－01－28. http://www.hibernage.org.