

文章编号:1001-9081(2008)09-2446-03

基于 Hibernate 框架的数据持久化层的研究及其应用

夏 贇,李志蜀

(四川大学 计算机学院,成都 610064)
(bettercloud@126.com)

摘 要:详细探讨了目前流行的 Hibernate 映射框架,结合某通信企业的电子运行维护系统(E-OMS)的开发,提出了基于 Hibernate 的数据持久层解决方案,并给出具体实现。根据系统实施效果,分析了该方案的优缺点并总结了一些开发方面的建议。

关键词:J2EE; 数据持久化层; Hibernate; 电子运行维护系统

中图分类号: TP311.12 **文献标志码:** A

Research of Hibernate frame-based data persistence and its application

XIA Yun, LI Zhi-shu

(School of Computer Science, Sichuan University, Chengdu Sichuan 610064, China)

Abstract: The currently popular Hibernate mapping framework was discussed in detail in this paper. With the development of a certain communication enterprise's Electronic-Operation Maintenance System (E-OMS), a solution on the hibernate-based data persistence was proposed and its implementation was also given. According to the effect of the system implementation, the advantages and disadvantages of this solution were analyzed, and some proposals in development were summarized.

Key words: J2EE; data persistence; Hibernate; Electronic-Operation Maintenance System (E-OMS)

0 引言

由于对象只能存在于内存中,而内存不能永久保存数据。如果要保存对象的状态,需要进行对象的持久化,即把对象存储到专门的数据存储库中^[1]。数据持久化是企业级应用系统开发中的一个重要环节,在进行 J2EE^[2] 企业级开发时,通常将其三层体系结构扩展成四层,即表示层、业务层、持久层和数据库层^[3]。

目前,已经出现了很多种数据持久化的解决方案。在 java 领域,可以直接通过 JDBC 编程来访问数据库,虽然运行效率高,但是在 Java 程序中嵌入大量 SQL 语句使得开发工程浩大,难以维护;也可采用 EJB CMP 方式,但使用比较复杂、难以掌握,且商用容器的性能一般;还可以采用 JDO 方式,但其对数据模型有一定限制,很多厂商如 IBM、BEA、Oracle 等都拒绝使用 JDO 规范^[4]。

在这种情况下,持久化层出现了一种非商业性的持久化中间件可供选用,可通过 XML 配置文件将数据库中的关系表映射到持久化类文件,实现 ORM(对象/关系映射),这就是 Hibernate。

1 Hibernate 持久化框架

1.1 Hibernate 简介

Hibernate 是一个功能强大 ORM 框架工具,解决了面向对象的应用程序和面向关系的数据库系统之间存在的“抗拒不匹配”^[4],通过 XML 配置文件,将 Java 持久化类映射到 SQL 关系数据库表结构,允许开发人员使用面向对象的思想对关系数据库进行操作^[5],从而使应用系统与数据库分离,不

仅提高了系统开发效率,也提高了系统运行效率。

此外,Hibernate 对 JDBC 做了轻量级封装^[6],不仅提供 ORM 映射服务,还提供数据查询和数据缓存功能,实现了广义的数据持久化,即业务数据的保存、更新、删除、加载以及查询。

1.2 Hibernate 原理

Hibernate 通过创建持久化类来完成对象关系映射,再通过一系列 XML 映射文件,将持久化类与关系数据库的表一一对应起来,从而实现应用系统业务数据的持久化。所谓持久化类,就是需要存储到数据库的 Java 对象模型,数据库中的表则是其在关系数据库中的存储形式。持久化类包括一系列属性,对应表结构中的列(即字段),而类的对象,则对应表结构中的行(即记录)。此外,持久化类中还提供与属性对应的 set() 和 get() 方法,用于存取数据。

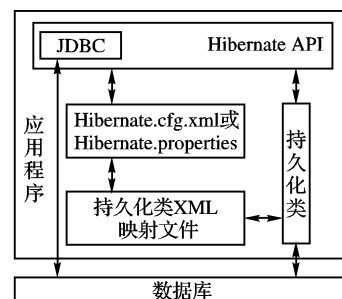


图 1 Hibernate 工作原理

Hibernate 配置文件有两种格式^[7]:一种是 XML 格式的文件(Hibernate.cfg.xml),一种是 Java 属性文件(Hibernate.properties),采用“键 = 值”的形式。在应用系统的开发中,Hibernate 配置文件中包含了 JDBC 连接数据库的驱动信息。

收稿日期:2008-03-27;修回日期:2008-05-29。

作者简介:夏贇(1983-),女,四川峨眉山人,硕士研究生,主要研究方向:计算机网络、信息系统;李志蜀(1946-),男,重庆人,教授,博士生导师,主要研究方向:计算机网络、智能控制、软件测评。

在应用程序访问数据库时,首先读取 Hibernate 配置文件,再通过配置文件找到持久化类的 XML 映射文件,根据该映射文件得到相应的持久化类,从而可以对类进行操作,使用 Hibernate API 进行数据库访问。由于 Hibernate 只对 JDBC 做了轻量级封装,应用程序可以使用 Hibernate API 对数据库进行操作,也可以直接绕过 Hibernate 使用 JDBC 完成数据库操作^[8]。

2 Hibernate 在电子运行维护系统中的应用

2.1 系统简介及架构设计

电子运行维护系统 (Electronic-Operation Maintenance System, E-OMS) 用于网络维护部门指挥、调度、管理日常生产任务,支持移动网络运行维护的跨地域、跨部门业务运作。该应用系统通过企业内部数据网络与运行维护管理需要的应用软件,将各级网络维护部门有机连接起来,从而实现对各级网络维护部门日常生产任务的电子化、流程化管理。从业务上,该系统分为工单管理模块,作业计划管理模块,值班管理模块,业务论坛模块,信息发布模块,图形报表模块,知识经验库模块,统计考核模块以及系统配置模块等。

E-OMS 系统采用 Struts1.1 应用框架构建 MVC(模型-视图-控制器)三层架构,对业务数据的操作则使用 Hibernate 框架进行持久化。

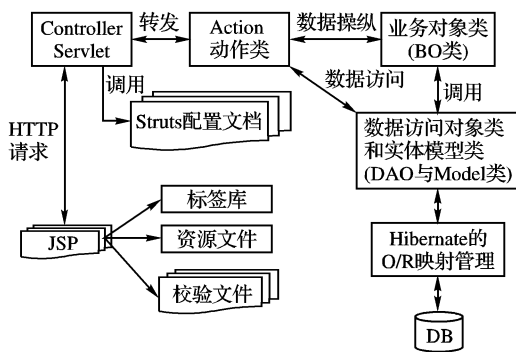


图 2 E-OMS 整体架构

该应用系统使用 Hibernate3.0,拓展了传统的 MVC 设计模式三层架构。视图由 JSP 来呈现用户界面,JSP 包含 Struts 自定义标签,资源文件和校验文件。控制器则由 Controller Servlet, Struts 配置文件和 Action 动作类组成。Hibernate 将模型组件细化为业务逻辑层,数据持久化层和数据库层。业务逻辑层由业务对象类(BO 类)调用数据库访问对象类(DAO 类)和数据持久化类(PO 类即 Model 类)来实现业务方法。数据库持久化层则由 Hibernate 提供 O/R 映射,通过 Hibernate API 直接操作数据库持久化类对象,实现与数据库的交互。

2.2 创建持久化类及其映射文件

以 E-OMS 系统的作业计划模块为例,详细阐述 Hibernate 实现数据持久化的机制。

持久化类是需要将其实例保存到数据库中的 Java 实体类,符合 Java Bean 的规范,包含一些属性和对属性的访问方法。其中属性的含义分为实体本身的特性和与其他实体的关联两部分,除了无意义的主属性只能被获取外,每一个属性都对应着两个外部对象访问方式,即 set() 和 get()。作业计划模块中,作业计划模板的持久化类定义如下。

```
public class TawwpModelPlan
    implements Serializable, DataObject {
```

```
private String id; //标识
private String name; //作业计划模板名称
private Set tawwpModelExecutes = new HashSet();
//对应的作业计划模板执行内容集合
private Set tawwpYearPlan = new HashSet();
//对应的年度作业计划信息
... //其他属性
public TawwpModelPlan() {}
public String getId() { return id; }
public void setId( String id ) { this. id = id; }
... //其他 set() 和 get() 方法
}
```

从作业计划的业务要求中可知,作业计划模板包含多项作业计划模板执行内容,是一对多的关系。此外,每年的年度作业计划按照作业计划模板生成,因此作业计划模板与年度作业计划也是一对多关联的。

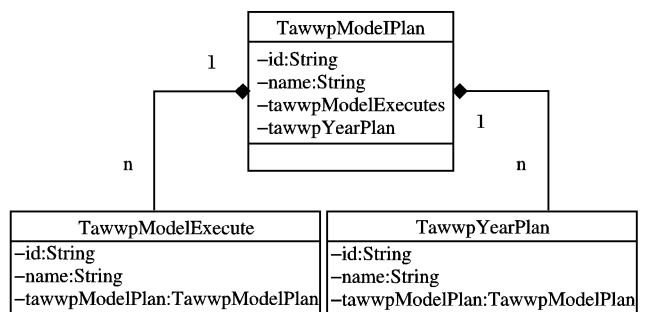


图 3 TawwpModelPlan、TawwpModelExecute 和 TawwpYearPlan 的类图

Hibernate 通过 XML 格式的描述文件,将持久化类映射到关系数据库的表,将类之间的关系映射为表之间的关系。创建作业计划模板实体类的 XML 映射文件,内容如下:

```
<hibernate-mapping>
<class name = "com. eoms. workplan. model. TawwpModelPlan"
table = "TAW_WP_MODELPLAN" dynamic-update = "true"
dynamic-insert = " true " select-before-update = " false "
optimistic-lock = "version" >
< id name = "id" column = "ID" type = "java. lang. String"
length = "32" unsaved-value = "null" >
< generator class = "uuid. hex" > </generator> </id>
< property name = "name" type = "java. lang. String" update
= "true" insert = "true" access = "property" column =
" NAMES " length = "100" not-null = "true" />
< set name = "tawwpModelExecutes" lazy = "true" inverse
= "true" cascade = "all" sort = "unsorted" >
< key column = "MODEL_PLAN_ID" > </key>
< one - to - many class = "com. eoms. workplan. model.
TawwpModelExecute" /> </set>
... //其他属性映射
</class>
</hibernate-mapping>
```

在 TawwpModelPlan. hbm. xml 文件中,对 tawwpModelExecutes 和 tawwpYearPlan 都进行了配置。使用 TawwpModelPlan 的 id 作为另外两个表的外键 MODEL_PLAN_ID,并将 inverse 属性设置为“true”,就可以完成 TawwpModelPlan 类对象与 TawwpYearPlan 类对象以及 TawwpModelExecute 类对象的一对多映射。

2.3 Hibernate 的配置

Hibernate 的配置文件除了包含数据库连接的驱动配置信息,还包含了持久化类的映射文件以及其他一些初始化信息,用于保证 Hibernate 正常完成应用程序的业务数据的持久化。XML 格式的配置文件可直接声明映射文件,无需调用 Configuration 类的 addClass() 方法来加载映射文件,从而提高

了应用程序的可维护性。E-OMS 系统的配置文件 hibernate.cfg.xml 主要代码如下所示:

```
<hibernate-configuration >
  <session-factory >
    <property name = "show_sql" > true </property >
    <property name = "dialect" >
      org.hibernate.dialect.InformixDialect </property >
    <property name = "hibernate.connection.driver_class" >
      com.informix.jdbc.IfxDriver </property >
    <property name = "hibernate.cache.provider_class" >
      org.hibernate.cache.EhCacheProvider </property >
    <mapping resource = "com/eoms/workplan/model/
      TawwpModelPlan.hbm.xml" / >
    <mapping resource = "com/eoms/workplan/model/
      TawwpModelExecute.hbm.xml" / >
    <mapping resource = "com/eoms/workplan
      /model/TawwpYearPlan.hbm.xml" / >
    ... //其他映射配置
  </session-factory >
</hibernate-configuration >
```

在配置文件中, show_sql 属性用于配置是否在控制台显示 SQL 语句, 方便程序调试。Dialect 用于配置本地数据库方言, E-OMS 中选用 IBM 公司的 Informix 数据库方言。Hibernate 几乎支持所有流行的数据库。此外, 还有配置数据库驱动和 Hibernate 二级缓存机制的属性。< mapping > 元素指定了需要加载的 Hibernate 持久化类映射文件。

2.4 Hibernate 操作持久化类

在 E-OMS 的开发中, 将连接数据库, 创建 SessionFactory 实例 session, 以及管理多线程 session 等操作都封装到指定文件 com.eoms.db.hibernate.HibernateUtil.java 中。

```
Configuration cfg = new Configuration().configure("/conf/hibernate.
  cfg.xml");
SessionFactory sessionFactory = createSessionFactory( cfg);
```

初始化过程结束后, 得到了全局的 SessionFactory 的实例, 就可以通过其 openSession() 方法从中获取 Session 实例, 进行对象的操作: query()、load()、insert()、delete()。此外, Hibernate 拥有一种功能非常强大的查询语言 (HQL, Hibernate Query Language), 这种语言与 SQL 非常相似, 但却是完全面向对象的, 查询的是持久化对象, 而不是数据库的记录, 可以用它来支持多态、继承、关联等关系。

```
Session s = HibernateUtil.currentSession();
String hSql = "";
hSql = "from TawwpModelPlan as m where m.deleted=0 ";
Query query = s.createQuery( hSql);
query.setCacheable( true);
```

由示例代码可以看出, Hibernate 通过提供丰富的 API 接口和类对象的公开方法以及 HQL, 能够轻松实现数据库表之间的关联, 完成数据持久化。从而避免了使用 SQL 语句进行底层的数据库操作的繁琐工作, 将开发人员的精力转移到业务处理方面, 大大提高了系统开发效率。

3 系统实施情况分析

3.1 系统实施情况

E-OMS 系统硬件条件如下: 数据库服务器为两台配置为 6 × 1.2 GHz CPU, 12 GB 内存, 标配硬盘的 SUN FIRE V880 做系统的数据库用服务器。两台数据库服务器通过共享磁盘阵列实现双机热备功能, 双机热备软件主要选用 VERITAS CLUSTER SERVER, 其能够自动检测应用或服务器故障, 并可

将其在备用服务器上快速重新启动, 使之具备故障切换功能, 保证系统的可靠性和可用性。Web 服务器为一台配置为 4 × 1.2 GHz CPU, 8 GB 内存, 标配硬盘的 SUN FIRE V880 UNIX 服务器。目前该系统实施效果良好, 已经在该通信公司的 20 多个省级集团公司运行使用, 获得了非常高的用户满意度。

3.2 Hibernate 的优势与不足

在系统整体架构中, Hibernate 完美地扮演着业务逻辑层到数据库层之间的中间件角色。Hibernate 框架符合模块化开发的思想, 降低了系统耦合度, 提高了系统的可维护性。其次, Hibernate 映射机制可以在很大程度上简化编程工作, 节约开发成本, 大大提高工作效率。同时, Hibernate 自身提供一些优化策略, 用于提高数据库的性能, 如与数据库交互时的多重最优化, 包括对象的缓存、有效外部连接的获取、必要时 SQL 语句的执行、集合过滤的使用等。

尽管 Hibernate 是一个非常优秀的数据持久化框架, 但它仍存在着一些不足。由于 Hibernate 映射机制会强制用户使用对象来描述关系型数据, 因此会在运行时产生大量冗余数据存放于内存。此外, 基于 Hibernate 机制的数据库访问是封装了 JDBC API 的, 因此相较 JDBC, 运行效率较低。最后, Hibernate 类包中对数据库对象的支持有限, 只支持对数据库表中数据的操作。

4 结语

在 E-OMS 系统开发中, 使用 Hibernate 框架提供数据持久化层构建的灵活支持, 解决了传统的并发访问控制低问题, 简化开发过程, 使系统具有较好的可扩展性和良好的可维护性。在开发过程中, 总结出了以下几点开发方面的建议。

- 1) 应用系统中数据库表相对较多、表结构内容明确时, 使用 Hibernate 优势明显。
- 2) 当系统对查询性能要求不是很高或者经常访问数据量不是很大的时候, 考虑 Hibernate。反之则考虑 JDBC。
- 3) Hibernate 2.0 不支持一些特殊查询和大批量数据更新, 可考虑直接使用 JDBC 查询语句或者更新为 Hibernate 3.0。
- 4) Hibernate 版本更新时, 要及时同步更新配置文件和持久化类的映射文件。
- 5) 应用系统业务要求持久化的实体较多时, 最好将持久化类及其映射文件存放于同一个目录下, 便于维护和更新。

参考文献:

- [1] 孙卫琴. 精通 Hibernate Java 对象持久化技术详解[M]. 北京: 电子工业出版社, 2005.
- [2] Java EE: Do more with less work[EB/OL]. [2008-02-01]. http://java.sun.com/j2ee/index.jsp.
- [3] 徐长盛, 戴超, 谢立. J2EE 数据持久化的研究[J]. 计算机应用与软件, 2006, 23(4): 56-57.
- [4] 杜玲玲. 基于 Hibernate 框架的数据持久化应用开发[J]. 微计算机信息, 2008, 24(6): 271-273.
- [5] 覃宇, 马铁军. Hibernate 数据持久化技术在轮胎企业 MES 中的应用[J]. 计算机工程与应用, 2006, 42(35): 208-210.
- [6] 高昂, 卫文学. 基于 Hibernate 与 Struts 框架的数据持久化应用研究[J]. 计算机应用, 2005, 25(12): 2818-2819.
- [7] 钱忠胜. 基于 Hibernate 的数据持久化研究及其应用[J]. 微计算机信息, 2007, 23(24): 242-244.
- [8] BAUER C, KING G. Hibernate in action[M]. Greenwich: Manning Publications Co., 2005.