

基于 Spring 和 Hibernate 的数据库 同步设计和实现

姚 瑶¹, 帅立国^{2,3}, 姜昌金¹

(1. 东南大学 自动化研究所, 江苏 南京 210096;

2. 东南大学 机械学院, 江苏 南京 210096;

3. 兰州理工大学 计算机与通讯学院, 甘肃 兰州 730056)

摘 要: Spring 是以控制反转和面向方面编程为基础的轻量级框架; Hibernate 主要用来实现对象持久化。文中介绍了 Spring 和 Hibernate 的主要特性; 讨论在采用 Teradata 数据仓库作为数据结算的主数据库, 以 Oracle 数据库作为存放配置文件的从数据库的结算管理系统中, 基于 Spring 和 Hibernate 的架构完成数据库同步的方案。

关键词: Spring; Hibernate; 数据库同步; 结算管理系统

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2007)12-0190-04

Design and Implementation of Database Synchronization Based on Spring and Hibernate

YAO Yao¹, SHUAI Li-guo^{2,3,4}, JIANG Chang-jin¹

(1. Institute of Automation, Southeast University, Nanjing 210096, China;

2. Dept. of Mechanism, Southeast University, Nanjing 210096, China;

3. School of Computer and Telecommunication, Lanzhou Univ. of Tech., Lanzhou 730056, China)

Abstract: Spring is a lightweight framework based on the theoretic of IoC and AOP. Hibernate is mainly a persistent service. Introduces the main characteristics of Spring and Hibernate; and analyzes the strategy of database synchronization based on the framework of Spring and Hibernate in a settlement management system. This management system uses Teradata data warehouse as the main database of the settlement data, and uses Oracle database as the accessory database of the information data.

Key words: Spring; Hibernate; database synchronization; settlement management system

0 引 言

Teradata 数据仓库作为 TB 级的数据库, 从不同的业务系统中抽取源数据, 并保留大量的历史数据作为分析应用。针对数据仓库进行的大量查询都非常复杂, 包括预先规划好的业务报表查询以及随机动态查询。由于数据仓库的操作量大, 并且同时支撑多个业务分析系统, 如果作为唯一的后端支撑结算管理系统将影响管理系统的运行效率。所以将系统中由前端输入的配置信息数据表移植到单一支撑结算管理系统的 Oracle 数据库中, 结算数据由 Teradata 从业务系统进行后端抽取, 结算时进行两个数据库的同步, 从而不影响

整个管理系统的运行效率。

根据整个管理系统的架构, 在这里利用 Spring 的依赖注入和 Hibernate 持久化对象的特点完成数据库的同步工作, 做到进行结算操作时触发同步, 从而保证系统的可靠运行。

1 Spring 的特点介绍

Spring 是 MVC 模式的技术, 是以控制反转和面向方面编程为基础的轻量级框架。当应用控制反转 (Inversion of Control, IoC) 时, 对象都是被动地给予它们的依赖, 而不是主动地去创建或寻找它们的依赖对象^[1]。

Spring 框架使用了分层的思想, 提供 7 个已经被定义好的模块: 核心容器, Spring 面向方面编程, Spring MVC 框架, Spring DAO, Spring ORM, Spring 上下文和 Spring Web。其中, 核心容器提供 Spring 框架最为基

收稿日期: 2007-03-03

作者简介: 姚 瑶 (1983-), 女, 江苏常州人, 硕士研究生, 主要从事基于数据仓库的系统应用研究; 帅立国, 副教授, 博士, 主要从事虚拟现实技术研究。

本、重要的功能,并且提供 IoC 容器,主要组件是 BeanFactory,是工厂模式的实现^[1]。BeanFactory 使用 IoC 将应用程序的配置和依赖性规范与实际的应用程序代码分开^[2]。一般说来,bean 应该遵守 JavaBean 规范,但是如果使用构造器依赖注入来连接 bean 的时候,就不需遵守 JavaBean 规范。

BeanFactory 实际上是实例化的,配置和管理众多 bean 的容器。这些 bean 通常会彼此合作,因而它们之间会产生依赖。BeanFactory 使用的配置数据可以反映这些依赖关系。每一个 bean 的依赖表现为属性、构造函数参数,或者当用静态工厂方法代替普通构造函数时工厂方法的参数。这些依赖将会在 bean 真正被创建出来后提供给 bean。每一个属性或者构造函数参数要么是一个要被设置的值的定义,要么是一个指向 BeanFactory 中其他 bean 的引用。Spring 在 BeanFactory 创建的时候要校验 BeanFactory 中每一个 Bean 的配置。Spring 会在 BeanFactory 装载的时候检查出错误,包括对不存在 bean 的引用和循环引用。

2 Hibernate 的特点介绍

Hibernate^[3]是一个开放源代码的纯 Java 的对象关系映射和持久性框架,它允许用 XML 配置文件把普通 Java 对象映射到关系数据库表中,它是对象/关系映射(ORM, Object/Relational Mapping)的解决方案。Hibernate 对 JDBC 进行了非常轻量级的对象封装,支持 Java 语言许多高级特性:如内省、多态、继承等。相对于使用 JDBC 和 SQL 来手工操作数据库,使用 Hibernate,可以大大减少操作数据库的工作量。

Hibernate 的技术本质是一个数据库服务的中间件,使用时具有很大的灵活性。但同时它的体系结构比较复杂,提供了好几种不同的运行方式。在轻型体系中,应用程序提供 JDBC 连接,并且自行管理事物,这种方式使用了 Hibernate 的一个最小子集;在全面体系结构中,对于应用程序来说,所有底层 JDBC/JTA 的 API 都被抽象了。

Hibernate 本身是个独立的框架,它不需要任何 Webserver 或 application server 的支持,可以使用在任何 Java 语言可以使用的地方。通常是将 Hibernate 框架和其他开源框架如 Struts、Spring 等联合使用,实现复杂的商业系统。在基于 MVC 设计模式的 Java Web 应用中,Hibernate 可以作为应用的数据访问层或持久层^[4]。Hibernate 通过配置文件(hibernate.properties 或 hibernate.cfg.xml)和映射文件(*.hbm.xml)把 Java 对象或持久化对象(Persistent Object, PO)映射到数据库中的数据表,然后通过操作 PO,对数据表中的数据

进行增、删、改、查等操作^[5]。其结构体系见图 1。

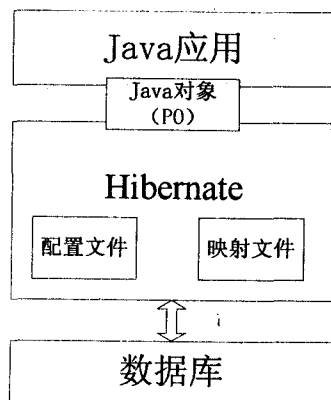


图 1 Hibernate 体系架构

3 同步方案

3.1 数据库复制

首先要实现将同步部分的数据表从 Teradata 数据仓库复制到 Oracle 数据库中,建立副本数据库。即要实现原结算管理系统的后端配置信息数据库从 Teradata 上移植到 Oracle 上, Teradata 数据仓库定期从业务系统抽取源数据,根据从 Oracle 上同步获得的配置信息进行结算,再将结算结果数据同步到 Oracle 上,这就是整个系统双数据库的构想。在这里采用 Microsoft SQL Server 2000 的数据导入导出工具实现数据库的复制。分别选择数据源为 Teradata 的数据库驱动,目标数据为 Oracle 数据库驱动,两库采用完全相同的表结构,完成数据库的复制工作。对于字段定义不同而无法导入的数据表,可以先在 Oracle 中建立相同结构的表,然后将 Teradata 中的表利用导出工具导出成 Excel 格式,再将 Excel 表数据导入 Oracle 中已建好的表中。

3.2 建立同步信息表持久化对象

将需要同步的数据表信息保存在表 SYN_TABLE_LIST 中。SYN_TABLE_LIST 表中字段设计为编号、源表名、目的表名、处理同步的 bean 名称、同步类型等。其中同步的类型用于标记是从 Teradata 同步到 Oracle 中的结算数据信息表,还是从 Oracle 同步到 Teradata 中的,从应用前端输入、修改的配置信息表。

建立 SYN_TABLE_LIST 表的持久化类。在该类中依据 SYN_TABLE_LIST 表的字段定义各私有属性变量,通过 get 和 set 方法进行属性值的获取和设置。持久化对象类似普通的 JavaBeans,唯一特殊的是它们与 Hibernate 的 Session 接口相关联。

设计类到关系型数据库的映射文件,用来把 PO

与数据库中的数据表, PO 的属性与表字段一一映射起来, 这是 Hibernate 的核心文件。首先, 在 hibernate.properties 文件中配置数据库的连接信息, 包括: 数据库驱动、连接用户名、密码、url、数据库连接池配置等; 其次, 在 hibernate.cfg.xml 文件中配置对应每个 PO 的配置文件的名称; 最后, 在每个 PO 对应的 *.hbm.xml 配置文件中将持久化对象的每个属性和表的对应字段进行映射, 下面为其一般配置形式:

```
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 2.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd">
<hibernate-mapping>
    <class name="..." table="...">
        <id name="..." type="..." column="...">
            <generator class="...">
                <param name="...">...</param>
            </generator>
        </id>
        <property name="..." type="..." column="..." length="..." />
        <property... />
        ...
    </class>
</hibernate-mapping>
```

其中 property 配置的是持久化对象的属性, name 为属性名, type 是属性的类型, column 是对应表中字段名。

配置好 SYN_TABLE_LIST 表的持久化对象后, 建立连接到表 SYN_TABLE_LIST 的 DAO 类, DAO^[6]是一个常用的设计模式, 通过创建一个 DAO 类来封装对持久化类的操作。该类实现系统底层对数据库的操作, 可以在该类中定义所有对数据表的操作。这里创建一个接口和它的实现类, 然后将它们交由 Spring 管理。这在 Spring 的应用上下文中加以定义。Spring 可以解决应用对象与硬编码源的绑定, 这通过将源(如 JDBC 数据源或 Hibernate 会话工厂)在应用上下文中定义为 bean 来实现。需要存取源的应用对象只通过 bean 关联接收对预先定义的实例的关联。

最后, 建立对表 SYN_TABLE_LIST 的事务层处理对象, 通过 Hibernate 的 Session 接口和 Transaction 接口, 调用建立的 DAO 对象实现对表 SYN_TABLE_LIST 的操作。Session 接口是 Hibernate 的核心接口, 用来操作 PO。Transaction 接口用来管理 Hibernate 事务, 它的主要方法有 commit 和 rollback, 可以由 Session 的 beginTransaction 方法生成。

3.3 实现表同步

新增所有同步相应表的 bean 类的接口类 ITableSynchronizer, 在其中分别定义从 Oracle 同步到 Teradata 的方法和从 Teradata 同步到 Oracle 的方法。

建立同步相应表的 bean 类, 将所有表和对持久化类都在 Hibernate 中作映射配置。在对应 bean 类中进行从源数据库选取出数据, 对持久化对象进行插入的操作, 从而实现数据库的同步。

新建类 TableSynchronizerManager, 实现反向控制(也叫依赖注入), 从而获取同步相应表的 bean 类。首先导入 org.springframework.beans.factory.BeanFactory 和 org.springframework.context.support.ClassPathXmlApplicationContext 包。一般通过使用 BeanFactory 接口以及它的子接口来使用 Spring, 为了使用的方便, Spring 提供了 BeanFactory 的一个扩展: ApplicationContext。使用 ApplicationContext, 可以减少需要编写的代码的数量, 并且 ApplicationContext 也增加了一些新的功能。如果在 Web 工程中使用 Spring, 甚至可以让 Spring 自动加载 ApplicationContext, 而无需自己编写代码来创建它。Spring 为 ApplicationContext 接口提供了三种实现: FileSystemXmlApplicationContext, ClassPathXmlApplicationContext 和 XmlWebApplicationContext。在这里 BeanFactory 采用如下方式实现方式:

```
ClassPathXmlApplicationContext ctx =
    new ClassPathXmlApplicationContext(new String[]{"bean-config.xml"});
```

```
BeanFactory factory = (BeanFactory) ctx;
```

通过调用 BeanFactory 的 getBean 方法, 根据从 SYN_TABLE_LIST 表获取的对应表的 bean 类名称, 返回同步相应表的 bean 类。

在 Spring 中, 使用宣告方式, 使得简单的部件通过配置来产生复杂的系统, 典型的情况是在 XML 中进行配置。一个最基本的 BeanFactory 配置由一个或多个它所管理的 Bean 定义组成。这里配置在 bean-conf.xml 文件中, 根节点 beans 中包含一个或多个 bean 元素。下面为其一般配置形式:

```
<? xml version="1.0" encoding="UTF-8"? >
<!DOCTYPE beans PUBLIC "-//SPRING/DTD BEAN/EN"
    "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
    <bean id="..." class="...">
        ...
    </bean>
    <bean id="..." class="...">
        ...
    </bean>
```

```
</bean>
...
</beans>
```

其中的 Bean 定义包括的内容有:

* classname:这通常是 bean 的真正的实现类。但是如果一个 bean 使用一个静态工厂方法所创建而不是被普通的构造函数创建,那么这实际上就是工厂类的 classname;

* bean 行为配置元素:它声明这个 bean 在容器的行为方式(比如 prototype 或 singleton,自动装配模式,依赖检查模式,初始化和析构方法);

* 构造函数的参数和新创建 bean 需要的属性;

* 和这个 bean 工作相关的其他 bean;比如它的合作者(同样可以作为属性或者构造函数的参数)。

3.4 数据库同步调度

到这里为止,所有同步的准备工作都已完成。新增一个数据库同步调度类 DBSynchronizer。在该类中定义两个标记以区分是将 Oracle 的数据同步到 Teradata 数据仓库还是将 Teradata 数据仓库同步到 Oracle。将两个数据库的连接信息作为参数传入。定义同步方法,调用该类的同步方法时将遍历 SYN_TABLE_LIST 表中所有同步表的信息,根据标记连接不同的数据库作为源库,获取数据插入到目的数据库中。

当前端用户点击结算功能时,系统首先传入 Oracle 的连接信息参数,调用同步调度类中将 Oracle 同步到 Teradata 的同步代码,在该段代码中将遍历所有符合标记的 SYN_TABLE_LIST 表中同步 bean 的名称,利用 Spring 的依赖注入,调用对应表的同步类代码,实现配置数据表的同步。然后系统调用 Teradata 中的数据抽取程序,从源系统抽取数据,再调用结算程序,获

(上接第 189 页)

所示。从图中可以得到结论,该电路的软件仿真结果符合设计要求,实现了预定功能。

在经过软件仿真之后,将完整程序下载到 FPGA 芯片,进行了相应的硬件调试和验证,进一步证明所设计电路完全符合预定设计要求。

4 结 论

使用 VHDL 硬件描述语言,在 QuartusII5.0 软件环境下,采用自顶向下的模块化设计思路,设计了一种出租车计价系统。通过软件仿真和硬件测试,证明该系统可以模拟实际的出租车计价器,完成计程、计费、计时和显示等功能,如果能进一步完善功能,如增添税控打印机功能等^[6],可以做到实用化和市场化。

取结算结果表。最后系统传入 Teradata 的连接信息,执行与第一次同步相似的流程,将结算结果表同步到 Oracle 数据库中以供前端查询,从而完成一次双库的同步操作。

4 结束语

通过实现从前端更新的数据配置信息和后端结算的结果数据的同步,保证了前端查询和更新配置信息的速率,也确保后端执行结算的正确性。这种基于数据仓库的双库同步,利用了现有的数据仓库资源,使得前端的管理系统可以基于一个相对小型的数据库系统,节约了资源,确保了效率。同时利用系统现有的 Spring 和 Hibernate 构架,快速便利地建立起数据同步的机制。

参考文献:

- [1] 周彩兰,李素芬,孙琳. Hibernate 在 Spring 中的研究与应用[J]. 计算机技术与发展,2006,16(10):62-67.
- [2] 史胜辉,王春明. Struts + Spring + Hibernate 三种架构在管理信息系统中的应用[J]. 中国管理信息化,2006(12):17-19.
- [3] 邹继成. Struts 与 Hibernate 实用教程[M]. 北京:电子工业出版社,2006:112-119.
- [4] 胡晓丹,魏长军. 基于 Struts + Spring + Hibernate 架构的能源管理系统的研究[J]. 计算机与现代化,2006(12):25-30.
- [5] 夏听,曹晓钢,唐勇. 深入浅出 Hibernate[M]. 北京:电子工业出版社,2005.
- [6] 李林红,沈文轩. 基于 Hibernate 和 Spring 的持久层 OR 映射技术[J]. 鞍山科技大学学报,2006(8):371-375.

参考文献:

- [1] 唐波,潘英俊,蔡秀梅,等. 基于 MC68HC705 单片机的多功能出租车计价器[J]. 自动化与仪表,2001,16(4):65-67.
- [2] 肖鹏,陈伟民. 一种新型多功能出租车计价器方案设计[J]. 自动化与仪器仪表,2000(3):50-51.
- [3] 胡大友. 新型出租车计价器模块的应用[J]. 国外电子元器件,1995(6):30-32.
- [4] 路而红. 专用集成电路设计与电子设计自动化[M]. 北京:清华大学出版社,2004.
- [5] 胡振华. VHDL 与 FPGA 设计[M]. 北京:中国铁道出版社,2003.
- [6] 汪清明. 出租车计价器税控打印机的设计[J]. 微计算机信息,2001,17(4):53-55.