

# Java 环境下基于 Hibernate 的 GIS 数据库访问策略

林大地<sup>1</sup>, 江 南<sup>2</sup>, 邹志强<sup>2,3</sup>, 白明白<sup>3</sup>, 赵郑慧<sup>3</sup>

(1. 南京大学城市与资源学系, 江苏 南京 210093; 2. 中科院南京地理与湖泊研究所, 江苏 南京 210008; 3. 南京邮电大学, 江苏 南京 215200)

**摘要:**全关系型数据存储模式将 GIS 数据的空间数据和属性数据统一存储于关系数据库中, 分析 Java 应用程序通过 JDBC 和 SQL 语言访问关系数据库中存储的 GIS 数据的不足, 提出基于 Hibernate 的解决方案, 并给出地理信息共享系统的开发实例。

**关键词:**Java; GIS; Hibernate; 对象/关系映射

GIS 数据包括属性数据与空间数据, 最初的 GIS 数据库组织模式为混合型: 利用文件系统进行空间数据的管理, 属性数据则采用关系型数据库管理, 两者通过关键字关联, 实现整个数据库的管理操作<sup>[1]</sup>。随着 GIS 数据量的不断增大, 功能模块不断复杂化, 这种模式已不能满足 GIS 数据统一管理的要求。全关系型数据库模式能较好地解决这个问题, 它通过加入空间数据字段等方法, 将属性和空间图形数据统一存储于关系型数据库的二维表中。在这种模式下, GIS 软件与大型商用关系型数据库管理系统的集成采用空间数据引擎 (ArcSDE, SpatialWare 等) 实现。全关系型数据库模式已成为 GIS 数据库建设的主要模式, 全关系型数据库及其访问语言 SQL 已成为 GIS 数据存储、操作及访问的标准<sup>[2]</sup>。目前, 一般采用 SQL 语言和 Java 语言规范应用程序访问数据库的 API, 即 JDBC 直接访问 RDBMS 中存储的 GIS 数据。

## 1 “JDBC + SQL”开发模式的不足

Java 应用中, 访问 GIS 数据库的关键在于持久层的设计<sup>[3]</sup>。而直接利用 JDBC 开发持久层难以把关系表记录完整地映射到持久对象的关系上, 因为存在 1:N 关系的持久对象的查询, 即 1+N 次对数据库的 SQL 操作, 也不能直接按对象的方式对持久对象编程, 这对于多元、海量、复杂的 GIS 空间数据必然导致系统运行速度缓慢, 且需要编写大量的数据库操作代码, 对开发人员要求较高。本文介绍一种基于对象/关系映射 (Object/Relational Mapping, ORM) 框架 Hibernate 的访问策略, 它能将对象映射到全关系型的 GIS 空间数据库中, 通过操作这些对象和 Hibernate 提供的一些基本类达到访问 GIS 数据库的目的, 从而提高系统开发效率, 加快系统运行速度<sup>[4]</sup>。

## 2 基于 Hibernate 的解决方案

Hibernate 是 JDBC 轻量级的对象封装, 它是一个独立的对象持久层框架, 与 Application Server 没有必然的联系, Hibernate 可以用在任何 JDBC 可以使用的场合<sup>[5]</sup>。总体上看, Hibernate 通过 properties 或 XML 文件配置运行时环境信息, 通过 XML mapping 文件进行类到数据库表之间的映射, 应用程序则通过 Hibernate 持久化的对象类操作数据库, 从而避免直接使用 JDBC 和 SQL 进行数据的操作 (图 1)。

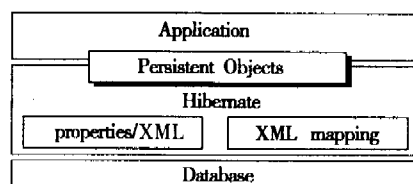


图 1 Hibernate 原理

Fig. 1 The principle of Hibernate

从 GIS 设计角度看, 可以认为 Hibernate 是 Java 应用程序与 GIS 全关系型数据库的中间件 (图 2)。在这种模式下, Hibernate 封装了底层的细节, 而向上层应用提供了 Session、SessionFactory、Transaction 等接口。

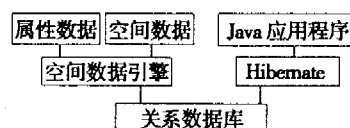


图 2 基于 Hibernate 的 GIS 中间件体系结构

Fig. 2 The GIS middleware architecture based on Hibernate

## 3 应用实例

笔者根据 Hibernate 在该共享系统中的应用, 以访问 shape 文件 block 经 ArcSDE 导入生成的数据库表为例, 阐述基于 Hibernate 的 GIS 数据库访问策略。

### 3.1 数据库表建设

该项目的关系数据库管理系统采用 Oracle8i, blocks.shp (包括它的索引文件和 dBASE 表), 由 ArcSDE 导入表空间 SDE 中生成表 BLOCKSTABLE, 同时生成辅助表 F13 和 S13。BLOCKSTABLE 的 BLOCKSCOLUMN 列为标准的 NUMBER 型字段, 存放索引值; 空间数据存放在表 F13 中; 而索引信息则存放在表 S13 中。以访问 F13 表为例, 开发如下。

### 3.2 配置 Hibernate

Hibernate 需要 XML 文件或 properties 文件声明运行时环境信息, 包括使用的数据库、数据库连接的各个参数、数据源、连接池大小、缓存处理、输出 Hibernate 生成的 SQL 语句、使用的 mapping 文件等配置信息, 本例使用 hibernate.cfg.xml 文件声明。

```
<property name="connection.url">
    jdbc:oracle:thin:@localhost:1521:oradb
</property>
<property name="connection.username">sde</property>
<property name="connection.password">admin</property>
```

收稿日期: 2006-01-05; 修订日期: 2006-02-26

作者简介: 林大地 (1980-), 男, 硕士研究生, 主要研究方向为 J2EE、网络 GIS。E-mail: lyndadi@126.com

```
<property name="connection.driver-class">
    oracle.jdbc.driver.OracleDriver
```

```
</property>
```

mapping 文件的声明语句为:

```
<mapping resource="com/niglas/data/F13.hbm.xml"/>
```

### 3.3 编写持久类

编写持久类 F13.java, 以便将表 F13 的字段映射到该持久类中。持久类符合 JavaBean 的规范, 包含一些属性(如 fid)以及与属性相对应的 get 和 set 方法(如 getFid() 和 setFid())。属性与需要映射的字段一一对应。

### 3.4 编写 mapping 文件

mapping 文件是一个 XML 文件, 文件名与持久类的类名相同, 扩展名为 "hbm.xml", 用以声明对象和关系数据之间的映射。在运行时, Hibernate 通过 mapping 文件生成各种 SQL 语句。mapping 文件有很多配置选项, 最重要的是要声明持久类、映射的表名和需要映射的字段。

```
<class name="com.niglas.data.F13"
    table="F13"> //声明持久类和映射的表名
    <property name="fid" />
    <property name="eminx" />
    <property name="emaxx" />
    <property name="eminy" />
    <property name="emaxy" />
    <property name="eminz" />
    <property name="emaxz" /> //声明需要映射的字段
```

### 3.5 通过对象和 Hibernate API 操作 GIS 数据库

(1) 创建 Configuration 类的实例, 其构造方法默认读取配置文件 hibernate.cfg.xml 或 hibernate.properties。调用 Configuration 类的 addClass() 方法读取持久类的 mapping 文件映射信息。

(2) 调用 Configuration 实例的 buildSessionFactory() 方法, 创建一个 SessionFactory 实例, 将 Configuration 对象包含的配置信息存储到 SessionFactory 对象的缓存中。

(3) 调用 SessionFactory 实例的 openSession() 方法, 创建 Session 实例, 通过 Session 接口的 save()、update()、delete()、load()、find() 等方法操作 GIS 数据库。

此外, 在查询方面, Hibernate 支持多种检索对象的方式,

并装备了一种极为有力的查询语言, 即 HQL。它是完全面向对象的, 具备继承、多态和关联等特性。

### 3.6 "JDBC+SQL"模式与 Hibernate 的比较

在持久层映射关系复杂的情况下, Hibernate 的运行效率远远高于 "JDBC+SQL" 模式<sup>[6]</sup>。笔者开发的地理数据共享系统提供了根据经、纬度判断查询的功能, 在同一服务器 (CPU 2.40 GHz, 内存 2.0 Gk) 及查询参数 (单线程、相同坐标范围、1:6 持久层映射关系) 下, 利用 Hibernate 策略访问数据服务器上的 GIS 数据库并输出查询结果仅需 1 s, 而 "JDBC+SQL" 模式需 8 s (数据经过多次测试求平均值而得)。

Hibernate 的访问策略也优于 "JDBC+SQL" 模式。在上述查询实例中, 要实现该功能, Hibernate 只需编写 5 行代码 (配置好 Hibernate 运行环境后), 而 "JDBC+SQL" 模式却需要编写近百行的标准 SQL 语句。

## 4 结语

Hibernate 通过 GIS 数据的持久化, 将表字段映射到 Java 对象中, 通过持久对象和 Hibernate API, 以面向对象的思想操作 GIS 数据库, 大大减轻了开发人员的工作量。此外, 在 GIS 数据的表关联映射复杂时, Hibernate 同样提供了 <one-to-one>、<many-to-one> 及 collection 的 <one-to-many>、<many-to-many> 等元素声明, 显著提高了 Java 应用程序访问 GIS 数据库的速度, 实现了查询优化, 并节约了开发成本。

### 参考文献:

- [1] 许捍卫, 冯学智. 空间数据存储机制研究[J]. 计算机应用研究, 2003(2): 39-40.
- [2] 杨晨毅, 刘吉平. 基于 SDE 的 GIS 空间和属性数据在 RDBMS 中的集成[J]. 计算机仿真, 2003, 20(11): 110-112.
- [3] 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, 29(22): 93-95.
- [4] HIBERNATE ORGANIZATION. Hibernate Reference Documentation[EB/OL]. <http://www.hibernate.org>, 2004.
- [5] ELLIOTT J. Hibernate: A Developer's Notebook[M]. O'REILLY, 2004.
- [6] BODOFF S, GREEN D, HAASE K, et al. The J2EE Tutorial[EB/OL]. <http://java.sun.com/j2ee/learning/tutorial>, 2003.

## GIS Database Access with Hibernate in the Java Environment

LIN Da-di<sup>1</sup>, JIANG Nan<sup>2</sup>, ZOU Zhi-qiang<sup>2,3</sup>, BAI Ming-bai<sup>3</sup>, ZHAO Zheng-hui<sup>3</sup>

(1. Department of Urban and Resources Sciences, Nanjing University, Nanjing 210093;

2. Nanjing Institute of Geography and Limnology, Chinese Academy of Sciences, Nanjing 210008;

3. Nanjing University of Posts and Telecommunications, Nanjing 215200, China)

**Abstract:** The total-relationship organization mode integrates the GIS spatial data and attribute data into the RDBMS, which becomes the main storage pattern. To access GIS data in RDBMS in the Java environment, the method of using JDBC API and SQL program language directly is adopted usually. However, it has some shortcoming in the development and execution efficiency. Referring to present situation and future development direction of GIS and Java program, accessing GIS Database with Hibernate in the Java environment is suggested in this paper. It establishes a mapping between the objects and relational database, and accesses the GIS database by operating those objects and Hibernate API, which acts as a middleware between the Java application and GIS database. It greatly increases the development efficiency and system execution speed. These two methods are comparatively discussed by an instance in the development of Geographic Information Sharing System, and two sets of data present the gap.

**Key words:** Java; GIS; Hibernate; ORM