

源码分析报告——H2 Database介绍

1. 序言

作为首次正式接触面向对象思想和Java语言的人而言，阅读、分析相关优秀项目的源码对该思想的学习是深刻的。本次源码分析的项目是H2 Database，本篇报告将主要对该项目的主要功能、流程和模块进行简要介绍。

2. 项目介绍

2.1 H2 Database是什么？

H2 Database（以下简称H2）是由Java编写的、开源的轻量级嵌入式数据库。它可以容易地嵌入到项目中，以及配置成内存数据库运行。

项目地址：<https://github.com/h2database/h2database>

2.2 H2部分功能及特性

- 平台无关性：H2完全采用Java编写。而由于Java语言的特性——即Java语言创建的可执行二进制程序能够直接运行于多个平台，H2运行可以不受平台限制。
- 嵌入式：这是H2最常用的功能。H2可以很方便地嵌入到其它项目中（导入依赖即可），方便地存储少量结构化数据（例如姓名、日期、地址等这些高度组织和整齐格式化的数据）。关于结构化数据和非结构化数据，可以参考IBM博客的一篇文章：[Structured vs. Unstructured Data: What's the Difference?](#)
- 内存数据库：H2支持在内存中创建数据库和表。众所周知，内存读写速度要比磁盘多许多倍。这是内存数据库的一大优势。不过当进程结束时，这些数据会丢失。但H2可以将数据持久化，将其保存在磁盘中。——这可以根据使用者的需求，选择数据去留。
- 支持全文检索：H2内置了全文搜索和基于Apache Lucene的全文检索。
- 适用于单元测试：基于H2轻量级、读写速度快、即用即消（通过搭建内存数据库）、嵌入式等特性，对于单元测试（检验程序模块，数据不需要保存）来说相当方便。
- 2种连接方式：嵌入式(本地)连接、使用TCP/IP服务器模式(远程连接)。
- 3种运行模式：嵌入式模式、服务器模式和混合模式。

以上仅列举了H2功能及特性的一小部分，限于篇幅不再过多展开。在这里笔者给出H2官网链接，供读者查阅：[H2 Database Engine](#)

3. 功能分析与建模——MVStore

3.1 简介

MVStore全称“multi-version store”，是H2较新版本使用的默认存储引擎。之后我们将对这个模块进行深入探讨。

3.2 一个极其简单的示例

让我们来看一个示例（源自官方文档，有些许修改）。

```
import org.h2.mvstore.*;

public static void main(String[] args) throws Exception {
    // open the store (in-memory if fileName is null)
    MVStore s = MVStore.open("E:/Java/H2Test");

    // create/get the map named "data"
    MVMap<Integer, String> map = s.openMap("data");

    // add and read some data
    map.put(1, "Hello World!");
    map.put(3, "Hello Java!");
    System.out.println(map.get(1));
    System.out.println(map.get(3));

    // close the store (this will persist changes)
    s.close();
}
```

一开始，我们建立了一个MVStore类的对象s（在路径E:/Java下，名字叫H2Test）。接着我们创建了一个MVMap对象（名字叫data），存放在s中。MVMap是MVStore中特有的键值映射表，在映射表内我们可以再插入一些键值映射，写和读的方法和HashMap类似。最终的打印结果也很显然：

```
Hello World!
Hello Java!
```

可以看到，MVStore的存储方式是非关系型的。关系型数据库的数据可以看作一个二维表格；而MVStore是键值存储，在之后我们会看到，其采用B-树数据结构进行存储。对于较少的数据，这种存储方式快速而灵活。