**Mybatis缓存**

分为1级缓存和2级缓存,2级缓存在sessionfactory中，1级缓存在session中。

1. session中的缓存在session连接关闭并被连接池回收时清理，或者手动清理
2. sessionfactory中的缓存需要在mybatis配置文件中手动配置，配置如下：

**[html]** [view plain](http://blog.csdn.net/tanga842428/article/details/52698657) [copy](http://blog.csdn.net/tanga842428/article/details/52698657)

* 1. **<setting** name="cacheEnabled" value="true"**/>**

**[html]** [view plain](http://blog.csdn.net/tanga842428/article/details/52698657) [copy](http://blog.csdn.net/tanga842428/article/details/52698657)

* 1. **<mapper** namespace="dao.userdao"**>**
  2. ...  select statement ...       <!-- Cache 配置 -->
  3. **<cache**
  4. eviction="FIFO"
  5. flushInterval="60000"
  6. size="512"
  7. readOnly="true" **/>**
  8. **</mapper>**

flushInterval：刷新轮询时间，每隔这个时间后缓存被清理一次。

**Hiberante缓存**

**一、why（为什么要用Hibernate缓存？）**

[**hibernate**](http://lib.csdn.net/base/javaee)是一个持久层框架，经常访问物理[**数据库**](http://lib.csdn.net/base/mysql)。

为了降低应用程序对物理数据源访问的频次，从而提高应用程序的运行性能。

缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据。

**二、what（Hibernate缓存原理是怎样的？）**

Hibernate缓存包括两大类：Hibernate一级缓存和Hibernate二级缓存。

**1.Hibernate一级缓存又称为“Session的缓存”。**

Session内置不能被卸载，Session的缓存是事务范围的缓存（Session对象的生命周期通常对应一个数据库事务或者一个应用事务）。

一级缓存中，持久化类的每个实例都具有唯一的OID。

**2.Hibernate二级缓存又称为“SessionFactory的缓存”。**

由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。

第二级缓存是可选的，是一个可配置的插件，默认下SessionFactory不会启用这个插件。

Hibernate提供了org.hibernate.cache.CacheProvider接口,它充当缓存插件与Hibernate之间的适配器。

**什么样的数据适合存放到第二级缓存中？**

1) 很少被修改的数据

2) 不是很重要的数据，允许出现偶尔并发的数据

3) 不会被并发访问的数据

4) 常量数据

**不适合存放到第二级缓存的数据？**

1) 经常被修改的数据

2) 绝对不允许出现并发访问的数据，如财务数据，绝对不允许出现并发

3) 与其他应用共享的数据。

**3.Session的延迟加载实现要解决两个问题：正常关闭连接和确保请求中访问的是同一个session。**

Hibernate session就是[**Java**](http://lib.csdn.net/base/javase).sql.Connection的一层高级封装，一个session对应了一个Connection。

http请求结束后正确的关闭session（过滤器实现了session的正常关闭）；延迟加载必须保证是同一个session（session绑定在ThreadLocal）。

**4.Hibernate查找对象如何应用缓存？**

当Hibernate根据ID访问数据对象的时候，首先从Session一级缓存中查；

查不到，如果配置了二级缓存，那么从二级缓存中查；

如果都查不到，再查询数据库，把结果按照ID放入到缓存删除、更新、增加数据的时候，同时更新缓存。

参考：<http://www.cnblogs.com/wean/archive/2012/05/16/2502724.html>

**Hibernate与Mybatis对比**

**首先简单介绍下两者的概念**

Hibernate ：Hibernate 是当前最流行的ORM框架，对数据库结构提供了较为完整的封装。

Mybatis：Mybatis同样也是非常流行的ORM框架，主要着力点在于POJO 与SQL之间的映射关系。

其次具体从几个方面说一下两者的区别：

**1.两者最大的区别**

针对简单逻辑，Hibernate和MyBatis都有相应的代码生成工具，可以生成简单基本的DAO层方法。

针对高级查询，Mybatis需要手动编写SQL语句，以及ResultMap。而Hibernate有良好的映射机制，开发者无需关心SQL的生成与结果映射，可以更专注于业务流程。

**2.开发难度对比**

Hibernate的开发难度要大于Mybatis。主要由于Hibernate比较复杂、庞大，学习周期较长。

而Mybatis则相对简单一些，并且Mybatis主要依赖于sql的书写，让开发者感觉更熟悉。

**3.sql书写比较**

Mybatis的SQL是手动编写的，所以可以按需求指定查询的字段。不过没有自己的日志统计，所以要借助log4j来记录日志。

Hibernate也可以自己写SQL来指定需要查询的字段，但这样就破坏了Hibernate开发的简洁性。不过Hibernate具有自己的日志统计。

**4.数据库扩展性比较**

Mybatis由于所有SQL都是依赖数据库书写的，所以扩展性，迁移性比较差。

Hibernate与数据库具体的关联都在XML中，所以HQL对具体是用什么数据库并不是很关心。

**5.缓存机制比较**

相同点：Hibernate和Mybatis的二级缓存除了采用系统默认的缓存机制外，都可以通过实现你自己的缓存或为其他第三方缓存方案，创建适配器来完全覆盖缓存行为。

不同点：Hibernate的二级缓存配置在SessionFactory生成的配置文件中进行详细配置，然后再在具体的表-对象映射中配置是那种缓存。

MyBatis的二级缓存配置都是在每个具体的表-对象映射中进行详细配置，这样针对不同的表可以自定义不同的缓存机制。并且Mybatis可以在命名空间中共享相同的缓存配置和实例，通过Cache-ref来实现。

两者比较：因为Hibernate对查询对象有着良好的管理机制，用户无需关心SQL。所以在使用二级缓存时如果出现脏数据，系统会报出错误并提示。

而MyBatis在这一方面，使用二级缓存时需要特别小心。如果不能完全确定数据更新操作的波及范围，避免Cache的盲目使用。否则，脏数据的出现会给系统的正常运行带来很大的隐患。

**6.总结**

Hibernate与MyBatis都可以是通过SessionFactoryBuider由XML配置文件生成SessionFactory，然后由SessionFactory 生成Session，最后由Session来开启执行事务和SQL语句。

而MyBatis的优势是MyBatis可以进行更为细致的SQL优化，可以减少查询字段，并且容易掌握。

Hibernate的优势是DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳。

# 1 简单简介

    1.1    **Hibernate 框架**

**Hibernate**是一个开放源代码的对象关系映射框架,它对JDBC进行了非常轻量级的对象封装,建立对象与数据库表的映射。是一个全自动的、完全面向对象的持久层框架。

    1.2    **Mybatis框架**

**Mybatis**是一个开源对象关系映射框架，原名：ibatis,2010年由谷歌接管以后更名。是一个半自动化的持久层框架。

# 2 两者区别

**2.1 开发方面**

        在项目开发过程当中，就速度而言：

            hibernate开发中，sql语句已经被封装，直接可以使用，加快系统开发；

            Mybatis 属于半自动化，sql需要手工完成，稍微繁琐；

        但是，凡事都不是绝对的，如果对于庞大复杂的系统项目来说，发杂语句较多，选择hibernate 就不是一个好方案。

**2.2 sql优化方面**

        Hibernate 自动生成sql,有些语句较为繁琐，会多消耗一些性能；

        Mybatis 手动编写sql，可以避免不需要的查询，提高系统性能；

**2.3 对象管理比对**

        Hibernate 是完整的对象-关系映射的框架，开发工程中，无需过多关注底层实现，只要去管理对象即可；

        Mybatis 需要自行管理 映射关系；

**2.4 缓存方面**

|  |
| --- |
| **相同点：**  Hibernate和Mybatis的二级缓存除了采用系统默认的缓存机制外，都可以通过实现你自己的缓存或为其他第三方缓    存方案，创建适配器来完全覆盖缓存行为。  **不同点：**  **Hibernate的二级缓存配置在SessionFactory生成的配置文件中进行详细配置，然后再在具体的表-对象映射中配置是那种缓存。**  MyBatis的二级缓存配置都是在每个具体的表-对象映射中进行详细配置，这样针对不同的表可以自定义不同的缓存机制。并且Mybatis可以在命名空间中共享相同的缓存配置和实例，通过Cache-ref来实现。 |

**比较：**

**Hibernate**具有良好的管理机制，用户不需要关注SQL，如果二级缓存出现脏数据，系统会保存，；

           Mybatis 在使用的时候要谨慎，避免缓存CAche 的使用。

#### Hibernate优势

1. Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。
2. Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便。
3. Hibernate数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。
4. Hibernate有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳。

#### Mybatis优势

1. MyBatis可以进行更为细致的SQL优化，可以减少查询字段。
2. MyBatis容易掌握，而Hibernate门槛较高。

## 一句话总结

**Mybatis**：小巧、方便、高效、简单、直接、半自动化

**Hibernate**：强大、方便、高效、复杂、间接、全自动化