

HIBERNATE与MYBATIS

开发速度的对比

- Mybatis框架相对简单很容易上手
- Hibernate上手较难，但基本的sql语句已经被封装好了，不需要写sql语句

开发工作量的对比

- 有相应的代码生成工具
 - Mybatis采用半自动化的方式，对高级查询需要手动编写SQL语句，以及ResultMap
 - Hibernate使用ORM（对象关系映射）来将Java对象映射到数据库表，有良好的映射机制，无需关心SQL的生成与结果映射，只需专注于业务流程

SQL优化方面

- Hibernate提供了一种称为HQL（Hibernate查询语言）的高级查询语言，可以方便地进行数据库访问控制
 - Hibernate的查询会将表中的所有字段查询出来，这一点会有性能消耗
 - 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
- MyBatis则需手动编写SQL语句，并支持动态SQL。可以按需求指定查询的字段
 - 这使得开发人员可以更直接地控制执行的SQL语句，从而提高了灵活性和性能

性能和扩展性

- 由于MyBatis采用半自动化的方式，它通常比Hibernate更轻量级，更适合对性能和扩展性要求较高的项目
- 而Hibernate则更适合对ORM支持和复杂业务逻辑要求较高的项目

日志方面

- Hibernate具有自己的日志统计
- Mybatis本身不带日志统计，使用Log4j进行日志记录

对象状态管理的对比

- Hibernate具有完整的对象状态管理机制，可以在操作数据库时自动维护对象状态的变化，Hibernate的开发者关注对象的状态（state），不必考虑SQL语句的执行
- MyBatis需要对对象进行详细的管理，MyBatis则需要开发人员手动维护对象状态，例如新增、更新或删除对象时需要调用相应的方法

集成

- Hibernate是全自动的ORM框架，因此在集成到现有应用程序中时需要更多的配置和调试
- MyBatis则具有更好的集成性，并且可以与其他技术栈（如Spring）轻松集成

缓存机制

- Hibernate一级缓存是Session缓存
- Hibernate二级缓存是SessionFactory级的缓存
- MyBatis默认情况下是没有开启缓存的，要开启二级缓存,你需要在你的SQL映射文件中添加一行: <cache/>

MYBATIS优势

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

HIBERNATE优势

- Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。
- 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。
- 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。
- 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。

总结

- Hibernate功能强大，数据库无关性好，O/R映射能力强，如果你对Hibernate相当精通，而且对Hibernate进行了适当的封装，那么你的项目整个持久层代码会相当简单，需要写的代码很少，开发速度很快，非常爽。
- Hibernate的缺点就是学习门槛不低，要精通门槛更高，而且怎么设计O/R映射，在性能和对象模型之间如何权衡取得平衡，以及怎样用好Hibernate方面需要你的经验和能力都很强才行。
- iBatis入门简单，即学即用，提供了数据库查询的自动对象绑定功能，而且延续了很好的SQL使用经验，对于没有那么高的对象模型要求的项目来说，相当完美。
- iBatis的缺点就是框架还是比较简陋，功能尚有缺失，虽然简化了数据绑定代码，但是整个底层数据库查询实际还是要自己写的，工作量也比较大，而且不太容易适应快速数据库修改。