现有框架使用问题

UserSession 仅适用于玩家与服务器直接相连的情形，换句话说，它就是纯粹意义上的玩家会话。

关于会话，分两大类

一类，服务器会话

另一类，玩家会话（直接面向玩家，不直接面向玩家），这两者的差别差别？能否整合？

UserMinaSession 与 InnerUserMinaSession 异同点

相同点：

都是表示玩家会话，可通过userId获取

不同点：

创建时机不一样，直接面向玩家的服务器，在连接建立时便创建了session，ioSessionId于userId对应UserMinaSession

不直接面向玩家的服务器，ioSessionId 对应多个 userId。连接建立时创建的是InnerServerMinaSession

InnerUserMinaSession 何时创建呢，如何创建？对于这样的服务器，同一个连接会跟多个玩家对应，因此，如果需要创建内部玩家session的话，必须要在数据包中提供玩家身份属性(如userId，如果不是消息头公共属性，就需要通过过滤器的方式写入固定位置的userId属性)。这样才能将数据包跟具体的玩家进行匹配

那么问题来了，userId该如何设置的，走框架消息头，走Filter，走其他

关于Handler，

直接面向玩家的服务器(Login,Gateway)

面向内部服务器的服务器,分两个类型

1. 服务器内部需要建立玩家会话，同时也需要建立服务器会话
2. 服务器内部只需要建立服务器会话，考虑改进现有SeverHandler

Dao层

目前缓存机制，以入库为例，先写db，紧接着写入cache。

BaseDao 实现是否存在问题，add时为什么还要查一遍

现有：

1. HibernateDao.getParameters 可以对@Term注解部分启动时分析并加入缓存，可以减少cpu计算，每次扫描全部域，检查是否使用该注解。Field.getAnnotations 是个同步方法

整体对性能提升不大，不是瓶颈。Jvm对首次调用的反射接口加载并扫描class，之后将加入到自身缓存中，缓存是软引用SoftRefrence。除非内存不足时，才会释放。

持久化框架 使用hibernate 是否合适？

《移动鹿鼎记》使用中碰到的问题

内部session测试步骤

使用合并后的mrd分支版本，结合主干进行测试