

## 软件复用——讨论课01 复用解决方案

1352960 张航

### 长连接心跳机制

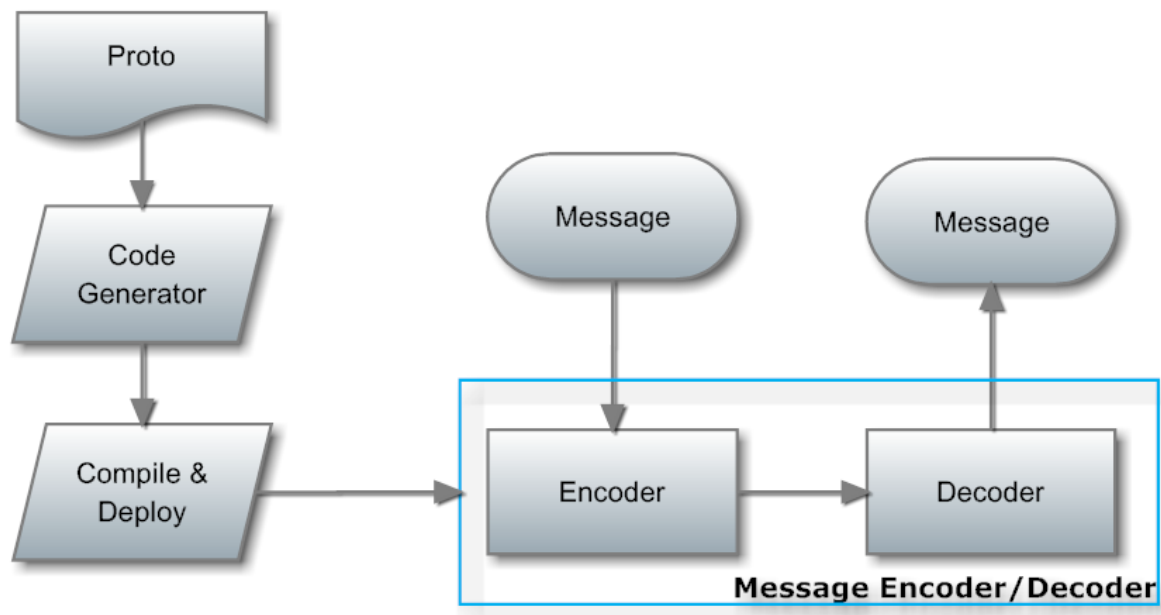
使用netty来达到实现Client与Server的长连接通讯和心跳检测的目的。（JDK1.8、netty5）

基本思路：netty服务端通过一个Map保存所有连接上来的客户端SocketChannel,客户端的Id作为Map的key。每次服务器端如果要向某个客户端发送消息，只需根据ClientId取出对应的SocketChannel,往里面写入message即可。添加了idleStateHandler用于监听链接idle，如果连接到达idle时间，这个handler会触发idleEvent，之后通过重写userEventTriggered方法，完成idle事件的处理。心跳检测通过IdleEvent事件，定时向服务端放送Ping消息，检测SocketChannel是否终断。

### 消息

#### protobuf

protobuf协议是由Google制定的，主要用于其内部的rpc调用和文件编码。与其他的编码协议如xml，json相比，protobuf有着更好的传输效率和压缩比率。原生的protobuf包括两部分内容：基于二进制的消息编码协议和基于proto元数据的代码生成器。首先，需要根据每条消息来编写对应的proto文件，然后使用google提供的代码生成器，基于proto文件来生成相应的编码器和解码器，然后使用生成的编/解码器来进行编/解码操作。对应的流程如下图：



这种方式的优势是代码静态生成，运行时不需要proto文件信息，而且可以根据具体的信息内容对代码进行优化。但缺点也十分明显：使用复杂（涉及到代码生成、编译、部署），改动成本高昂（需要重新生成、编译代码，并对代码进行部署），需要生成大量新代码（每个消息都需要一个独立的编码/解码器）。