1. Handler复用



@ChannelHandler.Sharable标识为可共享





1. 耗时操作引入业务线程池

*// xhandler 专属线程池*EventLoopGroup xhandlerGroup = **new** NioEventLoopGroup();

pipeline.addLast(xhandlerGroup, xHandller);



1. 分批推送

防止I/O线程被一个连接长时间占用。

导致用户体验变差。

1. 线程模型、责任链、ByteBuf  
     
   ByteBuf 高性能：对象复用、零拷贝、内存复用(1. )、

内存复用补充学习(1. 用C语言写一段内存申请的代码  
2. 后面的缓存中间件 memcached内存管理机制)

byte[] a  
byte[] b  
1. 创建新对象byte[] c = new byte[]

2. system.arraycopy  
  
ByteBuf ---- 逻辑buffer

对象复用

一定记得排查 是否 释放对象

对象才会到 对象池



