# 数据流

## IO模型

BIO(Blocking IO):同步并阻塞IO，一个连接一个线程

1：每个请求都需要独立的线程完成数据 Read，业务处理，数据 Write 的完整操作问题。

2：当并发数较大时，需要创建大量线程来处理连接，系统资源占用较大。

3：连接建立后，如果当前线程暂时没有数据可读，则线程就阻塞在 Read 操作上，造成线程资源浪费。

NIO(Non Blocking IO):同步非阻塞IO，一个请求一个线程

1：请求会注册到多路复用器Selector上，多路复用器轮询到连接有IO请求时才启动一个线程处理。

2：Netty封装了JDK的NIO，

AIO(Asynchronous IO):异步非阻塞IO，一个有效请求一个线程

1：

# Netty组件

## Bootstrap、ServerBootstrap

Bootstrap 意思是引导，一个 Netty 应用通常由一个 Bootstrap 开始，主要作用是配置整个 Netty 程序，串联各个组件，Netty 中 Bootstrap 类是客户端程序的启动引导类，ServerBootstrap 是服务端启动引导类。

## Future、ChannelFuture

在 Netty 中所有的 IO 操作都是异步的，不能立刻得知消息是否被正确处理。但是可以过一会等它执行完成或者直接注册一个监听，具体的实现就是通过 Future 和 ChannelFutures，他们可以注册一个监听，当操作执行成功或失败时监听会自动触发注册的监听事件。

## Channel

# Netty