Netty 核心概念

作者：289306290
原文链接：https://ld246.com/article/1582884333152
来源网站：链滴

netty 核心概念:

Channel
管道,是对 Socket 的封装,包含了一组 API,简化直接与 Socket 进行操作的复杂性.

EventLoopGroup
EventLoopGroup 是一个 EventLoop 池,包含很多 EventLoop
Netty 为每个 Channel 分配了一个 EventLoop,用于处理用户连接请求、对用户请求的处理等所有事。EventLoop 本身只是一个线程驱动,在其生命周期内
只会绑定一个线程,让该线程处理一个 Channel 的所有 IO 事件.
一个 Channel 一旦与一个 EventLoop 相绑定,那么在 Channel 的整个生命周期内是不能改变的。一个 EventLoop 可以与多个 Channel 绑定。
即 Channel 与 EventLoop 的关系是 n:1, 而 EventLoop 与线程的关系是 1:1。

ServerBootStrap
用于配置整个 Netty 代码,或将各个组件关联起来,服务端使用的事 ServerBootStrap,而客户端使用 BotStrap

ChannelHandler 与 ChannelPipeline
ChannelHandler 是对 Channel 中的数据的处理器,这些处理器可以是系统本身定义好的编码解码器,可以是用户自定义的，这些处理器会被统一
添加到一个 ChannePipeline 的对象中,然后按照添加的顺序对 Channel 中的数据进行依次处理。

ChannelFuture
Netty 中所有的 IO 操作都是异步的,所以 Netty 中定义了一个 ChannelFuture 对象作为异步操作的"言人",标识异步操作本身.如果想获取到该
异步操作的返回值。可以通过该异步操作对象的 addListener()方法为该异步操作添加监听器，为其册回调,当结果出来后马上调用执行.
Netty 的异步编程模型都是建立在 Future 与回调概念之上的。

Netty 执行流程大致流程:

Server 启动,从 parentGroup 中选出一个 NioEventLoop 对指定 port 的连接进行监听。
Client 连接指定 Server 的 port，创建 Channel。
Netty 从 childGroup 中选出一个 NioEventLoop 与该 Channel 绑定,用于处理该 Channel 中有的操作.
Client 通过 Channel 向 Server 发送数据包 ByteBuffer
Pipeline 中的处理器依次对 Channel 中的数据包进行处理
Server 如需向 Client 发送数据,则需将数据经 Pipeline 中的处理器处理形成 ByteBuffer 数据包</i>
Server 将数据包 ByteBuffer 通过 Channel 发送给 Client
