Netty 是什么

Netty 是一款基于 NIO 开发的**高性能和异步事件驱动**的网络通信框架，支持高并发，传输速度快，封装好。

高并发： 对比于 BIO，并发性能得到了很大提高。

传输快： Netty 的传输依赖于零拷贝特性，尽量减少不必要的内存拷贝，实现了更高效率的传输。

封装好： Netty 封装了 NIO 操作的很多细节，提供了易于使用调用接口。

典型的应用有：阿里分布式服务框架 Dubbo，默认使用 Netty 作为基础通信组件，还有 RocketMQ 也是使用 Netty 作为通讯的基础。

Netty 高性能的表现

IO 线程模型： 同步非阻塞，用最少的资源做更多的事。

内存零拷贝： 尽量减少不必要的内存拷贝，实现了更高效率的传输。

内存池设计： 申请的内存可以重用，主要指直接内存。内部实现是用一颗二叉查找树管理内存分配情况。

串形化处理读写： 避免使用锁带来的性能开销。

高性能序列化协议： 支持 protobuf 等高性能序列化协议。

Netty 的零拷贝

Netty 的零拷贝主要包含三个方面：

Netty 的接收和发送 ByteBuffer 采用 DIRECT BUFFERS，使用堆外直接内存进行 Socket 读写，不需要进行字节缓冲区的二次拷贝。如果使用传统的堆内存（HEAP BUFFERS）进行 Socket 读写，JVM 会将堆内存 Buffer 拷贝一份到直接内存中，然后才写入 Socket 中。相比于堆外直接内存，消息在发送过程中多了一次缓冲区的内存拷贝。

Netty 提供了组合 Buffer 对象，可以聚合多个 ByteBuffer 对象，用户可以像操作一个 Buffer 那样方便的对组合 Buffer 进行操作，避免了传统通过内存拷贝的方式将几个小 Buffer 合并成一个大的 Buffer。

Netty 的文件传输采用了 transferTo 方法，它可以直接将文件缓冲区的数据发送到目标

Channel，避免了传统通过循环 write 方式导致的内存拷贝问题。

TCP 粘包/拆包的问题解决

TCP 是一个“流”协议，所谓流，就是没有界限的一长串二进制数据。TCP 作为传输层协议并不不了解上层业务数据的具体含义，它会根据 TCP 缓冲区的实际情况进行数据包的划分，所以在业务上认为是一个完整的包，可能会被 TCP 拆分成多个包进行发送，也有可能把多个小的包封装成一个大的数据包发送，这就是所谓的 TCP 粘包和拆包问题。

粘包问题的解决策略： 由于底层的 TCP 无法理解上层的业务数据，所以在底层是无法保证数据包不被拆分和重组的，这个问题只能通过上层的应用协议栈设计来解决。业界的主流协议的解决方案，可以归纳如下：

1.消息定长，报文大小固定长度，例如每个报文的长度固定为 200 字节，如果不够空位补空格

2.包尾添加特殊分隔符，例如每条报文结束都添加回车换行符（例如 FTP 协议）或者指定特殊字符作为报文分隔符，接收方通过特殊分隔符切分报文区分；

3.将消息分为消息头和消息体，消息头中包含表示信息的总长度（或者消息体长度）的字段；

4.更复杂的自定义应用层协议。

Netty 粘包和拆包解决方案： Netty 提供了多个解码器，可以进行分包的操作，分别是：

LineBasedFrameDecoder（回车换行结尾）

DelimiterBasedFrameDecoder（添加特殊分隔符报文来分包）

FixedLengthFrameDecoder（使用定长的报文来分包）

LengthFieldBasedFrameDecoder（按照应用层数据包的大小，拆包）

Reactor 模式

反应器设计模式(Reactorpattern)是一种为处理并发服务请求，并将请求提交到一个或者多个服务处理程序的事件设计模式。当客户端请求抵达后，服务处理程序使

用多路分配策略，由一个非阻塞的线程来接收所有的请求，然后派发这些请求至相关的工作线程进行处理。

李林峰：Netty 系列之 Netty 高性能之道