什么是Netty

Netty 是一个基于NIO的客户、服务器端的编程框架，使用Netty 可以确保你快速和简单的开发出一个网络应用，例如实现了某种协议的客户、服务端应用。Netty相当于简化和流线化了网络应用的编程开发过程，例如：基于TCP和UDP的socket服务开发

netty能够受到青睐的原因有三：

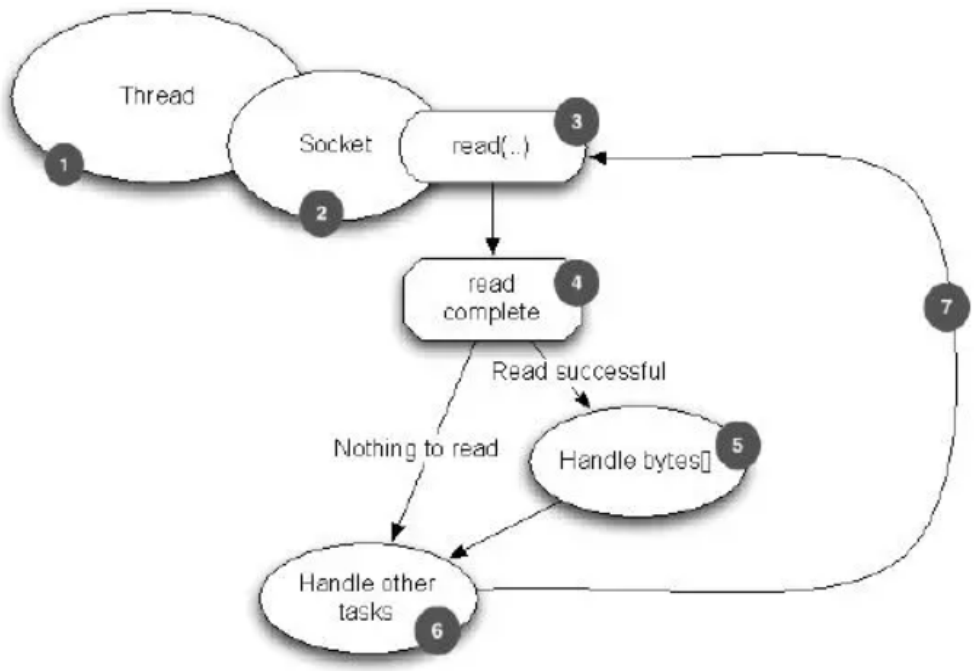
1. 并发高
2. 传输快
3. 封装好

高并发

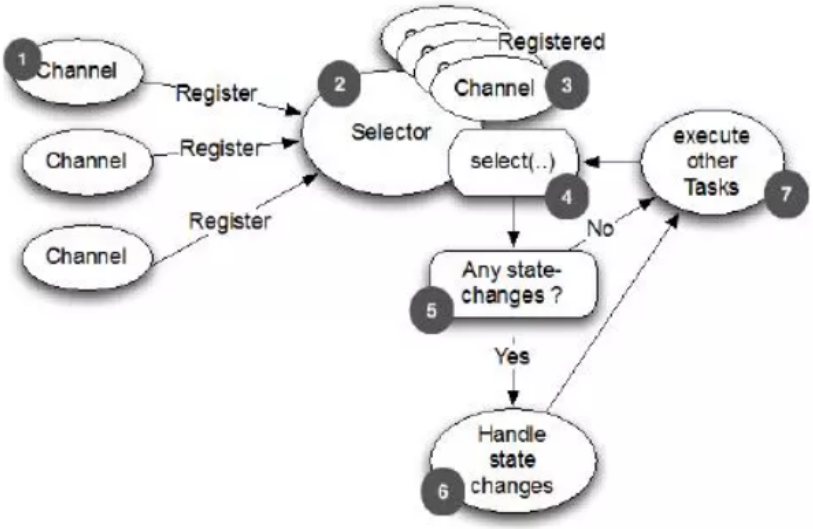
NIO和BIO的区别

在BIO中，等待客户端发数据这个过程是阻塞的，这样就造成了一个线程只能处理一个请求的情况，而机器能支持的最大线程数是有限的，这就是为什么BIO不能支持高并发的原因。

而NIO中，当一个Socket建立好之后，Thread并不会阻塞去接受这个Socket，而是将这个请求交给Selector，Selector会不断的去遍历所有的Socket，一旦有一个Socket建立完成，他会通知Thread，然后Thread处理完数据再返回给客户端——**这个过程是不阻塞的**，这样就能让一个Thread处理更多的请求了

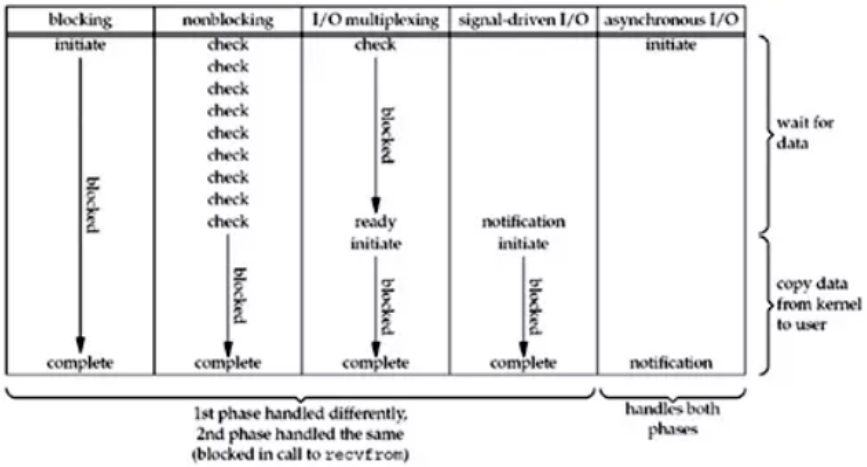


BIO的处理流程



NIO的处理流程

五种IO模型的处理流程：



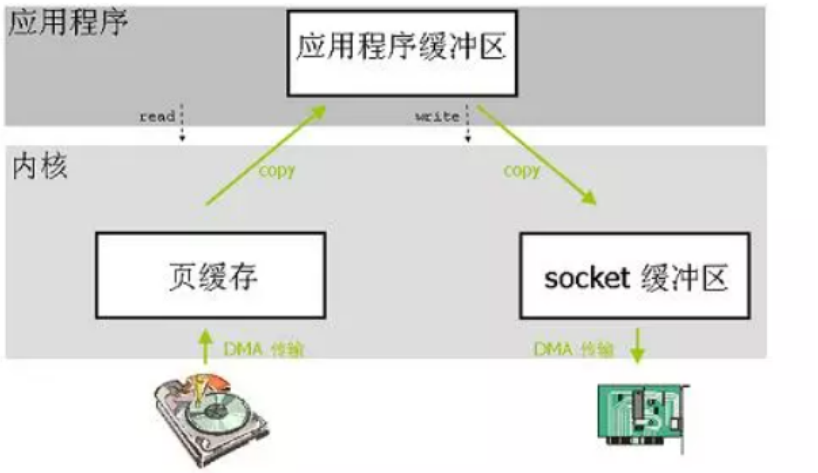
五种常见的IO模型

- BIO，同步阻塞IO，阻塞整个步骤，如果连接少，他的延迟是最低的，因为一个线程只处理一个连接，适用于少连接且延迟低的场景，比如说数据库连接。
- NIO，同步非阻塞IO，阻塞业务处理但不阻塞数据接收，适用于高并发且处理简单的场景，比如聊天软件。
- 多路复用IO，他的两个步骤处理是分开的，也就是说，一个连接可能他的数据接收是线程a完成的，数据处理是线程b完成的，他比BIO能处理更多请求。
- 信号驱动IO，这种IO模型主要用在嵌入式开发，不参与讨论。
- 异步IO，他的数据请求和数据处理都是异步的，数据请求一次返回一次，适用于长连接的业务场景。

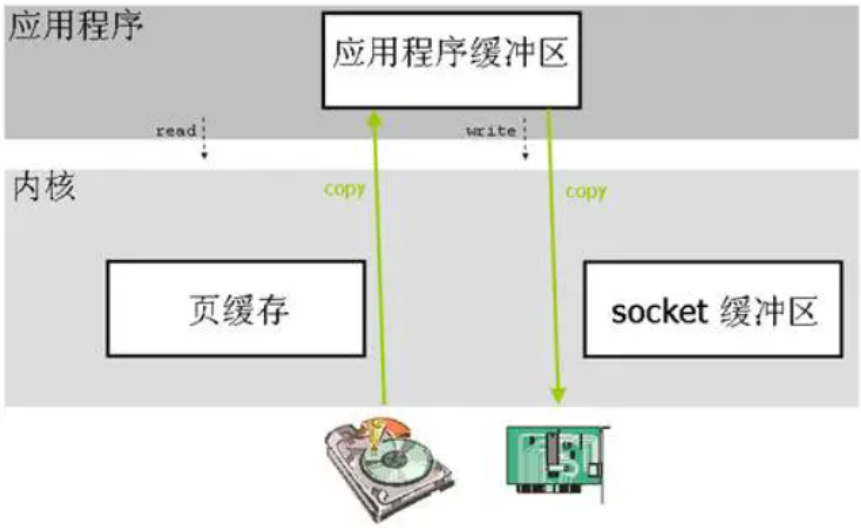
传输快

Netty的传输快其实也是依赖了NIO的一个特性——*零拷贝*。我们知道，Java的内存有堆内存、栈内存和字符串常量池等等，其中堆内存是占用内存空间最大的一块，也是Java对象存放的地方，一般我们的数据如果需要从IO读取到堆内存，中间需要经过Socket缓冲区，也就是说一个数据会被拷贝两次才能到达他的的终点，如果数据量大，就会造成不必要的资源浪费。

Netty针对这种情况，使用了NIO中的另一大特性——零拷贝，当他需要接收数据的时候，他会在堆内存之外开辟一块内存，数据就直接从IO读到了那块内存中去，在netty里面通过ByteBuf可以直接对这些数据进行直接操作，从而加快了传输速度。



传统数据拷贝

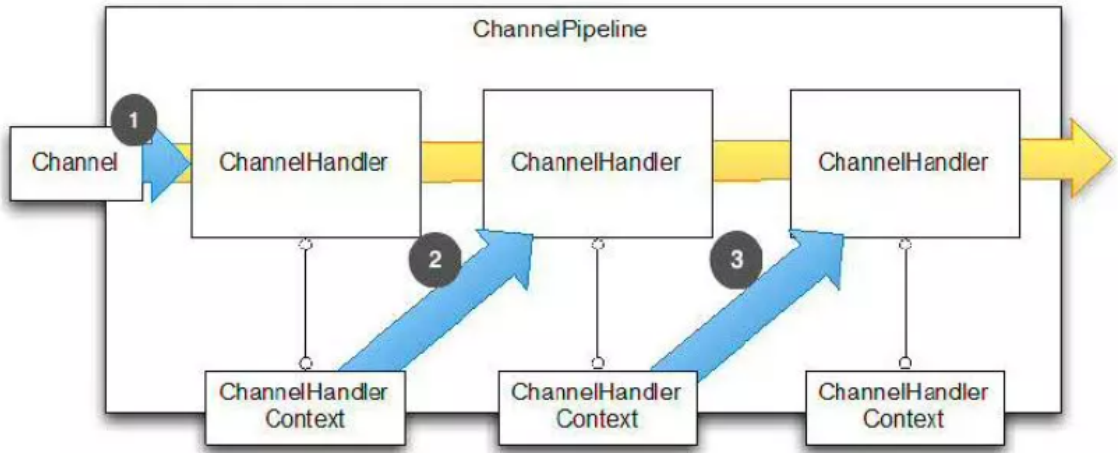


零拷贝

封装好

- Channel

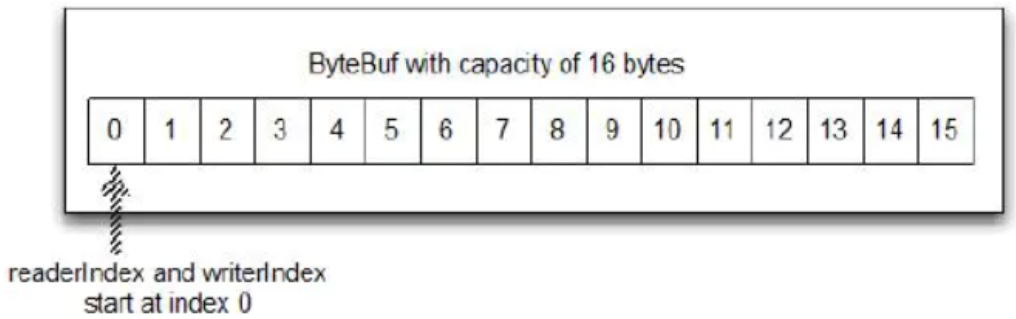
数据传输流，与channel相关的概念有以下四个，上一张图让你了解netty里面的Channel。



- Channel，表示一个连接，可以理解为每一个请求，就是一个Channel。
- **ChannelHandler**，核心处理业务就在这里，用于处理业务请求。
- ChannelHandlerContext，用于传输业务数据。
- ChannelPipeline，用于保存处理过程需要用到的ChannelHandler和ChannelHandlerContext。

- ByteBuf

ByteBuf是一个存储字节的容器，最大特点就是**使用方便**，它既有自己的读索引和写索引，方便你对整段字节缓存进行读写，也支持get/set，方便你对其中每一个字节进行读写，他的数据结构如下图所示：



他有三种使用模式：

1. Heap Buffer 堆缓冲区

堆缓冲区是ByteBuf最常用的模式，他将数据存储在堆空间。

2. Direct Buffer 直接缓冲区

直接缓冲区是ByteBuf的另外一种常用模式，他的内存分配都不发生在堆，jdk1.4引入的nio的ByteBuffer类允许jvm通过本地方法调用分配内存，这样做有两个好处

- 通过免去中间交换的内存拷贝, 提升IO处理速度; 直接缓冲区的内容可以驻留在垃圾回收扫描的堆区以外。
- DirectBuffer 在 -XX:MaxDirectMemorySize=xxM大小限制下, 使用 Heap 之外的内存, GC对此”无能为力”,也就意味着规避了在高负载下频繁的GC过程对应用线程的中断影响.

3. Composite Buffer 复合缓冲区

复合缓冲区相当于多个不同ByteBuf的视图，这是netty提供的，jdk不提供这样的功能。

除此之外，他还提供一大堆api方便你使用，在这里我就不一一列出了，具体参见ByteBuf字节缓存。

- Codec

Netty中的编码/解码器，通过他你能完成字节与pojo、pojo与pojo的相互转换，从而达到自定义协议的目的。

在Netty里面最有名的就是HttpRequestDecoder和HttpResponseEncoder了。