

先弄清楚什么是**内核缓冲区**与什么是**用户缓冲区**？

内核缓冲区是指只有操作系统所具有操作权限的内存资源部分中的一个部分用作数据读写的内存资源。

这部分的资源主要是与用户缓冲区、以及底层物理辅助外存或者是网络资源进行交互。它的目的也是为了降低读写的系统开销，因为无论是谁需要IO操作都需要操作系统在内核状态下进行。

用户缓冲区是与内核缓冲区以及程序进行交互的，因为程序需要IO操作时，需要到内核状态，然后将需要操作的数据与内核缓冲区进行交互，但是如果频繁的切换状态，必然也会造成性能下降。

在看看JVM：

在《深入理解在Java虚拟机》中讲到JDK 1.4中新加入了NIO（New Input/Output）类，引入了一种基于通道（Channel）与缓冲区（Buffer）的I/O方式，**它可以使用Native函数库直接分配堆外内存**[size=16px]，然后通过一个存储在Java堆里面的DirectByteBuffer对象作为这块内存的引用进行操作。这样能在一些场景中显著提高性能，因为避免了在Java堆和Native堆中来回复制数据。

（[不使用直接内存时，为什么对文件进行读写操作时需要在Java堆和Native堆来回复制？ - 知乎 (zhihu.com)](https://www.zhihu.com/question/436881266/answer/1654267585)）

[Netty对零拷贝(Zero Copy)三个层次的实现 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/88599349)

再理解两个概念：

Native堆-堆外内存：也就是Java堆外的用户空间中的堆内存

Java堆-堆内内存：就是jvm中的堆

（[JAVA直接内存（堆外内存）\_懒猫的脚印-CSDN博客\_java 堆外内存](https://blog.csdn.net/y3over/article/details/88791958?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control)）

[用Java实现JVM第九章《本地方法调用》 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/103567048)

下面我画个图：（直接缓冲区是要做到Java层面的零次拷贝，也就是减少了堆外内存与堆内内存拷贝的次数）



[深入理解NIO零拷贝及用户空间与内核空间切换 - 简书 (jianshu.com)](https://www.jianshu.com/p/f28eeca6fdae)

**所谓的零拷贝：其实指的是通道到通道的直接数据传输，不要在内核缓冲区与用户缓冲之间以及用户缓冲区与用户缓冲区之间进行不必要的数据重复拷贝。而能否将通道的数据直接传送到通道，Java都是调用的本地方法，也就是这完全依赖于操作系统，如果操作系统并不支持那么Java也就无法实现零拷贝。**

[java IO专题二： NIO读取文件并通过socket发送，最少拷贝了几次？堆外内存和所谓的零拷贝到底是什么关系\_mb5ffbc824a2582的技术博客\_51CTO博客](https://blog.51cto.com/u_15081050/2593340)

[深入理解NIO零拷贝及用户空间与内核空间切换 - 简书 (jianshu.com)](https://www.jianshu.com/p/f28eeca6fdae)

***（这篇文章很重要）***

**零拷贝，可以认为有内核层面的零拷贝和java层面的零拷贝两种。而所谓的"零"并不是一次拷贝都没有，而是在不同的场景下尽可能减少拷贝次数。因此利用DirectBuffer减少拷贝次数和利用channel.transferTo，都可以算作是"零拷贝"的一种表现。当然真正被认可的零拷贝可能就只有transferTo了。**

**而且将通道的数据直接输出或者输入到通道一般都是FileChannel进行操作。也就是是针对于文件的操作。**

**再有一句：一般是只有Buffer类才会提供堆外内存的分配，而一般对象与自己建立缓冲区数组都是建立在堆内内存中。**