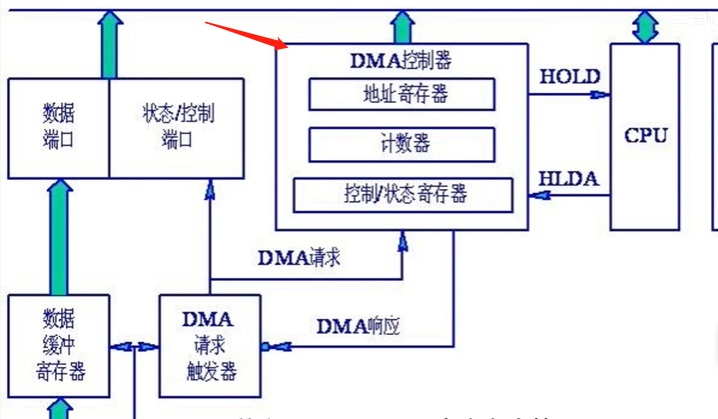
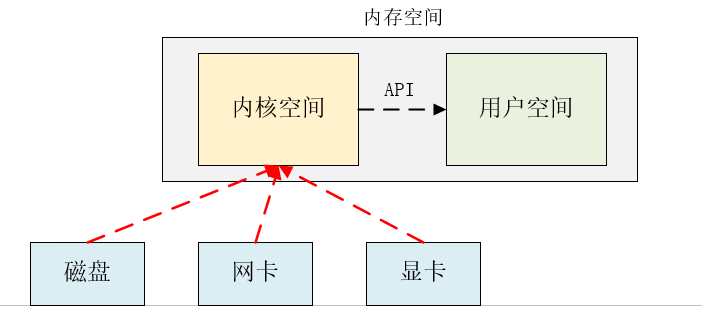
## 零拷贝原理

1. DMA介绍

DMA又叫直接内存访问（Direct Memory Access），是独立于CPU的一个电子元器件，可以在不消耗CPU性能的情况下，进行内存和外用设备（磁盘，显示器，网卡等）数据的沟通

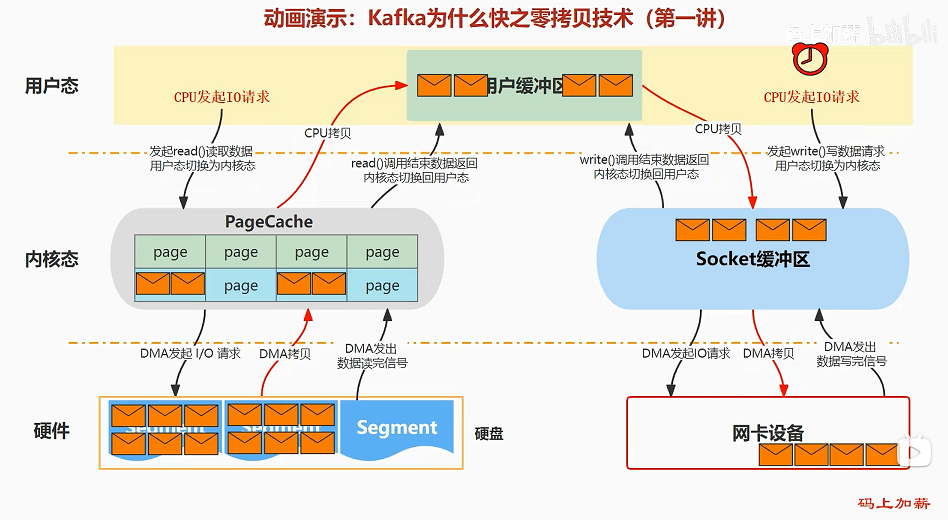


1. 数据的内核态和用户态
2. 为防止用户代码对计算机系统造成严重的损坏，计算机系统不允许用户代码直接调用系统的磁盘文件、网卡，显卡等，而是通过计算机系统给定的API来进行间接调用。
3. 这就导致计算机内存（RAM）分为大致的两块：用来存储用户代码的用户空间（如JVM）

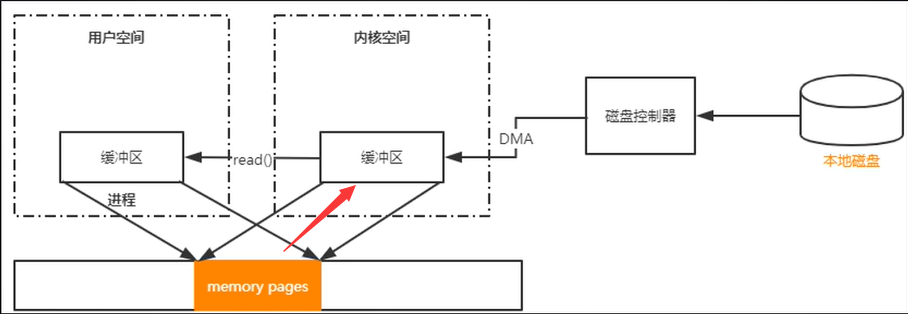
其中数据在内核空间称为内核态，在用户空间称为用户态

下面将以Kafka发送数据为例，讲解零拷贝的几种方式

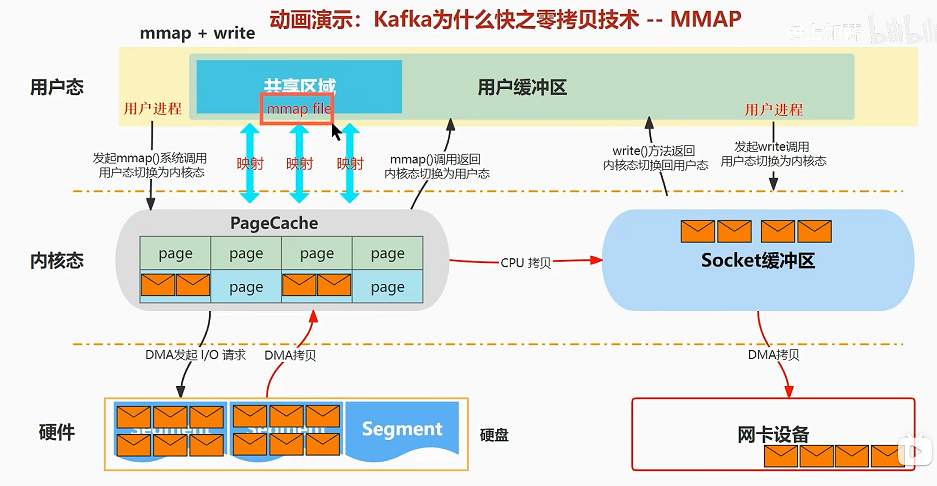
1. 传统的数据拷贝过程
2. 当用户空间向内核空间发送read（）命令时，用户线程会发生阻塞，CPU会收到命令，并向DMA发送数据拷贝的命令；
3. DMA收到数据拷贝命令后，会将外接设备（磁盘，网卡等）中的数据，拷贝至内核空间中，并通知CPU数据已经拷贝完成；
4. 然后CPU会将内核空间的数据，拷贝至用户空间，并通知用户空间，数据已拷贝完毕；
5. 用户空间收到拷贝完毕的指令后，线程由阻塞状态变为运行状态，并由用户空间向内核空间发送write()指令；
6. 内核空间在收到write()指令后，CPU会将用户空间的数据拷贝至内核空间的Socket缓冲区，并发送指令给DMA；
7. DMA将内核空间Socket缓冲区的数据，拷贝至外接设备（如磁盘、网卡等）
8. 整个过程发生了4次数据拷贝，2次CPU拷贝，2次DMA拷贝，效率较低



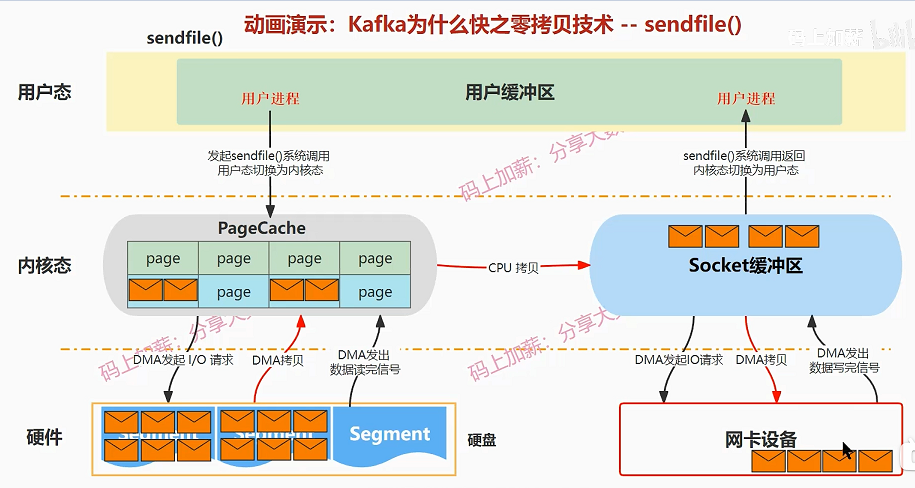
1. 零拷贝方式---mmap拷贝
2. MMAP其实就是将内核空间的数据，同用户空间的虚拟内存地址进行映射，以达到内核空间和用户空间共享内存数据的效果，来减少拷贝次数的目的。



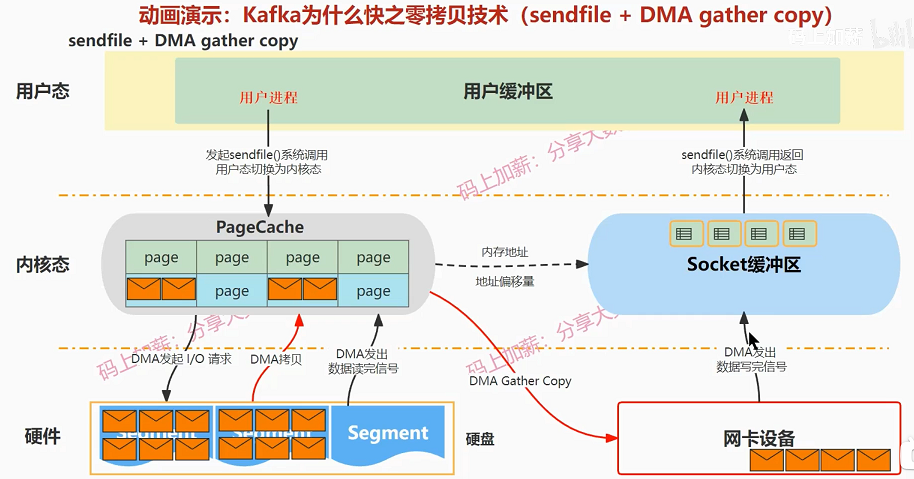
1. 利用mmap函数进行零拷贝的过程如下：
2. 首先，用户空间会向内核空间发送mmap函数，同时mmap函数会将用户空间的虚拟内存和内核空间的内存建立映射关系；
3. 同时，CPU在收到mmap指令后，会向DMA发送指令，将外接设备（磁盘，网卡）等的数据拷贝至内核空间中；
4. 然后DMA向CPU发送拷贝完毕相应，CPU向用户空间发送拷贝完毕指令；
5. 然后用户空间向内核空间发送write指令，CPU在接收到write指令后，会直接在内核空间中，将数据拷贝至socket缓冲区中；
6. 然后CPU向DMA发送指令，由DMA将数据拷贝至网卡等设备中
7. 整个过程只有3次拷贝，1次CPU拷贝，2次DMA拷贝,4次上下文切换（用户态，内核态之间切换）



1. 零拷贝方式--sendfile
2. sendfile是在mmap的基础上，减少了上下文切换的次数（即内核态和用户态之间的切换），来提高效率。（上下文切换也是消耗时间的）
3. 具体过程如下：
4. 用户空间向内核空间发送sendFile函数；
5. CPU在收到sendFile函数后，向DMA发送指令，由DMA将数据从外接设备（磁盘，网卡等）拷贝至内核空间；
6. 然后CPU会直接将内核空间的数据拷贝至socket缓冲区，并向DMA发送指令，由DMA将数据发送至外接设备（磁盘，网卡等）；
7. 然后DMA返回拷贝完成响应，CPU向用户空间返回sendFile执行完成相应；
8. 整个过程3次数据拷贝，1次CPU拷贝，2次DMA拷贝，2次上下文切换



1. 零拷贝方式---sendFile+DMA gather拷贝
2. 在Linux2.4之后，对sendFile函数进行了改进，数据拷贝过程无CPU参与，具体过程如下：
3. 用户空间向内核空间发送sendFile函数；
4. CPU在收到sendFile函数后，向DMA发送指令，由DMA将数据从外接设备（磁盘，网卡等）拷贝至内核空间；
5. 然后DMA gather技术会将内核空间数据的地址以及偏移量，发送到socket缓冲区
6. 然后DMA就会根据socket缓冲区中的数据地址以及偏移量，将内核空间中的数据拷贝至外接设备（网卡，磁盘等）
7. 然后DMA返回拷贝完成响应，CPU向用户空间返回sendFile执行完成相应；
8. 整个过程2次DMA数据拷贝，2次上下文切换,全程无CPU拷贝的参与，大大提高了效率；



1. 零拷贝方式---splice
2. splice函数就是在内核缓冲区建立数据和socket缓冲区的通道，由DMA完成数据到socket缓冲区的传送
3. 具体过程如下：
4. 用户空间向内核空间发送splice函数;
5. CPU在收到splice函数后，向DMA发送指令，由DMA将数据从外接设备（磁盘，网卡等）拷贝至内核空间；
6. 然后splice函数会创建从内核空间数据到socket缓冲区的通道，并且由DMA将数据拷贝至socket缓冲区中，进而拷贝至网卡；
7. 然后DMA返回拷贝完成响应，CPU向用户空间返回sendFile执行完成相应；
8. 整个过程2次DMA数据拷贝，2次上下文切换,全程无CPU拷贝的参与

