

蚂蚁二面，面试官问我零拷贝的实现原理，当场懵了...

原创：占小狼 占小狼的博客 10月28日

点击上方蓝色字体，选择“设为星标”

优质文章，及时送达

"狼哥，面试又跪了，碰到了知识盲区"

"哪个？"

"一面还可以，二面面试官问我零拷贝的原理，懵逼了...这块内容没去研究过"

"哦，这个知识点，我之前应该有讲过，你没注意到？"

"这东西工作中用不到，可能被我忽略了"

"啧啧啧..."

"哎，有空和我讲讲？"

"先从简单开始，实现下这个场景：从一个文件中读出数据并将数据传到另一台服务器上？"

"为啥写这个？"

"你先写"

"行..."

1分钟后

"我写了伪代码"

```
File.read(file, buf, len);  
Socket.send(socket, buf, len);
```

"这里涉及到了几次数据拷贝？"

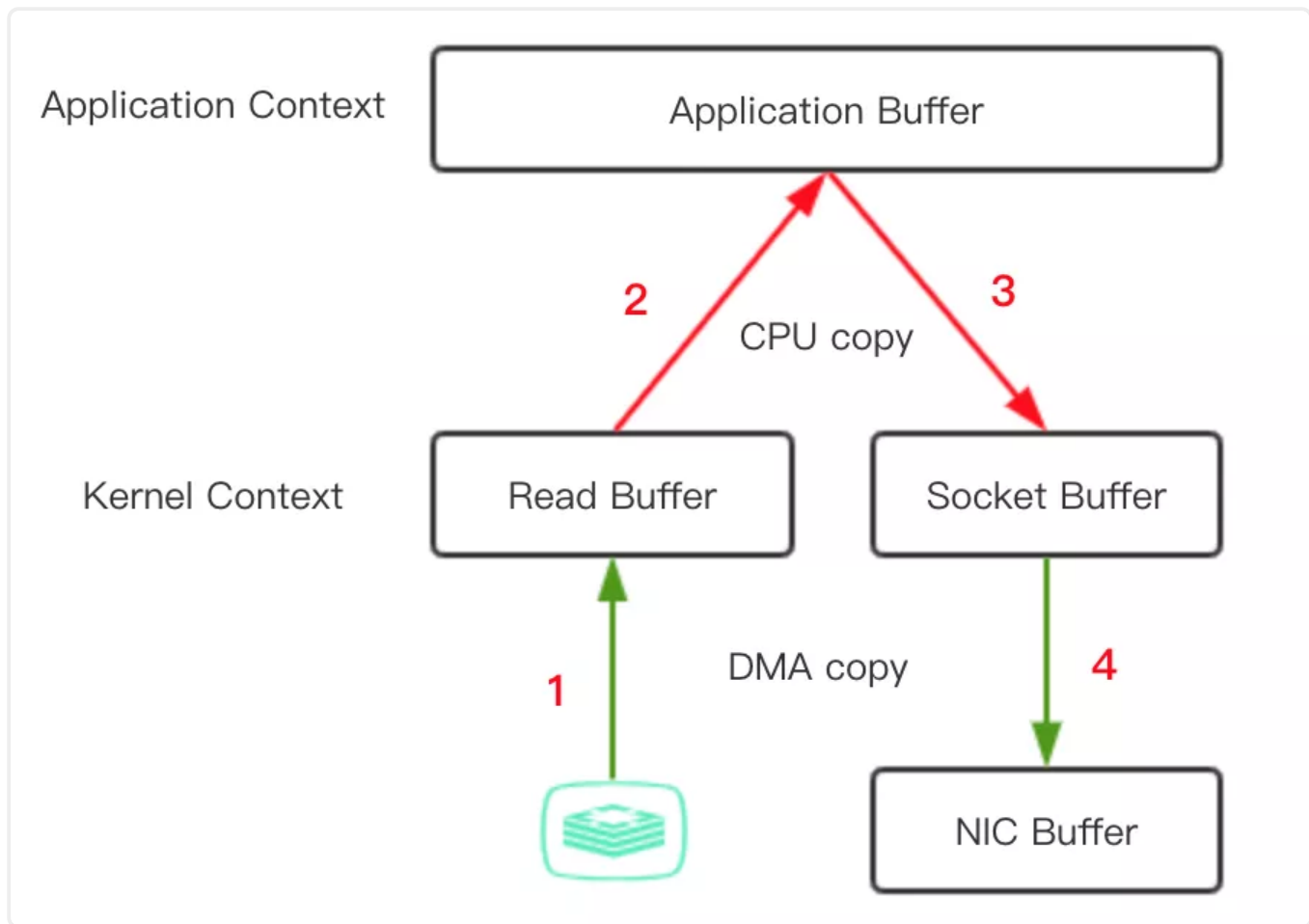
"2次？磁盘拷贝到内存，内存拷贝到Socket？"

"emmm，怪不得挂了，不冤"

"这种方式一共涉及了4次数据拷贝，知道用户态和内核态的区别吗？"

"了解"

"行，文字有点干瘪，你先看这个图"



1、应用程序中调用 `read()` 方法，这里会涉及到一次上下文切换（用户态->内核态），底层采用DMA（direct memory access）读取磁盘的文件，并把内容存储到内核地址空间的读取缓存区。

2、由于应用程序无法访问内核地址空间的数据，如果应用程序要操作这些数据，得把这些内容从读取缓冲区拷贝到用户缓冲区。`read()` 调用的返回引发一次上下文切换（内核态->用户态），现在数据已经被拷贝到了用户地址空间缓冲区，如果有需要，可以操作修改这些内容。

3、我们最终目的是把这个文件内容通过Socket传到另一个服务中，调用Socket的 `send()` 方法，又涉及到一次上下文切换（用户态->内核态），同时，文件内容被进行第三次拷贝，这次的缓冲区与目标套接字相关联，与读取缓冲区无关。

4、`send()` 调用返回，引发第四次的上下文切换，同时进行第四次拷贝，DMA把数据从目标套接字相关的缓存区传到协议引擎进行发送。

"整个过程中，过程1和4是由DMA负责，并不会消耗CPU，只有过程2和3的拷贝需要CPU参与，整明白了？"

"我消化一下..."

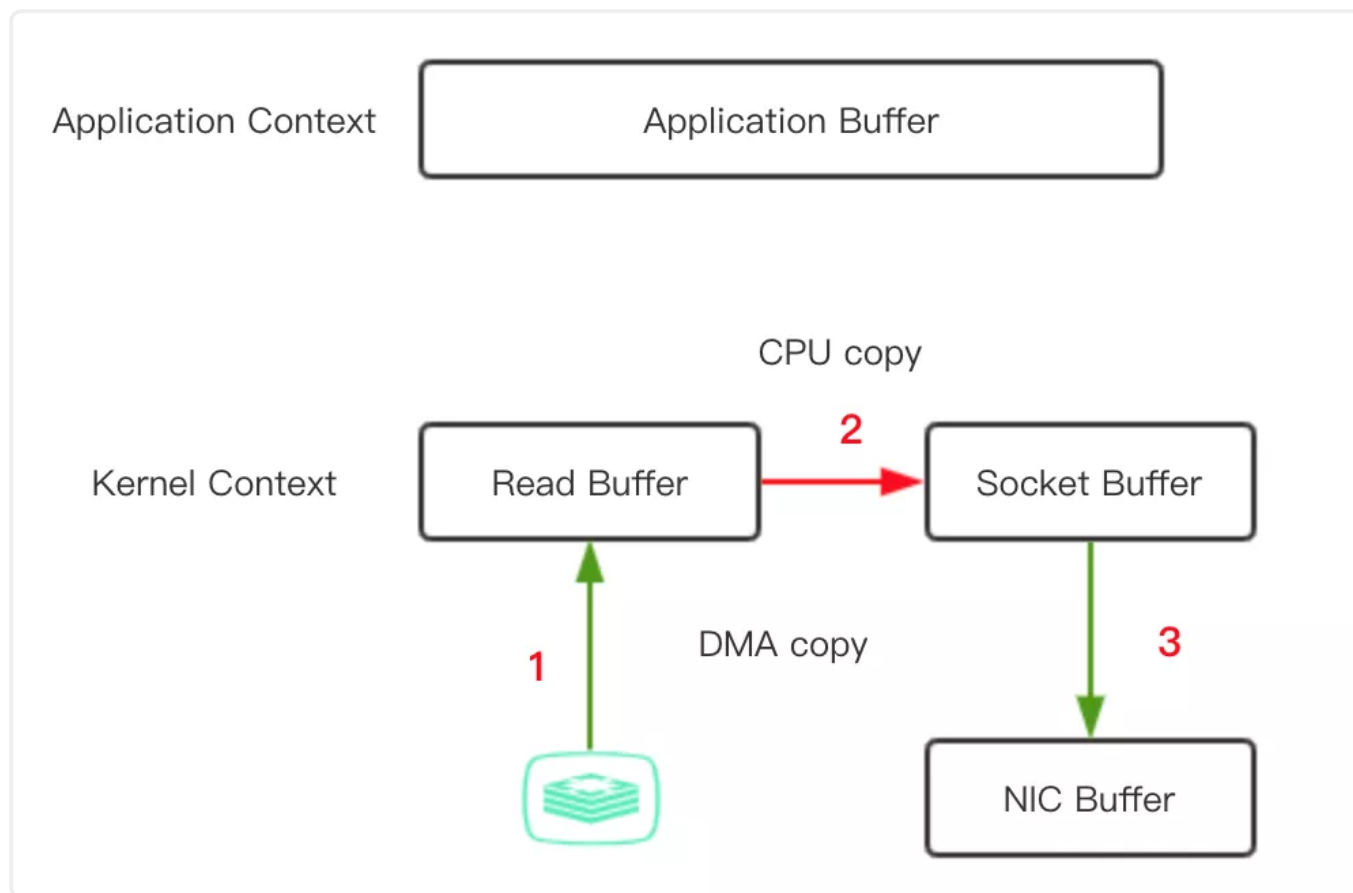
半小时后...

"狼哥，感觉这个过程中好几次的拷贝都是多余的，很影响性能啊"

"对，所以才有了零拷贝技术"

"具体咋实现？"

"慢慢来，如果在应用程序中，不需要操作内容，过程2和3显然是多余的，如果可以直接把内核态读取缓存缓冲区数据直接拷贝到套接字相关的缓存区，是不是可以达到目的？"



这种实现，可以有以下几点改进：

- 上下文切换的次数从四次减少到了两次
- 拷贝次数从四次减少到了三次（其中DMA copy 2次，CPU copy 1次）

"怎么实现？"

"在Java中，FileChannel的transferTo() 方法可以实现这个过程，该方法将数据从文件通道传输到给定的可写字节通道， 上面的 file.read() 和 socket.send() 调用动作可以替换为 transferTo() 调用"

```
public void transferTo(long position, long count, WritableByteChannel target);
```

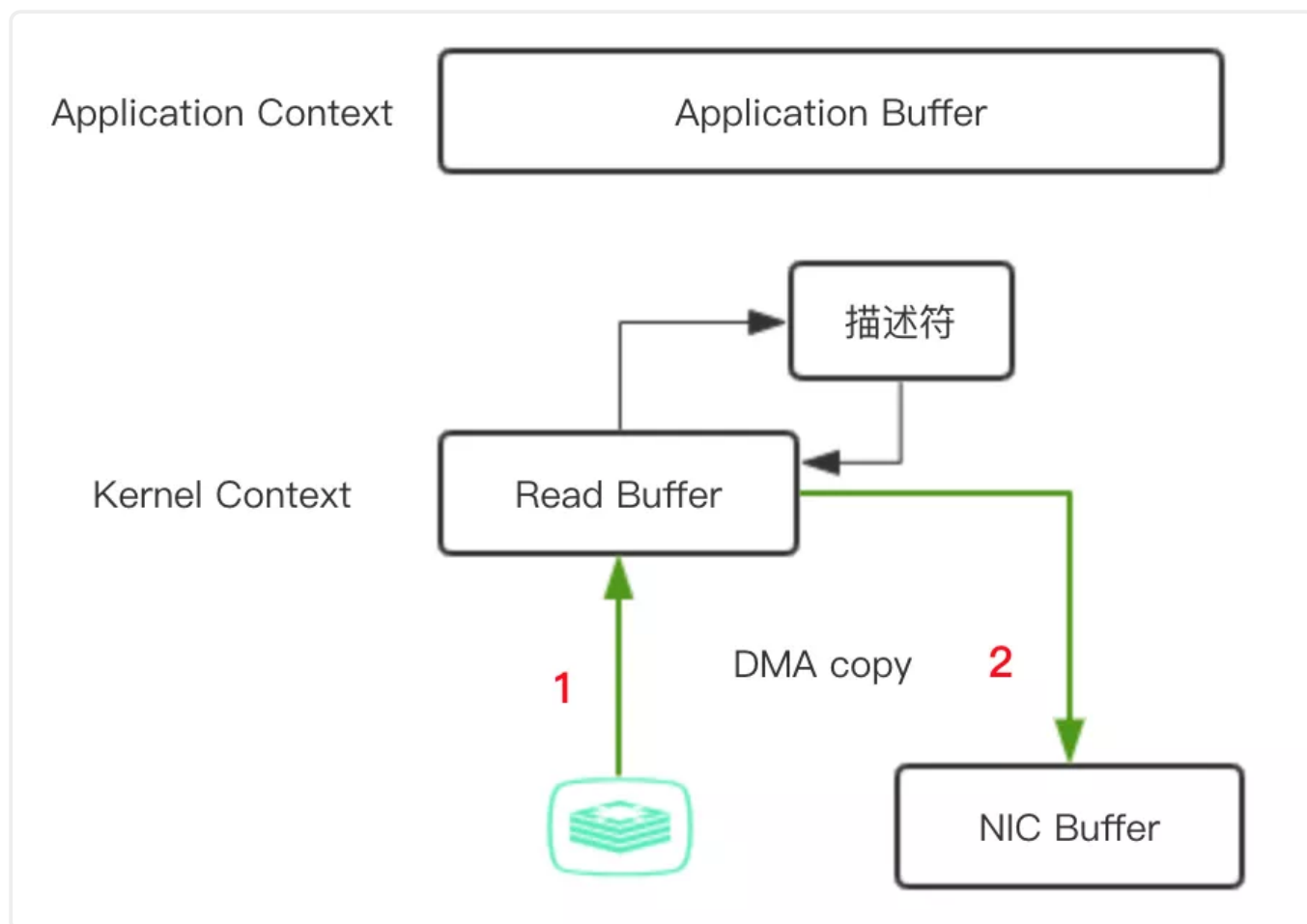
在 UNIX 和各种 Linux 系统中，此调用被传递到 sendfile() 系统调用中，最终实现将数据从一个文件描述符传输到了另一个文件描述符。

"这样确实改善了很多，但还没达到零拷贝的要求（还有一次cpu参与的拷贝），还有其它黑技术？"

"对的，如果底层网络接口卡支持收集操作的话，就可以进一步的优化。"

"怎么说？"

在 Linux 内核 2.4 及后期版本中，针对套接字缓冲区描述符做了相应调整，DMA自带了收集功能，对于用户方面，用法还是一样，只是内部操作已经发生了改变：



具体过程：

- 1、transferTo() 方法使用 DMA 将文件内容拷贝到内核读取缓冲区。
- 2、避免了内容的整体拷贝，只把包含数据位置和长度信息的描述符追加到套接字缓冲区，DMA 引擎直接把数据从内核缓冲区传到协议引擎，从而消除了最后一次 CPU 参与的拷贝动作。

近期热文

- [面试官问：MySQL的自增ID用完了，怎么办？](#)
- [ArrayList插入1000w条数据之后，我怀疑了jvm...](#)
- [面试官问：Redis 内存满了怎么办？我想不到！](#)



最后，分享一份面试宝典《**Java核心知识点整理.pdf**》，覆盖了JVM、锁、高并发、反射、Spring原理、微服务、Zookeeper、数据库、数据结构等等。

获取方式：点“**在看**”，关注公众号并回复 **Java** 领取，更多内容陆续奉上。

明天见(。·ω·。)/♡