如何把 committer 頭銜 zero-copy 到履歷上

低 cp 值高技術門檻的不歸路

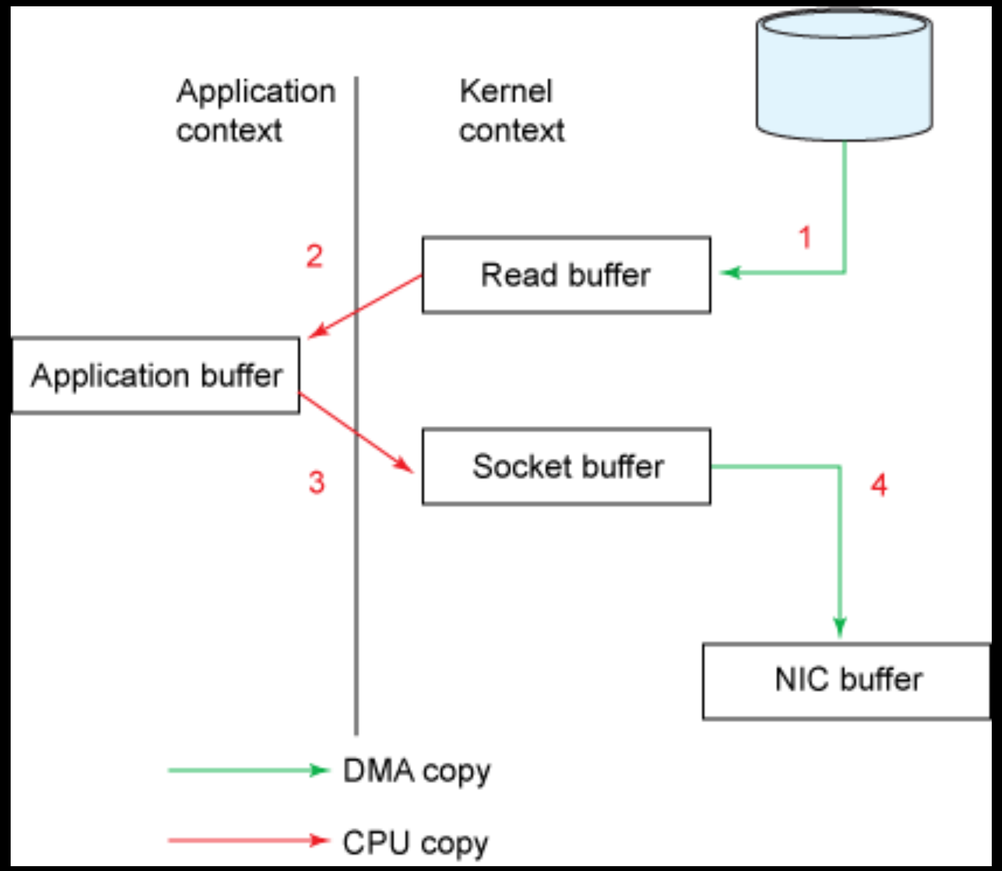
蔡嘉平

從 file -> network
傳統手法

① var buf = ByteBuffer.allocate(100);

② File.read(fileDesc, buf, len);

③ Socket.send(socket, buf, len);



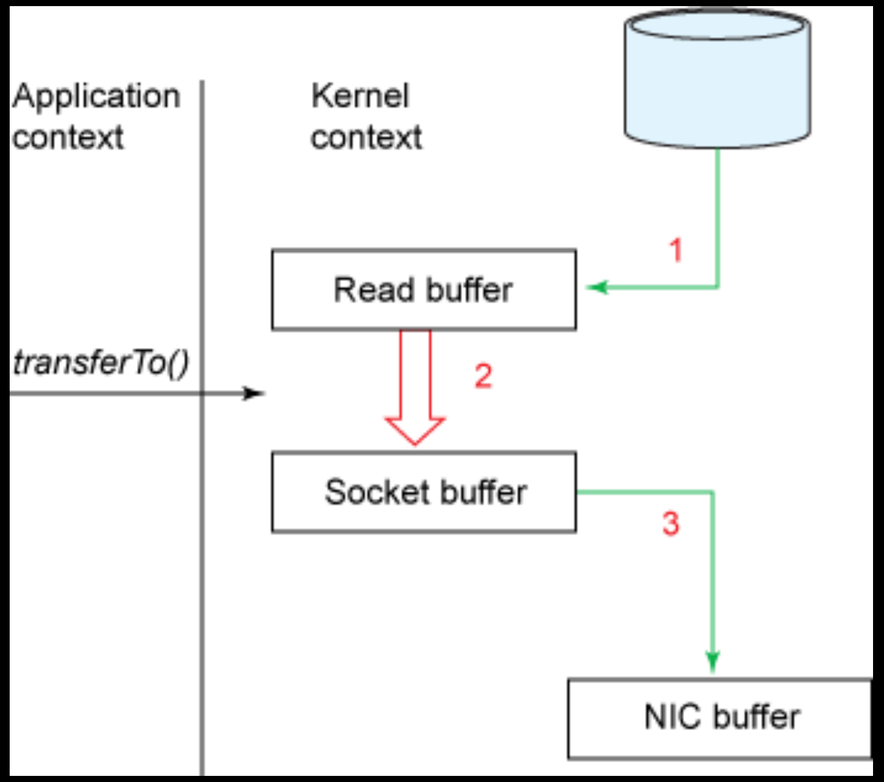
https://developer.ibm.com/articles/j-zerocopy/

從 file -> network
Zero-Copy

① var fileChannel = file.getChannel();

② var socketChannel =
socket.channel();

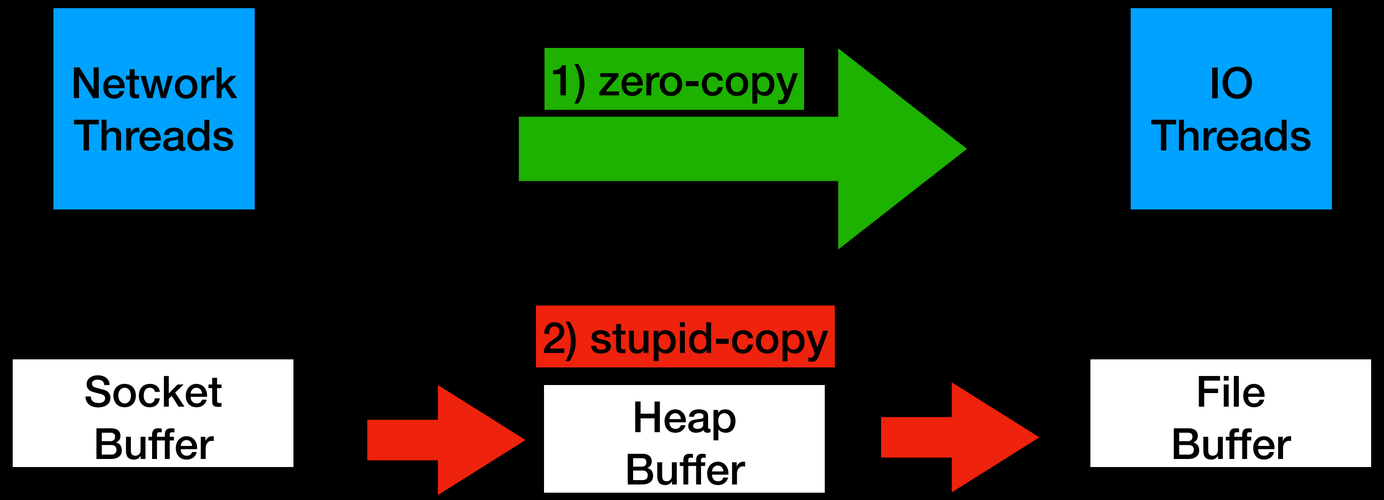
③ fileChannel.transferTo(position, count,
socketChannel);



https://developer.ibm.com/articles/j-zerocopy/

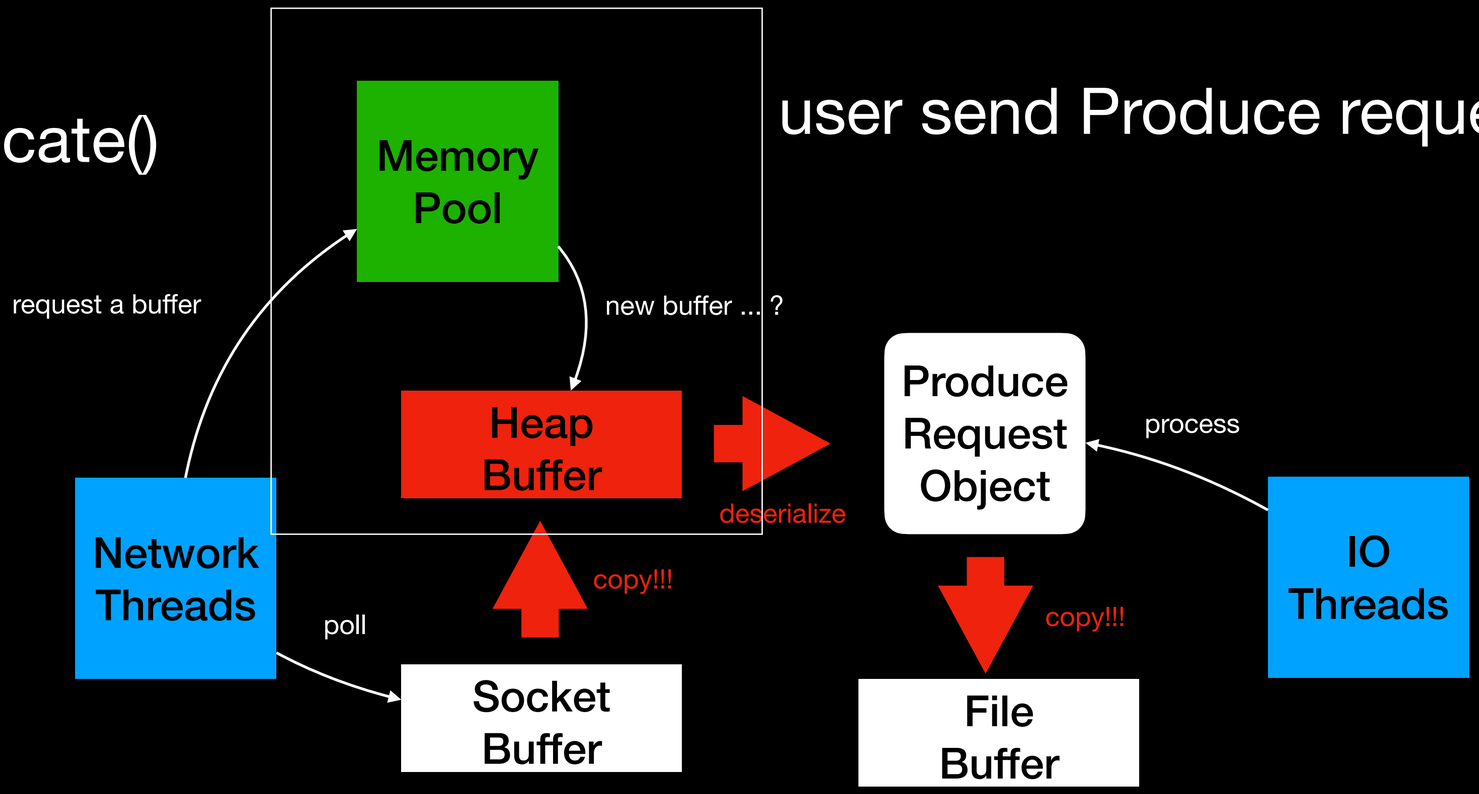
File size	Normal file transfer (ms)	transferTo (ms)
7MB	156	45
21MB	337	128
63MB	843	387
98MB	1320	617
200MB	2124	1150
350MB	3631	1762
700MB	13498	4422
1GB	18399	8537

https://developer.ibm.com/articles/j-zerocopy/


Network Threads
1) zero-copy
IO Threads
2) stupid-copy
Socket Buffer
Heap Buffer
File Buffer

Zero-Copy 的殘酷真相

HeapBuffer.allocate()



user send Produce request to server

所以到底哪裡是 zero-copy ???

誇飾 [編輯]

56 種語言

條目 討論 臺灣正體 閱讀 編輯 檢視歷史 工具

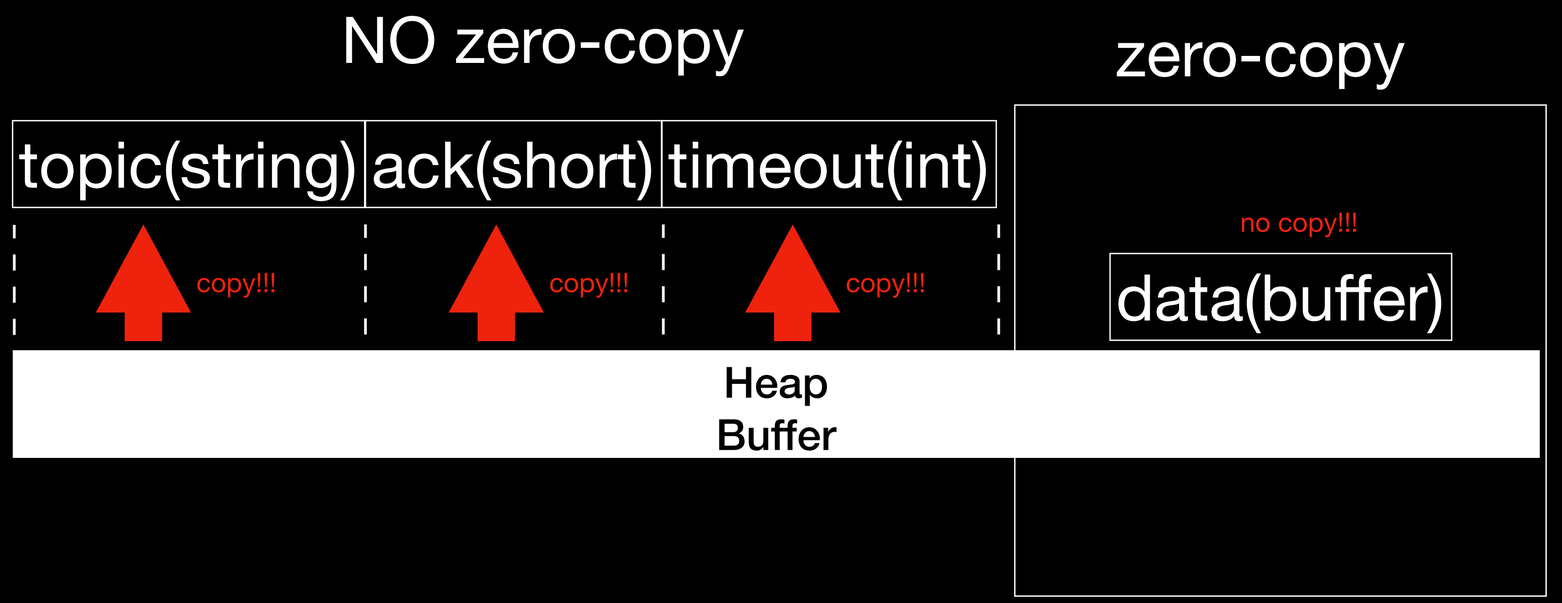
維基百科，自由的百科全書

誇飾，又稱**誇張**（英語：Hyperbole），是一種修辭方法。語文中，將客觀之人、事或物的特點，透過主觀情意，故意用誇大鋪張地渲染與鋪飾描述的手法，使它與真正的事實相差很遠，以加深讀者的印象上[1]。

使用誇飾修辭必須注意主觀方面是出自於作者的情意之自然流露，還有客觀方面不致於會被誤為是事實。誇飾修辭可以使句子或文章呈現言過其實、一鳴驚人的效果。如果運用得當，不但可以使再也普通不過的句子，變為新奇鮮明，同時也能夠聳動讀者的情感，加強印象，彰顯作者所要表達的情意，藉以打動讀者的心坎，領略作者的真意。[2]

zero-copy = less-copy

```
public class ProduceRequestData implements ApiMessage {
    String transactionalId;  26 usages
    short acks;  13 usages
    int timeoutMs;  13 usages
    TopicProduceDataCollection topicData;  23 usages
```



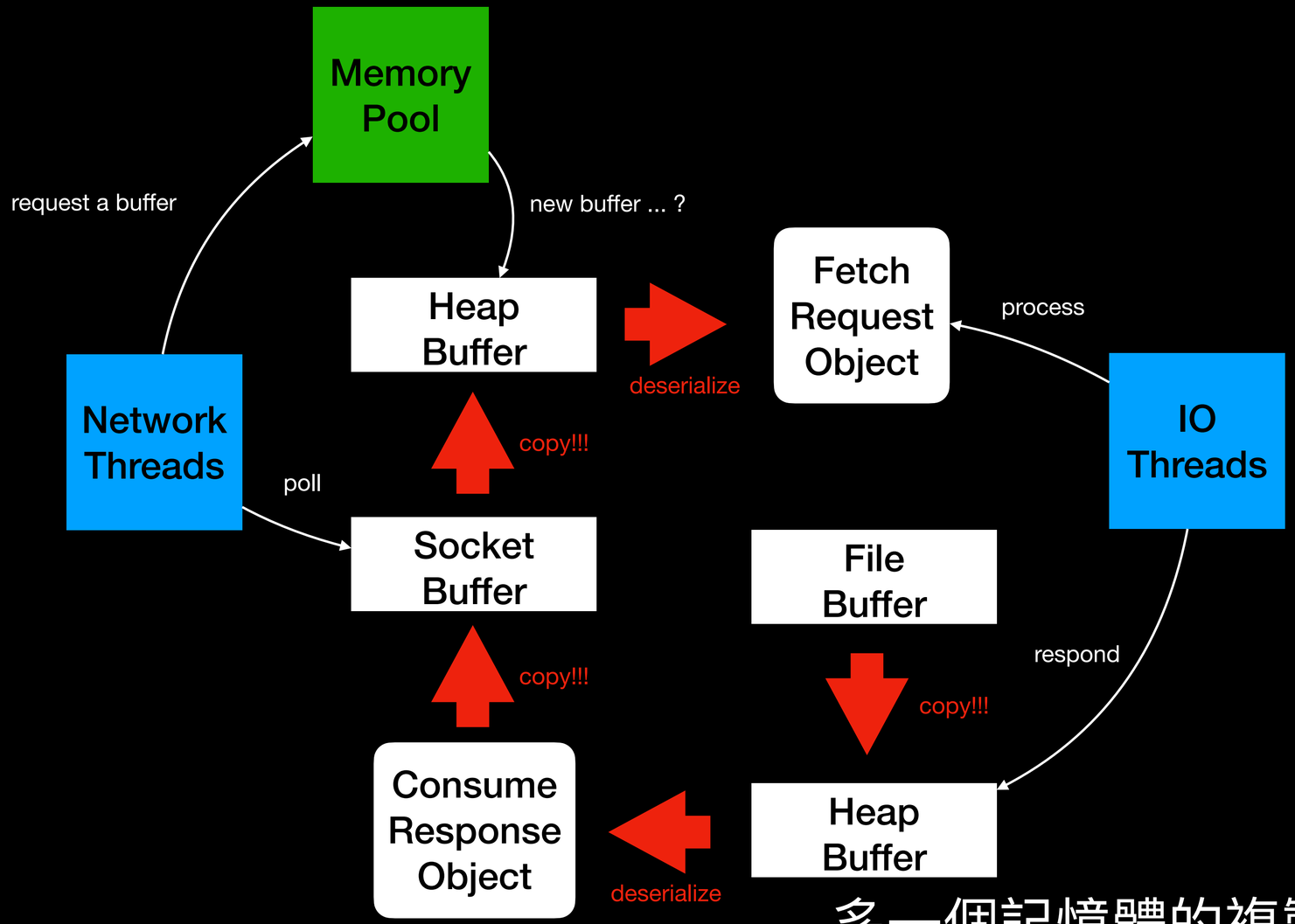
buf = data from network

```java
@Override  11 usages  Nikolay
public ByteBuffer readByteBuffer(int length) {
    ByteBuffer res = buf.slice();
    res.limit(length);

    buf.position( newPosition: buf.position() + length);

    return res;
}
```

less-copy

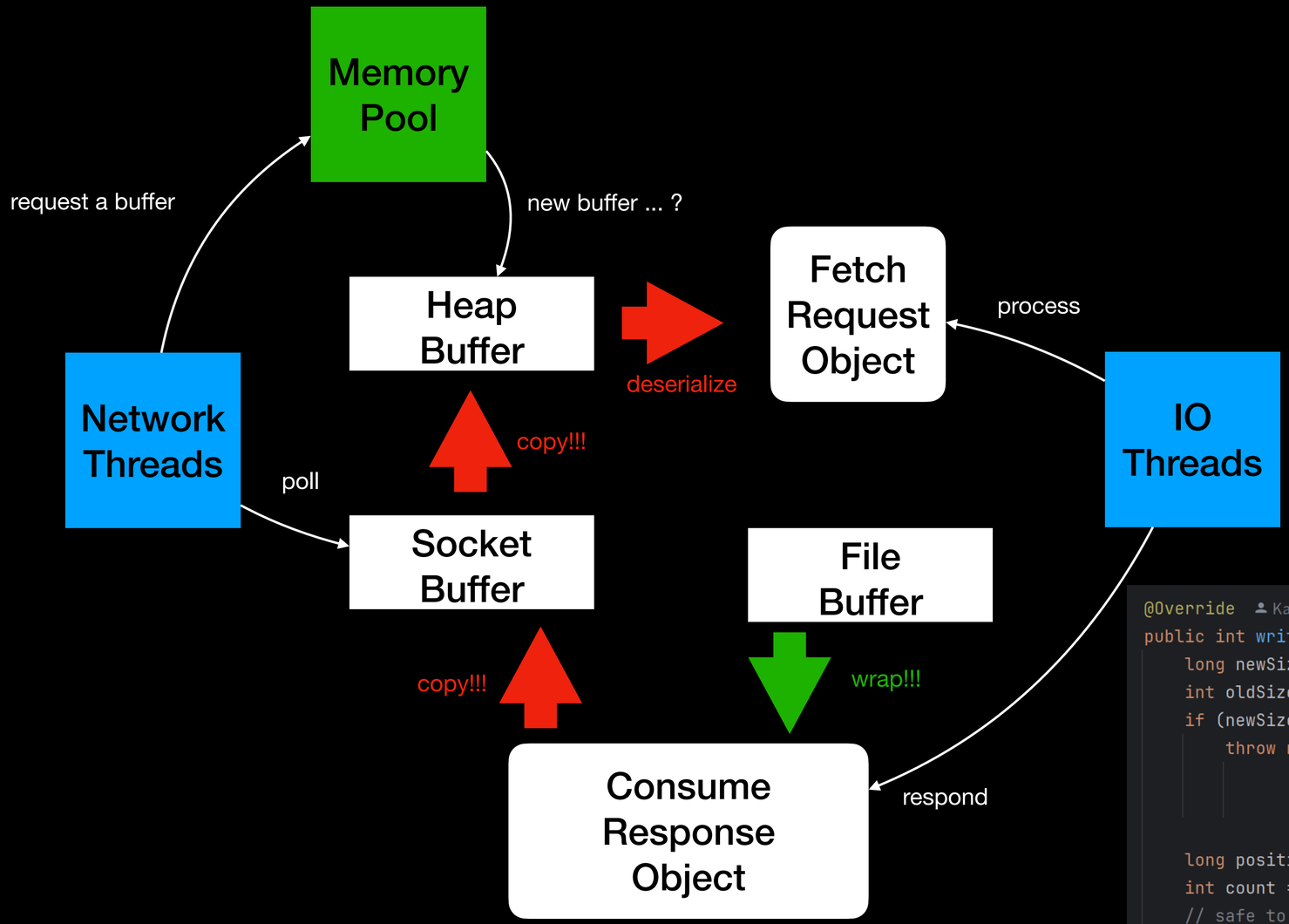
https://zh.wikipedia.org/zh-tw/%E5%BE%A9%E4%BB%87%E8%80%85%E8%81%AF%E7%9B%9F2%EF%BC%9A%E5%A5%A7%E5%89%B5%E7%B4%80%E5%85%83

Zero-Copy 的殘酷真相



多一個記憶體的複製

Zero-Copy 的殘酷真相



Socket Buffer

```
@Override  Kamil Szymanski +2
public int writeTo(TransferableChannel destChannel, int offset, int length) throws IOException {
    long newSize = Math.min(channel.size(), end) - start;
    int oldSize = sizeInBytes();
    if (newSize < oldSize)
        throw new KafkaException(String.format(
                "Size of FileRecords %s has been truncated during write: old size %d, new size %d",
                file.getAbsolutePath(), oldSize, newSize));

    long position = start + offset;
    int count = Math.min(length, oldSize - offset);
    // safe to cast to int since `count` is an int
    return (int) destChannel.transferFrom(channel, position, count);
}
```

File Buffer

Request has no Zero-Copy
Response has Less-Copy

接下來，聊聊 committer?

為何你需要參與開源?

源來適你

參與開發的諸多好處

免去繁瑣的求職考試

透過參與開源專案，直接展現您的實力，開啟職涯新篇章。

在真實專案中提升技能

您貢獻的程式碼將被廣泛應用，在實際使用中不斷精進您的開發能力。

擁有技術決策權

參與社群討論，引導技術發展方向，打造更符合您需求的軟體。

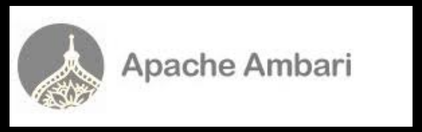
為台灣軟體實力正名

透過參與國際開源專案，讓世界看見台灣的軟體開發實力。

才怪
你只是想要一個很屌的頭銜而已


YUNIKORN


Apache Ozone™


Apache Ambari


APACHE POLARIS™


APACHE
GRAVITINO

你想要取得熱門技術的頭銜?

要跟一群大公司的資深員工競爭
你的臉沒存在感沒人會理你
你很難累積有價值的貢獻
要投入非常多的時間
日夜作息顛倒
沒有標準

只是當打工仔而已...去刷題吧

只是想練功？我不信

I also reviewed many of his pull requests and I was never really satisfied by them. I found that he often missed fundamental things while working on the new consumer.

I find him a little sloppy in PR reviews. There is frequent occasions, in which he approves a PR, and I review it afterward, and still leave a lot of comments, and we need one or two more rounds with the contributor before we can actually merge the PR.

熱門技術的頭銜很不容易

最好有大神幫你背書

Working with him on many of his PRs I sometimes feel like he did not fully think through the changes he's making, or he missed a small part of the task.

I do not believe xxx meets the technical bar for commitership. His contributions related to Kafka Connect have been seriously underwhelming. I'd be afraid for the quality of our code base if he was allowed to merge changes.

I think that he does not meet the bar yet from a technical point of view. I don’t trust him to merge code at this point given the quality of his reviews.

開源擴展職涯都是特殊案例

用相同的毅力去做其他事情你也會成功

你還是很想做開源貢獻?

今天就開始選一個專案開始貢獻