Report:

1. The time required to copy the file using *read_write* varies with the size of the buffer specified because of overhead and number of times to copy. If the buffer size is larger than the data, then all the data will be copied once which is the most efficient approach. However if the buffer size is smaller, it will fill up the buffer up to that size until all data is copied; this can take multiple times and will be slower as a result: multiple read() and write() calls add significant overhead and data is copied multiple times. With *memmap*, the data is copied once and the overhead is pretty much the same everytime as a result.

2. The mistake is that when calling write() it will use the user-specified buffer size and truncate extra bytes to the buffer. Because of this, if the buffer exceeds the data size it will add on some truncated bytes to the end of the destination file. To correct this, obtain the source file's size and use that as the buffer size to pass to write().