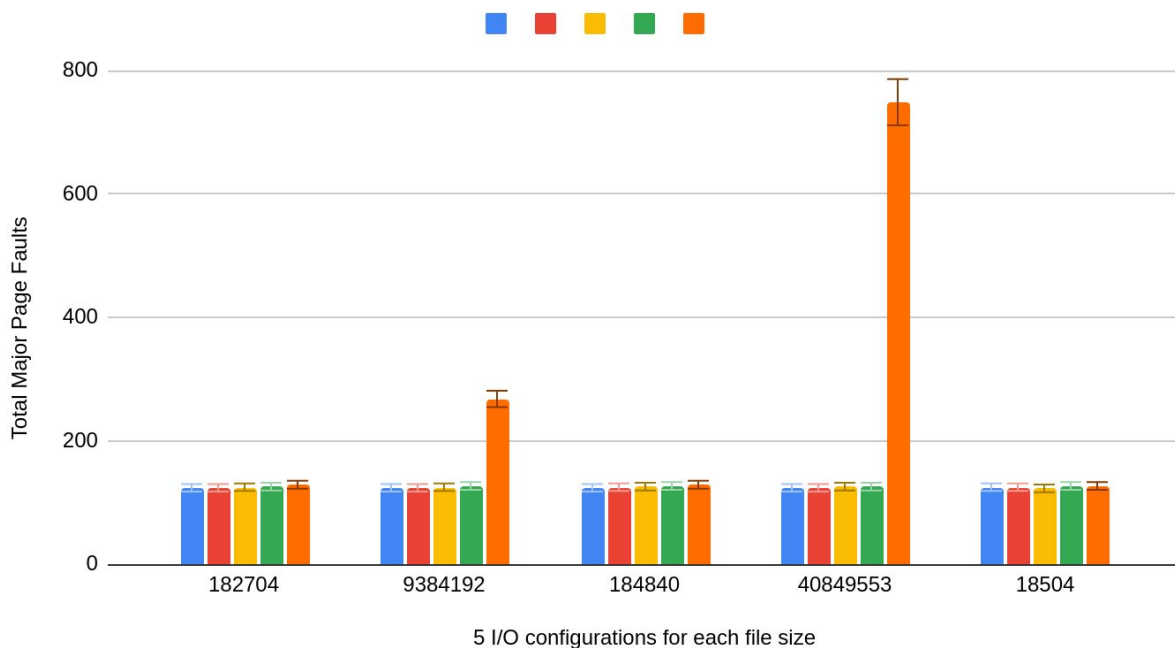


Performance Analysis

Comparing the number of total major page faults to the size of the file and the I/O configuration used to read the file.

Several files were read for testing the number of page faults for each file size. Overall, regardless of the size of the file the number of page faults is relatively constant when using the `read()` I/O. even with the larger buffer there was usually only an increase of 1 or 2 page faults overall. The biggest difference is with the `mmap` I/O on large file sizes. With large file sizes like sets 2 and 4, the number of major page faults caused by the disk I/O is much larger than when we used `read()`. This is likely due to the fact that when using `mmap` we use the whole file, rather than counting chunk by chunk.

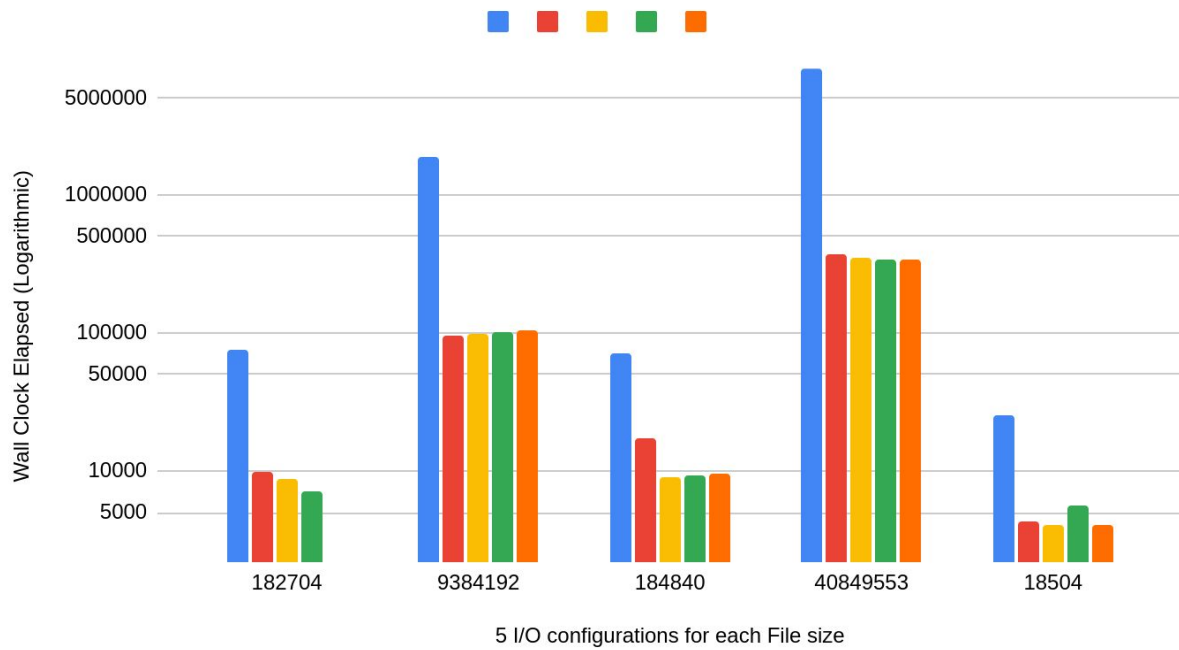
Major Page Faults vs File Size & I/O configuration



Comparing the number of total elapsed wall clock time across several different I/O operations and several different files.

Those same files that were previously analysed to compare major page faults were also analysed in a similar fashion. An important note is that the Y axis scale on the next graph is actually logarithmic to help visualize the data. In this data we find that using a smaller chunk size the time to read the whole file is much longer. With all the sets of data for each file it is clear to see that the first bar is significantly larger than the others. This data shows that using `mmap` also does not necessarily give us a huge boost in speed, where its performance pretty much matches using `read()` with chunk sizes of 4k and 8k. Only in smaller files does `mmap` make a performance increase, as the files get larger the difference between `read()` and `mmap()` lessens.

Wal Clock Elapsed vs File Size & I/O Configurations



Results

There is really no best practice that we can identify in this case. Using mmap or read() should depend on the system you are designing it for. If you are time constrained I would recommend read() with a larger page size or mmap. If you need to minimize the number of major page faults I would recommend not using mmap.