CS 3013 Project 4

Benjamin Hetherington

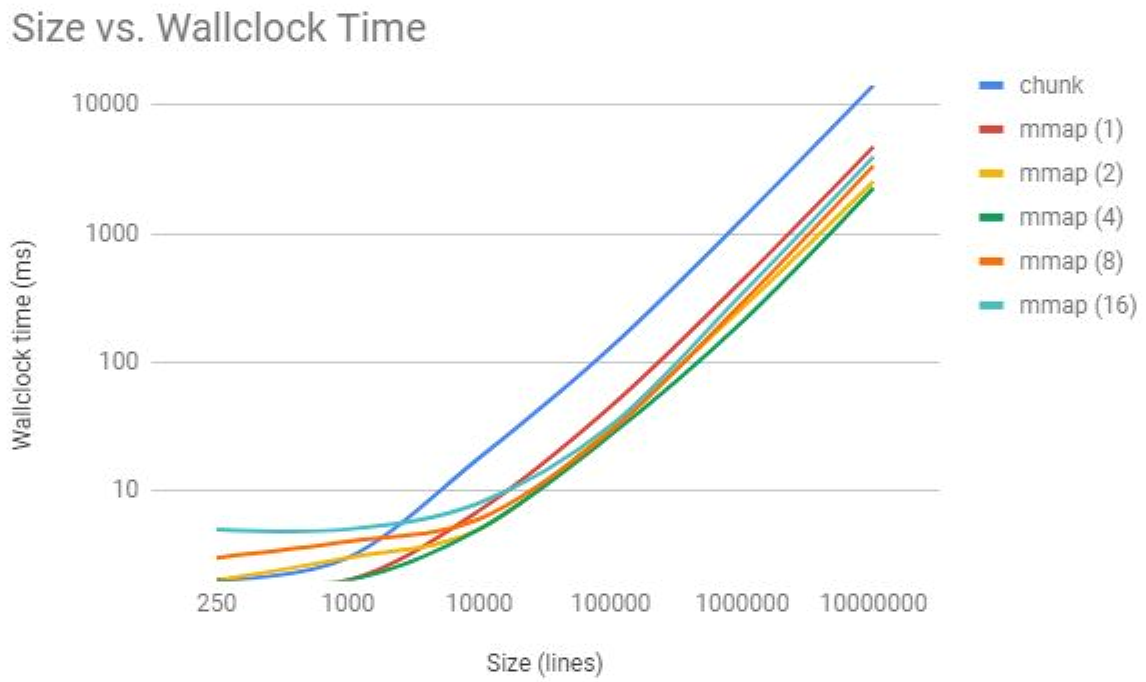
Configuration

The experiments were performed on a virtual machine running Linux with a 4 core CPU.

Wallclock Time Analysis

A search for the string “aba” was performed on files of various sizes, ranging from 250 lines up to 10,000,000 lines, where each line was 25 characters. At lower line counts, running with fewer threads and chunking appeared to be approximately equal. However, as the number of lines scaled up, the amount of time the chunked approach took with the default chunk size of 1024 bytes grew more rapidly than any of the memory-mapped approaches.

With regards to the memory-mapped approaches, the number of threads that performed consistently best was 4. Given that the CPU of the virtual machine that these tests were performed on has 4 cores, I do not believe this is a coincidence.



Major Page Fault Analysis

My virtual machine did not, unfortunately, report major page faults.

Conclusion

Based on these tests, it would appear that the overall best approach would be to search using the multithreaded memory-mapped approach. With that said, more threads beyond the number of cores the CPU has does not appear to provide any speed increase, rather, it appears to slow the process down, likely due to the fact that managing more threads requires overhead, and a CPU with only 4 cores can not get the full performance benefits of using more than 4 threads at once. However, this is based purely off of the wallclock time data, as I was unable to gather any for the number of major page faults, so I do not know how each implementation impacts the number of faults generated, or the patterns that arise from them.