

1. The threaded version is dramatically faster than the nonthreaded version, though it is not 4 times faster. This is because the code is not 100 percent parallelizable, in addition to the fact that in my implementation, all threads must complete before thread reassignment. The performance of the multithreaded version is still dependent on the directory searched, though it can hide the time it takes for some larger directories by running other threads in the background, sort of like servicing a cache hit under a miss. This implementation could be improved if all threads did not have to complete before thread reassignment. That way, very large directories would have even less of an impact of the performance.

2. The multi threaded version performs worse than the original in cases where the thread management overhead outweighs the benefit from distributing the workload. Larger workloads, deeper and wider directories, would benefit more from the multithreading, and could potentially outperform the nonthreaded version.

3. Running the program the second time likely improves the runtime because the data associated with the program is all in high level cache, and all the memory translation pages are probably cached as well.