Coarse-Grain Results Reflection



From the plot that was generated from my benchmark results, it appears as if the program slowed down as each of the 16 jobs were completed. I didn’t expect the multi-threaded version of this program to run so slowly. A few things come to mind as to why this may be:

1. Perhaps the file sizes/contents were different enough and that they caused the slowdown and variation in completion times.
2. There could have been an issue in the way I implemented the lock\_guard() function in my countWords function that each thread was calling.
3. I was locking in the wrong place, or could have been locking too much of the hash table which caused the other threads to have to wait for their chance to increment the count variable.

I suppose it makes sense that even if the program can run in parallel, the different threads will still have to wait for their turn to access the shared count variable, which would lead to a slowdown rather than speedup.