Assignment 3: Static loop scheduler

Question 1: Why do you think some measurements are so erratic?

It can be observed that as the n increases, the speedup in case of thread method increases. This could be because the number of iterations for each thread increases and every thread needs a mutex only to update the result. But in the iteration method, each thread can update the result anytime after it finishes its iteration, however, other threads will have to wait until it releases its mutex after it completes the iteration and hence it has a low speedup. As the proportion of parallel to serial execution increases, greater will be the speedup as the number of processors increases.

Question 2: Why is the speedup of low intensity runs with iteration-level synchronization the way it is?

According to the plot, when the intensity is low and n increases, speedup remains constant no matter how much the number of threads increase. There is one fixed time overhead, which is getting the mutex and running the function. When there is larger dataset and as the n increases along with the function cost, the speedup will also increase. Since the intensity is low, the function cost is less and hence the speedup is also less. So even if the n increases, the function cost is still less and so is the speedup.

Question 3: Compare the speedup of iteration-level synchronization to thread-level synchronization. Why is it that way?

The speedup of thread-level synchronization increases as n increases. But that’s not the case in iteration-level synchronization. As the parallel to serial execution increases, the speedup will also grow as the number of processors increase. Greater the parallelism, greater will be the speedup. In thread-level synchronization, each thread needs a mutex to update the result after it completes its iteration. In iteration-level synchronization, each thread waits for the other thread to release its mutex after it finishes its iteration which will take a while longer to compute and hence it has comparatively less speedup.