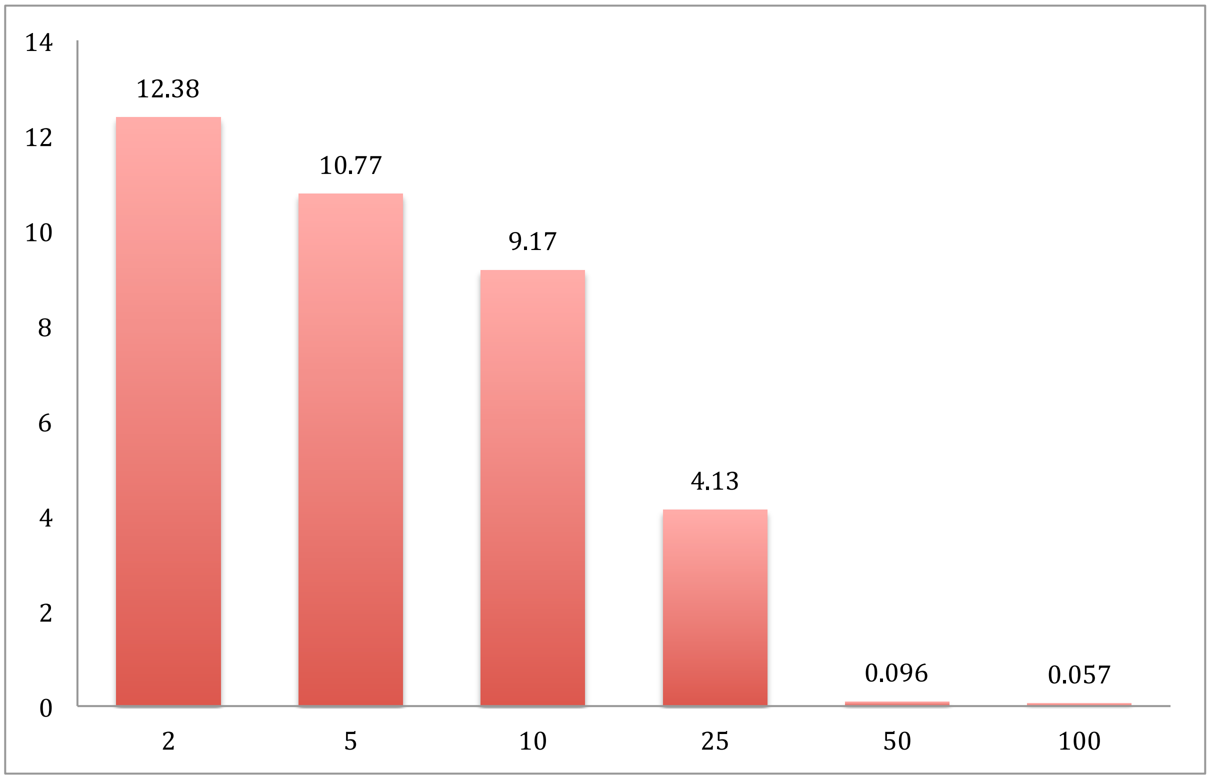
Jessica Donahue
Parallel Computing
4/6/16

Lab 2 Report

N = 1000

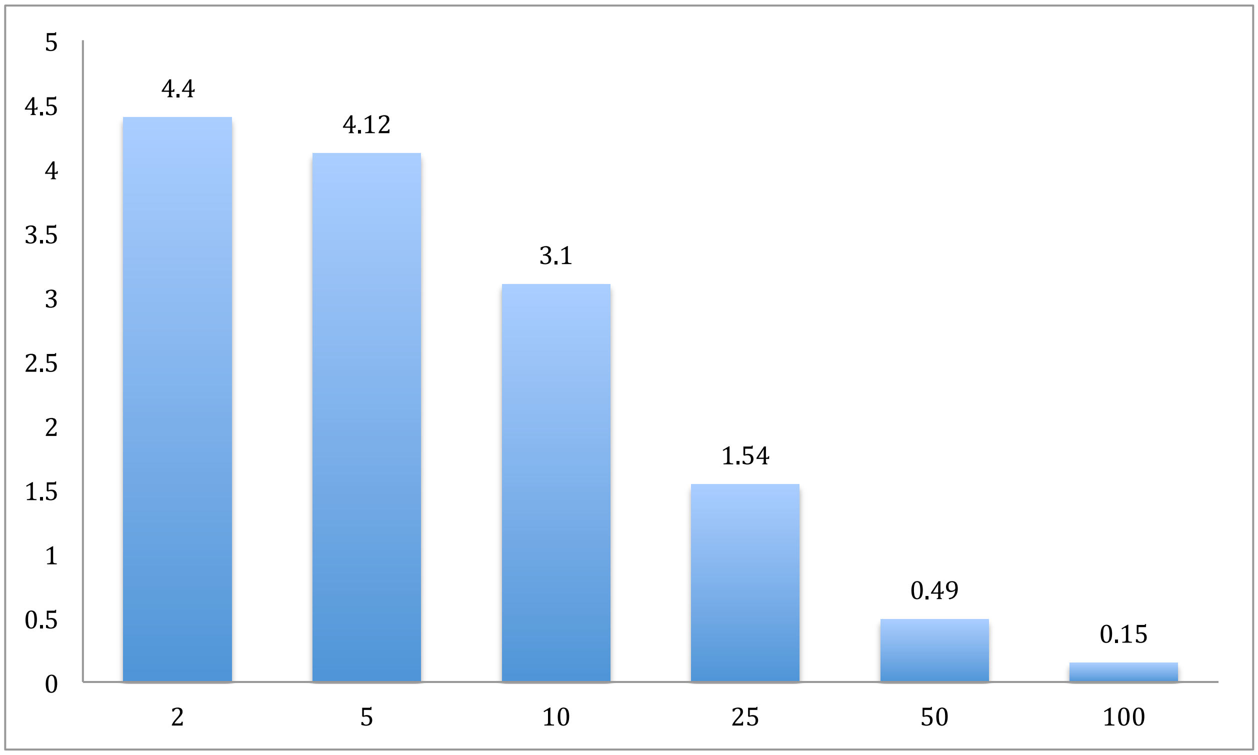


Conclusions:

The above graph depicts the speedups relative to one thread with an increasing number of threads. It is clear from the graph that 2, 5, 10, and 25 threads all give positive speedups; however, 50 and 100 threads cause an increase in performance time and no speedup. This is caused by thread overhead. Thread creation and termination are expensive operations. Therefore, the performance time of the program increases as we increase the number of threads if we are not achieving efficient parallelism to counteract this overhead. In the *genPrimes* function, I have a nested for loop that marks the numbers in the array that are prime. Only one thread can update a location in the array at a time, thus I use *#pragma omp critical* in this section. This permits only the master thread to run this code; therefore, all the threads other than the master thread created in this loop cannot be used. Therefore, the time used for thread creation is wasted because this section of the program cannot execute in parallelism and thus, increases the thread overhead.

This is why when there are more threads generated in my program, there is less parallelism (due to threads being idle) and greater thread overhead. This explains why the speedup decreases as we increase the number of threads, and eventually becomes negative for 50 and 100 threads.

N = 10000



Conclusions:

The above graph shows a similar pattern as the previous graph, with no speedups for 50 and 100 threads and positive speedups for 2, 5, 10, and 25 threads that decrease with increasing thread count. However, the speedups are not as large for when N is 10,000 than when previously 1,000. The larger data size causes an increase of performance time and an increase of thread overhead. Since the thread amount is the same for when N is 1,000 and for when it is 10,000, there is more work distributed to the threads when N = 10,000 because there is more work to be divided up among the threads. Thus, when these threads become idle during the critical section, there is more work that isn't being executed. This is why there is less of a speedup in this graph than in the previous. Also, the performance time greatly increases when there is 50 or 100 threads because the cost of creating this amount of threads outweighs the benefits of the parallelism.