CS 111 Lab 4: Graphs

Cristen Anderson, UID: 904400797

March 11, 2016

Introduction

These graphs were created to demonstrate trends between number of iterations, number of threads, and average time per operations for two different tester programs. For each data point, 5 different trial values were averaged. All times are in nanoseconds. Graphs were created using Microsoft Excel.

Part 1

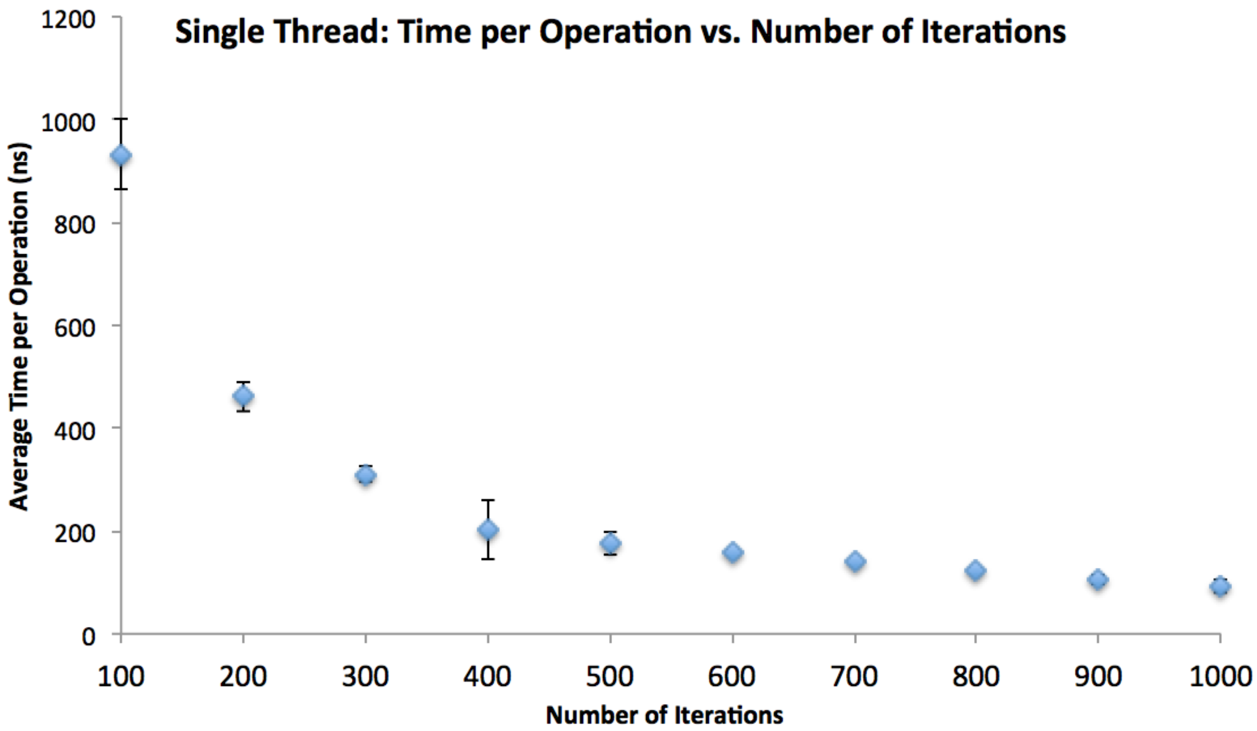

Figure 1: This plot shows my data for time per operation vs. number of iterations for a single thread. As I increased the number of iterations, the average time per operations went down. The reason behind this is described in answers.txt.

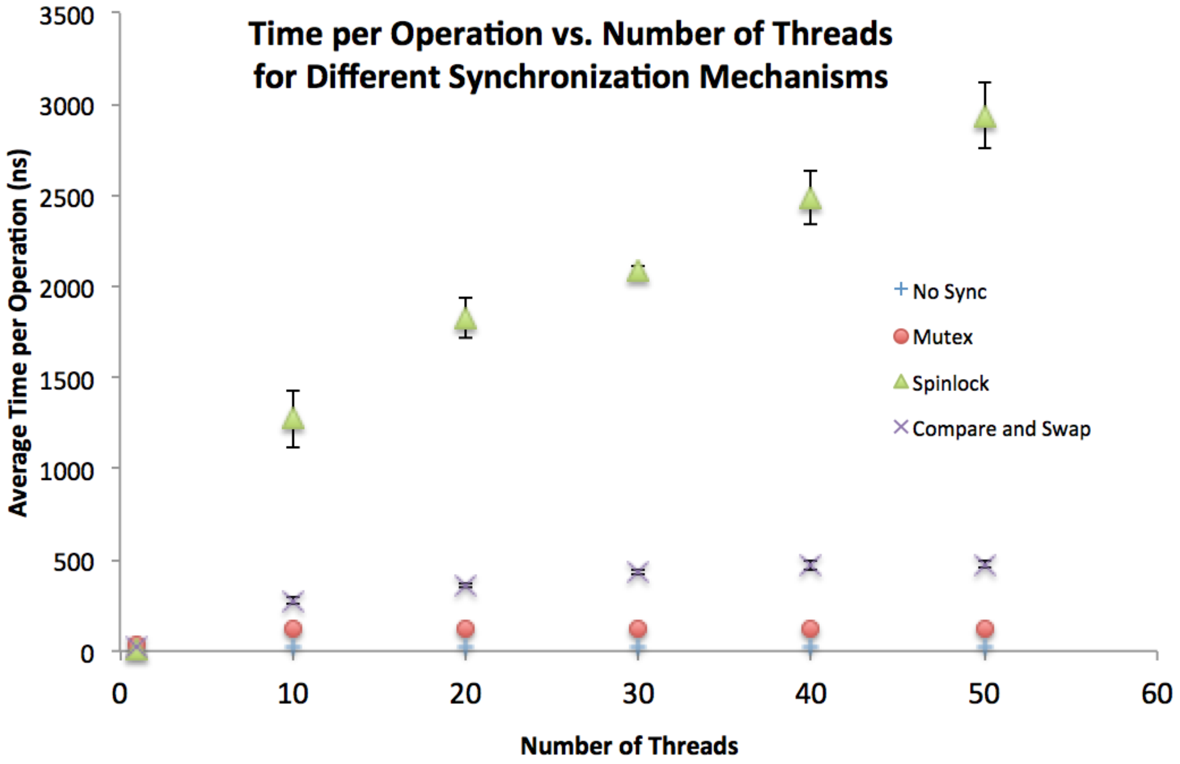

Figure 2: This plot shows average time per operation for 3 different methods of synchronization, compared to no synchronization, for different numbers of threads. I used 100000 iterations for each of these data points, in order to decrease the effect of the thread overhead on the average time per operation. From this data, spinlock clearly has the highest cost increase when more threads are added, while other synchronization methods only increase slightly in cost with more threads.

Part 2

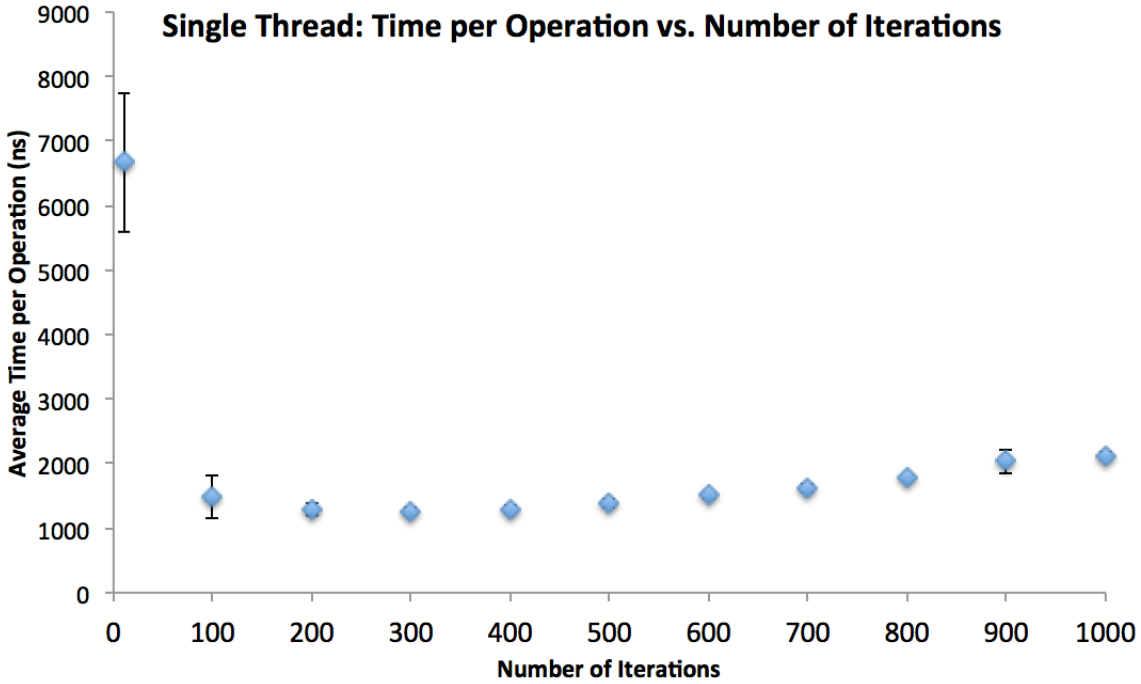

Figure 3: This plot shows my data for time per operation vs. number of iterations for a single thread. As I increased the number of iterations from 1 to 100, the average time per operations went down. However, it went up again from 100 to 20000 iterations. The reason behind this is described in answers.txt.

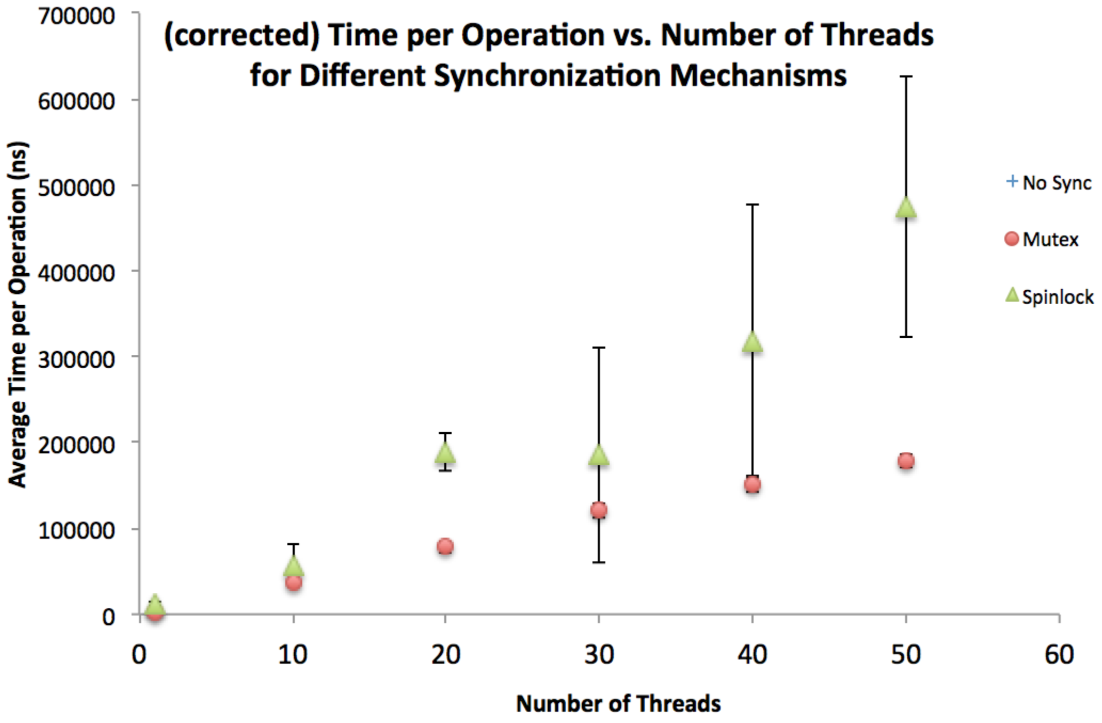

Figure 4: This plot shows my data for time per operation vs. number of threads for 1000 iterations. It was corrected by subtracting the thread creation overhead time from the total time. The time per operation grows with number of threads because they must wait for locks.

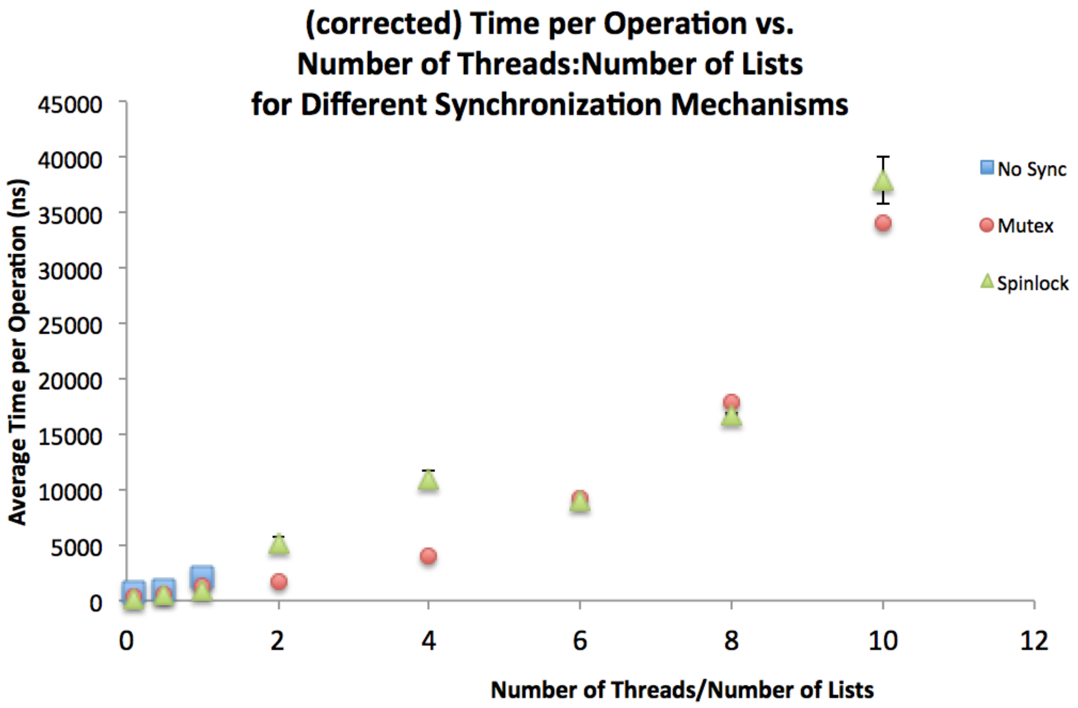

Figure 5: This plot shows my data for time per operation vs. number of threads:number of lists for 1000 iterations. An analysis of these trends is in answers.txt.