

Homework 5

Due Tuesday, February 5, 2019

This homework has four parts. In each you can change the code I provide to work better with what you write, including fixing possible bugs. Your code should be written to be thread safe. Use the Scholar queue to do your timing runs.

Part 1:

Using the code in `hw5Loop.c`, write a main procedure that has a parallel for loop that calls `doWork` on each element of `wA`. Run using static scheduling with the default block size, static scheduling with a block size of 50, dynamic scheduling with a default blocksize, dynamic scheduling with a blocksize of 50, and guided scheduling. For each time how long it takes the loop to run. What can you conclude about the different scheduling strategies for this loop?

Part 2:

For the scheduling strategy that works best for Part 1, quadruple the number of threads that execute the loop. Does the loop run faster or slower?

Part 3:

Using the code in `hw5Section.c`, write a parallel section code with `T` sections, where `T` is the number of threads supported by your hardware. Each section should have a loop that is pulling work from the `wA` array. The work in each element of `wA` should be performed once, and only once. Compare the time with the sequential time for the loop.

Part 4:

Using the code in `hw5Q.c`, write a main routine that initializes and adds work to the work queue, and then add a loop within a parallel construct that pulls work from the work queue. Compare the execution time with the sequential execution time.