Concurrency and Parallelism

GEI 2023

Lab 1 – Integer Array

Write a program in which several threads share an integer array that is initially filled with 0s. Each thread should iterate a given number of times, increasing a random position in the array by 1. If the program is working correctly, the sum of the values in the array should be the number of iterations times the number of threads.

You can download a the sequential version of this program from github, that runs from the command line, and accepts the following options:

- `-i n`, sets the number of iterations.
- `-t n`, sets the number of threads.
- `-s n`, sets the size of the array.
- `-d n`, sets a delay between operations in the critical section.

Add the following features:

Part 1 (**Create the threads and protect the array**)

Create the number of threads specified in the options, and protect the array using a mutex. Do not use global variables to share the array or the mutex.

Each thread should be identified by a number from 0 to the number of threads - 1, which should be used each time the threads print a message.

Part 2 (**One Mutex per Position**)

Each position in the array can be incremented independently. Modify your program so that each position is protected by a different mutex.

Part 3 (**Add threads that move values between positions**)

Add another set of threads (with the same size as the original one) that randomly select two positions in the array, and decrement the first one, and increment the second one. These threads should also iterate the number of times specified in the options.

When a movement is performed, it should not be visible to other threads. That is, if the sum of the array before the movement is n, no other thread should see a diferent sum unless a third thread performs an increase.

Part 4 (**Iterations**)

Change the behaviour of the iterations so that it specifies the number of iterations for each set of threads instead of each individual thread. That is, if the number of iterations is set to 100, the total number of increases and movements should be 100 each.

If the program is working correctly, the sum of the array at the end should be the number of iterations.

Count the iterations using a shared counter.

Submission

The submission deadline is February 19. The sequential program is available at github classrom at https://classroom.github.com/a/oxRzrn1i. When you first access the assignment you will be able to create your lab group. Name your group using the logins of the members separated by a hyphen. Once the first member of the group has created the repository, you should be able to add the second member.