Parallel programming

Business Modeling and Distributed Computing, 2020-2021

Assignment 2 - Pthreads

Let's consider a 2 dimensional matrix U stored in raw-major format. Raw major format means that a matrix of m lines x n columns is stored in a contiguous space of memory (i.e. an array), such that after element U[i, n-1] it follows the element U[i+1, 0].

In each iteration k, the algorithm needs to compute novel values U[i, j], $0<i<m-1$, $0<j<n-1$ with the following formula:

$U[i, j] = 0.25 \times (U[i-1, j] + U[i+1, j] + U[i, j-1] + U[i, j+1])$

The algorithm starts from the second line, second column (element U[1, 1]) and goes to right computing the element U[1,2], up to the element U[1, n-2], after that it moves to the third column (element U[2, 1]) and so on. It implied that when computing the element U[i, j], elements on the left and top should already be available.

Please write a POSIX Pthreads program that takes as input:

- Number of threads
- The number of iterations k
- Values m and n (which should be big enough)
- The initial matrix U. At the start, the matrix U contains values only on the first and the last rows and columns, the remaining elements being zero

You could read the values m and n from the parameters of the program and generate randomly the initial matrix. You could perform the parallelization on the columns of the matrix, by assigning a successive number of columns to each thread.

Your POSIX Pthreads program should compute the matrix U after running k iterations of the algorithm.

Please perform a performance analysis, by varying the number of threads and the problem size, and comment the results, indicating what sort of scalability you encountered. You should present an average over multiple runs, to account for measuring errors or outside effects. Time measurement should exclude threads creation.

When submitting the results, please archive the following items:

- The source code
- A document explaining your implementation and the performance analysis