Assignment 1
SPM course a.a. 23/24

March 21, 2024

Wavefront computation

Suppose you want to parallelize an algorithm that exhibits an upper-triangular wavefront computation pattern on an NxN matrix. Each element of an upper diagonal (starting from the leading diagonal) can be computed independently. Instead, distinct upper diagonals have to be computed serially in order (first, all elements of the diagonal k, and once the computation has completed, all elements of the diagonal k+1). See **Figure 1**.

We are not interested in how the real computation on the elements of the diagonal is actually done (usually through a stencil referring to previously computed elements); for this reason, consider the toy sequential C++ code (named *UTWavefront.cpp*) in which the computation is just a "waste of time" for a given amount of microseconds. In other words, the computational cost associated with a single diagonal element is artificial and can be varied by changing the *min* and *max* parameters of the provided sequential code.

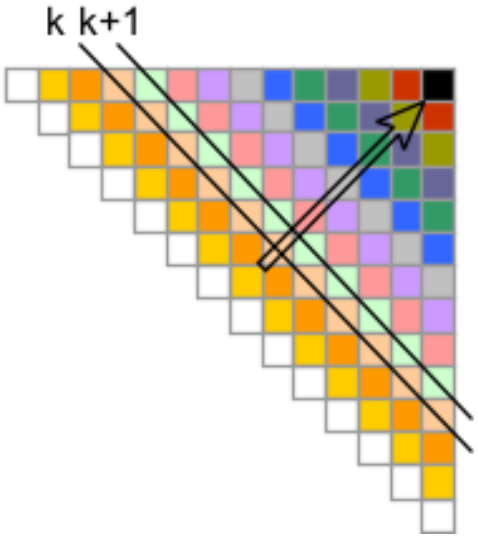

Figure 1. Representation of the upper-triangular wavefront computation pattern. Elements of the diagonal *k+1* can be computed only after all previous diagonal (from 0 to *k*) elements have been computed. Elements laying on the same diagonal can be computed independently.

Implement a parallel version of the UTWavefront.cpp code using C++-thread. Evaluate the weak and strong scalability by also varying the task computation granularity (i.e., min and max parameters) on the machines of the course (one or both -- spmcluster.unipi.it and/or spmnuma.unipi.it). NOTE: To synchronize all threads at the end of the computation of a diagonal, use the C++20 *std::barrier* class.

Once you have the parallel code, **email it to the teacher along with a brief document** (PDF format, max 4 pages) describing how to compile and execute the code, the performance figures obtained, comments, problems faced, solutions adopted, etc.

Deadline: Two weeks. However, no later than the end of the lectures (end of May 2024)