Introduction to multiprocessor architecture:

Report Assignment 2

# Part 1:

# Part 2:

1. Algorithm explanations
   1. Version 1: Simple process parallelization [algorithm.c]

* The first optimization level is to consider the 2-dimensional array as an index array, and this is already done with the provided clause
* The next optimization is straight forward parallelizing the main work in the nested for structure. To achieve this, we used the collapse(n) directive which allow to collapse the two-level nested for into individual operations distributed on the threads.
  1. Version 2: Parallelize with sharing optimization [algorithm2.c]

The next step is to consider the cache access and to make parallelize row by row so that we avoid accessing to many times to data that is not in the cache anymore. To achieve this, we separate the process in two helper functions updateRow and copyRow. The first one takes care of updating the values on the grid and the second one copies the values from the input to the output. We create first the threads with #pragma opm parallel and then we parallelize the two loops with #pragam omp for since we have two loops its more performant to separate the omp for declaration instead of using the #pragma omp parallel for directive on the outer loop.

* 1. Version 3: With cache blocs size optimization [algorithm3.c] [number for the if to define]

Use of an if statement to avoid parallelizing on to small array where the overhead brought by the merging of the threads and distribution of the work would be too important compared to the advantage of parallelization.

1. Optimization analysis
2. Presentation of the results