Programming Parallel Computers

Aalto 2023

CP2a: instruction-level parallelism ★

Please note that you can still submit, but as the course is already closed, your submissions will not be graded.

To get started with the development, **download the code templates**, unzip the file, edit cp.cc, and run `./grading test` or `./grading benchmark` to try it out — see the **instructions** for more details!

Upload your solution as a file here…

Please upload here the file **cp.cc** that contains your solution to task CP2a.

Choose File No file chosen

… or copy-paste your code here

Submit

Your submissions

Your submissions to CP2a will appear here; you can simply **reload** this page to see the latest updates.

What you will need to do in this task

Please read the **general instructions for this exercise** first. Here are the additional instructions specific to this task:

> Parallelize your solution to CP1 by exploiting **instruction-level parallelism**. Make sure that the performance-critical operations are pipelined efficiently. Do not use any other form of parallelism yet in this exercise. Please do all arithmetic with **double-precision** floating point numbers.
>
> For this technical exercise, we have disabled auto-vectorization.

What I will try to do with your code

I will first run all kinds of tests to see that your code works correctly. You can try it out locally by running `./grading test`, but please note that your code has to compile and work correctly not only on your own computer but also on our machines.

If all is fine, I will run the benchmarks. You can try it out on your own computer by running `./grading benchmark`, but of course the precise running time on your own computer might be different from the performance on our grading hardware.

Benchmarks

Name	Operations	Parameters
benchmarks/1	1,004,000,000	nx = 1000, ny = 1000
the input contains 1000 × 1000 pixels, and the output should contain 1000 × 1000 pixels		
benchmarks/2	16,016,000,000	nx = 1000, ny = 4000
the input contains 4000 × 1000 pixels, and the output should contain 4000 × 4000 pixels		

Here “operations” is our rough estimate of how many useful arithmetic operations you will **at least** need to perform in this benchmark, but of course this will depend on exactly what kind of an algorithm you are using.

Grading

In this task your submission will be graded using **benchmarks/2**: the input contains 4000 × 1000 pixels, and the output should contain 4000 × 4000 pixels.

The point thresholds are as follows. If you submit your solution no later than on **Sunday, 07 May 2023, at 23:59:59 (Helsinki)**, your score will be:

Running time	Points
≤ 7.000 sec	1
≤ 6.000 sec	2
≤ 5.000 sec	3

If you submit your solution after the deadline, but before the course ends on **Sunday, 04 June 2023, at 23:59:59 (Helsinki)**, your score will be:

Running time	Points
≤ 6.000 sec	1
≤ 5.000 sec	2