

Name: Do Le Tuan Minh

Student Number: 50366862

Email: le.do@tuni.fi

COMP.CE.350 Multicore and GPU Programming Lab Work Autumn 2021

Part 2

The performance of the code with different optimization, in average:

Optimization	Total frametime (ms)	Satellite moving (ms)	Space coloring (ms)
Original code without optimizations	1639.7	306.7	1259.17
Best optimization of original code with OpenMP	147.5	48.5	82.83
OpenCL, 1x1 workgroup	102.7	7.7	83.7
OpenCL, 4x4 workgroup	24.5	7.3	8.3
OpenCL, 8x4 workgroup	22.75	7.25	5
OpenCL, 8x8 workgroup	20.83/22.67	7	5.3
OpenCL, 16x16 workgroup	19.83/23.16	7.16	4.83/5.5

With 80x80 dimension, the performance of the optimization are:

Optimization	Total frametime (ms)	Satellite moving (ms)	Space coloring (ms)
Original code without optimizations	334	325.83	7.7
Best optimization of original code with OpenMP	54	51.17	2.17
OpenCL, 1x1 workgroup	17.3	6.6	1.3
OpenCL, 4x4 workgroup	16.28	6.5	0.3

OpenCL, 8x4 workgroup	16.67	6.67	1
OpenCL, 8x8 workgroup	15.83/16.71	5.5/6.28	0.57
OpenCL, 16x16 workgroup	16.67	6	0.67

The work group's different sizes have different performance because different work group's sizes contain different work items, which each execute the kernel. Theoretically, the larger the work group size means the better in performance. In practice the best performance are greater or multiply of 8, which explained 8x4, 8x8 and 16x16 work group size had better performance than smaller sizes.

All of the GPU versions ran faster than the CPU counterparts according to the above statistics.

FEEDBACK

- 1) The good of the exercise was learning different types of parallelization method, using different kind of software implementation, tried different optimization and analyze them.
- 2) There were not much to improve for the exercise. In my opinion, I could have written my code cleaner. For the course in the future, providing more and earlier example for OpenCL could be more benefit, since OpenCL is quite difficult and require lots of time to understand and use.
- 3) I have learned different implementation of parallel programming through OpenCL and OpenMP. Observing code runs faster was satisfying.
- 4) The first difficult of the exercise was setting up the environment for running OpenCL. There were a lot of choices and with none-experiences with OpenCL, it was difficult to choose and could be messy to get everything works.

The other difficult of the exercise is learn how OpenCL works. Theoretical part was straight forward but applying it to practice was difficult and easy to get lost.

Finally, working alone made things more complicated and overwhelmed.

- 5) Approximately 2 hours per day for 15-16 days, so in average 30-32 hours in total to complete this exercise.