Report on task 1

Name: Borisova Kseniya
Group: 19137
E-mail: k.borisova@g.nsu.ru

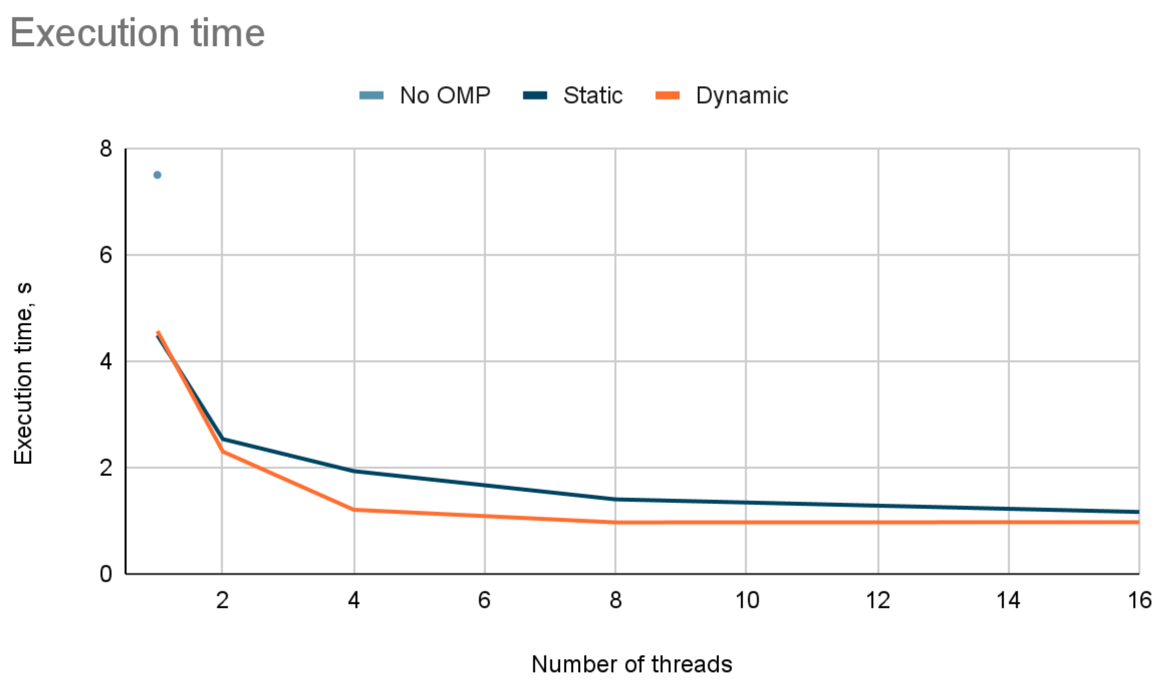
Configuration

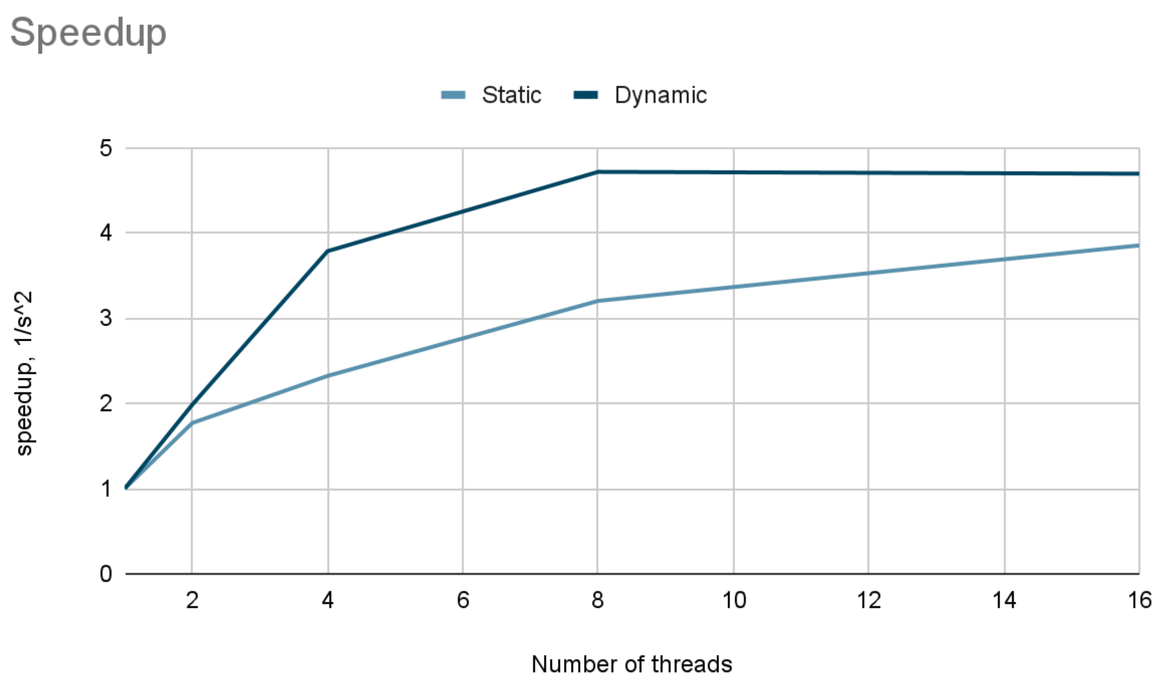
Hardware configuration	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz, ОЗУ 15Gi
Software configuration	Ubuntu 21.04, GNU C++

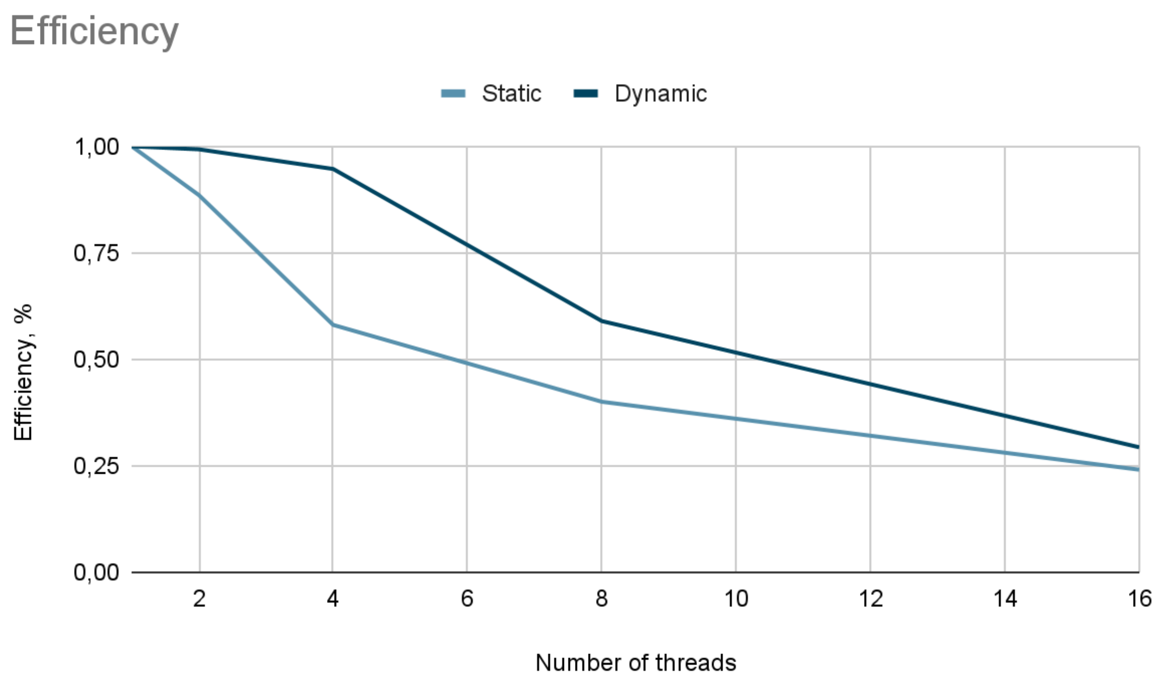
In this lab work, I looked at the effectiveness of the ray tracing program using OpenMP. Calculations were carried out in three variants:

1) Without OpenMP
2) With OpenMP and `schedule(static)` in the main computational loop
3) With OpenMP and `schedule(dynamic)` in the main computational loop

Each of the options with OpenMP was tested on 1, 2, 4, 8, 16 threads. The results on execution time, speedup and efficiency are presented below.







Conclusion:
Using OpenMP gives a speed gain of almost 2 times. After 8 threads, there is an exit to the execution plateau (this is due to the fact that there are 8 cores on the processor). The dynamic schedule without setting up the collapse operator proved to be the most effective. It has less running time, more speedup and greater efficiency on any number of threads listed.

Dynamic scheduling of threads is more efficient due to the fact that each of the pixels requires a different calculation time and a dynamic schedule can provide threads with new iterations as the previous ones are executed. On the contrary, a static schedule defines iterations for threads at the beginning, hence they finish their work at different times depending on the load progress.

My Github repository: https://github.com/kseniabrsv/Supercomputing_course2022