

Programming in CUDA C: the mandelbrot set

Use your exercise from last week as the starting point. If you used the Fortran version for OpenMP, you now have to consider the C version for CUDA C.

Exercise 3:

1. Implement a CUDA kernel for generating the mandelbrot set on the GPU. Setup the GPU \rightarrow CPU transfer of the resulting output. Run the executable and check the output in the mandelbrot.png file. Hints: Work initially with only one thread until the result is correct. Then extend to 1D blocks and subsequently to 2D blocks.
2. Introduce separate timings of the kernel and the GPU \rightarrow CPU transfer of the output (or use `nvprof` on the command line). Examine how the size of the mandelbrot set generated and launch configuration influence the runtime.
3. Calculate the speed-up you obtain compared to the OpenMP version from week 2. Calculate both the speed-up including transfer and excluding transfer. Is the speed-up as you would expect?
4. How close are we to the peak performance of the used GPU? Hints: You can use `nvprof --metrics flops_sp` to find the flop count of your kernel. Alternatively, summing all values in the result image gives the number of iteration of the innermost loop. You can count the flops done in the innermost iteration and multiply this with the number of iterations.