Assignment2: Heatdiffusion openCL

Given is a C program for implementing the approximation of a simple heat diffusion scheme in 1D (three-point stencil). This assignment explores how the heat diffusion algorithm from the first assignment can be parallelised using openCL.

Your tasks are to:

- develop an openCL version
- explore and potentially improve the effectiveness of your implementation on a GPU
- explore and potentially improve the effectiveness of your implementation on a multicore CPU
- provide an extensive performance analysis

This assignment should be done by teams of 4 students. How you distribute the work within the team is up to you. However, you need to declare who did which part. Ideally, you perform all performance measurements on a single system. If that constitutes a logistic problem, you can use different hardware provided you clearly specify which hardware has been used for which measurements.

Make sure that you:

- specify exactly what hardware is being used (CPU/GPU version, clock frequency, memory, etc.)
- specify exactly what software is being used (compiler version, compiler flags, etc.)
- specify exactly which parameters (HEAT and EPS) you are using
- repeat each experiment at least 5 times and report average time as well as the variability (error bars)

1 Task 1: openCL

Program an openCL version of the heat diffusion algorithm from Assignment 1. Make sure you check for errors on all openCL calls; did memory allocations suceed? did transfers succeed? did kernel executions complete without errors? etc. Make sure that your workgroup size divides the overall thread pool size in all dimensions.

Hand-out: 03/06/2020 Submission deadline: 16/06/2020 24:00

2 Task 2: tune for GPU performance

Try to improve the runtime performance you obtain on your GPU. If you have no openCL capable GPU in your system, you may use the shootout system at https://lb.oi.pe. Some attempts here are sufficient here, e.g., is your convergence check done in parallel as well? What is a good work group size for your GPU? Describe what you tried and what runtime effects you observed.

3 Task 3: tune for CPU performance

Try to improve the runtime performance you obtain on your CPU. As in Task 2, some attempts here are sufficient here. Describe what you tried and what runtime effects you observed.

4 Task 4: Performance

Provide a short discussion of your overall findings: Compute the absolute performance you achieved on the two systems (CPU and GPU). For that purpose, we use FLOPS (floating point operations per second) as measure. You may ignore the floating point operations for the convergence check, i.e., you can assume you make $n*5*num\_its$ floating point operations in total, where n is the length of the vector and num_its is the number of iterrations performed.

Measure at least 3 different vector sizes and 3 different heat values. Do these values affect the absolute performance on either system?

5 Task 5: Team

Provide a short description on how you divided up the work, i.e., who did what?