Universität Heidelberg
Institut für Technische Informatik

Holger Fröning
Lorenz Braun

GPU Computing
Term 2016/2017 (Winter)

Exercise 9

- **Return electronically until Thursday, 19.01.2016 14:00**
- **Include name on the top sheet. Hand in only one PDF.**
- **A maximum of three students is allowed to work jointly on the exercises.**
- **Hand in source code if the exercise required programming.**

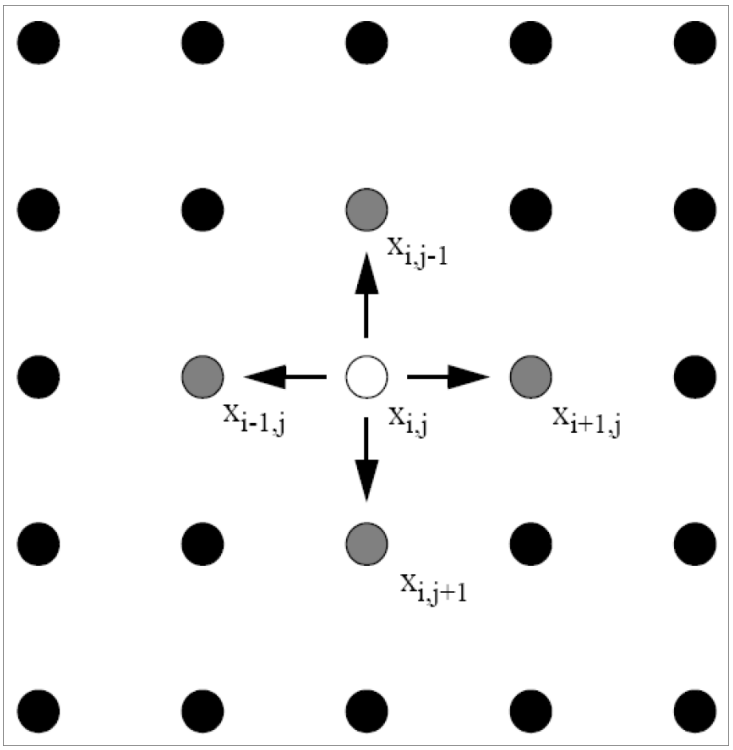
9.1 Reading

- Justin Holewinski, Louis-Noël Pouchet, and P. Sadayappan. 2012. High-performance code generation for stencil computations on GPU architectures. *In Proceedings of the 26th ACM international conference on Supercomputing* (ICS '12). ACM, New York, NY, USA, 311-320.

(15 points)

Heat Relaxation

Relaxation is a mathematical technique used for modeling or the simulation of dynamic processes (heat distribution, material yielding, etc.). In this technique, the solution of a multi-dimensional function is mapped to a discrete grid. The value of a given grid point is dependent on the values of the previous time step, usually of the point itself and its surrounding neighbors. Implement a program, which calculates the new value of grid points as average of the point itself and its four direct neighbours.



The value of a grid point for the next time step t+1 calculates as follows, note that i and j are the coordinates of this grid point:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \Phi \cdot ((-4) \cdot x_{i,j}^{t} + x_{i+1,j}^{t} + x_{i-1,j}^{t} + x_{i,j+1}^{t} + x_{i,j-1}^{t}) \qquad \Phi = \frac{24}{100}$$

- Dynamically allocate a grid of NxN single precision floating point values. The allowed value range is [0,127].
- Inject heat in the topmost grid points (j=0), with i in between [N/4, 3N/4]. These grid points are set statically to 127. The value of all other grid points at the borders is statically set to 0.
- The size of the grid N shall be configurable using a command line parameter.

9.2 Stencil Code – Naïve implementation

- Write an unoptimized GPU program that performs the described calculation.
- Test your program extensively.
- Optionally: try to visualize the output to check correctness.

(20 points)

9.3 Stencil Code – Optimized implementation

- Optimize the previous program using techniques explained in the lecture or in the reading. It's up to you to choose appropriate techniques.
- Test your program extensively.
- Compare the performance results to the unoptimized version. Report performance numbers in terms of the achieved memory bandwidth [GB/s].

(40 points)

Total: 75 points