Heidelberg University
Institute of Computer Engineering
Computer Engineering Group

Holger Fröning
Bálint Soproni

Introduction to High-Performance and Distributed Computing
Winter Term 2023/2024

Exercise 4

- **Hand in via Moodle until 23:59 on Monday 27 November, 2023**
- **Include all names on the top sheet. Hand in a single PDF.**
- **Please compress your results into a single archive (`.zip` or `.tar.gz`).**
- **Please employ the following naming convention `hpc<XX>_ex<N>.{zip,tar.gz}`, where `XX` is your group number, and `N` is the number of the current exercise.**
- **A maximum of three students is allowed to collaborate on the exercises.**
- **In case an exercise requires programming:**
  - **include clean and documented code**
  - **include a Makefile for compiling**

4.1 Reading

Read the following paper and provide a review as explained in the first lecture (see slides):

- Knüpfer, Andreas, et al. "Score-p: A joint performance measurement run-time infrastructure for periscope, scalasca, tau, and vampir." Tools for High Performance Computing 2011. Springer, Berlin, Heidelberg, 2012. 79-91.

(10 points)

4.2 Heat Relaxation – Sequential Implementation

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 neighbors.

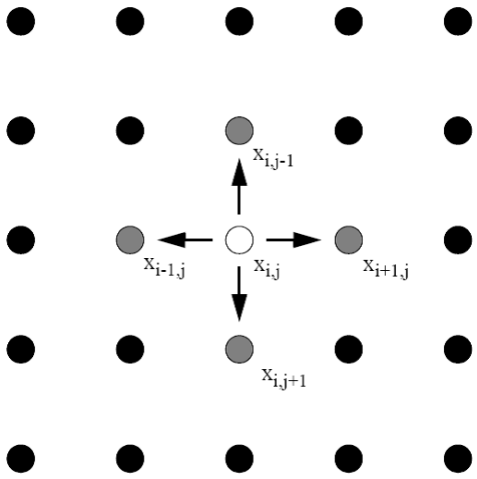

Figure 1: Example grid with coordinates.

Holger Fröning
Bálint Soproni

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:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \Phi \cdot ((-4) \cdot x_{i,j}^{t} + x_{i+1,j}^{t} + x_{i-1,j}^{t} + x_{i,j+1}^{t} + x_{i,j-1}^{t}) \quad (1)$$

$$\Phi = \frac{24}{100} \quad (2)$$

Excercise

- The size of the grid N shall be configurable using a command line parameter.
- Dynamically allocate a grid of $N \times N$ double precision floating point values. The allowed value range is $[0, 127]$.
- Inject heat in the topmost grid points ($j = 0$), with i in between $[\frac{N}{4}, \frac{3N}{4}]$. These grid points are set statically to 127. The value of all other grid points at the borders is statically set to 0.
- Write a suitable sequential program that performs the described calculation. Test your program extensively.
- Try to visualize the output to check correctness.

(25 points)

4.3 Heat Relaxation - Experiments

- Measure the average time for grid sizes of 128×128, 512×512, 1024×1024, $2k \times 2k$, $4k \times 4k$. Report the average time of one iteration by performing for instance 100 iterations, measuring the time with a suitable function (e.g., *gettimeofday()* in Linux) and dividing by the number of iterations. Do not include time for initialization or output. Use compiler-specific optimizations to minimize the runtime.
- Determine the number of FLOPs achieved for each of the grid sizes from above.
- Is this program computationally bound or memory-bound? Provide a detailed explanation!
- Interpret results!

Grid size	Time/iteration	Flops total	GFLOP/s
128×128			
512×512			
1024×1024			
$2k \times 2k$			
$4k \times 4k$			

(15 points)

4.4 Heat Relaxation – Pre-considerations for parallelization

- In the next exercise we will parallelize this application using message passing (obviously). For preparation, develop now a suitable partitioning and task model.
  Propose a suitable model, and try to answer the following questions:
  - How would you decompose this problem into sub-tasks? Which dependencies between tasks do exist?
  - Is 1D or 2D partitioning more suitable?
  - Are there opportunities to leverage overlap between computation and communication?
- Explain in detail and include a drawing to visualize your approach!

(15 points)

Total: 65 points