UNIVERSITY OF AMSTERDAM

Informatics Institute
Systems and Networking Lab
Parallel Computing Systems Group

Dr.ir. Ana Varbanescu
Dr. Clemens Grelck
Julius Roeder, MSc
Misha Mesarcik, MSc
February 1, 2021

Programming Multi-core and Many-core Systems

— Assignment 1 —

Assignment 1.1: Sequential C base version of heat dissipation simulation

Your initial task is to write a sequential C program that simulates heat dissipation on the surface of a cylinder according to the specification given in the lab project description. Pay attention to both maintainability and extensibility of your code, but first and foremost aim at "fast" code. For example, avoid function calls in (inner) loop bodies and consider the cost of different memory storage layouts for the matrices involved in the simulation.

Hint: In C multi-dimensional arrays are stored in memory in row-major order.

Assignment 1.2: Performance impact of compilation

Explore the impact of compilers and compiler optimisations on the effective runtime performance of your heat dissipation simulation code. Try out both the GNU C compiler `gcc` and the Intel C compiler `icc` and compare the performance of the codes they generate.

Experiment with different optimisation levels, and try to improve performance by adapting code generation to the DAS-5 hardware. Consult the documentation of the compilers for specific performance-enhancing optimisations and explore their effect. (The sky is the limit...)

Assignment 1.3: Experimentation with sequential heat dissipation code

Run experiments with your sequential base implementation of the heat dissipation simulation with the following parameter sets:

- square surfaces (at least) with M=N=100, 1000, 2000;
- rectangular surfaces (at least) with N=100 and M=50, 200, 1000, 2000;
- rectangular surfaces (at least) with N=50, 200, 1000, 2000 and M=100.

Follow the guidelines for experimentation set out in the project description. Observe and compare, in particular, the FLOP/s achieved. Does the orientation of the cylinder surface in the non-sqare experiments affect the performance achieved? Explain your observations with references to the architecture of the computing system used.

Assignment 1.4: Vectorisation (SIMD)

Hand-vectorise your heat dissipation simulation code using intrinsics for the DAS-5. Compare the performance of your hand-vectorised code with your original code version. Comment on the improvement or lack thereof.
Hint: do the compilers possibly automatically vectorise your code?

Use compiler flags, pragmas and/or code-rewriting to have the compiler automatically vectorise your original code. Comment on your findings in terms of performance and productivity.
Hint: check compiler reporting and/or the generated assembly code.

Did compilers succeed to perform vectorisation?
Did the performance improve?
Which version is better, manually or automatically parallelised?

Submission due date: Tuesday, February 9, 2021