

Parallel Computing

Institut für Informatik der Ludwig-Maximilians-Universität München
Vorlesung und Übung im Wintersemester 2017/18
Prof. Dr. D. Kranzlmüller, Dr. K. Furlinger

Work sheet 3

Due Sunday, 03. December 2017, 23:59 am on Uniworx.

Please submit your answers **in a single PDF document** if not stated otherwise.

1. Project assignment: Optimized matrix product implementation

Solutions are due in **three weeks**. You may work in groups of up to three students.

- The overall goal is to implement matrix-matrix multiplication (matrix product, DGEMM) optimized for the SuperMUC, restricted to square matrices.
- If you missed the previous tutorial sessions, contact tobias.fuchs@nm.ifi.lmu.de for SuperMUC login credentials.
- As a starting point, a simple benchmark of an unoptimized implementation and a Makefile using default compiler flags is attached to this assignment.
Run `make` to compile the benchmark application.
You presumably only need to change the implementation of the function `mmult`.
- Try to achieve the best performance for $N=1000 \times 1000$ matrices.
Experiment with any thinkable performance tweaks, such as:
 - Blocking / tiling to reduce cache misses
 - Changing memory layout of matrix values
 - Using a different compiler (icc, gcc, clang, vc, ...)
 - Compiler flags (-O2/-O3/...) and annotations, see <http://www.lrz.de/services/compute/supernuc/programming/#TOC4>
 - Loop unrolling
 - SSE / AVX assembly / intrinsics
 - Be creative!¹
- Submit your best implementation variant (including Makefiles and any other prerequisites needed to compile and run it) and achieved MFLOPS as an archive file (zip / tar / tgz / ...).

¹ Graduate students at Berkley invented automatic performance tuning in the 1990s while working on a similar homework assignment:
[http://research.microsoft.com/en-us/um/redmond/about/collaboration/Par-lab book/Chapter-11.pdf](http://research.microsoft.com/en-us/um/redmond/about/collaboration/Par-lab%20book/Chapter-11.pdf)