February 12th, 2025

Oblig 2 IN3030/IN4330 – v2025
Cache-Friendly Parallelization of Matrix Multiplication

This compulsory assignment (Oblig) is about Matrix Multiplication and Caching.

Given two matrices `A` and `B` of size `N` x `N`, you are to implement and run three variants of Matrix Multiplication `A x B`: one using the classic algorithm, one where `B` is transposed, and one where `A` Is transposed. For each version, you should implement and run both a sequential and a parallel version of Matrix Multiplication and measure and report on the speedup achieved.

You are to run with matrix sizes of `100x100`, `200x200`, `500x500`, and `1000x1000`.

Your program must be able to run regardless of the number of cores and should utilize them all by default; you can also optionally set a maximum number of cores as a parameter to the program.

In your solution include a check that you get the correct results, based on the sequential algorithm. Beware, that due to rounding errors, the results might differ very slightly—for floating point numbers `(a+b)+c` may not be exactly the same as `a+(b+c)`.

Show the speedup results both in table form and show a graphical representation of the speedup results.
You should achieve a speedup for at least some of the larger matrices.
Furthermore, show a graph comparing all six variant – use the variant that is the fastest (for large `N`) as a baseline, so that the others are depicted relative to the fastest.

You must use the published precode that we provide to fill the two arrays to be multiplied. The precode is available on the course web page.

Be aware that the sequential code can be slow, especially for larger matrices.

You must write a short report explaining how you did the parallelization, how you synchronized and why (or why not) you achieved a speedup. Comment on why the speedup varies with different sizes of the matrices. Include the table and graphs in the report. It should also contain a short user's guide explaining how to execute your code. The submitted code must be executable by following the user's guide.
The report is to be delivered in PDF format. Your report, the Java code and any other file are to be uploaded as a *single* zip file or tar file. The uploaded code must be compilable and runnable. At least some of your speedups should be greater than one. For the report, use the template given below.

For IN4330 students: You must, in addition, run your program on two *different* machines, *e.g.*, your own laptop and one of Ifi's machines. Thus, the results and graphs are in two versions: one set for each machine. Comment on the difference.

Note: for the highly skilled parallel programmer – there are even better methods than this one, *e.g.*, *tiling* but we are requesting that you do the simpler transposition solution described above ;-)

Tiling will be discussed briefly in a later lecture.

Delivery: Deadline is February 25th, 2025 at 23:59:00 CET – that is 23:59:00 local time in Oslo.

Deliver in devilry.

Template for report:

1. Introduction
2. Sequential Matrix Multiplication – *short description*
3. Parallel Matrix Multiplication – *how you did the parallelization*
4. Measurements – *includes discussion, tables, graphs*
5. User guide – *how to run your program*
6. Conclusion – *just a short summary*

Appendix:

- Your Java code
- Any supporting files, *e.g.*, makefile, whatever …