

GPU Programming 2019/20

Assignment 1

Due: 10/11/2019, 23:59

In this assignment we implement parallel matrix-matrix multiplication using `std::thread`. Your tasks are as follows:

- 1.) Implement the function `void MatrixMultiplicationSerial(const double * A, const double *B, double * C, const unsigned int size)` which performs a serial version of matrix-matrix multiplication for squared matrices with dimensions $\text{size} \times \text{size}$. Here we interpret the arrays `A` and `B` as the input matrices and the array `C` as the output matrix ($A \cdot B = C$). Make sure your implementation accesses memory in an efficient manner. (3 points)
- 2.) Initialize the variable `num_threads` with the number of threads at disposal in your machine. Use `std::thread::hardware_concurrency()`. (1 point)
- 3.) Implement `void MatrixMultiplicationParallel(...)`, extending the function implemented in 1.). Perform the matrix multiplications with `num_threads` threads. (4 points)
- 4.) Measure the performance of your matrix-matrix multiplication for $\text{size} = 2^j$ with $j = 6 \dots 11$ for the serial version and the parallel version. Generate a plot of the results. (3 points)

Bonus: Implement a parallel matrix-matrix multiplication for rectangular matrices. Check the compatibility of the dimensions of the input matrices. (2 points)

Please submit your implementation and the graph before the deadline to gpu@isg.cs.ovgu.de. Your code has to compile and run with the given CMake file on the machines in G29-426.