Prof. Dr. Kai Diethelm
Faculty of Applied Natural Sciences and Humanities
Technical University of Applied Sciences Würzburg-Schweinfurt

Summer Semester 2024

Parallel Programming
Assignment 2

Deadline: Thursday April 25, 2024, 23:59 CEST

General Notes:

- For the programming assignments, you may use either of the programming languages C, C++, or Fortran. On the FANG HPC system, the corresponding compilers can be invoked by the commands `gcc`, `g++` and `gfortran`, respectively.
- In order to submit your solutions, follow the instructions given in the course webpage in our Elearning system.
- Question 3 is a bonus question. If you decide not to answer it, this will not have any negative effects on your grade. If you do answer it, you may collect some extra points that can help you obtain a better final grade.

  The final grading scheme will look as follows:

Percentage of total collectible points from all question combined except for bonus questions	Grade
$\geq 95\%$	1.0
$\geq 90\%$ and $< 95\%$	1.3
$\geq 85\%$ and $< 90\%$	1.7
$\geq 80\%$ and $< 85\%$	2.0
$\geq 75\%$ and $< 80\%$	2.3
$\geq 70\%$ and $< 75\%$	2.7
$\geq 65\%$ and $< 70\%$	3.0
$\geq 60\%$ and $< 65\%$	3.3
$\geq 55\%$ and $< 60\%$	3.7
$\geq 50\%$ and $< 55\%$	4.0
$< 50\%$	5.0 (fail)

Mathematical Background: Given two matrices

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1q} \\ b_{21} & b_{22} & \cdots & b_{2q} \\ \vdots & & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nq} \end{pmatrix}$$

of dimension $(m \times n)$ and $(n \times q)$, respectively, their product $C = A \cdot B$ is an $(m \times q)$ matrix given by

$$C = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1q} \\ c_{21} & c_{22} & \cdots & c_{2q} \\ \vdots & & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mq} \end{pmatrix} \quad \text{with} \quad c_{jk} = \sum_{\ell=1}^{n} a_{j\ell} b_{\ell k} \quad (j = 1, 2, \ldots, m; k = 1, 2, \ldots, q). \qquad (*)$$

Questions:

1. Write a function `matrixProduct` whose parameter list contains the following arguments:

   - `m`, `n`, `q` (input)—positive integers with the dimensions of the matrices A and B as given above,
   - `a`, `b` (input)—a suitable structure of double precision numbers with the coefficients of the matrices A and B,
   - `cs`, `cp`, `dc` (output)—a suitable structure of double precision numbers with the coefficients of the matrices outlined below,
   - `ts`, `tp` (output)—double precision variables to be filled with timing information as specified below.

   The output variables `cs`, `cp` and `dc` as well as `ts` and `tp` should be filled with data in the following ways:

   (a) In the first part of the function body, you should implement some code that fills the variable `cs` with the coefficients of the matrix $C = A \cdot B$ using the formula given in equation $(*)$.

   - Write this part of the code in a sequential way, without any parallelization.
   - Write the complete code by yourself. Do not use a matrix multiplication function imported from some library.

   In addition, fill the output variable `ts` with the amount of time that your sequential matrix multiplication code requires to run.

   (b) In the second part of the function body, you should also implement some code that fills the variable `cp` with the coefficients of the matrix $C = A \cdot B$ using the formula given in equation $(*)$.

   - Write this part of the code in a parallel manner, using OpenMP. Identify a loop that admits such a parallelization.
   - Write the complete code by yourself. Do not use a matrix multiplication function imported from some library.

   In addition, fill the output variable `tp` with the amount of time that your parallel matrix multiplication code requires to run.

   (c) In the last part of the function body, you should write some code that fills the variable `dc`. This variable should contain the coefficients of the difference matrix `cs` − `cp`.

   (300 Points)

2. Write a main program that takes two command line arguments. The command line arguments should be interpreted as character strings containing file names. The program should first read the coefficients of the matrices A and B from these two files in the same way as the program from Question 2 of Assignment 1. Next, your program should call your function `matrixProduct` from Question 1 of this assignment. Finally, your program should

   (a) print the matrices `cs` and `cp` computed by your function in a reasonably formatted manner,

   (b) print the largest and the smallest coefficient of the matrix `dc` computed by your function,

   (c) print the timing information provided by the function's output variables `ts` and `tp`.

   All output should go to the standard output channel.

   (50 Points)

3. (*Bonus question*) Write another main program with the same functionality as the one from Question 2, except that your output should only contain the information specified in items (b) and (c) of Question 2, but not (a). However, instead of reading in the matrices with your approach from Question 2 of Assignment 1, use a suitable I/O function from the *MatrixMarket* library.

   The *MatrixMarket* library I/O functions require the files to be in a specific data format. To test your program, you may find a large number of suitable input data files in this format, for example, in the *Harwell-Boeing Matrix Collection*.

   (100 points)