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

Summer Semester 2024

Parallel Programming
Assignment 6

Deadline: Monday, June 17, 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.
- To submit your solutions, follow the instructions given in the course webpage in our Elearning system.

Background: In the OpenMP chapter of the lectures, we had discussed the Gauss-Jacobi method for (approximately) solving large systems of linear equations. In particular, we had derived the sequential form of the algorithm and a parallel version using OpenMP. An MPI version of this algorithm was the topic of Assignment 5.

The Schur complement algorithm for solving certain special kinds of linear equation systems was presented at the end of the MPI chapter (slides 157 and 158).

Questions:

1. Take the MPI program that you wrote to solve Assignment 5 and convert it into a program with hybrid parallelism, using MPI and OpenMP.

   Modify, if necessary, the termination condition for the Gauss-Jacobi iteration in such a way that the user may input a maximal number of iterations and an upper bound for the difference of two consecutive approximate solutions. The iteration should be stopped as soon as one of these criteria is fulfilled.

   Create a number of large equation systems whose solutions you know and test the program with these systems. Please also upload the input and output of the program to your directories.

   Provide detailed comments explaining the MPI and OpenMP commands that you use.

   Test your program on the compute node `fang-s010` with

   - 3 processes and 3 threads per process,
   - 5 processes and 3 threads per process,
   - 3 processes and 5 threads per process,
   - 7 processes and 2 threads per process,
   - 2 processes and 7 threads per process.

   Report the results and the run times.

   (600 Points)

2. (*Bonus question*) Implement the Schur complement method for solving linear equation systems in the form indicated on slide 158, i.e. with an OpenMP parallelization of the second step and a hybrid parallelization of the third step.

   Create a number of large equation systems whose solutions you know and test the program with these systems. Please also upload the input and output of the program to your directories.

   Provide detailed comments explaining the MPI and OpenMP commands that you use.

Test your program on the compute node `fang-s010` with

- 3 processes and 3 threads per process,
- 5 processes and 3 threads per process,
- 3 processes and 5 threads per process,
- 7 processes and 2 threads per process,
- 2 processes and 7 threads per process.

Report the results and the run times.

(400 Points)