Lab Introduction to High Performance Computing(WI4049TU)

Mathematics Physics Group, Faculty Electrical Engineering,
Mathematics, and Computer Science(EEMCS/EWI)
Delft University of Technology

1 Introduction

MPI is undoubtedly the most popular programming model for distributed platforms such as clusters and supercomputers. The lab exercises will be performed on Delftblue supercomputer. The emphasis will be on designing efficient parallel algorithms and on performance optimization. The lab course score will count for 50% of the final grade. At the end, a final report should be handed in, which contains answers and measurements, it should also include a final version of the code. The score will be determined based on analysis and discussions.

1.1 MPI Introduction

This small "Hello world" exercise will guide you into the MPI world from scratch. If you are a newbie in MPI, you should not skip it.

2 Exercise

There are 4 separate exercises [1] [2] to give you hands-on experience in high performance programming and you can optimize the code step-by-step.

(0) Introductory exercises: Measuring the times of MPI_Send and MPI_Recv pair as a function message length; Make a parallel MPI implementation for the matrix-matrix product.
(1) Poisson solver: This exercise will teach you to program a fairly complex application-Poisson kernel and let you learn to use MPI mechanisms. After that, you are supposed to experiment with your parallelized code on Delftblue.
(2) Finite elements simulation.
(3) The eigenvalue solution by Power Method on GPU using Cuda.

---

[1] The exercise specifications and the initial code can be downloaded from Brightspace
[2] You can also download this electronic instruction copy from Brightspace.

2 How to Use the Parallel Platform

A convenient way to work on the Delftblue is **to use the browser to access login.delftblue.tudelft.nl, the Open OnDemand environment** allows you to upload/download files. After login, click 'Files' and select "Home Directory", you will see the files and directories of your home directory /home/Netid. With the option "_Open in terminal" you can open a terminal to compile and execute your program. (PS. alternatively, for Linux users, you can use (e.g., from your Windows laptop)

```
ssh -X netid@login.delftblue.tudelft.nl)
```

1. Compile
   You need to first compile your MPI program on Delftblue. After you have logged onto Delftblue, type the following to load the usual package and the openmpi modules:
   module load 2024r1 openmpi
   Now you can use mpicc to compile your programs:
   mpicc myprog.c -o myprog
2. Execute
   Executing your program has to be done using 'srun' (Slurm scheduler)
   srun --ntasks P --cpus-per-task=1 --mem-per-cpu=1GB --account=Education-EEMCS-Courses-WI4049TU myprog (here: myprog=name of the executable file)
   or use a convenient alternative command:
   srun -n P -c 1 --mem-per-cpu=1GB myprog
   where P is the number of MPI processes (=task or programs) to be executed in parallel. The variable cpus-per-task controls how many cores are used for each process, it is set to 1 unless multiple threads within one process are created (e.g., in hybrid MPI+OpenMP programming). The variable mem-per-cpu reserves an amount of memory for each process (in case of 1 process per core), it can be specified in terms of MB or GB, 1 GB is more than enough for the lab exercises.
   Other useful additional options:
   --nodes=2, this will reserve a total of 2 nodes with a total of 96 cores (each node has 2x24 cores). Don't do this too often (waste resources), only for specific tests.
   --account=Education-EEMCS-Courses-WI4049TU, when the schedule complains about claiming too much resources or run time, this might help (though with no guarantee).

   For more information about using Delftblue, please visit DelftBlue Documentation (tudelft.nl)
   For example, you can find explanations about Crash-course on DelftBlue for absolute beginners, Linux Command Line Basics Course and DelftBlue

Supercomputer Course.

Note that if you do not want to input these command lines each time you can log onto the server, and use an editor to append these lines to your local .bashrc file. You can also define a script for use when you have to give many values/choices.
Editing: there are plenty of options you can choose. For example, in the OnDemand environment, you can select 'edit' of a file which opens a new window for editing. Or in the command terminal (after opening one with 'Open in terminal'), you can use nano, **vi**, vim or gedit (gedit is easier for newers) for editing: *vi filename.c* or *nano filename.c*.
Online manual: If you want to know the usage of a certain command/program, type e.g., *man mpicc* for an explanation, and if you don't the exact name of a command you can also try with *man –k keyword* (with keyword is about the function of the command you are searching).
Be careful that a typo mpicc -o filename1.c filename1 can result in your c program being overwritten and lost.

◇ Follow Steps into the MPI World! in the "Helloworld" example.

3 Lab report

- A final report on the lab exercises must be submitted which contains all your answers to the questions in the three exercises.
- Any figures, analysis, and tools illustrating or demonstrating your answers will benefit your overall grades.
- The final report should be completed independently. It should have the following information on the front cover: your name(s), studentnumber(s), email address(s), and submission date.
- Submit your report to Lab reports in Assignments on Brightspace.
- Submission Deadline: **February 3rd 2024**.

4 Assistance

The lecturer and TAs will assist during the Friday lab sessions. If you have any questions/suggestions about the lab exercises, you can contact the TAs: A.Carraro@student.tudelft.nl .

5 Grading Rules

The following rules will be followed to grade your assignments.

The base grades for satisfying the answers of the exercises are Introductory exercise (0.5), Poisson solver **(**1.75), Finite elements simulation **(**1.0), and eigenvalue solution by Power Methodon GPU **(**1.75). You will get a grade of 5.0 points when you finish all the basic questions.

A multiple of 0.5 points will be added to your total grade (until 10.0), according to the shining points (e.g., performance analysis, optimization, discussion, and clarifying figures).