HPC: assignment 1

Guillaume Trojani

February 1, 2021

Overall comments

This scripts uses dynamic allocation of memory using C function malloc and calloc, mostly as an exercise for myself to recall concepts of pointers, etc... I've not used many in-built libraries of C++ because I am only really familiar with the C base. Hopefully this will change eventually. Because of lack of time, no attempt has been made at optimising the code or timing it. Also I have not work on the last exercise. The script comes in 3 files, a main file "matrix_exercise.cpp", a library file with taylored matrix functions "matrix_operation.h" and a make/run file "mpi_run.sh".

Parralelisation strategy

In solving this exercise, I have written an MPI program that can split the work of matrix multiplication of $N \times N$ matrices between *nproc* processors. We have only implemented the case where $nproc|N$. In such cases, we define $quot := N/nproc$ to be the number of rows that will be sent to each processors and perform calculations. Before doing so, we compute the matrix $B \in \mathbb{Z}^{N\times N}$ in the root process and cast it to all other processes. We split the matrix $A \in \mathbb{Z}^{N\times N}$ by rows of size *quot* as

$$\text{A\_quot\_rows} \in \mathbb{Z}^{2\times N}$$

and perform a matric multiplication with the whole matrix B, giving us $\text{C\_quot\_rows} \in \mathbb{Z}^{2\times N}$. These are the components of the matrix multiplication AB, 'quot' rows by 'quot' rows. We then gather the data back to to the array of pointers D using MPIgather. The script will output the normal solution of matrix multiplication for the problem of size N, as well as the parralelised version (for checks).

Script calls

The script is called by passing arguments *nproc* and N to the bash script "/mpi_run.sh". No effort has been made to time the code using in-built functions as of yet.

Example 0.0.1. * In command line, running "./mpi_run.sh 2 4", calls for the script with 2 processors and 4 rows.

I have attached the file in the submission.