Riccardo Basso
Davide Ponzini

MPI Barrier

Introduction

For the MPI Barrier Lab, we always worked through ssh.
All tests, edits to the .cpp or creation of scripts, have been done through ssh and not locally.
The most difficult part was exercise 6, especially because we had to implement our own barrier and we had to think about a good way to do it.
We will also provide some charts containing our results, so we can compare the performance of exercises 5 and 6.

Exercises 1, 2, 3, 4

These exercises were quite easy for us and did not take a lot of time.
We just had to compile and run or sometimes modify short lines of code.
But we've experimented how to work with MPI and how to use barriers.

The source codes can be found inside the various exercise folders.

Exercises 5, 6

These are the most difficult exercises because for the exercise 6 we had to implement our own barrier, without using the library function.

We have also evaluated performance as a function of the number of processes, and compared it against the performance of the native MPI_Barrier() function used in exercise 5.

Both exercises have been run using a bash script, script.sh, which is provided on both folders.
Alternatively, for exercise 6, a Makefile has been provided: just by invoking the command `make n=<N>` the source code is compiled and executed on several machines, their amount depending on the value of N. If no value is provided for N, then it will be executed on 2 machines.

Results

Here we provide our results, plotting the time on the vertical axis and the number of machines on the horizontal one. The blue line indicates the performance of exercise 5's library barrier function, the orange one instead is the performance of the barrier implemented by us:
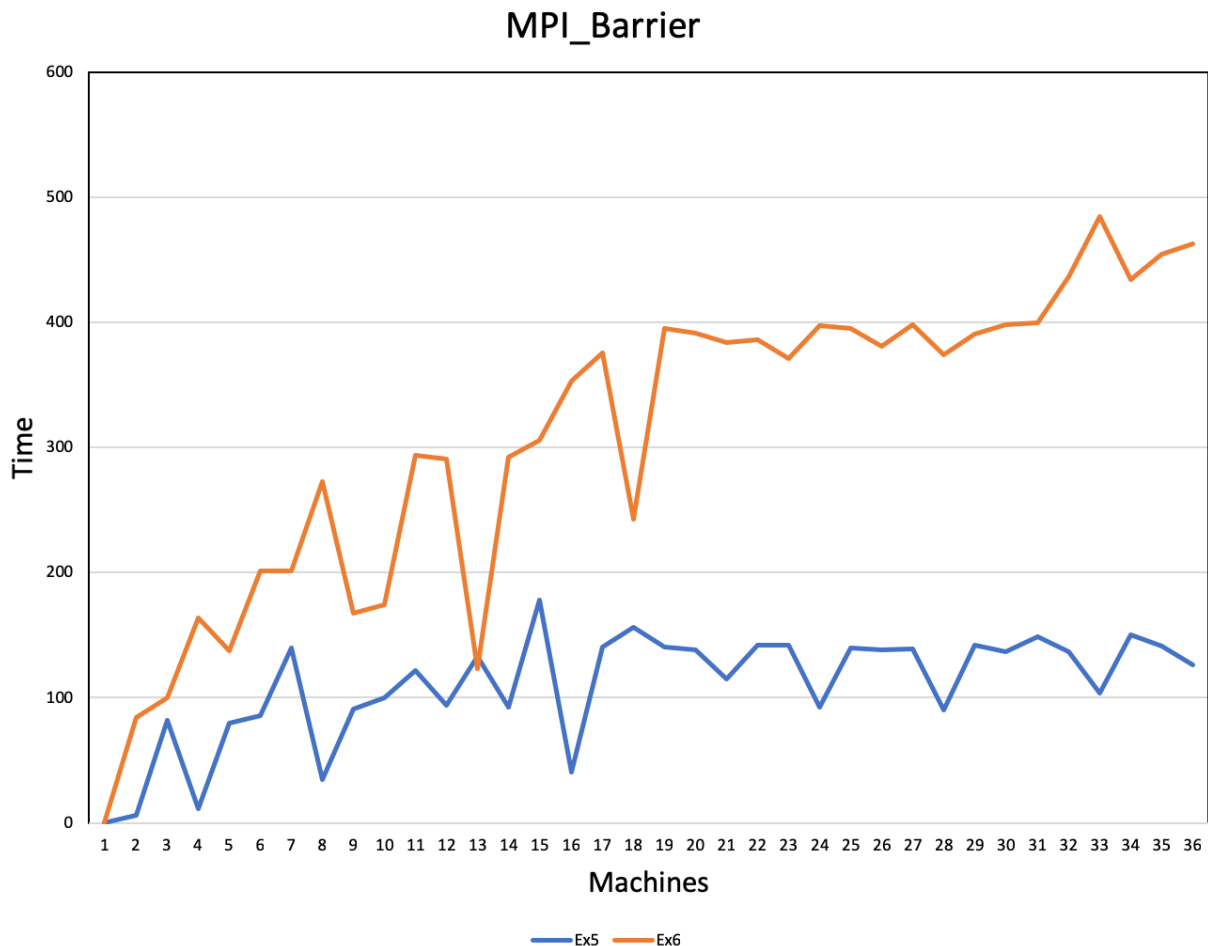
Every single data that we've plotted here is available inside the archive. The two output files are provided in folders 5 and 6.

As you can see, performance for the MPI_Barrier() function in the exercise 5 are better than the one implemented by us.

Also remember that the time is expressed in microseconds.