

Lab 4. MPI – generalities, simple examples

Main goals of the assignment

- Learn about the standard in programming the message passing paradigm.
- Understand the principles of single program multiple data.
- Understand the principles of collective communication.

Overview

Reminder

OpenMP: to use all cores available on our desktop/laptops/servers to speedup the response time of a particular application;

MPI: to use the computation capacity of multiple computers (even millions of them, see Top500).

What is MPI

Please read the presentation from the textbook - page 291 (bottom of the page).

First example

1. Write the Hello world code from the textbook - page 294 (bottom of the page)
2. Understand the need of the basic functions in MPI (See the textbook at page 296).
3. Compile the code; in command line, for example: `> mpicc hello.c -o hello`
4. Run the parallel code, e.g. `> mpirun -np 2 ./hello` and the one-process version `./hello`.
5. Understand the role of the parameter ‘-np’. Modify the number of processes and discuss how to use more machines (using hostfile/machinefile).

Second example

1. Imagine that we want to sum up all the numbers from 1 to 1000. Now admittedly, multiple processes are not needed for this job, but the point of this exercise is for you to see how we partition the problem. To effectively sum up all the numbers from 1 to 1000 using multiple processes, we partition the sums across the processes. If we use two processes, then we want that process 0 sums from 1 to 500, and process 1 sums from 501 to 1000, and then at the end, the two values are added together to obtain the total sum of all numbers from 1 to 1000. Given p processes, the problem of summing is partitioned into p subproblems, and then at the end all processes send their information to process 0 for the final accumulation of the results.
2. Implement the parallel code. See the code from page 313. See also the difference from a sequential code.
3. Understand the parameters of the point-to-point communication functions in MPI (See the textbook at the bottom of page 303).
4. Replace the code concerning the exchange of messages with the collective communication and computation `MPI_Reduce` using `MPI_SUM` (See the textbook page 309).