

MPI Programming exercise: timing

Start with the `time_avg` program, which computes the average time taken for a computation.

The make file has a handy feature: if you type `make` it will not actually do anything, but it will tell you all its capabilities. (Take a look at the makefile. How is this effect attained?)

Compile the program with `make time_avg` and run it in an `idev` session: `ibrun time_avg`.

Now do the following experiments.

1 Timer accuracy

The first thing you have to wonder about when you're doing timings is: how much does the timer disturb the measurements. Find a way to measure how much time the timer itself takes.

2 Collecting results

Instead of just computing the average time for a computation, gather all the timings on the root process and print them all.

3 Trace output

Do the following steps *on a login node* (you may want to keep two windows open):

1. `make total_clean` to remove all binaries
2. `module load tau`
3. Rebuild your program. Do you notice anything different about the output?
4. Now try to get an `idev` session on more than one node: `idev -N 2`.
5. Instead of running with `idev`, do `make idevrun`. (Inspect the makefile to see what magic is going on there!)

6. You'll see that a directory has been created. If you now type `make tauplots`, a file with extension `slog2` is created.
7. Invoke the `jumpshot` program on that `slog2` file.
8. What does it all mean? Experiment with zooming in; see what the different colours mean.

4 Report!

Write up a report of what you've done here. Ideally, your report will have some code snippets and screen shots. Note: every question/suggestion in this assignment is of course an opportunity for writing a paragraph or two.