

Parallel computation: MPI exercises

The following exercise is to be done with your own code, as applicable.

In case you do not have a code to work with, or it is unsuitable for some particular exercise, you can find a sample toy application; a 2D heat equation solver provided via the course web page (see the appendix for details how to compile and run it).

1. Porting applications

Move all your files to Sisu. Adapt your makefiles or other compilation scripts to comply with the Sisu environment. Remember: the compilers are always referred to as `ftn`, `cc` and `CC`. Build your application.

- Prepare a batch job script and submit the job. Make sure it runs correctly.
- (Optional): Measure the strong scaling curve and optionally a weak scaling curve of your application using a representative input dataset (matching also the available core count). Anyone working with the heat equation solver should increase the program size from the default (e.g. to a 1024x1024 grid).

2. Compiler optimization

Recompile your application with the following flags and record the wall-clock time (or other meaningful timing information, e.g. time for one simulation step).

Remember to verify the output.

CCE	Time	GNU	Time	Intel	Time
(no flags)		(no flags)		(no flags)	
-O3		-O3		-O3	
-O3 -hfp3 -hipa4		-Ofast -funrollall-loops		-Ofast -fpmodel fast=2	

If the compilation time of your application is high, feel free to fill in some selected boxes only. The empty boxes are reserved for combinations of your own.