MPI Environment on the RWTH Cluster CLAIX

Before You Start

Before you start, log into one of our Claix-2023 cluster frontends:

- **login23-1.hpc.itc.rwth-aachen.de**
- **login23-2.hpc.itc.rwth-aachen.de**
- **login23-3.hpc.itc.rwth-aachen.de**
- **login23-4.hpc.itc.rwth-aachen.de**

using `ssh` or a corresponding client.

Then, download the MPI lab archive from the PPCES 2026 website and extract it to a suitable location. The files in the archive extract to the `ppces2026-MPI-(all|C|Fortran)/` prefix.

C/C++

```
% mkdir -p ~/PPCES2026/
% cd ~/PPCES2026/
% wget https://hpc.rwth-aachen.de/ppces/ppces2026-MPI-labs-C.tar.gz
% tar -xvf ppces2026-MPI-labs-C.tar.gz
```

Fortran

```
% mkdir -p ~/PPCES2026/
% cd ~/PPCES2026/
% wget https://hpc.rwth-aachen.de/ppces/ppces2026-MPI-labs-Fortran.tar.gz
% tar -xvf ppces2026-MPI-labs-Fortran.tar.gz
```

Both C and Fortran

```
% mkdir -p ~/PPCES2026/
% cd ~/PPCES2026/
% wget https://hpc.rwth-aachen.de/ppces/ppces2026-MPI-labs-all.tar.gz
% tar -xvf ppces2026-MPI-labs-all.tar.gz
```

The lab archive contains skeleton code for the exercises described below. Intermediate solutions are provided where appropriate. Sample solutions to all problems are also provided in the **solutions** folder. We would advise you to not look at the solutions before you have tried your best to solve each exercise on your own.

Building the example code

The CLAIX shell environment uses the `module` system to provide different versions of specific software. By default the *Intel MPI* and *Intel compiler* modules are loaded. It is easiest to copy the `make.def.intel` for Intel compiler & MPI to `make.def` in the respective `common/` directory of your exercises.

Each problem comes with a Makefile with the following targets: ##### `default` target

This target build the respective exercise executable.

```
% make
```

`clean` target This target removes all build files, including the executable itself.

```
% make clean
```

`run` target This target can be used in an interactive session starting 4 processes with the given executeable in the current allocation.

```
% make run
```

`batch` target This target submits the given executable as a jobscript with SLURM, the RWTH worlkload manager. It will default to using 2 nodes with 2 processes each for a total of 4 processes. You may inspect the batchfile in `common/slurm.batch`.

```
% make batch
```