

Hands-on: Collective communication and Tofu Barrier Interconnect

- Choose either C/C++ (`c`) or Fortran (`fortran`) samples. Both of them are fine, as well.

C/C++

How to compile and how to execute

1. Compile program

- We have three kinds of kernels in `elm3/`, `elm6/`, and `elm6.sep/`.
- The executable file (`run.x`) is generated in each directory.

```
$ cd c/elm3
$ make # make -f Makefile.own # if using own compiler
$ ls
run.x ...
$ cd c/elm6
$ make # make -f Makefile.own # if using own compiler
$ ls
run.x ...
$ cd c/elm6.sep
$ make # make -f Makefile.own # if using own compiler
$ ls
run.x ...
```

2. Run program

- A job script (`task.sh`) to execute the program is located in `results/`.
 - BE CAREFUL: The program works when the number of MPI tasks is greater than 1.
 - In the job script, we set the number of used nodes as 12 (= 1 Tofu). You can change it into a smaller value, but one is not meaningful.
 - In the job script, three kinds of kernels (`elm3`, `elm6`, and `elm6.sep`) are successively executed. Each of the results is saved in the corresponding directory, like `elm3`.
- You can run the program either:

```
## To run as a batch job
$ cd c/results
$ pjsub task.sh
## Or, to run in an interactive job
$ cd c/results
$ bash task.sh
```

- The job in the Exercises will be completed within 2 minutes.
 - For safety, we set the elapsed time of the jobs cript as 4 minutes.

Exercises A

- E1: Check `elm3/main.c`. Which kinds of MPI collective communication routines are there? Among them, which can use communication with the features of the Tofu barrier interconnect?
- E2: As for `elm6/main.c` and `elm6.sep/main.c`, repeat similar consideration to the above.
- E3: According to the MPI statistical information of the program (the standard error file), check whether the Tofu barrier communication is allowed or not.
- E4: Compare the performance (Elapsed time of main part in the standard output file) of `elm3` to the others.

Exercises B (advanced)

- E5: Try `results.wo_tbi`. Compare the result to that in `elm3` of `results/`.

Fortran

How to compile and how to execute

1. Compile program

- We have three kinds of kernels in `elm3/`, `elm6/`, and `elm6.sep/`.
- The executable file (`run.x`) is generated in each directory.

```
$ cd fortran/elm3
$ make # make -f Makefile.own # if using own compiler
$ ls
run.x ...
$ cd fortran/elm6
$ make # make -f Makefile.own # if using own compiler
$ ls
run.x ...
$ cd fortran/elm6.sep
$ make # make -f Makefile.own # if using own compiler
$ ls
run.x ...
```

2. Run program

- A job script (`task.sh`) to execute the program is located in `results/`.
 - BE CAREFUL: The program works when the number of MPI tasks is greater than 1.
 - In the job script, we set the number of used nodes as 12 (= 1 Tofu). You can change it into a smaller value, but one is not meaningful.
 - In the job script, three kinds of kernels (`elm3`, `elm6`, and `elm6.sep`) are successively executed. Each of the results is saved in the corresponding directory, like `elm3`.
- You can run the program either:

```
## To run as a batch job
$ cd fortran/results
$ pjsub task.sh
## Or, to run in an interactive job
$ cd fortran/results
$ bash task.sh
```

- The job in the Exercises will be completed within 2 minutes.
 - For safety, we set the elapsed time of the job script as 4 minutes.

Exercises A

- E1: Check `elm3/main.F90`. Which kinds of MPI collective communication routines are there? Among them, which can use communication with the features of the Tofu barrier interconnect?
- E2: As for `elm6/main.F90` and `elm6.sep/main.F90`, repeat similar consideration to the above.
- E3: According to the MPI statistical information of the program (the standard error file), check whether the Tofu barrier communication is allowed or not.
- E4: Compare the performance (Elapsed time of main part in the standard output file) of `elm3` to the others.

Exercises B (advanced)

- E5: Try `results.wo_tbi`. Compare the result to that in `elm3` of `results/`.