

04_pure-mpi/01_pingpong

Hands-on: IMB: Pingpong (Send and Recv)

- Both of C/C++ and Fortran users will perform the common benchmarks, although IMB is written by C and C++.

How to execute

1. Edit a job script

- Before trying this hands-on, you need to do `00_imb` to compile Intel MPI benchmark.
- A job script to execute the program is `task.sh` in `cs32c1256.bw_node/`. It contains three kinds of measurement settings on `Pingpong` benchmark.
 - The first run is standard.
 - The second run is the same as the first, but the maximum of the message length is shorter than the first (See `msg.txt`).
 - The third run is measurement with setting `OMPI_MCA_btl_tofu_eager_limit` as the minimum value.
- edit `task.sh` (modify `--gname` option).
- Edit `BINDIR` variable in `task.sh` before the execution in `cs32c1256.bw_node/`. You need to write your installed location of IMB binary (e.g., `IMB-MPI1`) there.

2. Run program

- You can run the program either:

```
## To run as a batch job
$ cd cs32c1256.bw_node
$ pjsub task.sh
## Or, to run in an interactive job
$ cd cs32c1256.bw_node
$ bash task.sh
```

- The job in the Exercise will be completed within 3-4 minutes.
 - For safety, we set the elapsed time of the jobs cript as 6 minutes.

Exercises A

- E1: Confirm that the node and MPI settings in `task.sh` are `node=2`, `by-node`, and `PPN(Process-per-node)=1` (i.e., `#PJM -L "node=2"`, `#PJM --mpi "rank-map-bynode"`, and `#PJM --mpi "max-processor-node=1"`). In this case, how are the two processes distributed over nodes?
- E2: Find region of the message length in which the measured bandwidth is almost flat (i.e, the measured latency is almost linear).
 - We note that the measured results of IMB are shown in the standard output (e.g., `out-***`).

- E3: Check the MPI statistical information and the communication mode in point-to-point (P2P) communication. Compare the latency in the first run to the others, with respect to the message length.
 - We note that the MPI statistical information is shown in the standard error (e.g., `err-***`).

Exercises B (advanced)

- E4: Check the job statistical information file (`*.stat`). How are the node and ranks allocated?
- E5: Try `cs32c1256.bw_cmg`. In turn, one will measure the bandwidth between CMGs (i.e., intra-node communication performance).
- E6: Check `simple-p2p`. This directory contains two kinds of simple implementation of `Pingpong`, in `std/` and `sync/`. The former is equivalent to `PingPong` in IMB, composed of `MPI_Send` (i.e., standard communication mode) and `MPI_Recv`. The latter is different, composed of `MPI_Ssend` (i.e., synchronized communication mode) and `MPI_Recv`. Thus, the latter may be expected to always use the Rendezvous protocol, independent of the message length. Compare the results between `std/` and `sync/`. Also, confirm that the results in `std/` is similar to the measurement results with IMB's `Pingpong`.