Exercises - MPI

Day 12

Exercises: MPI #2

- Write a parallel ‘hello world’ program using MPI:
  - Each processor should output it’s rank, processor name and the total number of processors.
  - Make only one processors write the ‘hello world!’.
  - The I/O from the processors generally come in random order; Change the code such the order is ordered according to the rank (not trivial! Consider writing IO to disk).

Exercises: MPI #2

- Write a program for MPI ping-pong: process 0 sends a message (fx. one double precision number) to process 1 (ping). After receiving this message, process 1 sends the message back to process 0 (pong).
  - Repeat this 10 times and compute the average and min/max bandwidth (message size/transfer time).
  - Vary the message size (from one double precision number to 2**18) and estimate the latency and bandwidth between the processors.

Exercises: MPI #2

- Hint:
  - Use the module:
    - `module add studio`
    - `module add mpi/3.1.3-oracle-12u6`
    - Compile using: `mpif90 –free –O3 example.f`
    - Link using: `mpif90`.
  - Let the odd processor send and then receive and let the even processor first receive and then send.
  - Use t1 = MPI_Wtime() to perform the timing.
  - Assume the time *T* for sending a message of size *n* is:

$$T(n) = T_{latency} + n/bandwidth$$

Exercises: MPI #2

- Hint:
  - Gnuplot can fit your data:

```
gnuplot
f(x) = a + x*b
fit f(x) ‘data.dat’ u 1:2 via a,b
p f(x) w l, ‘data.dat’ u 1:2 w lp
```

Exercises: MPI #2

- Hint:
  - LSF queueing: bsub < `submit.sh`

```
#!/bin/sh
#BSUB -J job-name
#BSUB -q hpc
#BSUB -n 2 # specify number of cores
#BSUB -W 10:00 # specify maximum wall clock time
#BSUB -u s000000@student.dtu.dk
##BSUB -R “span[hosts=1]” # reserve all cores on one node
#BSUB -R “span[ptile=1]” # reserve one core per node
#BSUB -R “rusage[mem=16GB]”
#BSUB -B # send notification at start
#BSUB -N # send notification at end
module add studio
module add mpi/3.1.3-oracle-12u6
mpirun -np ${LSB_DJOB_NUMPROC} ./myexefile
```