

Lab #5 MPI Programming IV

Purpose: to learn how to use non-blocking send and receive, and balance workload among different processes.

1. (35) Computing performance can be improved by overlapping the communication and computation of your program. In addition, in a heterogeneous computing environment with various computing nodes and dynamic workload, balancing workload among different processes/machines can also improve the overall computing performance. In this lab, you are required to rewrite lab#3 and your program should

- Take a master-slave model. All three matrices exist only on the master process at the start and finish of the computation. The master process manages a pool of available tasks. Whenever a slave process has no work to do, it requests an available work from the master process until all work is completed.
- Have two buffers. While one buffer is used to send or receive data, the other one is used to do computation, and vice versa. While the master can use blocking send and receive, all slaves must use *non-blocking* send and receive.
- If $n \leq 16$, your program should print out the matrix A, B, and C.
- Use the master process to record the total computation time. Set appropriate n and n_{pes} shown in the following table. Run your program and fill out the form with relevant computation time.

	4 processes	8 processes	16 processes
n=256			
n=512			

- Compare the above table with the table you got from lab#3.