4.3 Non-blocking point-to-point operations

1. Irregular data exchange, the numbers of send and receive of a processor do not need to match. “these are the expected incoming messages”, without having to wait for them in sequence; declare the outgoing messages without having to do them in any particular sequence.

2. Non-blocking calls, MPI_Isend, MPI_Irecv.

Non-blocking routines introduce new problems, the actual sending cannot be confirmed so the send buffer may not be safe to overwrite; to send multiple messages with non-blocking calls you have to allocate multiple buffers.

4.4 More about point-to-point communication

1. Message probe, you do not know how big the message is, notice there is a message and use MPI-Get_count to determine the buffer size.

2. Wildcards in the receive call, the status parameter allows you to inspect the message after it was received.

3. Persistent communication

4. Buffered communication.

5. MPI_Request, an opaque pointer, MPI_Request_free.

4.5 Souces

Questions

1. Why does the MPI_Irecv not yield an MPI_Status object? Don’t understand the explanation.

2. How to understand the illustration in figure 4.14?

Exercise

4.5

The communication happening on different nodes takes more time than that on the same node.