# Exercises - Lecture 6

March 11, 2011

**1. Ping pong**

Write a simple MPI program to measure the cost of point-to-point communication, that is, using MPI_Send and MPI_Recv. (Hint, you can involve two MPI processes, between which a message is repeatedly sent forward-and-backward a number of times.)

Use timing call MPI_Wtime() to measure the average time taken for one round of communication, i.e. sending and receiving. Investigate how the time taken varies with the size of the message.

**2. MPI overhead**

From the above exercise you'll notice that point-to-point communication has a constant start-up overhead, independent of the size of the transferred message. This should mean that sending a certain number of small messages is much more expensive than sending one accumulated large message. Write a simple MPI program to verify this.

**3. MPI derived datatypes**

Write a simple MPI program that divides a $2D$ grid into $P$ vertical slices. Each "thin" $2D$ slice is assumed to be extended with one layer of ghost points on the both left and right sides. We also assume that the underlying data structure on each process is a contiguous $1D$ array. Each process is supposed to exchange two boundaries with its left and right neighbors. The whole exchanging procedure is shown in Figure 1. Use MPI derived datatype to define the vertical boundaries for MPI communication.

**4. Wave equation**

On page 26 of this week's lecture slides, you will find details about how to solve $1D$ wave equation in parallel. Make a MPI implementation that should be able to deal with arbitrary number of points correctly. Computing kernel is as follows, where you can set $dt = dx$.

```
t = dt;
while (t<T){
  t += dt;
  for (i=1; i<=M; i++)
    up[i] = 2*u[i]-um[i]+((dt*dt)/(dx*dx))*(u[i-1]-2*u[i]+u[i+1]);

  up[0] = value_of_left_BC(t); // enforcing left BC
  up[M+1] = value_of_right_BC(t); // enforcing right BC

  /* preparation for next time step: shuffle the three arrays */
  tmp = um;  um = u;  u = up;  up = tmp;
}
```

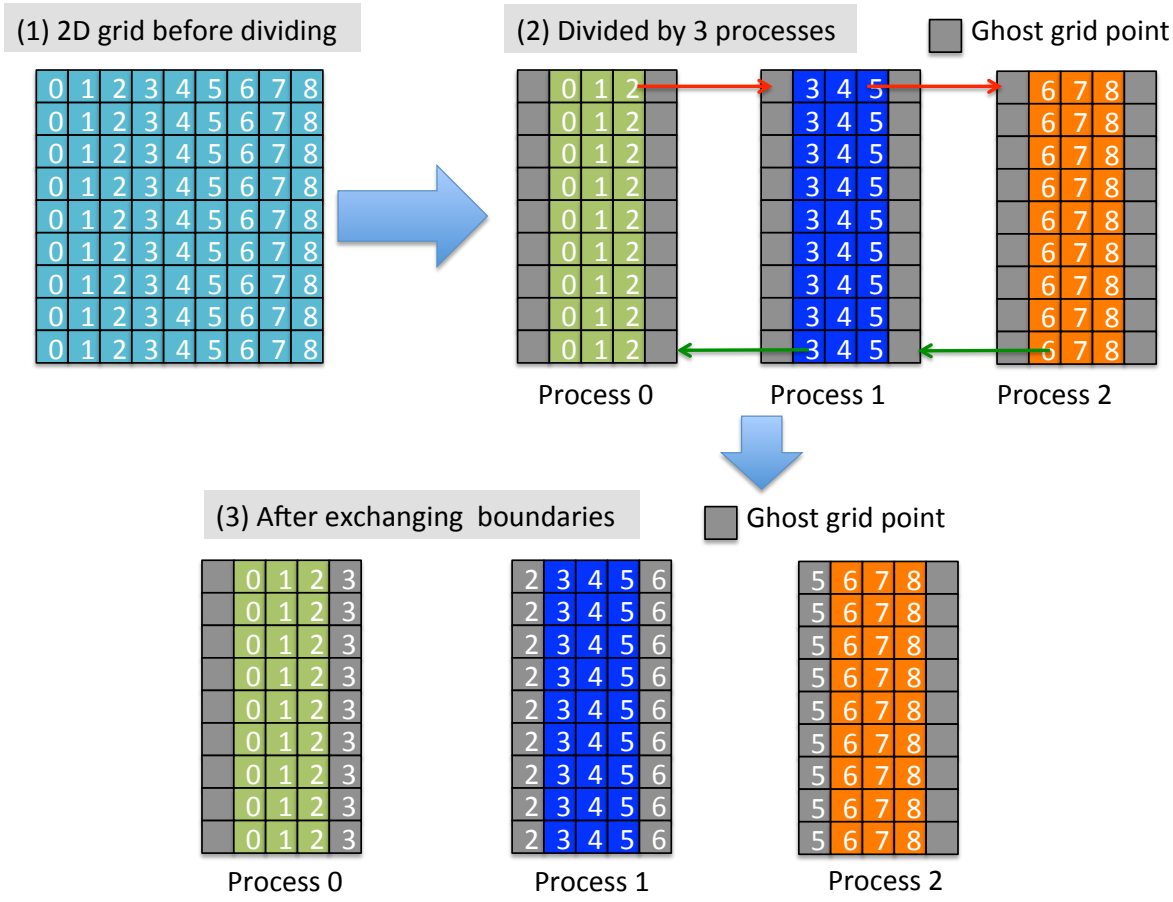


Figure 1: Illustration of exchanging boundaries among three processes.