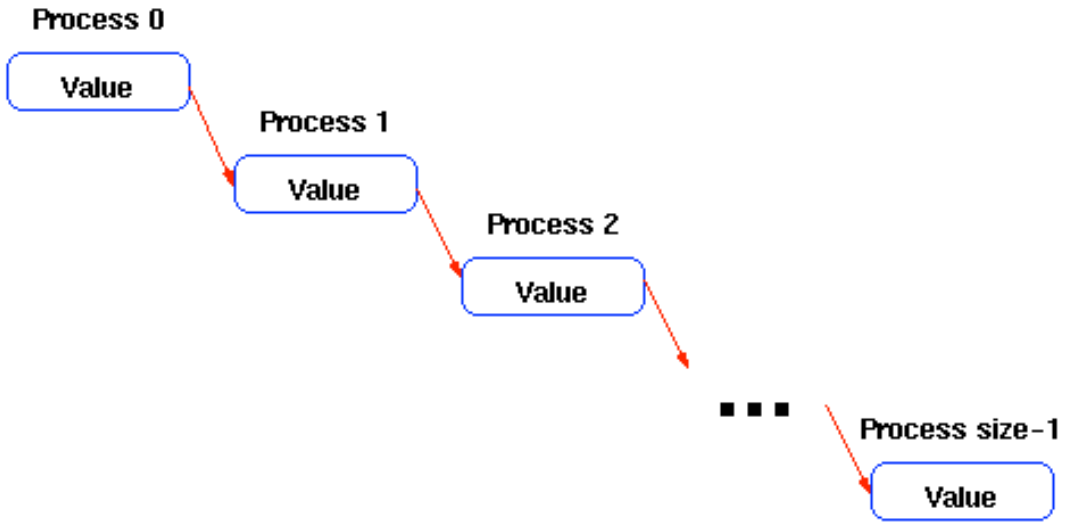
Simple Parallel Data Structures-2
by
William Gropp and Ewing Lusk

1. Exercise: Sending in a ring (broadcast by ring)

Write a program that takes data from process zero and sends it to all of the other processes by sending it in a ring. That is, process i should receive the data and send it to process i+1, until the last process is reached.



Assume that the data consists of a single integer. Process zero reads the data from the user.

You may want to use these MPI routines in your solution:
MPI_Send MPI_Recv

2. Exercise: Using topologies to find neighbours

In the ring example we assume that the "next" process is the one with rank one greater than our rank. That is, process i sends to process i + 1. This may not be the best choice of "next" process, particularly when using a communicator other than MPI_COMM_WORLD. MPI provides topology routines to find a good ordering of processes, particularly for simple linear orderings such as needed here. The assignment is to replace the use of "rank+1" and "rank-1" (where rank refers to the rank in MPI_COMM_WORLD of the calling process) with values computed using MPI_Cart_shift.

You may want to use these MPI routines in your solution:
MPI_Cart_create MPI_Cart_shift