

PHYS405
Advanced Computational Physics
Parallel Computing

Assignment # 4
Due: Friday, October 23, 2009

Purpose: Learn how to implement a truly parallel algorithm.

Note: Please identify all your work.

This assignment consists in building your own broadcast code based on a binary tree. The code `print_tree.c` (see web pages under *Global Operations*) provides a simple implementation of a binary tree based on node zero as being the root. It generates the node numbers from which any node is to receive the information and to which to send this information.

Part A

Write a function `My_Bcast()` that performs a broadcast similar as the one provided by MPICH2. It should follow the same syntax as `MPI_Bcast()`, namely

*int My_Bcast(void *buffer, int count, MPI_Datatype datatype, int root, MPI_Comm comm)*

This function should be based on the skeleton code `print_tree.c`. It should therefore assume that `root` is process zero (0). If not, it should print out an error message quoting this fact and return an exit code one (1).

Test carefully your routine for various process numbers and different `count` and `datatype` in the call.

Part B

Instrument your code for timing measurements. Modify your main code to broadcast the information three (3) different ways: 1 - sequentially via a `for` loop, 2 - via your `My_Bcast()` and 3 - via the MPICH2 `MPI_Bcast()`. Broadcast 10,000 integers 100 times and take an average over the 100 broadcasts to smooth out the time fluctuations. Repeat for 10,000 *double*. Tabulate your timings and comment on the results.

Part C

Modify `My_Bcast()` to be able to broadcast from any process (the root process) within the virtual machine. Check your code carefully for different `root` processes and various number of processes.