

Message passing and MPI

Assignments and problems
Graham Fagg
CS-594 Spring 2005

Rules

- Two weeks to complete assignments and problems. Hand in by midday 23rd Feb 2005.
- If you have problems and do not ask for help (until 10 minutes before the deadline) beware!
- Hand-in written work either on paper or Email to me fagg@cs.utk.edu
- Code is to be tarred with makefiles, output etc and a MD5 signature sent to me.
 - Md5 mywork.tar |& mail fagg@cs.utk.edu
 - Code is to run on hydra or torc machines. If I cannot verify it by remaking it and running it then I will assume it does not!
 - Broken code with comments gets more points than non working non commented code.
 - A short description of design is always needed.

Part A Correctness and buffering?

- Proc 0 Proc 1
- MPI_Send (data,size.. 1) MPI_Send (data,size.. 0..)
- MPI_Recv (indata,insize..1..) MPI_Recv (indata, insize.. 0..)

Above is a head to head send. This might or might not work depending on the system, MPI implementation or other factors such as timing.

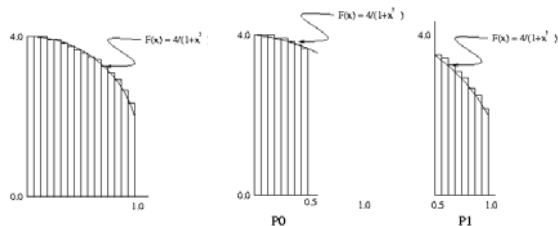
1. Write a paragraph on why the above is an incorrect [non-deterministic] MPI code.
2. Re-write the above code in 3 different ways to make it work (hint, there are 4 simple ways)

Extra under any implementation at what data size does the above break?

Part B Collecting with collectives

1. Write a SPMD code with 1 master and 3 slaves that calculates pi using code handed out in class.
 - Use p2p calls.
 - Master sends out slices
 - Slaves calculate integral slices
 - Master sums to get pi.
2. Change the above to use collective calls
 - Master sends out slices all at once
 - Master receives all slices at once
 - Master sums to get pi.
3. Change 2 so that the master does not have to sum the data explicitly but the collective for receiving the partial results does this.

Part B



Part C

Emulate Shared memory via threads and message passing

Use three process.

One holds an NxM array of numbers and two vectors, one length N and one length M

The first process creates the matrix and clears the vectors, it then enters a loop waiting on a single scalar. It also creates a handler thread that waits on a recv and processes requests.

The other two process uses MPI send/recv calls to perform shmput and shmget operations.

Part C

Emulate Shared memory via threads and message passing

The other process should sum all the columns and rows of the matrix and place these results in the first processes vectors

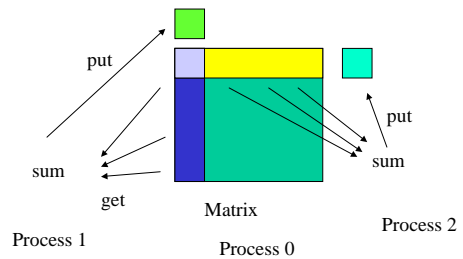
one does rows, one does columns

A put or get operation should contain

I, J (or x,y) the operation (get,put) and the value if needed.

Part C

Emulate Shared memory via threads and message passing



Part C

- Extra marks if you can design something that is very efficient (allows overlapping accesses)
 - Note each get/put is for a single value at a time though.
- Write a page on the design and performance issues as well as any other important details (such as locking or concurrent read/writes)