MATH 700: INTRODUCTION TO PARALLEL COMPUTING USING MPI

MADHU NAYAKKANKUPPAM

HOMEWORK 5

Assigned: **October 23, 2001** Due: **November 06, 2001**

Name: ______________________________
ID #: ______________________________
E–mail: ______________________________

INSTRUCTIONS

- This homework has **two (2) pages**. Make sure you have the entire homework.
- **The first page of your submitted homework must be this page. Staple** all the sheets of your solution on the *top, left corner.* Please save trees by using both sides of the paper, but start every question on a fresh page.
- Leave your completed assignment on the lectern in the classroom *at the start of class on the due date.* **Late homework will generally not be accepted.**

For instructor's use only:
Score: $\boxed{\quad /20}$

#	d	Type	Count	Time (original)	Time (remapped)

Reading exercise (0 points): Read Ch. 7 and 13 of the textbook.

Background work (0 points): Familiarise yourself with the MPI's timing routines (see Appendix of textbook or the MPI Reference Manual) and C's string manipulation routines (see `/usr/include/string.h` and the C bible by Kernighan and Ritchie.)

MPI_Reduce on a hypercube (15 points): Implement `MPI_Reduce` on a general d–dimensional hypercube. Incorporate timing code in your program so that, once the `MPI_Reduce` is completed, the root process can also print out the time taken.

Remapping processes (5 points): Clearly, whatever the communication patterns between processes, it seems logical to have communicating processes on the same node whenever possible. Given that our Beowulf cluster has four dual CPU nodes, incorporate *process remapping* in your original `MPI_Reduce`. Write your code so that you can switch between both versions easily, for example, with a command line argument.

What to turn in: Turn in the following:

1. Using an example (say, a 3–dimensional hypercube), **explain clearly** your implementation of `MPI_Reduce`. This must include details of communication patterns between processes, which Sends/Recvs you used and why.
2. Explain how you implemented *process remapping.*
3. Tabulate your experimental observations as shown below: where # is serial number, d is the dimension of the hypercube, Type is the `MPI_Datatype` you reduced, Count is the number of such elements, and Time, the time taken for the Reduce operation (indicate units, *e.g.* milliseconds). Comment on your experimental observations. (Make sure there are enough rows in the table, and that d and Count are judiciously chosen in order to justify your observations.)
4. Attach a listing of your source code.
5. Store all your files in the directory `xxxx`, where `xxxx` are the last 4 digits of your ID number. Include a README text file in this directory, if you want to add special comments or notes for me (*e.g.* compilation notes). Create a gzipped tar file by executing:

   ```
   % tar cvf xxxx.tar xxxx
   % gzip xxxx.tar
   ```

   Send me the resulting file (`xxxx.tar.gz`) as an e-mail attachment (madhu@math.umbc.edu) before the due date. The subject line should read: *Math 700: HW 5.* Please do not make a submission until you are certain that this is the version you want to submit, and that it tallies with the hard–copy you will hand in. (However, if you must make multiple submissions, only the latest submission will be considered.)