Homework 3

Tom Kelliher, CS 325

30 points, due Mar. 2, 2012

The Assignment

The objective of this assignment is to use Open MPI to parallelize the serial n-body solver accompanying this assignment. Sample data is also provided.

Your solution should work for up to 12 bodies. The root process, and only the root process, will read the data file, initialize the `current` array, and print out the initial state of the system. The root process will then broadcast `n`, the number of bodies, to the communicator, as well as the number of time steps and the time step interval. (Depending upon how you implement the broadcast, this could require multiple broadcasts, because the root process needs to broadcast both `int` and `double` values.) At this point, processes with rank greater than or equal to `n` should terminate, first calling `MPI_Finalize()`.

For each time step, the root will broadcast `current`. Because `current` is an array of `struct`s, the datatype to use in the broadcast should be `MPI_BYTE`, and the count will be `sizeof(current)`. Each process will use `current` to compute the next state position and velocity of one body. Process i computes this information for body i. After the computation is performed, each process sends the next state information for its body to the root process. The root process then updates current from the received information and the next time step begins.

After the final time step, the root process prints out the final state of the system.

Documentation Requirements

Minimally, your program must contain your name and an overview at its top similar to this:

```
/**************************************************************************
 * Tom Kelliher, Goucher College.
 *
 * mandelbrot.c --- Serial program to generate a PPM image
 *                  representation of the Mandelbrot set for some
 *                  rectangular portion of the complex plane.
 *
 * Due to the use of sqrtf(), the math library needs to be compiled-in:
 *
 *    gcc -lm -o mandelbrot mandelbrot.c
 *
 * The PPM image is written to stdout.  It will be HUGE, so it is
 * advisable to pipe the output to pnmtojpeg (on a Linux system)
 * and redirect the filter's output to a file:
 *
 *    ./mandelbrot | pnmtojpeg > image.jpg
```

```
 *
 * The pack()/unpack() routines, unnecessary in a serial environment,
 * _might_ be useful in a message passing environment, but that is a
 * claim that ought to be tested.
 ************************************************************************/
```

If the program you submit for a grade does not work, I will need further documentation within the program itself in order to assign partial credit. (You risk receiving **no** partial credit if you do not provide sufficient documentation.) One component of this documentation should be an analysis, to the best of your ability, of why the program is not working.

Compiling and Running Your Program

Due to the use of `sqrt()`, the math library will need to be linked into your program:

```
mpicc -lm -o parNbody parNbody.c
```

You will also need to include the math library's header file, `math.h`. The program will expect a command line argument corresponding to the name of the data file:

```
mpirun parNbody nbodyData
```

The format of this data file is documented in the serial n-body program. Here's an example of a complete data file:

```
6 10000000 0.01
1.0E4 0.0 0.0 0.0 0.0
1.0E4 100.0 0.0 0.0 0.0
1.0E4 -100.0 0.0 0.0 0.0
1.0E4 0.0 100.0 0.0 0.0
1.0E4 0.0 -100.0 0.0 0.0
1.0E4 100.0 100.0 0.0 0.0
```

Seeking Assistance

I strongly encourage you to begin this assignment as soon as it is issued and to come to my office if you need assistance. If you email me to describe a problem, describe the problem carefully and attach your source code. If extensive debugging will be required to determine the cause of the problem, I will let you know that you will need to visit me in person to resolve the problem. (In other words, I will not debug your program for you.)

Submitting Your Assignment

By the start of class on the 2nd, send me your source code. Your parallel n-body program should produce **exactly** the same output as the serial n-body program. You should anticipate that I will test your program with several different data files. Debugging messages in your source code should be commented-out. Late work will be penalized 15% per day, with the weekend counting as one day. (An unforeseen circumstance preventing you from finishing an assignment on time, while rare, may warrant a request for a deadline extension. Any such request must be made in writing, state the length of the extension requested, explain the nature of the unforeseen circumstance, include a strong justification for the requested extension, and be made at least 24 hours in advance of the assignment deadline. The strength of your justification and the nature of your unforeseen circumstance will determine whether or not the request is granted.)