

CSI 702 FALL 2007 FINAL Exam

The test is closed book, closed notes, no web access. Answers should be brief and concise. Please use additional paper to answer these questions, and write your name at the top of each page. **Good luck!**

Honor Code Certification

Name :

Time started:

Time completed:

I certify that I have abided by the GMU honor code in taking this examination. The work on this exam is my own. I have received no assistance from other persons in completing this exam. I have not consulted any sources of information other than a calculator.

Signature:

1. (30 pts) Parallel Programming Complexity and Efficiency

Write a short one page essay to answer the following questions.

- (a) One of the MPI assignments involved creating an N-body code. This project had the most impressive speedup of any of the homework assignments. Explain the reasons why this code is well suited toward parallelism. (There are several key reasons, so be brief but complete.)
- (b) One of the MPI assignments involved implementing a simple 1-D diffusion equation. The speedups on this project were terrible. In some cases, the code ran much slower on 4 machines than on one. Explain why this code scaled so poorly and how you might make this type of code run more efficiently on a Beowulf cluster.
- (c) Another MPI assignment involved creating a parallel sort. Most students that tried this had a very difficult time making this program work at all. When it was working, it often didn't scale very well. Explain what makes this algorithm more complex and difficult to make scale across many CPUs.

2. (20 pts) The following code segment calculates forces, potentials, and kinetic energies for particles in a molecular dynamics simulation. Make notations on the code with the appropriate OpenMP extensions so this code executes efficiently and correctly on a

multi-core computer. Be as explicit as possible. Can you make this code run efficiently using OpenMP for a dual core machine? How could changing the blocking affect your conclusions?

```
! from http://www.openmp.org/drupal/samples/md.html

do i=1,np
  ! compute potential energy and forces
  f(1:nd,i) = 0.0

  do j=1,np

    if (i .ne. j) then

      call dist(nd,box,pos(1,i),pos(1,j),rij,d)

      ! attribute half of the potential energy to particle 'j'
      pot = pot + 0.5*v(d)

      do k=1,nd

        f(k,i) = f(k,i) - rij(k)*dv(d)/d

      enddo

    endif

  enddo

  ! compute kinetic energy
  kin = kin + dotr8(nd,vel(1,i),vel(1,i))

enddo
```

3. (15 pts) Optimization

- (a) (5 pts) One of the projects presented in class involved simulating a colliding galaxy. In this project, they were able to get run of a given size to do one step in about one hours using ten nodes. Their code scaled nearly linearly with the number of processors. In a code I use, I can do the same problem on a serial computer at a speed of one step every 3 minutes. How can one code run so much more efficiently than the other?
- (b) (5 pts) List five different types of "code tuning" that can be done to make a serial code run faster.
- (c) (5 pts) Explain why we don't optimize code before we profile.

4. (15 pts) Compare the following HPC hardware solutions. Explain the advantages, disadvantages and limitations to each approach. What are the advantages and disadvantages of using special purpose hardware for High Performance computing?

- (a) FPGA (Fully Programmable Gate Arrays)
- (b) GPGU (General Purpose Graphics Units)
- (c) The GRAPE project

5. (20 pts) Computational Requirements

A single threaded code you use to calculate a given evolutionary sequence takes approximately 12 hours to run on a 2.4 GHz Pentium D machine. A typical checkpoint file takes 240 Mb, and there are 30 files per run generated. It may be possible to reduce stored data to 1/3 of a Mb for each the 30 timesteps.

You are drawn into a collaboration that requires you to run simulations for a large set of parameter space. This includes

$$\begin{aligned}
 rmin &= \{0.1, 0.25, 0.50, 0.75, 1.0, 1.25, 1.50, 1.75, 2.0, 2.5, 3.0, 4.0, 5.0\} \\
 v/vmax &= \{1, 1.2, 1.5, 1.75, 2.0, 2.5\} \\
 i &= \{-90, -75, -60, -45, -30, -15, 0, 15, 30, 45, 60, 75, 90\} \\
 w &= \{-90, -60, -30, 0, 30, 60, 90\} \\
 mr &= \{0.1, 0.2, 0.33, 0.4, 0.5, 0.66, 0.75, 0.80, 1.0, 1.2, 1.33, 1.5, 2.0, 2.5, 5.0, 10\} \\
 bs &= \{0.2, 0.4, 1.0\}
 \end{aligned}$$

(1)

- (a) (5 pts) Assuming this is a numerically intensive code, estimate how many floating point calculations are done in a typical run.
- (b) (5 pts) Calculate the total number of runs, the total computational time, and the storage space required for these runs.
- (c) (5 pts) Is this project well suited for grids, clusters or other hardware? What changes would be needed to make this run effectively on the "best" platform?
- (d) (5 pts) Based on your experience in this class, why is parallel programming harder than programming in serial? If this is true, why (and when) is parallel programming worth the effort? What changes in the future might make this easier for scientists?