Computing the Universe '99: Parallel Programming

July 1-5, 1999

1. Numerical Integration

   Write a parallel implementation of a program to perform a numerical integration using the trapezoid rule. Use it to evaluate the integral in problem 2 of Exercise 2 for the ($\Omega_m = .4, \Omega_\Lambda = 0$) model. Compare the parallel results with your serial results for correctness.

2. Direct N-body force calculation

   The attached *C* program (taken from *Using MPI* by Gropp, Lusk, and Skjellum) is an $O(N^2)$ algorithm for computing the forces on a collection of particles. To compute all the interactions, it copies the particles around in a ring. As it stands, the program initializes the particles using a random number generator, and only prints out forces for particles on one processor. (Also note that the program requires the number of particles to be evenly divided by the number of processors.)

   (a) Modify this program so that it reads particle data from an ASCII file of the format:

   ```
   NParticles
   Mass1 X1 Y1 Z1
   Mass2 X2 Y2 Z2
   Mass3 X3 Y3 Z3
   ...
   ```

   And writes the calculated forces to a new file in the format:

   ```
   NParticles
   ```

```
ForceX1
ForceX2
...
ForceY1
ForceY2
...
ForceZ1
ForceZ2
...
```

(b) Use this program to produce forces for the particles in the attached `k7e4.mxyz` file (a king model). Compare run times and correctness on 1, 2, and 4 processors. (The later will require the cooperation of your group members.) All files can be found in the directory: /hosts/a/home/trq/hw5.

(c) Modify the program so that it can handle different numbers of particles on different processors. Compare the run times on 3 and 6 processors.

Hints: Let one processor do all the file I/O and ship data to and from the other processors. The output file described above can be read by the tipsy `readvector` command. This may be useful for debugging along with the tipsy binary file `k7e4.nat`.

Extra: There are other things that can be done with this program. *e.g.* it could be turned into a full N-body integrator using leap-frog. There are also performance enhancements to be made, such as employing the symmetry of the interparticle forces: $F_{ij} = -F_{ji}$.