Using Parallel Computers for Very Large N-body Simulations : Shell formation using 180K particles

John Salmon
Caltech Concurrent Computing Project
Peter Quinn
Mt. Stromlo and Siding Springs Observatories
Mike Warren
Caltech Concurrent Computing Project

Abstract

We have used a parallel version of the Barnes-Hut C treecode running on a parallel supercomputer (a 512 node NCUBE) at the Caltech Concurrent Computing Project to follow the evolution of systems with very large numbers of particles (N>180,000). Many important problems in astrophysics demand large N and in particular we have studied the fully selfconsistent formation of stellar shells around an elliptical parent galaxy via the merger with another elliptical of one tenth its mass. The final system shows a spread in shell radii similar to that in observed shell galaxies. Major changes in the structure and dynamics of the primary have resulted from the merger.

1 Large N problems : the motivation

Traditionally, N-body simulations of collisionless systems have been run with the largest number of particles possible given the limitations on machine time and memory. This push to large N was motivated by the need to control the collisionality of the simulation over the desired number of dynamical times. The arrival of methods with computational times that scaled like $NLogN$ instead of N^2 (PM, PPPM, Treecodes) meant that for a fixed computing resource,10,000 - 100,000 particles could be evolved with high dynamical resolution and in runs of only a few tens of hours on machines like the CRAY 1. In trying to achieve very large N, most of the NlogN codes run up against serious memory constraints on machines like the CRAY X-MP . In an attempt to address this memory problem we have developed the Barnes-Hut treecode on a parallel architecture in which individual nodes have substantial amounts of memory. The 512 node NCUBE at the Caltech Concurrent Computing Project has 512K of memory per node giving a total machine memory of 256 MgB which is more than ten times the typical maximum process size on the X-MP.

2 Tree Codes on Parallel Computers

The basic problem in designing a parallel algorithm is the decision of "who gets what data, when?" Given an appropriate organization of the data, termed a "decomposition", one is usually able to carry over the basic sequential algorithm with little or no change. That is, each processor behaves as though it is performing the original sequential algorithm on a subset of the complete data set.

The simplest decomposition to consider is one in which each processor is assigned all the bodies in a region of space. The processor boundaries are chosen to assign approximately equal work loads to the processors. Given a decomposition of the particles which balances the load, it is tempting to think we can construct the tree on each processor, merge them into a collective tree which is copied to each processor and proceed with the force evaluation. Unfortunately, this leads to severe problems with the available memory. One is limited to treating problems whose entire tree fits into the 512 kbytes available to each processor. Thus, this approach provides extremely good utilization of the parallel processor when the total number of bodies is less than a few thousand, but it doesn't work at all for large numbers of bodies.

At this point, it is crucial to realize that by the very nature of the Barnes-Hut (1) algorithm, it is not necessary to store the entire tree in each processor. Since each processor is concerned with computing forces only within a restricted region of space, it will need to access the most detailed levels of the tree only in a restricted region. Distant regions of space may be accounted for with much less data.

Thus, the parallel force calculation algorithm proceeds as follows :

- Determine processor boundaries by recursive bisection of the estimated work.
- Build the tree for the local particles.
- Communicate to obtain only those details of other processors' trees which may be necessary for force computations.
- For each local particle descend the tree and compute the force.

The performance of the parallel algorithm may be measured several ways. Of primary interest is the overall performance. Counting all of the overheads, load imbalances, i/o operations, etc., the system computes the force on our 180,000 bodies in approximately 200 seconds ($\theta = 0.8$,monopole). In addition, it is of interest how this figure scales with the number of processors. Obviously, we were not able to evaluate the forces on a 180,000 body system on a single processor with 512 kbytes of memory. Nevertheless, extrapolation from a series of runs with fewer particles indicates that we obtained a speedup of 300 on 512 processors. Figure 1 shows the speedup resulting from the use of parallel processing as a function of the number of processors, as well as the number of bodies. It is typical of such plots that for a fixed problem size, the speedup begins to level off and the addition of more processors does little to reduce the overall running time. On the other hand, if the problem size is allowed to grow with the number of processors, the speedup continues to increase up to very large numbers of processors. Thus, large parallel computers are generally well matched to large problems, but are a poor match to small problems.

3 Shell Formation

Since the early 1980s, stellar shells around ellipticals have been recognized as the product of a collision between a large elliptical and a compact and/or dynamically cold companion galaxy. Models of shell formation by (2; 3; 4) have all used rigid potential models for the primary and hence have ignored important selfconsistent effects like dynamical friction and tidal stripping. As a result several important features of shells galaxies, like the small radii of the innermost shells, have remained unexplained. In order to address important dynamical effects such as the orbit decay and progressive tidal distruction of the companion as well as the selfconsistent effects on the primary, we have undertaken a study of shell formation using the treecode on the NCUBE.

Figure 2 shows the distribution of companion particles near the end of the simulation. The core of the companion stayed intact for five pericentre passages and was destroyed in the final pass. The following is a summary of the main points of the simulation with regard to shell production :

- Shell making particles are removed from the companion after each close encounter with the core of the primary. As a consequence the final shell distribution is complex, consisting of several overlapping shell systems.
- Since the companion core continues to lose orbital energy over the course of the simulation, the final shells are confined to smaller radii than the shells that resulted from earlier pericentre passages . The innermost shells are within 0.3 effective radii and the outer shells extend to more than 20 effective radii which gives a shell radius range similar to many observed shell galaxies.
- Because of the complex nature of the multiple shell system, it is very difficult in this case to apply the techniques outlined by Hernquist and Quinn(5) to probe the underlying potential using the distribution of shell radii.

4 Conclusion

Many problems in galaxy formation and evolution will require the use of very large numbers of particles in N-body simulations. The model of shell formation we ran has addressed a number of outstanding issues in shell formation; in particular the radius of the inner most shells. The simulation has also given us valuable new insights into the effects shell making mergers have on the primary galaxies. These effects may explain many generic features of ellipticals (photometric and kinematic axis alignment, rotation curve shape, see Quinn and Zurek, this volume). The simulation also teaches us about the complex process of tidal disruption and the dynamics of mergers. By comparing this simulation with others on the formation of counterrotating cores by Balcells and Quinn(6) and the cosmological simulations of Quinn et al.(7), we can start to see common processes emerge that may tell us a great deal about the way mergers work in general and how they may have influenced the properties of galaxies along the Hubble sequence.

Acknowledgements

We would like to thank the Caltech Concurrent Computing Project for providing travel funds and CPU time on the nCUBE. This work was supported in part by Department of Energy, Applied Mathematical Sciences grant DE-FG03-85ER25009.

References

J. Barnes and P. Hut, “A hierarchical O(NlogN) force-calculation algorithm,” *Nature*, vol. 324, p. 446, 1986.

P. J. Quinn, “On the formation and dynamics of shells around elliptical galaxies,” *Ap. J.*, vol. 279, p. 596, 1984.

C. Dupraz. and F. Combes *Astr. & Ap.*, vol. 166, p. 53, 1986.

L. Hernquist and P. J. Quinn, “Formation of shell galaxies: I. spherical potentials,” *Ap. J.*, vol. 331, p. 682, 1988.

L. Hernquist and P. J. Quinn, “Shells and dark matter in elliptical galaxies,” *Ap. J.*, vol. 312, p. 1, 1986.

M. Balcells and P. J. Quinn, “The formation of counterrotating cores in elliptical galaxies,” *Ap. J.*, vol. 361, p. 381, 1990.

P. J. Quinn, J. K. Salmon, and W. H. Zurek, “Primordial density fluctuations and the structure of galactic haloes,” *Nature*, vol. 322, p. 329, 1986.