Astrophysical N-body simulations on GRAPE-4 Special-Purpose Computer

Junichiro Makino* and Makoto Taiji
*Department of Information Science and Graphics,
Department of Earth Science and Astronomy,
College of Arts and Sciences, University of Tokyo,
Tokyo 153, Japan
Email: makino@chianti.c.u-tokyo.ac.jp

Abstract

We report on resent astrophysical N-body simulations performed on the GRAPE-4 (Gravity Pipe 4) system, a special-purpose computer for astrophysical N-body simulations. We first review the astrophysical motivation, the algorithm, the structure of the GRAPE system, and the actual performance. The GRAPE-4 system consists of 1692 pipeline processors. The peak speed of one pipeline processor is 523 Mflops and that of the total system is 884 Gflops. The performance obtained is 529 Gflops for the simulation of two massive black holes in the core of a galaxy with 700,000 stars,

1 Introduction

Many astronomical objects are well approximated as the system of self-gravitating particles. The solar system, open clusters, globular clusters, galaxies, clusters of galaxies are all expressed as the system of N particles interacting only through gravity.

The N-body simulation technique is a most useful tool for the study of self-gravitating N-body systems. The calculation cost of N-body simulation increases rapidly as we increase the number of particles. The reason is that the gravity is a long-range attractive force. We cannot neglect the contributions of distant particles to the force on a particle. In many other particle simulations, the force is of short-range nature, as in the case of the van der Waals force, or the contributions of the distant particles can be neglected.

In order to accelerate astrophysical N-body simulation we developed a series of special-purpose hardwares for the force calculation [18]. The

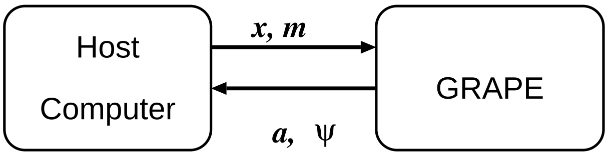

Figure 1: Basic structure of a GRAPE system

basic idea is shown in figure 1. The special-purpose hardware calculates the interaction between particles. The host computer (a workstation) performs all other calculations and I/O. We call the special-purpose hardware GRAPE (GRAvity PipE), since its main computing unit is the hardwired pipeline to calculate the gravitational force between particles.

In this paper, we report on a large astrophysical N-body simulation performed on GRAPE-4, a massively parallel GRAPE system. Present GRAPE-4 system consists of 1692 pipeline processors. Its peak speed is 884 Gflops. A more detailed description of GRAPE-4 is in [12].

The structure of the present paper is the following. In section 2, we overview the algorithms used to calculate the gravitational force between particles. In section 3, we describe the GRAPE-4 system. In section 4 we report on two very large simulations recently finished/currently running on GRAPE-4 system.

2 Astrophysical N-body Simulations

As we stated in the introduction, many astrophysical systems are well expressed as the system of N particles interacting only through gravity. However, this fact does not necessarily means we can use single algorithm to study various systems. In the following we briefly describe several examples of astrophysical N-body systems and numerical methods for them.

2.1 The Solar System

The long-term stability of the solar system is a very old, but still not completely solved, problem. Recent progress in the understanding of the stability comes mainly from very long and accurate direct integration of the orbit of outer planets. In particular, Sussman and Wisdom [19] followed the orbits of five planets for 850 M years and found that the Lyapnov number for the orbit of Pluto is positive. Their calculation was performed on the Digital Orrery [3], a special-purpose computer for the solar system. In this problem, the number planets is small and the orbital timescales of planets are not widely different. The accuracy must be very high since the simulation covers tens of millions of orbits. Thus, high-order integrators with a fixed stepsize is used and forces between planets are directly calculated.

It is very difficult to run this kind of simulations efficiently on vector processors or MPPs, since the degree of parallelism is small. After the Digital Orrery, most simulations have been performed on single-processor RISC workstations.

2.2 Cosmological Simulations

The large scale structures of the universe is believed to be formed through the gravitational instability of the uniformly expanding universe. The simulation of this process is necessary to compare the theories and observations. In this type of simulation, the number of particles must be very large to represent the smooth matter distribution. The duration of the simulation, measured by the dynamical timescale of the system, is rather short, since the dynamical timescale of an expanding universe is its age itself. As a result, total number of timesteps is rather small (less than 1,000). We do not need very high accuracy in either the force calculation or the orbit integration.

As far as we are mainly interested in the large scale structure, we can use one common timestep for all particles. If we are interested in the evolution of smaller subsystems (*e.g.*, formation of galaxies), however, the timesteps for particles in high-density region must be much smaller than that for other particles.

Since the required accuracy is not very high, it is possible to apply fast potential solvers such as the particle-mesh (PM) scheme, particle-particle particle-mesh (P^3M) scheme (see, *e.g.*, [9]), the Barnes-Hut tree algorithm ([4], [5], [10], [14]) or the fast multipole algorithm(FMM) [8], [6]. Calculation cost of these schemes are $O(N \log N)$, except for the FMM scheme whose cost is $O(N)$. PM and P^3M schemes are most widely used. BH tree is used by several researchers. To our knowledge, FMM has not been used for astrophysical simulations yet.

It is easy to achieve reasonable performance for PM and P^3M schemes on vector processors or MPP. Salmon and warren developed a high-performance implementations of tree algorithm on MPP. They reported simulations using $> 10^7$ particles on Touchstone Delta.

Simulations of two interacting galaxies and groups or clusters galaxies do not need very high accuracy, since the orbital timescale of a galaxy (10^8 years) is not very small compared to the age of the universe (10^{10} years). For these systems, the tree algorithm on workstations or conventional vector machines is mainly used.

2.3 Globular clusters and galactic nuclei

Globular clusters are collection of 10^5-10^6 stars orbiting around each other. Our galaxy has ~ 100 such objects. Both in view of the number of the stars and the orbital timescale, the globular clusters lie in between the solar system and the universe. The typical stars have the orbital timescale of 10^6 years, but the stars in the central region have much shorter timescales (10^4 years or less depending on the evolutionary stage). Thus, it is crucial to allow different particles to have different timesteps. The individual timestep algorithm [1] has been widely used in this field.

In the individual timestep scheme, each particle has it own timestep Δt_i and maintains its own time t_i. To integrate the system, one first selects the particle for which the next time $(t_i + \Delta t_i)$ is the minimum. Then, one predicts its position at this new time. Positions of all other particles at this time must be predicted also. Then the force on that particle from other particles is calculated following Newton's law of gravity. The position and velocity of the particle is then corrected.

This individual timestep scheme is an extremely powerful method when applied to the late stage of the evolution of star clusters. The gain in the computational speed can be $O(N)$ for many systems. This $O(N)$ gain, however, implies that only a small number of particles have small timesteps. It is therefore difficult to apply fast algorithms such as the FMM scheme, since these schemes are fast only when the forces on all particles are calculated at each timestep. McMillan and Aarseth [13] implemented high-accuracy treecode with individual timestep. For practical number of particles, however, this code turned out to be slower than the direct method on 1-processor Cray YMP.

The implementation of the individual timestep algorithm on massively parallel processors is nontrivial, since the communication between nodes must be very fast in order to use large number of processors efficiently. Spurzem [16] implemented individual timestep algorithm on a Cray T3D. His implementation requires that each processor has complete copy of the total system and therefore is not scalable to large number of processors. In addition, it requires a fast broadcast mechanism, which also limits the scalability of the algorithm.

The largest number of particle reported in the literature for the simulation of long-term evolution of a globular cluster is 10,000 [17]. This calculation took two CPU months of 1-processor Cray YMP.

In the above calculation, the Ahmad-Cohen [2] neighbor scheme (henceforth referred to as the AC scheme) is used. In this scheme, each particle has maintain the list of its neighbor particles, in the way similar to the neighbor list scheme in molecular dynamics simulations. This list is updated in the timestep longer than the timestep for the integration of orbit. At each timestep, only the forces from the particles in the list are calculated. The forces from the rest of the system is calculated only when the list is updated. This algorithm has the theoretical gain in the computational cost of $N^{1/4}$, compared with the standard algorithm. In practice, for 10,000 particles the AC scheme requires about 10 times less number of floating point operations compared to the standard scheme.

We recently finished the simulation of globular cluster with 32,768 particles, using the 1/4 of the GRAPE-4 system. The calculation cost scales roughly as $N^{3.2}$, even with the AC scheme. Thus, our calculation requires about 50 times more computer power than reported in [17].

The evolution of the central region (nucleus) of galaxies is in many ways similar to that of globular clusters, but is more complex. The globular cluster is for the first-order approximation a correction of point mass particles. The physical mechanisms that drive the evolution of the galactic center includes star formation, physical interaction between stars and between stars and massive black holes.

The structure of the galactic nucleus containing massive black holes recently attracted the attention of many researchers, since the result of the Hubble Space Telescope indicates that the structure of the nuclei of many elliptical galaxies cannot be described either with the model based on distribution of normal stars or the model where one black hole has grown by eating the stars. In addition, it is highly likely that many ellipticals contain more than one black holes in the center, since certain fraction ellipticals are believed to be formed through merging of two galaxies.

Thus, simulation of the galactic nucleus containing two or more black holes is necessary to understand the structure and evolution of the elliptical galaxies. This simulation requires the individual timestep algorithm and very accurate orbit integration.

The largest number of particle reported in the literature for the simulation of galaxy with two black holes is 32770 [7], which used a BH tree algorithm with constant stepsize. They did not follow the late stage of evolution where two black holes form a binary system. We have done several simulations with 16386 particles, using GRAPE-3 [11].

For simulation of a galaxy with single central black hole, the extension of SCF method has been used [15]. The extension of this method to handle more than one massive object seems to be difficult.

3 The GRAPE-4 system

3.1 Architecture

GRAPE-4 is designed to run the individual timestep algorithm with very high speed. The GRAPE-4 system consists of a host computer and four clusters. One cluster has one host-interface board (HIB), one control board (CB), and 9 processor boards (PB). HIB and CB handles the communication between the processor boards and the host. The processor boards perform the force calculation. Each processor board houses 48 HARP (Hermite AccceleRator Pipeline) chips, which are custom LSI chips to calculate the gravitational force and its first time derivative. The theoretical peak speed of GRAPE-4 is 854 Gflops, for the clock frequency of 32 MHz.

3.2 Parallelism and calculation flow

The total number of pipeline chips of GRAPE-4 is 1672. Each pipeline calculates the forces on two particles, using the technique we call "virtual multiple pipeline". A simplest algorithm to use this number of chip is to let them calculate forces on 3344 different particles. With this approach, however, the degree of parallelism is too large to run the individual timestep algorithm efficiently, unless the number of particles is unpractically large. Therefore, we used a different approach: Each of the 36 processor boards calculates the forces on same 96 particles, but from different set of particles. Thus, the total force on a particle is computed as 36 partial sums. The summation within a cluster is performed on CB and summation of contributions from different clusters is performed on the host computer.

3.3 Processor board

A processor board consists of the particle data memory, one PROMETHEUS LSI chip and 48 HARP LSI chips. The particle data memory stores the data of particles which exert the force. The PROMETHEUS LSI is used to calculate the position (and velocity) of particles at a specified time. This function is necessary to use the individual timestep. The HARP LSI chips calculate the gravitational accelerations and their first time derivatives for particles. One board has 48 HARP chips. 8 HARP chips are packaged in one custom MCM package. One PB board houses 6 MCMs. We use 47 out of 48 chips for actual calculation in order to utilize MCMs with one defect chips. Each HARP chip calculates the forces on two different particles simultaneously. So one board calculates the forces on 94 particles in parallel.

3.4 Hardware cost

The total cost of GRAPE-4 is 150 M JYE, of which 80 M JYE is spent for the production of the hardware. Remaining 70 M JYE is spent for the designing (including the cost of design software and workstations to run the software). The cost per Gflops is 0.17 M JYE, which is, with the present exchange rate of 1 dollar = 100 JYE, 1700 dollars.

4 Performance

We report the performance statistics for a recent simulation on GRAPE-4. The performance numbers are based on the wall-clock time obtained from UNIX system timer on the host computer (a DEC AXP 3900). The total number of floating point operations is calculated as $49Nn$, where N is the number of particles and n is the number of individual steps. Each pairwise interaction costs 49 flops, using the Livermore Loops prescription of 1 square root = 1 division = 4 flops.

We simulated the evolution of the galactic nucleus containing massive black hole pair. The galaxy is modeled with 700,000 equal-mass particles. The black holes are modeled as two point-mass particles with mass 2.2×10^4 times larger than that of other particles. The relativistic effect was negligible in our simulation.

We performed the simulation for 15.8 dynamical time units, for which the number of individual steps was 2.505×10^9. The whole simulation, including on-the-fly analysis of the orbital elements of black holes and file operations, took 45.15 hours, resulting in the average computing speed of 529 Gflops. A full-size GRAPE-4 with 4 clusters (peak speed 854 Gflops) was used for this calculation. The overall efficiency is about 60 %.

This simulation uses 43 times more particles compared to the simulation previously reported [11]. This difference makes it possible to follow the evolution of the orbital element of the pair black hole far more accurately. The evolution of the eccentricity, which was too noisy in 16k runs, is now clearly observed.

5 Comparison with other calculations

As far as the computing speed measured in the Gflops rate is concerned, the performance we obtained is the fastest, since the measured speed of 528 Gflops is faster than the theoretical peak speed of any existing computer (as of September 1995). Thus, as far as the same algorithm is used, our calculation on GRAPE-4 is faster than any other calculation.

When we compare the above performance with the performance of other computers, however, we must take into account the fact that the algorithm we used is not the most efficient one, at least in terms of the number of floating point operations. As we stated in section 2.3, the AC scheme would, theoretically, requires about 25 times less number of floating point operations for a 700k particle simulations. Thus, if someone can achieve the sustained speed of 20 Gflops for the AC scheme, the effective speed would be the same. Such an implementation currently does not exist. However, whether such an implementation is possible or not is an interesting question.

In the following, we will make rather optimistic estimate for the performance of the AC scheme on present vector-parallel computers and massively parallel computers.

A single-processor YMP achieves about 100 Mflops for the AC scheme. This is ~30% of the peak performance. If we scale this ratio to a Cray T932 (which is not shipped yet), we get 20 Gflops. Thus, when a T932 is actually delivered, it might be able to run the simulation at roughly the same speed as the present GRAPE-4 system.

In practice, the performance of a T932 is likely to be somewhat lower, because of the following three reasons. In order to achieve the speedup factor of 25, the AC scheme requires $O(N^{1.75})$ memory, which is about 200GB for 700k particles. The largest memory available on T932 is 8GB. This difference causes rather large loss in the performance (a factor of 2 or more). The other reason is that the efficiency of T932 is likely to be lower than that of YMP, since the relative memory bandwidth is a factor of 2 smaller for T932. Finally, to expect a result of single-processor run to be scalable to 32 processors is far too optimistic. Thus, We believe it is reasonable to estimate the performance of T932 to be in the 5–10 Gflops range.

Now we consider MPPs. The implementation by Spurzem [16] achieved the efficiency of about 10%. Thus, if we scale this result to a 1024-processor T3D, we get about 15 Gflops, which is close to 20 Gflops. However, Spurzem's implementation requires the memory of $O(N^{1.75}n)$, where n is the number of processors. Thus it is not practical at all. A practical implementation of the AC scheme would require significantly more communication, resulting in the loss in the efficiency. It is shown from simple argument that the communication would limit the total performance if $Rn > 10N^{5/12}$, where R is the ratio between the calculation speed (in flops) and the broadcast bandwidth (in words/sec). If we assume $R = 2$ and $n = 1024$, for 700k particles the communication would still nearly as expensive as the calculation itself.

6 Acknowledgements

We are grateful to Daiichiro Sugimoto and Toshikazu Ebisuzaki who have been leading the GRAPE project from the start. We thank Piet Hut for many invaluable discussions concerning the software and computational cost of N-body simulations. We also thank Toshiyuki Fukushige for helping the development of the GRAPE-4 system. This work was supported by the Grant-in-aid for Specially Promoted Research (04102002) of the Ministry of Education, Science, and Culture.

References

[1] S. J. Aarseth, *Monthly Notices Roy. Astron. Soc.*, Vol. 126, p. 223, 1963.

[2] A. Ahmad and L. Cohen, J. Comput. Phys., Vol 12, p. 389. 1973

[3] J. H. Applegate, M. R. Douglas, Y. Güsel, P. Hunter, C. L. Seitz, and G. J. Sussman, *IEEE Transactions on Computers*, Vol C34, p 822, 1985.

[4] J. E. Barnes and P. Hut, *Nature* Vol. 324, p. 446, 1986.

[5] J. E. Barnes, *J. Comp. Phys.*, Vol. 87, p. 161, 1990.

[6] J. A. Board, Z. S. Hakura, W. D. Elliott and W. T. Rankin, in Proceedings of the 7th SIAM Conference on Parallel Processing for Scientific Computing, et. D. H. Bailey et al, SIAM, 1995, p 295.

[7] F. Governato, M. Colpi and L. Maraschi, preprint, 1994.

[8] L. Greengard, The rapid evaluation of potential field in particle systems, MIT Press, 1988.

[9] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, Adam Hilger, New York, 1988.

[10] J. Makino, *J. Comp. Phys*, Vol. 87, p. 148, 1990.

[11] J. Makino, T. Fukushige, S. K. Okumura and T. Ebisuzaki, *Publ. Astron. Soc. J.*, Vol. p. 45, 303, 1993.

[12] J. Makino, M. Taiji, T. Ebisuzaki and D. Sugimoto, in *Proceedings of Supercomputing '94*, IEEE Computer Society Press, Los Alamitos, 1994, pp 429-438.

[13] S. L. W. McMillan and S. J. Aarseth, *Astrophys. J.*, Vol. 414, p. 200, 1993.

[14] J. K. Salmon, M. P. Warren and G. S. Winckelmans, Intl. J. Supercomputer Appl. Vol 8, p 129, 1994.

[15] S. Sigrudsson, L. Hernquist and G. D. Quinlan, Astrophys. J., Vol 446, p 75.

[16] R. Spurzem, A parallel implementation of an Aarseth N-body integrator on the CRAY T3D, in preparation.

[17] R. Spurzem and S. Aarseth, in preparation.

[18] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki and M. Umemura, *Nature*, Vol. 345, p. 33, 1990.

[19] G. J. Sussman and J. Wisdom, Science, Vol 241, p. 433, 1988.

Biographical Sketches

Junichiro Makino received the B.S., M.S., and Ph.D. degrees in systems science from University of Tokyo in 1985, 1987, and 1990, respectively. From 1990 to 1994 he was a research associate at the University of Tokyo. Since 1994 he has been an associate professor in the Department of Information Science and Graphics of College of Arts and Sciences at University of Tokyo. His Email address is makino@chianti.c.u-tokyo.ac.jp

Makoto Taiji received the B.S., M.S., and Ph.D. degrees in physics from University of Tokyo in 1987, 1989, and 1992, respectively. Since 1992 he has been a research associate at the Department of Earth Science and Astronomy of College of Arts and Sciences at University of Tokyo. His Email address is taiji@chianti.c.u-tokyo.ac.jp