A 29.5 Tflops simulation of planetesimals in Uranus-Neptune region on GRAPE-6

Junichiro Makino
Department of Astronomy, School of Science, University of Tokyo,
Tokyo 113-0033, Japan
Email: makino@astron.s.u-tokyo.ac.jp
Eiichiro Kokubo
National Astronomical Observatory of Japan, Mitaka, Tokyo 181-8588, Japan
Toshiyuki Fukushige,
Department of General System Studies, College of Arts and Sciences, University of Tokyo,
Tokyo 153-8902, Japan
and Hiroshi Daisaka
Department of Astronomy, School of Science, University of Tokyo

Abstract

As an entry for the 2002 Gordon Bell performance prize, we report the performance achieved on the GRAPE-6 system for a simulation of the early evolution of the protoplanet-planetesimal system of the Uranus-Neptune region. GRAPE-6 is a special-purpose computer for astrophysical N-body calculations. The present configuration has 2048 custom pipeline chips, each containing six pipeline processors for the calculation of gravitational interactions between particles. Its theoretical peak performance is 63.4 Tflops. The actual performance obtained was 29.5 Tflops, for a simulation of the early evolution of outer Solar system with 1.8 million planetesimals and two massive protoplanets.

0-7695-1524-X/02 $17.00 (c) 2002 IEEE

1 Introduction

In this paper, we report the performance of 2048-chip GRAPE-6 system for the simulation of the protoplanet-planetesimal system of the outer Solar system. The simulated system consists of 1,799,998 planetesimals and two protoplanets. The algorithm used is the block individual timestep algorithm [McM86, Mak91], where each particle has its own time and timesteps. Since the required accuracy for the time integration, and therefore that of the gravitational interaction, are both high, we used direct summation for the force calculation. The GRAPE-6 configuration used consisted of a cluster of 16 Linux PCs (with AMD Athlon XP-1800+ CPUs), each with 4 GRAPE-6 processor boards. Each of the GRAPE-6 processor boards houses 32 processor chips. The PC cluster performs the time integration and GRAPE-6 boards perform the force calculation. The achieved performance was 29.5 Tflops.

This paper will be organized as follows. In section 2, we present some scientific background for the planetesimal-protoplanet system. In section 3 we discuss what kind of algorithms we can use for simulations of such systems. In section 4, we describe the basic idea of GRAPE systems, which are specially designed for this kind of computer simulations. In section 5 we describe the architecture of the GRAPE-6 system. In section 6 we present the performance achieved. Section 7 is for summary.

2 The Uranus-Neptune region

In the standard scenario of the Solar system formation, the building blocks of planets are km-sized bodies called planetesimals (e.g., [Sa69, HNN85]). Planetesimals are formed from the solid component (dust) in the solar nebula. While orbiting the sun, planetesimals accrete to form terrestrial (rocky) and uranian (icy) planets and cores of jovian (gaseous) planets. This process is called planetary accretion. Planetary accretion is an important process of planet formation that controls the basic architecture of a planetary system and the formation time scale of planets.

The formation time scale of planets increases with the distance from the sun because the surface number density of planetesimals decreases and because the dynamical time increases with the distance from the sun. One of the most serious problems in the standard scenario is the formation of Neptune. A simple estimate implies that the formation time scale of Neptune is longer than the age of the Solar system. It is an open question how planetary accretion proceeds in the Neptune region.

The Neptune region is also important in the origin of comets. It is widely accepted that the so-called Oort cloud, which is a spherical shell-like reservoir of comets surrounding the solar system, is formed by gravitational scattering of planetesimals mainly by Neptune. In the formation process of Neptune, some planetesimals are accreted and others are scattered away from the solar system by Neptune. This scattering efficiency is an important key to understand the planetesimal evolution in the Neptune region.

The gravitational relaxation of planetesimal orbits due to mutual gravitational interaction is an elementary process that controls the planetesimal evolution. The formation time scale and the formation rate of comets depend on the velocity distribution of planetesimals.

In the present paper, we simulate the orbital evolution of planetesimals due to mutual gravitational interactions and scattering by proto-Uranus and proto-Neptune.

The system of units is chosen such that the Astronomical Unit (AU), Solar mass, and the gravitational constant are all unity. In this system of units, 1 year is 2π time units.

We distribute 1.8 million planetesimals in a ring region with inner radius of 15 AU and outer radius of 35 AU. The mass distribution of the planetesimals follows $N(m)dm \propto m^{-2.5}$, which is a stationary distribution found by numerical simulations and confirmed by simple analytic argument. The higher and lower cutoff masses are 3.84×10^{-11} and 1.192×10^{-9}, respectively. The surface mass density of planetesimals is proportional to $r^{-1.5}$, where r is the distance from the sun. The amount of planetesimals is consistent with the standard Solar nebula model [Ha81].

To model the interaction between proto-Neptune (and Uranus) and planetesimals, we place two massive protoplanets (mass 3×10^{-5}) at 20AU and 30AU on non-inclined circular orbits. All gravitational interactions (except for the Solar gravity, which is treated as an external potential field) is softened using softening radius 0.008AU. This softening is two orders of magnitude smaller than the Hill radius of the protoplanets and therefore the effect on the scattering cross section is negligible.

3 Algorithmic requirements

First of all, we need a large number of planetesimals, for the following reasons. To model the interaction between protoplanets and planetesimals, the mass ratio between them must be as large as possible to guarantee that the thermal relaxation due to interaction between planetesimals does not mask the effect of protoplanets. For this purpose, ideally we want to use the real mass for both protoplanets and planetesimals, but this is still impractical with the present speed of available computing resources. The second reason is just to obtain good resolution for spatial structure of the distribution of planetesimals.

Another requirement is that the simulation code should be able to handle a wide range in timescales. In the Uranus-Neptune region, the orbital period of protoplanets and planetesimals is of the order of 100 years. However, when two planetesimals or a planetesimal and a protoplanet undergo close encounters, the timescale can go down to a few hours. Thus, the timescale ranges six orders of magnitudes.

It is crucial to be able to assign different timesteps to different particles, and also to change the timestep of each particle adaptively (using individual timestep algorithms, [Aar63]). This wide range of timescale also means that we need to integrate particles with short timescale with high accuracy to maintain reasonable overall accuracy of the result.

This combination of the requirement for the large number of particles and wide range of timescale results in one of the hardest kind of problems in computational science. If all we need is just a large number of particles, but with similar timescales, we can use Barnes-Hut tree algorithm or Fast Multipole method to reduce the calculation cost per timestep from $O(N^2)$ to $O(N \log N)$ or $O(N)$. However, it is very difficult to achieve high efficiency with these algorithms when the timesteps of particles vary widely.

It is not impossible to combine the individual timestep approach with a tree algorithm [MA93], and such calculation scheme has been used in the field of cosmological N-body simulations. However, the actual gain in the calculation speed turned out to be rather small, even for relatively favorable situation of the cosmological context [SYW01] where the range of the timescale is not as wide as that in our problem of planetesimals.

Even without tree algorithm, the implementation of the individual timestep algorithm on massively parallel computers turned out to be a difficult problem. The reason is that we need

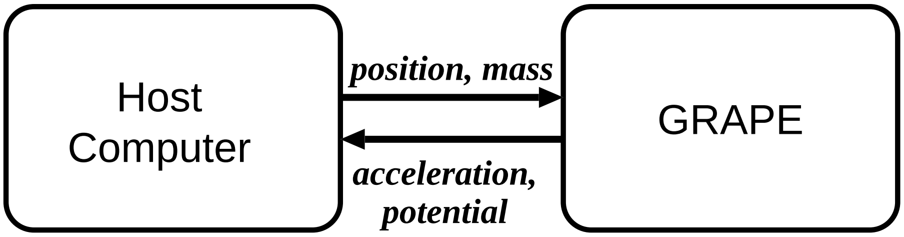

Figure 1: Basic concept of a GRAPE system.

a very low-latency, high-bandwidth network which can perform broadcast or global reduction, or preferably both. No one has yet achieved the effective speed of more than 10 Gflops for individual timestep algorithm implemented on MPPs.

4 GRAPE hardware

4.1 Basic concept

Many astrophysical systems share the same characteristic of exhibiting a wide range in timescales. This problem arises directly from the fact that the gravitational force is an attractive force with no characteristic scale length.

In order to accelerate N-body simulations with individual timestep algorithms, we have developed a series of special-purpose hardwares for the force calculation [SCM$^+$90, MT98]. Figure 1 shows the basic structure of our GRAPE (GRAvity piPE) system.

The direct force calculation is well suited for acceleration by specialized hardware, because of its simplicity. A GRAPE system consists of a general-purpose frontend and special-purpose hardware. The special-purpose part consists of custom-design pipeline chips for gravitational force calculation.

4.2 Individual timestep

The use of individual timesteps adds extra complexity to the hardware and software, but still we can achieve pretty high performance.

In the software side, it is necessary to extract as much parallelism as possible. This is achieved by using the so-called blockstep method [McM86, Mak91], in which timesteps of particles are forced to powers of two and the scheduling algorithm is changed so that all particles with the same updated time are integrated in parallel.

In the hardware side, the entire hardware must be designed so that it can deliver reasonable performance when asked to evaluate the forces on relatively small number of particles. Even with the blockstep method, the average number of particles which can be integrated in parallel might be as few as one hundred or less, even for $N = 10^5$ or larger. We let multiple pipelines to calculate the force on one same particle, but from different subsets of particles. The partial forces are then summed up using a reduction tree hardware.

For particles with time different from the current time, their positions must be predicted using predictor polynomials. The pipeline for this predictor should also be implemented in hardware.

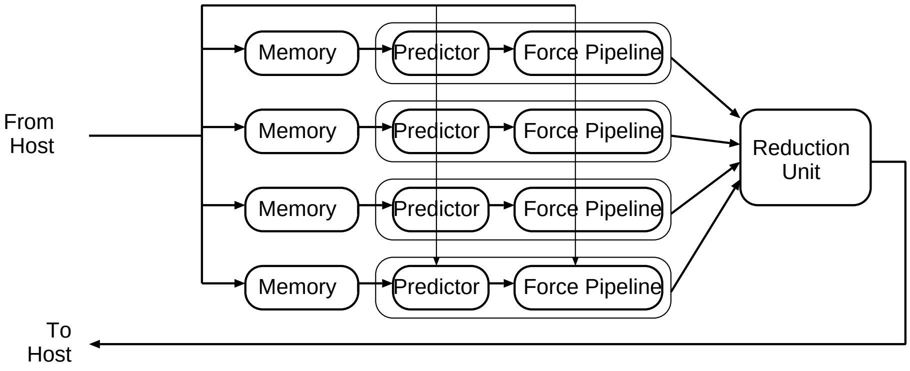

Figure 2: Parallel GRAPE pipelines for individual timestep algorithm.

Thus, actual hardware for individual timestep algorithm is somewhat more complex than the simple outline in figure 1. We show the concept in figure 2.

4.3 Parallelization of the host

The important advantage of GRAPE architecture is that the speed of communication between the host and GRAPE and the speed of calculation of the host computer need not to be very high compared to the speed of GRAPE hardware. The reason is simply that GRAPE performs $O(N)$ operation per particle per timestep, while the host performs $O(1)$ operations. Even so, since we can achieve the speed of order of 10–100 Tflops for GRAPE hardware, a single workstation with the effective speed of several hundred Mflops is too slow as a host. The speed of communication is also a problem, since currently the communication speed is limited by the peak bandwidth of the PCI bus of the host.

A naive approach to have multiple hosts each with its own GRAPE hardware does not work. Figure 3 show such a simple configuration. The problem here is that each GRAPE hardware, and therefore each host, must have the information of all particles in the system to calculate the total force on a particle. In other words, each host need to receive all informations of particles updated at the end of each timestep. This means the amount of communication is not reduced when we increase the number of host computer. Since the speed of host-host communication is also limited essentially by the PCI bandwidth, this limitation means the parallel system configured in the way shown in figure 3 is no better than a single host, as far as the communication bandwidth is concerned.

One way to solve this problem is let the GRAPE hardwares to exchange the data by themselves. We can achieve this by adding input ports to memory units in a GRAPE hardware. Figure 4 shows the smallest such configuration, where we have two hosts and two GRAPEs. Each GRAPE has two independent memory units, one host port, one data-out port and one data-in port. One memory unit is connected to the host port and the other to the data-in port, The data out port simply emits everything sent from the host. The data-in port receives the data from the other GRAPE.

Note that with this approach the host computers do not have to exchange any particle data. They still have to synchronize at the beginning of each timestep, but no further communication is necessary.

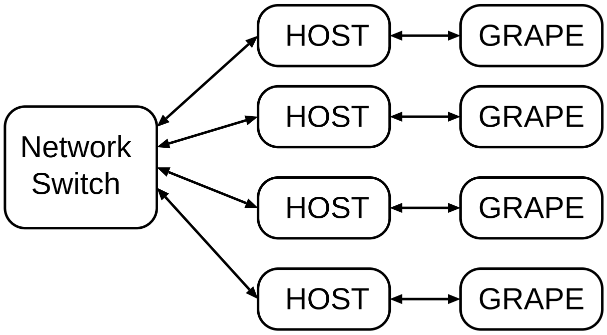

Figure 3: Simple way to use multiple host for multiple GRAPE hardware.

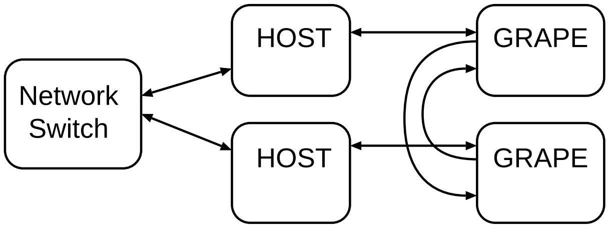

Figure 4: Parallel host with data exchange between GRAPE hardwares.

In this way, if we have p hosts, each GRAPE need to have $p-1$ data in/out ports and p memory units. We implement such a structure by separating the basic processing units and the network interface unit as shown in figure 5. Here, we show a GRAPE processor consisting of one network board (NB) and four processor boards (PBs). Each PB may be a parallel GRAPE system with multiple processor chips, but it has only one input and one output ports.

An NB has a configurable network for transferring data from the host or other network boards. Using four NBs, we can connect four host computers to 16 processor boards. The network can be configured in three modes, broadcast, 2-way multicast and point-to-point. Thus, we can use a 4-host, 16-processor board system as single entity, as two units, and as four separate units.

We can also cascade multiple NBs in a tree structure to increase the ports both to processor boards and to the hosts. Thus, we can construct arbitrarily large systems (if our budget allows), without being limited by the communication bandwidth between host computers.

Another way to solve the communication problem is to configure host computers themselves in a 2-dimensional network (figure 6). For example, if we have 16 host computers each with their own GRAPE hardware, but not with special network boards described above, we can configure the 16 hosts to a 4 × 4 matrix, and use only 4 hosts (in one row or one column) as real hosts to do time integrations and use other 12 hosts just to emulate the network boards described above. This approach gives us even more flexibility in operation, since we can divide the 2-dimensional matrix of nodes to any rectangular submatrix (down to single node) and use each of them to run separate programs. The disadvantage of this approach is that we need reasonably fast communication between the nodes in the same column (as well as those in the same row). The theoretical peak speed of Gigabit Ethernet is barely okay.

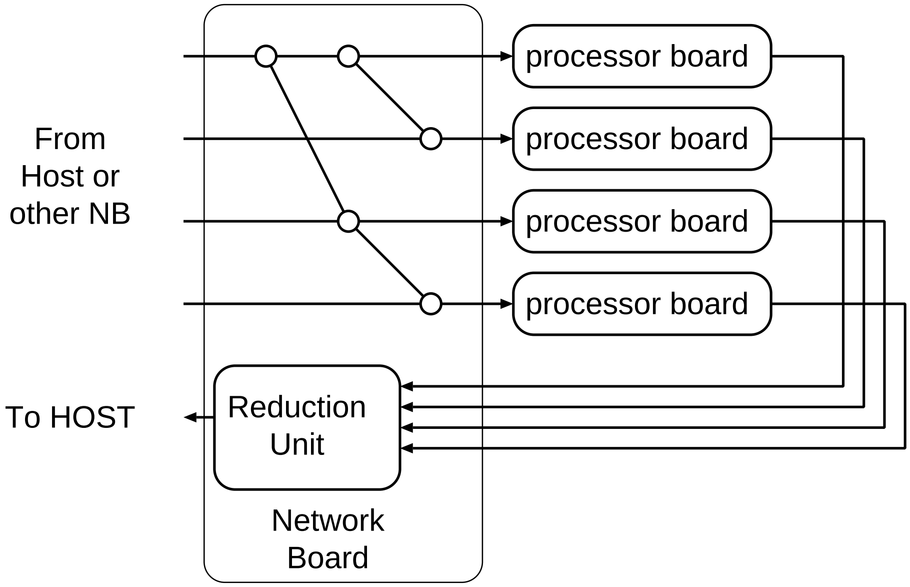

Figure 5: Scalable GRAPE architecture with multiple ports to the host.

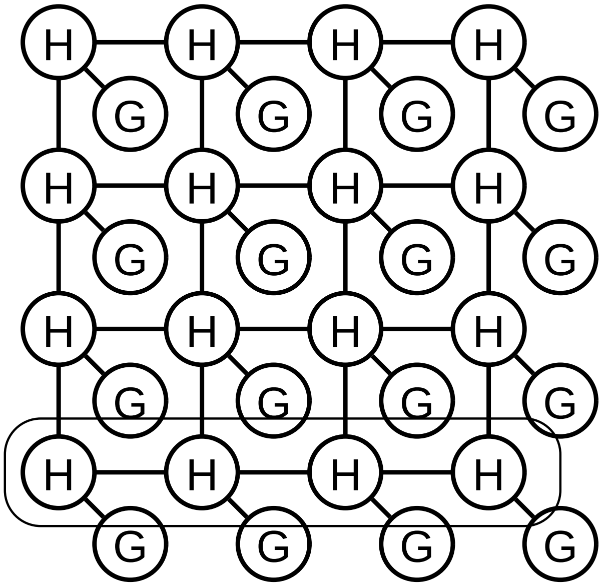

Figure 6: 2-dimensional network of host-GRAPE pairs.

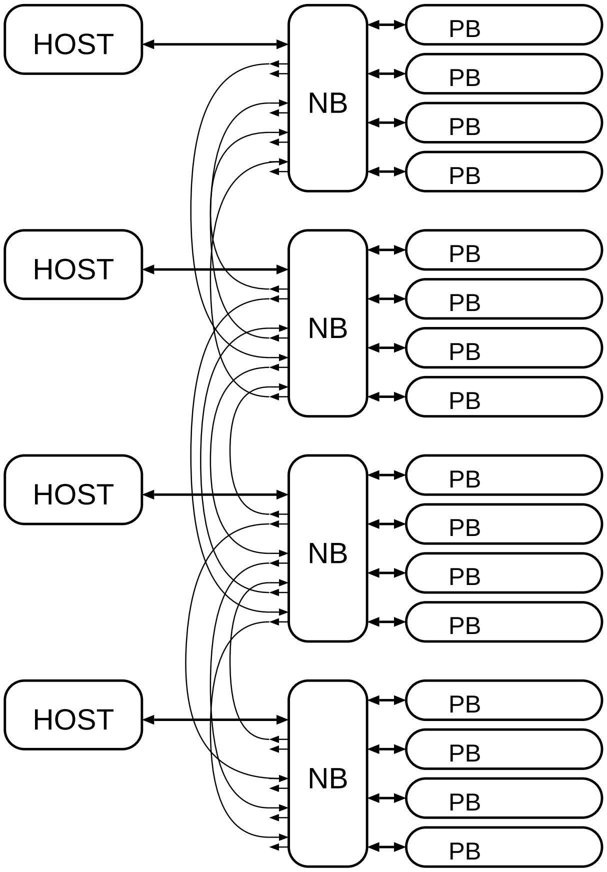

Figure 7: A GRAPE-6 cluster.

5 The GRAPE-6 system

5.1 Overall architecture

GRAPE-6 implements the parallel architecture discussed in the previous section. At present, the total system consists of 16 host computers each with four PBs and one NB. Four host computers are configured to form a 4 × 4 cluster, using dedicated network discussed in the previous section (figure 7). Communication between clusters is through Gigabit Ethernet on host computers. Thus, we adopted the hybrid of the two solutions discussed in the previous section. In the following, we call a system of single host, single node and 4 processor boards as "node", and a 4-node system with hardware network a "cluster".

5.2 Node structure

One node has one host-interface board (HIB), one network board (NB), and 4 processor boards (PB). Each NB has one uplink and four downlinks, two output port to other network board, and three cascade links from/to other network boards in the same cluster. Thus, 16 PBs are connected to the host through network of NBs (see figure 7). HIB and NB handles the communication between PBs and the host.

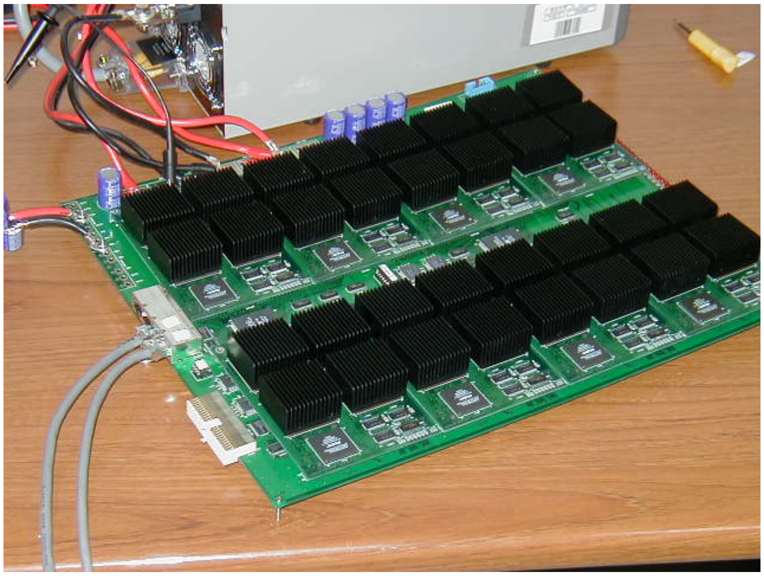

Figure 8: The GRAPE-6 processor board under testing.

The PBs perform the force calculation. Each PB houses 32 GRAPE-6 processor chips, which are custom LSI chips to calculate the gravitational force and its first time derivative. Figure 8 shows the photograph of a PB. Four processor chips and eight memory chips are mounted on a daughter card, and eight daughter cards are mounted on a processor board.

A single GRAPE-6 processor chip integrates six pipeline processors for the force calculation, one pipeline processor to handle the prediction, network interface and memory interface (see figure 9). One force pipeline can evaluate one particle-particle interaction per cycle. With the present pipeline clock frequency of 90MHz, the peak speed of a chip is 30.7 Gflops. Here, we follow the convention of assigning 38 operations for the calculation of pairwise gravitational force, which is adopted in resent Gordon-Bell prize applications. GRAPE-6 calculates the time derivative, which adds another 19 operations. Thus, the total number of floating point operations for one interaction is 57.

For the link between boards, we have adopted a fast semi-serial link with LVDS signal level. Data transfer rate through a link is 90 MB/s. It uses four pairs of twisted-pair cables (the same as the standard cable for 100Mbit Ethernet). We adopted DS90CF364AMTD and DS90C363AMTD from National Semiconductor as the LVDS devices.

The structure of an NB is essentially the same as that of the processor board, but it carries links to the next level of the tree (either NB or PB) instead of the processor chips and switching network for input data. Figure 10 shows the network board.

5.3 The entire system

The complete GRAPE-6 system consists of four clusters. Figure 11 shows the network structure. 16 host computers are connected through Gigabit network. For host computers. we used Linux running PCs with Athlon XP 1800+ CPU and 1 GB DDR main memory (ECS K7S6A mainboard). For Gigabit NIC, we used Planex GN1000TC based on NS83820 chip.

Figure 12 shows the entire system. The maze of wires in the two racks in the front side are

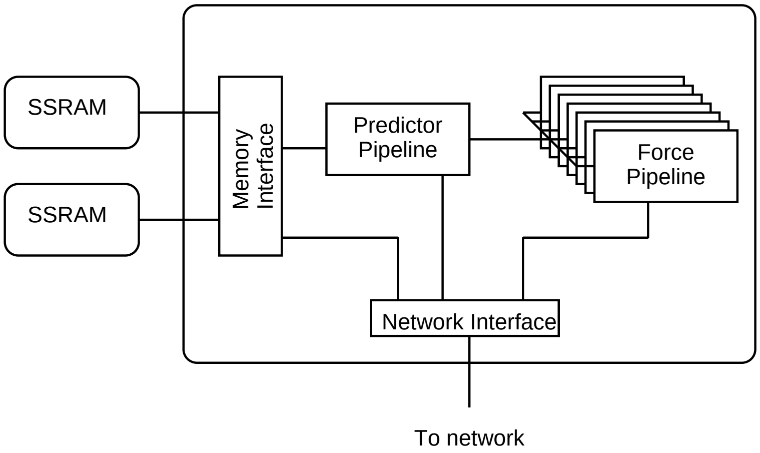

Figure 9: The GRAPE-6 processor chip.

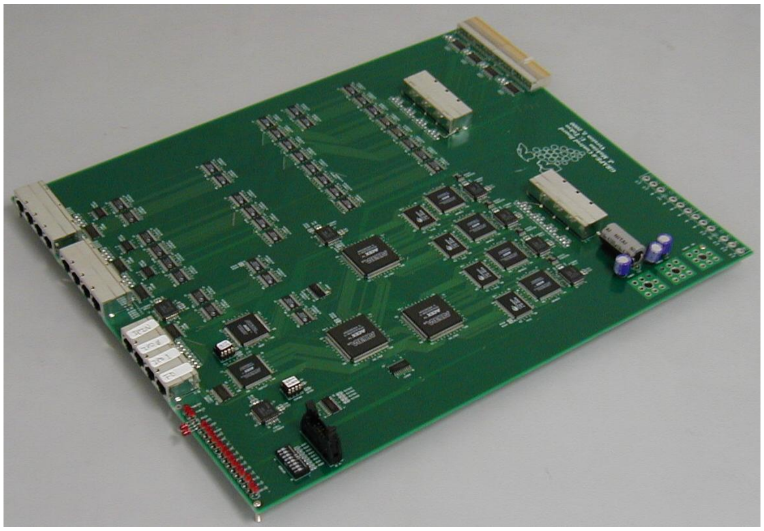

Figure 10: The GRAPE-6 network board.

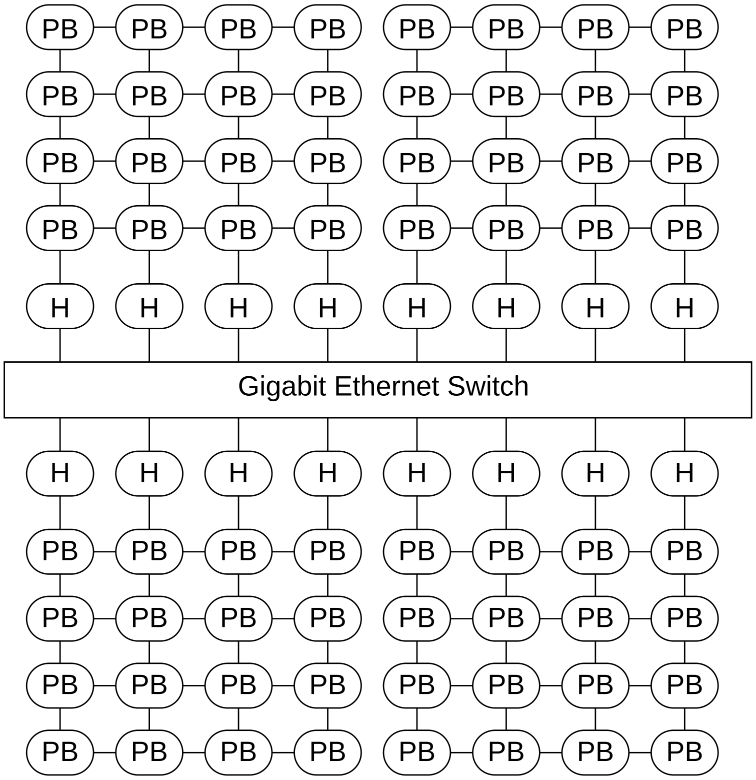

Figure 11: The GRAPE-6 System.

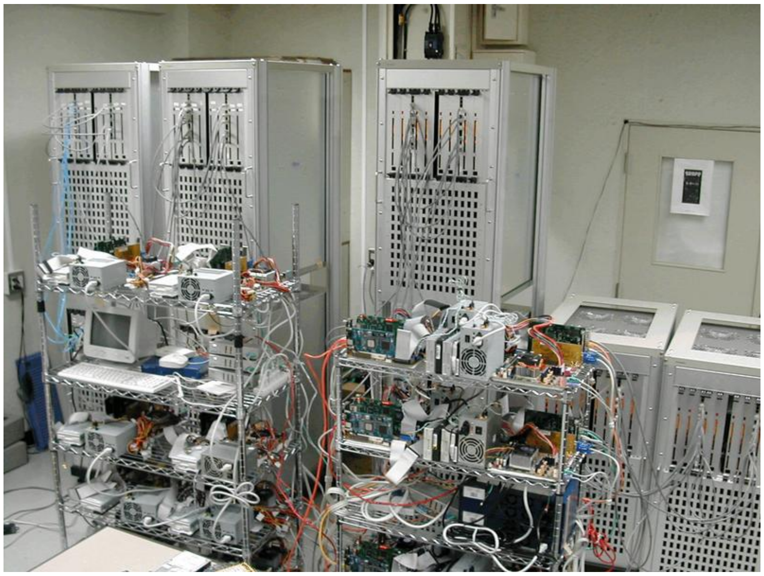

Figure 12: The GRAPE-6 System.

actually the host PCs. GRAPE-6 PBs and NBs are in the five racks. Three racks in the left house two subracks and two racks in the right house one each. Each of these eight subracks hoses 8 PBs and 2 NBs.

In this configuration, the connection between PBs and NB are through custom-designed backplane. The mechanical dimensions of the backplane and the backplane connectors on PB and NB are that of Compact PCI standard.

6 Simulation and Performance

We have simulated the evolution of a system of 1.8 million planetesimals and two protoplanets circling around the sun.

We performed a simulation for 40000 dynamical time units, for which the number of individual steps was 3.6267×10^{10}. The whole simulation, including file operations, took 35.053 hours. The total number of floating point operations is $3.6267 \times 10^{10} \times 1799999 \times 57 = 3.7210 \times 10^{18}$, since one particle-particle interaction amounts to 57 floating point operations. The resulting average computing speed is 29.5 Tflops. Figure 13 shows the distribution of planetesimals at $T = 800$ and that at $T = 30400$. Gap of the distribution is formed near the radius of protoplanets. We are currently analyzing the result.

7 Summary

In this paper, we present the performance achieved for an astrophysical N-body simulation with individual timestep and direct force calculation on the GRAPE-6 special-purpose computer. The achieved performance number is 29.5 Tflops for a simulation with 1.8 million particles.

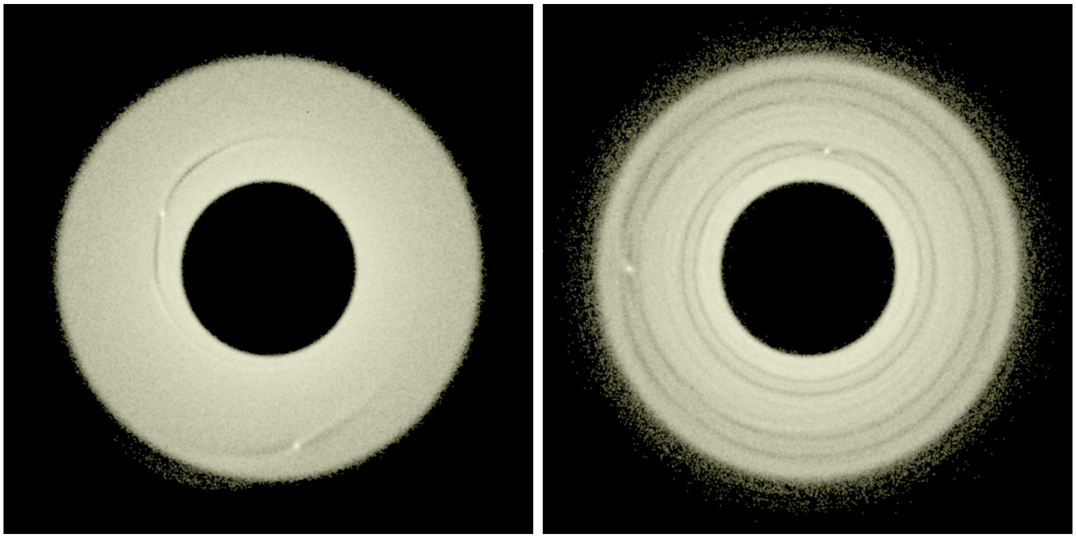

Figure 13: Distribution of planetesimals at $T = 800$ (left) and at $T = 30400$ (right).

This work is supported by the Research for the Future Program of Japan Society for the Promotion of Science (JSPS-RFTF97P01102).

References

[Aar63] Aarseth Sverre J. (1963) *MNRAS* 126: 223–255.

[Ha81] Hayashi, C. (1981) *Prog. Theor. Phys. Suppl.* 70: 35–53.

[HNN85] Hayashi, C., Nakazawa, K. and Nakagawa, Y. (1985) Formation of the solar system. In Protostars and Planets II (D. C. Black and M. S. Mathews, Eds.), pp. 1100-1153. Univ. of Arizona Press, Tucson.

[MA93] McMillan S. L. W. and Aarseth S. J. (1993) *ApJ* 414: 200–212.

[Mak91] Makino J. (1991) *PASJ* 43: 859–876.

[Mak97] Makino J. (1997) *ApJ* 478: 58+.

[McM86] McMillan S. L. W. (1986) In Hut P. and McMillan S. (eds) *The Use of Supercomputers in Stellar Dynamics*, pages 156–161. Springer, New York.

[MT98] Makino J. and Taiji M. (1998) *Scientific Simulations with Special-Purpose Computers — The GRAPE Systems.* John Wiley and Sons, Chichester.

[Sa69] Safronov, V. S. (1969) Evolution of the Protoplanetary Cloud and Formation of the Earth and the Planets, Nauka, Moscow.

[SCM$^+$90] Sugimoto D., Chikada Y., Makino J., Ito T., Ebisuzaki T., and Umemura M. (1990) *Nature* 345: 33–35.

[SYW01] Springel, V., Yoshida, N., and White, S. D. M. (2001) *New Astronomy* 6: 79-117.