A 11.55 Tflops simulation of black holes in a galactic center on GRAPE-6

Junichiro Makino* and Toshiyuki Fukushige
*Department of Astronomy, School of Science, University of Tokyo,
Tokyo 113-0033, Japan
Email: makino@astron.s.u-tokyo.ac.jp
Department of General System Studies,
College of Arts and Sciences, University of Tokyo,
Tokyo 153-8902, Japan

Abstract

As an entry for the 2001 Gordon Bell performance prize, we report the performance achieved on the GRAPE-6 system for the simulation of the dynamical evolution of a multiple black hole system in the galactic center. GRAPE-6 is a special-purpose computer for astrophysical N-body calculations. The present configuration has 1024 custom pipeline processors, each containing six pipeline processors for the calculation of gravitational interactions between particles. Its theoretical peak performance is 31.52 Tflops. The actual performance obtained on the present 1024-chip system was 11.55 Tflops, for a simulation of massive black holes embedded in the core of a galaxy with 1.1 million stars. This is an improvement in the performance of more than a factor of eight compared to our entry last year, which was 1.349 Tflops for 768k stars.

1 Introduction

In this paper, we report the performance of a 1024-chip GRAPE-6 system for the simulation of a multiple black hole system in the core of a galaxy. The simulated system consists of 1,099,997 "normal" stars and 3 "black hole" stars. The algorithm used is the block individual timestep algorithm [McM86, MH91], where each star has its own time and timesteps. Since the required accuracy for the time integration, and therefore that for the gravitational interaction, are both high, we used direct summation for the force calculation. The GRAPE-6 configuration used consisted of four host computers (Linux boxen with 1.7 GHz Intel Pentium 4 processors) and 32 GRAPE-6 processor boards, each with 32 processor chips. The host processors perform the time integration, and GRAPE-6 boards perform the force calculation. The achieved performance was 11.55 Tflops.

This paper will be organized as follows. In section 2, we describe why we want to model multiple black hole systems in galactic centers. 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 computers for this kind of simulation. In section 5 we discuss strategies to parallelize the individual timestep algorithm. In section 6 we describe the basic design of the GRAPE-6 system, along with the network structure. In section 7 we present the performance achieved. Section 8 is for summary.

2 Black holes in galactic cores

There is rapidly growing evidence for supermassive black holes (SMBHs) of the mass of 10^6 to 10^9 solar masses in the centers of many galaxies. For a recent review see Kormendy and Richstone[KR95]. There are too many examples to list here; indeed, there are only a few galaxies for which observations indicate that a central SMBH does not exist[KM93]. These SMBH are believed to be the central engines for quasars and AGN (active galactic nuclei), both of which emit vast amount of radiation in wide range of wavelengths. Thus, how these SMBHs are formed is one of the most important questions of modern theoretical astrophysics.

However, we know rather little about the formation mechanism of SMBHs. In fact, our theoretical understanding has not advanced much beyond the scenarios described by Rees[Ree78, Ree84] in the early 1980s. In the famous diagram by Rees, there were basically two paths from gas clouds to massive black holes. The first is direct monolithic collapse. The second is via the formation of a star cluster, with subsequent runaway collisions leading to black hole formation. Previous numerical studies, however, have demonstrated that neither path is likely.

Very recently, Matsumoto et al. [MTK$^+$01] have found bright compact X-ray sources in the central region of nearby galaxy M82 using data from the Chandra X-ray Observatory. The brightest source (No. 7 in their Table 1) had the estimated lower limit mass of $700M_\odot$ (solar mass). This is the *first* detection of a black hole with a mass much greater than $100M_\odot$ but less than $10^6M_\odot$ (intermediate-mass black hole, IMBH).

Harashima *et al.* [HIT$^+$01] observed the same region in the infrared using the CISCO instrument on the SUBARU telescope. They identified a number of young compact star clusters, at least four of them coinciding with the X-ray sources within the position uncertainty of Chandra and SUBARU. The logical conclusion from these observations is that most of Chandra X-ray sources, including the brightest one with an Eddington mass of $700M_\odot$, are formed in young star clusters.

This finding contradicts the theoretical prediction that black hole formation is unlikely in star clusters. However, if we examine the assumptions made for that theoretical prediction, it is clear that the assumptions are not adequate for the star cluster found in M82. For the star cluster in M82, it is quite likely that multiple collisions between stars in the core of the cluster led to the formation of supermassive stars and then IMBHs.

However, even if an IMBH $> 700M_\odot$ is found in a star cluster in M82, it is still a few orders of magnitude smaller than SMBHs found in other galaxies. The question here is whether they are relate.

Theoretically, it is plausible that IMBHs are the building blocks for SMBHs. The reason is that the star clusters so close to the galactic center would sink toward the center of the galaxy through dynamical friction. Since there are several other similar star clusters with central X-ray sources, and because more would have been formed in the past, there seem to be a sufficient number of IMBHs that reached to the center of the galaxy through the lifetime of M82 to form an SMBH. Other galaxies would also have had similar experiences.

If the IMBHs are carried to the center of the galaxy by their host cluster, the next question is whether they can merge to form SMBHs. This question is actually a rather old one, posed and studied in very different contexts. Numerical studies [ME96, Qui96, Mak97, QH97] seem to suggest that such merging is unlikely if there are only two black holes. The reason is that if there are only two black holes, they would eventually sweep out all the stars in their neighborhood and leave themselves in an empty space. A binary in an empty space is perfectly stable and

would follow no further evolution. The separation of the two black holes at which this sweeping out occurs is not small enough for the gravitation wave radiation to have a significant effect.

If there are more than two black holes, however, the evolution would become entirely different [HR92, ME94]. For example, it is very unlikely that three black holes form a dynamically stable system. There are basically two possibilities for the final outcome. One is that one or more black holes will be ejected from the galaxy through three-body interaction of black holes. The other possibility is that during the three-body interaction two black holes come close enough that they merge through gravitational wave radiation. Thus, even after black holes have kicked out all nearby stars, they can still evolve in a complex way. There is, however, no simulation study on the evolution of triple black hole system, except for the one we reported in our Gordon Bell entry last year[MFK00], which was not yet conclusive. Thus, this year we decided to perform simulations similar to what we performed last year, but with a larger number of particles.

3 Algorithmic requirements

For detailed discussion of the requirement for the numerical procedure, we refer the readers to our SC2000 article [MFK00]. Here, we repeat the basic nature of the problem.

First, we need a large number of stars to reduce the "particle noise" or the two-body relaxation effect. Second, the simulation should cover a very wide range in timescales. Typical stars in a galaxy have orbital timescales of 10^8 years, and therefore the timestep necessary to integrate the orbit of these stars accurately is around 10^5 years. The orbital timescale of the black holes can be anywhere between 10^6 years and 10 minutes. Thus, it is crucial to assign different timesteps to different stars (using individual timestep algorithms, [Aar63]). This short timescale also requires rather high accuracy for time integration and therefore for the calculation of gravitational interaction.

It is not impossible to combine the individual timestep approach with a tree algorithm [MA93] and achieve high accuracy at the same time. However, in order to obtain astrophysically useful results from such a code, parallelization is necessary in order to obtain high calculation speeds, and that poses a difficult problem.

Even without the 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 a very low-latency, high-bandwidth network. No one has yet achieved the effective speed of more than 10 Gflops for the individual timestep algorithm implemented on MPPs. Dorband[Dor01] recently reported the speed equivalent to 4 Gflops on a 128-processor Cray T3E for $N = 128k$.

4 GRAPE hardware

4.1 Basic concept

Black holes in the center of a galaxy form just one example of tough problems in astrophysics. 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 hardware for the force calculation [SCM+90, MT98]. Figure 1 shows the basic structure of our GRAPE (GRAvity piPE) system.

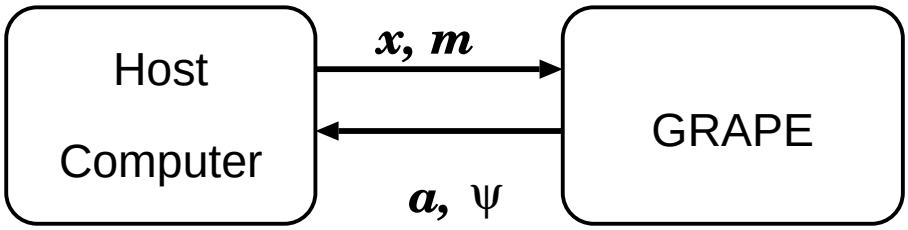

Figure 1: Basic concept of a GRAPE system.

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.

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.

5 Parallelization on host computers

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. Thus, we need to use multiple host computers in parallel.

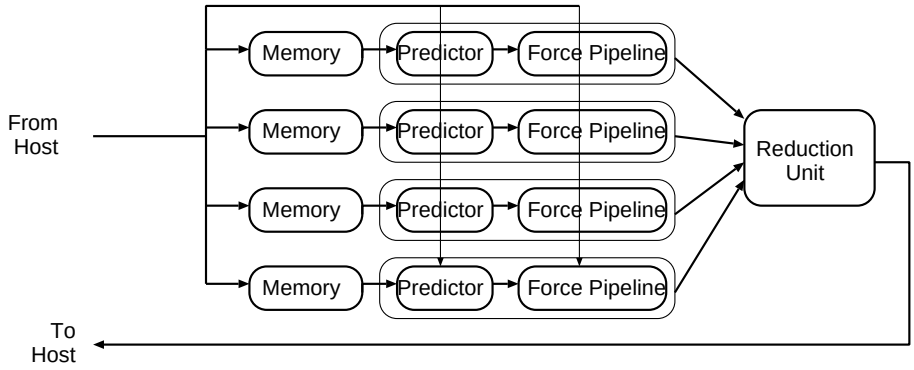

Figure 2: Parallel GRAPE pipelines for individual timestep algorithm.

To use multiple host computers is, however, a rather difficult task, since we are faced with the same limitations as that limit the efficiency of general-purpose MPP systems for the individual timestep algorithm. In this section, we first review the problem and describe the solution we adopted for GRAPE-6.

5.1 Traditional parallel algorithms and their limitation

Consider the case that we want to integrate a system of N particles using p processors. We consider the case where processors are connected through some communication network without physically sharing the memory, like in the case of most of MPPs and PC clusters.

Traditionally, two different algorithms have been used to parallelize the direct summation method. One is the ring algorithm (sometimes called also as systolic algorithm), and the other is what we call here as a "copy" algorithm.

In the ring algorithm, N particles are divided into p subsets, each consisting of N/p particles. Each of the p processors takes care of one subset. To obtain the forces on particles, each processor need to know the positions of all other particles in the system. This can be achieved either by passing through the data through one-dimensional ring, or by letting processors broadcast their data one by one. In either case, all processors send and receive $O(N)$ data in each timestep.

Note that we cannot use this algorithm with the individual timestep algorithm which which the number of particles integrated at one timestep is much smaller than N. Instead, we can pass around the particles for which we want to calculate the force, and let processors calculate forces from their particles to particles it received.

In the copy algorithm, each processor has a complete copy of the system. They then calculate the forces on its share of particles and integrate their orbits. After integration is done, they communicate the updated particles either by ring communication or broadcast. With this scheme, individual timestep is rather easy, since communication is needed only for updated particles.

In both algorithms, all processors send and/or receive all particles updated in each blockstep. Thus, for a given number of particles N, there is a theoretical limit for the calculation speed, which is determined by the communication speed of a single processor. The maximum number of processors we can use is $O(N)$, with a rather small coefficient which reflects the ratio between the calculation speed and communication speed. If we increase the calculation speed of a single processor without changing the communication speed, the number of processors we can use is

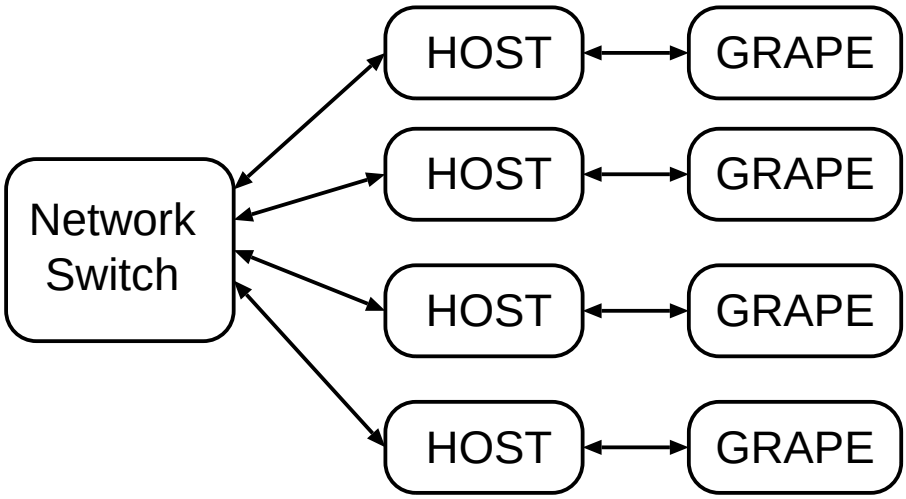

Figure 3: Simple way to use multiple host for multiple GRAPE hardware.

reduced, and the maximum speed we can achieve remains the same.

Even with general-purpose MPPs, this is a rather severe limitation. With GRAPE hardware, this becomes a critical issue since what GRAPE does is exactly to increase the calculation speed without changing anything else. Thus, a naive approach to have multiple hosts each with its own GRAPE hardware (see figure 3) does not work.

5.2 Solution with GRAPE-6

One way to solve this problem is to let the GRAPE subsystems to exchange the data by themselves. We can achieve this by adding more input port to to each of GRAPE subsystems. Figure 4 shows the smallest of such a 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. Thus, communication bottleneck is completely removed, and with p host computers we can achieve p times faster communication. Of course, this is simply because we added p input ports to a GRAPE hardware.

We implemented 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 (PBs). The network can be configured in three modes, broadcast, 2-way multicast and point-to-point. Thus, we can use a 4-host, 16-PB system as a single entity, as two units, and as four separate units.

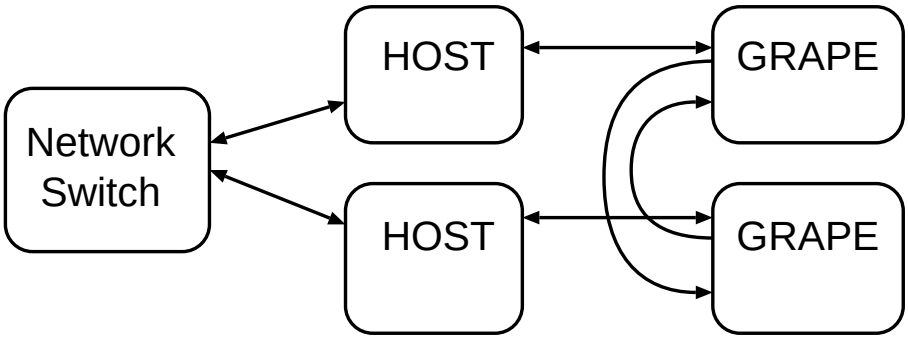

Figure 4: Parallel host with data exchange between GRAPE hardwares.

Figure 6 shows such a 4-host, 16-PB system. Each NB has 4 input ports. One input port has two associated output ports, and each of other three has one associated output port. By connecting NBs as shown in the figure, we can let each NB receive data from all four hosts and transfer them to four PBs connected to it.

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.

6 The GRAPE-6 system

6.1 Architecture

GRAPE-6 implements the parallel architecture discussed in the previous section. In the current plan, the total system will consist of 8 clusters each with 8 PBs. The 8 PBs are connected to hosts through Three NBs. Three NBs form a 8-input, 8-output network. In the following, we call a group of eight processor boards connected to a single host a "cluster".

One cluster has one host-interface board (HIB), three network boards (NB), and 8 processor boards (PB). Each NB has one uplink and four downlinks. Thus, 16 PBs are connected to the host through two-level tree network of NBs (see figure 7). HIB and NB handles the communication between PBs and the host.

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, and network and memory interfaces (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 30.8 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[WSB+97, MFK00]. GRAPE-6 calculates the time derivative, which adds another 19 operations. Thus, the total number of floating point operations for one interaction is 57.

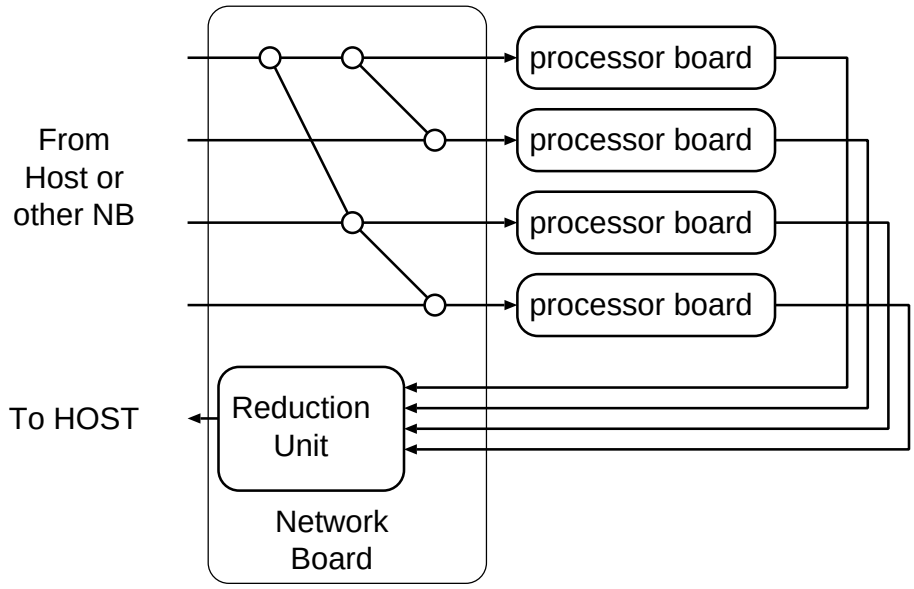

Figure 5: Scalable GRAPE architecture with multiple ports to the host.

For the link between boards, we have adopted a fast semi-serial link with LVDS (low-voltage differential signal) signal level. The 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 100 Mbit 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. Figure 10 shows the network board.

6.2 Development status

As of the time of writing, we have a 32-PB system with four host computers (figure 11). In this configuration, we have two 16-PB, 4-NB networks connected to the hosts through two separate ports. Thus, each host has two PCI interface cards. The host computers used have Intel Pentium 4 CPUs (1.7 GHz) and 256MB DRDRAM memory, and connected through 100BT Fast Ethernet. Each of the four large card cages in figure 11 houses eight PBs and two NBs. Connections between PBs and an NB go through the backplane connectors. All connections between NBs and host computers are through twisted-pair cables.

We used usual MPICH/p4[1] over TCP/IP as the message passing library. The total number of chips used for the calculation is 1024 and the theoretical peak speed of the system is 31.52 Tflops.

6.3 Comparison with our entry last year

Figure 12 shows the GRAPE-6 system we used for our Gordon Bell prize entry for year 2000. In this configuration, we used PBs with 16 chips. So the peak performance was only 2.9 Tflops.

[1] `http://www-unix.mcs.anl.gov/mpi/mpich/`

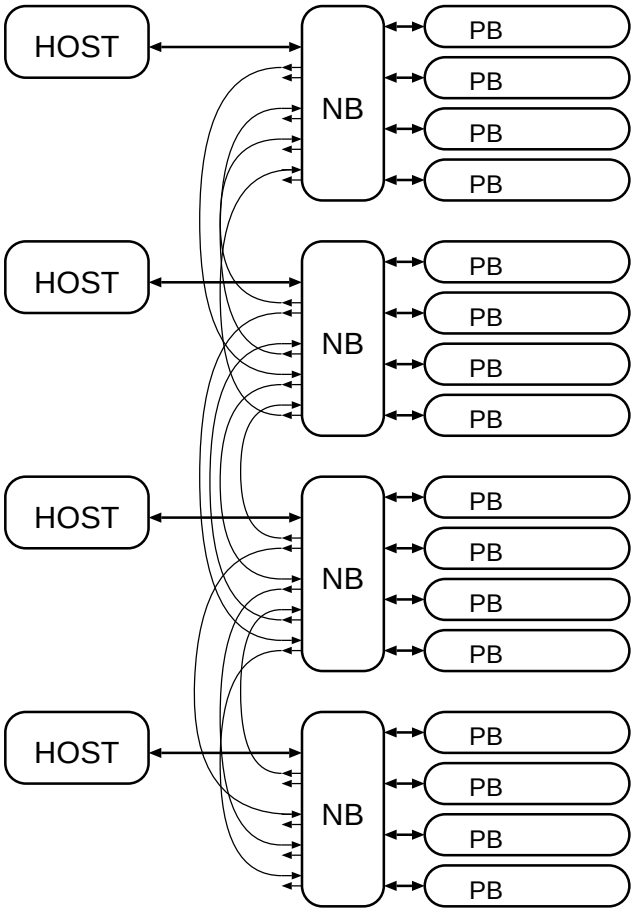

Figure 6: A 4-host, 16-PB system.

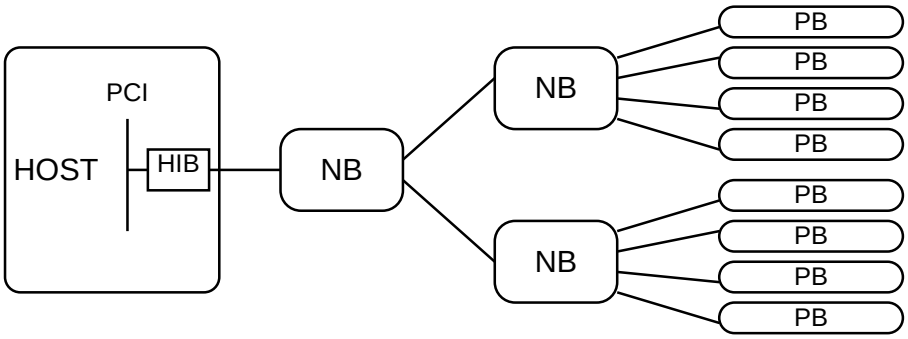

Figure 7: A GRAPE-6 cluster. Interface to the multicast network are not shown.

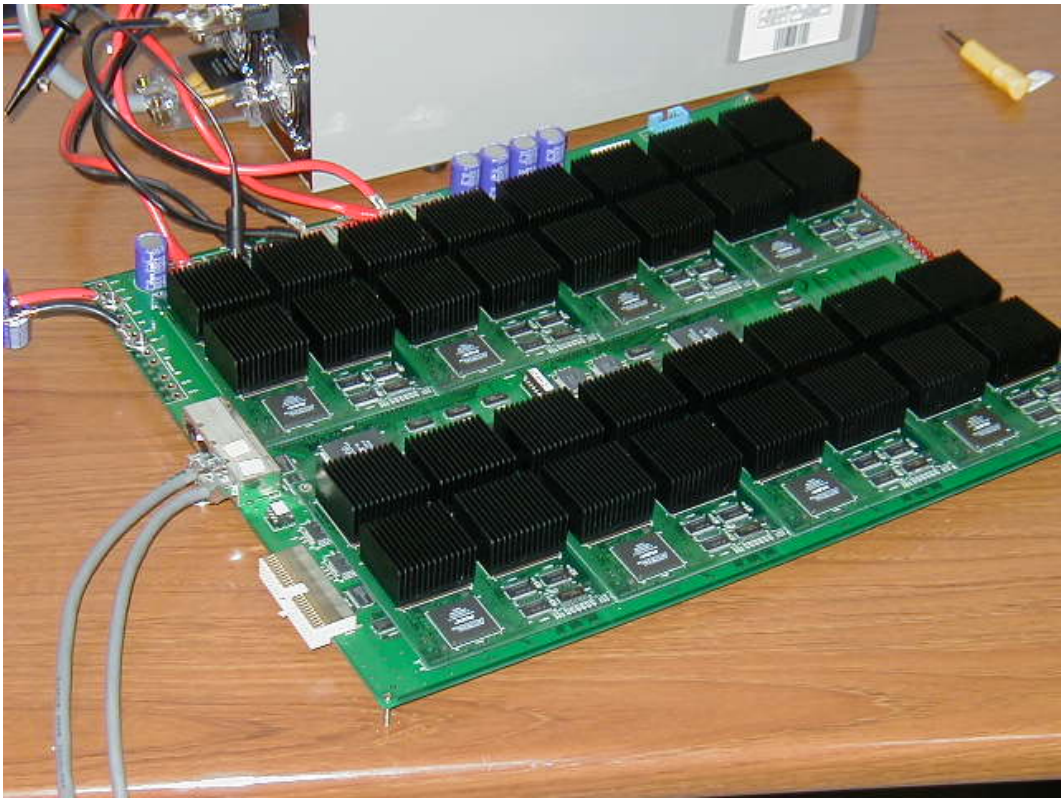

Figure 8: The GRAPE-6 processor board under testing.

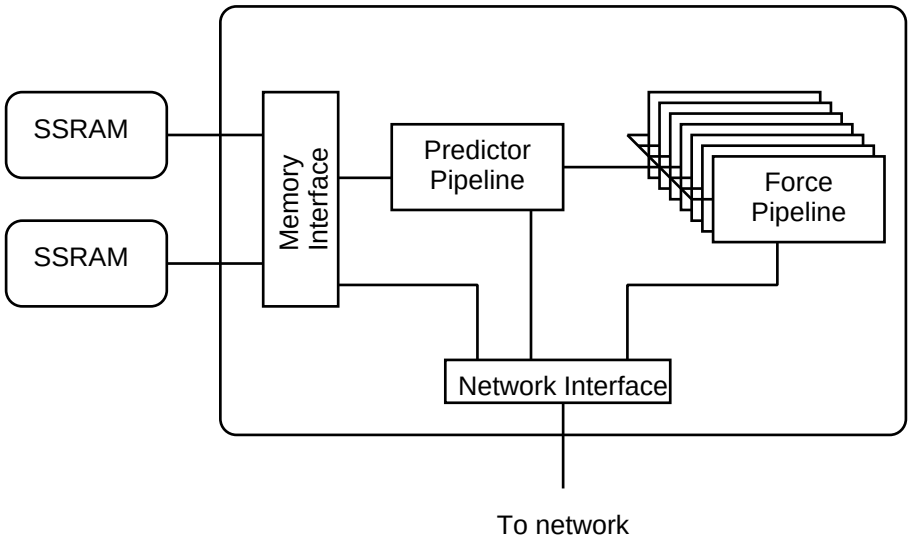

Figure 9: The GRAPE-6 processor chip.

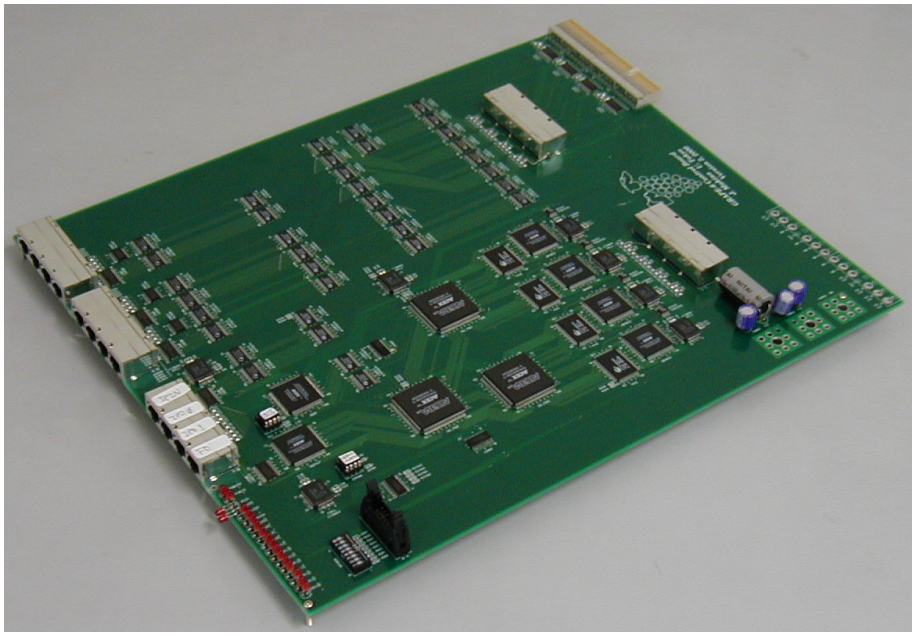

Figure 10: The GRAPE-6 network board.

Table 1: Comparison between our entries this and last years.

	last year	this year
Peak speed	2.9 TF	31.52 TF
No. of chips	96	1024
No. of PBs	6	32
Network	Simple tree	Scalable
Host	1 Alpha CPU	Cluster of 4 Pentium 4 boxen
Achieved speed	1.349 TF	11.55 TF

Also, the connection between NBs were not used, and therefore we could not use a PC cluster as the front end. Table 1 summarizes the difference.

Clearly, we improved the peak and achieved performances primarily by increasing the number of processor boards and number of chips. In order to realize this large increase in parallelism, we developed a scalable network to avoid the communication bottleneck of a single host computer.

7 Simulation and Performance

We have simulated the evolution of a galactic nucleus containing triple massive black holes. In our simulation, we modeled a galaxy with 1,099,997 equal-mass stars. The black holes are modeled as three point-mass particles each with a mass of 1% of the total mass of the system. The relativistic effects were negligible in our simulation.

We performed a simulation for 10 dynamical time units, for which the number of individual

Figure 11: The present GRAPE-6 configuration used for the simulation.

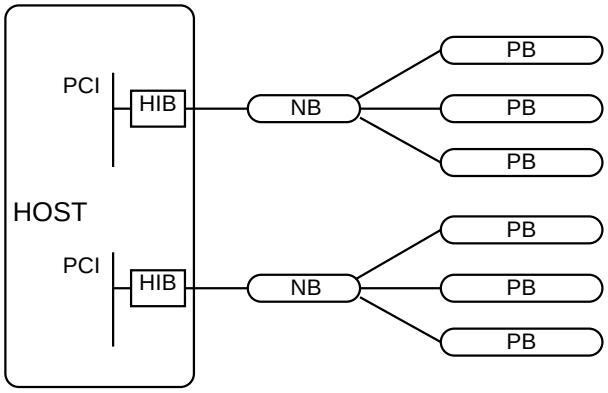

Figure 12: The GRAPE-6 configuration used last year.

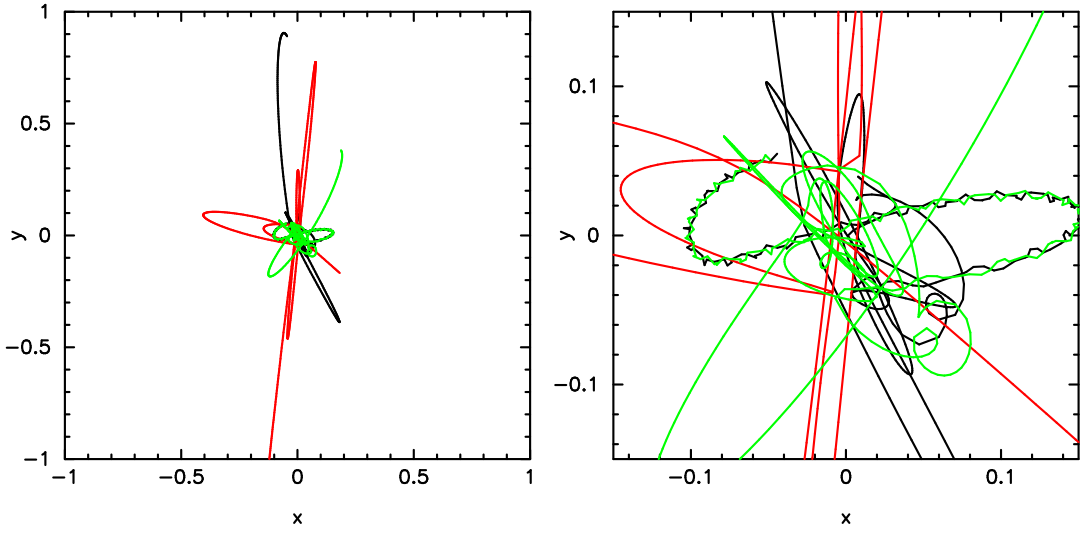

Figure 13: The trajectories of the black-hole particles projected onto the x-y plane. The right panel is the enlargement of the central region.

steps was 4.426×10^9. The whole simulation, including file operations, took 6.671 hours. The total number of floating point operations is $4.426 \times 10^9 \times 1,099,999 \times 57 = 2.775 \times 10^{17}$, since one particle-particle interaction amounts to 57 floating point operations. The resulting average computing speed is 11.55 Tflops.

Figure 13 shows the trajectories of black hole particles. Initially, all black holes are far from the center, but they quickly sink toward the center due to the dynamical friction from other field stars. Eventually, through complex three-body interaction between black holes, two of them form a binary system. Through further interaction with the third black hole, this binary become more tightly bound.

8 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 11.55 Tflops for a simulation with 1.1M particles.

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.

[Dor01] Dorband E. N. (2001) *Systolic and Hyper-Systolic algorithms for the gravitational N-body problem*, Master's thesis, The State University of New Jersey.

[HIT+01] Harashima T., Iwamuro F., Tsuru T., Maihara T., Motohara K., Terada H., Matsumoto H., Matsushita S., and Kawabe R. (2001) Near-IR spectroscopy of a starburst galaxy M82 with SUBARU. Talk at ASJ annual meeting (2000-S02a).

[HR92] Hut P. and Rees M. J. (1992) *MNRAS* 259: 27P–30P.

[KM93] Kormendy J. and McClure R. D. (1993) *AJ* 105: 1793–1812.

[KR95] Kormendy J. and Richstone D. (1995) *Ann. Rev. Astron. Astroph.* 33: 581+.

[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.

[ME94] Makino J. and Ebisuzaki T. (1994) *ApJ* 436: 607–610.

[ME96] Makino J. and Ebisuzaki T. (1996) *ApJ* 465: 527–533.

[MTES97] Makino J., Taiji M., Ebisuzaki T., and Sugimoto D. (1997) *ApJ* 480: 432-446.

[MFK00] Makino J., Fukushige T., and Koga M. (2000) In *The SC2000 Proceedings*, CD–ROM. IEEE, Los Alamitos, CA.

[MH91] Makino J. and Hut P. (1991) *ApJ* 383: 181–191.

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

[MTK+01] Matsumoto H., Tsuru T., Koyama K., Awaki H., Canizares C., Kawai N., Matsushita S., and Kawabe R. (2001) *ApJL* 547: L25–L28.

[PMMH01] Portegies Zwart S. F., Makino J., McMillan S. L. W., and Hut P. (January 2001) *ApJL* 546: L101–L104.

[QH97] Quinlan G. D. and Hernquist L. (1997) *New Astronomy* 2: 533–554.

[Qui96] Quinlan G. D. (1996) *New Astronomy* 1: 35–56.

[Ree78] Rees M. J. (1978) *The Observatory* 98: 210–223.

[Ree84] Rees M. J. (1984) *Ann. Rev. Astron. Astroph.* 22: 471–506.

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

[WSB+97] Warren M. S., Salmon J. K., Becker D. J., Goda M. P., and Sterling T. (1997) In *The SC97 Proceedings*, CD–ROM. IEEE, Los Alamitos, CA.