

A 1.349 Tflops simulation of black holes in a galactic center on GRAPE-6*

Junichiro Makino*, Toshiyuki Fukushige and Masaki Koga

*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 2000 Gordon Bell performance prize, we report the performance achieved on a prototype GRAPE-6 system. GRAPE-6 is a special-purpose computer for astrophysical N -body calculations. The present configuration has 96 custom pipeline processors, each containing six pipeline processors for the calculation of gravitational interactions between particles. Its theoretical peak performance is 2.889 Tflops. The complete GRAPE-6 system will consist of 3072 pipeline chips and will achieve a peak speed of 100 Tflops. The actual performance obtained on the present 96-chip system was 1.349 Tflops, for a simulation of massive black holes embedded in the core of a galaxy with 786,432 stars. For a short benchmark run with 1,400,000 particles, the average speed was 1.640 Tflops.

1 Introduction

The structure and dynamics of the central region of galaxies is an important field of research in astrophysics. Of particular interest is the existence of massive black holes. Observations using the Hubble Space Telescope (*e.g.*, [vv98]) and large interferometer arrays (*e.g.*, [MMH⁺95]) have shown that many galaxies host central massive objects with a mass million or even billion times larger than that of the mass of our sun.

The most direct observational evidence for the existence of a massive black hole is for the presence of a black hole in the very core of our own galaxy [EOG99]. Here, thanks to our close proximity, observers can now see individual stars with a distance of less than 0.01 parsecs from the central massive object and measure their radial (line of sight) velocities as well as tangential velocities (through proper motion measurements). All evidence is quite consistent with the existence of a massive black hole with a mass of $2.7 \times 10^6 M_{\odot}$ (here M_{\odot} is one solar mass) at the center of the galaxy.

The evidence for the existence of black holes in distant galaxies is less direct. Nonetheless, there is now little doubt that most galaxies host massive central objects, which are most likely

*0-7803-9802-5/2000/\$10.00 (c) 2000 IEEE.

massive black holes. Among these massive black holes, those in large elliptical galaxies are known to be the most massive. The mass of black holes in normal spiral galaxies is typically several million solar masses. The mass of black holes in elliptical galaxies is much larger, typically more than a hundred million solar masses.

How are such massive black holes formed? How is their formation process related to the formation process of the galaxy itself? Why do elliptical galaxies have more massive black holes than spiral galaxies? In order to answer these questions, we need to interpret observations with the help of numerical experiments (simulations).

The standard theory of the formation of elliptical galaxies is that they are formed through merging of smaller galaxies, either spiral galaxies or elliptical galaxies [TT72]. Numerical simulations of galaxy merging have demonstrated that this theory explains many observed features of elliptical galaxies very well.

When two spiral galaxies merge, the gas in the disk of a galaxy is strongly perturbed by the gravitational force from the other galaxy, and tends to concentrate into the central region of the galaxy well before two galaxies actually merge. Such a high-density gas results in very high star forming activity, which is observed in many interacting galaxies. Also, some of the gas would eventually be absorbed by the central black hole.

Thus, when two galaxies finally merge, the black holes have become much more massive than their original mass, and most of the gas in the galaxies is lost. These black holes will settle in the center of the merger galaxy, through the dynamical friction from other stars in the galaxy. In other words, the galaxy that is the product of the merger of two galaxies is likely to have two black holes in the center. What will happen to these two black holes?

These black holes form a bounded pair (binary), and the binding energy of the binary increases over time through dynamical friction. If the dynamical friction remains effective until the separation of the two black holes in the binary becomes small enough for the gravitational radiation to shrink the separation further, the two black holes will merge. Such mergings of two massive black holes would emit strong gravitational wave which will be observable through space-based gravitational wave detectors such as LISA [JCR94].

However, N -body simulations of the evolution of binary black holes in the center of a galaxy [ME96, Qui96, Mak97, QH97] have suggested a different picture. Before the separation of the black holes becomes small enough, the binary black hole expels almost all the stars in its vicinity, resulting in a region devoid of stars. As a result, dynamical friction from field stars becomes inefficient.

N -body simulations of binary black holes in the center of the galaxy pose numerically very tough problems. There are several reasons why these are tough. In order to obtain a reliable result, we need to use as many particles (stars) as possible, to reduce the numerical relaxation effects. Here, numerical relaxation denotes the effect that stars change their velocity randomly through encounters with other stars. The strength of this effect is proportional to $1/N$, where N is the number of stars in the system. In the limit of $N \rightarrow \infty$, this effect becomes zero. In real galaxies, this effect exists, but is very small. In numerical simulations with, say, 10^5 stars, this effect is several orders of magnitude larger than that in real galaxies.

Another reason is that 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 period of the black hole binary can be anywhere between 10^6 years and 10 minutes. Of course, we do not need to follow a binary with an orbital period of 10 minutes for a long time, since the black

holes would merge in a few orbital periods anyway. However, in order to see if gravitational radiation becomes effective or not, we need to follow binaries with an orbital period of about one year. Thus, it is crucial to assign different timesteps to different stars (using individual timestep algorithms, [Aar63]).

This requirement to handle a wide range of the timescales leads to the additional requirement of very high accuracy, simply because black holes and stars close to black holes are integrated for a very large number of orbits. The gravitational forces on particles should be calculated to high accuracy. The orbits should also be integrated with high accuracy.

These two requirements of a wide range in timescales and high accuracy makes the numerical simulation very difficult. In particular, no one has succeeded to use the tree algorithm for this kind of problem. The tree algorithm is now widely used for simulations of the formation and evolution of galaxy clusters, galaxy groups and individual galaxies. However, in these calculations the central regions of galaxies are not correctly modeled.

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.

In fact, even without tree algorithm, the implementation of the individual timestep algorithm on massively parallel computers turned out to be indeed a difficult problem. The reason is that we need a very low-latency network which can perform broadcast or global reduction, or preferably both. The fraction of the stars with very small timestep is small, but since they have such small timesteps, a large fraction of the total calculation time is spent to obtain gravitational forces on them. So we have to efficiently parallelize the calculation of the force on a small number of particles. One can get good performance on vector processors, shared-memory multiprocessors, and small-scale distributed-memory parallel computers with a very low-latency network, but with present large MPPs, it's very difficult to achieve good performance.

So far, the largest simulation of a black hole binary in the galactic core published is those by GRAPE-4 [Mak97] with 256K particles to express galaxies. Quinlan and Hernquist [QH97] reported similar simulation with 100K particles, but with spherical expansion method with only spherically symmetric term in the expanded potential. Thus, in their calculation they could not simulate the merging process of two galaxies. In addition, it is rather difficult to test the validity of the calculation, since the neglect of the non-spherical component of the potential is clearly an oversimplification.

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 hardwares for the force calculation [SCM+90, MT98]. The direct force calculation is well suited for acceleration by specialized hardware, because of its simplicity. The use of individual timesteps adds extra complexity to the hardware and software, but still we can achieve pretty high performance. In the case of special-purpose computers, it is not too difficult to implement a fast network for broadcast and reduction. So the latency problem of general-purpose massively parallel computers is solved by hardware.

It should be noted that the use of GRAPE hardware is not limited to direct summation.

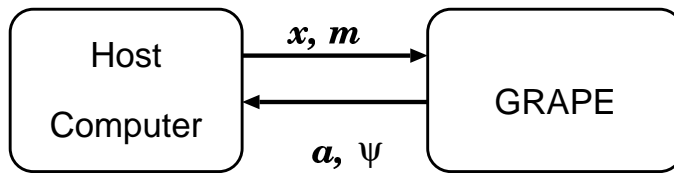


Figure 1: Basic idea of GRAPE

GRAPE can achieve pretty high performance for the tree algorithm, as demonstrated by our entry for the Gordon Bell prize last year.

In 1995, we completed GRAPE-4[MTES97] with a peak speed of 1.08 Tflops. The direct N -body simulations on this system were awarded the Gordon Bell prize in 1995 (112 Gflops) and 1996 (333Gflops). In 1997, we started the development of the GRAPE-6 system. The full GRAPE-6 system will be completed by year 2001, and will achieve a peak speed of 100 Tflops. In this paper, we report the performance of 1/48 of the total system, which is currently up and running as a testbed. In section 2, we describe the basic concept and history of our GRAPE project. In section 3, we describe the GRAPE-6 system and its present status. In section 4, we report the simulation performed on GRAPE-6.

2 GRAPE Project and its history

2.1 GRAPE architecture

In principle, the direct N -body simulation of stellar systems is very simple and straightforward. The only thing one has to do is to numerically integrate the following system of equations of motion:

$$\frac{d^2 x_i}{dt^2} = - \sum_{j \neq i} G m_j \frac{x_j - x_i}{|x_j - x_i|^3}, \quad (1)$$

where x_i and m_i are the position and mass of the particle with index i and G is the gravitational constant. The summation is taken over all stars in the system. Unless the total number of stars in the system is very small, the calculation and summation of the gravitational forces dominates the total calculation cost.

We have developed a series of special-purpose computers for N -body simulations, which we call GRAPE (GRAvity PipE). Figure 1 gives the basic idea. The host computer, which is usually a general-purpose workstation running UNIX, sends the positions and masses of particles to the GRAPE hardware. Then GRAPE hardware calculates the interaction between particles. What GRAPE hardware calculates is the right hand side of equation (1). When we use the Hermite scheme (Makino [Mak91]), the first time derivative of the force must also be calculated. Figure 2 shows the block diagram of the pipeline unit to calculate the force and its time derivative.

In order to combine the individual timestep scheme with GRAPE hardware, one modification of the basic architecture is necessary. Figure 3 shows the change. In the individual timestep scheme, particles has their own times and timesteps. Thus, to calculate the force

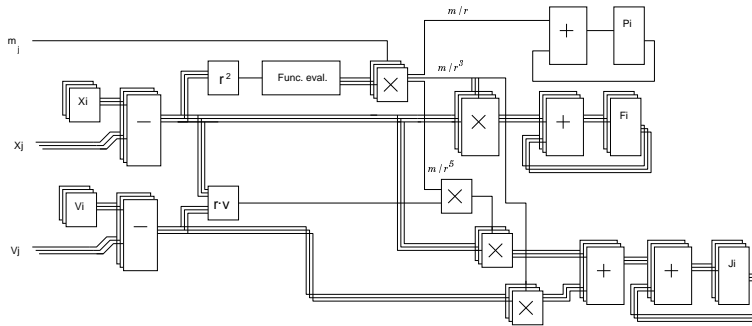


Figure 2: Pipelined processor for the calculation of force and its time derivative. Reproduced from Makino *et al.* 1997 [MTES97]).

on a particles i at its own time t_i , we need the positions and velocities of all other particles at time t_i . However, other particles have their own times, which are generally different from t_i . In the individual timestep scheme, we use the predictor based on the polynomial extrapolation to obtain the positions and velocities of other particles. Each particle has, in addition to position and velocity at its own time, acceleration and its time derivative, so that we can construct a third-order predictor. This third-order predictor allows us to use the fourth-order time integration scheme, either two-step Hermite scheme or four-step variable-stepsize Adams-Moulton scheme for second-order equations.

This prediction must also be done on the GRAPE hardware, since otherwise the amount of the calculation the host computer has to do becomes comparable to that of GRAPE, and therefore the speedup will be too limited.

In the original individual timestep algorithm, each particle choses its timestep freely, based on some timestep criterion. Thus, timesteps and times of different particles are all different, and time integration proceeds in an “event-driven” manner, where one particle with minimum $t_i + \Delta t_i$ is integrated. This scheme has one obvious problem that the degree of the parallelism in force calculation is limited to N , since we can calculate the force on only one particle. We can force several particles to share exactly the same time, by forcing their timesteps to be powers of two [McM86]. This modified individual timestep scheme is often called the blockstep scheme. This scheme is useful both on GRAPE and general-purpose computers.

In the modified architecture shown in figure 3, the particle memory keeps all data necessary for prediction, for all particles in the system. At each timestep, the host computer sends the position and velocity of particles to be updated to GRAPE, and GRAPE calculates the forces on them and sends the results back to the host. If the number of pipelines is smaller than the number of particles in the block, this step is repeated until the forces on all particles in the block are obtained. Then the host performs the orbit integration using these calculated forces, and updates the data of the integrated particles in the particle memory of the GRAPE hardware.

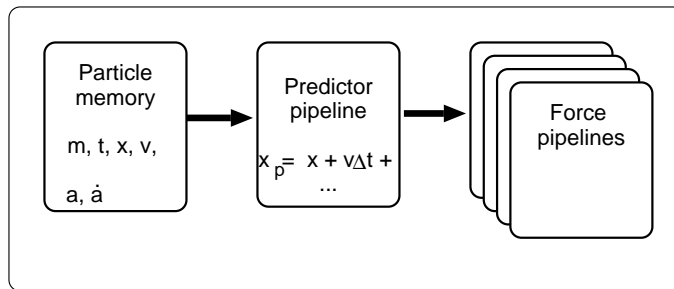


Figure 3: GRAPE for individual timestep

2.2 Project history

In 1989, we started the development of GRAPE systems. The first hardware, GRAPE-1, [IMES90] was a single-pipeline system with low-accuracy force calculation (r.m.s relative accuracy of 2%), without the predictor hardware. The force calculation pipeline was implemented using integer adder/subtractor IC chips and ROM chips (for lookup tables). It was the proof-of-the-concept machine to demonstrate that theoretical astrophysicists (we) could construct a real hardware. It had the peak speed of 240 Mflops, and turned out to be useful for a wide range of astrophysical problems where high accuracy is not very important.

GRAPE-2 [IEMS91] is again a single-pipeline system, but with high accuracy. The pipeline was implemented using commercial floating-point ALU chips such as TI 8847. It had the peak speed of 40 Mflops. For this machine, we did not include the predictor hardware, since with the blockstep scheme it was possible to perform prediction operations on the host computer without significant loss of the efficiency. Of course, this is simply because GRAPE-2 was not so fast.

GRAPE-3[OME⁺93] is our first multiple-pipeline system, where low-accuracy force calculation pipeline similar to that of GRAPE-1 was integrated into a custom chip. The GRAPE-3 chip was fabricated in $1\mu\text{m}$ rule and had about 100K transistors. The die size was 9mm by 9mm. It has single pipeline unit with operating clock frequency of 10 MHz, resulting in the speed of 380 Mflops/chip. The full GRAPE-3 system consisted of two boards with 24 chips each, for the aggregated peak speed of 18 Gflops. A smaller version of this machine, GRAPE-3A, which has 8 GRAPE-3 chips with the clock speed of 20 MHz, has been commercially available since 1993 and more than 100 GRAPE-3A boards have been manufactured and are used by a number of researchers worldwide.

GRAPE-4 (Makino *et al.* [MTES97]) is the first GRAPE hardware to implement the full predictor hardware with high accuracy. In GRAPE-4, one processor board houses one predictor units and 96 (virtual) force calculation pipelines. The total system in the maximum configuration consisted of 36 boards organized into four clusters, and different boards calculated the forces on the same set of 96 particles. In this way, we met the requirement that the number of forces calculated in parallel is small, even though the number of pipelines is large.

Summation of 9 forces from processor boards in the same cluster is taken care by the communication hardware, and final summation of the forces from four clusters is handled by

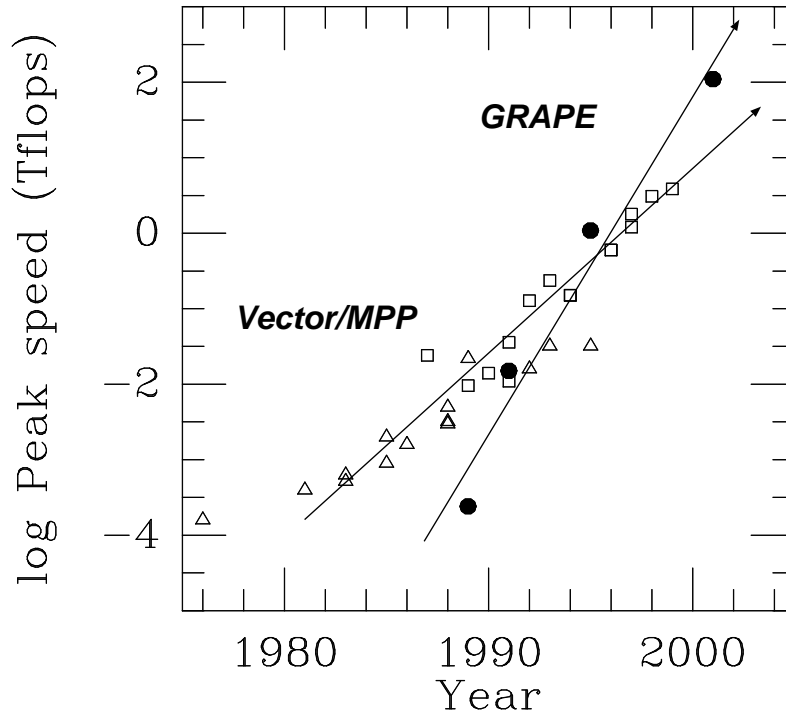


Figure 4: The evolution of GRAPE and general-purpose parallel computers. The peak speed is plotted against the year of delivery. Filled circles, open triangles and open squares denote GRAPEs, vector processors, and parallel processors, respectively.

the host.

A board houses 48 force calculation pipeline chips (GRAPE-4 chips) and one predictor pipeline chip (PROMETHEUS chip). GRAPE-4 chip can calculate one interaction in every three clock cycles, since it processes x , y , and z components in three consecutive clock cycles. It actually calculates the forces on two different particles, using six clock cycles. As a result, a GRAPE-4 chip logically looks like containing two pipelines operating at the clock frequency half of the actual one. We named this feature Virtual Multiple Pipeline (VMP). With VMP, we can reduce the memory bandwidth without sacrificing the performance.

GRAPE-4 was completed in 1995, and has been used by many researchers for the study of many astrophysical problems [HM99, MT98].

Figure 4 shows the peak speed of our GRAPE systems and that of representative high-performance general-purpose computers.

3 The GRAPE-6 system

3.1 Architecture

The GRAPE-6 system will consist of a host computer and 12 clusters. Each cluster has two I/O ports, one for local communication and the other for multicast through clusters. The multicast network is necessary to implement the individual timestep algorithm efficiently, but

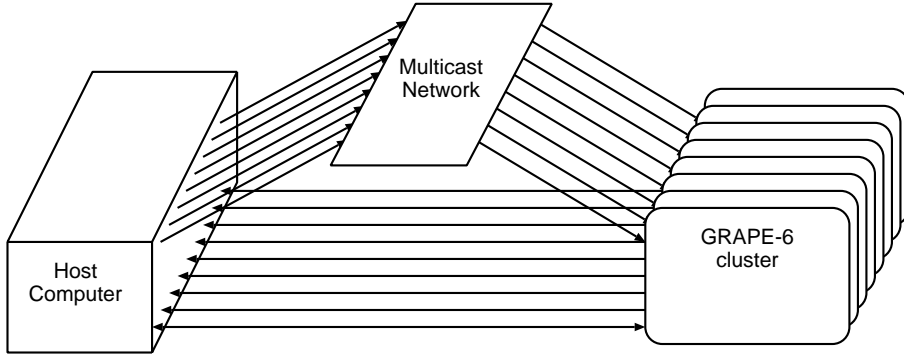


Figure 5: Overall structure of GRAPE-6 system. Eight clusters are shown.

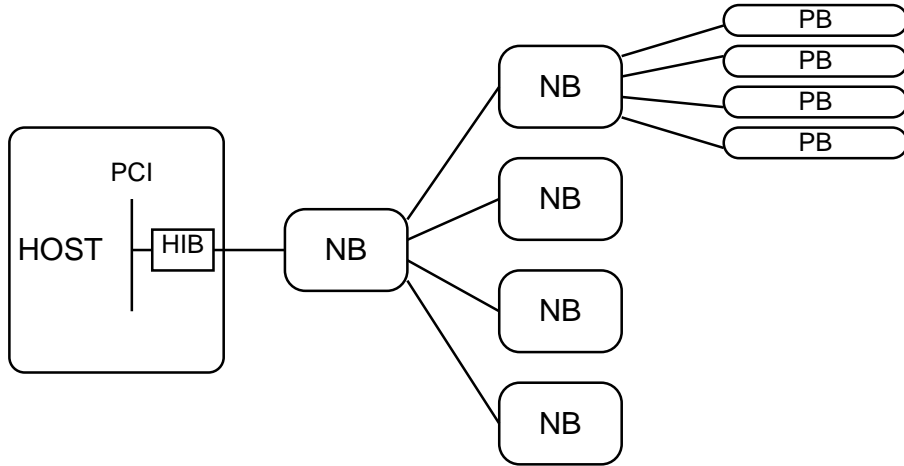


Figure 6: A GRAPE-6 cluster. First four PBs are shown. Other 12 PBs and interface to the multicast network are omitted.

in this paper we do not discuss the details, and concentrate on a single cluster. Figure 5 shows the overall structure of the GRAPE-6 system.

One cluster has one host-interface board (HIB), five network boards (NB), and 16 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 6). HIB and NB handles the communication between PBs and the host.

The PBs perform the force calculation. Each PB houses 16 GRAPE-6 processor chips, which are custom LSI chips to calculate the gravitational force and its first time derivative. Figure 7 shows the photograph of a PB. Two processor chips and four memory chips are mounted on a daughter card, and eight daughter cards are mounted on a processor board. Three large chips on board are the FPGA chips for the reduction network.

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

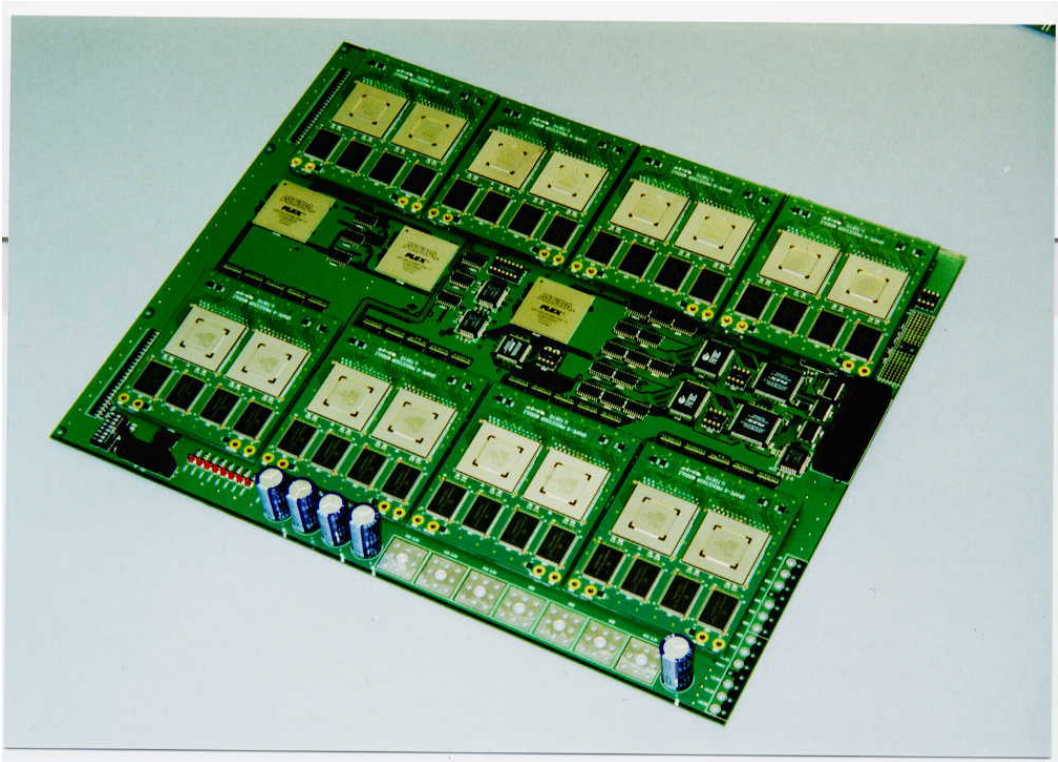


Figure 7: The GRAPE-6 processor board.

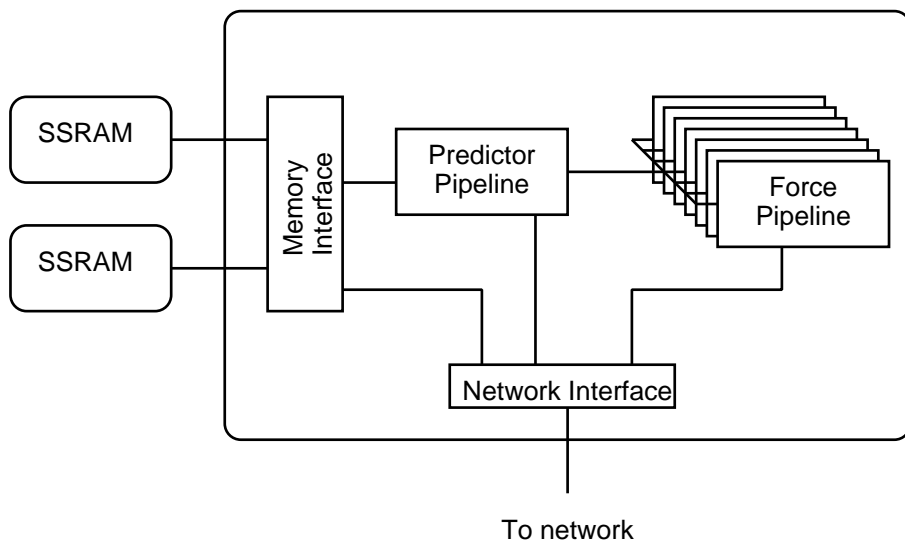


Figure 8: The GRAPE-6 processor chip.

(see figure 8). One force pipeline can evaluate one particle-particle interaction per cycle. With the present pipeline clock frequency of 88MHz, the peak speed of a chip is 30.1 Gflops. Here, we follow the convention of assigning 38 operations for the calculation of pairwise gravitational force, which is adopted in recent 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.

Each pipeline processor of GRAPE-6 calculates the force on eight particles simultaneously, using the virtual multiple pipeline (VMP) architecture introduced with GRAPE-4. In addition, six pipelines in one chip calculate the forces on six different sets of eight particles. Thus, one chip calculates the force on 48 particles. This architecture reduces the required memory bandwidth per chip drastically. The data stored for one particle is 64 bytes. Therefore, if six pipelines require different data, one chip requires 384 bytes per cycle, or 33.8GB/sec of the memory bandwidth. In our architecture, the bandwidth requirement is reduced by a factor of 48 to 0.7 GB/s, since the same data is reused 48 times to calculate force on 48 particles.

All processor chips in a cluster calculate the force on the same set of 48 particles, from different sets of particles. Thus, we need a fast network to first broadcast the 48 particles to all processor chips, and then summing up the result calculated on 256 chips in a cluster. The broadcast in hardware is pretty simple to implement in hardware. The summation network is implemented using FPGA chips as node points. Each FPGA chip implements a sequencer and 4-input adder. This FPGA and other circuits on board operate at 1/4 of the clock speed of the chip. The data width for the network on board is 32 bits. So the communication speed of any link in the network is 88MB/s.

For the link between boards, we have adopted a fast semi-serial link with LVDS signal level, which can achieve the above data rate with 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.

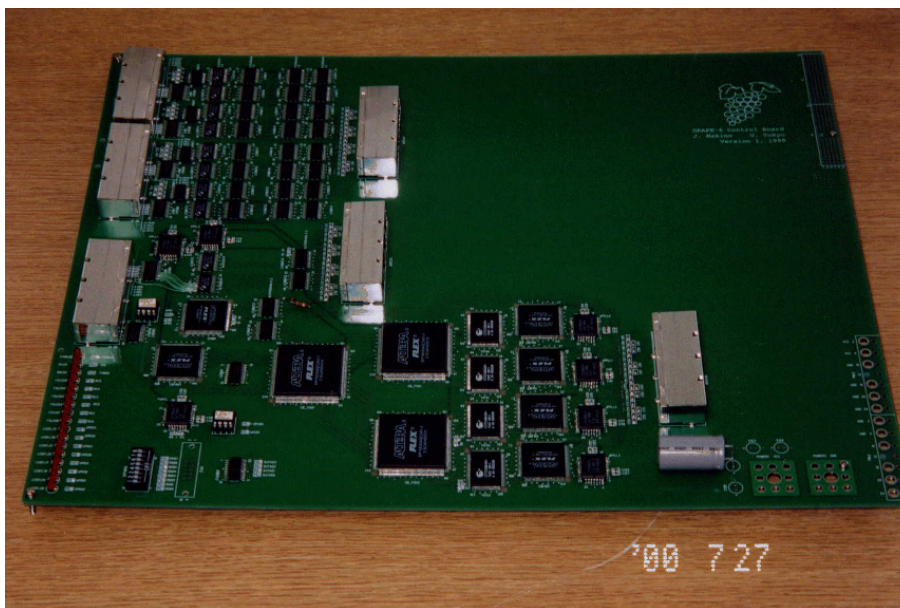


Figure 9: The GRAPE-6 network board.

The structure of a 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 9 shows the network board.

3.2 Difference from GRAPE-4

From the viewpoint of a user, the primary difference between GRAPE-4 and GRAPE-6 is the speed. The peak speed of GRAPE-6 (when completed) will be 100 times faster than that of GRAPE-4. In order to achieve a reasonable sustained performance, the peak communication bandwidth to the host computer is improved by a factor of 20. On the other hand, what is calculated is essentially the same for GRAPE-4 and 6. Both calculate the gravitational force and its time derivative, using predicted positions and velocities of particles stored in local memory on the side of the GRAPE hardware.

The 100-fold increase in the peak speed mainly comes from the performance improvement of the processor chip. The GRAPE-4 force calculation pipeline chip has single pipeline unit running at 32 MHz clock, which needs three clock cycles to evaluate one interaction. The GRAPE-6 chip has six pipelines running at 90 MHz, which can evaluate one interaction per clock cycle. Thus, GRAPE-6 chip is 50 times faster than GRAPE-4 chip.

The 18-times increase in the number of pipelines (from one third to six) was made possible by the miniaturization of the device (from $1\ \mu\text{m}$ to $0.25\ \mu\text{m}$). The improvement in the clock frequency also comes primarily from the same miniaturization of the device.

However, it should be noted that this improvement of a factor of 50 is achieved only through a drastic change in the overall architecture of the hardware. In GRAPE-4, a force calculation chip contains just one pipeline with no control logic. The particle data is fed from the predictor pipeline (PROMETHEUS chip), which also has a rather simple pipeline

architecture. The interface and control logic were implemented using FPGA chips on the processor board, and there is only one memory unit on a board.

In GRAPE-6, however, a processor chip integrates multiple force calculation pipelines, a predictor pipeline, a memory interface unit, control logics and communication interface. In other words, GRAPE-6 chip integrates one GRAPE-4 processor board. In fact, the architecture of the GRAPE-6 chip shown in figure 8 is identical to that of the GRAPE-4 processor board, except for details such as the number of pipelines and the width of the memory data bus.

This integration has various side effects, but the most important one from the viewpoint of the designer is that now one force on a particle is obtained as partial sums on a number of chips (in a full GRAPE-6 clusters, 256 chips on 16 processor boards), instead of just 9 chips in the case of GRAPE-4. In GRAPE-4, 9 processor boards are connected to a shared bus with the data width of 96 bits, and a control board sequentially reads the registers of chips on processor boards and accumulates the results using a floating-point adder chip.

With GRAPE-6, it is clear that we cannot take the same approach, since to accumulate 256 forces would take too long time. The only alternative is to construct an adder tree. The problem with the adder tree is that we need a rather large number of adders. an adder tree for 256 input would need 127 adders if constructed with 2-input adders. If we use a floating-point adder chip for the adder tree, the cost of the adder tree would be comparable or higher than the cost of the processor chips. Moreover, most of floating-point adder chips has become out of production. On the other hand, implementing 64-bit floating-point adders to FPGAs is not impossible, but would require rather large and expensive chips and throughput would be low.

To solve this difficulty, we chose to use fixed point number format for the accumulation of the force (and its time derivative as well). Since we have to handle a very wide range in distance and therefore in force, a naive use of the fixed point format would result in a very long data format to avoid overflow and underflow. To keep the length of the accumulator reasonably short, we decided to let each particle to have scaling coefficient of the force on it, which effectively works as the exponent for the calculated force. As a result, we can use fixed point adders both for the adder tree and the final accumulator of the force calculation pipeline. The host computer has to provide an appropriate scaling factor for each particle, but it can be easily calculated from the force on it at the previous timestep. Thus, the change in the application program is very small.

3.3 Development status

As of the time of writing, we have a 6-PB system, where three processor boards are connected to one network board. Two network boards are directly connected to the host computer without using the next level of network board. Figure 10 shows the structure of this system. This system has a theoretical peak speed of 2.889 Tflops. The host computer is a UP-2000 based Alpha (EV6, 667 MHz) box with two processors and 2 GB of memory, running Compaq Tru64 UNIX.

Figure 11 shows the photograph of the current six-board system. One can clearly see that it is a small system still in development phase. However, this small system has the theoretical peak speed close to three Teraflops. The only “general-purpose” machines that are faster than this small box are huge and expensive ASCII machines.

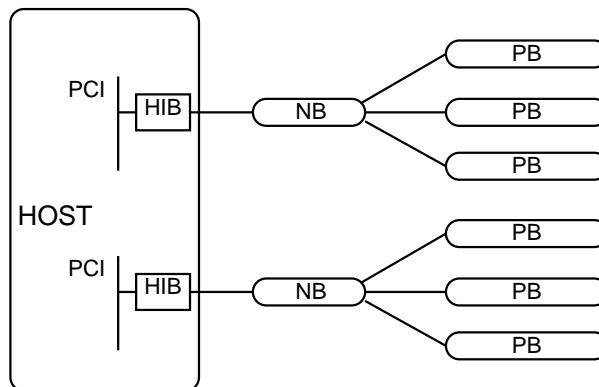


Figure 10: The present testbed GRAPE-6 configuration used for the simulation.

4 Simulation and Performance

We have simulated the evolution of a galactic nucleus containing triple massive black holes. As we stated in the introduction, currently available numerical simulations suggest that when two galaxies merge, their central black holes will be left as a binary system. Thus, when the remnant of a galaxy merger again merges with a third galaxy, the final merged object is likely to host three massive black holes.

The evolution of a triple black hole system is entirely different from that of a binary system, since 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 (gravitational slingshot, [SVA74]). The other possibility is that during the three-body interaction two black holes come close enough that they merge through gravitational wave radiation [HR92, ME94]. Thus, even after black holes have kicked out all nearby stars, they can still evolve in a complex way. On the other hand, a bound pair of two black holes is always dynamically stable. Thus, once a binary has kicked out all nearby stars, the evolution would be completely halted.

As we have described in the introduction, several numerical studies on the evolution of the massive black hole binary have been published, but to our knowledge there is no self-consistent N -body simulation of the black hole triples embedded in the galactic core, though the potential importance was first pointed out a number of years ago [SVA74] and studied in a number of papers both theoretically and numerically as a pure three-body problem in the fixed background potential (see [Val99] for a recent example).

The primary reason for this non-existence of the direct numerical work is the same difficulty of simulation as we described in the introduction for the case of the black hole binary. Though the evolution of the black holes might be different, the numerical difficulties are all the same. Since the difficulties are the same, the same solution, the combination of GRAPE hardware and individual timestep algorithm, works rather well for this problem.

In our simulation, the galaxy is modeled with 786,432 equal-mass stars. The black holes are modeled as three point-mass particles with a mass of 1% of the total mass of the system. The relativistic effects were neglected in our simulation. The gravitational force from black holes, both on field stars and on black holes, are calculated on the host computer to ensure



Figure 11: The six-board GRAPE-6 configuration used in this study. Eight processor boards are visible but only six were actually used in the calculation reported in this paper. The black box in front is the host computer (a two-processor Alpha box)

the best possible accuracy. The force from field particles is calculated on GRAPE-6.

We performed a simulation for 7.875 dynamical time units, for which the number of individual steps was 3.238×10^9 . The whole simulation, including on-the-fly analysis of the orbital elements of black holes and file operations, took 29.88 hours. The total number of floating point operations is $3.238 \times 10^9 \times 786,434 \times 57 = 1.451 \times 10^{17}$, since one particle-particle interaction amounts to 57 floating point operations. The resulting average computing speed is 1.349 Tflops.

The calculation reported here is the first large simulation of black hole triples in the core of the galaxy, and as such, the result turned out to be rather surprising. Figure 12 shows the evolution of the orbital elements for the most strongly bounded pair of binaries. The criterion of the selection of the pair is that the binding energy is the maximum. The binding energy of two black holes i and j is defined as

$$E_b = G \frac{M\mu}{r_{ij}} - \frac{1}{2} m v_{ij}^2, \quad (2)$$

where M is the total mass $m_i + m_j$, μ is the reduced mass defined as $\mu = m_i m_j / M$, and r_{ij} and v_{ij} are the relative distance and relative velocity of the pair. We regard this most strongly bounded pair as binary and plotted the time evolution of its semi-major axis and eccentricity. The evolution of the semi-major axis is not much different from those in the simulations of black hole binaries. However, the behavior of the eccentricity is quite different. In simulations of black hole binaries, it was found that the eccentricity remains almost constant during the evolution, and typically eccentricity is not very large. Thus, black holes do not come close to each other, and the effect of gravitational radiation remains small. In figure 12, however, we can see that the eccentricity e comes very close to 1 rather frequently (note that we plotted $1 - e$). The effect of the gravitational radiation is proportional to $(1 - e)^{-3.5}$. Thus, if $e = 0.99$, the effect of gravitational radiation is 10^7 times stronger than in the case of a circular binary. Thus, it is quite likely that this bound pair emits strong gravitational radiation resulting in the merging of two black holes.

Since the variation of the eccentricity is highly stochastic, we need more simulations from different initial conditions, and it is desirable to incorporate the effect of the gravitational radiation directly into the simulation to see if it actually works. However, our present result clearly demonstrates the importance of the triple interaction.

As a short benchmark run, we tried the same simulation with 1,400,000 stars for 0.25 time units. For this run, the total number of timesteps is 1.607×10^8 , and calculation took 2.17 hours. The average calculation speed was 1.64 Tflops.

5 Summary and Discussion

In this extended abstract, we present the performance achieved with astrophysical N -body simulations with individual timestep and direct force calculation on the GRAPE-6 special-purpose computer. The achieved performance number is 1.349 Tflops for a long calculation with 768k particles, and 1.64 Tflops for a short test calculation with 1.4 million particles.

The number of particles used is fairly large. Therefore, fast algorithms such as the Barnes-Hut tree algorithm might in principle outperform the direct summation algorithm we adopted here. However, as we've discussed in the introduction, there is no massively-parallel implementation of tree algorithm and individual timestep with accuracy high enough for this kind

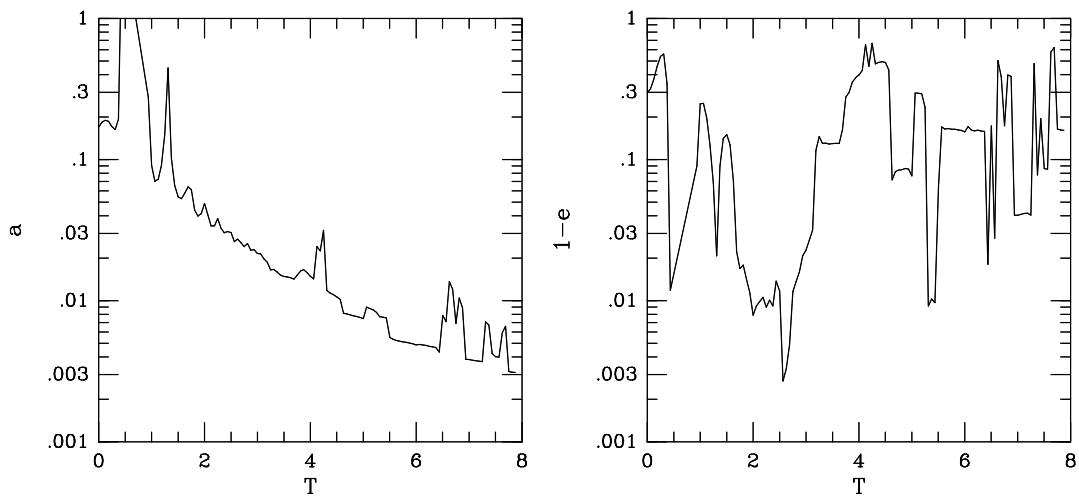


Figure 12: The time evolution of the semi-major axis (left panel) and eccentricity (right panel) of the most strongly bounded pair of black holes.

of problem. Therefore, we believe the speed we achieved is the fastest one can do with currently available combinations of hardware and software.

We thank Piet Hut for reading the manuscript. 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) Dynamical evolution of clusters of galaxies, i. *MNRAS*126: 223–255.
- [EOG99] Eckart A., Ott T., and Genzel R. (December 1999) The sgr a* stellar cluster: New nir imaging and spectroscopy. *A&A*352: L22–L25.
- [HM99] Hut P. and Makino J. (1999) Astrophysics on the grape family of special-purpose computers. *Science* 283: 501–505.
- [HR92] Hut P. and Rees M. J. (1992) Constraints on massive black holes as dark matter candidates. *MNRAS*259: 27P–30P.
- [IEMS91] Ito T., Ebisuzaki T., Makino J., and Sugimoto D. (1991) A special-purpose computer for gravitational many-body systems: Grape-2. *PASJ*43: 547–555.
- [IMES90] Ito T., Makino J., Ebisuzaki T., and Sugimoto D. (1990) A special-purpose n-body machine grape-1. *Computer Physics Communications* 60: 187–194.
- [JCR94] Jafry Y. R., Cornelisse J., and Reinhard R. (1994) Lisa - a laser interferometer space antenna for gravitational-wave measurements. *ESA Journal* 18: 219–228.

- [MA93] McMillan S. L. W. and Aarseth S. J. (1993) An $o(n \log n)$ integration scheme for collisional stellar systems. *ApJ*414: 200–212.
- [Mak91] Makino J. (1991) Optimal order and time-step criterion for aarseth-type n -body integrators. *ApJ*369: 200–212.
- [Mak97] Makino J. (1997) Merging of galaxies with central black holes. ii. evolution of the black hole binary and the structure of the core. *ApJ*478: 58+.
- [McM86] McMillan S. L. W. (1986) The vectorization of small- n integrators. 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) Triple black holes in the cores of galaxies. *ApJ*436: 607–610.
- [ME96] Makino J. and Ebisuzaki T. (1996) Merging of galaxies with central black holes. i. hierarchical mergings of equal-mass galaxies. *ApJ*465: 527–533.
- [MMH⁺95] Miyoshi M., Moran J., Herrnstein J., Greenhill L., Nakai N., Diamond P., and Inoue M. (January 1995) Evidence for a black-hole from high rotation velocities in a sub-parsec region of ngc4258. *Nature*373: 127+.
- [MT98] Makino J. and Taiji M. (1998) *Scientific Simulations with Special-Purpose Computers — The GRAPE Systems*. John Wiley and Sons, Chichester.
- [MTES97] Makino J., Taiji M., Ebisuzaki T., and Sugimoto D. (1997) Grape-4: A massively parallel special-purpose computer for collisional n -body simulations. *ApJ*480: 432–446.
- [OME⁺93] Okumura S. K., Makino J., Ebisuzaki T., Fukushige T., Ito T., Sugimoto D., Hashimoto E., Tomida K., and Miyakawa N. (1993) Highly parallelized special-purpose computer, grape-3. *PASJ*45: 329–338.
- [QH97] Quinlan G. D. and Hernquist L. (November 1997) The dynamical evolution of massive black hole binaries - ii. self-consistent n -body integrations. *New Astronomy* 2: 533–554.
- [Qui96] Quinlan G. D. (1996) The dynamical evolution of massive black hole binaries i. hardening in a fixed stellar background. *New Astronomy* 1: 35–56.
- [SCM⁺90] Sugimoto D., Chikada Y., Makino J., Ito T., Ebisuzaki T., and Umemura M. (1990) A special-purpose computer for gravitational many-body problems. *Nature* 345: 33–35.
- [SVA74] Saslaw W. C., Valtonen M. J., and Aarseth S. J. (June 1974) "the gravitational slingshot and the structure of extragalactic radio sources". *ApJ*190: 253–270.
- [TT72] Toomre A. and Toomre J. (December 1972) Galactic bridges and tails. *ApJ*178: 623–666.

- [Val99] Valtonen M. J. (July 1999) "jets, hot spots, and sidedness in double radio sources in the slingshot theory". *ApJ*520: 97–104.
- [vv98] van der Marel R. P. and van den Bosch F. C. (November 1998) Evidence for a $3 \times 10^8 m_{\odot}$ black hole in ngc 7052 from hubble space telescope observations of the nuclear gas disk. *AJ*116: 2220–2236.