GRAPE-4: A Special-Purpose Computer for Gravitational N-body problems

Junichiro Makino* Makoto Taiji† Toshikazu Ebisuzaki†
Daiichiro Sugimoto†

Abstract

We describe GRAPE-4, a special-purpose computer for gravitational N-body simulations. In gravitational N-body simulations, almost all computing time is spent for the calculation of interaction between particles. GRAPE-4 is a specialized hardware to calculate the interaction between particles. It is used with a general-purpose host computer that performs all calculations other than the force calculation. With this architecture, it is relatively easy to realize a massively parallel system. In 1991, we developed the GRAPE-3 system with the peak speed equivalent to 14.4 Gflops. It consists of 48 custom pipelined processors. In 1992 we started the development of GRAPE-4. The GRAPE-4 system will consist of 1920 custom pipeline chips. Each chip has the speed of 600 Mflops, when operated on 30 MHz clock. A prototype system with two custom LSIs has been completed July 1994, and the full system is now under manufacturing.

1 Introduction.

In this paper, we describe the GRAPE-4 system, a special-purpose computer for astrophysical N-body simulations. Many astrophysical objects are well approximated as a gravitational N-body system, in which point mass particles interact only through gravity. The solar system, open clusters, globular clusters, galaxies, clusters of galaxies, and superclusters are all well expressed as gravitational N-body systems.

The N-body simulation is the most powerful tool to investigate the behavior of these astrophysical objects. N-body simulations are, however, very expensive since the number of interactions is proportional to the square of the number of particles.

In the following, we give a few examples of the difference between what is possible with present computers and the real systems. Planets are believed to be formed from smaller planetesimals. It is believed that there were about 10^{10} planetesimals, while the number of particles used in simulations is 10^3 or less.

Globular clusters typically contain $10^5 - 10^6$ stars. However, the largest simulation ever tried for the long-term evolution of globular clusters is with only 10,000 particles. A typical galaxy contains $\sim 10^{11}$ stars, but simulations are usually performed with 10^5 or less stars. In the study of large scale structures of the universe, the number of stars considered would be 10^{16} or more, while simulations are usually performed with 10^6 or less particles.

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

†Department of Earth Science and Astronomy, College of Arts and Sciences, University of Tokyo, Tokyo 153, Japan

The reason why we can use much larger number of particles for simulations of galaxies compared to globular clusters is quite simple; the orbital period is quite different. The typical orbital timescale of stars in a galaxy is 10^8 years, while that of stars in globular cluster is 10^6 years. Thus we need at least 10^2 times as many timesteps for the simulation of globular clusters, compared to the simulation of galaxies.

There is another reason why we can use larger number of particles for simulation of galaxies or large-scale structures, compared to that of the planetesimals or globular clusters. If the required accuracy is not very high, it is possible to use the particle-mesh (PM) algorithm to calculate the potential field and then calculate the force on each particles (see, *e.g.*, [7]). However, for planetesimals or globular clusters, the PM scheme is not appropriate. The simulation of the long-term evolution of these systems requires very accurate calculation of the interparticle force, which is not practical with the PM scheme. Moreover, the number density and orbital timescale of stars in a globular cluster range over many orders of magnitude in a single cluster. As a result, a fixed resolution of the PM scheme is not appropriate. The tree algorithm does not help much, since the required accuracy of the force is high ([9],[17]).

For these problems, therefore, it is necessary to apply the straightforward algorithm, in which the gravitational force on a particle is honestly calculated by summing up the contributions from all other $N-1$ particles in the system. Thus, practically all N-body simulations of planetesimal systems and globular clusters have been performed using the direct force calculation.

In the rest of this paper, we describe the GRAPE-4 system, which is a special-purpose computer for astrophysical N-body simulation with the direct force calculation. In section 2, we describe the basic stricture of a GRAPE system and give a brief summary of our project. We describe the hardware design of GRAPE-4 in section 3, and the present status of the project in section 4. The peak speed of GRAPE-4 will be 1.15 Tflops, and the machine will be ready by early 1995. Section 5 is for discussion.

2 The GRAPE architecture

The direct force calculation is well suited for acceleration by specialized hardwares, because of the following two reasons. First, the force calculation itself is pretty simple. We calculate the difference in the positions of two particles $\Delta\mathbf{r}_{ij}$, and calculate the Newtonian gravitational force $-Gm_j\Delta\mathbf{r}_{ij}/|\Delta r_{ij}|^3$, where G is the gravitational constant and m_j is the mass of particle j. Total number of floating point operations involved is 17 plus one division and one square root operation. With the state-of-the-art VLSI technology, a fully hardwired pipeline to calculate the gravitational force can fit into one custom LSI chip.

Secondly, we can perform the rest of the calculation, such as the orbit integration of particles, on the general-purpose host computer, provided that the communication between the host computer and the specialized hardware is not too slow. This is because the force calculation dominates the total calculation cost.

Figure 1 shows the basic structure of a GRAPE system. GRAPE stands for Gravity Pipe. The host computer sends the positions and the masses of particles. GRAPE hardware calculates the gravitational accelerations and the potential energy of particles and sends it back to the host computer.

The simulation program itself runs on the host computer. The special-purpose hardware is used through the calls to a few library functions, which calculate the force between particles. It is not very difficult to modify existing programs so that it uses the special-

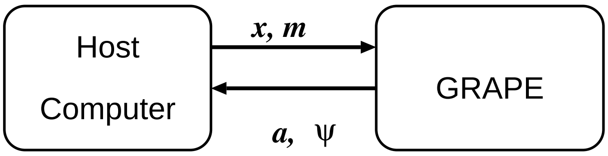

Fig. 1. *Basic structure of a GRAPE system*

purpose hardware, since the task is the same as to utilize a software library.

To apply the above idea for simulations of globular clusters or planetesimal, some modification of the hardware is necessary. The most widely used algorithm for simulation of these systems is the individual timestep scheme (for review, see [1]), in which each particle has it own time and timestep. The integration of the system proceeds as follows. First, one selects particle i for which its next time (present time plus present timestep $t_i + \Delta t_i$) is the minimum. Then, one predicts the positions of all particles at this $t_i + \Delta t_i$ and calculates the acceleration of particle i, applies the corrector based on this new acceleration and calculate the new timestep. Then one selects the next particle to integrate. A variable-stepsize predictor-corrector scheme is used to predict and correct the positions of particles. This scheme is extremely efficient when the cluster becomes highly inhomogeneous.

In order to use the individual timestep algorithm on the special-purpose hardware, one has to implement the predictor in hardware. This, again, is easy to implement since the predictor is just a polynomial of the time. A pipelined hardware to evaluate the polynomial is quite simple.

The first machine we build, GRAPE-1 [10], had one pipeline that operated on a 8 MHz clock. Its theoretical peak speed was equivalent to 240 Mflops. GRAPE-1 used the number format with the accuracy lower than that of the standard floating point, in order to reduce the hardware cost. This decision resulted in short development time, but its application is rather limited due to the low accuracy. We build GRAPE-2 [11] and GRAPE-2A [12], which use the standard floating point format. Their peak speed was 40 and 180 Mflops, respectively. The force calculation pipelines of these machines are assembled with off-the-shelf LSI chips.

For GRAPE-3 [18], we implemented the force calculation pipeline in a custom LSI chip. We again used the low-accuracy number format similar to the format used in GRAPE-1, because the budget was severely limited. The original GRAPE-3 board had 24 custom chips and its speed was equivalent to 7.2 Gflops. We build two boards that worked in parallel. Thus the peak performance of the GRAPE-3 was 14.4 Gflops. More than 20 copies of a minor-changed version of GRAPE-3, GRAPE-3A, have been made and are being used in more than 10 institutes both within and out of Japan.

In year 1992, we started the development of GRAPE-4. It will consist of 40 processor boards. Each processor board houses 48 pipeline chips. Thus the total system consists of 1920 chips and its planned peak speed is 1.15 Tflops. In the rest of this paper, we describe the architecture of GRAPE-4 and present status of the project.

3 The architecture of GRAPE-4

Figure 2 shows the overview of the GRAPE-4 system. The GRAPE-4 system consists of a host computer and five clusters. One cluster has one host-interface board, one control board, and eight processor boards. We use a DEC Alpha workstation as the host computer.

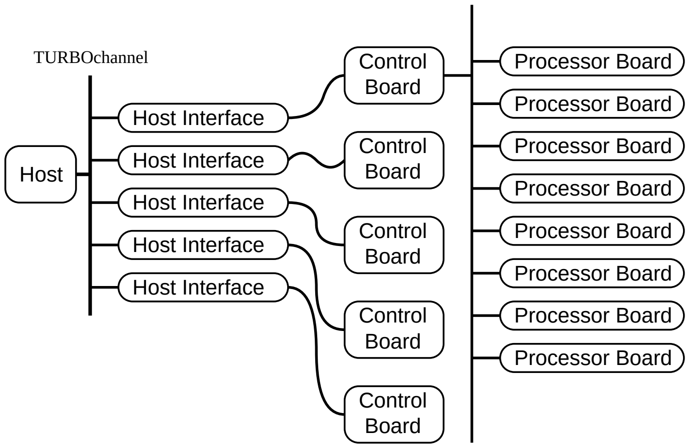

FIG. 2. *The GRAPE-4 system*

The host-interface board is connected to the host by TURBOchannel, DEC's own I/O bus. It's peak transfer rate is 100 MB/s. The host interface board and the control board are connected with a 32 bit parallel interface. Physical connection between them is a coaxial flat cable. The control board and processor boards are connected by a backplane bus with 96 bit data width. A simple synchronous protocol is used on the backplane bus.

Figure 3 shows the structure of a processor board. The particle data memory stores the data of particles which exert the force. The predictor LSI is used to calculate the position (and velocity) of particles at a specified time. This function is necessary to use the individual timestep algorithm ([1], [14]). The force calculation LSI chips calculate the forces on particles. Each chip calculates the forces on two particles, using the "virtual multiple pipeline" (VMP). With VMP, the clock period of the pipeline LSI is two times as that of the system clock, and it calculates the forces on two different particles at alternate clock cycles. From the outside, one force calculation chip looks as if it has two pipelines. The advantage of this architecture is that we can increase the performance of the pipeline chip without increasing the system clock cycle. The predictor chip is a gate array and the pipeline chip is cell-based. Both are fabricated by LSI Logic Corp. The design rule is 1 μm and the gate count is about 50k and 100k for predictor and force calculation chips, respectively.

One board has 48 pipeline chips. These chips are packaged in an MCM (multi-chip module) package made by Kyocera. Thus one board has 6 MCM packages. Since each chip calculates the force on two particles, one board calculates the forces on 96 different particles. In the original individual timestep algorithm, we can calculate the force on only one particle at a time. In order to allow multiple particles to be advanced at the same time, we use a hierarchical timestep algorithm [14], in which the timesteps of particles are forced to powers of two.

The control board has two main functions. The first one is to distribute the data received from the host computer to processor boards. The second one is to sum up the calculated force and potential from processor boards. The summed result is transferred to the host computer through the host interface board. This summation process is necessary to reduce the bandwidth of the communication with the host computer [16].

The theoretical peak speed is 1.15 Tflops, for the system clock of 15 MHz and the pipeline clock of 30 MHz. Half of the peak speed will be achieved for the simulation of 5×10^4 particles.

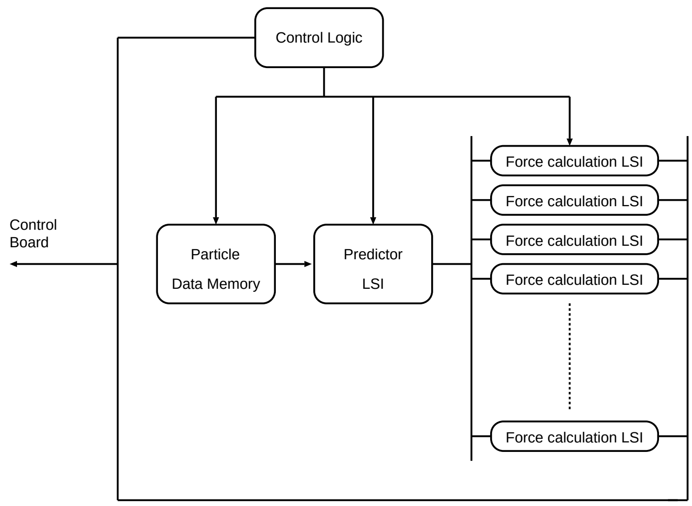

Fig. 3. *Processor board of GRAPE-4*

4 Status of the Project.

We started the design of the custom chips in July 1992. The first engineering samples of the custom chips were delivered in August 1993. We confirmed the normal operation of both chips with 50 MHz clock for the pipeline chips (25 MHz for the board). As of September 1994, the prototype system which consists of one host interface board, one control board and one processor board has been completed and is working as designed.

5 Discussion

Our GRAPE project is not the first project to develop special-purpose computers for N-body simulations. Bakker et al. ([4], [3]) developed DMDP, and Fine et al. [6] developed FASTRUN. These machines are specialized to MD simulations.

DMDP and FASTRUN have similar architectures. Both have the force calculation pipeline similar to one pipeline of GRAPE-4. The peak speed is also similar (100-200 Mflops). However, both of DMDP and FASTRUN have additional hardware for time integration of particles, as well as hardware support for sophisticated algorithms such as linked-list or neighbor scheme. As a result, both machines are very large, comprising 10-20 large printed-circuit boards. In addition, the time needed to develop the hardware was rather long ($\sim$ 5 years). Our GRAPE-4 system will take three years, which is rather long because it is our first massively-parallel machine. Smaller machines such as GRAPE-3 took less than 1 year to develop.

The long development time has a large negative impact on the performance of any computer, since the speed of general-purpose computer doubles roughly in every one and half years. Thus, five years corresponds to about a factor of 10 in performance.

We believe that the lifetime of a special-purpose machine is longer if it designed to work in cooperation with the host computer, compared to a stand-alone design in which everything is done on the special-purpose computer. The design of the hardware would be much simpler and therefore faster. The cost of the hardware would be cheaper. The total performance might be limited by the speed of the host computer. This limitation, however, also implies that the total performance of the system would be improved as the faster host computer become available.

If we compare our approach with the traditional approach of using fast general-purpose computer, it is clear that our approach is better, at least for gravitational N-body simulations that require relatively high accuracy. GRAPE-4 offers 100-1000 times faster raw performance compared to general-purpose computers. It is true that on general-purpose computers one can use more sophisticated algorithm to improve the speed. For example, one can use the neighbor scheme, in which the force on a particle is divided into the force from neighbor particles and that from distant particle, and the latter is updated in longer timestep [2]. The theoretical gain by the neighbor scheme is $O(N^{1/4}$ ([8], [15]), which is relatively small. In addition, the actual gain in the computer time is smaller than the theoretical factor, because the memory access pattern of the neighbor scheme is not regular. Thus, the advantage of the specialized hardware is large enough to overcancell the algorithmic loss. In the case of the Barnes-Hut tree algorithm, we can apply the vectorized algorithm developed by Barnes [5] to use GRAPE [13].

References

[1] S. J. Aarseth, in Multiple Time Scales, Academic Press, New York, 1985.

[2] A. Ahmad and L. Cohen, *A Numerical Integration Scheme for the N-Body Gravitational Problem*, J. Comp. Phys., 12 (1973) pp. 389-402.

[3] A. F. Bakker and C. Bruin, *Design and Implementation of the Delft Molecular-Dynamics Processor*, in Special Purpose Computers, Academic Press, New York, 1988.

[4] A. F. Bakker, C. Bruin, F. van Dieren, and H. J. Hilhorst, *MOLECULAR DYNAMICS OF 16000 LENNARD-JONES PARTICLES*, *Phys. Lett.*, 93A (1982) pp. 67-69, 1982.

[5] J. Barnes, *A Modified tree code: Don't laugh, it runs*, J. Comp. Phys., 87 (1990) pp. 161.

[6] R. Fine, G. Dimmer and C. Levinthal, *FASTRUN: A special purpose, hardwired computer for molecular simulation*, Proteins, 11 (1991) pp. 242-253.

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

[8] P Hut, and J. Makino, *Performance analysis of direct N-body calculations,* Astrophys. J. Suppl., 68 (1988) pp. 833-856.

[9] P Hut, J. Makino, and S. L. W. McMillan, *Modeling the evolution of globular star clusters*, Nature, 336 (1988), pp. 31-35.

[10] T. Ito, J. Makino, T. Ebisuzaki and D. Sugimoto, *A special-purpose N-body machine GRAPE-1*, Computer Physics Communications, 60 (1990) pp. 187-194.

[11] T. Ito, J. Makino, T. Ebisuzaki and D. Sugimoto, *A special-purpose computer for gravitational many-body systems GRAPE-2*, Publ. Astron. Soc. J., 43 (1991) pp. 547-555.

[12] T. Ito, J. Makino, T. Fukushige, T. Ebisuzaki, S. K. Okumura and D. Sugimoto, *A Special-Purpose Computer for N-body Simulations: GRAPE-2A*, Publ. Astron. Soc. J., 45 (1993) pp. 339-348.

[13] J. Makino, *Treecode with a special-purpose processor*, Publ. Astron. Soc. J., 43 (1991) pp. 621-638.

[14] J. Makino, *A modified Aarseth code for GRAPE and vector processors*, *Publ. Astron. Soc. J.*, 43 (1991) pp. 859-876.

[15] J. Makino and S. J. Aarseth, *On a Hermite Integrator with Ahmad-Cohen Scheme for Gravitational Many-Body Problems*, Publ. Astron. Soc. J., 44 (1992) pp. 141-151.

[16] J. Makino, E. Kokubo and M. Taiji, *HARP: A Special-Purpose Computer for N-body Problem*, Publ. Astron. Soc. J., 45 (1993), pp. 349-360.

[17] S. L. W. McMillan and S. J. Aarseth, *An $O(N \log N)$ integration scheme for collisional stellar systems*, Astrophys. J., 414 (1993) pp. 200.

[18] S. K. Okumura, J. Makino, T. Ebisuzaki, T. Fukushige, T. Ito, D. Sugimoto, *Highly Parallelized Special-purpose Computer GRAPE-3*, Publ. Astron. Soc. J., 45 (1993), pp. 329-338.