

GRAPE AND PROJECT MILKYWAY

JUNICHIRO MAKINO

Department of Astronomy, University of Tokyo, Tokyo, 133-0031, Japan

E-mail: makino@astron.s.u-tokyo.ac.jp

(Received February 1, 2005; Accepted March 15, 2005)

ABSTRACT

We overview the GRAPE (GRAVity piPE) project. The goal of the GRAPE project is to accelerate the astrophysical N -body simulations. Since almost all computing time is spent for the evaluation of the gravitational force between particles, we can greatly accelerate many N -body simulations by developing a specialized hardware for the force calculation. In 1989, the first such hardware, GRAPE-1, was completed, with the peak speed of 120 Mflops. In 2003, GRAPE-6 was completed, with the peak speed of 64 Tflops, which is nearly 10^6 times faster than GRAPE-1 and was the fastest computer at that time. In this paper, we review the basic concept of the GRAPE hardwares, the history of the GRAPE project, and two ongoing projects, GRAPE-DR and Project Milkyway.

Key words : celestial mechanics—methods: N -body simulations

I. INTRODUCTION

The N -body simulation technique, in which the equations of motion of N particles are integrated numerically, has been one of the most powerful tools for studying astronomical objects, such as the solar system, star clusters, galaxies, clusters of galaxies, and large-scale structures of the universe.

From the theoretical point of view, N -body simulations can be classified into two categories: simulations of collisional systems and that of collisionless systems. In the case of collisional simulations, the evolution of the system is driven by the two-body relaxation process.

An example of such collisional systems is a star cluster, which consists typically of 10^4 to 10^6 stars. Their dynamical timescale is relatively short. For example, the dynamical timescale of galactic globular clusters is around 1 Myrs. Thus the relaxation timescale is typically around 1 Gyrs, and the two-body relaxation process drives the evolution of star clusters.

If the evolution of the system is driven mainly by the two-body relaxation with the timescale much longer than the dynamical timescale, it sounds like a waste of time and computing power to model them directly by means of N -body simulations. If we can construct an appropriate stochastic model which describes the slow change of the distribution function, we can, in principle, use the timestep of the order of the relaxation timescale, and there seems to be no need for expensive N -body simulations.

The direct numerical integration of the Fokker-Planck (FP) equation (Cohn 1979) is one of such methods that

model the slow change of the distribution function. It is a very useful tool to study the evolution of idealized clusters, such as the problem of the core collapse of isolated cluster made of equal-mass particles and gravothermal oscillations.

However, FP models have their limitations. For example, until recently it can only handle spherically symmetric systems with isotropic distribution functions. Takahashi (1996) made it possible to integrate the system described by $f(E, J)$, that is, spherically symmetric system with anisotropic velocity distribution, and Einsel & Spurzem (1999) introduced the system with $f(E, J_z)$. Real star clusters in the galactic tidal field have three-dimensional structure, which requires full six-dimensional description of the phase space. Whether or not such an FP model is practical, or even possible, is not known.

Even for simple and idealized systems, without the help of N -body simulations it is quite difficult to calibrate more approximate models. On the other hand, if N -body simulations with necessary number of particles are too expensive, we have to live without them.

The exponential increase in the computer power in the last half century helped us to make N -body simulations practical. However, the use of direct N -body simulations in collisional stellar dynamics is rather unique in computational science. A significant fraction of large-scale simulations have been done on a series of special-purpose computers designed and built specifically for collisional stellar dynamics by one group of astrophysicists. We call the hardware we developed “GRAPE”, from GRAVitational Pipe. Here, “pipe” stands for the pipelined processor for gravitational interaction calculation.

In the rest of this paper, we briefly overview the past history of the GRAPE project and two ongoing projects, GRAPE-DR project and Project Milkyway.

Proceedings of the 6th East Asian Meeting of Astronomy, held at Seoul National University, Korea, from October 18-22, 2004.

II. GRAPE PROJECT

(a) Basic Concept

The gravitational N -body problem is expressed by the following set of equations of motion:

$$\frac{d^2 \mathbf{x}_i}{dt^2} = - \sum_{j \neq i, 1 \leq j \leq N} G m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3}, \quad (1)$$

where \mathbf{x}_i and m_i are the position and mass of particle i , G is the gravitational constant, and N is the number of stars. With the simple direct summation method the calculation cost of the right hand side of the equation scales as $O(N^2)$, and dominates the calculation cost for $N > 10$.

For applications where low accuracy is acceptable, one can use fast methods like Barnes-Hut tree algorithm (Barnes & Hut 1986) or FMM (Greengard & Rokhlin 1987). However, for many collisional simulations, it is hard to apply these algorithms (Makino et al. 2003).

In our GRAPE project, we accelerate the calculation of gravitational interaction between particles by developing a computer specialized for that operation.

Figure 1 shows the basic structure of a GRAPE system. The calculation of the interaction between stars is handled by the special-purpose computer, while all other calculations, such as the time integration of stars, I/O, analysis and diagnostics are handled by a host computer. For the host computer, we used either a UNIX-running workstation or a PC (usually with Linux).

This hybrid architecture has several important advantages. First of all, since the special-purpose part is dedicated to a single function, we can use a highly optimized architecture for that part. From the user's point of view, even more important is that most part of his/her program still runs on the general-purpose computer. Therefore, it is relatively easy to develop programs and it is also possible to adopt existing programs.

Figure 2 shows the basic structure of a GRAPE pipeline processor. It is designed to perform the calculation of the gravitational forces from particles (with indices j) to particle i , in other words, the right hand side of equation (1).

This pipeline architecture means we can use a large

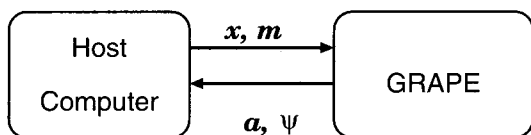


Fig. 1.— Basic concept of a GRAPE system.

number of arithmetic units in parallel. In a general-purpose computer, it is difficult to integrate more than a few arithmetic unit in a chip, and circuits for the floating-point arithmetic unit usually consume very small fraction of the total transistor count. For example, latest microprocessors such as Intel Pentium 4 or Itanium 2 has around 10^8 transistors, while a floating point arithmetic unit requires around 10^5 transistors. Thus, around 99.9% of all transistors are used for something other than the arithmetic units. These include caches, control logic and bus interfaces etc.

With the GRAPE pipeline architecture, it is easy to implement one or even multiple pipeline processors into a single chip, resulting in hundreds of arithmetic units in a chip, utilizing almost all available transistors for arithmetic units. Thus, compared to general-purpose microprocessors, a custom-design processor with GRAPE architecture can perform nearly 1,000 times more floating point operations per clock cycle. Even if the clock speed is slower by more than a factor of 10, a GRAPE processor would still be much faster than a general-purpose microprocessor. Thus, overall performance will be much higher, if the production cost of the chip is not too higher than that of microprocessors.

(b) Project History

We started the development of GRAPE-type machines in 1989. The first machine, GRAPE-1, was an experimental hardware with a very short word format (relative force accuracy of 5% or so), and was not really suited for simulations of collisional systems. However, its exceptionally good cost-performance ratio made it useful for simulations of collisionless systems (Okumura et al. 1991; Funato et al. 1992). Also, we developed an algorithm to accelerate the Barnes-Hut tree algorithm using GRAPE hardware (Makino 1991), and developed GRAPE-1A (Fukushige et al. 1991), which was designed to achieve good performance with the treecode. Thus, the GRAPE approach turned out to be quite effective, not only for collisional simulations, but also for collisionless simulations as well as SPH simulations (Umemura et al. 1993; Steinmetz 1996). GRAPE-1A and its successors, GRAPE-3 (Okumura et al. 1993) and GRAPE-5 (Kawai et al. 2000), have been used by

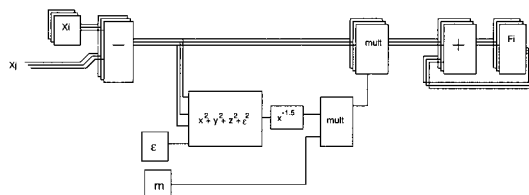


Fig. 2.— Basic structure of a GRAPE pipeline processor.

researchers worldwide for many different problems.

GRAPE-2 (Ito et al. 1991) is the first machine for collisional stellar dynamics. It was constructed using LSI chips for floating-point arithmetic operations, and had the peak speed of 40 Mflops. GRAPE-4 (Makino et al. 1997) was a single-LSI implementation of GRAPE-2, or actually that of HARP-1 (Makino et al. 1993), which was designed to calculate force and its time derivative. A single GRAPE-4 chip calculated one interaction in every three clock cycles, performing 19 operations. Its clock frequency was 32 MHz and peak speed of a chip was 608 Mflops.

In 2002, we completed GRAPE-6 (Makino et al. 2003). It is a direct successor of GRAPE-4, with main difference in the speed. The processor chip of GRAPE-6 integrates six force-calculation pipelines, each of which can calculate one interaction per clock cycle. The clock frequency of GRAPE-6 is 90 MHz. Thus, a single GRAPE-6 chip is around 50 times faster than a single GRAPE-4 chip. The total system with 2,048 chip offers the theoretical peak speed of 64 Tflops.

The concept of GRAPE can be applied to other field of science. An obvious example is classical molecular dynamics, where the classical equations of motion for atoms interacting with other atoms are integrated numerically. There were a number of attempts to develop special-purpose computers for molecular dynamics, and some of them used the pipeline architecture rather similar to GRAPE pipeline (Bakker & Bruin 1988; Fine et al. 1991). Unfortunately, none of them were very successful, primarily because the hardware design was too complex and application range was too limited. For the discussion on their problems, see Makino & Taiji (1998). We also applied the GRAPE architecture to molecular dynamics, starting with GRAPE-2A (Ito et al. 1994). It was followed by the custom-chip version, MD-GRAPE (Fukushige et al. 1996), and then by massively parallel MDM (Narumi et al. 1999). MDM and GRAPE-6 had the peak speed and efficiency rather close to each other, and in year 2001 small prototype versions of them shared the Gordon Bell Prize for performance. Currently, an even faster Protein Explorer is under development (Taiji et al. 2003).

III. GRAPE-DR

In July 2004, we were awarded the grant to develop the next-generation GRAPE system, which we call GRAPE-DR. The “DR” part comes from the Data Reservoir project, the goal of which is to develop an infrastructure for the data sharing over fast long-distance network. In this section we briefly outline the goal and architecture of GRAPE-DR.

(a) The Goal

The goal of GRAPE-DR project is to develop a massively-parallel computer for scientific applications, with the peak speed of 2 Petaflops by the end of FY

2008. The application range is not limited to astrophysical N -body problem. In fact, it is not even limited to particle-based simulations.

This of course means we have changed the basic architecture of the hardware. Instead of the hardwired pipeline designed to calculate the Newtonian gravity between particles, we use a large number of very simple processors connected through a simple network. Each processor consists of floating-point adder/subtractor, floating-point multiplier, an integer arithmetic unit and a register file. These processors “emulate” the GRAPE pipeline processors in software.

One might think that this move from hardwired pipeline to programmable processor array is against the concept we described in the previous section, which is certainly not entirely wrong.

With the current GRAPE-DR design, only about 1/3 of the total transistors is used for the arithmetic units. In addition, each processor has both an adder and a multiplier, but in many applications only one of them is used for most of the time. Thus, we lose at least a factor of five or so in the efficiency of the use of the transistors.

The reason why we decided to change the processor architecture is primarily economical rather than technological. The initial development cost of a custom chip has become too high to be covered by a budget for a research project within astronomy. To be specific, the initial cost of the chips for GRAPE-3, 4 and first MD-GRAPE was around 200K USD. With GRAPE-6, it becomes more than 1M USD. Right now manufacturers would require 5M USD or more. As the miniaturization of the transistors continue, the design and fabrication become more and more difficult, which resulted in the higher initial cost.

There are three possible solutions for this problem (maybe more). One is to try to convince the funding agency that the research project is worth the money. This option does not always work. The second option is to reduce the initial cost by adopting new design methods like the use of large FPGA (Field Programmable Gate Array) or structured ASIC. These new methods offer much lower initial cost, but with much higher mass-production cost (limitation in the maximum available size and intrinsically low transistor usage). The third option is to widen the application range. For example, making the pipeline processor which is good for gravity, molecular-dynamics applications, and other applications such as boundary-element methods is certainly possible. In this way, asking for a large enough funding becomes somewhat easier than with the first option.

A practical problem with the third option is that designing a pipeline processor which can be used for multiple applications is very difficult. We need a designer who knows all details of all target applications. Instead, we decided to make the processors programmable, so that the chip designer need not know every

details of every application. Of course, as we stated above, this decision results in the performance hit of around a factor of five, but that would still probably be better than what we can get with the second solution. Also, the system might turn out to be useful for the applications we have not even thought of.

The current plan for GRAPE-DR is to complete the machine by the end of FY 2008 (March 2009), with the theoretical peak speed of 2 Petaflops. The machine will consist of 2048 processor chips, each with around 1024 simple processors. The processor chip will be fabricated using 90nm technology (similar to that used for microprocessors available in 2005). Clock frequency will be around 1 GHz or less. With this modest clock frequency the power consumption will not be too large, around 30W per chip.

IV. PROJECT MILKYWAY

Project Milkyway is a new project initiated by Wada and Tomisaka of NAOJ and several others, including the author. The goal of the project is to model the formation and evolution of a typical spiral galaxy (namely the Milkyway galaxy) with the resolution much higher than previously tried. Currently, state-of-the-art Nbody+SPH simulation of galaxy formation typically uses around 10^5 particles (gas and dark matter), which means the mass of one particle is around $10^6 M_\odot$. With such a low resolution, clearly we cannot resolve the thin gas disk, spiral arms and star forming regions, but without modeling these structures, it is certainly difficult to understand the evolution of galaxies.

The goal of the project (in five years) is to improve the mass resolution by at least three orders of magnitude. The GRAPE-DR system will provide the necessary computing power, but there are many scientific/technological challenges. To name just a few, the treatment of radiation, better treatment of star formation, development of efficient and accurate algorithm with individual timestep and so on.

GRAPE has been used in many of Nbody+SPH simulations of the formation of galaxies (Hut & Makino 2003). In these calculations, since GRAPE made the calculation of gravity much faster, remaining calculations such as the SPH interaction between gas particles and the calculation of chemical equilibrium becomes the bottleneck. We believe the programmable architecture of GRAPE-DR will enable us to accelerate these calculations and thus open a new horizon in the theoretical understanding of galaxy formation.

REFERENCES

- Bakker, A. F., & Bruin, C., 1988, in *Special Purpose Computers*, ed. B. J. Alder (San Diego: Academic Press), 183–232
- Barnes, J., & Hut, P., 1986, *Nature*, 324, 446
- Cohn, H., 1979, *ApJ*, 234, 1036
- Einsel, C., & Spurzem, R. 1999, *MNRAS*, 302, 81
- Fine, R., Dimmler, G., & Levinthal, C., 1991, *PROTEINS: Structure, Function, and Genetics*, 11, 242
- Fukushige, T., Ito, T., Makino, J., Ebisuzaki, T., Sugimoto, D., & Umemura, M., 1991, *PASJ*, 43, 841
- Fukushige, T., Taiji, M., Makino, J., Ebisuzaki, T., & Sugimoto, D., 1996, *ApJ*, 468, 51
- Funato, Y., Makino, J., & Ebisuzaki, T., 1992, *PASJ*, 44, 613
- Greengard, L., & Rokhlin, V., 1987, *Journal of Computational Physics*, 73, 325
- Hut, P., & Makino, J., eds., 2003, *Astrophysical Supercomputing using Particle Simulations* (San Francisco: ASP)
- Ito, T., Ebisuzaki, T., Makino, J., & Sugimoto, D., 1991, *PASJ*, 43, 547
- Ito, T., Fukushige, T., Makino, J., Ebisuzaki, T., Okumura, S. K., Sugimoto, D., Miyagawa, H., & Kitamura, K., 1994, *PROTEINS: Structure, Function, and Genetics*, 20, 139
- Ito, T., Makino, J., Ebisuzaki, T., & Sugimoto, D., 1990, *Computer Physics Communications*, 60, 187
- Kawai, A., Fukushige, T., Makino, J., & Taiji, M., 2000, *PASJ*, 52, 659
- Makino, J., 1991, *PASJ*, 43, 621
- Makino, J., Fukushige, T., Koga, M., & Namura, K., 2003, *PASJ*, 55, 1163
- Makino, J., Fukushige, T., Okumura, S. K., & Ebisuzaki, T., 1993, *PASJ*, 45, 303
- Makino, J., & Taiji, M., 1998, *Special Purpose Computers for Scientific Simulations – The GRAPE systems* (Chichester: John Wiley and Sons)
- Makino, J., Taiji, M., Ebisuzaki, T., & Sugimoto, D., 1997, *ApJ*, 480, 432
- Narumi, T., Susukita, R., Ebisuzaki, T., McNiven, G., & Elmegreen, B., 1999, *Molecular Simulation*, 21, 401
- Okumura, S. K., Ebisuzaki, T., & Makino, J., 1991, *PASJ*, 43, 781
- Okumura, S. K., Makino, J., Ebisuzaki, T., Fukushige, T., Ito, T., Sugimoto, D., Hashimoto, E., Tomida, K., & Miyakawa, N. 1993, *PASJ*, 45, 329
- Steinmetz, M., 1996, *MNRAS*, 278, 1005
- Taiji, M., Narumi, T., Ohno, Y., Futatsugi, N., Suenaga, A., Takada, N., & Konagaya, A., 2003, in *The SC2003 Proceedings* (Los Alamitos, CA: IEEE), CD-ROM
- Takahashi, K., 1996, *PASJ*, 48, 691
- Umemura, M., Fukushige, T., Makino, J., Ebisuzaki, T., Sugimoto, D., Turner, E. L., & Loeb, A., 1993, *PASJ*, 45, 311