Dynamical Evolution of Dense Stellar Systems
Proceedings IAU Symposium No. 246, 2007
E. Vesperini, M. Giersz & A. Sills, eds.

doi:10.1017/S1743921308016165

Current Status of GRAPE Project

J. Makino

Division of Theoretical Astronomy, National Astronomical Observatory of Japan, 2-21-1
Osawa, Mitaka, Tokyo, 181-8588
email: makino@cfca.jp

Abstract. I'll summarize the current status of GRAPE project. GRAPE-6, completed in 2002, has been used by a number of people, for a wide variety of problems such as planet formation, star cluster dynamics, galactic nuclei, and cosmology. In 2004, we started the development of the next-generation machine, GRAPE-DR. GRAPE-DR has a architecture radically different from that of previous GRAPEs. It does not have hardwired pipeline for gravitational force calculation but a large number of small and simple programmable processors. This change made it possible to apply GRAPE-DR to a wide range of problems to which GRAPE was not efficient, and at the same time it helps us to explore new algorithms for N-body simulations. The GRAPE-DR chip was completed in 2006, and second prototype board was completed in May 2007. We hope to have full production-level board commercially available by the end of year 2007. A single board will offer the theoretical peak speed of 2 Tflops, about 20 times as that of a single PCI card version of GRAPE-6.

Keywords. methods: n-body simulations, globular clusters: general

1. 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 the study of astronomical objects, such as the solar system, star clusters, galaxies, clusters of galaxies, and large-scale structures of the universe.

In particular, for the study of the dynamical evolution of star clusters, N-body simulation is now an essential tool. There are certainly faster methods like Monte-Carlo or direct integration of the Fokker-Planck equation. However, they need a number of assumptions and simplifications, which need to be tested and justified through the comparison with the result of N-body simulations. On the other hand, the main limitation of N-body simulations is their high computational cost. Calculation cost scales as $O(N^{3.3})$, where N is the number of particles. The calculation cost per time step is $O(N^2)$, if we use simple direct summation. We have additional power of 1/3, since the average size of the time steps must be small enough so that one particle would not move the distance larger than the distance to its nearest neighbor. The final power of 1 comes from the ratio between the relaxation timescale and dynamical timescale.

In the last two decades, we have developed a series of special-purpose computers, GRAPE (GRAvity piPE). The basic idea behind the GRAPE system is to build specialized hardwares which calculate the gravitational interaction between particles, and connect then to usual general-purpose computers. All calculations other than the calculation of interactions are done on the side of the general-purpose computers. Since most of the computing time is spent on the calculation of interactions, we can accelerate the overall calculation just by accelerating the interaction calculation, and the high-performance hardware which calculates only the gravitational interaction is relatively easy to design.

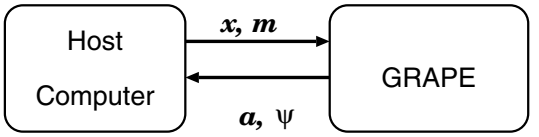

Figure 1. Basic concept of a GRAPE system.

In the rest of this paper, we briefly overview the GRAPE project and the ongoing GRAPE-DR project.

2. GRAPE project

2.1. *Basic Concept*

In our GRAPE project, we accelerate the calculation of gravitational interaction between particles by developing a computer specialized for that operation. Fig. 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 very 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. For GRAPE designs from GRAPE-1 (Ito *et al.* 1991) to GRAPE-6 (Makino *et al.* 2003), we adopted a fully pipelined processor designed specifically for the calculation of gravitational interaction between particles.

This fully-pipelined architecture means almost all transistors on a chip are used to implement arithmetic units, and each arithmetic unit can be optimized to a specific function assigned to it. Latest microprocessors such as Intel Core 2 Quad or Quad-core AMD Opteron has around 10^9 transistors, while a floating point arithmetic unit requires around 10^5 transistors. These microprocessors typically have eight arithmetic units. Thus, around 99.9% of all transistors are used for something other than the arithmetic units. On the other hand, a GRAPE-6 chip, consisting of only 10^7 transistors, have around 400 arithmetic units. Thus, the peak performance of a GRAPE chip is much higher than that of a general-purpose microprocessor made with the same technology, or the technology 5–10 years more advanced.

2.2. *Project history*

We started the development of GRAPE-type machines back 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. 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 researchers worldwide for many different problems.

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

https://doi.org/10.1017/S1743921308016165 Published online by Cambridge University Press

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 640 Mflops.

A major difference between GRAPE-4 and previous machines was its size. GRAPE-4 integrated 1728 pipeline chips, for a peak speed of 1.08 Tflops. The machine was composed of 4 clusters, each with 9 processor boards. A single processor board housed 48 processor chips, all of which shared a single memory unit through another custom chip to handle predictor polynomials. GRAPE-4 chip used two-way virtual multiple pipeline, so that one chip looked like two chips with half the clock speed. Thus, one GRAPE-4 board calculated the forces on 96 processors in parallel. Different boards calculated the forces from different particles, but to the same 96 particles. Forces calculated in a single cluster were summed up by a special hardware within the cluster.

In 2002, we completed GRAPE-6 (Makino *et al.* 2003). It is a direct successor of GRAPE-4. 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 special-purpose computer for the long-range interaction between particle can be applied to other particle-based simulations. In fact, 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, Dimmler & Levinthal 1991). 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). An even faster Protein Explorer was completed in 2006 (Narumi *et al.* 2006).

3. GRAPE-DR

3.1. *Problem with special-purpose architecture*

In July 2004, we were awarded the grant to develop the next-generation GRAPE system, which we call GRAPE-DR. This grant was, however, not for a special-purpose computer for astrophysical N-body simulation, but for a programmable massively-parallel processor. In the following we overview what is GRAPE-DR and why we chose to develop such a system.

As we summarized in the previous section, as far as the achieved speed is concerned, GRAPE hardwares have been pretty successful. Moreover, in the field of the dynamical evolution of star clusters, most of recent N-body simulations were performed on GRAPE hardwares. Thus, at the time of the completion of GRAPE-6, it was clearly desirable to develop next-generation GRAPE hardware. However, there was one quite practical limitation. The initial cost to design and fabricate a custom LSI chip has been increasing exponentially. For the processor of GRAPE-4 (year 1992), we paid around 200K USD as the initial cost. For GRAPE-6 (1997), we paid around 1.5M USD. A new design in 2004 would have costed at least 5M USD, and that in 2008 nearly 10M USD. The total amount of grant for GRAPE-4 was 2.5M USD, and that for GRAPE-6 was 5M. We need the total grant at least three times as much as that for the initial cost of the chip, in order to make a machine with reasonable price-performance ratio. Thus, the grant of around 15M USD was necessary to start the development of the next-generation GRAPE in 2003-4.

https://doi.org/10.1017/S1743921308016165 Published online by Cambridge University Press

This amount of money was way too much for the project to develop a computer which can be used in a relatively narrow field within the theoretical astrophysics.

There were several possible approaches for this problem. One would be just to forget about building a custom processor and relax. I sometimes think this might be the best approach, but there are still several other options. The second one is to use FPGAs, or field-programmable gate-array chips. An FPGA chip is a mass-produced LSI, in which a user can "program" arbitrary logic circuits. An FPGA chip consists of a number of logic elements and a programmable network which connects them. A logic element is essentially a small SRAM table with 4-5 address bits. Each logic elements can realize any combinatorial logic circuits with 4-5 inputs, and by programming the connection network one can implement more complex circuits.

The advantage of FPGA chips is that we do not have to pay the initial development cost of the chip. The disadvantage is that the size of the logic circuit which can fit into an FPGA chip is much smaller than that can fit into a custom LSI chip made using the same manufacturing technology. A logic element requires at least a few hundred transistors, while a logic gate in a custom chip requires around 10. This difference also results in the difference in the speed. Thus, using an FPGA chip for high-accuracy gravitational force calculation has been difficult.

The third approach is to try to get larger grant, for example through international collaborations. This is the way followed by many big projects in basic science, and observational astronomy is no exception. However, this is not the way to develop a special-purpose computer, since a very important requirement for the project to develop a special-purpose computer is that the development timescale must be short, in order to be able to outperform general-purpose computers. In the case of big projects in particle physics or observational astronomy, the long development time is not a fatal issue, since there are no other facilities which can do the same experiment or observation. However, in the case of a computer, the difference is essentially in just the speed. If the machine is not faster than general-purpose computers at the time of the completion, it has no value.

The fourth approach is to design a machine which can be used for applications other than the calculation of gravitational interaction between particles. There are again several approaches to achieve this goal, but with any approach, it is clear that the performance that can be achieved is significantly less that what is possible with traditional GRAPE design specialized to just one function. Even so, this approach can still be better than other approaches, in particular that of using FPGAs.

3.2. *The GRAPE-DR architecture*

Fig. 2 shows the basic structure of the new programmable GRAPE. It consists of a number of processing elements (PEs), each of which consists of an FPU and a register file. They all receive the same instruction from outside the chip, and perform the same operation.

Compared to the classic SIMD architecture such as that of TMC CM-2, the main difference are the followings.

a) PEs do not have large local memories.

b) There is no communication network between PEs.

We introduce these two simplifications so that a large number of PEs can be integrated into a single chip. If we want to have a large memory connected to each PE, the only economical way is to attach DRAM chips. However, once we decide to use external memory chips, it becomes very difficult to integrate large number of processors into a chip, since an external memory with sufficient bandwidth is practically impossible to add.

https://doi.org/10.1017/S1743921308016165 Published online by Cambridge University Press

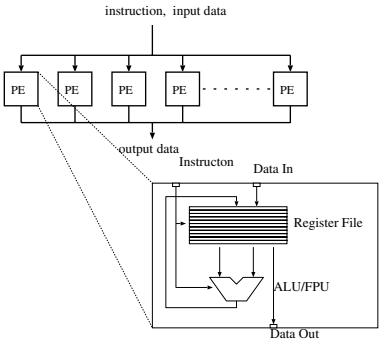

Figure 2. Basic structure of an SIMD processor.

A communication network is not very expensive, as far as it is limited into a single chip. A two-dimensional mesh network would be quite natural, for physically two-dimensional array of PEs on a single silicon chip. However, such a two-dimensional network poses a very hard problem, if we try to extend it to multi-chip systems. Again, external wires would be too costly.

If we eliminate the inter-PE communication network right from the beginning, we have no problem in constructing multi-chip systems, since PEs in different chips need not be connected.

Thus, this simple architecture has two advantages. First, we can integrate a very large number of PEs into a single chip. Second, it is easy to construct a system with multiple chips. As a result, we can construct a system with very high peak performance.

One problem with this architecture, when used as GRAPE, is that the number of processors is too large. A single chip can integrate several hundred PEs, and in order to use one PE efficiently, it is desirable to calculate the forces on several particles in one PE. Thus, the number of particles on which the forces are calculated in parallel becomes more than one thousand, which is generally too large for the numerical simulation of star clusters with individual timestep.

Of course, this problem was already there with GRAPE-4 or GRAPE-6, and we solved this problem by adding a reduction network, which takes the summation of partial forces calculated on many pipelines.

In the case of GRAPE-4, one processor board houses 48 pipeline chips and calculates forces on 96 particles in parallel. An additional summation circuit on another board takes the summation of forces from up to nine processor boards. In the case of GRAPE-6, one processor chip calculates the forces on 48 particles, and one board houses 32 processor chips. We added a reduction tree on each processor board. It takes the summation of forces calculated on 32 processor chips.

In the case of GRAPE-DR, each chip has 512 PEs, which is already a large number. So we added a reduction tree to each processor chip. We divided 512 processors to 16 groups each with 32 PEs, and added a reduction tree which takes the summation of results from these 16 groups. This reduction tree must be programmable. One node of the reduction tree of GRAPE-DR chip consists of the floating-point addition unit and integer ALU, which have the same logic design as those used in PEs, and the instructions are given from outside the chip essentially in the same way as the instructions to the PEs are supplied.

Thus, the processor has the architecture shown in Fig. 3. We added a buffer memory to each processor group, so that it can store the data sent from the external memory or the host computer.

https://doi.org/10.1017/S1743921308016165 Published online by Cambridge University Press

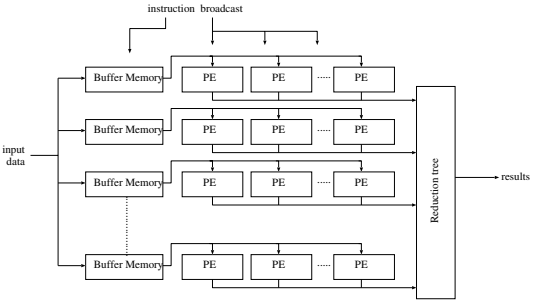

Figure 3. Modified SIMD architecture.

In this way, PEs in different blocks can calculate the forces from different particles. In addition, the reduction network allows multiple PEs in different blocks to calculate the force on the same particle from different particles. Thus, the efficiency for small-N systems or short-range force is greatly improved. In the following, we call these blocks of PE as broadcast blocks (BBs) and the buffer memory as broadcast memory (BM).

Note that the hardware cost of the buffer memory and reduction network is very small, since their cost is proportional to the number of blocks, which is a small fraction of the number of PEs.

We call this architecture GRAPE-DR, or Greatly Reduced Array of Processor Elements with Data Reduction.

4. Design of the GRAPE-DR chip

In this section, we overview the design of the GRAPE-DR chip. It integrates 512 simple processing elements (PEs), organized into 16 broadcast blocks. Each PE can do one floating-point addition and one multiplication in single precision per clock cycle, or one addition and one multiplication in double precision in every two clock cycles. The clock frequency is 500MHz and the theoretical peak performance is 512Gflops in single precision and 256 Gflops in double precision.

4.1. *PE architecture*

We designed the PE architecture so that the hardware is simple and yet can achieve high performance. Fig. 4 shows the architecture of a PE of the GRAPE-DR chip. A PE consists of a floating-point adder, a floating-point multiplier, an integer ALU, a three-port general-purpose register file (GP reg), a single-port local memory, and an additional dual-port working register (T register). The T register is used to store temporary values. The local memory augments the size of the register file. The address generator for the local memory supports the indirect addressing, by arrowing the content of the T register to be used as the address of the local memory. It also supports constant-stride access during vector operations. Storing of the results to memory units (GP reg and local memory) can be controlled by mask registers. Mask registers can store the flag output of the integer ALU and the floating-point adder.

Each PE has two fixed inputs, PEID and BBID. PEID gives the index of PE within its broadcast block, while BBID gives the index of the broadcast block itself. Using these fixed-number inputs and mask registers, we can control individual PEs independently.

The broadcast memory is dual-ported. With the current GRAPE-DR design, the data in the broadcast memory can be written directly to all of GP register, T register and

https://doi.org/10.1017/S1743921308016165 Published online by Cambridge University Press

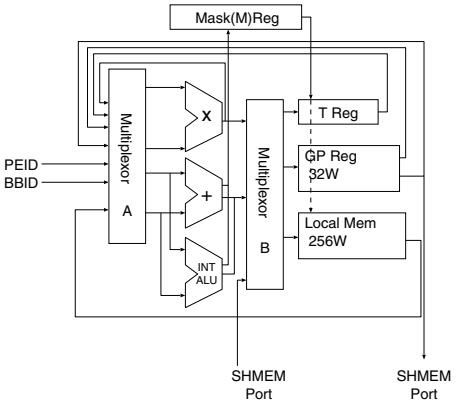

Figure 4. Structure of a Processor Element.

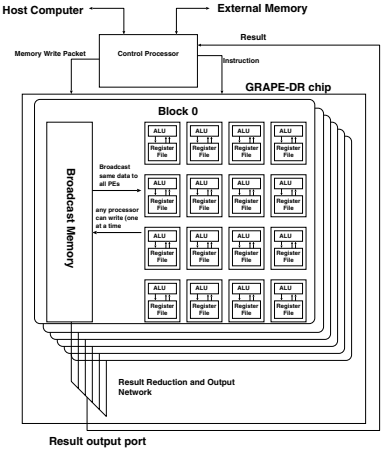

Figure 5. Overall Architecture of the GRAPE-DR chip.

the local memory, while only the data in the GP register can be transferred to to the broadcast memory.

The basic data format is a 72-bit floating-point format, with 1-bit sign, 11-bit exponent and 60-bit mantissa. We call this format double-precision. It also supports single-precision format with 24-bit mantissa. The integer ALU operates on 72-bit integers. The floating-point adder unit also work in 72-bit double-precision data, but it has the flag to round the output to single-precision format. Also, it has the flag to handle unnormalized numbers, for both the input and output.

The integer ALU can perform most of basic integer arithmetic and logical operations, including shift operations.

4.2. *Chip architecture*

Fig. 5 shows the overall architecture. We so far showed PEs in a two-dimensional grid, but from hardware point of view it is more appropriate to regard the structure as a two-level hierarchy. The chip consists of multiple broadcast blocks (BBs) of PEs. Each block consists of PEs and a broadcast memory (BM). All BBs receive the same data and instruction from outside the chip. The outputs of BBs are reduced by the reduction network.

https://doi.org/10.1017/S1743921308016165 Published online by Cambridge University Press

All communication to and from PEs are through BMs. To write a data to one PE, first we write that data to the BM, and then transfer it to the PE's memory (or registers). To read out the data in a PE, first we let it to transfer the data to the BM, and then use the reduction network to output the data.

The reduction network has the binary tree structure, and each tree node has the floating-point adder and integer ALU of the same design as those of PEs. Thus, we can apply many different reduction operations, such as summation, multiplication, max, min, and, or etc.

4.3. *Programming GRAPE-DR*

In the case where we use this new GRAPE-DR processor as the replacement of traditional special-purpose GRAPE chip for particle-particle interaction calculation, programming is not a very large issue. We can simply write down the microcode for the gravitational force calculation, and communication library routines etc. This is not much different from the softwares necessary for traditional GRAPE hardwares. The only difference is that we also need to write the microcode, which is just several tens of lines.

For other kind of particle-particle interactions, the development process is quite similar to that for gravitational force calculation, and much of the communication library codes can be recycled. Thus, it is more efficient to let some software generate the communication library from higher-level specifications, in the way similar to the PGPG system (Hamada, Fukushige & Makino 2005). Also, writing the horizontal microcode by hand is hard, even for just a few lines. We have developed a simple symbolic assembly language, in which the program is written in a more or less human-readable way. Compiler languages are also under development.

4.4. *Parallel GRAPE-DR system*

So far, we have discussed the design of a single chip. In practice, in order to achieve a reasonable performance, it is necessary to use many of these chips for one application. In other words, we need to discuss how to construct a large parallel system.

We continued the approach we used for previous GRAPE hardwares (Fukushige, Makino & Kawai 2005). The GRAPE-DR hardware will be designed as a relatively small attached processor for UNIX/Linux running workstations or PCs, and large parallel systems will be constructed just by assembling large PC clusters in which each node is connected to small GRAPE-DR hardware.

One GRAPE-DR card will house 4 processor chips, each with its own off-chip memory. The data transfer speed between the host and GRAPE-DR card can be the bottleneck, but current fast interface standards like 8-lane PCI-Express would offer reasonable bandwidth, at least for the current GRAPE-DR chip.

We plan to complete a 4096-chip system by early 2009. It will have the theoretical peak performance of 2 Pflops for single precision and 1 Pflops for double precision. Most likely, it will be a 512-node system each with two GRAPE-DR cards.

5. Development status

We finished the physical design of the GRAPE-DR chip by the end of 2005, and received the first sample chips in May 2006. Left panel of Fig. 6 shows the top-level layout image of the chip. Each white square is one PE. The die size is 18mm by 18mm.

We have developed the GRAPE-DR test board (see the right photo of Fig. 6), which houses one GRAPE-DR chip around the same time and confirmed the operation of the chip with both the test vectors and for real applications. The test board consists of one

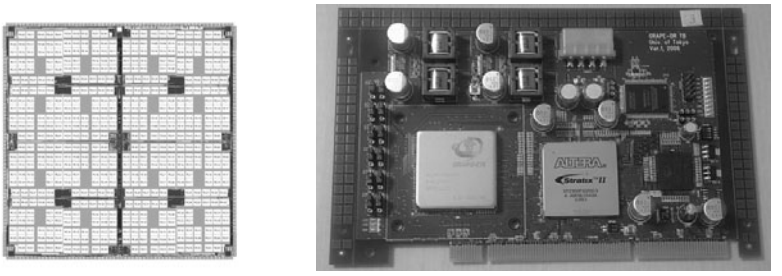

Figure 6. GRAPE-DR chip layout (left) and GRAPE-DR test board (right).

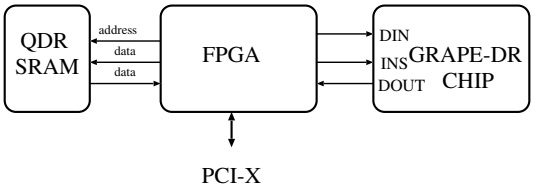

Figure 7. GRAPE-DR chip test board block diagram.

GRAPE-DR chip, one FPG chip (Altera Stratix II), and one memory chip. The interface to the host is PCI-X, and we used the IP core from PLDA. Fig. 7 shows the block diagram of the GRAPE-DR test board.

We have finished the development of the second board with PCI-Express interface and large on-board memory with DDR2 DRAM.

The measured maximum power consumption of the GRAPE-DR chip was 65W.

6. Discussion

6.1. *Comparison with related projects*

The design of ClearSpeed CX600 is quite similar to GRAPE-DR. It consists of 96 PEs, each with integer ALU, FMUL, FADD, integer MAC, 5-port register file and 6KB of memory. Compared to GRAPE-DR, the main difference is the lack of the support for the hierarchical structure (broadcast memory and reduction network). Since the number of PEs in the CX chip is still relatively small, the reduction network might not be crucial for the application performance. Its peak speed for matrix multiplication is 25 Gflops, which is about 1/10 of that of a GRAPE-DR chip.

Recent GPUs with the so-called "Unified Shader" architecture, in particular nVidia GeForce 8800, can be used as GPGPU (General-Purpose GPU), in the way rather similar to a GRAPE-DR processor chip. The peak performance numbers of GeForce 8800 and GRAPE-DR chip are rather similar. The former can perform 128 single-precision multiplications and 128 multiply-and-add operations also in single precision, at the clock speed of 1.35GHz. Thus, the theoretical peak performance is 518 Gflops. The peak performance of a GRAPE-DR chip is 512 Gflops. The transistor count of GeForce 8800 is 681M, while that of GRAPE-DR is 450M. Both are manufactured using TSMC 90nm process. Compared to GPUs, a GRAPE-DR chip lacks the fast external memory. In practice, it is not easy to use the large external memory of GPUs efficiently, since the communication bandwidth of a GPU with its host is rather slow. Compared to a GRAPE-DR chip, a GPU chip lacks the reduction tree, hardware support for double-precision operation, and few other minor things which help to achieve reasonable performance for the used as

https://doi.org/10.1017/S1743921308016165 Published online by Cambridge University Press

GRAPE. At this point, it is not clear how a GRAPE-DR board compares with GPGPU for real applications.

Acknowledgments

The authors thank Toshiyuki Fukushige, Yoko Funato, Piet Hut, Toshikazu Ebisuzaki, and Makoto Taiji for discussions related to this work. The GRAPE-DR chip design was done in collaboration with IBM Japan and Alchip company. We thank Ken Namura, Mitsuru Sugimoto, and many others from these two companies. The design of the control processor on the prototype board was done by Takeshi Fujino. This research is partially supported by the Special Coordination Fund for Promoting Science and Technology (GRAPE-DR project), Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

Bakker, A. F. & Bruin C., 1988, in: B. J. Alder (ed.), *Special Purpose Computers*, (San Diego: Academic Press), p. 183

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., Makino, J., & Kawai, A. 2005 *PASJ*, 57, 1009

Fukushige, T., Taiji, M., Makino, J., Ebisuzaki, T., & Sugimoto, D. 1996 *ApJ* , 468, 51

Hamada, T., Fukushige, T., & Makino, J. 2005 *PASJ*, 57, 799

Ito, T., Makino, J., Ebisuzaki, T., & Sugimoto, D. 1990 *Computer Physics Communications*, 60, 187

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

Kawai, A., Fukushige, T., Makino, J., & Taiji, M. 2000 *PASJ*, 52, 659

Makino, J. *PASJ*, 43, 621

Makino, J., Fukushige, T., Koga, M., & Namura, K. 2003 *PASJ*, 1991, 55, 1163

Makino, J., Fukushige, T.K. Okumura, S., & Ebisuzaki, T. 1993 *PASJ*, 45, 303

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

Narumi, T., Ohno, Y., Okimoto, N., Koishi, T., Suenaga, A., Futatsugi, N., Yanai, R., Himeno, R., Fujikawa, S., Taiji, M., & Ikei M. 2006 in: *Proceedings of SC06*, (ACM Press), CD-ROM

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. *MNRAS*, 278, 1005

Umemura, M., Fukushige, T., Makino, J., Ebisuzaki, T., Sugimoto, D., Turner, E. L., & Loeb, A. 1993 *PASJ*, 45, 311