Protein Explorer: A Petaflops Special-Purpose Computer System for Molecular Dynamics Simulations

Makoto Taiji[1]
taiji@gsc.riken.go.jp

Tetsu Narumi
taiji@gsc.riken.go.jp

Yousuke Ohno
ohno@gsc.riken.go.jp

Noriyuki Futatsugi
foota@gsc.riken.go.jp

Atsushi Suenaga
suenaga@gsc.riken.go.jp

Naoki Takada
ntakada@gsc.riken.go.jp

Akihiko Konagaya
konagaya@gsc.riken.go.jp

High-Performance Biocomputing Research Team, Bioinformatics Group,
Genomic Sciences Center, Institute of Physical and Chemical Research (RIKEN)
61-1 Ono-cho, Tsurumi, Yokohama, Kanagawa, 230-0056 Japan

Abstract

We are developing the 'Protein Explorer' system, a petaflops special-purpose computer system for molecular dynamics simulations. The Protein Explorer is a PC cluster equipped with special-purpose engines that calculate nonbonded interactions between atoms, which is the most time-consuming part of the simulations. A dedicated LSI 'MDGRAPE-3 chip' performs these force calculations at a speed of 165 gigaflops or higher. The system will have 6,144 MDGRAPE-3 chips to achieve a nominal peak performance of one petaflop. The system will be completed in 2006. In this paper, we describe the project plans and the architecture of the Protein Explorer.

1. Introduction

The history of modern computing began with the invention of a universal machine and the formulation of a clear definition of the computing procedures on it. The universal computer enables us to utilize essentially the same machine for various applications, and thus the hardware can be designed independently of the applications. This feature makes the development of a computer easier, since the engineering can be focused on the design of a single architecture. In fact, the universal computer may have a wider range of applications universal computer may have a wider range of applications than any other machine in history. Now, as the transistor density in microchips has reached its critical point, there is little difference between the technologies used in conventional personal computers and those used in high-performance computers; the two species of computers have begun to converge. In light of these facts, it would appear to be very difficult to compete with the conventional computer. However, we are going to face two obstacles on the road of the high-performance computing with the conventional technology, the memory bandwidth bottleneck and the heat dissipation problem. These bottlenecks can be overcome by developing specialized architectures for particular applications and sacrificing some of the universality.

The GRAPE (GRAvity PipE) project is one of the most successful attempts to develop such high-performance, competitive special-purpose systems[26, 17]. The GRAPE systems are specialized for simulations of classical particles such as gravitational N-body problems or molecular dynamics (MD) simulations. In these simulations, most of the computing time is spent on the calculation of long-range forces, such as gravitational, Coulomb, and van der Waals forces. Therefore, the special-purpose engine calculates only these forces, and all the other calculations are done by a conventional host computer connected to the system. This style makes the hardware very simple and cost-effective. This strategy of specialized computer architecture predates the GRAPE project. It was pioneered for use in molecular

*To whom all correspondence should be addressed

SC'03, November 15-21, 2003, Phoenix, Arizona, USA

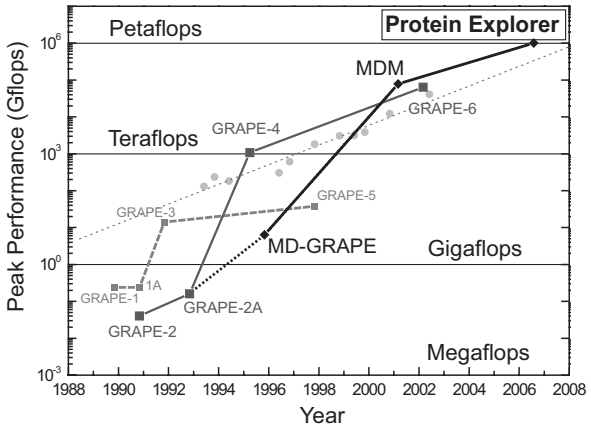

Figure 1. The growth of the GRAPE systems. For astrophysics, there are two product lines. The even-numbered systems, GRAPE-2, 2A, 4, and 6, are the high-precision machines, and the systems with odd numbers, GRAPE-1, 1A, 3, and 5, are the low-precision machines. GRAPE-2A, MD-GRAPE, MDM, and PE are the machines for MD simulations.

dynamics simulations by Bakker et al. in the Delft Molecular Dynamics Processor (DMDP) [2] and by Fine et al. in the FASTRUN processor [5]. However, neither system was able to achieve effective cost-performance. The most important drawback was the architectural complexity of these machines, which demanded much time and money in the development stages. Since the technology of electronic devices continues to develop very rapidly, speed of development is a crucial factor affecting the cost-to-performance ratio.

Figure 1 shows the advancement of the GRAPE computers. The project started at the University of Tokyo and is now run by two groups, one at the University of Tokyo and one at the RIKEN Institute. The GRAPE-4[27] built in 1995 was the first machine to break the teraflops barrier in nominal peak performance. Since 2001, the leader in performance has been the MDM (Molecular Dynamics Machine)[18, 20, 19] at RIKEN, which boasts a 78-Tflops performance. At the University of Tokyo, the 64-Tflops GRAPE-6 was completed in 2002[15, 14]. To date, five Gordon-Bell Prizes (1995, 1996, 1999-2001) have been awarded to simulations using the GRAPE/MDM systems. The GRAPE architecture can achieve such high-performance because it solves the problem of memory bandwidth bottleneck and lessens the heat dissipation problem. Based on these successes, in 2002 we launched a project to develop the 'Protein Explorer' (PE), a petaflops special-purpose computer system for molecular dynamics

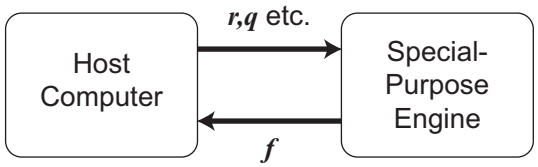

Figure 2. The basic architecture of the Protein Explorer and the other GRAPE systems.

simulations, particularly protein simulations. The Protein Explorer is a successor of GRAPE-2A[9], MD-GRAPE[28, 7], and MDM/MDGRAPE-2[18], which are also machines for molecular dynamics simulations.

In recent years there have been very rapid advances in structural genomics, and many three-dimensional structures of proteins and other biomolecules have been solved. In Japan, for example, the national 'Protein 3000' project was started in 2002 with the goal of solving the structures of 3,000 proteins by the year 2007. The PE project is motivated and funded by the Protein 3000 project. Such molecular dynamics simulations by means of high-performance, special-purpose computers will be a very powerful tool for applying the new structural knowledge to advances in bioscience and biotechnology. The main targets of the PE are the high-precision screening for drug design[24] and the large-scale simulations of huge proteins/complexes[22, 25]. The Protein Explorer system will appear in early 2006. In this paper, we describe the hardware architecture of the Protein Explorer system and estimate its performance.

2. Outline of the Protein Explorer

First, we will describe the basic architecture of the Protein Explorer and what the PE calculates. Figure 2 shows the basic architecture of the PE and the other GRAPE systems. The system consists of a general-purpose commercial computer and a special-purpose engine. In the case of the PE, it consists of special-purpose boards attached to a host PC cluster. The host sends the coordinates and the other data of particles to the special-purpose engine, which then calculates the forces between particles and returns the results to the host computer. In the molecular dynamics simulations, most of the calculation time is spent on nonbonded forces, i.e., the Coulomb force and van der Walls force. Therefore, the special-purpose engine calculates only nonbonded forces, and all other calculations are done by the host computer. This makes the hardware and the software quite simple. For hardware we need to consider only the calculation of forces, which is rather simple and has little variation. In the DMDP, on the other hand, almost all of

the work in the MD simulation was done by the hardware, so that the hardware was highly complex and very time-consuming to build. In the GRAPE systems no detailed knowledge on the hardware is required to write programs and a user simply uses a subroutine package to perform force calculations. All other aspects of the system, such as the operating system, compilers, etc., rely on the host computer and do not need to be specially developed.

The communication time between the host and the special-purpose engine is proportional to the number of particles, N, while the calculation time is proportional to its square, N^2, for the direct summation of the long-range forces, or is proportional to NN_c, where N_c is the average number of particles within the cutoff radius of the short-range forces. Since N_c usually exceeds a value of several hundred, the calculation cost is much higher than the communication cost. In the PE system, the ratio between the communication speed and the calculation speed of the special-purpose engine will be 0.25 Gbytes/sec·Tflops = 0.25 bytes for one thousand operations. This ratio is fairly small compared with those in the commercial parallel processors. Such a low communication speed is adequate to make efficient use of the special-purpose engine.

3. What Protein Explorer calculates

Next, we describe what the special-purpose engine of the PE calculates. The PE calculates two-body forces on i-th particle $\boldsymbol{F}_i$ as

$$\boldsymbol{F}_i = \sum_j a_j g(b_j r_s^2) \boldsymbol{r}_{ij}, \tag{1}$$

where $\boldsymbol{r}_{ij} = \boldsymbol{r}_j - \boldsymbol{r}_i, r_s^2 = r_{ij}^2 + \epsilon_i^2$. The vectors $\boldsymbol{r}_i, \boldsymbol{r}_j$ are the position vectors of the i, j-th particles and ϵ_i is a softening parameter to avoid numerical divergence. For the sake of convenience, we hereafter refer to the particles on which the force is calculated as the 'i-particle', and the particles which exert the forces on the i-particle as the 'j-particle'. The function $g(\zeta)$ is an arbitrary smooth function. For example, in the case of Coulomb forces, the force is given by

$$\boldsymbol{F}_i/q_i = \sum_j \frac{q_j}{r_{ij}^3} \boldsymbol{r}_{ij}, \tag{2}$$

where q_i, q_j are the charges of the i-th particle and the j-th one, respectively. This can be calculated by using $a_j = q_j, b_j = 1, g(\zeta) = \zeta^{-3/2}, \epsilon_i = 0$. The multiplication by q_i is done by the host computer.

In the case of a force by Lennard-Jones potential, the force between particles is given by

$$\boldsymbol{f}_{ij} = \left[\frac{A_{ij}}{r_{ij}^8} - \frac{B_{ij}}{r_{ij}^{14}} \right] \boldsymbol{r}_{ij}, \tag{3}$$

where A_{ij} and B_{ij} are constants determined from the equilibrium position and the depth of a potential. These constants depend on species of i-th and j-th particles. This force law can be evaluated by choosing $g(\zeta), a_j, b_j, \epsilon_i$ as follows.

$$\begin{aligned} g(\zeta) &= \zeta^{-4} - \zeta^{-7}, \\ a_j &= A_{ij}^{7/3} B_{ij}^{-4/3}, \\ b_j &= \left(\frac{A_{ij}}{B_{ij}} \right)^{1/3}, \\ \epsilon_i &= 0 \end{aligned} \tag{4}$$

The other potentials, including the Born-Mayer type repulsion ($U(r) = A \exp(Br)$), can also be evaluated.

In addition to Equation (1), the PE can also calculate the following equations:

$$c_i = \sum_j a_j g(\boldsymbol{k}_i \cdot \boldsymbol{r}_j), \tag{5}$$

$$\boldsymbol{F}_i = \sum_j \boldsymbol{k}_j c_j g(\boldsymbol{k}_j \cdot \boldsymbol{r}_i + \phi_j), \tag{6}$$

$$\phi_i = \sum_j a_j g(b_j r_s^2). \tag{7}$$

Equations (5) and (6) are used to calculate wave-space sums in the Ewald method[6]. The potential energies and isotropic virials are calculated by Equation (7).

Virial tensors, which are necessary for simulation under tensile stress using the Parrinello-Rahman method[23], can be calculated from forces. At first, in the case of open boundary conditions, virial tensors are defined by

$$\Phi = \sum_i \boldsymbol{F}_i : \boldsymbol{r}_i, \tag{8}$$

which we can calculate directly from forces $\boldsymbol{F}_i$. In the case of periodic boundary conditions, it has been well established that a different formula must be used[1] :

$$\Phi = \frac{1}{2} \sum_{i,j,\alpha} \boldsymbol{f}_{ij}^{\alpha} : (\boldsymbol{r}_i - \boldsymbol{r}_j^{\alpha}), \tag{9}$$

where α denotes neighbor cells. To calculate this, a force $\boldsymbol{F}_i$ must be divided into several forces from each cell α as

$$\boldsymbol{F}_i^{\alpha} = \sum_j \boldsymbol{f}_{ij}^{\alpha}, \tag{10}$$

and then the virial tensor (9) can be evaluated as

$$\Phi = \frac{1}{2} \sum_{i,\alpha} \boldsymbol{F}_i^{\alpha} : (\boldsymbol{r}_i - \boldsymbol{R}_{\alpha}), \tag{11}$$

where $\boldsymbol{R}_{\alpha}$ is a translation vector to a cell α.

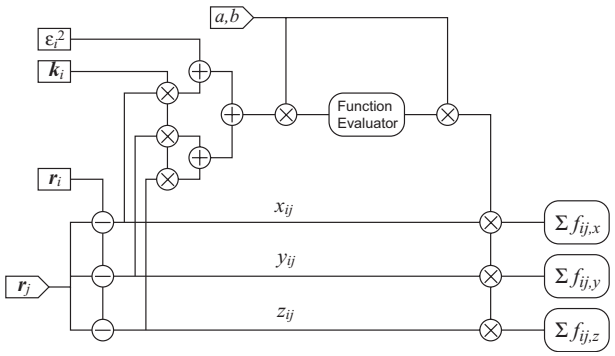

Figure 3. Block diagram of the force calculation pipeline in the MDGRAPE-3 chip.

There are several clever algorithms for efficient calculation of the long-range forces. In high-precision calculations such as those in molecular dynamics simulations, these algorithms use the direct summation for the near-field forces. Therefore, the GRAPE can accelerate these algorithms. There are also several implementations of these algorithms, i.e., the tree algorithm[13], the Ewald method[6, 20], and the modified fast multipole method[11, 12].

4. MDGRAPE-3 Chip

In this section, we describe the MDGRAPE-3 chip, the force calculation LSI for the Protein Explorer system. We start from the explanation of the LSI because it is the most important part of the system.

4.1. Force Calculation Pipeline

Figure 3 shows the block diagram of the force calculation pipeline of the MDGRAPE-3 chip. It calculates Equations (1), (5), (6), and (7) using the specialized pipeline. It will consist of three subtractor units, six adder units, eight multiplier units, and one function-evaluation unit. It can perform about 33 equivalent operations per cycle when it calculates the Coulomb force[10]. In the case the function-evaluation unit calculates $x^{-3/2}$, which is equivalent to 16 floating point operartions. The count depends on the force to be calculated. Most of the arithmetic operations are done in 32-bit single-precision floating-point format, with the exception of the force accumulation. The force $\boldsymbol{F}_i$ is accumulated in 80-bit fixed-point format and it can be converted to 64-bit double-precision floating-point format. The coordinates $\boldsymbol{r}_i$, $\boldsymbol{r}_j$ are stored in 40-bit fixed-point format because this makes the implementation of periodic boundary condition easy, and the dynamic range of the coordinates is relatively small in molecular dynamics simulations.

The function evaluator, which allows calculation of an arbitrary smooth function, is the most important part of the pipeline. This block is almost the same as those in MD-GRAPE [28, 7]. It has a memory unit which contains a table for polynomial coefficients and exponents, and a hardwired pipeline for the fourth-order polynomial evaluation. It interpolates an arbitrary smooth function $g(x)$ using segmented fourth-order polynomials by the Horner's method

$$\begin{aligned} g(x_0 + \Delta x) &= (((c_4[x_0]\Delta x + c_3[x_0])\,\Delta x + c_2[x_0]) \\ &\quad \Delta x + c_1[x_0])\,\Delta x + c_0[x_0], \end{aligned} \tag{12}$$

where x_0 is the center of the segmented interval, $\Delta x = x - x_0$ is the displacement from the center, and $c_k[x_0]$ are the coefficients of polynomials. The coefficient table has 1,024 entries so that it can evaluate the forces by the Lennard-Jones (6–12) potential and the Coulomb force with a Gaussian kernel in single precision.

In the case of Coulomb forces, the coefficients a_j and b_j in equation (1) depend only on the species of j-particle, and they are supplied from the memories. Actually, the charge of the j-th particle q_j corresponds to a_j, and $b_j = 1$ is constant. On the other hand, in the case of the van der Waals force, these coefficients depend on the species of both the i-particle and the j-particle, as we can see in Equation (4). Thus, these coefficients must be changed when the species of the i-particle changes. Since it consumes a lot of time to exchange them with each change in species, the pipeline contains a table of the coefficients. The table generates the coefficients from the species of both i-particles and j-particles. The number of species is 64, which is sufficient for most biomolecular simulations.

4.2. j-Particle Memory and Control Units

Figure 4 shows the block diagram of the MDGRAPE-3 chip. It will have 20 force calculation pipelines, a j-particle memory unit, a cell-index controller, a force summation unit, and a master controller. The master controller manages the timings and the inputs/outputs of the chip. The j-particle memory unit holds the coordinates of j-particles for 32,768 bodies and corresponds to the 'main memory' in general-purpose computers. Thus, the chip is designed using the memory-in-the-chip architecture and no extra memory is necessary on the system board. The amount of memory is 6.6 Mbits and will be constructed by a static RAM. The same output of the memory is sent to all the pipelines simultaneously. Each pipeline calculates using the same data from the j-particle unit and individual data stored in the local memory of the pipeline. Because the two-body force calculation is given by

$$\boldsymbol{F}_i = \sum_j \boldsymbol{f}(\boldsymbol{r}_i, \boldsymbol{r}_j), \tag{13}$$

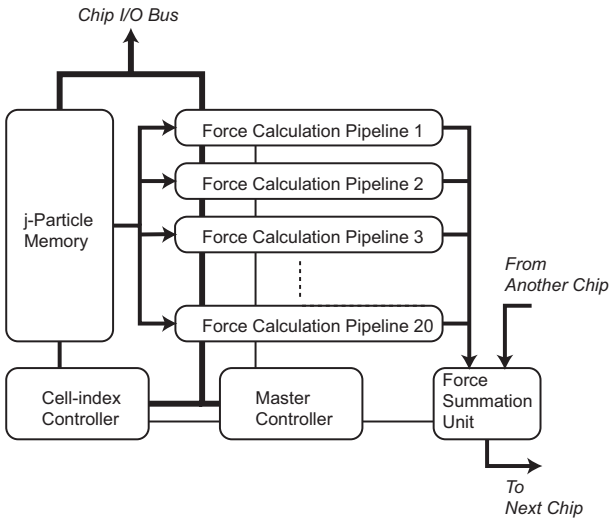

Figure 4. Block diagram of the MDGRAPE-3 chip.

for the parallel calculation of multiple $\boldsymbol{F}_i$, we can use the same $\boldsymbol{r}_j$. This parallelization scheme, 'the broadcast memory architecture', is one of the most important advantages of the GRAPE systems. It enables the efficient parallelization at low bandwidth realized by simple hardware. In addition, we can extend the same technique to the temporal axis. If a single pipeline calculates two forces, $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i+1}$, in every two cycles by timesharing, we can again use the same $\boldsymbol{r}_j$. This means that an input $\boldsymbol{r}_j$ is needed every two cycles. We refer to this parallelization scheme as a 'virtual multiple pipeline'[16], because it allows a single pipeline to behave like multiple pipelines operating at half speed. The merit of this technique is the reduction of the 'virtual frequency'. The requirement for the memory bandwidth is reduced by half, and it can be decreased further by increasing the number of the 'virtual' pipelines. The hardware cost for the virtual pipeline consists of only the registers for the coordinates and the forces, which are very small.

In the MDGRAPE-3 chip there are two virtual pipelines per physical pipeline, and thus the total number of virtual pipelines is 40. The physical bandwidth of the j-particle unit will be 2.5 Gbytes/sec, but the virtual bandwidth will reach 100 Gbytes/sec. This allows quite efficient parallelization in the chip. The MDGRAPE-3 chip has 340 arithmetic units and 20 function-evaluator units which work simultaneously. The chip has a high performance of 165 Gflops at a modest speed of 250 MHz. This advantage will become even more important in the future. The number of transistors will continue to increase over the next ten years, but it will become increasingly difficult to use additional transistors to enhance performance. In fact, in the general-purpose scalar processor, the number of floating-point units in the CPU seems to reach a ceiling at around 4–8 because of the limitation in the bandwidth of the memory, that of the register file, and the difficulties in software. On the other hand, in the GRAPE systems the chip can house more and more arithmetic units with little performance degradation. Thus, the performance of the chip exactly follows the performance of the semiconductor device technology, which is the product of the number of transistors and the speed. In the general-purpose processor, the performance increases roughly 10 times every 5 years, a rate slower than that in the semiconductor device, which is about 30 times every 5 years. Therefore, a special-purpose approach is expected to become increasingly more advantageous than a general-purpose approach. The demerit of the broadcast memory parallelization is, of course, its limitation with respect to applications. However, we could find several applications other than particle simulations. For example, the calculation of dense matrices[21] and the dynamic programming algorithm for hidden Markov models are also possible to be accelerated by the broadcast memory parallelization.

The size of the j-particle memory, 32,768 bodies = 6.6 Mbits, is sufficient for the chip in spite of the remarkable calculation speed, since the molecular dynamics simulation is a computation-intensive application and not a memory-intensive application like the fluid dynamics simulations. Of course, the number of the particles often exceeds this limit, but we can use many chips in parallel to increase the capacity. We can divide the force $\boldsymbol{F}_i$ on the i-th particle as

$$\boldsymbol{F}_i = \sum_{k=1}^{n_j} \boldsymbol{F}_i^{(k)} \tag{14}$$

by grouping j-particles into n_j subsets. Thus, by using n_j chips we can treat n_j times more particles at the same time. However, if a host computer collects all partial forces $\boldsymbol{F}_i^{(k)}$ and sums up them, the communication between the host and the special-purpose engines increase n_j times. To solve this problem, the MDGRAPE-3 chip has a force summation unit, which calculates the summation of the partial forces. In the PE system, the MDGRAPE-3 chips on the same board will be connected by a unidirectional ring, as explained later. Therefore, each chip receives the partial forces from the previous chip, adds the partial force calculated in the chip, and sends the results to the next chip. The force summation unit calculates the sum in the same 80-bit fixed-point format as used for the partial force calculations in the pipelines. The result can be converted to a 64-bit floating-point format. The summation of the force subsets on different boards is done by the host computer. If the number of particles still exceeds the total memory size, the forces must be subdivided into several subsets, and the contents of the j-particle memory should be replaced for each subset. In this case the number of particles is quite

huge, so the overhead for communication/calculation is not as important.

The j-particle memory is controlled by the cell-index controller, which generates the address for the memory. To calculate short-range forces, it is unnecessary to calculate two-body interactions with all particles. There exists a cut-off r_c so that forces can be ignored when $r > r_c$. If we calculate contributions inside the sphere $r < r_c$, the cost of force calculations decreases by $(L/r_c)^3$, where L is the size of a system. Thus, when $L \gg r_c$, the efficiency increases substantially. The standard technique for this approach is called the cell index method[1]. In this method a system is divided into cells with sides l. When we calculate the forces on the particles in a cell, we treat only particles in cells within the cutoff.

The implementation of the cell-index method is almost the same as for the MD-GRAPE/MDGRAPE-2 systems. The host computer divides particles into cells and sorts them by the cell indices. Thus, the particles which belong to the same cell have sequential particle numbers. Thus the address of the particles in a cell can be generated from the start address and the end one. The cell-index table contains the start addresses and the end ones for all cells. The controller generates the cell number involved in the interaction and looks up the table to obtain the start address and the end one. Then the counter generates the sequential address for the j-particle memory. When the count reaches to the end address, the same process is repeated for the next cell. It can also translate cells according to the periodic boundary condition, and supports the virial tensor calculation. As shown in equation (11), we need to evaluate the forces separated by each mirror image to calculate a virial tensor. The cell-index controller controls the force calculation pipelines to calculate these partial forces.

Gathering all the units explained above, the MDGRAPE-3 chip has almost all elements of the MD-GRAPE/MDGRAPE-2 system board except for the bus interface. The chip is designed to operate at 250 MHz in the worst condition (1.08V, 85°C, process factor = 1.3). It has 20 pipelines and each pipeline performs 33 equivalent operations per cycle, and thus the peak performance of the chip will reach 165 Gflops. This is 12.5 times faster than the previous MDGRAPE-2 chip. In the typical condition (1.2V, 65°C, process factor = 1.0) it will work at 460 MHz with the peak performance of 300 Gflops. It will be possible to design a chip with the speed of gigahertz at the typical condition, however, the power dissipation will become about 80W and the number of transistors per operation will increase at least 80%. Since the parallelization is very efficient in our architecture, the gigahertz speed will cause a lot of difficulties in the power dissipation and the layout but will bring no performance gain. Therefore, we choose the modest speed of 250 MHz at worst and 460 MHz at typical. The chip will be made by Hitachi Device Development Center HDL4N 0.13 μm technology. It consists of 6M gates and 10M bits of memory, and the chip size will become about 220 mm^2. It will dissipate 20 watts at the core voltage of +1.2 V. Its power per performance will be 0.12 W/Gflops, which is much better than those of the conventional CPUs. For example, the thermal design power of the Pentium 4 3GHz processor produced by a 0.13 μm process is 82 watts, while its nominal peak performance is 6 Gflops. Therefore, its power per performance is about 14 W/Gflops, which is a hundred times worse than that of the MDGRAPE-3. There are several reasons for this high power efficiency. First, the accuracy of the arithmetic units is smaller than that of the conventional CPUs. Since the number of gates of a multiplier is roughly proportional to the square of the word length, this precision has a pronounced impact on the power consumption. Secondly, in the MDGRAPE-3 chip, 90% of transistors are used for the arithmetic operations and the rest are used for the control logic, (the transistors for the memory are not counted). The specialization and the broadcast memory architecture make the efficient usage of silicon possible. And lastly, the MDGRAPE-3 chip works at the modest speed of 250 MHz. At gigahertz speed, the depth of the pipeline becomes very large and the ratio of the pipeline registers tends to increase. Since the pipeline registers dissipate power but perform no calculation, there is a lessening of power-efficiency. Thus, the GRAPE approach is very effective to suppress power consumption. This is another important advantage of the GRAPE. When the feature size of the semiconductor decreases, almost all things are improved except for the power density. Therefore, the power dissipation will become important in the future, and the GRAPE architecture can alleviate the problem.

5. System Architecture

In this section, the system architecture of the Protein Explorer is explained. Figure 5 shows the block diagram of the PE system. The system will consist of a PC cluster with special-purpose engines attached. For the host PC cluster we plan to use a future release of Itanium or Opteron CPU. The choice will be made based on both calculation performance and I/O performance. The system will have 256 nodes with 512 CPUs. By the end of 2004, we expect the nominal peak performance of the CPU to reach at least 10 Gflops, and the performance of the host to be 5 Tflops. The required network speed between the nodes is 10 Gbit/sec, which will be realized by Infiniband, 10G Ethernet, or future Myrinet. The network topology will be a two-dimensional hyper-crossbar.

Each node will have the special-purpose engines with 24

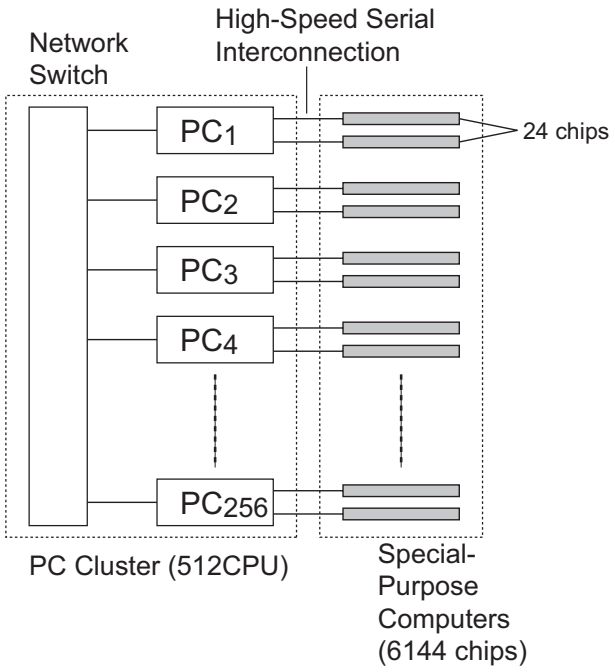

Figure 5. Block diagram of the MD-GRAPE system.

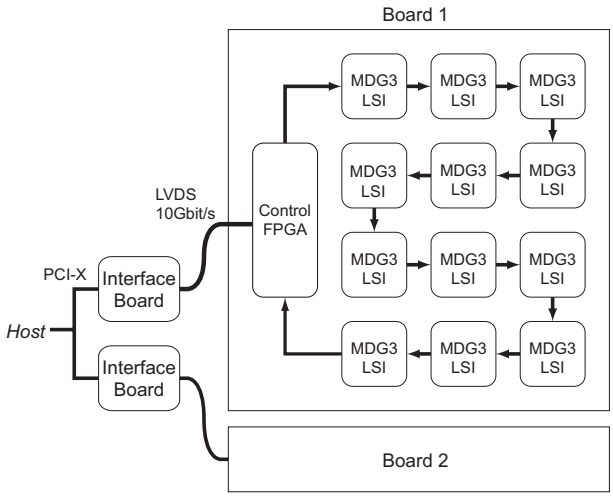

Figure 6. Block diagram of the Protein Explorer board. 'MDG3 LSI' denotes the MDGRAPE-3 chip.

MDGRAPE-3 chips. Since the MDGRAPE-3 chip has a performance of 165 Gflops, the performance of the nodes will be 3.96 Tflops. With 256 nodes the system reaches to a petaflops. The MDGRAPE-3 chips are connected to the host by two PCI-X buses at 133 MHz. Figure 6 shows the block diagram of the Protein Explorer board. The board has twelve MDGRAPE-3 chips, and each chip is connected in serial to send/receive the data. Since the memory is embedded in the MDGRAPE-3 chip, the board will be extremely simple. The speed of the communication between the chips will be 1.3 Gbytes/sec, which corresponds to an 80-bit word transfer at 133 MHz. For these connections 1.5V-CMOS I/O cells will be used. The board has a control FPGA (or ASIC) with a special bus with 1 Gbytes/sec peak speed. Two boards with 24 MDGRAPE-3 chips are connected to the host via two independent PCI-X buses. The boards and the PCI-X buses are connected by the high-speed serial interconnection with the speed of 10 Gbit/sec for both up and down streams. It is realized by the bundle of four 2.5 Gbit/sec LVDS channels.

The boards and PC motherboards will be mounted in a 19" rack, which can house 6 nodes. Therefore, the system will consist of 43 racks in total. The total power dissipation will be about 150 KWatts and it will occupy 100 m^2. Since the special-purpose computer is very efficient, we need no gymnasium even with a petaflops system.

6. Software

As mentioned above, in the GRAPE systems a user does not need to consider the detailed architecture of the special-purpose engine and all the necessary compute services are defined in the subroutine package. The following is a typical procedure of simulation using the Protein Explorer system. In this example, we assumed a PE board calculates the Coulomb and van der Waals forces under the free boundary condition. Table 1 summaries the library routines for this example.

1. call `pe_start_calc(n, r, rmax, q, maxspecies, speceis, amatrix, bmatrix)`. In the library, the following operations are performed.
   (a) Convert '`r`' to 40-bit fixed-point format using '`rmax`'.
   (b) Convert '`q`' to 32-bit single-precision floating-point format.
   (c) Generate command sequence for the PE board, and set it to the host memory where DMA (Direct Memory Access) can be used for.
   (d) Issue the start command to the PE board.
2. Perform other force calculations, such as bonding force.
3. call `pe_wait_calc(force)`. In the library, the following operations are performed.

Table 1. Example of the library routines for PE system

Subroutine name	Variables	Description
`pe_start_calc`	`int n`	Number of particles
	`double r[][3]`	Positions of particles
	`double rmax`	Maximum value of coordinates of particle positions
	`double q[]`	Charges of particles
	`int maxspecies`	Number of species
	`int species[]`	Species of particles
	`double amatrix[]`	Matrix for a_j in Equation (4)
	`double bmatrix[]`	Matrix for b_j in Equation (4)
`pe_wait_calc`	`double force[][3]`	Sum of the Coulomb and van der Waals forces on particles

   (a) Wait until the PE board finish the calculation.
   (b) Multiply charge (q_i) to the results from the board.
   (c) Add the Coulomb and van der Waals forces, and return them.

4. Add other forces such as bonding force to '`force`'. Then calculates orbits of particles, and increments a time.

The PE board performs the following operations after it receives the start command.

1. Get command sequence from the host memory using DMA. In the command sequence, data such as positions of particles are also packed in it.
2. Write positions, charges, and species of j-particles to the chip. Each chip has `n`/12 j-particles.
3. Broadcast positions of 40 i-particles to all the chip.
4. Issue the calculation command to all the chip.
5. Each chip calculates partial forces on 40 i-particles.
6. Read the result of the calculation from the chip. The sum of 12 partial forces are performed automatically between the chip.
7. Transfer the result to the host memory using DMA.
8. Repeat from 3 to 7 for all i-particles.
9. Tell the end of the calculation to the host.

The calculation of the chip and the data transfer between the host and the chip are performed simultaneously. Moreover, the operations of the PE board and the bonding force calculation on the host can be performed simultaneously.

Several popular molecular dynamics packages have already been ported for MDM/MDGRAPE-2, including Amber-6[4], CHARMM[3], and DL_POLY[29]. They can be used in the PE with small modifications. For Amber-6 and CHARMM, the parallel code using MPI also works with the MDM/MDGRAPE-2. The interactive molecular dynamics simulation system on the MDM/MDGRAPE-2 has also been developed using Amber-6[19]. It uses a three-dimensional eyeglass and a haptic device to manipulate molecules interactively.

7. Performance Estimation

In this section we discuss the sustained performance of the PE by means of a simple model. We define the speed of the MDGRAPE-3 chip as $f_{\rm GRAPE}$ Gflops, the sustained performance of the host CPU as $f_{\rm host}$ Gflops, the sustained communication speed of the host CPU and the special-purpose engine as $f_{\rm comm}$ Gbytes/sec, and the sustained communication speed between the host CPUs as $f_{\rm MPI}$ Gbytes/sec. Then, the calculation time $T_{\rm PE}$ in the GRAPE, the calculation time in the host computer $T_{\rm host}$, the communication time between the host and the GRAPE $T_{\rm comm}$, and the communication time between the hosts $T_{\rm MPI}$ are given by

$$\begin{aligned} T_{\rm PE} &= \frac{33N^2}{f_{\rm GRAPE} n_{\rm GRAPE}} \\ T_{\rm host} &= \frac{800N}{f_{\rm host} n_{\rm host}} \\ T_{\rm comm} &= \frac{88N}{f_{\rm comm}\sqrt{n_{\rm host}}} \\ T_{\rm MPI} &= \frac{96N}{f_{\rm MPI}\sqrt{n_{\rm host}}}, \end{aligned} \tag{15}$$

where the time unit is a nanosecond, N denotes the number of particles, and $n_{\rm GRAPE}$ and $n_{\rm host}$ are the number of the MDGRAPE-3 chips and the number of host CPUs, respectively. This model is based on the direct summation algorithm, which will show the best sustained performance. Figure 7 shows the sustained performance of the Protein

Explorer. The total time $T = T_{\rm PE} + T_{\rm host} + T_{\rm comm} + T_{\rm MPI}$ and the efficiency $T_{\rm PE}/T$ are plotted. Here the parameters are given as $f_{\rm GRAPE} = 165$ (Gflops), $f_{\rm host} = 1.5$ (Gflops), $f_{\rm comm} = 0.5$ (Gbytes/sec), and $f_{\rm MPI} = 0.5$ (Gbytes/sec). We assume $n_{\rm GRAPE} = 6,144$ and $n_{\rm host} = 512$ in the petaflops system, and $n_{\rm GRAPE} = 12$ and $n_{\rm host} = 1$ in the 2-Tflops system with one CPU. We ignore $T_{\rm MPI}$ for the 2-Tflops system. In the petaflops system, the efficiency reaches 0.5 at $N = 5 \times 10^5$, and the system with one million particles can be simulated by 0.05 sec/step. In the target applications of the PE, the throughput of systems with several ten thousand particles is important. In the single CPU system with 2 Tflops, the efficiency reaches 0.5 at $N = 4 \times 10^4$, and the system with $N = 4 \times 10^4$ particles can be simulated by 0.08 sec/step. Therefore, we can simulate such systems for 1.1 nsec/CPU/day, and 0.55 μsec/system/day when the simulation timestep is 1 fsec. For high-precision screening of drug candidates, a simulation of 10 nsec will be required for each sample. If the system is used for this purpose, we can treat 50 samples per day, which is very practical. For larger systems, the use of the Ewald method or the treecode is necessary. With the Ewald method, a system with ten million particles can be treated with 0.3 sec/step. The system will be useful for the analysis of such very-large biocomplexes. The sustained performance of the GRAPE systems is very good compared with that of general-purpose computers in real applications as well. The measured performance of a MDGRAPE-2 board (53 Gflops) was twice as fast as that of a PC cluster with 128 CPUs (Pentium III 1GHz) for $N > 3 \times 10^4$[8]. Considering that the cost-performance of the PE is 10–100 times better than that of the commercial PC, the PE will be very competitive.

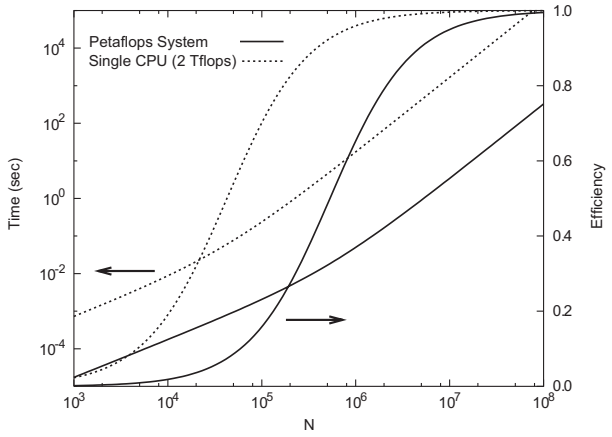

Figure 7. The sustained performance of the Protein Explorer with the direct summation algorithm. The calculation time per step (left axis) and the efficiency are plotted. The solid line and dashed one indicate those of the petaflops system and the 2-Tflops system, respectively.

8. Summary and Schedule

The Protein Explorer system is a special-purpose computer system for molecular dynamics simulations with a petaflops nominal peak speed. It is a successor of MDM/MDGRAPE-2, the GRAPE system for MD simulations. We are developing an MDGRAPE-3 chip for the PE that will have a peak performance of 165 Gflops. The chip will have 20 force calculation pipelines and a main memory of 6 Mbits. It will be made using 0.13 μm technology and will operate at 250 MHz in the worst case. The system will consist of 6,144 MDGRAPE-3 chips to achieve a petaflops. The host computer is a PC cluster with 256 nodes, each having 2 CPUs. The special-purpose engines are distributed to each node of the host.

A sample LSI of MDGRAPE-3 will appear in March 2004, and a system of 100 Tflops performance will be built by March 2005. The system will be complete in the first half on 2006, but it will be accelerated to be ready on 2005 if additional funds are available. The total cost is about 20 million US dollars including the labor cost. The price will be less than ten US dollars per gigaflops. The cost-performance is thus at least ten times better than that of general-purpose computers, even when compared with the relatively cheap BlueGene/L (140 US dollars/Gflops). The PE is expected to be a very useful tool for the biosciences, especially for structural genomics, nanotechnology, and the wide range of material sciences.

Acknowledgements

The authors would like to express their sincere gratitude to their many coworkers in the GRAPE projects, especially to Prof. Junichiro Makino, Prof. Daiichiro Sugimoto, Dr. Toshikazu Ebisuzaki, Prof. Tomoyoshi Ito, Dr. Toshiyuki Fukushige, Dr. Atsushi Kawai, Dr. Sachiko Okumura, and Dr. Ryutaro Susukita. This work was partially supported by the 'Protein 3000 Project' contracted by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, Oxford, 1987.

[2] A. F. Bakker and C. Bruin. Design and implementation of the Delft molecular-dynamics processor. In B. J. Alder, editor, *Special Purpose Computers*, pages 183–232. Academic Press, San Diego, 1988.

[3] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4:187–217, 1983.

[4] D. A. Case, D. A. Pearlman, J. W. Caldwell, T. E. Cheatham III, W. S. Ross, C. Simmerling, T. Darden, K. M. Merz, R. V. Stanton, A. Cheng, J. J. Vincent, M. Crowley, V. Tsui, R. Radmer, Y. Duan, J. Pitera, I. Massova, G. L. Seibel, U. C. Singh, P. Weiner, and P. A. Kollman. *Amber 6 Manual*. UCSF, 1999.

[5] R. Fine, G. Dimmler, and C. Levinthal. FASTRUN: a special purpose, hardwired computer for molecular simulation. *PROTEINS: Structure, Function and Genetics*, 11:242–253, 1991.

[6] T. Fukushige, J. Makino, T. Ito, S. K. Okumura, T. Ebisuzaki, and D. Sugimoto. A special purpose computer for particle dynamics simulations based on the Ewald method: WINE-1. In V. Milutinovix and B. D. Shriver, editors, *Proceedings of the 26th Hawaii International Conference on System Sciences*, pages 124–133, Los Alamitos, 1992. IEEE Computer Society Press.

[7] T. Fukushige, M. Taiji, J. Makino, T. Ebisuzaki, and D. Sugimoto. A highly-parallelized special-purpose computer for many-body simulations with arbitrary central force: MD-GRAPE. *Astrophysical J.*, 468:51–61, 1996.

[8] N. Futatsugi, N. Okimoto, A. Suenaga, H. Hirano, T. Narumi, N. Takada, A. Kawai, R. Susukita, K. Yasuoka, T. Koishi, H. Furusawa, M. Taiji, T. Ebisuzaki, and A. Konagaya. A high-speed and accurate molecular dynamics simulation system with special-purpose computer: MDM - effective calculational approach of biomolecules -. in preparation, 2003.

[9] T. Ito, J. Makino, T. Ebisuzaki, S. K. Okumura, and D. Sugimoto. A special-purpose computer for N-body simulations: GRAPE-2A. *Publ. Astron. Soc. Japan*, 45:339, 1993.

[10] A. H. Karp. Speeding up n-body calculations on machines lacking a hardware square root. *Scientific Programming*, 1:133–141, 1992.

[11] A. Kawai and J. Makino. Pseudoparticle multipole method: A simple method to implement a high-accuracy tree code. *Astrophysical J.*, 550:L143–L146, 2001.

[12] A. Kawai, J. Makino, and T. Ebisuzaki. Performance analysis of high-accuracy tree code based on pseudoparticle multipole method. Submitted to Astrophysical J., 2003.

[13] J. Makino. Treecode with a special-purpose processor. *Publ. Astron. Soc. Japan*, 43:621–638, 1991.

[14] J. Makino, T. Fukushige, and K. Nakamura. GRAPE-6: The massively parallel special-purpose computer for astrophysical particle simulations. in preparation, 2003.

[15] J. Makino, E. Kokubo, T. Fukushige, and H. Daisaka. A 29.5 Tflops simulation of planetesimals in Uranus-Neptune region on GRAPE-6. In *Proceedings of Supercomputing 2002*, 2002. in CD-ROM.

[16] J. Makino, E. Kokubo, and M. Taiji. HARP: A special-purpose computer for N-body simulations. *Publ. Astron. Soc. Japan*, 45:349, 1993.

[17] J. Makino and M. Taiji. *Scientific simulations with special-purpose computers*. John Wiley & Sons, Chichester, 1998.

[18] T. Narumi, R. Susukita, T. Ebisuzaki, G. McNiven, and B. Elmegreen. Molecular Dynamics Machine: Special-purpose computer for molecular dynamics simulations. *Molecular Simulation*, 21:401–415, 1999.

[19] T. Narumi, R. Susukita, A. Kawai, T. Koishi, N. Takada, A. Suenaga, N. Futatsugi, H. Furusawa, K. Yasuoka, N. Okimoto, M. Taiji, T. Ebisuzaki, , and A. Konagaya. A high-speed and accurate molecular dynamics simulation system - development of the Molecular Dynamics Machine -. in preparation, 2003.

[20] T. Narumi, R. Susukita, T. Koishi, K. Yasuoka, H. Furusawa, A. Kawai, and T. Ebisuzaki. 1.34 Tflops molecular dynamics simulation for NaCl with a special-purpose computer: MDM. In *Proceedings of Supercomputing 2000*, 2000. in CD-ROM.

[21] Y. Ohno, M. Taiji, A. Konagaya, and T. Ebisuzaki. MACE : MAtrix Calculation Engine. In *Proc. 6th World Multiconference on Systemics, Cybernetics and Informatics SCI*, pages 514–517, 2002.

[22] N. Okimoto, K. Yamanaka, A. Suenaga, Y. Hirano, N. Futatsugi, T. Narumi, K. Yasuoka, R. Susukita, T. Koishi, H. Furusawa, A. Kawai, M. Hata, T. Hoshino, and T. Ebisuzaki. Molecular dynamics simulations of prion proteins - effect of Ala117 $\rightarrow$ Val mutation -. *Chem-Bio Informatics Journal*, 3:1–11, 2003.

[23] M. Parrinello and A. Rahman. Polymorphic transitions in single crystals: a new molecular dynamics method. *J. Appl. Phys.*, 52:7182–7190, 1981.

[24] A. Suenaga, M. Hatakeyama, M. Ichikawa, X. Yu, N. Futatsugi, T. Narumi, K. Fukui, T. Terada, M. Taiji, M. Shirouzu, S. Yokoyama, and A. Konagaya. Molecular dynamics, free energy, and SPR analyses of the interactions between the SH2 domain of Grb2 and ErbB phosphotyrosyl peptides. *Biochemistry*, 42:5195–5200, 2003.

[25] A. Suenaga, N. Okimoto, N. Futatsugi, Y. Hirano, T. Narumi, A. Kawai, R. Susukita, T. Koishi, H. Furusawa, K. Yasuoka, N. Takada, M. Taiji, T. Ebisuzaki, and A. Konagaya. A high-speed and accurate molecular dynamics simulation system with special-purpose computer: MDM - application to large-scale biomolecule -. in preparation, 2003.

[26] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, and M. Umemura. A special-purpose computer for gravitational many-body problems. *Nature*, 345:33, 1990.

[27] M. Taiji, J. Makino, T. Ebisuzaki, and D. Sugimoto. GRAPE-4: A teraflops massively-parallel special-purpose computer system for astrophysical N-body simulations. In *Proceedings of the 8th International Parallel Processing Symposium*, pages 280–287, Los Alamitos, 1994. IEEE Computer Society Press.

[28] M. Taiji, J. Makino, A. Shimizu, R. Takada, T. Ebisuzaki, and D. Sugimoto. MD-GRAPE: a parallel special-purpose computer system for classical molecular dynamics simulations. In R. Gruber and M. Tomassini, editors, *Proceedings of th 6th conference on physics computing*, pages 609–612. European Physical Society, 1994.

[29] http://www.dl.ac.uk/TCSC/Software/DL_POLY/main.html.