Project Description – Proposal Section C

1. Introduction

For a substantial subset of problems in computational physics, special-purpose hardware has the potential to combine ease of use with efficient implementation of basic numerical algorithms, and may provide the best path to continued high computational performance. In the particular case of astrophysical N-body calculations, the so-called "GRAPE" hardware (GRAvity PipE: Sugimoto et al. 1990; Ito et al. 1991) has yielded breakthrough performance in the field of stellar dynamics, providing a simple software interface for its users while delivering performance in excess of 1 Tflops. The developers of the current (GRAPE-4) system were awarded the 1995 Gordon Bell Prize for their efforts. The purpose of this proposal is to investigate how this highly successful teraflops system might be expanded into the petaflops domain.

Several key features have contributed to the success of the GRAPE series of special-purpose computers. Two particularly critical aspects are their simple design (with the concomitant short development time scale) and the straightforward software interface (access to the hardware through FORTRAN and C library calls). In addition, their integration into large-scale simulations requires minimal restructuring of existing programs, and GRAPE has been incorporated into a wide variety of codes, including those involving high-order integrators, individual particle time-step schemes, and hierarchical (tree) data structures. The modularity and scalability of the GRAPE machines ranges from the one-of-a-kind, massively parallel GRAPE-4, to small-scale, multi-chip boards in use at many institutions worldwide.

Our goal here is to determine (1) precisely where the bottlenecks in the present approach lie, and under what circumstances they become apparent, (2) to what extent the GRAPE design can be scaled up to achieve even higher performance, and (3) how we can broaden the base of applicability of the hardware into other areas of computational science, in order to attract multi-disciplinary support. We plan to address these issues by presenting a design for a next-generation GRAPE system that achieves a balance between raw performance and a wider range of applications, while retaining the key elements of previous success.

1.1. *Computational Bottlenecks in N-body Simulations*

N-body simulations are used in many areas of computational science. Their importance lies in the fact that they provide a largely assumption-free means of determining the behavior of systems whose dynamics is dominated by long-range interparticle forces. Once a system enters the regime where linearized treatments can no longer describe its evolution, a fully self-consistent treatment of its nonlinear development becomes essential; N-body simulations are the indispensable tool for following that development. Astrophysical N-body examples (involving gravitational interactions) span the entire range of astronomical research, from the formation of the solar system, to the evolution of star clusters, to the formation of large-scale structure in the universe (see §2). We will concentrate on astrophysical applications within this proposal, in large part because that is the context within which the existing GRAPE hardware was designed. However, many non-astrophysical N-body applications exist as well, perhaps the most important being molecular dynamics, and these will also be pursued in our design study.

Even when some approximation can be tolerated and efficient algorithms are used, as discussed in more detail below, problems which entail the computation of long-range interparticle forces can rapidly come to consume all CPU cycles available on any given computer. Problems with dynamic ranges of 10^7 in mass, 10^{14} in length, and 10^{21} in time are not uncommon in astronomy, and simulations of these systems can expand to fill all available computational space simply by pushing farther toward the ideal dynamic range. The following example illustrates this point for the prototypical GRAPE application.

For star-cluster simulations, the calculation must extend for $\mathcal{O}(N)$ particle orbits. In the absence of any simplifying assumptions, determination of the force on a particle requires $\mathcal{O}(N)$ operations. One might try to reduce this cost by using treecodes or multipole schemes (Barnes & Hut 1986, Greengard & Rokhlin 1987) , but the high precision required and large dynamic ranges involved make these schemes currently uncompetitive with the best direct-summation algorithms (e.g., McMillan & Aarseth 1993). With N particles, $\mathcal{O}(N)$ operations per particle, and $\mathcal{O}(N)$ orbits, the total number of operations over the entire simulation scales at least as $\mathcal{O}(N^3)$. In fact, accuracy tolerances become more stringent as the length of the simulation increases, so a more realistic scaling is $\mathcal{O}(N^{3.5})$ (Hut, Makino, & McMillan 1988). The largest value of N that has been simulated on conventional supercomputers is $\sim 10^4$ (Spurzem & Aarseth 1996); the GRAPE-4 can tackle problems with $N \sim 2-5\times10^4$; real globular star clusters contain $\sim 10^5 - 10^6$ stars. Because the memory requirements for these simulations are only a few tens of Mbytes, the calculation in this case is purely limited by the CPU speed.

In problems of this type, the determination of an immense number of interactions presents a substantial barrier to the inclusion of more realistic physics. Removal of this bottleneck will allow the physics, rather than the numerics, to become the prime focus of the investigation.

1.2. *The GRAPE Project*

The essential idea behind GRAPE is to perform the time-critical calculation in hardware (using a custom LSI chip), then operate many copies of that hardware in parallel to achieve very high speeds. Each chip calculates the total interaction $\sum_j \mathbf{a}_{ij}$ between a target particle i and a list of sources j, using pipeline architecture to efficiently sum the individual (inverse-square) pairwise interactions $\mathbf{a}_{ij}$. From a few to thousands of chips can be arranged in parallel, with the flexibility of spreading a single target calculation over many chips or assigning a different target to each chip. The simplicity of the inverse-square gravitational force makes it relatively easy to implement this interaction in hardware. However, some GRAPE systems are designed to evaluate arbitrary central forces, specified in the form of look-up tables. Given a long list of sources, stored once per step, then reused for a long list of targets, the communication time between host and GRAPE is negligible compared to the calculation time and a call to the GRAPE hardware becomes an extremely efficient replacement for software force evaluation.

In typical use, the host computer sends to the GRAPE the positions and masses of all source particles. Looping over target particles, the host then sends a target position and receives the forces accumulated ¿from all sources. For N particles as both source and target, there are $\mathcal{O}(N)$ values communicated compared to $\mathcal{O}(N^2)$ calculations. On some GRAPE systems, copies of particle positions and velocities are maintained in memory on the GRAPE hardware in order to minimize the total traffic between GRAPE and the host. Some versions of the hardware also calculate the potential and the jerk (derivative of the acceleration) on the target, and can return a list of neighbor sources (i.e., sources lying within a user-specified distance of each target). The rest of the calculation (moving the particles, timestep control, scheduling, etc.) is performed on the host. The distribution of computing load is such that the 1 Tflops GRAPE-4 system can be driven at better than 50% peak speed by a 100–200 Mflops host for $N \sim 2 - 5 \times 10^4$.

Modifying an existing program to use the GRAPE hardware is straightforward, and entails minimal changes. Subroutine and function calls (written in C or FORTRAN) to the GRAPE hardware replace the force-evaluation functions already found in existing N-body codes. Communication between host and GRAPE is accomplished through a collection of about a dozen interface routines. The force evaluation code which is replaced typically consists of only a few dozen lines at the lowest level of an algorithm. Thus, use of the GRAPE calls only for small, localized changes which in no way inhibit future large-scale algorithm development. The GRAPE interface has been successfully incorporated into the Barnes-Hut tree algorithm (Barnes & Hut 1985; Makino 1991) and the P^{3}M scheme (Hockney & Eastwood 1988; Brieu, Summers, & Ostriker 1995).

Building on these ideas of simplicity of function, design, and interface, development of the GRAPE projects has proceeded rapidly (see Table 1). The initial GRAPE-1 machine (240 Mflops) in 1989 demonstrated the feasibility of the basic approach, while the GRAPE-1A improved the communication bandwidth for a broader set of applications. At this point it was recognized that not all applications needed full IEEE precision, and two parallel tracks emerged, with the high-precision GRAPE-2 made from commercially available chips and the lower-precision GRAPE-3 developed as a custom LSI chip. The chips used in current-generation GRAPE systems are manufactured by LSI Logic Corp.

The GRAPE-3 showed scalability in its 48 parallel chips (15 Gflops total), as well as modularity in the 8-chip GRAPE-3A boards sold to institutions worldwide (including Princeton, Cambridge, and Max Planck). The so-called "HARP" (Hermite AcceleratoR Pipeline) machines demonstrated a more complex force calculation implemented in hardware, and led to the 1692-pipeline GRAPE-4 machine that achieved 1.08 Tflops peak speed in summer 1995. The Gordon Bell Prize awarded to Makino & Taiji in December 1995 was for a simulation of binary black holes in the center of a galaxy, a calculation performed on 1/6 of the GRAPE-4 machine the previous spring. A third development track is represented by the GRAPE-2A and MD-GRAPE machines, which include a user-loadable force look-up table that can be used for arbitrary central force laws (targeted at molecular dynamics applications). Overall, the pace of development has been impressive: 10 special-purpose machines with a broadening range of applications and a factor of 4000 speed increase in just over 6 years.

1.3. *Proposal: The Next Generation of GRAPE Systems*

The design of the GRAPE hardware is such that, once all pipelines are filled, each chip produces one new interparticle interaction (corresponding to ~ 60 floating-point operations) every three clock cycles. For a clock speed of 30 MHz, a peak chip speed of ~ 0.6 Gflops is achieved. We note that the present chip represents 1992 technology (1 μm fabrication line width). Even if no changes were made in the basic design, we estimate that advances in fabrication technology would permit more transistors per chip and increased clock speed, enabling a 50–100 MHz, 10–30 Gflops chip in a 1996 start (0.35 μm line width), and a 100–200 MHz, 50–200 Gflops chip with 1998 (0.25 μm) technology. Based on these projected performance improvements, we conclude that $\sim 10^4$ GRAPE-6 chips of 100 Gflops each could be combined to achieve petaflops speeds by the year 2000.

Thus, remarkably, petaflops-scale computing (for N-body problems, at least) seems attainable within five years, without the need for major breakthroughs in either chip design or fabrication technology. However, the anticipated increase in cost ($\sim$\$10 million, compared to $\sim$\$1 million for the GRAPE-4) requires that we broaden substantially our application base. We must assess the desirability, feasibility, and performance impact of redesigning all or part of the GRAPE system, and of adding new functionality, to allow new problems to be studied. In addition, we must reassess algorithms currently in use to optimize the performance gain realized by the proposed new system. The next generation GRAPE system has exciting prospects, but ones which must be carefully considered.

2. Applications

An important part of this design study is to explore and identify applications which can utilize the GRAPE hardware, as well as to consider how the hardware can be generalized to encompass a larger range of problems.

2.1. *General Characteristics*

Applications which can have an efficient GRAPE implementation have several defining characteristics. The primary requirement is that the application be dominated by the calculation of a pairwise interaction over many pairs. Although we are accustomed to thinking in terms of particle calculations, any discretized element may be considered. The interaction between elements must be dependent only on the distance between them, and perhaps also on their relative velocity and some scaling factor, such as mass or charge. The total interaction on each element must be a summation over a large number ($\geq 10^3$) of pairwise interactions to make communication time with the hardware inconsequential. To maximize the impact of the GRAPE speed-up, the calculation of this interaction should be a large fraction of the total CPU time of the full calculation. These characteristics probably mean that the simulation cost will scale as a large power of the number of elements, typically N^{2-3}. Note that we are considering here the total CPU time, not just the cost of a single timestep (as is often quoted for algorithms).

Finally, the largest problem one can run should ideally be constrained by the speed and number of cycles (CPU limited), and not by the size of the available memory (RAM limited). RAM limited applications will likely run faster with GRAPE, and perhaps allow more parameter space to be explored, but GRAPE will not enable larger problems to be broached. Some problems may also benefit simply by being able to move ¿from crowded supercomputers to local workstations, avoiding file transfers, job queues, and limited time and memory allocations.

For each potential application, our study will assess the suitability and impact of the GRAPE-6 by considering several critical points:

- The status and success of previous GRAPE work in this area.
- Detailed estimates of computing requirements, and identification of current and future bottlenecks.
- How the problem might be adapted to the simple structure of GRAPE.
- How GRAPE might be generalized to better fit the problem.
- Requirements placed on the host computer by the problem.
- Advantages one might expect over a general-purpose supercomputer implementation in the year 2000.

2.2. *Astrophysical Applications*

Within the general area of astrophysics, we identify five basic research topics which are heavily dependent on the computation of interparticle gravitational forces. We discuss the problem of globular cluster dynamics in detail, as it was the prime focus of previous GRAPE development, then briefly give descriptions and notes on the others.

- *Globular cluster dynamics.* Globular clusters are dense concentrations of about a million stars. They are found in all large galaxies and appear to be among the oldest objects in the universe. To a dynamicist, they represent a fantastic laboratory for studies of the long-term evolution of systems of gravitating point masses. As stated earlier, the computation cost of this problem scales as $N^{3.5}$. Hut et al. (1988) estimated that following a 60,000-body system for its entire dynamical lifetime would require a total of $\sim 10^{19}$ floating-point operations, and the original GRAPE development targeted this problem. The GRAPE-4 machine enabled the first numerical confirmation of theoretically predicted chaotic oscillations in the rapidly evolving cores of these systems. Scaling to more realistic globular cluster parameters ($N \sim 5 \times 10^5$, say) indicates a need for a petaflops-class machine. As the hardware is tailored to this problem, we expect our scaling estimates to be reliable. A 10 Gflops host with 2 Gbyte/s I/O bandwidth will be required to achieve 50% peak speed for $N \sim 5 \times 10^5$. Extrapolating current performance trends, we estimate that general-purpose supercomputers will be unable to address this problem until at least the year 2010.

- *Dynamics of binary black holes in galactic nuclei.* The centers of galaxies, or galactic nuclei, are regions of extremely high stellar density. Their dynamics becomes particularly interesting when one includes the possibility that most galactic nuclei may contain a black hole with mass of order a million times that of normal stars. These black holes are thought to have formed by the merging of smaller black holes, raising the complex dynamical problem of determining the behavior of two massive black holes orbiting within a dense stellar field. Some progress can be made using a smoothed potential to represent the stars, but unraveling the detailed dynamics of the stellar field (since that is what is observable in telescopes) requires high accuracy N-body studies. The CPU bottlenecks here are similar to those in the globular cluster problem.

- *Planet formation and planetary ring dynamics.* In planet formation studies, the particles represent small, asteroid-scale bodies which collide and merge to build up larger bodies, eventually leading to planets. Planetary ring dynamics is concerned with the long-term evolution of a system of ring particles, in the presence of planetary tidal fields and perturbations and resonances due to moons. In either case, the key point is that a wide range of mass scales ($\sim 10^7$) must be covered, with the number of objects increasing exponentially with decreasing mass. A petaflops machine will enable a realistic simulation of dynamical friction, where a single heavy particle is surrounded by a sea of 10^6 particles, each with a mass ten thousand times smaller, for the period required to form planets ($\sim 10^5$ dynamical time scales).
- *Galaxy evolution and interactions.* The dynamics of galaxies is another problem which covers a very wide range in mass and length scales. One is interested in the dynamics responsible for the characteristic morphological patterns of galaxies, as well as in the complex interactions that occur when galaxies collide and merge. In these applications, tree codes (Barnes & Hut 1986) have been extremely useful, because some degradation in the accuracy of long-distance forces can be tolerated without affecting the character of the solution. The tree algorithm has been adapted for use with the GRAPE architecture (Makino 1991).
- *Large-scale structure and galaxy formation.* Following the formation of structure on the largest scales of the universe down to the collapse of individual galaxies is a problem whose entire range has yet to be covered in a single simulation. Current work with the GRAPE-3A hardware shows that supercomputer-class simulations can be moved to local workstations (Brieu et al. 1995). The largest simulations in this field are close to being RAM limited, but CPU limits will become more serious as increased resolution imposes stricter timestep constraints. An important consideration for this problem (and also galaxy dynamics) is the incorporation of smoothed particle hydrodynamics (SPH, see §4.3) which is a key ingredient to continued advancement of the scientific results from these simulations.

In December 1995, our group held a workshop at the National Center for Supercomputer Applications (NCSA) to discuss possible astrophysical applications of the GRAPE-6 hardware. One result of this workshop is presented in Table 2. This table gives rough size estimates for the largest simulations possible on general-purpose supercomputers today and in the year 2000, along with corresponding figures for the GRAPE-4 (today) and the GRAPE-6 (in 2000), with an additional estimate of the time advantage gained using the GRAPE-6 hardware. In our comparisons, we assume that peak supercomputing speeds on the order of several Tflops will be available by 2000, and that high-end workstation performance will be in the 1 Gflops range. Although we tried to adopt pessimistic assumptions in our estimates, significant advantage clearly shows in several problem areas, with others requiring more evaluation. Determining the details and refining our rough estimates will be a central part of our study.

2.3. *Other Applications*

Astrophysics is not the only research area that can make use of the GRAPE hardware. Particle methods are in use in a range of applications. One field of prominence and extensive supercomputer usage is molecular dynamics. A second area is vortex methods for incompressible fluid-dynamical simulations. We will use our professional contacts and NCSA to seek out experts in these and other fields. We must learn both the characteristics of the problems and the algorithms presently used in their solution in order to evaluate the usefulness of a GRAPE implementation. When possible, ports of these codes to GRAPE hardware can be tested. If warranted, we may hold a workshop at NCSA for interested researchers in the targeted fields.

The bridge to the molecular dynamics community is already under construction (Higo et al. 1994). The main difference between the calculations considered in the previous section and those used in molecular dynamics is the fact that molecular dynamics force laws do not always have an r^{-2} dependence. Molecular dynamics applications require a more generalized force law, rather than a hard-wired inverse square law. The recently completed MD-GRAPE hardware (Taiji et al. 1995, Fukushige et al. 1996) allows the user to load a look-up table onto the chip, enabling the computation of an arbitrary pairwise force law. Such a feature would also be useful in speeding up the implementation of the P^3M algorithm (§4.1). The large size of the molecular dynamics community may provide the critical mass to support wide implementation of this form of the hardware.

3. Hardware Design Options

As a basis for discussion of improvements, we first describe the current system in a little more detail. The heart of the GRAPE-4 machine is the HARP chip, which calculates the gravitational force and its first time derivative. One HARP chip consists of 400k transistors and is fabricated in 1 μm technology. To utilize the individual timestep algorithm, another custom chip, the Prometheus predictor chip, predicts the position and velocity of all source particles based on the target particle's timestep. One Prometheus chip, 48 HARP chips (47 plus one spare), and a memory unit are packaged together on each processor board.

The processor boards are connected in four clusters, each consisting of 9 processor boards and a controller board, connected by a backplane bus with a peak transfer rate of 192 Mbyte/s. The controller boards are connected via an interface board to the TURBOchannel I/O bus of a DEC Alpha 3000 workstation. The connection between the controller board and the interface board is a coaxial flat cable with 32 bit data width and a bandwidth of 64 Mbyte/s. The TURBOchannel has 100 Mbyte/s peak bandwidth which, along with the main memory bandwidth of the DEC 3000, constitutes the limiting bandwidth of the current configuration. Data transfer within the host takes more time than the actual data transfer between the host and the GRAPE.

In summary, the present GRAPE-4 system is composed of a DEC Alpha workstation connected to 4 clusters, with a total of 36 processor boards and 1692 HARP chips. A schematic overview of the system, along with the logical layout of each chip, are shown below (figures taken ¿from Makino et al. 1993; note that the first diagram depicts a slightly different GRAPE-4 configuration, with 16 boards per cluster and 16 processors per board).

Schematic block diagram of the GRAPE-4 system.

Logical architecture of the HARP chip. The symbols x, v, and m represent particle positions, velocities, and masses, respectively. The Σ registers accumulate the acceleration, jerk (two sums) and potential on particle i.

3.1. *Chip Design Options*

The basic options for chip design have been well defined during GRAPE development (see Table 1). The odd-numbered GRAPE machines use limited-precision data paths for applications which do not require full IEEE precision throughout the calculation. Note that the width of the data path is the single most important factor determining the size, and therefore the cost, of the processor chip. The even-numbered machines use full-precision data paths and have progressed to incorporate the hermite integration and individual timestep schemes. The MD-GRAPE (and GRAPE-2A) includes a user-loadable look-up table for the force law. In the interest of maximizing the number of applications, our study will consider the consequences of trying to combine all three tracks in a single chip.

The GRAPE-4 contains all the functionality of the the GRAPE-3, but it is overkill for the problems GRAPE-3 was designed to attack. The major issue here is the considerable extra expense for the full-precision data path and the hermite and individual timestep hardware. Furthermore, the I/O objectives of the two machines are somewhat at cross purposes, as will be considered below. The inclusion of a look-up table for the force poses a more difficult challenge. In MD-GRAPE, the look-up table takes up about 1/3 of the total area of the chip. This size is necessary to provide full single-precision accuracy for a fairly wide dynamic range, and to support two independent tables (as would be needed for acceleration and jerk calculations). Using such a large percentage of transistors for a table will limit the number of pipelines possible, and hence the speed of the chip.

The merit of these options must be carefully evaluated against the performance penalty and additional development cost of the chip, as well as development time for the total system. By comparing designs optimized for each type of machine, we can address the inefficiencies of a combination design.

3.2. *I/O Options*

For the current GRAPE-4 system, a new host interface board to connect the controller board to a PCI bus is under development. With a host that supports multiple PCI buses and much improved memory bandwidth (a DEC 8400 system), the I/O performance of the present system will be improved by nearly a factor of 10.

For the next generation, even more improvement is required. High-precision, long-term simulations require I/O bandwidth on the order of 1 Gbyte/s, but with a communication latency on the order of microseconds because the target lists can sometimes be short when using individual particle timesteps. For low precision, short-term simulations, much longer latency is acceptable, while much higher bandwidth (probably exceeding 10 Gbytes/s; see the P^3M discussion in §4.1) is desirable to push to very large particle counts. Thus, the communication architecture of GRAPE-6 must balance the somewhat competing goals of short latency and high bandwidth.

We will attempt to evaluate the I/O options likely to be available on supercomputers in the first decade of the next century, when GRAPE-6 will be useful. Many outstanding practical problems need to be addressed: what physical interface will be used (serial/parallel, copper/optical), whether one of the emerging standards will be adopted (ATM, Fiber-Channel, SCI, etc.), what logical interface will be adopted, and so on. The advice and expertise of staff at NCSA will prove invaluable here. Substantial design effort will be necessary to find a combination which maximizes the benefits over various applications.

3.3. *Host Computer Requirements & Architecture Options*

Demands made on the host computer depend strongly on the application. The problem areas outlined above have requirements for high flops count, fast data transfer, and large memory on the host computer. The architecture that will be used in high-performance computers in the year 2000 is not entirely clear. Present industry trends seem to point toward SMP-based CC-NUMA clusters. NCSA has adopted this architecture as its strategy and has both the HP/Convex Exemplar and SGI Power Challenge Array available. However, real-world testing of this architecture has just begun, and we need to evaluate other possible front-end architectures, such as the IBM SP-2 or the Cray T3E. Researchers at IBM's T. J. Watson Labs have expressed specific interest in our project and its adaptability to the SP-2.

The organization of the GRAPE-6 system as a whole is also an open question. The design of GRAPE-4 was simplified by having 4 clusters which operated independently, but a single I/O channel will be incapable of handling the traffic from future large simulations. A machine like the SP-2 with a board or cluster attached to each processor can provide scalable I/O, but data motion between processors may become a bottleneck. The optimal configuration depends very much on such factors as the technology available to connect multiple boards, the number of chips that can be integrated on a single board, and the communication latency between processors on the host.

Finally, the GRAPE-6 system will likely have a minimum modularity that will be useful in very small configurations, such as 10–50 chips (a single board). Like the GRAPE-3A board currently in use at many institutions, a moderately priced (∼\$20,000) GRAPE-6A teraflops board can act as a "turbocharger" for workstations, allowing wide dissemination of the fruits of the development effort. Such boards might reduce loads on the supercomputing centers (for certain applications) by offering similar performance, but with much more convenience, at an affordable price. These goals place constraints both on the minimum size of the on board memory and on the board structure.

4. Algorithm Studies

The basic idea behind special-purpose hardware is that a brute-force approach can sometimes be more efficient than a sophisticated software attack on a problem. One then has the legitimate worry that an unanticipated new algorithm will leapfrog the hardware before it is complete. To avoid premature obsolescence, one must continue developing ways of incorporating advanced algorithms into the hardware, and adapting these algorithms to utilize the GRAPE functionality. In a sense, the algorithms are more fundamental than the applications, for once an algorithm has a GRAPE implementation, then all applications that use that algorithm are automatically ported. Algorithm studies will enable GRAPE to combine the best of both hardware and software development.

4.1. *Wide Dynamic Range Particle Algorithms*

GRAPE can be useful in particle algorithms which need wide dynamic range in length scales. In these cases, the algorithm invariably boils down to a direct-summation calculation over a local region, coupled with some more approximate method for larger-scale interactions. Efficiency and accuracy are balanced in setting the size scale of the local region. For large or very dense local regions, the direct summation calculation can consume considerable CPU time, and a GRAPE implementation is warranted.

The spatially adaptive Barnes-Hut tree algorithm has a GRAPE implementation (Makino 1991), but with a potentially limiting bottleneck: the construction of the tree data structure is performed on the host computer and can take up to 5% of the total CPU time per step. Hence, the potential for speed-up over a general-purpose machine may be limited to around a factor of 20. The exact breakdown of the CPU distribution for the different problem areas at various stages of clustering and for large simulations needs to be defined. Tree code development (e.g., Dubinski 1996) may provide a software route around this bottleneck, and it is not inconceivable to implement tree construction in hardware.

A GRAPE implementation of the P^3M (particle-particle/particle-mesh) algorithm, running on a SPARCstation 10 with a four-chip GRAPE-3A board, reaches a peak speed about one third that of a vectorized single-processor Cray C90 P^3M code. This code has already served to move some simulations from supercomputing centers to local workstations. While this is already impressive performance, straightforward improvement can be gained by going to parallel techniques to speed up the fast Fourier transforms performed on the front end, and by using the MD-GRAPE hardware to provide the desired truncated force law. The GRAPE code will be directly compared against research codes running on a variety of supercomputer platforms. This code also needs to be tested to estimate the necessary balance between host and GRAPE speed for GRAPE-6 class simulations. Another potential bottleneck, for which detailed estimates are essential, is the I/O bandwidth, because the algorithm uses many calls to GRAPE per timestep, each with different sets of particles. Pushing toward billion-particle P^3M simulations will provide the maximum bandwidth constraint (worst-case estimates are in the ~ 20 Gbyte/s range).

Preliminary analysis indicates that the fast multipole method (FMM: Greengard & Rokhlin 1987) can be adapted to GRAPE, but no coding or tests of its efficiency have been made. Possible drawbacks to FMM are the large amount of memory required and the efficiency of using individual particle timesteps. This potentially important algorithm merits further study.

4.2. *Smoothed Particle Hydrodynamics*

An important algorithm for many astrophysical applications is the use of smoothed particle hydrodynamics (SPH) to follow compressible fluid flows. In some areas, major scientific questions now hinge on how fluid-dynamical processes shape structure development. SPH is a particle-based approach to gas dynamics which determines local quantities by kernel estimation over nearby particles. The lists of neighbor particles returned by the GRAPE hardware allow for an efficient SPH implementation in an existing GRAPE gravity code.

Several important questions remain about current GRAPE and SPH implementations (Umemura et al. 1993, Steinmetz 1996). First, the neighbor lists on the GRAPE hardware are of fixed length, so, if too many particles satisfy the neighbor criterion, the list overflows and is of no use. The percentage of such occurrences, and the performance hits they incur, must be characterized for various classes of problems. Second, the most efficient GRAPE and SPH implementations utilize a fixed number of neighbors for smoothing, which may lead to unphysical effects. Situations can develop where the hydrodynamics is resolved on scales much smaller than the gravitational softening scale, and a feedback loop between density and cooling could lead to excessive collapse. Careful tests of the numerical effects introduced and studies of alternative sampling techniques are warranted.

Finally, some researchers have proposed creating a hardware implementation of SPH. To assess the feasibility of such an undertaking, it will first be necessary to define an 'optimal' SPH version to hardwire, obtaining maximal flexibility from the board. The likely longevity of SPH's distinct advantages, in the face of ongoing improvements in adaptive mesh refinement techniques, must also be considered.

5. Summary & Discussion

We propose a design study of a petaflops-class computer for application to particle simulations in astrophysics and other areas of computational science. The crux of the system is special-purpose hardware which performs the most CPU-intensive part of a simulation (interparticle force computation), leaving the remainder of the calculation to a front-end computer. This basic structure has been refined in a series of GRAPE machines over the last 7 years, culminating in the 1.1 Tflops GRAPE-4 last summer. Over 40 GRAPE machines of various sizes are in use world-wide today. Initial estimates predict that a petaflops GRAPE-6 machine costing $10 million dollars is feasible by the year 2000. This proposed study will examine those prospects in detail.

The justification and support for building such a machine will have to come from a broad base of science and computation. We have identified a number of problem areas in astrophysics, ranging in scale from the formation of planets to the dynamics of stars to the large-scale structure of the universe, which are well suited to GRAPE methods of solution. Preliminary estimates (Table 2) suggest that substantial gains are likely over conventional general-purpose supercomputers available in 2000, with a time advantage of over a decade in some cases. Outside astrophysics, the subjects most likely to benefit from a special-purpose computational engine of this sort are molecular dynamics and vortex methods in CFD. We will obtain robust estimates of the applicability of the GRAPE-6 to each of the target problems. We will also explore which software algorithms mesh with future GRAPE design and can scale well to larger simulations. Adapting the best algorithms to utilize GRAPE as well as incorporating advanced options on board will take advantage of both software and hardware improvements.

These applications have a broad range of accuracy requirements and front-end computational loads, meaning that all aspects of the GRAPE-6 system—hardware and software performance, communications bandwidth, and front-end capability—must be carefully evaluated in the light of potential usage. We will attempt to determine the desirability and the feasibility of introducing new special-purpose components into the proposed hardware design—for example, to speed up fluid-dynamical portions of simulation codes, or to allow arbitrary force laws to be computed. A key issue will be to ascertain whether our goals can be met with a single design without serious performance degradation, or if a series of more problem-specific designs is mandated. To allow for a variety of host computer architectures, we must determine whether the GRAPE-6 can be constructed so as to be distributed across many nodes of a massively parallel system if necessary. Finally, a smaller, workstation add-on version of the hardware will enable wide dissemination of our work to others, and will be explicitly included in the plans. Throughout, we will seek a hardware design that balances variety of applications against maximum performance while retaining the essential simplicity that has made GRAPE a success.

The result of our study will be a considered evaluation of, and complete outline for, the next-generation GRAPE system. We will identify the major design tradeoffs, and quantify them where possible with large simulations on current hardware. One or more design options will be outlined, specifying performance, application base, host computer requirements, interface requirements, and estimated total system cost. Our recommendations will choose the best paths to follow at various levels of funding. From this study will emerge an evaluation of special-purpose N-body hardware as a long-term high performance computing solution.

The PI team combines the necessary talents for a thorough study of all aspects of the proposal. Our ranks include the developers who have taken the GRAPE project from initial idea, through ‘proof of concept’ hardware, to a functioning Teraflops machine. Our group also includes those with the greatest experience using the GRAPE hardware and programming new algorithms for a wide range of applications, from stellar dynamics to cosmology. As demonstrated at our initial workshop at NCSA last December, we have established dialogues with, and generated enthusiasm from, researchers across many areas of astrophysics. Beyond our connections at NCSA, we also have had discussions and queries from workers at Pittsburgh Supercomputing Center, IBM's T. J. Watson Labs, and Bell Labs. At heart, we are astrophysicists who rely on computational tools and have a keen eye on enabling technologies that can advance our science. Our team combines the technological, computational, and scientific expertise required to make this project a success.

In closing, we address specifically the key criteria specified in the program announcement:

- *Originality, novelty, and practicality of design.*
  Special-purpose hardware has been tried before, with checkered success. Simplicity in both function and design is the key to the GRAPE series. The practicality of its design has been proved over the past seven years.
- *Programmability and ease of use.*
  Since the GRAPE hardware simply replaces a small section of a general-purpose application code with a functionally equivalent library call, it is extremely easy to use. Most of the application program runs on the front end and programmability is unimpaired.
- *Degree of integration of software and hardware into an effective unified system.*
  The GRAPE-4, with 1692 chips and specific hardware for hermite intergration and individual timesteps, amply demonstrates that hardware and software can be effectively integrated into a unified system.
- *Suitability for applications involving complex and/or dynamic data structures.*
  The functional simplicity of the GRAPE hardware places no restrictions on the complexity of host data representation. GRAPE systems have been incorporated into applications involving monolithic arrays as well as hierarchical (tree) data structures. The calls to GRAPE involve only the low level elements which are independent of the overall data organization.
- *Expected performance across a range of applications.*
  We have identified a number of very different astrophysical contexts in which the hardware may be usefully employed. In addition, other areas of computational science, such as molecular dynamics and vortex fluid dynamics methods, are likely to benefit from this technology.

Although GRAPE is a special-purpose system, it incorporates several features that may make it interesting to users outside the particle simulation community. GRAPE achieves high speed by integrating a number of floating point units on a single chip, then using many of these chips in parallel with little additional overhead. This is possible for several reasons: (i) N-body simulations are very compute-intensive, so the amount of memory required is relatively small, and its cost is negligible. (ii) GRAPE uses a hardwired pipeline specialized for the force calculation, and several pipelines can share memory, reducing memory bandwidth by a very large factor. (In GRAPE-4, it is 0.001

word/flop). (iii) The hardwired pipeline also make it possible to adjust the floating-point precision used in different parts of the chip so as to be "just suffient" everywhere. In a general-purpose system, if some operation needs 64-bit precision, in most cases everything must be done in 64 bit.

Table 1

Summary of GRAPE Hardware

Limited-Precision Data Path

Machine	Year	Peak Speed	Notes
GRAPE-1	1989	240 Mflops	Concept system, GPIB interface
GRAPE-1A	1990	240 Mflops	VME interface
GRAPE-3	1991	15 Gflops	48 Custom LSIs, 10 MHz clock
GRAPE-3A	1993	5 Gflops/board	8 chip version for distribution, 20 MHz, PCB implementation

Full-Precision Data Path

Machine	Year	Peak Speed	Notes
GRAPE-2	1990	40 Mflops	IEEE precision, commercial chips
HARP-1	1993	180 Mflops	"Hermite" pipeline
HARP-2	1993	2 Gflops	Evaluation system of the custom chips to be used in GRAPE-4
GRAPE-4	1995	1.1 Tflops	The Teraflops GRAPE, 1692 pipelines

Arbitrary Force Law

Machine	Year	Peak Speed	Notes
GRAPE-2A	1992	180 Mflops	Force look-up table
MD-GRAPE	1995	4 Gflops	Custom chip with force look-up table

Table 2

Particle Number of Largest Simulation Feasible
and GRAPE-6 Time Advantage†

Problem Area	1995 General Purpose Supercomputer	1995 GRAPE-4	2000 General Purpose Supercomputer	2000 GRAPE-6	GRAPE-6 Time Advantage
Planet Formation & Rings	5×10^3	5×10^4	5×10^4	10^6	7 years
Globular Cluster Evolution	10^4	5×10^4	5×10^4	5×10^5	10 years
Black Hole Binary in Galactic Nucleus	10^5	10^6	10^6	3×10^7	10 years
Galaxy Evolution & Interactions	10^6	2×10^6	3×10^7	10^8	3 years
Galaxy Evolution & Interactions with SPH	2×10^5	10^6	5×10^6	10^7	2 years
Large Scale Structure & Galaxy Formation	3×10^7	3×10^7	5×10^8	5×10^8	0 years**
Large Scale Structure & Galaxy Formation with SPH	4×10^6	3×10^6	10^8	3×10^8	3 years

† All numbers are rough estimates for initial evaluation purposes only

** Assumes RAM limited calculation