Implementation of Fast Multipole Algorithm on Special-Purpose Computer MDGRAPE-2

Nguyen Hai Chau, Atsushi Kawai and Toshikazu Ebisuzaki
Computational Science Division
Advanced Computing Center
RIKEN (Institute of Physical and Chemical Research)
2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan

ABSTRACT

N-body simulation is a time consuming task in which force calculation part is most dominant part. The simplest and most accurate algorithm for force calculation is direct summation which has time complexity $O(N^2)$. It is not practically suitable for large-scale simulations on most general-purpose computers. To cut down cost of force calculation one applies fast algorithms or performs force calculation on special-purpose hardware. GRAPE is a special-purpose computer designed for force calculation between point-charge or point-mass particles. It performs force calculation much faster than general-purpose computers of similar cost. However the time complexity of direct force calculation on GRAPE is still $O(N^2)$. In this paper, we deal with the implementation of fast multipole algorithm whose time complexity is $O(N)$ on special-purpose computer MDGRAPE-2. We present our experimental results for up to four millions particles system. Performance and accuracy of FMM on GRAPE is presented. Comparison of FMM with treecode and direct summation on GRAPE is also given.

Keywords: N-body simulation, fast multipole method, GRAPE special-purpose hardware, Anderson's method, P^2M^2 method, tree algorithm.

1. INTRODUCTION

N-body simulation was devised in 1950s and widely used since 1970s when digital computers became powerful and affordable. It is an orthodox method of studying particle systems nowadays. The N-body simulation problems require very high computational cost in which force calculation part is most expensive. The cost of direct summation algorithm for force calculation is $O(N^2)$, therefore calculation time grows as rapidly as the number of body N increases. There are two main approaches to reduce force calculation cost of N-body simulation. The first approach is to apply fast algorithms such as treecode [4] or fast multipole method [5] which reduces calculation cost down to $O(N.\log N)$ and $O(N)$, respectively. The second is to perform N-body simulation with special purpose hardware.

GRAPE (GRAvity PipE) [11], [12] is a special-purpose hardware for the calculation of force between point-mass or point-charge particles. A typical GRAPE system consists of a general-purpose computer (hereafter we refer as 'host computer') and a GRAPE board. GRAPE calculates force and the host computer performs everything else.

Although GRAPE can speed up force calculation significantly, about 100-1000 times faster than general-purpose computers of the same price, calculation cost is still proportional with N^2 for direct summation algorithm. Implementation of fast algorithms on GRAPE therefore becomes demanded for large-scale simulations. The first implementation of a fast algorithm (treecode) on GRAPE is presented in [10]. The implementation combines advantages of both fast algorithm and fast hardware and obtains typically 30-50 times faster than the treecode without GRAPE. This leads us to implement fast multipole algorithm on GRAPE.

In this paper we describe the first implementation of fast multipole method (FMM) on special-purpose hardware MDGRAPE-2, a member of the GRAPE family. Since GRAPE can only operate with point-charge or point-mass interaction, we modify the original FMM so that it can be run on GRAPE.

Remaining parts of the paper are organized as follows. Section 2 gives a summary of FMM and its variants. Section 3 deals with GRAPE system architecture briefly. In section 4 we present our implementation of modified FMM on GRAPE and experimental results. Finally, conclusions are given in section 5.

2. SUMMARY OF FMM, ANDERSON'S METHOD AND P^2M^2

The FMM is presented in [5] for two dimensional

case. In the following we describe briefly non-adaptive version of FMM for three dimensional case. We assume all particles are uniformly distributed in a unit cube.

Initially we construct a oct-tree structure by hierarchical subdivision of the cube. The subdivision process starts from the root cell at refinement level $l = 0$ of the tree which contains the whole system. The subdivision is repeated recursively for all child-cell and stops when r reaches an optimal refinement level n. We call this process as tree construction.

Next we form multipole expansions for all leaf cells then form multipole expansions for all non-leaf cells in coarser level by shifting and adding up expansions of their children. This step is called as upward pass as we traverse the tree from leaf to root. In this step we have performed multipole expansion transitions M2M.

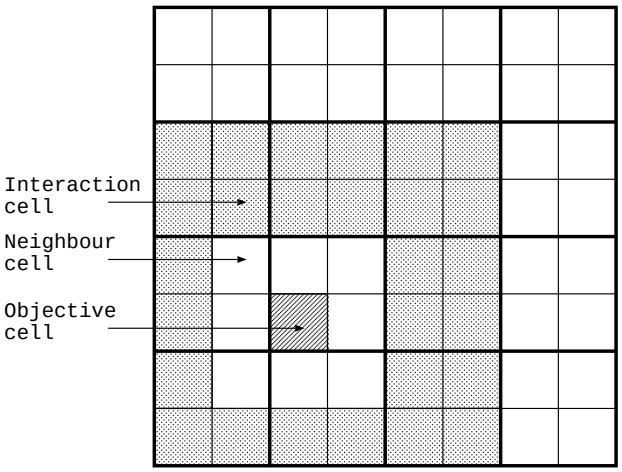

Figure 1: Neighbour and interaction list of hatched cell

The last step is downward in which we traverse the tree from root to leaf. First we form local expansion at geometric center of each cell due to potential field of its interaction list.

The interaction list of a cell is the set of cells which are children of the nearest neighbours of the cell's parent and which are not neighbours of the cell itself. Neighbour list of a cell is set of cells in the same level with the cell in question and have contact with the cell. Figure 1 shows an example of neighbour and interaction list of a cell.

The potential field due to interaction list of the cell are calculated by converting their multipole expansions to local expansion of the cell and adding them up (M2L conversion). Then we sum up local expansions at different refinement levels to obtain total potential field at all leaf cells.

Finally we calculate force on each particles in all leaf cells by summing up contribution of far field and near field force. The near field contribution is directly calculated by evaluating the particle-particle force. The far field contribution is calculated by evaluating local expansions at position of the particles (L2L transition). The time complexity of FMM is $O(N)$.

Anderson [1] proposed a formulation of the multipole expansion. The purpose of his method is to simplify the implementation of FMM. Here we briefly describe his method.

Anderson's method is based on the Poisson's formula. This formula gives solution of the boundary value problem of the Laplace equation. When potential on the surface of a sphere of radius a is given, the potential Φ at position $\vec{r}$ is expressed as

$$\Phi(\vec{r}) = \frac{1}{4\pi} \int_S \sum_{n=0}^{\infty} (2n+1) \left(\frac{a}{r}\right)^{n+1} P_n \left(\frac{\vec{s}\cdot\vec{r}}{r}\right) \Phi(a\vec{s}) ds \tag{1}$$

for $r > a$, and

$$\Phi(\vec{r}) = \frac{1}{4\pi} \int_S \sum_{n=0}^{\infty} (2n+1) \left(\frac{r}{a}\right)^{n} P_n \left(\frac{\vec{s}\cdot\vec{r}}{r}\right) \Phi(a\vec{s}) ds \tag{2}$$

for $r < a$. Here $\Phi(a\vec{s})$ is the given potential on the sphere surface. The area of the integration S covers the surface of a unit sphere centered at the origin. The function P_n denotes the n-th Legendre polynomial.

In order to use the formula as an replacement of the multipole expansion, Anderson proposed a discrete version of the formula, i.e., he truncated the right-hand side of the Eq. (1)–(2) at finite n, and replaced the integrations over S with numerical ones using the spherical t-design [6]. The relation he obtained is expressed as

$$\Phi(\vec{r}) \approx \sum_{i=1}^{K} \sum_{n=0}^{M} (2n+1) \left(\frac{a}{r}\right)^{n+1} P_n \left(\frac{\vec{s}_i\cdot\vec{r}}{r}\right) \Phi(a\vec{s}_i) ds \tag{3}$$

for $r > a$, and

$$\Phi(\vec{r}) \approx \sum_{i=1}^{K} \sum_{n=0}^{M} (2n+1) \left(\frac{r}{a}\right)^{n} P_n \left(\frac{\vec{s}_i\cdot\vec{r}}{r}\right) \Phi(a\vec{s}_i) ds \tag{4}$$

for $r < a$.

Anderson's method uses Eq. (3) and (4) for M2M and L2L conversion, respectively. The procedure of the tree construction part is the same as that of original FMM.

P^2M^2 is very similar to Anderson's method. The difference is that P^2M^2 uses the masses/charges distribution on the surface of a sphere instead of potential values. The pseudo particles must be distributed on the sphere so that they exactly express the coefficients of multipole expansion. Distribution of pseudo particles is given by spherical t-design [6]. We applied Makino's approach to define pseudo particles'

masses/charges. Positions of pseudo particles are fixed and masses/charges are obtained as follows:

$$M_j = \sum_{n=1}^{N} m_i \sum_{l=0}^{p} \frac{2l+1}{K} \left(\frac{|r_i|}{a} \right)^l P_l(cos\gamma_{ij}), \qquad (5)$$

where r_i is position of physical particle, R_j is K points on the unit sphere defined by t-design, γ_{ij} is the angle between R_j and r_i and P_l is l-th Legendre polynomial.

3. BRIEF DESCRIPTION OF THE GRAPE SYSTEM

GRAPE (GRAvity PipE) [11], [12] is a special-purpose hardware for the calculation of force between point-mass or point-charge particles. Figure 2 shows a typical GRAPE system consisting of a general purpose computer and a GRAPE hardware connected via a communication interface. The host computer sends positions and masses or charges of particles to GRAPE. GRAPE then calculates the gravitational or Coulomb forces due to these particles and results back to the host computer. GRAPE can speed up force

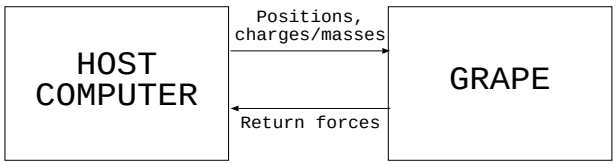

Figure 2: Basic structure of a GRAPE system.

calculation about 100-1000 times faster than general-purpose computers of the same cost and the time complexity of force calculation on GRAPE is $O(N^2)$ for direct summation algorithm. In contrast, the accuracy of direct force calculation on GRAPE is lower than that of host computer due to hardware limitation. For example, the error of relative force calculated by MDGRAPE-2 is around 10^{-8}. However this accuracy is far enough for almost all N-body simulations.

4. IMPLEMENTATION OF FMM ON SPECIAL-PURPOSE COMPUTER MDGRAPE-2

Since GRAPE can only deal with point-charge or point-mass interaction, only near-field part of original FMM can be implemented on GRAPE. Unfortunately, the far field force calculation is also a time consuming task if it is executed only on the host computer. Thus we modified FMM so that it can be run on GRAPE hardware using P^2M^2 and Anderson's method. In the following, we use notation p and q for expansion order of P^2M^2 and Anderson's method in Eq. (5) and (4), respectively. We will refer 'particle' as a real particle to distinguish with pseudo-particles given in P^2M^2 method. Summary of our modification of original FMM is as follows.

At first stage, instead of forming multipole expansions' coefficients, we distribute pseudo particles at points R_j on spheres located at geometrical centers of cells and calculate their masses/charges M_j based on spherical t-design [6] and Eq. (5). We will refer these spheres as P-spheres. The ratio between edge length of a cell and radius of its corresponding P-sphere is predefined. The number of pseudo-particles on the P-sphere depends on multipole expansion order p. We then use another type of sphere (hereafter refers as Φ-sphere) located in geometrical center of the cell with different radius from P-sphere. Spherical t-design is applied again to define $R_j^{'}$ points on Φ-sphere surface. Potential values at these points will be used for calculation of M2L and L2L stages. Number of $R_j^{'}$ points on Φ-sphere surface depends on q. The potential values Ψ_j at these $R_j^{'}$ points are accumulated during M2L and L2L stages. The contributions potential to values Ψ_j include direct potential from interaction list and far field potential.

In M2L stage, direct potential contributions at $R_j^{'}$ points of Φ-sphere of each non-leaf cell are calculated due to pseudo-particles corresponding to interaction list of the cell by GRAPE hardware.

In L2L stage, far field potential contributions are obtained by shifting local expansion on Φ-sphere of parent cell to center of child cells using inner-sphere approximation given in Eq. (4). This process is repeated for all level of the oct-tree and finally potential values at $R_j^{'}$ points of Φ-sphere of all leaf cells are obtained.

Force on each particle will be sum of near field and far field force. For each leaf cell, the near field force is calculated directly by GRAPE. This contribution includes force from particles belonging to the cell itself, the cell's neighbour list and pseudo particles on P-sphere of cell's interaction list.

The far field force is obtained by deriving force approximation from potential given in Eq. (4) as follows. Let $\vec{x} = (x, y, z)$ be position of a particle of the system. Let u be dot product $\vec{s_i}\vec{x_p}$ and r be length of vector $\vec{x}$. The x component of far field force on $\vec{x}$ is then expressed as:

$$-\frac{\partial \Phi}{\partial x} = \sum_{i=1}^{K} \sum_{n=0}^{M} \left(nxP_n(u) + \frac{ux - s_{ix}r}{\sqrt{1-u^2}} \frac{\partial P_n(u)}{\partial x} \right) (2n+1) \frac{r^{n-2}}{a^n} g(a\vec{s_i}) w_i, \quad (6)$$

where $g(a\vec{s_i})$ are potentials values at $R_j^{'}$ points on Φ-sphere, P_n is n-th Legendre polynomial and w_i are constant weight values. The y and z components are

obtained in the same way. The following gives detailed of our FMM implementation after the modifications mentioned above.

Tree construction stage.
Description: Assume particles are uniformly distributed on a unit cube. Choose a refinement level n, expansion order p of P^2M^2 and q of Anderson's method. We build an oct-tree by subdivision the cube into eight equals subcells. The subcells then divided into 8 smaller subcells. Repeat this process until we reach level n. By experiment, we found that $n \approx \log_8(N) - 1$ is the optimal refinement level for fastest force calculation.

M2M stage.
Description: Traverse the oct-tree from leaf to root to calculate masses/charges and positions of pseudo particles in all cells. At all leaf cells, calculate masses/charges and positions of pseudo particles based on mass and position of all real particles belonging to the cells. At non-leaf cells, masses and positions of particles are calculated based on masses and positions of pseudo particles of their children.

for $c = 1$ to 8^n begin
    Calculate pseudo particle's positions R_j and masses M_j of c based on masses (charges) and positions of particles belonging to c and Eq. (5).
end

for $l = n - 1$ to 0 begin
    for $c = 1$ to 8^l begin
        Calculate positions of points $R_j{}'$ on Φ-sphere of c.
        Calculate pseudo particle's position R_j and mass M_j of c based on pseudo particles' mass and position corresponding to child cells of c in level $l + 1$ and Eq. (5)

    end
end

M2L and L2L stages.
Description: Traverse the oct-tree from root to leaf in order to obtain Ψ_j potential values at all leaf cells. For each non-leaf cells, calculate potential contribution to Ψ_j at $R_j{}'$ of Φ-sphere surface due to all pseudo-particles corresponding to its interaction list. The far field potential contribution to Ψ_j are calculated repeatly while going down the oct-tree by using inner-sphere approximation given in Eq. (4). The range of integration is Φ-sphere of a non-leaf cell and evaluation points are $R_j{}'$ of its children. For each leaf cell, calculate force on each particle. Near field part is calculated by direct summation. Far field part is calculated by Eq. (6).

for $l = 0$ to $n - 1$ begin
    for $c = 1$ to 8^l begin
        Calculate potential contributions to Ψ_j at $R_j{}'$ on Φ-sphere surface of c due to pseudo particles corresponding to interaction list of c then accumulate these potential values.
        Using Eq. (4) to calculate far field potential contribution Ψ_j at $R_j{}'$ on Φ-spheres of all children d of c and accumulate the potential values to Ψ_j of d
    end
end

At finest level:
for $c = 1$ to 8^n begin
    Calculate force on particles due to near-field and far-field force. Near field force are calculated due to particles belonging to c, its neighbour list and pseudo particles belonging to its interaction list. Far field is calculated by Anderson's method. Force approximation is given in Eq. (6).
end

In the following we give our numerical results. We implemented FMM on GRAPE and tested the implementation on a system consisting of a MDGRAPE-2 board (192 GFlops equivalent) and a host computer (COMPAQ DS20E Alpha 21264/667MHz). We uniformly distributed particles within a unit cube centered at origin and measured time spend for one time step calculation for various accuracy. We tested the code with number of particles N from 65536 particles to 4194304 particles. We measured the relative error of potential and force averaged over all the particles. The averaged error is defined as follows:

$$e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{|\Phi_i' - \Phi_i|^2}{|\Phi_i|^2}}\ , \qquad (7)$$

where $\Phi_i^{'}$ is the potential value calculated by FMM, Φ_i is exact potential calculated by direct summation algorithm.

Fig. 3 shows calculation time for potential. From top to bottom, five curves are for direct summation, FMM with (p, q)=(1,1), (1,2), (2,2) and (2,3), respectively. The error e for each results are 6.8×10^{-3}, 3.6×10^{-4}, 1.6×10^{-3}, and 8.9×10^{-3}, respectively. Similarly, Fig. 4 shows calculation time for force.

The results show that the calculation time of our implementation scales as $O(N)$, and is achieving much better performance than direct summation code.

5. CONCLUSIONS

We have presented the implementation of fast multipole method on special-purpose computer

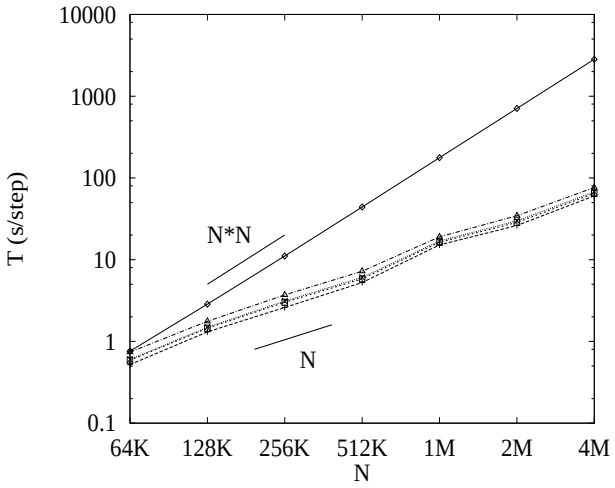

Figure 3: Potential calculation time is plotted against the number of particles N. The time is measured on a single MDGRAPE-2 board. From top to bottom, five curves are for direct summation, FMM with (p, q)=(1,1), (1,2), (2,2) and (2,3), respectively.

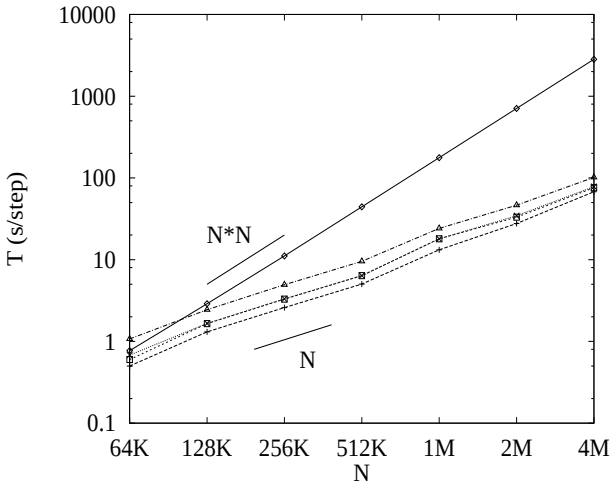

Figure 4: Same as Fig. (3) but for force calculation time.

MDGRAPE-2. Experimental results show that our implementation is scalable with number of particles in the systems for various accuracy. At present, FMM on GRAPE is still in development and its performance is not better than that of treecode for same accuracy. The FMM on GRAPE is slower than treecode on GRAPE roughly 10%-15%. The reason is that Anderson's force formula cannot be performed on GRAPE. This will increase calculation amount on the host computer leading to unbalance of calculation cost between the host computer and GRAPE. However we still have room for improvements and expect to obtain better performance in near future. The parallel implementation of FMM on GRAPE will be implemented soon.

REFERENCES

[1] C. R. Anderson, "An implementation of the fast multipole method without multipoles", Siam J. Sci. Stat. Comput., Vol. 13, No. 4, 1992, pp. 923-947.

[2] A. Appel, An efficient program for many-body simulation", SIAM J. Sci. Stat. Comput., 6(1), 1985, pp. 85-103.

[3] J. E. Barnes, "A modified tree code: Don't laugh; It runs", Journal of Computational Physics 87, 1990, pp. 161-170.

[4] J. E. Barnes, P. Hut, "A hierarchical $O(NlogN)$ force-calculation algorithm", Nature 324, (1986), pp. 446-449.

[5] L.Greengard, V. Rokhlin, "A fast algorithm for particle simulations", J. Comput. Phys., 73 (1987), pp. 325-348.

[6] R. H. Hardin, N. J. A. Sloane, "McLaren's improve snub cube and other new spherical design in three dimensions", Discrete and Computational Geometry, 15 (1996), pp. 429-441.

[7] Y .Hu, S. L. Johnsson, "A data-parallel implementation of $O(N)$ hierarchical N-body methods", Intl. J. of Supercomput. Appl. and High Perf. Comput., 10(1), 1996, pp. 3-40.

[8] A. Kawai, J. Makino, "Pseudo-particle multipole method: A simple method to implement a high-accuracy tree code", The Astrophysical Journal, 550, 2001, pp. L143-L146.

[9] J.Makino, "Yet another fast multipole method without multipoles - pseudo-particle multipole method", J. Comp. Phys., 151, 1990, pp. 910.

[10] J. Makino, "Treecode with a special-purpose processor", Publ. Astron. Soc. Japan 43, 1991, pp. 621-638.

[11] J. Makino, and M. Taiji, "Scientific Simulations with Special-Purpose Computers — The GRAPE Systems", Chichester: John Wiley and Sons, 1998.

[12] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, M. Umemura, "A special-purpose computer for gravitational many-body problems", Nature, 345, 1990, pp. 33-35.