

高速多重極法と専用計算機の併用による分子動力学計算高速化のための二つのアルゴリズム

網崎 孝志^{a*}, 豊田 新次郎^b, 宮川 博夫^c, 北村 一泰^c

^a 鳥取大学医学部保健学科, 〒 683-8503 鳥取県米子市西町 86

^b 富士ゼロックス株式会社 NBC, 〒 259-0157 神奈川県足柄上郡中井町境 430

^c 大正製薬株式会社, 〒 330-8530 埼玉県さいたま市吉野町 1-403

*e-mail: amisaki@grape.med.tottori-u.ac.jp

(Received: April 8, 2002; Accepted for publication: May 23, 2002; Published on Web: July 5, 2002)

タンパク質などの生体分子系の分子動力学シミュレーションにおいては、静電相互作用などの遠距離相互作用の求値に莫大な計算時間が費される。我々は、最近、この計算を高速化するための専用計算機 MD-Engine II を開発した。この計算機は、従来のものとは異なり、二体相互作用計算のための高速アルゴリズムとの併用を前提に設計されている。そのような併用を効率よく行なうためには、専用計算機側のハードウェアのみならず、それを制御するためのホスト計算機上のソフトウェアにおいても特別な機構が必要である。本論文では、そのような目的で開発した二つのアルゴリズムについて報告する。ひとつは、セルリスト管理法についてのものであり、効率のよい併用を実現するための基礎となるアルゴリズムである。そこでは、ホスト-専用計算機間の通信を低減するために、三次元巡回パツファを利用した。もうひとつは、専用計算機上のプロセッサへのタスク割り当てに関するもので、再帰二分法に基づく静的負荷分散アルゴリズムを開発した。本論文では、これらのアルゴリズムを紹介し、また、その有効性を検証するために行なった数値実験の結果を報告する。

キーワード：分子動力学法、専用計算機、高速多重極法、並列アルゴリズム、負荷分散

1 はじめに

分子動力学 (Molecular Dynamics; MD) 計算は、様々な物質の性質を解明する上で欠くことのできない手法となっている。しかしながら、タンパク質をはじめとする生体分子系のシミュレーションにおいては、静電相互作用などの遠距離相互作用の求値に莫大な計算時間が必要となり、このことが MD 計算の実用化の障害となっている。従来は、一定の切断距離を導入し、それより遠方の相互作用を無視するというカットオフ法が用いられていたが、カットオフにより生体分子系のシミュレーションの信頼性が失われることは広く認識されるようになってきた [1–3]。この問題を解決するため、静電相互作用求値に対する様々な高速アルゴリズムが開発されてきた [4–11]。一般に、遠距離相互作用

は二体相互作用の和として定義されるため、 N 個の粒子で構成される系に対する漸近計算量は $O(N^2)$ となる。高速アルゴリズムは、階層的セル構造の上での多重極展開に分割統治法を適用することにより、あるいは、空間メッシュ上での高速 Fourier 変換により、漸近計算量を $O(N \log N)$ 以下に低減する。特に、高速多重極法 (Fast Multipole Method; FMM) の計算量は $O(N)$ といわれている [5, 6]。しかしながら、これらの高速アルゴリズムは、いずれも一種の近似法であり、精密な求値を行なうには依然として多大な計算時間が必要となる。特に、島田ら [8] により報告されているように、精密な計算を行なうためには、近似法を適用しない部分、すなわち、近傍領域のサイズを増大させることが有効であるが、この近傍領域内での相互作用の直接計算に莫大な時間が必要となる。

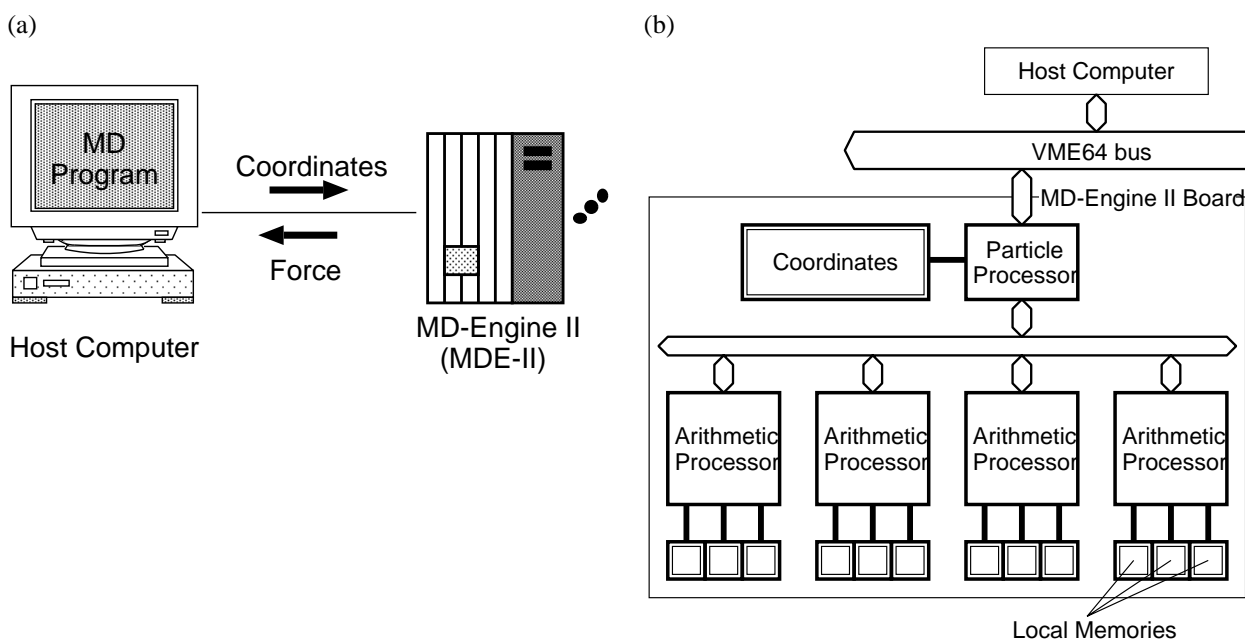


Figure 1. Configuration of MD-Engine II (MDE-II). (a) MDE-II can be viewed as an accelerator that calculates nonbonded pair interactions at high speed. On receipt of the coordinates and charges from the host computer, the MDE-II system calculates interactions (e.g., energy and force) using its dedicated hardware. (b) MDE-II consists of identical computation boards. On each board, a particle-processor and four arithmetic processors are mounted. The board is connected to a host computer via a VME64 bus. VME64 (Versa Module Europa) is a popular open-ended bus system and typically exhibits a throughput of 40 Mbytes/sec.

一方、特別なハードウェアにより MD 計算を高速化する試みもなされている。そのようなハードウェアには MD-Engine[12] や MD-GRAPe[13] などがある。これらの専用計算機は、二体相互作用の求値と総和のための演算パイプラインを備えた専用プロセッサを装備することにより、また、それらのプロセッサの多重化による並列処理により、高速化を実現している。しかしながら、その演算機構は、素朴な直接法に基づくものであり、漸近計算量は依然として $O(N^2)$ のままである。したがって、巨大分子系のシミュレーションに対応するためには、極めて多数のプロセッサで構成される大規模計算システムを構築する必要があり、コスト面で難点がある。

我々は、最近、高速アルゴリズムと組み合わせて利用することができる専用計算機 MD-Engine II (MDE-II) を開発した。その基本コンセプトは、高速アルゴリズムにより漸近計算量を低減した上で、精密計算を行なう場合の難点であった近傍領域の直接計算を専用計算機に担当させるというものである。従来の専用計算機的设计においては、このような作業分担が前提となっ

ていたため、仮に、高速アルゴリズムとの併用を行なっても、多重プロセッサによる並列化において、極めて深刻な効率低下が発生する。これに対し、MDE-IIには特別な機構を実装したため、このような並列効率低下を最低限に抑えることができると予想している。これにより、巨大分子系のための計算システムを、個々の研究室単位に低コストで構築することが可能になると考えている。

高速アルゴリズムとの併用を効率的に行なうためには、ハードウェア/ソフトウェアの両面で一体化したアルゴリズム開発が必要であった。併用時には、MDE-IIは高速アルゴリズムにおける近傍領域セルに存在する粒子に関する計算を担当する。このため、どの粒子が各粒子の近傍領域に存在しているかという情報を管理する必要がある。この情報は、MDE-II とそれを制御するホスト計算機の間で共有する必要があるため、それを管理するアルゴリズムも両方で整合性のとれたものではない。本研究では、粒子リストの管理に、従来より用いられているセルリストを採用し、また、セルリストをホストから MDE-II に効率よく転送する

ために三次元巡回バッファに基づくアルゴリズムを開発した。また、MDE-II は、一種の単一命令多重データ (Single-Instruction-Multiple-Data; SIMD) 型並列計算機とみなすことができる。本研究では、セルリスト管理法を前提とした上で、MDE-II の各演算プロセッサに、効率よく計算を割り当てるための負荷分散アルゴリズムを開発した。これは、再帰二分法に基づくものである。本論文では、これらのアルゴリズムを紹介し、また、その有効性を検証するために行なった数値実験の結果を報告する。

2 MDE-II の概要

MDE-II は、二体相互作用の和として定義される非結合性の相互作用 (エネルギーや力) を計算するための専用の計算機である。MDE-II は、一種のアクセラレータとみなすことができ、ホスト計算機から粒子座標や電荷を受信した後、計算開始の信号を受け取ると、相互作用の値を計算し、それをホスト計算機に返信する [Figure 1(a)]。MD シミュレーションを遂行するためには、それ以外にも、結合性相互作用計算、運動方程式の数値積分、圧力制御など様々な作業が必要になるが、これらの計算量はわずかなため、ホスト計算機上で実行する。他の専用計算機と同様、このように機能を限定することで、高速な計算能力が低コストで実現されている。

MDE-II は、複数枚の計算ボード [Figure 1(b)] で構成された並列計算機である。各ボード上には、各種のメモリの他、粒子プロセッサと 4 個の演算プロセッサが装備されている。粒子プロセッサは、座標メモリから座標や電荷を取り出して、それらを 4 個の演算プロセッサに同時に配信する。それをもとに、各演算プロセッサは、その演算パイプラインを用いて二体相互作用を高速に計算する。MDE-II のアーキテクチャの詳細については別途報告するが、ここでは、本論文の以後の議論において特に重要となることを列挙しておく。(i) 二体相互作用の演算時間は一定であり、変化しない：ボード 1 枚あたり、8 個の粒子に働く 2 種の二体相互作用を必ず 160 ns で計算する。この性質は、メモリ参照を演算パイプラインの各ステージに分散して割り当てることにより実現されたものであり、メモリキャッシュの性能に演算時間が左右されがちな通常の計算機に比べ MDE-II が優れている点のひとつである。(ii) 計算ボードはそれぞれ独立に動作するが、各ボード上の演

算プロセッサは同期して動作する。したがって、各ボードは粒子 8 個を単位として計算を行なう。このように MDE-II は、計算ボードとしてみると SIMD 型計算機であり、システム全体としてみると多重命令多重データ (Multiple-Instruction-Multiple-Data; MIMD) 型計算機である。(iii) 粒子プロセッサは、近傍領域粒子抽出機構を備えている。近傍領域に存在する複数の粒子の座標は、メモリ上では必然的に分断されて配置されるが、粒子プロセッサは、これらを途切れることなく連続して取り出してくる。

3 近傍粒子管理のためのアルゴリズム

本章では、まず、高速アルゴリズムとして FMM (高速多重極法) を取り上げ、それを例として、セルや遠方/近傍領域とは何かについて説明する。続いて、セルリストを利用した近傍粒子リスト管理法を説明し、最後に、それをホスト計算機と MDE-II の上に実現するために開発した三次元巡回バッファ・アルゴリズムについて報告する。本論文の主題は高速アルゴリズムと MDE-II の「併用」であるが、前述のように、遠方領域の計算は高速アルゴリズムが、近傍領域の計算は MDE-II が担当する。ここに報告するアルゴリズムと前述の近傍粒子抽出機構は、「併用」において中核をなすものである。

3.1 FMM の概要

一般に、高速アルゴリズムにおいては、空間を遠方と近傍の二つに分割し (Figure 2)、高速化技法は遠方領域にのみ適用される。例えば、FMM の場合は、遠方領域に存在する全ての粒子との相互作用を 1 個の多項式で近似する。この多項式は、局所展開と呼ばれ、各セルの多重極/局所展開の分割統治的変換により生成される。一方、近傍領域に存在する粒子との相互作用は、通常の方法により計算する。FMM の詳細については、文献 [5, 6, 14] などを参照されたい。

FMM には種々の変法があるが、本論文の主題に関係するのは近傍領域の取り方である。ある粒子に働く二体力の総和を計算するとき、この粒子を中心粒子という。また、中心粒子が存在するセルを中心セルという。中心セルとそれを取り囲む 26 個のセルの合計 $3 \times 3 \times 3 = 27$ 個のセルで近傍領域を構成するとき、それを I

型近傍とよぶことにする。さらに、それらの周りのセルも含めて、合計 $5 \times 5 \times 5 = 125$ 個のセルで近傍領域を構成するとき、それを II 型近傍とよぶ。元来の FMM は II 型近傍を前提に発表されたが、計算量の少ない I 型近傍が使われることもある。多重極展開に基づく他の高速アルゴリズムでも、これら二種の近傍領域が用いられることが多い。MDE-II の粒子プロセッサは、基本的には、これら二種の近傍領域に対応している。

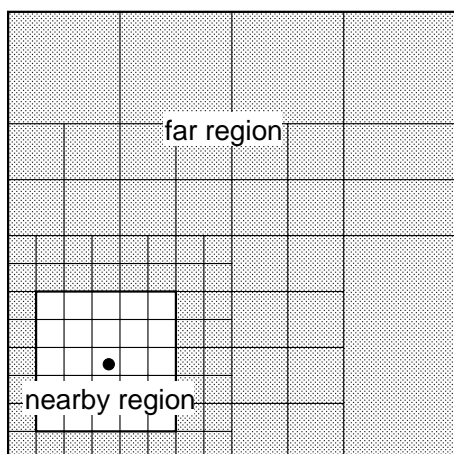


Figure 2. A hierarchical cell structure of the fast multiple method (FMM) in two dimensions. The far and nearby regions (gray and white boxes, respectively) are defined for each atom (closed circle). The contributions of atoms in the far region are evaluated using the multipole approximation method, while those of nearby atoms are directly evaluated using the naive $O(N^2)$ method.

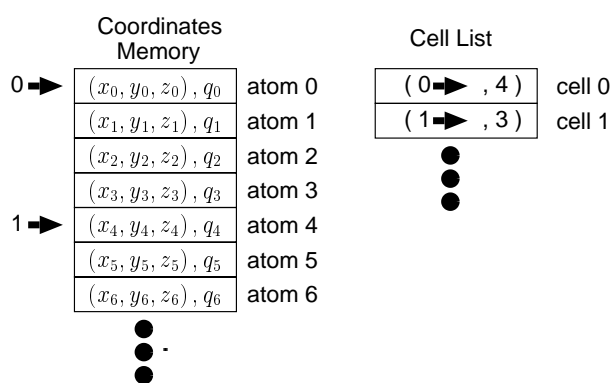


Figure 3. A cell-list method. The cell-list can be expressed using a list of pairs, each of which consists of a pointer to the first atom in that cell and number of atoms contained in that cell.

3.2 粒子リストとセルリスト

元来、セルリストは、カットオフ法において、各粒子の近傍に存在する粒子を選び出すために利用されていた。同じものは、高速アルゴリズムでの近傍領域粒子を抽出するためにも利用できる。

セルリストは、各セルに含まれる粒子のリストを並べたものである。通常のコンピュータでは、粒子リストを表現するために結合リストというデータ構造が用いられることが多い。しかし、結合リストには間接アドレッシングが必要であり、そのような高価なメモリ操作を導入すると、MDE-II の特長である「一定時間での演算の完了」が達成できなくなる。代わりに、本研究では、配列を用いてセルリストを実現した (Figure 3)。すなわち、メモリ上で、同一セルに含まれる粒子が連続するように、粒子座標と電荷を並び変えておけば、あるセルに含まれる粒子のリストは、その先頭粒子へのポインタとそのセルに含まれる粒子の数という対で表すことができる。そのような対を並べたものがセルリストであり、I 型近傍は 27 個の対で、II 型近傍は 125 個の対で表現できる。

注意すべきは、近傍領域に含まれる粒子は、メモリ上で必然的に分断されて配置されることである。コンピュータのメモリは線型であるため、その上にどのような順序で並べようとも、近傍領域を構成する全てのセルの粒子を、メモリ上で連続して配置することはできない。MDE-II の粒子プロセッサには、そのようにメモリ上で分断された粒子を、途切れることなく連続して取り出してくる機構がある。この近傍粒子抽出機構は、ベクトル計算機での gather 操作に対応するものであるが、この機構のため、高速アルゴリズムとの効率のよい併用が可能となる。

3.3 三次元巡回バッファ

高速アルゴリズムとの併用において、MDE-II が近傍領域の計算を担当するとき、その計算は、基本的に、セルを単位として行う。すなわち、ひとつのセルに存在する全ての粒子について、それらに作用する力を求め終わった後、次のセルの計算に移る。このとき、中心となるセルが変われば、近傍領域も変化するため、新たな近傍領域セルリストを MDE-II に転送する必要がある。このとき、27 個あるいは 125 個のセルのデータを全て転送するのは無駄なので、ホスト計算機は、新たに近傍領域となったセルのデータだけを送信し、MDE-II は

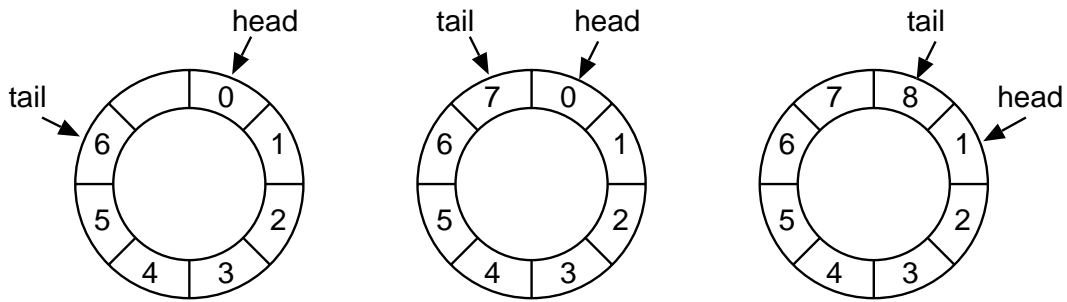


Figure 4. A cyclic buffer in one-dimensional case.

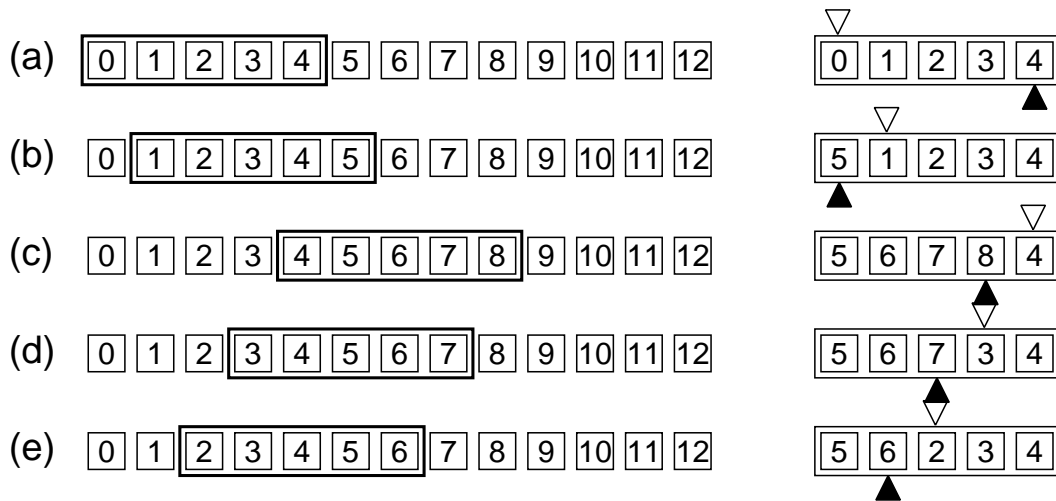


Figure 5. A cyclic buffer represented as a window moving around data. The open and solid triangles represent the positions of head and tail pointers, respectively.

それを既に近傍領域ではなくなったセルのデータに重ね書きする。本研究では、これを実現するために、三次元巡回バッファに基づくアルゴリズムを考案した。これを説明するために、まず、一次元の巡回バッファについて解説する。

Figure 4 (左) に、一次元の巡回バッファを示す。データを格納するための M 個 (この例では $M = 8$) の要素からなる配列と、先頭と末尾の位置を示すポインタで構成される。もともと、巡回バッファは、先入れ先出しの待ち行列などを実現するためのものであり、その場合には、データの最大個数が M を超えないことが保証されていなければならない。もし、 M 個のデータをバッファに投入した状態 (図中央) で、さらにデータを末尾に追加すると、そのデータは以前の先頭のデータの位置に重ね書きされてしまう (図右)。逆に、常にこのような重ね書きを行なうものとする、一次元巡回バッファは、データの上を移動していくウィンドウとみなすことができる [Figure 5 の (a) (b) (c) の遷移]。さらに、逆方向への移動 [Figure 5 の (c) (d) (e) の遷移] も許可したものが、双方向の一次元巡回バッファで

ある。MDE-II で利用するのは、これを三次元に拡張したもので、 x, y, z の各軸に対して、それぞれ先頭と末尾の位置を示すポインタを持っている。

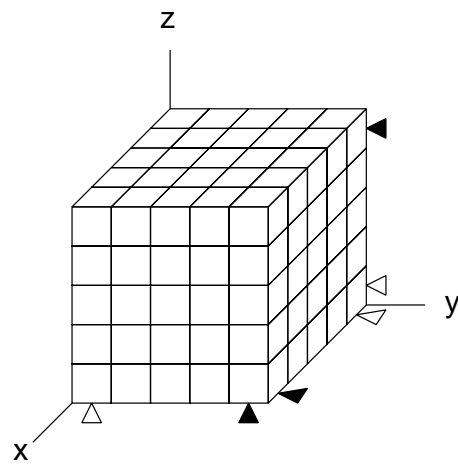


Figure 6. A cyclic buffer in three-dimensional case. The open and closed triangles for each axis indicate the positions of head and tail pointers, respectively.

Figure 6 に II 型近傍の三次元巡回バッファの初期状態を示した。以上のアルゴリズムは粒子プロセッサに実装されているが、ホスト計算機上でも MDE-II と協調して動作するため、同じアルゴリズムによりセルの管理を行なう。

4 負荷分散法

前述のように、MDE-II の計算ボードは SIMD 方式で動作し、8 個の粒子に働く力を同時に計算する。このため、ひとつの中心セルに含まれる粒子に働く力の計算を、複数のボードに分割して担当させるよりも、単独のボードに担当させる方が有利である。また、近傍領域セルリストの転送量の観点からも、ひとつの中心セルに関する計算は、単独のボードに担当させる方が望ましい。以後、このようなセル単位の並列化における負荷分散を考える。

前述の近傍セルデータの転送量を最小にするためには、中心セルの移動は、セルの境界面を共有する方向に行なう必要がある。この条件を満足するような空間順序でセルを番号付けしておく [15]。また、同じ理由から、各ボードは、番号が連続するセルを担当することが望ましい。そこで、以下のような方針でセルをボードに割り当てる。

1 枚の計算ボードにより、中心セル c に含まれる全ての粒子に関する相互作用の値を求めるための時間は、 $\lceil N_c / 8 \rceil \times N_n \times 120$ で与えられる。ここで、 N_c はセル c に存在する粒子数、 N_n はセル c の近傍領域に存在する粒子の数である。また、 $\lceil x \rceil$ は x を下回らないような最小の整数を表す。MDE-II の演算パイプラインはストールしないように設計されているので、この時間は正確に予測することができ変動することはない。そこで、セル c に関する負荷量 l_c を

$$l_c = \left\lceil \frac{N_c}{8} \right\rceil \times N_n$$

で定義する。すると、ここでの負荷分散問題は以下のように形式的に表現することができる。

C 個の整数よりなる列 $L = \{l_0, l_1, \dots, l_{C-1}\}$ を連続する B 個の部分列に分割する。ただし、それぞれの部分列に含まれる整数の和を S_0, S_1, \dots, S_{B-1} とするとき、任意の S_i と S_j との差の最大値が最小となるような分割法とする。

ここで、 C は総セル数、 B はボード枚数である。この問題の厳密解を求めることは現実的ではないので、本研究では、以下のように再帰二分法 [16](pp.52-53) を変形したものを利用した。まず、 L の接頭部和を要素とする列 A を求める。すなわち、 A の第 k 要素 a_k について

$$a_k = \sum_{i=0}^k l_i$$

となるように A の各要素を定める。この列において、 $a_{C-1} \times \lceil B/2 \rceil / B$ に最も近い要素を二分探索により検出し、その位置で、 L を二つに分割する。これにより、 L は、負荷量の比がほぼ $\lceil B/2 \rceil / B$ と $\lfloor B/2 \rfloor / B$ となるような二つの列に分割される。この部分列のそれぞれを、 $\lceil B/2 \rceil / B$ 枚と $\lfloor B/2 \rfloor / B$ 枚のボードに担当させるわけである。なお、 $\lfloor x \rfloor$ は x を越えないような最大の整数を表す。以上の手続きを、得られた部分列それぞれに再帰的に適用することを繰り返し、最終的に、負荷量のほぼ均一な B 個の部分列を得る。

5 数値テスト

これまでに述べた近傍粒子抽出機構と計算ボード間負荷分散アルゴリズムの有効性の検証を行なうため、近傍領域の相互作用計算時間に関する数値実験を行なった。対象とした系は、 10^5 個の粒子を一辺の長さ 100 の立方体にランダムに配置したものである。この場合の数密度は 0.1 であり、長さの単位を Å にとれば、タンパク質-水系での典型的な値である。この系において、FMM の階層セル構造における分割レベルを $L=3$ あるいは $L=4$ にとった。すなわち、系を、 8^3 個あるいは 8^4 個のセルに分割した。それぞれについて、ポテンシャルエネルギーへの近傍領域からの寄与分を計算するための時間を計測した。近傍領域は I 型と II 型の 2 種類を用いた。ホスト計算機には Sun ワークステーション (UltraSPARC, 167MHz) を使用し、MDE-II システムと Sbus-VME バス変換器を介して接続した。粒子プロセッサの近傍領域粒子抽出機構が利用できる場合とできない場合について、計算ボードの枚数を、1, 2, 3 枚と変化させて時間を計測した。ただし、MDE-II には上記機構を利用せずに近傍領域だけの計算を行なう機能はないので、以下の方法で推定した。すなわち、MDE-II に転送する近傍領域セルのデータ ($3 \times 3 \times 3 = 27$ 個) のうち、ひとつだけを実際のセルのデータとし、残りの 26 個は、相互作用をもたない粒子 1 個だけを含むダミーのセルのものとした。ここで、相互作用をもたない

粒子とは、電荷ならびに van der Waals のエネルギーパラメータが 0 であるような架空の粒子である。

Table 1. Performance of MDE-II with or without a special hardware mechanism for selecting atoms in the nearby regions in the computation for a system of 10^6 random atoms.

B	Type	L	with	without
Execution Time (s)				
1	I	3	8.8	10.6
1	I	4	2.0	4.2
1	II	3	31.3	38.3
1	II	4	6.5	16.2
Parallel Efficiency (%)				
2	I	3	99.4	90.2
2	I	4	95.1	70.4
2	II	3	99.9	90.5
2	II	4	98.3	72.8
3	I	3	99.0	83.2
3	I	4	87.1	52.7
3	II	3	99.7	84.8
3	II	4	96.7	54.7

* B and L are the number of boards and recursive level of the FMM cell, respectively. “Type” represents the type of nearby regions.

これにより、近傍領域の中の 1 個の実セルについての相互作用を計算した。この実セルを残りの実セルに順に置き換えて計算することを繰り返し、近傍領域からの全相互作用を計算した。その後、ダミーセルの計算に要した時間を補正し、粒子プロセッサに近傍粒子抽出機構がない場合の計算時間を推定した。

結果を Table 1 に示す。この表では、ボード枚数 (B) が 1 枚のときには近傍領域のエネルギー求値に要した時間 (s) を、 $B \geq 2$ の場合には並列効率 (%) を示した。また、“Type” の欄のローマ数字 I と II は、それぞれ近傍領域の I 型と II 型を表している。また、 L の欄の値は、全空間を 8^L 個のセルに分割したことを示している。

並列効率に注目すると、近傍粒子抽出機構が利用できる場合は、一部の例外を除き、極めて高い値を示している。このことは、本研究で開発した負荷分散アルゴ

リズムが極めて有効に機能していることを示唆している。これに対し、近傍粒子抽出機構が利用できない場合の効率は概して低く、本機構の有効性を示唆している。特に、II 型近傍の $L=4$ の場合の差が顕著である。

近傍領域の種別 (I/II) 並びに L の値は、ともに、中心セルの粒子数 (N_c) と近傍領域中の粒子数 (N_n) の両方に同じように影響を与える。近傍領域の種別が同一であれば、 L の値が小さいほど N_c, N_n ともに大きくなる。 L の値が同一であれば、II 型近傍の場合の方が I 型近傍の場合よりも N_c, N_n ともに大きい。このうち、並列効率に大きく影響を与えるのは、近傍領域のサイズ N_n であると思われる。実際、 $L=4$ で I 型近傍の場合の効率は低い。これは、ホスト計算機から MDE-II への各種の通信時間に比べて、粒子 1 個あたりの計算量が小さい、すなわち、 N_n が少ないためと思われる。特に、近傍粒子抽出機構が利用できない場合は、近傍領域の各セル毎にセルデータを送信する必要があるため、並列効率が 52.7% にまで低下している。一方、II 型近傍で $L=4$ の場合は、I 型近傍に比べて N_n が大きいため、近傍粒子抽出機構が利用できる場合は効率の低下はあまりみられない。しかしながら、II 型近傍のセル数 (125) は I 型近傍のもの (27) に比べて多いため、近傍粒子抽出機構が利用できない場合は、先ほどと同様の原因により効率が大きく低下しているものと思われる。

一方、ボード数 1 枚の場合の実行時間をみてみると、ここでも、近傍領域のサイズが小さい場合に、近傍粒子抽出機構の有無による違いが大きい。

以上の結果は、本研究の近傍粒子抽出機構並びに負荷分散アルゴリズムの有効性を示しているものと思われる。

なお、ここでは、ボード単位の並列効率を議論したが、中心セルの粒子数 N_c は、ボード内での並列効率 (プロセッサ利用率) にも影響を与える。すなわち、前述のように、各計算ボードは同時に 8 個の粒子についての相互作用を計算する。このため、例えば、 $N_c=24$ であれば 3 回の計算ですむが、 $N_c=25$ であれば 4 回の計算が必要になる。前者での利用率は 100% であるが、後者では 82% 程度となる。実際、 $L=4$ の場合、各セルには平均で 24.4 個の粒子が存在することになり、最悪の場合、演算プロセッサの利用率は 82% となる。この数値は、極端な場合の例であり、実際には、利用率がそこまで低下することはないと思われる。一方、 $L=3$ の場合は、各セルに平均で 195.3 個の粒子が存在する。この場合に、同様に最悪時の利用率を計算すると 97% 程度となる。

6 おわりに

本論文では、分子動力学計算のための高速アルゴリズム FMM と専用計算機 MDE-II の効率のよい併用のために必要となる二つのアルゴリズムについて報告した。ひとつは、MDE-II の粒子プロセッサに近傍粒子抽出機構の一部として実装したセルリスト管理アルゴリズムであり、この機構を利用するためには、ホスト計算機上のソフトウェアにも同アルゴリズムを実装する必要がある。もうひとつは、計算ボード間の負荷分散のためのアルゴリズムである。また、数値実験により、近傍粒子抽出機構と負荷分散アルゴリズムが有効に機能していることを検証した。

上述のように、セルリスト管理アルゴリズムは近傍粒子抽出機構の一部として一体化しており、また、近傍粒子抽出機構と負荷分散アルゴリズムも、分割点となるセルの取り方やセルリストデータ転送のための通信量などにおいて互いに関連している。このため、本論文の数値実験では、個々のアルゴリズムや機構を分離して評価せず、全体としての評価を行なった。しかしながら、今後、より効率のよい計算システムの構築を行なうためには、より詳細な評価実験も必要になると思われる。

現在、我々は、MDE-II 計算ボードを組み込んだ複数の計算ノードで構成された並列計算環境上で、FMM などの高速アルゴリズムの並列処理を行なうことを計画している。そのようなシステムを構築するには、ハードウェア/ソフトウェア両面で様々な課題がある。しかしながら、本論文で報告した二つのアルゴリズムは、そのようなシステムにおいても、「併用」の中核ソフトウェアとして有効に機能するものと考えている。

なお、これらのアルゴリズムを実装したプログラム、並びに、それらと協調して動作する FMM のプログラム (論文 [11] の周期境界条件に対する機能を実装) の配布を希望される方は、著者のうちいずれかに連絡をいただきたい。

本研究の一部は、文部科学省科学研究費補助金 (C)13224070 並びに (C)13680743 により行なわれた。

参考文献

[1] R. J. Loncharich and B. R. Brooks, *PROTEINS*, **6**, 32-45 (1989).

- [2] H. Schreiber and O. Steinhauser, *Biochem*, **31**, 5856-5860 (1992).
- [3] M. Saito, *J. Chem. Phys.*, **101**, 4055-4061 (1994).
- [4] J. Barnes and P. Hut, *Nature*, **324**, 446-449 (1986).
- [5] L. Greengard and V. Rokhlin, *J. Comput. Phys.*, **73**, 325-348 (1987).
- [6] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge (1988).
- [7] M. Saito, *Mol. Simul.*, **8**, 321-333 (1992).
- [8] J. Shimada, H. Kaneko, and T. Takada, *J. Comput. Chem.*, **15**, 28-43 (1994).
- [9] J. W. Eastwood and R. W. Hockney, *J. Comput. Phys.*, **16**, 342-359 (1974).
- [10] T. Darden, D. York, and L. Pedersen, *J. Chem. Phys.*, **98**, 10089-10092 (1993).
- [11] T. Amisaki, *J. Comput. Chem.*, **21**, 1075-1087 (2000).
- [12] S. Toyoda, H. Miyagawa, K. Kitamura, T. Amisaki, E. Hashimoto, H. Ikeda, A. Kusumi, and N. Miyakawa, *J. Comput. Chem.*, **20**, 185-199 (1999).
- [13] T. Narumi, R. Susukita, T. Ebisuzaki, G. McNiven, B. Elmgreen, *Mol. Simul.*, **21**, 401-415 (1998).
- [14] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Academic Press, San Diego, CA (1996).
- [15] 空間順序とは、三次元空間を構成するセルを走査する順序をいう。いくつかの種類があるが、行順序、交替行順序、Morton 順序、Peano-Hilbert 順序などが代表的である。FMM の実現においては、Morton 順序が使われることが多い。本研究では、本文中に述べたように、セルの境界面を共有する方向に走査を行なう必要があるが、この条件を満足するのは、交替行順序と Peano-Hilbert 順序である。
- [16] I. Foster, *Designing and building parallel programs: concepts and tools for parallel software engineering.*, Addison-Wesley: Reading (1994).

Two Algorithms Designed for Realizing Efficient Combination of Fast Multipole Method and Dedicated Hardware for Molecular Dynamics Simulations

Takashi AMISAKI^{a*}, Shinjiro TOYODA^b, Hiroh MIYAGAWA^c and Kunihiro KITAMURA^c

^aFaculty of Medicine, Tottori University, 86 Nishi-machi, Yonago 683-8503, Japan

^bNBC Fuji Xerox Co., Ltd., 430 Sakai, Nakai-machi, Ashigarakami-gun 259-0157, Japan

^cTaisho Pharmaceutical Co., Ltd., 1-403 Yoshino-cho, Saitama 330-8530, Japan

**e-mail: amisaki@grape.med.tottori-u.ac.jp*

This paper concerns two algorithms designed for acceleration of molecular dynamics simulations using the fast multipole method (FMM) and special-purpose machine (MD-Engine II) for calculating pair interactions. One algorithm is based on a three-dimensional cyclic buffer, which enables efficient cooperation between FMM and MD-Engine II. The algorithm is implemented as part of the particle-processor of MD-Engine II. The processor specifically extracts a set of particles which are contained in nearby regions defined in FMM. The other algorithm is based on a recursive bisection for static load-balancing technique. The efficacy of these algorithms was confirmed by numerical tests.

Keywords: molecular dynamics, special-purpose machine, fast multipole method, parallel algorithm, load-balancing

