

ベクトル計算機における固体 MD 計算の高速化

板 倉 憲 一[†] 横 川 三 津 夫[†] 清 水 大 志[†]
君 塚 肇[†] 蕪 木 英 雄[†]

地球シミュレータは、640 の計算ノードを持ち理論ピーク性能は 40 Tflop/s である。プロセッサノードはピーク性能 8 Gflop/s のベクトルプロセッサ 8 個、16GB の共有メモリから構成される。本研究では地球シミュレータの計算ノードによる固体分子動力学法の計算プログラムのベクトル化と並列化を行い性能評価を行った。

分子動力学法では、カットオフ半径内の粒子が互いに影響を与え、その粒子ペアを行列を用いて表現することができる。ベクトル化に際して、この行列表現に compressed row form と jagged diagonal form を考える。jagged diagonal form はベクトル長が compressed row form よりも長くできるので、ベクトル化により適している。しかし、標準的な粒子対の情報から jagged diagonal form に変換するには時間がかかるため、全体の性能はより簡単な compressed row form よりも低下した。compressed row form では 8CPU での並列化により 2.4 から 2.7 倍のスピードアップとなった。

High Speed Calculation for Solid Molecular Dynamics on Vector Processors

KEN'ICHI ITAKURA,[†] MITSUO YOKOKAWA,[†] DAISHI SHIMIZU,[†]
HAJIME KIMIZUKA[†] and HIDEO KABURAKI[†]

The Earth Simulator which is under development has 640 processor nodes and its peak performance is 40 Tflop/s. Each node has 8 vector processors, each of which has 8 Gflop/s peak performance, and 16GByte shared main memory. In this study, we have evaluated performance of solid molecular dynamics simulation on an SMP node of the Earth Simulator.

In molecular dynamics simulation, each particle is influenced by all particles within a cut-off region and the representation of these pairs of particles is made by a matrix. Two matrix representations, compressed row form and jagged diagonal form, are considered for vectorization. The jagged diagonal form is better than the compressed row form in performance on a vector processor for the force calculation of every pairs, because the vector length of the former is longer than that of the latter. However, computational cost for converting the normal matrix form to the jagged diagonal form is quite expensive and the total performance in using the jagged diagonal form is low. with the jagged diagonal form is obtained. Speedup by parallelization with the compressed row form is 2.4 to 2.7 with 8 vector processors.

1. はじめに

地球シミュレータ¹⁾²⁾ は、気候・気象分野及び固体地球のシミュレーションを始めとした様々な大規模問題に挑戦するための超高速並列計算機である。この地球シミュレータは 2002 年 3 月の運用開始時点で世界最高速の計算機になるものと予想され、地球変動予測研究に大きな進展をもたらすことが期待されている。本体製作及び海洋科学技術センター横浜研究所(横浜市金沢区)に建設された地球シミュレータ棟への据付調整作業が進められている。

地球シミュレータはその主なターゲットアプリケー

ションである気候・気象分野のシミュレーションにおいて、実効性能として 5 Tflop/s が目標であるが、さまざまアプリケーションによる地球シミュレータの性能評価をすることが重要である。実際にこのようなアプリケーションプログラムの成果を得ると共に、計算機工学の分野においても、大規模なベクトル型並列計算機の利用方法を促進させるアルゴリズム解析・開発することが重要である。

本研究では、地球シミュレータの計算ノードを用いて分子動力学法の高速化を行う。分子動力学法は様々な分野で利用されており、基本的な計算アルゴリズムであるがその原理から大規模な粒子数での計算には非常に多くの計算パワーが必要とされる。ここでは、予備評価として 1 ノードにおけるベクトル並列処理と共有メモリ並列処理の高速化を行った結果について報告

[†] 日本原子力研究所
Japan Atomic Energy Research Institute

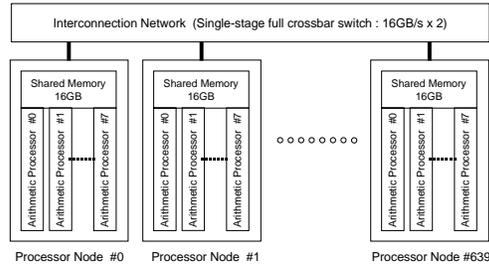


図1 地球シミュレータ概念図

2. 地球シミュレータのアーキテクチャ

地球シミュレータ概念図を図1に示す。地球シミュレータは640台の計算ノード(PN: Processor Node)を単段のクロスバネットワークで結合する分散メモリ型並列計算機である。地球シミュレータで想定されるアプリケーションプログラムでは、全てのプロセッサ間での通信が高速に行われる必要性があり、このような柔軟なネットワークとした。

PNはピーク性能8Gflop/sの計算用ベクトルプロセッサ(AP: Arithmetic Processor)を8台が16GBの主記憶を共有する共有メモリ型並列計算機である。したがって、全体ではAP 5120台、ピーク性能は40Tflop/s、主記憶10TBとなる。主記憶は、32台の主記憶ユニット(MMU: Main Memory Unit)から構成される。PNにはその他リモートアクセス制御装置(RCU: Remote Access Control Unit)及び入出力プロセッサ(IOU: I/O Processor)がある。主要なLSIには0.15 μ m CMOSテクノロジーを、冷却方式には空冷を採用した。

APはベクトル処理部(VU: Vector Unit)、スカラ処理部(Scalar Unit)、プロセッサネットワークユニット(PNU: Processor Network Unit)及びアドレス制御部(ACU: Address Control Unit)からなり、1LSI(20.79mm \times 20.79mm)で実装する。SUは4ウェイのスーパースカラであり128個の汎用レジスタ、2ウェイセットアソシエティブ方式の命令キャッシュとデータキャッシュをそれぞれ64KBづつ実装している。さらに、分岐予測や投機実行機能も持つ。VUは6種類(加減算、乗算、除算、論理、ビット列演算、ロード/ストア)のベクトル演算器と72個のベクトルレジスタからなるベクトル演算器セット8個で構成され、最大8Gflop/sの性能を有している。32個のMMUには主記憶素子としてDRAMベースの128Mbitの高速RAMを採用し、2048バンク構成とした。各々の

APは主記憶システムとの間に32GB/secのバンド幅を持っており、1PNで256GB/secを確保している。

地球シミュレータの基本ソフトウェアは、既存のUnix系オペレーティングシステムをベースにした。

地球シミュレータでのプログラミングでは1) AP内のベクトル化、2) PN内の共有メモリ型並列処理、3) PN間の分散メモリ型並列処理の3階層の並列化を利用する。1)および2)はコンパイラによるベクトル化及び並列化を利用し、ユーザによるディレクティブによってこれを促進させる。3)にはメッセージパッシングライブラリMPI2の利用、及びHPF2の日本拡張及び地球シミュレータ用の独自拡張機能を利用する。

3. 分子動力学法

原子力用材料の研究開発では、原子炉の高経年化の問題に関連して中性子照射を受けた材料が硬化する過程を原子レベルの微視的な立場からその原因を解明することが求められている。この過程では中性子よりはじき飛ばされた結晶中の格子間原子や空孔がそれぞれ集合して複雑なクラスタを形成し、結晶中の転位と相互作用することが考えられるが、実験での観察は困難なことから分子動力学法(MD)による素過程の数値シミュレーションが重要である。一般的に結晶中に存在する欠陥集合体及び欠陥間相互作用等の運動を原子レベルから明らかにするためには、大規模なMDによる数値シミュレーションが現在最も期待される手法となっている。

本研究では、将来のMDによる様々なシミュレーションの最適化のベースとして簡単化した条件での主要計算部分の高速化を行うことを目的としている。このため、扱うモデルは面心立方格子状の様に分布している固体銅原子とした。また、粒子間に働く力をポテンシャル関数としてEmbedded-Atom Method(EAM)³⁾モデルを用いている。系から受けるエネルギー E_{total} は他の粒子から受けるエネルギー E_i の総和として求められる。

$$E_{total} = \sum_i^n E_i$$

EAMでは E_i は以下のように与えられる。

$$E_i = \frac{1}{2} \sum_j \phi(r_{ij}) + F(\bar{\rho}_i)$$

$$\bar{\rho}_i = \sum_j \rho_{ij}$$

ここで、 r_{ij} は*i*番目と*j*番目の粒子間距離、 ϕ は粒子間に働く力の関数、 F は密度 ρ から求められるもう一つの粒子間に働く力である。この ϕ と F の力の関数の影響を与える粒子*j*はカットオフ半径があり、一般的には4つの隣接した結晶格子の距離で与える。こ

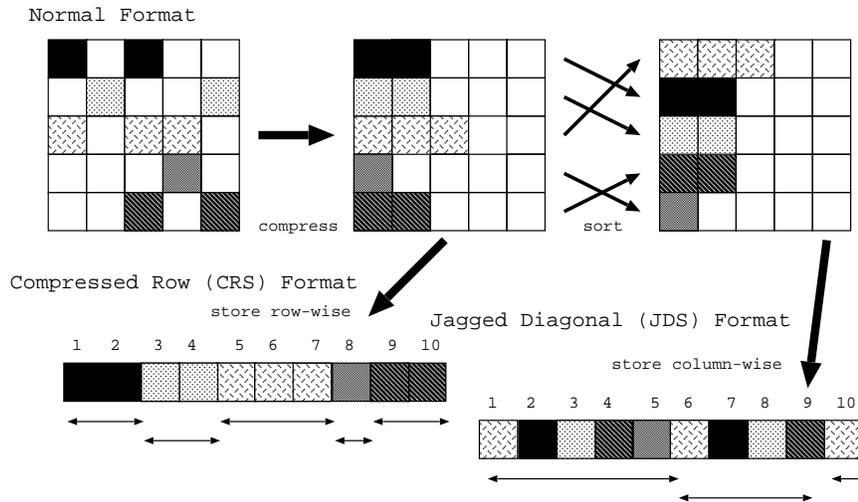


図2 粒子対の格納方法

ここで用いた銅原子間でのシミュレーションでは、カットオフ半径は 4.961 オングストローム とした。

MD では、ポテンシャル関数の働くカットオフ半径よりも広い半径内の粒子を予め各粒子毎に登録しておく、真のポテンシャル関数の働く粒子ペアを検索する時間を短縮するブックキーピング手法を用いるのが一般的である。ブックキーピング法では、あらかじめ粒子間力が影響する可能性のある原子対が求められている状態からカットオフ半径内の真の粒子対を求め、その粒子間力をそれぞれの粒子に作用させていく。本研究においても、ブックキーピング法を用いており、その半径はカットオフ半径の 1.4 倍とし、タイムステップ 1000 の間ブックキーピングの回数は最初の 1 回だけとした。

4. 性能評価

4.1 ベクトル化の手法

真の粒子対の情報の保持の方法は、行と列を粒子番号とする行列で表現すると疎行列になる。多数の粒子に対して MD 計算をするので、この行列の次元は大きい。したがって、圧縮形式を用いて粒子対を表現するのが普通である。ここでは、連立一次方程式の解法などで用いられる疎行列の圧縮方式である compressed row 形式 (CRS) と jagged diagonal 形式 (JDS) を用いた。CRS と JDS の形式を図 2 に示す。

CRS は行方向優先で非零要素の情報を 1 次元配列に入れる。さらに、同じサイズの列番号を入れた 1 次元配列と列の開始位置を記録した配列を用意する。JDS は行方向に圧縮した後、非零要素数でソートし、その結果を列方向優先で 1 次元配列に格納する。行の開始位置を記録する配列と行のソート後の行番号とオリジ

ナルの行番号の対応表が必要になる。それぞれのモデルでの力計算部分のコードを図 3, 4 に示す。

```

do i = 1, nx
  do k = i_str(i), i_str(i+1)-1
    f = (w(i) + w(j)) * u(k) + v(k)
    fx = f * dx(k)
    fy = f * dy(k)
    fz = f * dz(k)
    fxb(i) = fxb(i) - fx
    fyb(i) = fyb(i) - fy
    fzb(i) = fzb(i) - fz
    pb(i) = pb(i) + t(k)
    fxb(j_list(k)) = fxb(j_list(k)) + fx
    fyb(j_list(k)) = fyb(j_list(k)) + fy
    fzb(j_list(k)) = fzb(j_list(k)) + fz
    pb(j_list(k)) = pb(j_list(k)) + t(k)
  end do
enddo

```

図3 CRS による力計算のカーネルループ

疎行列に対する反復計算や MD の力の計算は、1 次元配列に入れた非零要素に対応する演算に用いる。計算のループはこの 1 次元配列をアクセスするが、CRS も JDS も二重ループとなる。ベクトル処理では最内側のループ長がマシンの命令ベクトル長よりも長い方が高い性能が得られる。

元の行列のサイズは粒子数の 2 乗になるが、1 行中の非零要素数はそれに比べると非常に小さく、ここで用いた銅原子の MD の場合は 100 程度になる。このため、CRS では、短ベクトル処理が多くなり地球シ

```

do i = 1, nl
  do j=1, j_str(i+1)-j_str(i)
    k = j+jstr(i)-1
    jj = jj_list(k)
    ii= sorttable(j)
    f = (w(ii)+w(jj)) * u(k) + v2(k)
    fx = f * dx2(k)
    fy = f * dy2(k)
    fz = f * dz2(k)
    fxb(ii) = fxb(ii) - fx
    fyb(ii) = fyb(ii) - fy
    fzb(ii) = fzb(ii) - fz
    pb(ii) = pb(ii) + t(k)
  end do
enddo

```

図4 JDSによる力計算のカーネルループ

ミュレータのベクトル長は 256 なので、この長さでは十分な実効性能が出せない。

これに対して JDS では最内ループの長さは粒子数である。各行の非零要素数がほぼ揃っている状態では、この非常に長いループ処理になる。メモリアクセスも連続となり、力の計算部分の最適化が非常に促進されると考えられる。

MD のような 2 粒子間の力の相互作用を計算する場合には、作用反作用の関係を利用し双方の粒子に働く力を一度に求める方法が一般的である。CRS では最内ループで各非零要素に対する力の計算を行う方向の粒子番号は 1 つであり、列方向の粒子番号は重複が無いのでベクトル化が可能である。しかし、JDS では列方向の粒子番号は重複があるかもしれない間接参照となりベクトル化ができない。このために、CRS では対象な疎行列の上半分だけを利用するが、JDS では全ての行列要素を利用し、力の計算量が 2 倍になるがベクトル化を促進する。

以上のようなベクトル化による高速化手法を行った後、更に、ノード内の共有メモリによる並列化を評価する。共有メモリによる並列化には、コンパイラによる自動並列化機能とそれを支援するコンパイラ指示行によって行った。

4.2 ベクトル化の評価

CRS と JDS のデータ格納形式に基づいて銅原子の相互作用をシミュレーションする MD のプログラムを Fortran90 で記述した。

粒子数として、32000、108000、256000 を用いた。

CRS と JDS の MD のプログラムを単体プロセスで評価した結果を表 1 示す。

粒子数 32000 の場合について、力の計算部分の処理時間とベクトル化効率を評価した結果、CRS では力の計算部分で 44.77% が消費されている。この部分の

表 1 ベクトル化の処理時間 [秒]

粒子数	CRS	JDS
32000	132.47	222.25
108000	483.19	748.22
256000	1299.88	1966.97

ベクトル化率は 97% だが、平均ベクトル長が 89.43 となっている。これが一つの粒子からカットオフ半径内の平均粒子数であり、ベクトル命令の最大ベクトル長 256 に対して 33% しか得られていない。

これに対して、JDS では力の計算部分はベクトル化率 99%、平均ベクトル長 253.01 となり非常に良いルーチンとなっている。しかしタイムステップ毎に行う JDS のテーブル作成の部分が処理時間の 71.70% を占めており力の計算の高速化の分を上まわる結果となった。疎行列を係数行列とする連立一次方程式の反復解法では疎行列自体は変形しないため、一度疎行列を JDS 形式で保持すると、その大きいベクトル長を持つ列方向を繰り返し利用でき JDS 形式を作成するオーバーヘッドは相対的に少なくなるため非常に有効であるが、これに対して、MD の真の粒子対の情報は 1 タイムステップごとに作成する必要があり、JDS 形式に変換するオーバーヘッドが力の計算の大部分を占める結果となった。

特に、このテーブル作成部分では、バケットソートを用いており、今の段階ではこの部分のベクトル化ができていないことがオーバーヘッドが大きくなった原因である。バケットソートのベクトル化に関しては文献 4) で行われており、この実装を組み入れることを検討したい。

4.3 並列化の評価

ここでは、地球シミュレータの 1PN 内での共有メモリモデルによる並列化を行った結果について示す。並列化にはコンパイラによる自動並列化を利用した。これは、ソースプログラム中に含まれるループ処理のような局所的な並列性を抽出し、それらを複数のタスク上で実行する機能であり、OpenMP のワークシェアリングの考え方と基本的に同じである。

ほとんどの場合、この並列化はループ処理に対してのみ適用されるが、地球シミュレータのプロセッサはベクトルプロセッサなので、ループ処理はまずベクトル化が適用される。長いループ処理はストリップマイニングによって分割されこれが並列化の対象になるが、結果としてベクトル長が短くなりベクトル処理の効率低下を招きやすい。

ここで用いた MD のコードでは、最も処理時間を要する力の計算部分において、カットオフ半径内にある粒子同士の演算を疎行列モデルに合せて行った。CRS と JDS の両方で、二重ループの処理になっており、内側のループをベクトル化、外側のループを並列化することが可能となった。もともとこの二重ループを一重

表 2 CRS 形式の並列処理時間 [秒]

粒子数	CPU 数							
	1	2	3	4	5	6	7	8
32000	142.75	89.92	73.00	64.41	58.78	56.20	53.03	51.38
108000	496.61	320.16	262.37	233.35	215.20	204.15	196.00	189.80
256000	1326.42	874.90	725.88	652.66	609.54	578.12	558.35	543.41

表 3 JDS 形式の並列処理時間 [秒]

粒子数	CPU 数							
	1	2	3	4	5	6	7	8
32000	223.17	217.42	216.95	215.42	214.75	214.51	214.32	225.01
108000	783.53	776.93	758.66	757.19	754.36	755.10	753.14	772.08
256000	1971.81	1926.17	1917.33	1904.51	1900.32	1903.79	1899.97	1914.41

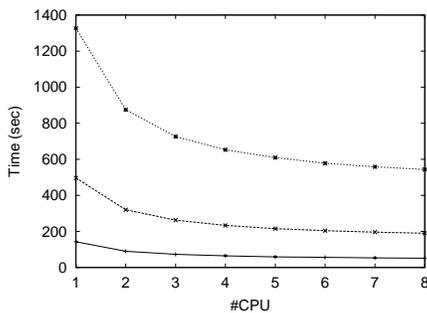


図 5 CRS の処理時間

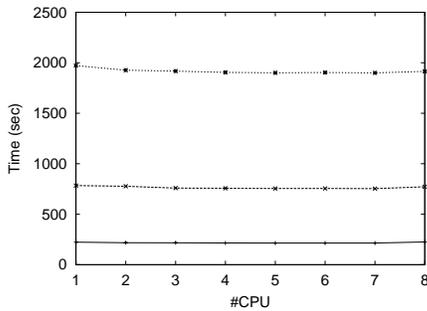


図 6 JDS の処理時間

化すると、その依存関係から単純なベクトル化が不可能であるため、ワーク領域を使う並列化を行う。

それぞれの疎行列モデルに対する並列化プログラムの処理時間の結果を表 2, 3 と図 5, 6 に示す。

並列化コンパイラの実行したコードでは、ワークシェアリングを行うためのループ範囲の計算と本体部分のサブルーチンの切り出しが行われており、これに伴うオーバーヘッドがかかる。このため、1CPU で並列化コードを実行した場合には、逐次版のコードの結果よりも若干時間が長くなっている。

CRS では、力の計算部分が主な時間消費部分であり、

この部分が並列化できている。並列化によるスピードアップは 32000 粒子で 2.7 倍/8CPU、256000 粒子で 2.4 倍/8CPU である。JDS でも力の計算部分は並列化できている。32000 粒子の場合に、4.5 倍/8CPU、256000 粒 4.1 倍/8CPU である。しかし、そもそもテーブルを作る部分がボトルネックとなっており、更に悪いことにこの部分はほとんど並列化できなかったため、CPU 台数によってスピードアップせず、CRS よりも悪い結果となった。バケットソートの並列化については、文献 5) などを参考に今後検討する。

5. おわりに

本研究では、地球シミュレータの 1 ノードを用いてベクトル計算機における分子動力学法の高高速化を行った。

MD の分子間力の計算にはカットオフ半径によって、粒子間の関係が非常に疎なものとなり、いわゆる疎行列を扱う数値計算手法が流用できるかどうか実際にプログラムコードを作成して評価を行った。jagged diagonal 形式による表現は力の計算のカーネル部分のベクトル化及び並列化によって高速化されたが、この形式への変換のオーバーヘッドが高いために、より単純な compressed row 形式の方が高速な結果となった。

今回のプログラムではメモリ効率が悪いので、1 ノードで 25.6 万粒子までしか計算できなかったが、100 万から 150 万粒子程度のシミュレーションが可能であり、複数ノードを用いることで更に大規模な計算が可能となる。今後、大規模分子動力学法の計算エンジンとして地球シミュレータを活用するための高速化手法をさらに開発していきたい。

謝辞

本研究に関して御議論頂いた、地球シミュレータ研究開発センターの諸氏及び関係各位に感謝致します。

参考文献

- 1) 谷 啓二, 横川 三津夫, 「地球シミュレータ計画」

- , 情報処理 Vol.41 No.3 pp. 249-254 (2000).
- 2) 横川 三津夫, 谷 啓二, 「地球シミュレータ計画」, 情報処理 Vol.41 No.4 pp. 369-374 (2000).
 - 3) “Intermetallic Compounds: Vol. 1, Principles and Practice”, Edited by J.H.Westbrook and R.L.Fleischer, Wiley&Sons Ltd.(1995).
 - 4) 村井 均, 末広 謙二, 妹尾 義樹, 「共有メモリ型ベクトル並列計算機上の高速整数ソーティングアルゴリズム」, 情報処理学会論文誌, vol.39, No.6, pp.1595-1602, 1998年6月.
 - 5) 横山 栄二, 安岡 孝一, 岡部 寿男, 金澤 正憲, 「分散メモリベクトル並列計算機上での高速整数ソーティングアルゴリズムの実装」, 情報処理学会論文誌 ハイパフォーマンスコンピューティングシステム, vol.42, No.SIG 9(HPS 3) ,pp.45-53, 2001年8月.
 - 6) “Multi-million particle molecular dynamics: I. Design considerations for vector processing”, D.C. Rapaport, Computer Physics Communications 62 pp.198-216 (1991).
 - 7) “Multi-million particle molecular dynamics: I-I. Design considerations for vector processing”, D.C. Rapaport, Computer Physics Communications 62 pp.217-228 (1991).