DEM におけるマルチカラー接触判定法の適用と マルチコア計算機による性能評価

Adaptation of Multicolor Particle Contact Detection Method for DEM and Performance Evaluation with Multicore Processors

片桐 孝洋^{a*}, 竹田 宏^b, 河村 祥太^c, 加藤 淳也^c, 堀端 康善^c

Takahiro Katagiri, Hiroshi Takeda, Shota Kawamura, Jyunya Kato, Yasuyoshi Horibata

Received 9 April 2014; Accepted 26 May 2014

We propose a new parallel implementation method for particle contact detection processes in the Discrete Element Method (DEM). Conventional implementations require mutual exclusion, which results in their parallel performance being reduced when the execution is highly threaded. By using multicolor contact grids, our implementation method, called the Multicolor Particle Contact Detection Method, accomplishes parallel execution without mutual exclusion. The results of an evaluation conducted indicate that the performance of our proposed method is superior to that of conventional methods.

Keywords: Discrete element method, Particle contact detection, Thread parallelism, Multicolor grids

1. はじめに

離散要素法 (Discrete Element Method (DEM))¹⁾ は、粉体シミュレーションなどの科学技術計算でよく使われている数値計算法である。DEM によるシミュレーションでは、粒子接触判定 (particle contact detection (PCD)) が必要になる。DEM では、すべての粒子が、その他の粒子と接触する可能性がある。そのことから、PCD の処理時間が多くの時間を占める。そこで我々の目的は、この PCD の処理時間を高速化することにある。

PCD の処理時間を高速化するために、接触判定格子がしばしば導入されるが、接触判定格子導入に伴う高速化手法の一つとして、並列化が検討されてきた。しかしながらこの並列化は、処理に内在するデータ依存により高い並列性を引き出すことが一般的には難しいことが知られている。

PCD の処理を並列化するには、いくつかの方法が知られている。その一つは、データ依存を保証するための相互排除をいれて並列化する方法²⁾ である。また、並列化を制御するデータ構造を導入する方法もある。たとえば、pair-wise lists を作る方法³⁾ が知られている。しかしながら、これらの方法はいずれも、相互排除が必要となる。相互排除の処理は並列数が増えるにつれて無視できない時間となるため、特に並列数が増える場合に高い性能が達成できなくなる。

そこで本論文では、PCD の並列化のための新しい並列実装方式を提案する。この提案方式は、PCD 処理において並列性を抽出するために、マルチカラー接触判定格子を利用する方法であり、マルチカラー接触判定法とよぶ。

マルチカラー格子を利用して並列性を抽出する方法

a 東京大学 情報基盤センター

(〒113-8658 東京都文京区弥生2-11-16)

Information Technology Center, The University of Tokyo
(2-11-16 Yayoi, Bunkyo-ku, Tokyo 113-8658, Japan)

b 株式会社アールフロー

(〒340-0015 埼玉県草加市高砂1-10-45高梨ビル)

R-flow Corporation, Ltd.

(Takanashi Bld., 1-10-45 Takasago, Soka, Saitama 340-0015, Japan)

c 法政大学 理工学研究科

(〒184-8584 東京都小金井市梶野町3-7-2)

Graduate School of Science and Engineering, Hosei University
(3-7-2 Kajino-cho, Koganei, Tokyo 184-8584, Japan)

* 責任著者連絡先 katagiri@cc.u-tokyo.ac.jp

は、連立一次方程式の反復解法である SOR 法の並列化においては、Red-Black 法⁴⁾として知られている方法である。一方、分子動力学法 (Molecular Dynamics (MD) method) においては、Spatial Decomposition Coloring (SDC)⁵⁾として知られている方法である。しかしながら、Red-Black 法は解くべき数学上の問題が異なり、かつ、演算対象が格子点上で固定されるため、DEM の粒子のように粒子が頻繁に移動する状況にない。そのため、扱う問題の特性が異なる。一方、SDC が対象としているのは MD であり、本論文で対象とする DEM とは異なる。また MD では、カットオフ判定のための領域に多数の分子が入ることがあり、このことで SDC による並列化でも負荷バランスが悪化することがある。そのため、負荷バランスの劣化が SDC の欠点の一つになっている。DEM におけるマルチカラー接触判定法では、接触判定のための格子幅を粒子径にとると、同一の接触判定格子に入る粒子数は高々数個になる。このことにより、SDC で生じるような負荷バランスの劣化の問題が DEM では生じないため、問題特性が異なるといえる。

我々は、報告⁶⁾でマルチカラー接触判定格子を導入することにより、OpenMP による並列化効率を改善できる可能性を示した。そこで本論文では、DEM におけるマルチカラー接触判定法のアルゴリズムを提案する。さらに、計算機アーキテクチャの異なる複数の並列計算機による性能評価により、提案するマルチカラー接触判定法の有効性を示すことを目的とする。

本論文の構成は以下のとおりである。まず 2 章で、DEM のためのマルチカラー接触判定法の説明をする。また、提案したマルチカラー接触判定法を用いて、OpenMP でスレッド並列化する実装について説明する。3 章は、近年の 3 種のマルチコア CPU を用いて、マルチカラー接触判定法を評価する。最後に本論文の結論を述べる。

2. マルチカラー接触判定法

2.1 DEM の処理と粒子接触判定

DEM において標準的な処理は論文⁷⁾に記載されている。Fig. 1 に、この標準的な DEM の処理⁷⁾における、PCD の流れ図を示す。

計算量を減らして粒子間接触を見つけるため、接触判定格子を用いる。このことで、PCD 処理における計算量を $O(N^2)$ から $O(N)$ に減らすことができる。ここで、 N は粒子数である。Fig. 1 の PCD 処理は、以下である：(i) 接触する粒子に関する力の計算、(ii)

Fig. 1 Flow of operation among the PCD processes in the standard DEM⁷⁾

それぞれの粒子についての力の足しこみ、(iii) 粒子速度と位置の更新。並列処理では、粒子同士の演算にデータ依存があるため、(i) 接触する粒子に関する力の計算、について考慮しなければならない。

2.2 PCD 処理におけるデータ依存

前節で説明したように、粒子接触判定時の演算におけるデータ依存は、接触する粒子数が増えるほど生じやすくなる。この節では、このデータ依存について説明する。

いま、 π_i を、 i 番粒子とする。ここで、 $i=1, 2, \dots, N$ である。 ϕ_j を、 j 番の接触判定格子とする。ここで、 $j=1, 2, \dots, N_{\text{grids}}$ であり、 N_{grids} は、接触判定格子の数である。これらの表記を使い、 ϕ_j に存在する粒子の集合を Π_j ($j=1, 2, \dots, N_{\text{grids}}$, $\Pi_j \in \Pi$) とする。ここで、 Π は全粒子の集合、 $\{\pi_1, \pi_2, \dots, \pi_N\} \in \Pi$ である。 Φ_i を、粒子 π_i ($i=1, 2, \dots, N$) と接触する粒子の集合とする。ここで、 $\Phi_i \in \Pi$ である。

いま、粒子 π_k と π_l との接触を考える。この接触について、力が双方の粒子に働く。ここで、粒子 π_k が粒子 π_l に対して働く力を $F_{k,l}$ とする。このとき、作用反作用の法則を利用して、粒子 π_k と π_l の間の力 ($F_{k,l}$ と $F_{l,k}$) の計算のための演算を減らすことができる。すなわち、粒子 π_l の力について、 $F_{l,k} = -F_{k,l}$ とする。そこで、力の計算は、粒子 π_k ($k < l$) のみ行うことにする。接触粒子の集合 Φ_i は、 $k < l$ の範囲内のみで定義する。この関係を説明したのが、Fig. 2 である。

Fig. 2 Action and reaction for two particles and their contact forces

Fig. 2 では、力の計算を並列化する場合、粒子 π_k と π_l との間の力を加算する時、この力が収納されている配列の加算に排他制御が必要となる。特に $F_{l,k}$ の値を計算する時に、排他制御が必要になる。一般的に、排他制御には、OpenMP などのスレッド並列化や Message Passing Interface (MPI) などのプロセス並列化に関わらず、大きな計算負荷（オーバーヘッド）がかかる。したがって、Fig. 2 の並列化時の最大の問題は、排他制御のための計算時間の削減にある。

2. 3 従来の並列アルゴリズム

大別して PCD 処理の並列化には、2つの方法がある。それは、粒子並列と格子並列である⁷⁾。これらの方法はプログラムレベルでみると、いずれも最外側のループの並列性を利用し、OpenMP や MPI で並列化するものである。

粒子並列では、それぞれの粒子における力の計算の並列性を利用する。したがって、最外ループは粒子番号で形成される。一方、格子並列では、格子レベルの並列性を利用する。したがって、接触判定格子番号により、最外ループが形成される。DEM における PCD 処理では、従来の並列アルゴリズムはすべて格子並列で作られている。本論文で提案する方法も、格子並列に基づいている。

2. 3. 1 相互排除を用いた並列化

Fig. 3 は、OpenMP による相互排除を用いた従来の格子並列による実装を示している。ここで、相互排除は、いくつかの方法で実装できる。たとえば、atomic 演算子による方法や、クリティカルセクションによる方法があげられる²⁾。Fig. 3 は、クリティカルセク

```

< 1> Make sets of  $\Pi_j$  ( $j=1, 2, \dots, N_{\text{grids}}$ ) for each grid,
and  $\Phi_i$  ( $i=1, 2, \dots, N$ ) for each particle in current
particle contact detection.
!$ OMP PARALLEL DO
< 2> DO  $j=1, N_{\text{grids}}$ 
< 3> DO WHILE ( $\Pi_j$  is not an empty set)
< 4>  $\pi_k \in \Pi_j$ 
< 5> DO WHILE ( $\Phi_k$  is not an empty set)
< 6>  $\pi_l \in \Phi_k$ 
< 7> Compute  $F_{k,l}$ . !This is Process (i) in Fig. 1.
!$ OMP CRITICAL! This is Process (ii) in Fig. 1.
< 8> Add  $F_{k,l}$  to array of force for  $\pi_k$ .
< 9> Add  $-F_{k,l}$  to array of force for  $\pi_l$ .
!$ OMP END CRITICAL
< 10>  $\Phi_k = \Phi_k - \{\pi_l\}$ 
< 11> END WHILE
< 12>  $\Pi_j = \Pi_j - \{\pi_k\}$ 
< 13> END WHILE
< 14> END DO

```

Fig. 3 Conventional grid parallelism implemented using OpenMP with critical section (Critical Version)

ンによる方法を用いた例である。atomic 演算子による実装は、ハードウェアによる atomic 演算のサポートがあり、かつ、OpenMP コンパイラによるハードウェア命令の利用がなされる場合は、クリティカルセクションによる実行よりも高速である²⁾。しかし、クリティカルセクションによる方法は、ハードウェア命令に依存せず実現できるため、相互排除を用いた従来実装の性能の特徴が示される。そのため、本論文ではクリティカルセクションによる実装を採用する。Fig. 3 の実装を、Critical Version とよぶ。

2. 3. 2 相互排除を用いない並列化

Fig. 3 では、接触する粒子間の力の計算のための演算量を減らすために、作用反作用の法則を利用した。しかしながら、この演算量削減のために生じたデータ依存関係により、相互排除を必要とする欠点が生じた。そこで、この欠点を取り除くための実装を考える。

相互排除を取り除くため、接触する粒子の集合 Φ_k を再定義して $\hat{\Phi}_k$ とする。ここで、集合 $\hat{\Phi}_k$ は、粒子 π_l と接触する粒子 π_k ($k > m$) だけでなく、粒子 π_m ($l < m$) を含むものとする。この上で、Fig. 3 の Critical 文を取り除く。Fig. 4 に、Fig. 3 から Critical 文を取り除いた実装の差分のみを示す。

Fig. 4 では、<7>行の力 $F_{k,l}$ の計算に関して、Fig. 3 の<7>行の計算の2倍の演算量を必要とする。なぜなら、計算対象の接触する粒子集合の要素数 $|\hat{\Phi}_k|$

```

< 5'> DO WHILE ( $\hat{\Phi}_k$  is not an empty set)
< 6'>    $\pi_i \in \hat{\Phi}_k$ 
< 7'>   Compute  $F_{k,1}$  !This is Process (i) in Fig. 1.
< 8'>   Add  $F_{k,1}$  to array of force for  $\pi_k$ .
< 9'>    $\hat{\Phi}_k = \hat{\Phi}_k - \{\pi_i\}$ 
<10'> END WHILE

```

Fig. 4 Conventional grid parallelism implemented using OpenMP without critical section (Redundant Computation Version)
Only the sections that are different from Fig. 3 are shown.

が、元の $|\Phi_k|$ より 2 倍増えているからである。これは、逐次演算において、力の計算部分を冗長にしていることにほかならない。このことから、Fig. 4 の実装を、Redundant Computation Version (冗長計算) とよぶ。

2. 4 マルチカラー接触判定格子

DEM での接触判定演算におけるデータ依存を考慮し、接触判定格子の演算の順序を逐次から変更する。ただし、簡単のため均一粒子径を仮定し、接触判定格子幅を粒子径と同じに取るものとする。

Fig. 5 に、逐次のデータ依存関係を緩和した 2 次元格子の例を示す。ここで、 $\Pi_1 = \{\pi_1\}$, $\Pi_2 = \{\pi_2\}$, $\Pi_4 = \{\pi_4\}$, $\Pi_5 = \{\pi_3, \pi_5\}$, $\Pi_9 = \{\pi_6\}$, および、 $\Phi_1 = \{\pi_2, \pi_3, \pi_4\}$, $\Phi_2 = \{\pi_3\}$, $\Phi_3 = \{\pi_5\}$ である。

Fig. 5 では、 $color=0$ の格子にある粒子については同時に計算しても、力の計算結果の収納と足しこみに

Fig. 5 Multicolor particle contact grids (2D grids)

```

< 1'> Make sets of  $\hat{\Pi}_c$  ( $c=1, 2, \dots, N_{\text{colors}}$ ) for each
      grid, and  $\Phi_i$  ( $i=1, 2, \dots, N$ ) for each particle in
      current particle contact detection.
< 2'> DO  $c=1, N_{\text{colors}}$ 
!$ OMP PARALLEL DO
< 3'> DO  $j=1, |\hat{\Pi}_c|$ 
...
< 7'>   Compute  $F_{k,1}$ . !This is Process (i) in Fig. 1.
< 8'>   Add  $F_{k,1}$  to array of force for  $\pi_k$ .
< 9'>   Add  $-F_{k,1}$  to array of force for  $\pi_i$ .
...
<14'> END DO
!$ OMP END DO PARALLEL
<15'> END DO

```

Fig. 6 Multicolor particle contact method

ついて、同一格子内にある粒子を除くすべての粒子間で衝突することがない。一方、同一格子内の粒子間接触力の計算は、格子並列では逐次処理されるため、相互排除処理は必要としない。この状況は、 $color=1, 2$, および、3 の格子にある粒子についても同様である。たとえば **Fig. 5** において、粒子 π_1 は粒子 π_6 に直接触れないため、**Fig. 5** の計算が可能となる。このことから、同じ色の接触判定格子の計算は、相互排除なしで並列処理が可能である。この色分けされた格子を、マルチカラー接触判定格子とよぶ。

2. 5 マルチカラー接触判定法

Fig. 5 における演算は、容易に多次元格子へ拡張可能である。たとえば、3D の格子では、最低 8 色の格子を用意すると 2D の **Fig. 5** と同様の演算が構築できる。

いま、マルチカラー接触判定格子の色の数を N_{colors} とする。色番号 c での格子に存在する粒子の集合を $\hat{\Pi}_c$ ($c=1, 2, \dots, N_{\text{colors}}$), $\hat{\Pi}_c \subseteq \Pi$, とする。これらの表記から、**Fig. 3** の計算法を再構築できる。**Fig. 6** に、マルチカラー接触判定格子を用いた新しい計算法を載せる。**Fig. 6** を、マルチカラー接触判定法とよぶ。

ここで、**Fig. 6** の <8'> 行と <9'> 行は、冗長計算を用いていないにもかかわらず、Critical 文がない。この理由は、マルチカラー接触判定法では、すでに述べたように同一色の格子内では、データの収納と足しこみについての衝突がスレッド間で起きない (逐次処理で行う) ことが保証されているためである。

3. 性能評価

3. 1 利用計算機

ここでは以下の 3 種類の計算機を用いて、提案手法を評価する。

東京大学情報基盤センターに設置されている FX10 スーパーコンピュータシステム (FX10) の 1 ノードを利用した。FX10 の CPU は、SPARC64 IX-fx (1.848 GHz) であり、1 ノードあたり 16 コアを有する。理論最大性能は、236.5 GFLOPS である。1 ノード当たりのメモリ量は、32 GB であり、総合的なメモリバンド幅は 85 GB/秒である。富士通 Fortran 90 コンパイラ version 1.2.1 P-id: T01641-02 を利用し、コンパイラオプションは “-Kfast, openmp” である。PCD 処理には、インライン展開して最適化を指示している。

次に、Intel Ivy Bridge を用いた。CPU は、Intel Xeon E5-2670 V2 (2.50 GHz)、2 ソケットあり、それぞれのソケット内に 10 コアを有する。ハイパースレッディング (HT) はオンで、1 コアあたり 2 つのスレッドを割り当て可能である。1 ノードで最大 40 スレッド実行ができる。1 ノードの理論性能は 400 GFLOPS、1 ノード記憶容量は 64 GB である。コンパイラは Intel Fortran version 14.0.2.144 Build 20140120、コンパイラオプションは “-ipo20 -O3 -openmp -mcmmodel=medium” を指定した。コアとスレッドの割り当て方式として、“KMP_AFFINITY=granularity=fine, compact” を指定した。これは、コアそれぞれに、順番に 2 スレッドずつ割り当てていく方式である。

最後に、Intel Xeon Phi を用いた。CPU は、Intel Xeon Phi 5110P (B1 stepping) (1.053 GHz)、60 コアである。HT はオンであり、最大で 1 コア当たり 4 つのスレッド割り当てができる。1 ノードの理論ピーク性能は 1 TFLOPS である。1 ノード記憶容量は 8 GB である。コンパイラは、Intel Fortran 14.0.2.144 Build 20140120、コンパイラオプションは “-ipo20 -O3 -openmp -mcmmodel=medium -mmiccl-align array64byte” を指定した。コアとスレッドの割り当て方式として、“KMP_AFFINITY=granularity=fine, balanced” を指定した。これは、スレッドをコアに均等になるように割り当てていく方式である。本実験では、Xeon Phi の物理 60 コアに、それぞれ 1 スレッドずつ割り当てて実行する。

以上の計算環境での逐次実行では、OpenMP を利用しないようにコンパイルして実行する。この理由は、コンパイラで OpenMP 並列化してスレッド数を 1 とする実行では、OpenMP のオーバーヘッドにより、OpenMP なしの逐次実行より遅いことがあるからである。この逐次実行より遅い時間を OpenMP での 1 スレッド実行時間に採用すると、並列性能の評価において公平な評価にならないためである。

3. 2 ベンチマークの設定

ここでは、重力場での粒子の自由落下現象を、3 次元の DEM を用いて解く。粒子間および粒子壁間接触力の計算では、簡単のため、法線方向の力のみを考慮し、接線方向の接触力は無視した。粒子の初期状態は、xyz 空間上の区間 [0,1], [0,1], [0,1] において、位置を乱数で設定した。なお乱数生成には、Fortran 90 の組み込み関数 random_number を利用した。乱数の種の設定は random_seed 関数を利用し、同一の値の “2” を設定した。乱数生成の実装が機種ごとで異なるため、全く同じ乱数列が生成されていない。また同一の機種でも、OpenMP 並列化による演算順序の変更により、時間刻みが進むにつれてスレッド数に依存し、同一の接触判定数にならない。したがって、機種ごと、および、並列数により、各時間刻みにおける粒子の接触判定数は同一にならないが、差はわずかで性能評価には影響を与えないと考えられる。たとえば、マルチカラー接触判定法において逐次実行で、第 1 時間刻みの接触判定数は、FX10 では、375,165 回であったが、Ivy Bridge と Xeon Phi では、374,614 回であった。

このベンチマーク問題での粒子数は 62,499,988 粒子である。粒子径は 0.002 m、粒子密度は 1000 kg/m³、ばね定数は 100 N/m、反発係数は 0.8 である。接線方向の接触力を無視しているため、摩擦力は働かず、摩擦係数は 0 である。解析領域は一辺 1 m の立方体で、粒子の充填率は約 26.2% である。また、シミュレーションにおける時間刻み幅は 0.0001286 秒である。

測定した時間 [秒] は、第 5 ステップから第 20 ステップまでの PCD 時間のみを蓄積して、その後、16 で除して平均化した時間 (平均ステップ時間) である。この評価を採用した理由として、シミュレーション開始直後は粒子衝突数が大きく時間がかかるため、実際のアプリケーションの時間を反映していない恐れがあるからである。

3. 3 性 能

Tables 1, 2, 3 に、FX10, Ivy Bridge, および Xeon Phi における性能をのせる。

Tables 1, 2, 3 では、いずれの計算機環境でも、逐次実行では Critical Version が最も高速である。これは、逐次では、排他制御のための時間がなく、かつ、演算量が少ないことによる。原理的には、演算量は Critical Version とマルチカラー接触判定法 (Multi-color) は同じである。しかし、Critical Version に対してマルチカラー接触判定法は、格子の色ごとに粒子

Table 1 Execution time in the FX10
Time is in seconds.

#Threads	Critical (Speedup)	Redundant (Speedup)	Multi-color (Speedup)
Sequential	34.42 (1.00x)	54.98 (1.00x)	38.09 (1.00x)
2	51.07 (0.67x)	30.30 (1.81x)	20.40 (1.87x)
4	38.03 (0.91x)	16.93 (3.25x)	10.99 (3.47x)
8	48.64 (0.71x)	11.66 (4.72x)	6.98 (5.46x)
16	78.47 (0.44x)	6.81 (8.07x)	4.12 (9.25x)

Table 2 Execution time in the Ivy Bridge
Time is in seconds.

#Threads	Critical (Speedup)	Redundant (Speedup)	Multi-color (Speedup)
Sequential	10.22 (1.00x)	17.07 (1.00x)	11.46 (1.00x)
2	11.40 (0.90x)	12.75 (1.34x)	8.49 (1.35x)
4	7.26 (1.41x)	6.79 (2.51x)	4.27 (2.68x)
8	7.15 (1.43x)	3.85 (4.43x)	2.50 (4.58x)
16	7.78 (1.31x)	2.18 (7.83x)	1.41 (8.13x)
20	7.96 (1.28x)	1.76 (9.70x)	1.14 (10.0x)
32	16.42 (0.62x)	2.43 (7.02x)	1.74 (6.59x)
40	19.48 (0.52x)	2.24 (7.62x)	1.67 (6.86x)

をアクセスする都合から、収納された粒子データの配列を連続にアクセスしない。このことから逐次処理においては、マルチカラー接触判定法では、キャッシュヒット率の悪さが性能劣化を引き起こしている可能性がある。

2 スレッド以上の並列実行では、提案法であるマルチカラー接触判定法の実行が、従来法に対して高速である。実行時間と台数効果の観点から、スレッド数が増加すると従来法よりもマルチカラー接触判定法が高速になっていく。マルチカラー接触判定法による台数効果は、計算機環境により異なるが約9.2倍 (FX10) ～約35倍 (Xeon Phi) である。

Table 3 Execution time in the Xeon Phi
Time is in seconds.

#Threads	Critical (Speedup)	Redundant (Speedup)	Multi-color (Speedup)
Sequential	90.26 (1.00x)	152.88 (1.00x)	109.26 (1.00x)
2	94.00 (0.96x)	364.92 (0.42x)	58.53 (1.87x)
4	60.14 (1.50x)	248.50 (0.62x)	31.96 (3.42x)
8	61.73 (1.46x)	330.73 (0.46x)	20.42 (5.35x)
16	71.51 (1.26x)	85.79 (1.78x)	8.47 (12.9x)
32	65.81 (1.37x)	103.79 (1.47x)	5.59 (19.5x)
60	66.98 (1.35x)	46.12 (3.31x)	3.06 (35.7x)

Critical Version では計算機環境によらず、スレッド数が増加すると性能が劣化する。これは、排他制御のオーバーヘッドのためと推測される。特に Table 3 の Xeon Phi での実行は、マルチカラー接触判定法の並列性能が、従来手法に比べ最大で約21倍 (60スレッド実行時、Critical Version に対して) と高い。

Table 3 の Xeon Phi では、スレッド並列化時において Redundant Version の実行時間が、スレッド数が増えるにしたがい増加することがある (たとえば、8 スレッド、32スレッドでの実行)。この理由は不明であるが、考えられる理由として、Xeon Phi ではメモリと CPU 間の相互結合網がリング・インターコネクトになっていることがあげられる。ループ内の配列アクセスの量が Critical Version とマルチカラー接触判定法に対して2倍となりアクセス量が多くなると、リング・インターコネクト内のトラフィックが増加して、メモリアクセス性能が劣化する可能性がある。そのため、スレッド数が増えるとメモリアクセス過多による性能劣化を生じることがあるため、と推察される。また、このリング・インターコネクト内のトラフィックの増加によるメモリアクセス性能の劣化により、FX10 や Ivy Bridge での Redundant Version の性能に対し、Xeon Phi での Redundant Version の性能は、相対的に悪くなっている可能性がある。

今回の実験条件では、62,499,988粒子を単精度でメモリ確保しているため、粒子情報のための配列確保だけでも 238 MB 以上の領域が必要となる。これは、以上の3機種最大のキャッシュサイズを超えている。そ

のため、粒子数を増加させて大規模シミュレーションする場合においても本実験結果に対してキャッシュの影響はなく、本実験と同程度のメモリアクセス性能となると予想される。したがって粒子数を増やしても、本実験と同程度の並列実行効率が期待できる。

以上の結論として、提案法であるマルチカラー接触判定法は、スレッド数が多い状況において特に効果的である。

4. おわりに

本論文では、DEMのためのマルチカラー接触判定法を提案した。マルチカラー接触判定法では、逐次処理におけるデータ依存関係を緩和するため、マルチカラー接触判定格子を用い、演算順序の変更を行うことで並列性を抽出し、DEMのPCD処理における並列性をより良く抽出する。マルチコアCPUを用いた性能評価から、マルチカラー接触判定法は従来法に対して、より良い並列性能を抽出できることを示した。

本論文では、均一粒子径を仮定したが、マルチカラー接触判定格子の幅を最大粒子径（あるいはそれ以上）に取ることで、マルチカラー接触判定法は粒子径分布がある場合に対してもそのまま適用できる。なお、3.2節の均一粒子径に対するベンチマークでは、近接粒子探索のための接触判定格子幅とマルチカラー判定格子幅をともに粒子径に等しく取ったが、一般には、マルチカラー判定格子幅を近傍粒子探索のための接触判定格子幅に合わせる必要はない。たとえば、近傍粒子探索のための接触判定格子幅を最大粒子

径よりも小さくとることによりPCDの効率を上げることができるとの報告があるが⁸⁾、マルチカラー接触判定法では、近傍粒子探索のための接触判定格子幅をマルチカラー判定格子幅と独立に設定してもよいため、このような方法との併用も可能である。これらの手法の評価については、今後の課題である。

PCD処理におけるデータアクセスパターンを最適化するため、粒子を整列することは有効であることが報告されている⁹⁾。この粒子の整列は、マルチカラー接触判定法でも有効と考えられる。粒子の整列に加えて、マルチカラー接触判定法のアクセスパターンに特化し、連続アクセスするためのデータ構造を導入することも、おそらく有効であると考えられる。この実装と評価は、今後の課題である。

一方、近年の並列計算機では、MPIとOpenMPを同時に用いて並列化する、ハイブリッドMPI-OpenMP並列化の研究が盛んである。本論文で提案したマルチカラー接触判定法は、ノード内におけるOpenMP並列化の観点で有効であるばかりでなく、ノード間のMPI並列化においても有効であると考えられる。マルチカラー接触判定法を、ハイブリッドMPI-OpenMP並列化に適用して評価することは、重要な今後の課題である。

〔謝辞〕 本研究の一部は、学際大規模情報基盤共同利用・共同研究拠点、および、革新的ハイパフォーマン
ス・コンピューティング・インフラの支援による。

References

- 1) Cundall P. A. and O. D. L. Strack : "A Discrete Numerical Model for Granular Assemblies", *Geotechnique*, **29**, 47-65 (1979)
- 2) Sakai, M. : "Numerical Simulation of Granular Flows", Maruzen (2012)
- 3) Nishiura, D. and H. Sakaguchi : "Parallel-vector Algorithms for Particle Simulations on Shared-memory Multiprocessors", *Journal of Computational Physics*, **230**, 1923-1938 (2011)
- 4) Evans, D. J. : "Parallel S. O. R. Iterative Methods", *Parallel Computing*, **1**, 3-18 (1984)
- 5) Hu, C., Y. Liu and J. Li : "Efficient Parallel Implementation of Molecular Dynamics with Embedded Atom Method on Multi-core Platforms", Int. Conf. Parallel Process. Work., pp. 121-129 (2009)
- 6) Kato, J., S. Kawamura, H. Takeda, T. Katagiri and Y. Horibata : "Parallelization of Computations for Particle Contact Detection with Multi-color Contact Detection Grid by OpenMP", *IPSJ SIG Notes*, **2013-ARC-207**, (7), 1-6 (2013)
- 7) Sakaguchi, H. and D. Nishiura : "Development of Hyper Intelligent Discrete Element Method (HiDEM) and its Application for Science and Industry", *JAMSTEC-RIFREE Special Issue*, 201-210 (2009)
- 8) Mio, H., A. Shimosaka, Y. Shirakawa and J. Hidaka : "Optimum Cell Size for Contact Detection in the Algorithm of Discrete Element Method", *J. Chem. Eng. Japan.*, **38**, 969-975 (2005)
- 9) Takagi, S., N. Wada, D. Oka, H. Takeda, T. Katagiri and Y. Horibata : "Parallelization of Particle Contact Detection with MPI and OpenMP with Respect To Particle Distribution", *IPSJ SIG Notes*, **2012-HPC-137**, (34) 1-6 (2012)