

jh130001-NA01

## 粉体解析アルゴリズムの並列化に関する研究

片桐孝洋 (東京大学)

**概要** 実用で扱う多くの粉体は、億、兆、あるいは、それ以上の膨大な数の粒子から構成されているにもかかわらず、逐次計算では計算時間の制約から 100 万個程度の粒子を解析することが限界となっている。そのため、粉体解析アルゴリズムの並列化が早急の課題となっている。特に粉体解析アルゴリズムでは、接触判定計算部分について高い並列性を確保するのが課題となっている。ところが、近年普及しているマルチコア計算機では、性能に大きく影響をおよぼすキャッシュメモリやローカルメモリが多階層化されている理由から、粉体解析ソフトウェアではスレッド並列化時に実行状況に依存し、並列性が確保できないことがある。そこで本研究では、主に接触判定計算部分において、近年普及したマルチコア計算機を用いて計算負荷の均等化による並列化効率の向上と、マルチコア計算機特有の階層キャッシュの最適化に焦点を当てた研究を行う。

### 1. 研究の目的と意義

#### (1) 研究目的

実用で扱う多くの粉体は、億、兆、あるいは、それ以上の膨大な数の粒子から構成されているにもかかわらず、通常の逐次計算では計算時間の制約から 100 万個程度の粒子を解析することが限界となっている。そのため、粉体解析アルゴリズムの並列化が早急の課題となっている。粉体解析では、接触判定計算の計算負荷が最も大きく高速化が必要となる。本研究では、粒子の接触判定計算を主な対象とし、解析領域内の粒子分布に依存しない手法の研究開発を目的とする。

一方、近年マルチコア計算機が広く普及している。このマルチコア計算機は、スーパーコンピューター (スパコン) に代表されるハイエンド計算機だけではなく、企業におけるものづくりで使われる PC まで使われている。したがって、マルチコア計算機において高い並列化性能を達成できる並列化方式を実現することが必須となっている。そこで本研究では、マルチコア計算機に向く並列化方式の研究を目的とする。

マルチコア計算機に向く並列化方式として、ノード内は OpenMP によるスレッド並列化、ノード間は MPI による分散並列化を行うハイブリッド MPI 実行形態が現在の並列化の主流になっている。ハイブリッド MPI 実行のためには、ノード内およびノード間において、それぞれ効率的な実装方式と

数値計算アルゴリズムを探求しなくてはならない。したがって、本研究の目的とする。

#### (2) 研究の意義

##### N 体問題や MD との問題性質の違い

宇宙物理分野で取り扱う粒子シミュレーションでは、粒子間の相互作用について遠方の粒子に働く力も取りあつかわなければならない。その相互作用の力を近似するため、高速多重極法 (FMM) 法や Barnes Hut 法などにより、計算量の削減と並列性の向上をしている。また、MD (Molecular Dynamics) の分野ではカットオフ距離がある。多数の分子がそのカットオフ距離内に入ること、負荷バランスの劣化を生じることがある。

以上の N 対問題や MD における並列化の問題と、粉体解析に利用される DEM (Discrete Element Methods) とでは、並列処理の観点で問題の性質がまったく異なる。粉体解析では、遠方の粒子の力の計算はほとんど必要とせず、粒子間の衝突による力の計算が主体となる。また、全ての粒子がすべての粒子と接触する可能性があるため、接触判定の計算量を FMM のような物理法則を利用して削減することができない。したがって DEM は、N 対問題などの宇宙物理分野のシミュレーションや MD とは問題の性質がまったく異なる。さらに計算負荷のばらつきの観点では、接触判定格子の大きさを粒子径とほぼ等しくすることで、接触判定格子内に入る粒子数を高々数個にすることができる。ま

た、粒子が存在する接触判定格子内の演算のみ行うことで、余分な演算が生じない。この理由から、負荷バランスの劣化を少なくすることが原理的に可能である。

このように MD と DEM の問題特性の違いから、並列化時の特性がまったく異なる。本研究では、粉体解析に用いられる DEM を取り扱う。

### 粉体解析の処理と本研究での焦点

粉体解析においては、粒子同士の接触判定計算の計算負荷が大きくなる。接触判定計算部分について、高い並列性を確保するのが課題となっている。ところが、近年普及しているマルチコア計算機では、性能に大きく影響をおよぼすキャッシュメモリやローカルメモリが多階層化されている。粉体解析ソフトウェアではスレッド並列化においても、実行状況により並列性が確保できないことが報告されている。

そこで本研究では、主に接触判定計算部分において、近年普及したマルチコア計算機を用いて計算負荷の均等化による並列化効率の向上と、マルチコア計算機特有の階層キャッシュの最適化に焦点を当てた研究を行う。このことは、粉体解析ソフトウェアの実行性能に大きく影響する事項であるので、研究の意義は大きい。

一方、上記スレッドレベルの高並列化の研究に加えて、近年開発されたスパコンによる実行では、1 万並列を超える並列性を確保しないとイケない。この状況では、MPI に代表される分散メモリ並列化への対応が必須となる。そこで第二の意義として、粉体解析における分散メモリアルゴリズムの設計を目的にする。そのため、通信量、通信回数などの最適化を行うことを目的にするが、これもシミュレーションの実行条件に依存する。そのため、どのようなシミュレーション条件で、どういう通信時間の削減方法が有効であるかも検証する。

以上の 2 つの意義を実現すると、近年の超並列実行の前提となる MPI によるプロセス実行と OpenMP によるスレッド実行の混合によるハイブリッド MPI-OpenMP 実行が実現される。この際、計算

機の全資源（コア）を使い切るためには、プロセス数とスレッド数の組合せが生じる。このことで、性能チューニングのためのチューニングパラメータが複雑化され、チューニングのコストが増大する。この理由から第三の意義として、自動性能チューニング (Auto-tuning、AT) 技術を適用することで、チューニングに関するコストの削減を狙う。

## 2. 当拠点公募型共同研究として実施した意義

### (1) 共同研究を実施した拠点名および役割分担

- **東京大学・情報基盤センター**: 取りまとめ、並列化、性能チューニング、自動チューニングの適用
- **法政大学・理工学部**: 数値計算法に関する助言、OpenMP および MPI を用いた実装、性能評価、および改善方式の提案
- **株式会社アールフロー**: 粉体解析アルゴリズムに関する助言

### (2) 共同研究分野

超大規模数値計算系応用分野

### (3) 当公募型共同研究ならではの事項など

並列化・性能チューニングを専門とするコンピュータ・サイエンス研究者、数値アルゴリズムを専門とする数理工学研究者、および企業における粉体解析ソフトウェアを開発している計算科学を基礎とする技術者が連携して同一課題に取りかかる。このことで、コンピュータ・サイエンスと計算科学の協調、および学際研究を実現し、より汎用的で適用範囲の広い方式の実現を狙う。

## 3. 研究成果の詳細と当初計画の達成状況

### (1) 研究成果の詳細について

粉体計算や荷電粒子のシミュレーションを扱う粒子解析においては対象とする問題ごとに様々な運動方程式を立てて計算するが、共通して行われる処理は粒子接触判定計算である。粒子接触判定計算では、全体の時間の大部分を占めると言われており、高速化が必須の処理となっている。特に近年の計算機においては、ノード内に 8 スレッド

以上の並列実行可能な CPU が普及し、並列化による高速化が課題となっている。

粒子接触判定計算においては、従来、接触判定計算の計算量を減らす方法が使われてきた。これは、接触判定計算を全粒子について行うのではなく、接触する可能性のある近傍の粒子についてのみ物理量の計算を行う方法である。この近傍の粒子を絞り込むために、接触判定格子を用いる方法が知られている。また、キャッシュを有効利用することで並列化時の台数効果を上げるため、粒子に番号を振ったうえで、粒子を接触判定時の演算が局所化するように並び替える方法が知られている。

### 提案法の概要

本研究では、DEM の接触判定計算において更なる高速化を達成するため、新しい接触判定計算の方法を提案する。DEM で必要となる接触する各粒子に働く接触力の足しこみ計算においては、従来は作用-反作用の法則を用いて演算量の削減を行っている。この従来法をスレッド並列化する時には、接触する相手の情報も同時に更新することになる。しかし、この操作は他スレッドが更新する情報の書き換えにほかならない。ゆえに、複数のスレッド間でデータの読み書きが競合するため、排他制御が必要となる。この排他制御は OpenMP を利用する場合、critical 指示子や atomic 指示子を入れるのが普通である。

この一方で、critical 指示子等を入れない並列化では、接触時に相手の情報を更新しない方法により排他制御を回避してきた。しかしこの方法は、critical 指示子を入れる方法に対し約 2 倍の演算量の増大になる。

つまり、排他制御する方法では演算量が 1/2 になるメリットがある反面、排他制御によるスレッド並列化のオーバーヘッドが生じる。一方、排他制御がない方法では演算量が増える反面、スレッド並列化時のオーバーヘッドが少ない。両者は欠点と利点があり、状況に応じて最適な方法が異なる。

そこで本研究では、演算量は critical 指示子による方法と同じく 1/2 にするが、排他制御を必要としない新しい方法を提案する。この方法を、**マルチカラー接触判定法**と命名した。マルチカラー接触判定法で用いる接触判定格子のことを、**マルチカラー接触判定格子**と呼ぶ。ここでは、従来の方法とマルチカラー接触判定法とのスレッド並列実行性能を比較することで、提案法の有効性を検証することを目的とした。

### 従来法および類似解法との違い

MD の分野では、マルチカラー接触判定格子を用いて、短距離の分子間力の計算を並列化法が提案されている。それを、Spatial Decomposition Coloring (SDC) と呼ぶ[1]。すでに説明したように、MD と DEM は数学上の問題が異なり、並列化の際のデータ依存の特徴も異なるため、本手法は SDC に対し独創性が主張できる。さらに、先行研究[1]では、SDC のスレッド並列化のみが性能評価されている。ハイブリッド MPI-OpenMP 実行での性能評価がされていない。この観点においても、本研究の独創性が主張できる。

一方、解くべき数学上の問題が全く異なるが、連立一次方程式の反復解法の 1 つである SOR 法において、Red-Black 法 [2] が知られている。Red-Black 法では並列性を抽出するため、計算格子に 2 色の色を付け、その色ごとに計算を行う。この色数を拡張することで、マルチカラー法に拡張できる。したがって考え方としては、マルチカラー格子を利用した並列化方式といえる。しかし Red-Black 法と本手法との違いは、数学上の解くべき問題が全く異なることに加えて、問題性質の違いが指摘できる。Red-Black 法では、演算対象となる物理変数は演算格子に張り付けられ、シミュレーション中に動くことは無い（ただし、動的に格子を変更する場合はそうでない。しかし、通常は動的に格子を変更する頻度は少ない）。演算対象が動かない特徴を利用していることから、並列処理時のデータ依存が DEM とは異なる。DEM では、演算対象の粒子は頻繁に移動する。したがって、

Red-Black 法に比べ、データ構造の最適化を含む性能最適化がより困難となる。そのため、並列性能においても、Red-Black 法による並列性能と比べることは不平等であるばかりか、参照データとしても意味をなさない。

### 接触判定計算

#### ● プログラム全体の流れ

本研究における接触判定計算までの、プログラム全体の手順は以下のとおりである。

まず初めに、粒子を疑似乱数（以後、乱数）で発生させて粒子を登録する。

次に、粒子番号の並び替えを行う。この並び替えは、データアクセスを局所化するために行うものであり、我々の先行研究により効果が実証されている。

#### ● 接触判定格子と粒子登録

本研究では、解析対象となる 3 次元領域に任意の数だけ粒子を発生させる。数値実験では、粒子の位置は乱数で決める。各粒子について接触判定計算を行う。この接触判定計算には、接触する可能性のある粒子を判定するために、接触判定格子を用いる。

接触判定格子では、計算領域を格子状に分割する。その後、格子内に入っている全粒子を、対応する判定格子上に登録する。格子幅を粒子直径と同じにすることにより、接触する可能性のある粒子を同一格子内と隣接格子内に存在する粒子に限定することができる。そのため、計算量を減らすことが可能となる。

#### ● 逐次計算の演算パターン

接触する粒子に対する接触力を更新するとき、自分自身の情報だけでなく、相手の情報も同時に更新する。これは、作用-反作用の法則を用いて演算量を 1/2 に削減するための処理となる。図 1 にプログラムの一部分を記載する。

図 1 では、 $n$  は接触力を計算している粒子番号、 $n1s \sim n1e$  は粒子番号  $n$  が入っている格子に存在す

る粒子群についての最初の粒子番号から最後の粒子番号、 $f_{xyz}$  は総接触力を収納している配列、そして  $f_x$ ,  $f_y$ ,  $f_z$  は粒子番号  $n$  に働く接触力を示している。図 1 からわかるように、自分の接触力である図 1<4>~<6>の  $f_{xyz}(*, n)$  の更新と、相手の接触力である図 1<7>~<9> $f_{xyz}(*, n1)$  の情報を更新する。

```

<1>do n=ns, ne
<2>  do n1=n1s, n1e
<3>  ...
<4>    fxyz(1,n)=fxyz(1,n)+fx
<5>    fxyz(2,n)=fxyz(2,n)+fy
<6>    fxyz(3,n)=fxyz(3,n)+fz
<7>    fxyz(1,n1)=fxyz(1,n1)-fx
<8>    fxyz(2,n1)=fxyz(2,n1)-fy
<9>    fxyz(3,n1)=fxyz(3,n1)-fz
<10>  ...
<11>end do
<12>end do
    
```

図 1 接触判定計算における競合

図 1 の処理について、図 1<1>のループ  $n$  を OpenMP 並列化する場合を考える。このとき、 $n$  について、粒子番号 0 番と粒子番号 1 番の粒子が、同一格子内にあるとする。また、粒子番号 0 番の粒子はスレッド 0、粒子番号 1 番の粒子はスレッド 1 で処理されるとする。

ここで、粒子番号 0 番の粒子の図 1 での処理は、 $n=0$  および  $n1=1$  となる。このとき、スレッド 0 において、 $f_{xyz}(*, n1)$  の更新（ここでの  $n1$  は 1）を行っている最中に、スレッド 1 では、 $f_{xyz}(*, n)$ （ここでの  $n$  は 1）の更新を行う状況がありうる。すなわち、スレッド 0 とスレッド 1 の間で、 $f_{xyz}(*, n)$  と  $f_{xyz}(*, n1)$  が同じ配列の要素を指しており、同時に値を更新しようとするタイミングがありうる。この状態では、スレッド 0 もしくはスレッド 1 の結果のみが反映され、逐次の結果と一致しなくなる。したがって、図 1 の処理は、排

他制御が必要である。

以降、図 1 の衝突判定計算において、スレッド間で値を同時に書き込むことを「競合」と呼ぶことにする。

### ● 競合の回避方法

図 1 をスレッド並列化した場合の競合について前節で説明した。ここでは、この競合の問題を回避する実装について述べる。従来、図 1 の競合を回避するため、OpenMP で提供される critical 指定子を用いたディレクティブである critical 文や atomic 文により排他制御を行うことで、この競合問題を制御してきた。

また、critical 文を使わない方法として、図 1 の演算について、接触時に相手の情報を更新しないようにすることで、競合を回避してきた。しかしながら、この方法では、接触力計算に関して冗長計算をしなくてはならない。その結果、演算量が 2 倍に増加する。

以上で説明した 2 つの実装のプログラムを、図 2 に示す。

図 2 <3>~<11>において、critical 文を使う従来法では、対象範囲を critical 文で囲み排他制御を行う。図 2<12>~<15>の冗長計算をする方法では、相手の接触力の値を更新しないため、 $f_{xyz}(*, n1)$ の式が存在しない。一方、図 2<16>~<23>では、本研究による提案法であるマルチカラー接触判定法での計算になる。マルチカラー接触判定法では、相手の接触力の更新である  $f_{xyz}(*, n1)$ の式があるが、critical 文は存在しない。

### ● マルチカラー接触判定格子

図 2 のプログラムを利用するため、DEM における接触判定格子において、同時に計算できる格子に同じ色を付ける。同じ色の接触判定格子単位の計算は、並列処理で実行可能である。また、格子内の処理は逐次で行うことを保証するものとする。

本稿で扱う DEM では、接触判定格子のサイズを粒子径と同程度にとると、注目する格子の近傍のみにデータ依存関係が生じる。したがって、近傍

の格子が異なるように接触判定格子を色づける。この色づけられた格子をマルチカラー接触判定格子と呼ぶ。

```

<1> do n1=n1s,n1e
<2>  ...
<3>  if( critical 文を使う実装 ) then
<4> !$omp critical
<5>   fxyz(1,n)=fxyz(1,n)+fx
<6>   fxyz(2,n)=fxyz(2,n)+fy
<7>   fxyz(3,n)=fxyz(3,n)+fz
<8>   fxyz(1,n1)=fxyz(1,n1)-fx
<9>   fxyz(2,n1)=fxyz(2,n1)-fy
<10>  fxyz(3,n1)=fxyz(3,n1)-fz
<11> !$omp end critical
<12> else if( 冗長計算を使う実装) then
<13>  fxyz(1,n)=fxyz(1,n)+fx
<14>  fxyz(2,n)=fxyz(2,n)+fy
<15>  fxyz(3,n)=fxyz(3,n)+fz
<16> else if(マルチカラー接触判定法) then
<17>  fxyz(1,n)=fxyz(1,n)+fx
<18>  fxyz(2,n)=fxyz(2,n)+fy
<19>  fxyz(3,n)=fxyz(3,n)+fz
<20>  fxyz(1,n1)=fxyz(1,n1)-fx
<21>  fxyz(2,n1)=fxyz(2,n1)-fy
<22>  fxyz(3,n1)=fxyz(3,n1)-fz
<23> end if
<24>  ...
<25>end do
    
```

図 2 接触判定計算の競合を回避するプログラム

図 3 に、3 次元の接触判定格子において、マルチカラー接触判定格子を実現した例を示す。

図 3 から、3 次元の接触判定格子では、少なくとも 8 色の色分けで、マルチカラー接触判定格子を作ればよいことがわかる。

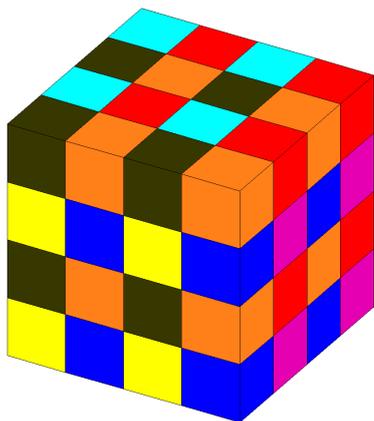


図 3 マルチカラー接触判定格子の例  
(3次元の接触判定格子の場合)

### 性能評価

#### ● 計算環境と初期設定 (OpenMP)

本性能評価では、東京大学に 2012 年に導入された FX10 スーパーコンピュータシステム（富士通 PRIMEHPC FX10, 以降 FX10 と呼ぶ）の 1 ノードを用いて行った。1 ノード内のコア数は 16 である。そのため、OpenMP による並列化は最大の 16 スレッドまで行う。

今回の性能評価では、粒子径は 0.002、粒子数は 6250 万、ステップ数は 20 に設定した。判定時に各粒子は接触数の情報を更新するが、3 次元の接触判定格子を利用する。そのため、1 粒子あたりの接触数情報の上限は 12 とする。

提案手法の有効性を見るため、表 1 のように 5 つのプログラムを作成した。

表 1. Method による実装方式の違い

Method	実装方式
Method 1	粒子並列 (排他制御)
Method 2	粒子並列 (冗長計算)
Method 3	接触判定格子並列 (排他制御)
Method 4	接触判定格子並列 (冗長計算)
Method 5 (提案法)	マルチカラー接触判定法 (マルチカラー)

ここで、粒子番号で最外ループを回し、その最外ループを並列化する方法を**粒子並列**と呼ぶ。一方、接触判定格子番号で最外ループを回し、その

最外ループを並列化する方法を**接触判定格子並列**と呼ぶ。

この時、表 1 では、並列化時のデータの競合を防ぐため critical 文を入れた粒子並列のプログラムを method1 とする。粒子並列において、相手の情報を書き込まず、計算量が 2 倍になるプログラムを method2 とする。また、接触判定格子並列化時のデータの競合を防ぐため、critical 文を入れたプログラムを method3 とする。接触判定格子並列において、相手の情報書き込まず、計算量が 2 倍になるプログラムを method4 とする。以上の method1~method4 は従来法である。

最後に、今回の提案手法であるマルチカラー接触判定格子を用いて並列化するプログラムを method5 とする。

#### ● 接触判定計算の性能評価 (OpenMP)

各 method において、逐次計算と並列計算 (スレッド数 2、4、8、および 16) において、計算時間の比較を行う。ここで計算時間とは、5 ステップ目以降の計算時間を測定し、その値を平均したものである。計算時間の結果を表 2 に示す。

表 2 では、method2、method4 では排他制御を用いていない理由から、method1、method3 のような性能の劣化が見られない。また、排他制御や冗長計算を行わないマルチカラー接触判定法を用いた method5 は、従来法である method1~method4 の手法よりも良い台数効果が確認できる。

表 2. 各 method の実行時間 [秒] (FX10)

THREADS	Method 1	Method 2	Method 3	Method 4	Method 5 (提案法)
逐次	81.46	58.24	34.13	56.00	36.38
2	57.22	58.74	56.49	58.44	37.36
4	48.48	39.90	47.82	39.04	24.02
8	54.62	31.79	54.48	30.59	18.15
16	81.09	29.15	81.33	27.87	16.14

表 2 から、16 スレッド時において、提案法を利用することで最大で 5.03 倍 (method3 の 81.33 [秒] に対する、method5 の 16.14 [秒]) の速度向上が得られることがわかる。

● 接触判定計算の性能評価 (ハイブリッド MPI-OpenMP)

ここでは、マルチカラー接触判定法をハイブリッド MPI-OpenMP 化をした場合の演算性能について評価する。なお本プロトタイプでは、MPI 並列化を行っているが、通信の最適化をしていない。接触判定格子のデータ依存関係を利用することで、各 MPI プロセスの隣接 MPI プロセスのみ通信をするだけで並列化が可能である。しかし、ここでは集団通信を利用した実装を行った。そのため、通信時間が理想的な実装に対して大きくなる。そこで、通信時間を評価対象から除く。

粒子間距離は 0.004、粒子密度 100、ばね定数 200、粒子数 7,812,499 の問題を設定した。時間刻みは 0.00025、20 ステップのシミュレーションを行う。自由落下の 3 次元問題であり、粒子の初期状態 (位置) は乱数で決定する。法線方向の力は計算しない。

以上の問題サイズを固定し、FX10 の 1 ノード (16 コア) と 12 コア (192 コア) の実行時間 (4 ステップ~20 ステップまでの衝突判定時間の総和時間) を評価する。すなわち、ストロングスケールでの評価である。

ハイブリッド MPI-OpenMP 実行の性能評価のため、全てのコアを使い切る状況で、MPI プロセス数と OpenMP スレッド数を変化させる。ここで、MPI から派生する OpenMP スレッド数は、1、2、4、8、および 16 とした。ここで、“PXY” という表記を用いる。ここで、X は MPI プロセス数、Y は MPI プロセスから派生する OpenMP スレッド数である。

図 4 に、1 ノードでの結果をのせる。

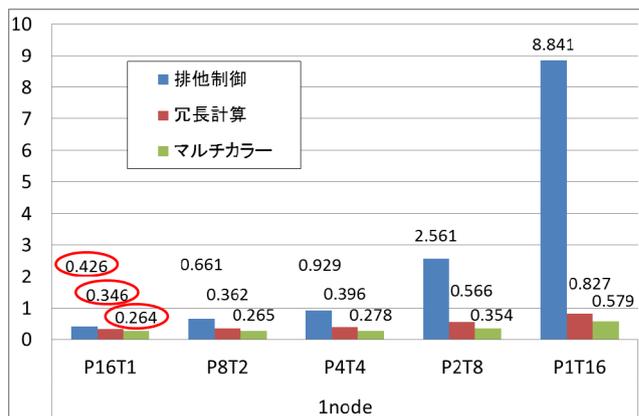


図 4 衝突判定計算の 1 ノードでの実行時間 (FX10)

図 4 から、提案するマルチカラー法が、すべての実行形態において高速となる。また、すべての method において、P16T1 の【ピュア MPI】実行が、ハイブリッド MPI-OpenMP 実行よりも高速である。スレッド数が増えるにつれ、排他制御を用いる方法 (method=3) は、性能が劣化 (実行時間が増加) していく。この傾向は、表 2 の OpenMP の評価と同様であるといえる。

図 5 に、12 ノードを利用した場合の実行時間をのせる。

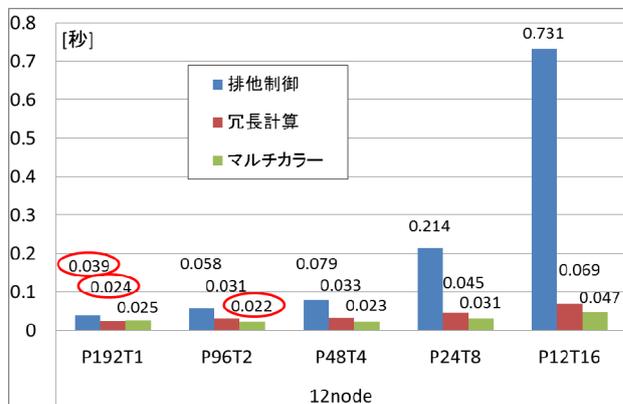


図 5 衝突判定計算の 12 ノードでの実行時間 (FX10)

図 5 から、提案法であるマルチカラー法がすべての実行形態で高速であることは変わらない。興味深いのは、スレッド数を増やしたときに最速となる実行形態が、1 ノードの時と変わることである。図 5 では、マルチカラー法以外では、P192T1 のピュア MPI 実行が最も高速であったが、提案法では P96T2 のハイブリッド MPI-OpenMP 実行が高速となった。この理由は、マルチカラー法が従来法

に比べて OpenMP 実行の並列化効率が高いことから生じたと予想される。

一方、1 ノードでの実行に対する 12 ノードでの台数効果は、最高速なハイブリッド MPI-OpenMP 実行の形態を考慮すると、排他制御、冗長計算、マルチカラーで、それぞれ、10.9 倍、14.4 倍、12.0 倍である。提案法は最高速であるが、台数効果も良好である。

ハイブリッド MPI-OpenMP 実行の目的は、スレッド実行数を増やすことで MPI プロセス数を減少させ、通信時間をピュア MPI 実行よりも削減させることにある。その効果は、超並列実行ほど顕著になる。図 5 の結果は DEM において、エクサスケールに向けた超並列実行でのマルチカラー法の優位性を表しているといえる。

今後の課題として、通信処理を最適化したうえで、全体時間の観点から提案手法を評価することがあげられる。

## 参考文献

[1] Hu, C., Liu, Y. and Li, J.: Efficient Parallel Implementation of Molecular Dynamics with Embedded Atom Method on Multi-core Platforms, International Conference on Parallel Processing Workshop, pp.121-129 (2009)

[2] Evans, D. J.: Parallel S.O.R. Iterative Methods, Parallel Computing, Vol.1, pp.3-18 (1984)

## (2) 当初計画の達成状況について

本計画では、以下の 3 フェーズを通して目的を実現する計画であった。

- **フェーズ I (2013 年 4 月～7 月)**: 東大の FX10 を用いて、単体性能の向上に関する研究を行う。具体的には、FX10 の詳細プロファイラをもちいて、スレッド並列化時のオーバーヘッドや SIMD 演算効率などの解析を行う。その解析を基に、性能改善が可能な実装方式や、また数値計算アルゴリズム上の工夫を行う。

- **フェーズ II (2013 年 8 月～11 月)**: MPI 並列化の設計を行い、また簡単なプロトタイピングを行う。
- **フェーズ III (2013 年 12 月～2014 年 3 月)**: フェーズ II のプロトタイピングをもとに、FX10 の 24 ノードを利用した最大限の並列数において性能評価を行う。詳細プロファイラにより、通信処理や単体性能のボトルネック解析をする。その結果に基づき、性能改善が可能な実装方式や通信アルゴリズムの提案を行う。また、AT 機能を実現する際の性能パラメタの見極めを行う。さらにその性能パラメタにおける AT を実施し、AT 機能の効果を検討する。

本報告において、フェーズ II までが終了していることを報告した。フェーズ III においては、通信最適化を行っていない都合から未完である。しかし DEM において、新規性のある並列化手法の提案を行うことができた。またプロトタイピングの性能評価の結果から、OpenMP およびハイブリッド MPI-OpenMP の並列化効果においても、従来法をしのご性能が実現可能なことを示した。本手法は、DEM において新規性があると判断される。そのため、DEM のシミュレーションの応用先である粉体工学の専門誌に、本手法の論文を投稿済みである。

## 4. 今後の展望

本研究で開発したマルチカラー接触判定格子を用いたマルチカラー接触判定法は、DEM において新規性のある手法と考えられる。また、DEM においてエクサスケールに向けた超並列実行で、さらなる優位性が主張できると予想される。さらには、Intel Xeon Phi に代表される、1 ノードあたり 200 スレッド並列を超える、「メニーコア型」の計算機に特に向く手法と考えられる。なお、提案手法におけるメニーコア型計算機での性能は、初期評価が終了しており、良好な性能が得られることを確認している。したがって、次世代の計算機においても、本手法は有力な方法であると考えられる。

今後の展望として、通信処理の最適化、および、

マルチカラー接触判定格子に向く通信実装方式の開発を行うことがあげられる。その上で、DEM の計算において、1 万並列を超える並列環境で、ハイブリッド MPI-OpenMP 実行の再評価することがあげられる。

以上に加えて、性能チューニングのためのパラメタが増えている。そのため、AT 技術の適用が期待される。マルチカラー接触判定法に向く AT 技術の開発も、今後の重要な課題となるだろう。

## 5. 研究成果リスト

### (1) 学術論文（投稿中のものは「投稿中」と明記）

- 片桐孝洋、竹田宏、加藤淳也、河村祥太、堀端康善：DEM におけるマルチカラー接触判定法、粉体工学学会誌（2014）（投稿中）

### (2) 国際会議プロシーディングス

#### (3) 国際会議発表

- Takahiro Katagiri, Hiroshi Takeda, Jyunya Kato, Shota Kawamura and Yasuyoshi Horibata: Towards Multicolor Particle Contact Detection Method for Hybrid MPI-OpenMP Execution in DEM, In Proceedings of VECPAR2014, A Poster, (2014) (Accepted)

#### (4) 国内会議発表

- 加藤淳也、河村祥太、竹田宏、片桐孝洋、堀端康善：マルチカラー接触判定格子を用いた粒子接触判定計算の OpenMP による並列化、情報処理学会研究報告 2013-HPC-142 (2013)

### (5) その他（特許，プレス発表，著書等）