

マルチカラー接触判定法のハイブリッド MPI/OpenMPによる並列化

河村, 祥太 / KAWAMURA, Shota

(出版者 / Publisher)

法政大学大学院理工学・工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学・工学研究科編

(巻 / Volume)

56

(開始ページ / Start Page)

1

(終了ページ / End Page)

8

(発行年 / Year)

2015-03-24

(URL)

<https://doi.org/10.15002/00011129>

マルチカラー接触判定法の ハイブリッド MPI/OpenMP による並列化

THE PARALLELIZATION OF MULTI-COLORED CONTACT DETERMINATION METHOD
BY HYBRID MPI/OPENMP

河村祥太

Shota KAWAMURA

指導教員 堀端康善

法政大学大学院理工学研究科電気電子工学専攻修士課程

Many of the powder treated in practice are formed from a large number of particles. However, Sequential computation has a limit of analysis of a million particles by the computation time constraints. Therefore, parallelization powder analysis algorithm has become urgent issue. Especially in powder analysis algorithm, ensuring the high parallelism for contact determination calculation part has become an issue. In this study, we perform the MPI parallelization using multi-color contact determination method which increases the thread parallelism of the computation in the node, and we aim to realize a hybrid MPI / OpenMP implementation to achieve high efficiency. As a result, the speed improvement of 435.7 times in 256 node run in a multi-color contact determination method has been confirmed.

Key Words : multi-colored contact determination method, Hybrid MPI/OpenMP

1. はじめに

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（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/OpenMP 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/OpenMP 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索しなくてはならない。

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。

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いけない。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、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とする。

以上から、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。

以上の背景から、DEM においては、特に粒子の接触判定計算において、ノード内で高いスレッド並列性を引き出すことが重要となる。そこで我々は、DEM におけるデータ依存関係を考慮し、接触判定格子に対するマルチカラー法を提案した。この方法を、マルチカラー接触判定法[1][2]と呼ぶ。

本稿では DEM において、マルチカラー接触判定法をノード内のスレッド並列化に用い、ノード間の MPI 並列化を行うハイブリッド MPI/OpenMP 実装の検討を行うことを目的とする。特に本稿では、このハイブリッド MPI/OpenMP 実行を行う場合において、演算効率の観点から、ハイブリッド MPI/OpenMP 実装においてもマルチカラー接触判定法が有効であることを示すことを目的にする。

2. 接触判定計算

(1) 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を図 1 に示す。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る。DEM における粒子の並び替えの有効性は、いくつか報告がされている[3][4][5]。

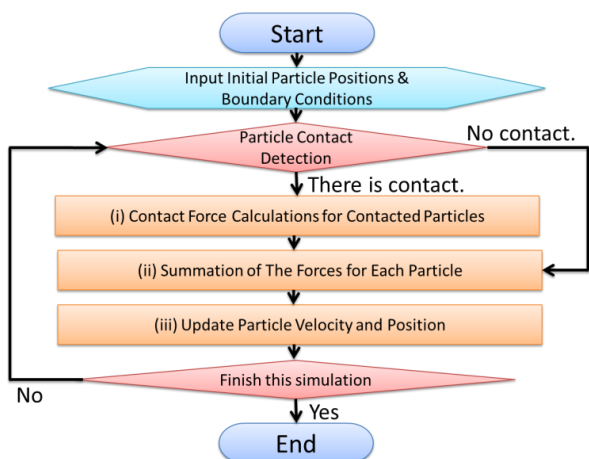


図 1 プログラム全体のフローチャート[6]

(2) 接触判定計算と接触判定格子

接触判定計算には、接触する可能性のある粒子を効率よく判定するために、接触判定格子を用いる。この接触判定格子では、計算領域を格子状に分割する。格子内に入っている全粒子を、対応する判定格子に登録する。図 2 に接触判定格子の例を示す。

接触判定格子の格子幅は、粒子直径と同じ、または粒子直径以上にとる必要がある。本研究では、格子幅

を粒子直径と同じに設定する。接触判定格子を用いることにより、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、接触判定のための計算量を減らすことが可能となるだけでなく、接触判定処理におけるデータ依存関係を緩和することができる。このことは、後述するマルチカラー接触判定法における前提となるため、重要である。

また、均一粒子直径を対象とした DEM では、接触判定格子内に入る粒子数は数個（1～3 個）程度であるため、極端な数の不均衡は起こる事はない。後述するが、このことも、スレッド実行時における負荷不均衡が生じない理由である。

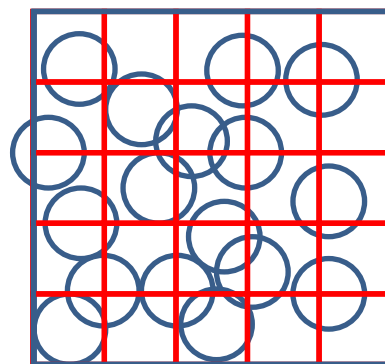


図 2 接触判定格子の例（2次元）

(3) 接触判定計算

いま、接触判定が終了し、接触する粒子に対する接触力の計算を行うことを考える。このとき、自分自身の接触力だけでなく、相手側の接触力も同時に更新しなくてはならない。このことが、並列処理において、排他制御を必要とする要因である。

また一方で、作用-反作用の法則から、相手側の接触力と自分の接触力は、向きが違うだけである。すなわち、いま自分の接触力を F とすると、相手側の接触力は $-F$ となる。したがって、この力の計算においては、作用-反作用の法則を用いて演算量を 1/2 に削減することができる。図 3 にプログラムの一部分を記載する。

ここで図 3 では、粒子 i における 3 次元の接触力が配列 $fxyz(*, i)$ に収納されており、現在計算された接触力の x, y, z 成分を fx, fy, fz としている。ここで、図 3 中の n は、現在注目している粒子番号である。一方、接触している粒子群の番号は $n1$ であり、その番号は $n1s \sim n1e$ である。

図 3 からわかるように、自分の接触力である図 3<4>～<6>の $fxyz(*, n)$ の更新と、相手の接触力である図 3<7>～<9>の $fxyz(*, n1)$ の情報をループ内で更新しているの、図 3 のループを並列実行すると、相手側の配列 $fxyz(*, n1)$ のデータを加算するときに、逐次と結果が異なることがある。そのため、排他制御が必要となる。

図 3 の処理について、図 3<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 3 での処理は， $n=0$ および $n1=1$ に相当する．このとき，スレッド 0 において， $fxyz(*,n1)$ の更新（ここでの $n1$ は 1）を行っている最中に，スレッド 1 において， $fxyz(*,n)$ （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， $fxyz(*,n)$ と $fxyz(*,n1)$ 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 3 の処理は，排他制御が必要である．

以降，図 3 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．

```
<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 3 作用・反作用を用いた接触力の計算

（４）競合の回避方法

図 3 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 3 の競合を回避するには，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競合の問題を制御してきた[7]．このように，OpenMP 等の排他制御を使う方法を，排他制御法とよぶ．

また，`critical` 文を使わない方法として，図 3 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．ただしこの方法では，接触力計算は相手側の接触力の再計算が必要となるため，演算量が 2 倍に増加する．そのため，この方法を冗長計算法とよぶ．

一方，後ほど説明するマルチカラー接触判定法では，接触判定格子のデータ依存関係を利用することで，冗長計算法を用いずに，演算量は排他制御法と同じ量で

実現が可能となる．また，データ依存関係を考慮しているので，排他制御も不要である．

以上の実装のプログラムの概略を図 4 に示す．図 4<3>～<11>では，`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 4<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， $fxyz(*, n1)$ の式が存在しない．一方，図 4<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 $fxyz(*, n1)$ の式があるが，`critical` 文は存在しない．

```
<1> do n1 = n1s, n1e
<2>  ...
<3>  if( 排他制御法 ) 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( 冗長計算法 ) 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 4 接触判定計算の実装概略

3. マルチカラー接触判定法

（１）概要

我々は，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した[2]．DEM で必要となる接触判定計算では，すでに説明したように，作用・反作用の法則により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ため，複数のスレッド間でデータの読み書きが競合する．結果として，排他制御が必要となることを述べた．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とも述べた。

一方で、冗長計算法は、相互排除法に対し約 2 倍の演算量の増大になる。つまり、排他制御法では演算量が $1/2$ 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、冗長計算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、演算量は相互排除法と同じく $1/2$ になるが、排他制御を必要としない方法である、マルチカラー接触判定法を提案した。

(2) マルチカラー接触判定法

まず、マルチカラー接触判定法に用いる接触判定格子を定義する。これを、マルチカラー接触判定格子と呼ぶ。

マルチカラー接触判定格子とは、従来の接触判定計算に用いる接触判定格子を用いるが、接触判定計算のためのアクセスの仕方が異なる接触判定格子のことである。ここではまず、1 次元の場合における例を示すことで、マルチカラー接触判定法のアルゴリズムを説明する。

接触判定格子上のデータ依存を回避するため、2 色に色分けする。接触判定格子への色の付け方を、図 5 に示す。まず、図 5 (a) の接触判定格子を 2 個ずつの単位格子 (1 次元単位格子) にグループ分けする。図 5 (b) は、1 次元単位格子を示す (i は 1 以上の整数)。図 5 (b) のように、1 次元単位格子の左位置 (格子番号 $2i-1$) の格子に対して色“1”を付与する。また、右位置 (格子番号 $2i$) の格子に対して色“2”を付与する。この操作を、すべての 1 次元単位格子に対して、単位格子内の同じ位置の格子に同じ色を付与する。このようにして、図 5 (c) に示したマルチカラー接触判定格子を定義することができる。

図 5 は、1 次元の解析対象領域に対するマルチカラー接触判定格子と格子番号を示している。

図 5 では、従来の接触判定格子のデータ・アクセスパターンは、粒子の接触判定をする理由から、格子につけられた番号 (格子番号) 順にアクセスする。図 5 の例では、従来法では、格子番号順に $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow \dots$ とアクセスしていく。

このとき、DEM での接触判定計算のデータ依存を考慮すると、対象格子番号 i の両隣格子 (隣接格子) である、格子番号 $(i-1)$ と格子番号 $(i+1)$ には依存関係があるが、それ以外の格子には依存関係がない。ここで、依存関係がないとは、データの競合が起き得ないために並列実行が可能であることをいい、依存関係があるとは、データの競合が起き得るために並列実行ができないことをいう。そのため、たとえば、格子番号 i と格子番号 $(i-2)$ と $(i+2)$ の計算は並列実行が可能である。

ここで注意点は、格子内にある粒子は接触しているため、データ依存がある。そのため、格子内にある粒子の接触判定計算は逐次 (スレッドごとに逐次) で行う。また、対象格子番号 i の接触判定を行うとき、格子番号 $(i-1)$ は格子番号 $(i-2)$ ととも接触判定を行うため、接触相手の情報を更新する際、格子番号 $(i-1)$ 内に存在する粒子はバッティングを起こし、正しい結果が得られない可能性がある。しかしながら、冗長計算をする場合を除いては、接触計算が 2 重に行われることを防ぐため、通常、格子番号 $(i-1)$ の格子は接触判定時に参照しないという条件を課すようになっている。したがって、このようなバッティングが起きることもない。

図 5 の例では、並列実行が可能である格子番号が奇数番の接触判定格子 i ($i=1, 3, 5, 7, 9, \dots$) の接触判定計算を、まず並列に行う。その後、格子番号が偶数番の接触判定格子 j ($j=2, 4, 6, 8$) の接触判定計算を並列に行う。以上の 2 工程を交互に繰り返し行う。このように演算するのが、マルチカラー接触判定法である。

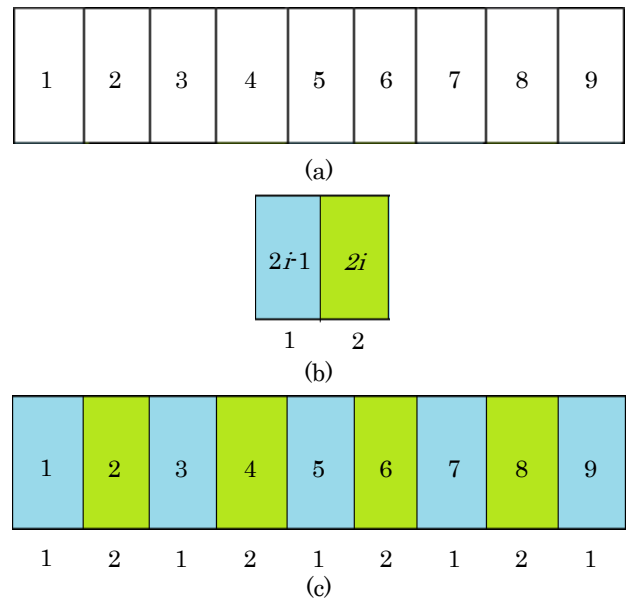


図 5 1 次元でのマルチカラー接触判格子の例

マルチカラー接触判定法の適用の前提は、接触格子の格子幅を粒子直径、あるいはそれ以上にとることである。そうすることで、同じ色の格子内にある粒子同士は接触することがない。そのためマルチカラー接触判定法では、自分と相手の接触力を更新する際に、原理上、競合が起きない。その結果、並列化をしても複数のスレッド間でデータの競合が起きない。

なお、マルチカラー接触判定法では、逐次の接触判定計算とは加減算の順序が異なる。ただし、数学上における演算結果の一致は保証される。

以上は 1 次元の解析対象に対するマルチカラー接触判定法の説明である。2 次元、3 次元に解析対象を広げ

る場合においても、マルチカラー接触判定法は、マルチカラー接触判定格子を2次元, 3次元に拡張することで拡張が可能となる。マルチカラー接触判定法を3次元問題に適用するには、図6のように、3次元にマルチカラー接触判定格子を拡張する。図6では、接触判定格子上のデータ依存を回避するため、8色に色分けする。

図6の8色の色分けを行い、接触判定計算の方法を3次元に拡張する。

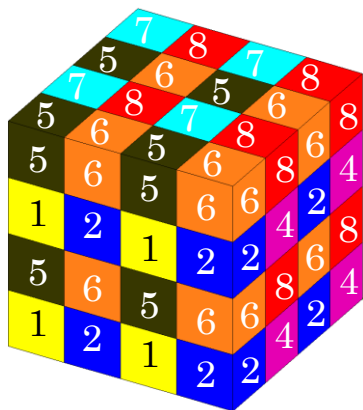


図6 3次元のマルチカラー接触判定格子例

(3) OpenMPの実装

ところで、マルチカラー接触判定法に限らず、従来手法においても、格子番号をそのままループにして回してOpenMP並列化すると、粒子が存在しない接触判定格子を割り当てられるスレッドがあるため、負荷不均衡が生じる。そのため、この実装では、OpenMPでの実行性能が出ない。以下にその例を示す。

```
!$omp parallel do
  do i=1, MAX_NUM_GRIDS
    格子番号 i における接触判定計算
  enddo
!$omp end parallel do
```

なお、MAX_NUM_GRIDSとは、接触判定格子の番号のうち、もっとも大きな格子番号である。

そこで本研究においては、接触判定計算に入る前に、粒子が存在しない接触判定格子の番号を取り除き、接触判定計算用のループの回転数を変更する。

いま、MAX_NONZERO_NUM_GRIDSを、粒子が存在する接触判定格子の数とする。また、CDG(j)を、ループ変数jにおける、対応する元の接触判定格子番号を収納した配列とする。この時、以下のOpenMP並列化を実装する。

```
!$omp parallel do private(i)
  do j=1, MAX_NONZERO_NUM_GRIDS
    i = CDG( j )
    格子番号 i における接触判定計算
  enddo
!$omp end parallel do
```

以上の実装により、ループjをOpenMP並列化する際には、対応する接触判定格子には必ず粒子が存在するため、スレッド間で負荷の不均衡が生じない。またすでに説明したが、メモリ量が許す範囲において、粒子直径と接触判定格子間隔を等しく設定することにより、接触判定格子内の粒子の個数は高々数個となるため、粒子が密集した場合においても、負荷の不均衡が生じない。

またこの実装は、マルチカラー接触判定法に限らず、排他制御法、冗長計算法にも適用できる。

マルチカラー接触判定法では、接触判定格子の色ごとに、接触判定格子を再構築することで、簡単に実装ができる。以下にその例(3次元の場合)を示す。

```
do icolor=0, 7
!$omp parallel do private(i)
  do j=1, MAX_NONZERO_NUM_GRIDS(icolor)
    i = CDG( j, icolor )
    格子番号 i における接触判定計算
  enddo
!$omp end parallel do
enddo
```

ここで、MAX_NONZERO_NUM_GRIDS(icolor)は、接触判定格子の色icolorにおけるマルチカラー接触判定格子のうち、粒子が空でない格子の数である。また、CDG(j, icolor)を、マルチカラー接触判定格子の色icolorにおける、ループ変数jにおける、対応する元の接触判定格子番号を収納した配列とする。以上より、マルチカラー接触判定法においても、スレッド実行時の負荷バランスの劣化が生じない。

(4) MPIの実装

MPI並列化を行うに当たり、以下の3つの方針がある。

a) 通信回数を最少にする方法

衝突判定格子の構造を元に、各MPIプロセスに割り当てる。例えば、3次元の衝突判定格子があるとき、X軸方向を分割した立方体を各MPIプロセスに割り当てることを考える。このとき、通信は2つの隣接プロセスしか生じない。そのため、最大で各MPIプロセス当たり2回の通信になる。

欠点は、各MPIプロセスに割り当てた衝突判定格子内に粒子が偏在するとき、MPI並列での負荷バランス

が悪くなり、並列性能が劣化することである。

b) 負荷バランスを最少にする方法

衝突判定格子の内部にある粒子数を考慮して、各 MPI プロセスに割り当てられる粒子数が均等になるように、各 MPI プロセスに担当となる衝突判定格子を割り当てる。この方法では、MPI 並列での負荷バランスが悪くなり、並列性能が劣化することがない。

しかし通信の観点では、通信回数が最少で 2 回であり、最大の通信回数は保証できない。また、粒子は頻繁に移動するため、割り当てた衝突判定格子を再度割り当て直さないと、負荷バランスの劣化を修正できない。そのため、通信量も増加する。

c) 2 次元にデータ分散する方法

衝突判定格子の構造を元に、各 MPI プロセスに計算領域を割り当てる。例えば、3 次元の衝突判定格子があるとき、X 軸方向、Y 軸方向を分割した領域を各 MPI プロセスに割り当てる。これは、(1) と通信の総量が同じでも通信回数を増やすことで、通信時間を計算時間の裏に隠せる事があり、計算時間を短くする事が可能である。

4. 性能評価

(1) 計算環境

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 256 ノードを用いて行った。1 ノード内のコア数は 16 であるため、OpenMP による並列化は最大の 16 スレッドまで行った。また、256 ノードを用いたので、MPI による並列化は、4096 プロセスまで行い、OpenMP と MPI の各組み合わせでのデータを取得した。

表 1. FX10 構成

項目		仕様
ノード	理論演算性能	236.5GFlops
	プロセッサ数 (コア数)	16
	主記憶容量	32GB
プロセッサ	プロセッサ名	SPARC64 IX-fx
	周波数	1.848GHz
	理論演算性能 (コア)	14.78GFlops
ソフトウェア	OS	Red Hat Enterprise Linux Server 6.1
	コンパイラ	Fujitsu Fortran Compiler

(2) 台数効果について

FX10 の 256 ノードを占有し、OpenMP では 1~16 スレッド、MPI では 1~4096 プロセスで並列実行したときの計算時間と台数効果について性能を評価する。ここで台数効果とは、並列処理の効果を示す一般的な指標である。式(1) のように「1 ノードのビュア MPI 実行時計算時間」に対する「並列実行における実行時間」の比で表される。

$$S = \frac{1 \text{ノードのビュア MPI 実行時計算時間}}{\text{並列計算に要する計算時間}} \quad (1)$$

なお、式 (1) における計算時間とは、接触判定計算に要する計算時間 T_c と粒子の接触力計算時間 T_f を合計した総計算時間 T_{ALL} の事を指す。

(3) 接触判定計算の性能評価

提案手法の有効性を見るために、表 2 のように 3 つのプログラムを用意した。

ここで、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。

接触判定格子並列化時のデータの競合を防ぐために critical 文を入れたプログラムを method1 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 method2 とする。以上の method1~method2 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 method3 とする。また、MPI の実装方法は第 3 章 4 節 b 項を使用した。

表 2. Method による実装方式の違い

Method	実装方式
Method 1	接触判定格子並列
Method 2	接触判定格子並列（冗長計算）
Method 3	マルチカラー接触判定格子並列

粒子直径は 0.0032、粒子数は 15,258,790、ステップ数は 20 に設定した。ま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ため、1 粒子あたりの接触数情報の上限は 12 とする。各実装方式において、逐次計算と並列計算（1~12 ノード、1~192 プロセス、1~16 スレッドの各組み合わせ）での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。各ノードの最高速の計算時間の結果を、以下の表 3 に示す。

表 3. 各実装方式の実行時間[秒]

node	MPIprocess × OMPthread	排他 制御	冗長 計算	マルチ カラー
1node	P16T1	0.893	0.760	0.556
2node	P32T1	0.447	0.374	0.278
4node	P64T1	0.227	0.184	0.144
6node	P96T1	0.155	0.123	0.098
8node	P128T1	0.120	0.091	0.073
10node	P160T1	0.096	0.070	0.060
	P80T2			
12node	P192T1	0.080	0.056	0.049

表 3 より，提案するマルチカラーが，すべての実行形態において高速であることがわかる．また，すべての実装方式において，P192T1 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，排他制御は，性能が劣化（実行時間が増加）していくことがわかる．

また，各実装方式において，各ノードの最高速のハイブリッド MPI/OpenMP での組み合わせの時間に対する，台数効果を以下の表 4 に示す．1 ノードのピュア MPI の実行時間を基準とする．

表 4. 台数効果

node	MPIprocess × OMPthread	排他 制御	冗長 計算	マルチ カラー
1node	P16T1	1.00	1.00	1.00
2node	P32T1	2.00	2.03	2.00
4node	P64T1	3.93	4.13	3.86
6node	P96T1	5.76	6.18	5.67
8node	P128T1	7.44	8.35	7.62
10node	P160T1	9.30	10.86	9.27
	P80T2			
12node	P192T1	11.16	13.57	11.35

排他制御では，高スレッド実行での性能劣化の発生を確認できた．すべてのノード実行時で，16 スレッド実行が最も悪く，排他制御にかかるコストの高さが原因であると考えられる．また，最高速なハイブリッド MPI/OpenMP 実行の形態での 1 ノードでの実行に対する 12 ノードでの台数効果は，排他制御，冗長計算，マルチカラーで，それぞれ 11.2 倍，13.6 倍，11.4 倍であることがわかる．提案法は最高速であるが，台数効果も良好である．

（４）通信処理を最適化したプログラムでの性能評価

第 3 節で最も性能が良かったマルチカラーについて性能評価を行った．また，第 3 節で使ったプログラムの通信処理を最適化し，実行時間の短縮を図った．プログラムの変更点は 2 点ある．1 つ目は MPI の実装方法を第 3 章 4 節 b 項から c 項への変更．2 つ目は通信サブルーチンを集団通信から，1 対 1 通信（ノンブロッキング）への変更である．

粒子直径は 0.0032，粒子数は 18,222,800，ステップ数は 50 に設定した．並列計算（1~256 ノード，1~4096 プロセス，1~16 スレッドの各組み合わせ）での計算時間の比較を行う．ここで計算時間とは，10 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．各ノードの最高速の実行時間の結果を，以下の表 5 に示す．また，実行時間のグラフを図 7 に，台数効果を図 8 に示す．

表 5. マルチカラー接触判定法の性能評価

node	MPIprocess × OMPthread	実行時間	台数効果
1node	P16T1	4.0957	1.000
2node	P16T2	2.2301	1.837
4node	P64T1	0.7367	5.560
8node	P64T2	0.4002	10.234
16node	P256T1	0.1553	26.373
32node	P256T2	0.0877	46.701
36node	P144T4	0.1211	33.821
64node	P1024T1	0.0289	141.720
72node	P576T2	0.0431	95.028
100node	P1600T1	0.0178	230.096
128node	P1024T2	0.0174	235.385
144node	P2304T1	0.0232	176.539
196node	P3136T1	0.0084	487.583
200node	P1600T2	0.0107	382.776
256node	P4096T1	0.0094	435.713

表 5 より，多ノード実行で，性能の飽和の発生を確認できた．また，256 ノードでの台数効果は，435.7 倍であり，良好な結果である．

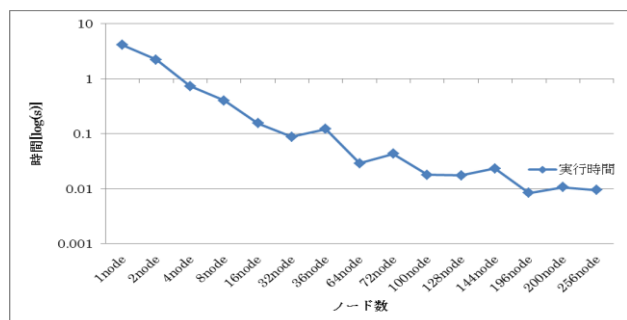


図 7 通信処理を最適化したプログラムでの実行時間

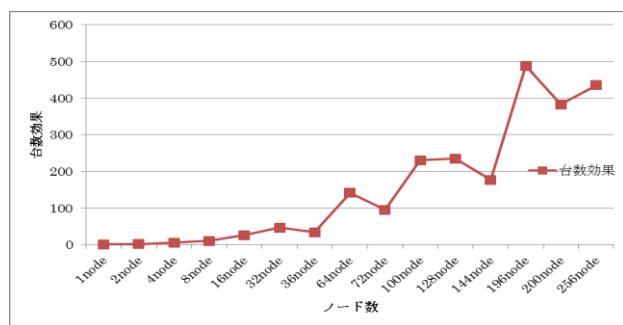


図8 通信処理を最適化したプログラムでの台数効果

5. 結論

本研究は、マルチカラー接触判定法をハイブリッド MPI/OpenMP 化し、従来法に対する性能評価を行った。

我々の先行研究[4]では格子並列を採用しているが、計算の効率化のために、粒子番号のリナンバリングを行っている。また、我々の先行研究[1]では、このリナンバリングに加え、格子のアクセスの仕方を工夫することで、スレッド並列化時にデータ競合が起らないようにしている。

今回の性能評価では、マルチカラー接触判定法はすべての結果で他の手法よりも計算時間が短く、優れていた。

また、ハイブリッド MPI/OpenMP による多ノード実行では、台数効果がスーパーリニアとなり、良好な台数効果となっている。今回の性能評価はストロングスケールで行っており、多ノード実行で扱うには小さすぎる問題である。そのため、多ノード実行では、扱うデータ量が少なくなり、キャッシュにヒットしやすくなることで良好な性能が得られたと考えられる。

今回の性能評価では、マルチカラー接触判定法は従来法より優れていた。また、多ノード実行では優れた台数効果を得ることができた。しかし各ノードとも、高スレッド実行での性能が劣化していた。これは、接触判定処理部分以外のプログラムで、並列化の最適化が実行されていないためであると考えられる。プログラム全体で、並列化の最適化が今後の課題である。

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。

参考文献

- 1) 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，第 199 回 ARC・第 142 回 HPC 合同研究発表会 (2013)
- 2) 片桐孝洋，竹田宏，河村祥太，加藤淳也，堀端康善：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，粉体工学会誌，第 51 巻 8 号，pp.564-570 (2014)
- 3) Yusuke Shigeto, Mikio Sakai : Parallel Computing in Computational Granular Dynamics by Using Multi-Core Processors -Practical Usage of the DEM in Complicated Powder Systems in Industries-, J. Soc. Powder Technol, Japan, 47, pp.707-716 (2010)
- 4) 和田直樹，高木翔，岡大樹，竹田宏，片桐孝洋，堀端康善：粒子接触判定計算の OpenMP による最適化，情報処理学会研究報告，Vol.2012-HPC-136, No.3(2012)
- 5) 高木翔，和田直樹，岡大樹，竹田宏，片桐孝洋，堀端康善：粒子分布を考慮した粒子接触判定計算の MPI および OpenMP による並列化，vol.2012-HPC-137-34(2012)
- 6) Sakaguchi, H. and Nishiura, D.: "Development of Hyper Intelligent Discrete Element Method (HiDEM) and its Application for Science and Industry", JAMSTEC-R IFREE Special Issue, 201-210 (2009)
- 7) 酒井幹夫編著：粉体の数値シミュレーション，丸善出版 (2012)