

GPUのキャッシュヒット率向上によるDEMの高速化

松井南実† 富永浩文† 中村あすか† 篠塚研太† 前川仁孝†

†千葉工業大学 情報科学部情報工学科

1 はじめに

DEM(Distinct Element Method)は、粒子の挙動を運動方程式に基づいて解析する手法であり、土木分野で土石流や液状化などのシミュレーションに用いられている。本手法は、粒子数が増えると粒子同士の衝突計算の増加により実行時間が長くなるため、GPUを用いた並列計算による高速化の研究が行われている[1]。GPUを用いたDEMでは、接触判定と相互作用力の計算を各粒子に対して行うため、メモリアクセスの負荷が大きい。このため、粒子のソートを行いGPUのメモリへのアクセスをコアレスにすることで、メモリアクセスの負荷を軽減する手法が提案されている[2][3]。これらの手法は、L1キャッシュを持つGPUを用いキャッシュ最適化を行うことで更に高速化が図れる。そこで本稿では、L1キャッシュを搭載したGPUであるFermiアーキテクチャを用いて、参照するデータが多く重複する粒子の計算を同一SM(Streaming Multi-processor)内に割り当てることで、GPUのキャッシュヒット率を上げ、DEMを高速化する手法を提案する。

2 GPUによるDEMシミュレーション

DEMは、接触粒子計算によって粒子同士の相互作用力を求め、速度と座標の計算を繰り返すことにより、粒子の挙動をシミュレーションする手法である。DEMにおける粒子の相互作用力は、接触している粒子同士にのみ働く。このため、接触粒子計算は、図1のように解析領域を格子で区切ることによって、計算する粒子*i*の周囲9格子内にある粒子に対してのみ接触判定を行い、不要な計算を削減する。図1中の各処理は、全粒子の計算が独立しているため、各粒子に対する計算を並列に行うことができる。

粒子計算が独立しているという特徴から、GPUを用いた[3]の手法では、1つの粒子計算に対して1スレッドをGPUのSP(Streaming Processor)に割り当て、並列に計算する。GPUの各SM内にキャッシュメモリが存在しないGPUでは、参照方向が格子の横方向に連続するように粒子番号をソートし、コアレスアクセスを行うことで、メモリアクセスの負荷を軽減する。

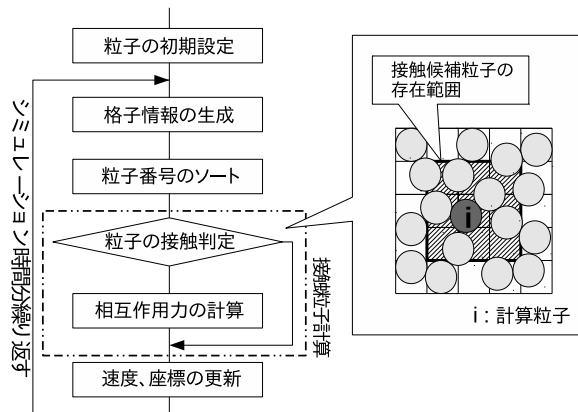


図1: DEMの流れ

3 キャッシュヒット率を向上させる手法

本稿では、Fermiアーキテクチャに搭載されているL1キャッシュのグローバルメモリに対するキャッシュヒット率を上げることで、接触粒子計算を高速に行う。Fermiアーキテクチャは、各SMがL1キャッシュを搭載しており、ブロック内の全てのスレッドでL1キャッシュを共有する。このため、ある粒子の接触粒子計算に必要なデータを、同一ブロック内のスレッド間で共有することで、各スレッドが粒子データを参照したときにキャッシュメモリ上のデータを用いることができる。

接触粒子計算に必要なデータをGPUのL1キャッシュから多く参照できるようにするために、解析領域内で近くにある粒子を同一ブロック内のスレッドで計算する。図2に、相互作用力が影響する粒子が存在する範囲を示す。図2中の斜線は領域内に存在する各粒子の接触判定を行う9格子範囲である。このため、図3の太線のように、周囲9格子が重複するような粒子をブロックにマッピングすることで、同一ブロック内のスレッドが空間的局所性の高い粒子データのアクセスを行うことができる。GPUの各ブロックに粒子を割り当てる際、どのブロックがどの粒子を計算するか指定する必要がある。このため、ブロックがどの粒子を参照するかという情報を作成し、接触判定のカーネル起動後、図3に示す例のように、その情報を用いて各ブロックに粒子を割り当て、ブロック内のスレッドで割り当

†Minami MATSUI †Hirobumi TOMINAGA †Asuka NAKAMURA
†Kenta SHINOZUKA †Yoshitaka MAEKAWA
†Department of Computer Science, Chiba Institute of Technology

てられた粒子を計算する。本手法により、1つの粒子に対して接触粒子判定と相互作用力の計算を多く行うような解析領域の場合、各ブロック内に割り当てられた粒子同士で共有するデータが多くなるため、キャッシュヒット率が向上し、高速化できると考えられる。

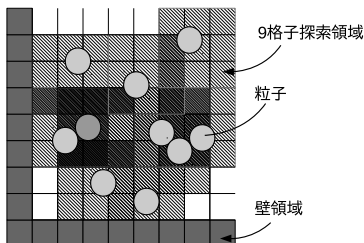


図2: 接触判定を行う9格子の重複範囲

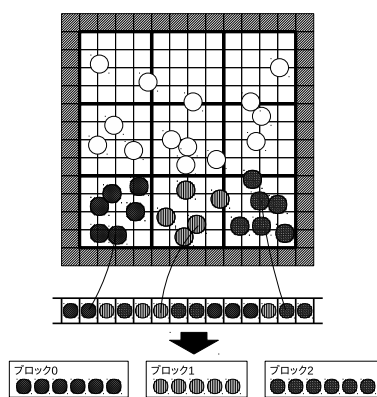


図3: 粒子データのメモリ配置

4 評価

領域分割によってキャッシュヒット率を向上させる手法の有効性を評価するために、領域分割を行う場合と行わない場合でDEMの実行時間を測定する。評価には、粒子数が2500, 4900, 6400, 8100, 10000の落下運動問題を用いる。評価で使用する環境は、CPUがAMD Phenom II X4 965, GPUがGeForce GTX 580, CUDA version4.0を用いる。表1に、実行ステップが6000回のDEMシミュレーションの実行時間を示す。表1より、キャッシュヒット率を上げるような領域分割を行う手法は、行わない場合と比較して実行時間が約2%高速化することが確認できた。

次に、粒子の挙動によって、接触粒子計算の時間がどのように変化するかを確認するために、ステップ時間毎の接触粒子計算の実行時間を測定する。図4に、粒子数を10000個としたときのシミュレーション時間毎の接触粒子計算の処理時間を示す。図4より、シミュレーション時間の前半では、両手法の実行時間に差異が見られないが、シミュレーション時間の後半では、提案手法の実行時間が最大で約16.1%短縮することが確認できる。これは、シミュレーション時間の後半に、粒

子同士の接触が多く起こり、同一ブロック内のスレッド間で共有するデータが多くなったためである。これにより、キャッシュヒット率が向上し、処理時間が高速化したと思われる。

表1: 実行時間の比較

| 粒子数 | 分割無 [s] | 分割有 [s] |
|-------|---------|---------|
| 2500 | 11.112 | 10.815 |
| 4900 | 23.573 | 23.378 |
| 6400 | 32.777 | 32.574 |
| 8100 | 40.652 | 40.252 |
| 10000 | 59.480 | 58.345 |

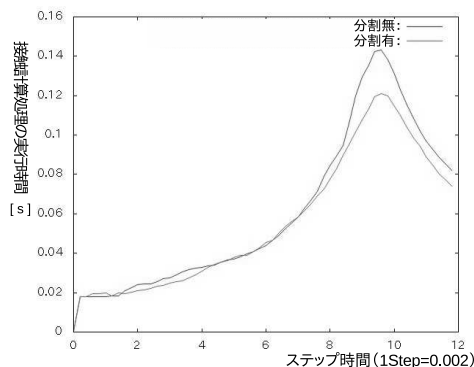


図4: 接触粒子判定計算の実行時間

5 おわりに

本稿では、GPUを用いたDEMの処理時間を、粒子計算時に空間的局所性のあるアクセスを行い、キャッシュヒット率を上げることによって、高速化する手法を提案した。評価の結果、接触粒子計算において、提案手法は、分割した領域内で計算する粒子が多くなる程キャッシュヒット率が上がり、実行時間が高速化することが確認できた。

参考文献

- [1] 原田隆宏, 田中正幸, 越塚誠一, 河口洋一郎: グラフィックスハードウェアを用いた個別要素法の高速化, 日本計算工学会論文集, Vol.2007, No.20070011(2007).
- [2] 原田隆宏, 政家一誠, 越塚誠一, 河口洋一郎: GPU上での粒子法シミュレーションの空間的局所性を用いた高速化, 日本計算工学会論文集, Vol.2008, No.20080016(2008).
- [3] 西浦泰介, 阪口秀: GPUを用いたDEMの高速化アルゴリズム, 日本計算工学会論文集, Vol.2010, No.20100007(2010).