

粒子法陽解法における AVX-512 命令を用いた ループ/SIMD 並列処理

Loop/SIMD Parallelization Using AVX-512 Instruction for MPS Method

富岡 我空†
Gaku Tomioka

齊藤 大輔†
Daisuke Saito

吉田 明正†
Akimasa Yoshida

1 はじめに

流体シミュレーションに用いられる粒子法 [1][2] は、津波シミュレーション [3] のような大規模シミュレーションにも用いることが可能であるが、莫大な計算を必要としており並列処理による高速化が期待されている。本稿では、非圧縮性流体を解析する粒子法 (MPS 法) の陽解法を対象とし、その高速化を目指す。陽解法では、圧力勾配項などを陽的な計算により求めるため計算量は少ないが、安定した解を得るためにタイムステップ t を小さくする必要があり、計算回数が増えてしまう [4]。粒子法の高速度は多くの研究 [5][6] が行われてきたが、粒子法陽解法においてデータ再配置を伴う SIMD 並列処理は提案されていない。そこで、本稿では OpenMP によるループ並列処理と AVX-512 命令による SIMD 並列処理を用いて高速化を実現する。また、SIMD 演算を適用するためにデータレイアウト変換と粒子データの再配置を行った。性能評価では、Intel Xeon Platinum 8358 上で 41,352 粒子の 3 次元水柱崩壊のシミュレーションを行い、提案手法の有効性を確認する。

2 ループ/SIMD 並列処理を用いた高速化

粒子法のタイムステップループ内での各処理に、ループ並列処理および SIMD 並列処理を適用する。各処理においては、並列化可能な 2 重の for ループが存在し、外側のループでは OpenMP によるループ並列処理を適用し、内側のループでは AVX-512 を用いた SIMD 並列処理を適用することにより、階層的な並列処理を実現する。

2.1 OpenMP を用いたループ並列処理

並列化可能な 2 重の for ループの外側のループに対して、OpenMP によるループ並列処理を実現する。指示文である `#pragma omp parallel for` を使用して、全ての並列化可能な for ループに対して並列化を行った。また、スレッドによる処理の不均衡が発生する関数には、動的にイタレーションの割り当てのために指示句として `schedule(dynamic, 64)` を使用した。

2.2 AVX-512 命令を用いた SIMD 並列処理

並列化可能な 2 重の for ループの外側のループに対して、Intel AVX-512 命令を用いて SIMD 並列処理を行う。512bit 長のレジスタにより、64bit 浮動小数点数を 8 要素同時に演算することができる。粒子法コードに AVX-512 intrinsics [7] を使用して SIMD 並列処理を適用する。条件分岐が必要な関数には、マスク演算を用いて

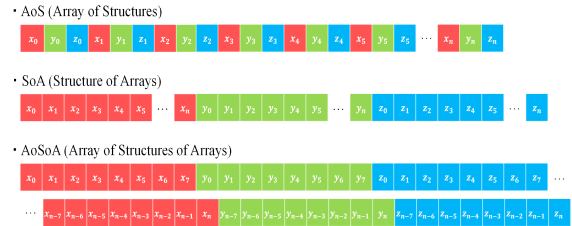


図 1 AoS/SoA/AoSoA データレイアウト。

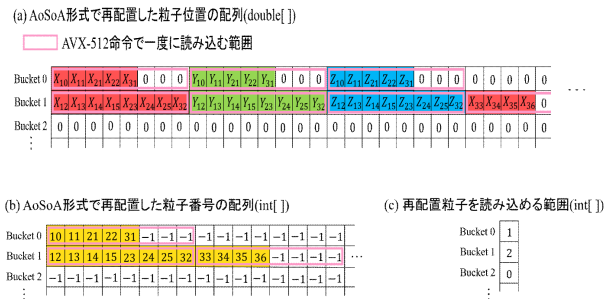


図 2 再配置のデータ構造。

条件に合わない要素を破棄し、条件に適合する要素のみを演算に使用する。

3 粒子の空間情報に対するデータレイアウト変換による高速化

SIMD 演算を行う際にメモリアクセスを向上させるため、データレイアウトの変換、近傍粒子探索におけるデータ再配置を適用する。

3.1 AoSoA 形式によるデータレイアウト

一般的な粒子法コードでは図 1 のような AoS 形式により粒子の空間情報を保存する。SIMD 演算を行う際には要素が連続する形式に変換する必要がある。本研究では x 座標の 8 要素、 y 座標の 8 要素、 z 座標の 8 要素を単位とする AoSoA 形式を採用した。SIMD 演算では 8 要素ずつアクセスすることができるためアクセス速度が向上する。

3.2 バケット法におけるデータ再配置の適用

粒子法コードにおいて AoSoA 形式によって粒子の空間情報を保持する場合、SIMD 演算を行わなわい時は

† 明治大学総合数理学部ネットワークデザイン学科
Department of Network Design, School of Interdisciplinary
Mathematical Sciences, Meiji University

```

1 for(int i=0;i<nP;i++){
2   if(Typ[i] == GST)continue;
3   //粒子iの所属するバケット番号を求める
4   int ix = (int)((Pos[i]*3 ] - MIN_X)*DBinv) +1;
5   int iy = (int)((Pos[i]*3+1] - MIN_Y)*DBinv) +1;
6   int iz = (int)((Pos[i]*3+2] - MIN_Z)*DBinv) +1;
7   int ib = iz*nBxy + iy*nBx + ix;
8
9   int bm=bktmax[ib];
10  bktnum[ib*100+bm]=i;
11  //位置,速度をAoSoA形式で配置する
12  bktpos[ib*64+bm/8*24+bm%8] = Pos[i]*3;
13  bktpos[ib*64+bm/8*24+bm%8+8] = Pos[i]*3+1;
14  bktpos[ib*64+bm/8*24+bm%8+16]=Pos[i]*3+2;
15  bktvel[ib*64+bm/8*24+bm%8] = Vel[i]*3;
16  bktvel[ib*64+bm/8*24+bm%8+8] = Vel[i]*3+1;
17  bktvel[ib*64+bm/8*24+bm%8+16]=Vel[i]*3+2;
18  bktmax[ib]++;
}

```

図 3 再配置を行うコード .

表 1 性能評価マシン構成 .

マシン	Intel Xeon Platinum (Ice Lake)
プロセッサ	Intel Xeon Platinum 8358
CPU コア	32cores, 2.6GHz
メモリ	256GB
OS	Ubuntu 20.04.4

メモリアクセスが非連続になりアクセス速度は低下する . また、SIMD 演算を適用する全ての関数はバケット法を用いて近傍粒子探索を行うため、通常は粒子のデータが連続に並んでいない . そこで、バケット内にある粒子の空間情報に対して、AoSoA 形式によるデータ再配置を導入する . AoSoA 形式による再配置の手法を図 2 に示す . バケットにある粒子に対して、バケット番号に対応した図 2 の (a) の位置に粒子データ、(b) に粒子番号、(c) にバケット内の粒子数を保存する . (a) では粒子データの xyz 成分を 8 要素単位の AoSoA 形式で保存する . 粒子データがないメモリの箇所には (a) では 0、(b) では -1 を置いておく .

次に再配置を行うコードを図 3 に示す . 4 ~ 7 行目では粒子のバケット番号を求める . 10 行目では粒子番号を図 2 の (b) に格納する . 12 ~ 17 行目では粒子の位置と速度データを格納する . AoSoA 形式による再配置を行ったことにより、SIMD 演算以外の処理では AoS 形式を使用し、SIMD 演算を行う近傍粒子との計算は AoSoA 形式を使用するためアクセス速度が向上する .

4 Xeon 搭載サーバー上での性能評価

性能評価では、Intel Xeon Platinum8358 を用いて粒子法陽解法の性能評価を行った .

4.1 性能評価環境

評価プログラムは粒子数 41,352 においてタイムステップを 0.0005[s] として 3 次元の水柱崩壊のシミュレーションを行った . 評価に用いたマシンの構成は表 1 に示す .

4.2 OpenMP/AVX-512 命令を用いた階層的並列処理の性能評価

OpenMP によるループ並列処理と AoSoA 形式による再配置を行い AVX-512 を用いた SIMD 並列処理を適用した手法と再配置を行わずループ並列処理のみを用いた手法による実行結果を図 4 に示す . まず、再配置を用

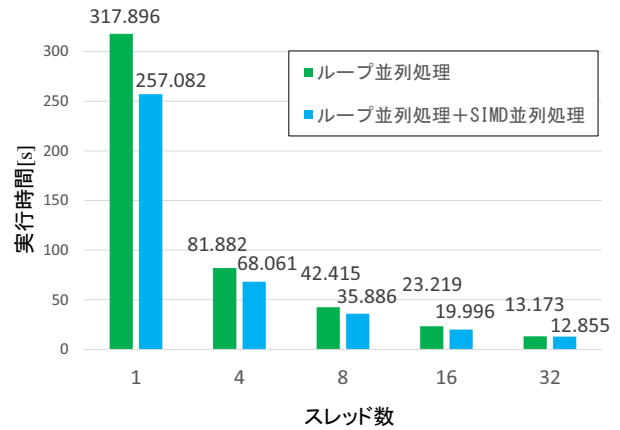


図 4 AoSoA 形式による再配置を用いた手法の性能評価 .

いたシングルスレッド実行の実行時間は 257.08[s] であり、ループ並列処理のみを用いた手法は 32 スレッドでの実行時間は 12.855[s] となり、シングルスレッド実行と比較して速度向上は 24.1 倍であった . AoSoA 形式による再配置を用いた手法では、32 スレッドでの実行時間は 13.173[s] となり、シングルスレッドでの実行時間 317.89[s] と比較して速度向上は 24.7 倍であった . これらの結果から、AoSoA 形式による再配置を用いたループ/AVX512 命令による並列処理の有効性が確認された .

5 おわりに

本稿では、陽解法による粒子法に対して OpenMP によるループ並列処理と AVX-512 命令による SIMD 並列処理を行った . 本手法では近傍粒子計算に SIMD 演算に適用するために AoSoA 形式による再配置を行った . Intel Xeon Platinum8358 にて 3 次元水柱崩壊のシミュレーションを用いた性能評価では、提案手法は再配置を使用しない実行時間と比較して 32 スレッドの実行時間は 24.7 倍の速度向上が得られ、本手法の有効性が確認された .

参考文献

- [1] 越塚誠一, 柴田和也, 室谷浩平 . 粒子法入門 流体シミュレーションの基礎から並列計算と可視化まで C/C++ソースコード付, 丸善出版, 2014.
- [2] プロメテック・ソフトウェア株式会社 . 粒子法 CFD ソフトウェア Particleworks の最近の産業応用事例, https://www.pcluster.org/ja/event/data/ws202109107/pccc-oss_tokura.pdf, 2021.
- [3] Kohei Murotani. Large Scale Numerical Simulation Reproducing of Tsunami Behavior against a Station Building, QR of RTRI, Vol.60, No.1, 2019.
- [4] 松永 拓也, 越塚 誠一 . 粒子法における時間進行法の改良, 日本機械学会論文集, 87 巻 895 号, 2021.
- [5] 齊藤大輔, 吉田明正 . 粒子法の近傍粒子探索における AoSoA 再配置を用いた SIMD 高速化, 情報処理学会研究報告, 2023-HPC-190-10, 2023 .
- [6] M N A Alamsyah, A Utomo and P H Gunawan Analysis OpenMP performance of AMD and Intel architecture for breaking waves simulation using MPS, Journal of Physics: Conference Series, Vol.971, 2018.
- [7] 北山洋幸 . 512 ビット・ベクトルプログラミング入門 AVX-512 命令を駆使しよう, カットシステム出版, 2021.