5J-02

疎行列を対象とした共役勾配法における マルチコア/AVX-512による並列処理

Parallelization with Multicore/AVX-512 for Conjugate Gradient Solver of Sparse Matrix

三村 卓也† 吉田 明正†
Takuya Mimura Akimasa Yoshida

1 はじめに

近年，数値シミュレーションにおいて，偏微分方程式を離散化して大規模な連立一次方程式を解く問題に帰着させることが知られている．問題の中には，疎行列を対象とするものが多く存在し，大規模な行列においては多くの計算時間を要する．共役勾配法（Conjugate Gradient 法）とその前処理の並列処理および疎行列計算に関する研究は活発に行われている [1][2][3][4]．本稿では，疎行列格納形式として CRS と ELL を SIMD 拡張し，OpenMP 指示文と AVX-512 命令を用いて CG 法の並列プログラムを実装した．Intel Xeon Platinum 8358 の 32 コア上での性能評価の結果，高い実効性能が得られ提案手法の有効性が確認された．

2 共役勾配法と疎行列の格納形式

本稿では，共役勾配法（CG 法）の並列処理において，CRS（Compressed Row Storage）形式と ELL（Ellpack-Itpack）形式を用いて，疎行列の非ゼロ要素を SIMD データ単位で格納し，OpenMP によるループ並列処理と SIMD 並列処理を階層的に行う．

2.1 CG 法のアルゴリズム

共役勾配法では，係数行列 A を正定値対称行列とし，連立一次方程式 Ax=b の解 x を求めることができる．図 1 にそのアルゴリズムを示す [5]．図 1 において，関数 mvm() は行列-ベクトル積を求め，関数 ip() はベクトルの内積を求める．

```
01: for(k=0;k<ITER;k++){
02:   mvm(A,p,Ap,size);
03:   alpha=ip(r,r,size)/ip(p,Ap,size);
04:   rk=ip(r,r,size);
05:   for(i=0;i<size;i++)
06:     x[i]=x[i]+alpha*p[i];
07:   for(i=0;i<size;i++)
08:     r[i]=r[i]-alpha*Ap[i];
09:   rk1=sqrt(ip(r,r,size));
10:   if(rk1/r0<=eps){
11:     break;
12:   }
13:   beta=ip(r,r,size)/rk;
14:   for(i=0;i<size;i++)
15:     p[i]=r[i]+beta*p[i];
16: }
```

図 1 CG 法のアルゴリズム

2.2 SIMD を考慮した CRS/ELL 形式による疎行列のデータ構造

本稿で提案する SIMD を考慮した CRS（Compressed Row Storage）形式は，図 2 に示すように，非ゼロ要素を格納する配列 val，その列インデックスを格納する配列 colind，各行の非ゼロ要素開始位置を格納する配列 rowptr から構成される．通常（SIMD でない）の CRS 形式では，(b) の val を double 型の配列としているが，提案手法では AVX-512 の場合に__m512d 型，AVX-256 の場合に__m256d 型を採用する．例えば，図 2(a) の (1)〜(10) の SIMD データ (double 型配列のエイリアス) は，(b) の (1)〜(10) のように配列 val に格納される．

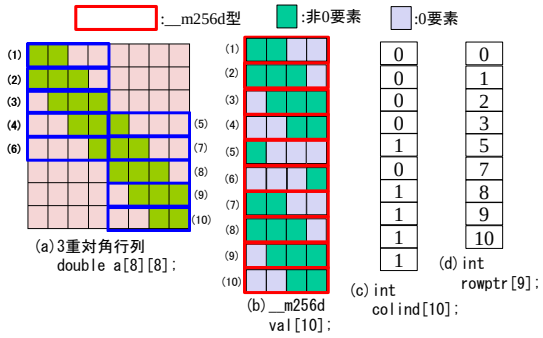

図 2 SIMD を考慮した CRS 形式.

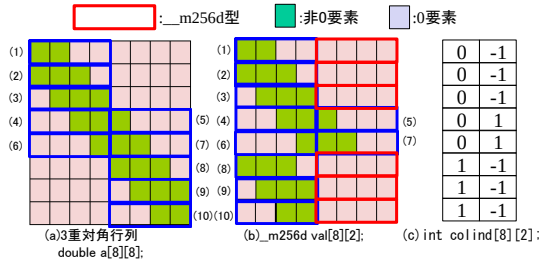

図 3 SIMD を考慮した ELL 形式.

また，提案する SIMD を考慮した ELL（Ellpack-Itpack）形式は，図 3 に示すように，非ゼロ要素を格納する配列 val，各行の列インデックスを格納する配列 colind から構成される．通常（SIMD でない）の CRS 形式では，(b) の val を double 型の配列としているが，提案手法では AVX-512 の場合に__m512d 型，AVX-256 の場合に__m256d 型を採用する．例えば，図 3(a) の (1)〜(10) の SIMD データは，(b) の (1)〜(10) のように配列 val に格納される．

3 CRS/ELL 形式を用いた CG 法のマルチコア/SIMD 並列処理

本稿では，CRS 形式と ELL 形式を用いた疎行列データに対して，マルチコア並列処理と SIMD 並列処理を階層的に組み合わせた並列処理手法を提案する．

3.1 CRS/ELL 形式における SIMD 並列処理

2.2 節で提案した SIMD を考慮した CRS/ELL 形式によるデータ構造を実装することにより，AVX-512 を用いた SIMD 並列処理を実現する．AVX-512 では__m512d 型を用いることにより，512 ビットの SIMD データにおいて，8 バイトの double 型変数 8 個を含めることがで

†明治大学大学院先端数理科学研究科ネットワークデザイン専攻
Department of Network Design, Graduate School of Advanced Mathematical Sciences, Meiji University

表 1　性能評価に用いる並列サーバの構成.

CPU	Intel Xeon Platinum 8358
コア数	32 コア
最大スレッド数	64 スレッド
メモリ	256GB
OS	Ubuntu20.04LTS
C コンパイラ	Intel one API 2022
オプション	-qopenmp -xICELAKE-SERVER -O2

表 2　Xeon 上での並列処理時間（非ゼロ要素率 0.09%）.

実行方法	1 スレッド	4 スレッド	32 スレッド
2 次元配列+OMP	42,181	11,892	3,405
CRS+OMP+SIMD	8 (5,273 倍)	6 (7,030 倍)	12 (3,515 倍)
ELL+OMP+SIMD	68 (620 倍)	31 (1,361 倍)	25 (1,687 倍)

(注) 単位は [ms]，( ) 内は逐次比.

きる．この結果，AVX-512 命令を用いることで，8 個の double 型変数を一度に計算することが可能になる．

3.2　マルチコアによるループ並列処理

OpenMP（Open Multi-Processing）は，共有メモリ型マシンで並列プログラミングを可能にする API である．本稿では CG 法の収束ループ内部において，並列化可能な for 文に OpenMP 指示文を加えて，32 コア上でループ並列処理を実現する．さらに，各コアのスレッド内部において，上述の SIMD 並列処理を適用する．これにより，SIMD を考慮した CRS/ELL データ構造において，マルチコアによるループ並列処理と SIMD 並列処理を組み合わせた階層的な並列処理を実現する．

4　CG 法のマルチコア/SIMD 並列処理の性能評価

本性能評価では，Intel Xeon サーバ上で，CRS 形式と ELL 形式による疎行列の格納を行い，マルチコア上でのループ並列処理と AVX-512 による SIMD 並列処理を階層的に組み合わせた CG 法の性能評価を行う．

4.1　性能評価環境

性能評価に用いる並列サーバの構成を表 1 に示す．実行方法としては，(1) 2 次元配列+OpenMP，(2)CRS+OpenMP+SIMD，(3)ELL+OpenMP+SIMD の 3 通りとする．なお，表中では OpenMP を OMP と表記している．

CG 法の性能評価に用いる正定値対称行列は，行列サイズを 10000 × 10000，非ゼロ要素率を 0.09%，0.99%，9.74%の 3 種類としている．

4.2　Intel Xeon 上での CG 法の性能評価

まず，非ゼロ要素率 0.09%における Xeon サーバ上での CG 法の並列処理時間を表 2 に示す．32 スレッド実行の場合，2 次元配列+OpenMP では 3,405[ms]（逐次比で 12 倍），CRS+OpenMP+SIMD では 12[ms]（逐次比 3,515 倍），ELL+OpenMP+SIMD では 25[ms]（逐次比 1,687 倍）となっている．

次に，非ゼロ要素率 0.99%における Xeon サーバ上での CG 法の並列処理時間を表 3 に示す．32 スレッド実行の場合，2 次元配列+OpenMP では 7,829[ms]（逐次比で 12 倍），CRS+OpenMP+SIMD では 13[ms]（逐次比で 7,452 倍），ELL+OpenMP+SIMD では 51[ms]（逐次比で 1,899 倍）となっている．

表 3　Xeon 上での並列処理時間（非ゼロ要素率 0.99%）.

実行方法	1 スレッド	4 スレッド	32 スレッド
2 次元配列+OMP	96,873	27,750	7,829
CRS+OMP+SIMD	22 (4,403 倍)	12 (8,073 倍)	13 (7,452 倍)
ELL+OMP+SIMD	584 (166 倍)	203 (477 倍)	51 (1,899 倍)

(注) 単位は [ms]，( ) 内は逐次比.

表 4　Xeon 上での並列処理時間（非ゼロ要素率 9.74%）.

実行方法	1 スレッド	4 スレッド	32 スレッド
2 次元配列+OMP	749,120	215,750	62,209
CRS+OMP+SIMD	1,012 (740 倍)	249 (3,009 倍)	130 (5,762 倍)
ELL+OMP+SIMD	73,322 (10 倍)	21,861 (34 倍)	3,295 (227 倍)

(注) 単位は [ms]，( ) 内は逐次比.

さらに，非ゼロ要素率 9.74%における Xeon サーバ上での CG 法の並列処理時間を表 4 に示す．32 スレッド実行の場合，2 次元配列+OpenMP では 62,209[ms]（逐次比で 12 倍），CRS+OpenMP+SIMD では 130[ms]（5,762 倍），ELL+OpenMP+SIMD では 3,295[ms]（227 倍）となっている．なお，非ゼロ要素率 9.74%の ELL+OpenMP+SIMD の実行が，CRS+OpenMP+SIMD に比べて遅くなっている原因としては，非ゼロ要素数の増加によりキャッシュミスが増えていると考えられる．

以上の性能評価の結果から，正定値対称行列である 3 種類の疎行列において，提案する CRS+OpenMP+SIMD 実行が最も高速でありその有効性が確認された．

5　おわりに

本稿では，CG 法の並列処理において，CRS 形式と ELL 形式を SIMD 拡張し，マルチコアによるループ並列処理と SIMD 並列処理を階層的に組み合わせた並列処理行手法を提案した．

Intel Xeon Platinum サーバ上での性能評価では，CRS+OpenMP+SIMD においては，非ゼロ要素率 0.99%の疎行列で最大 8,073 倍の速度向上率，ELL+OpenMP+SIMD においては，非ゼロ要素率 0.99%の疎行列で最大 1,899 倍の速度向上率が得られた．それゆえ，SIMD 命令を考慮した CRS/ELL 疎行列格納形式による CG 法の並列処理の有効性が確認された．

参考文献

[1] 久保田修司，　高橋大輔．GPU における格納形式自動選択による疎行列ベクトル積の高速化，情報処理学会研究報告 Vol.2010-ARC-192 No.19, 2010.

[2] 菱沼 利彰，井原 遊，高村 守幸，平野 哲，荻原 孝，岩田直樹，奥田 洋司，SX-Aurora TSUBASA における有限要素解析のための共役勾配法の性能評価．情報処理学会研究報告 Vol.2020-HPC-175 No.18，2020.

[3] H.Yoshizawa, D.Takahasi.　Automatic Tuning of Sparse Matrix-Vector Multiplication for CRS format on GPUs,　IEEE 15th International Conference on Computational Science and Engineering, 2012.

[4] T.Iwashita, N.Takemura, A.Ida, H.Nakashima.　A new fill-in strategy for IC factorization preconditioning considering SIMD instructions,　IEEE Trustcom/BigDataSE/ISPA, 2015.

[5] 寒川　光，藤野　清次，長崎　利夫，高橋　大介．IT Text HPC プログラミング．オーム社，2009.