# Intel/KNL における倍々精度疎行列ベクトル積の特性評価

土肥　樹[†]　伊藤　友太[†]　菱沼　利彰[‡]　藤井　昭宏[†]　田中　輝雄[†]

工学院大学[†]　　筑波大学[‡]

## 1. はじめに

大規模シミュレーションの核である反復解法では，疎行列ベクトル積（Sparse Matrix Vector : SpMV）が多くの時間を占める．反復解法の収束の改善には高精度演算が有効である [1].

高精度演算を実装する手法のひとつに倍々精度演算（DD 演算）がある．倍々精度演算は倍精度変数を 2 つ組み合わせて 4 倍精度演算を実現する手法であり，倍精度計算に対して 20~30 倍の演算量を必要とする．菱沼らは SIMD 拡張命令の AVX2 を用いて 1 命令で 4 つの倍精度演算を行い，DD 演算を高速化した [2].

一方，メニーコアプロセッサとして Xeon Phi がある．Xeon Phi は 60 以上のコアプロセッサをもつ．佐々木らは Xeon Phi の第一世代である Knights Corner（KNC）を用いて倍精度疎行列と倍々精度ベクトルの疎行列ベクトル積（DD-SpMV）の性能評価を行った [3].

Xeon Phi の第二世代である Knights Landing（KNL）は広帯域メモリ Multi-Channel DRAM（MCDRAM）を搭載していることが特徴のひとつとして挙げられる[4]．MCDRAM は単位時間当たりに転送できるデータが多いため，メモリ性能に制約を受けにくい [5].

KNL は SIMD 拡張命令の AVX-512 を搭載しており，1 命令で 8 つの倍精度演算を行うことが可能である．

本研究では AVX-512 を用いて DD-SpMV を実装し，SuiteSparse の疎行列を用いた DD-SpMV の分析と帯行列の DD-SpMV の 2 つの観点から KNL の特性評価を行った．

DD-SpMV の最内側で使われる倍々精度積和演算に必要な演算回数は，AVX-512 利用時は 20 回，非利用時は 32 回である．

Characterization of Double-Double Sparse Matrix Vector Multiplication on Knight Landing

Tatuski Doi[†], Yuta Ito[†], Toshiaki Hishinuma[‡], Akihiro Fujii[†] and Teruo Tanaka[†]

[†]Kogakuin University, [‡]University of Tsukuba

## 2. 特性評価環境

表 1　環境変数

| 変数名 | 値 |
|---|---|
| KMP_HW_SUBSET | 64c@2,4t |
| KMP_AFFINITY | compact |
| OMP_SCHEDULE | static |

評価環境は Intel Xeon Phi 7250，1.4GHz，68 コア，L2 キャッシュは 34MB，メモリバンド幅は 450GB/s（MCDRAM 利用時），コンパイラは Intel C/C++ Compiler 17.0.1 を用い，コンパイルオプションは “icc -qopenmp -no-vec -fp-model precise -O3”，並列化には OpenMP を用いた．環境変数を表 1 に示す．疎行列格納形式には Compressed Row Storage（CRS）形式を用いた [6].

本論文では，DD-SpMV の性能指標を 2×非零要素数/実行時間とした．割り込み処理の影響を受けるコアを使用しないため，計測に使用したコアは 64 コアとなる．したがって，スレッド数は 256 スレッドで計測した．

## 3. DD-SpMV の性能分析

### 3.1 SuiteSparse Matrix Collection を用いた性能分析

SuiteSparse Matrix Collection [7]から入手した 11 種の対称で正方な疎行列を用いて DD-SpMV 処理を分析した．疎行列を 1 行当たりの平均非零要素数順に番号付けし，名称，非零要素数(nnz)，行列サイズ(N)，1 行当たりの平均非零要素数を表 2 に示す．これらの疎行列のデータサイズはすべて MCDRAM に収まる．AVX-512 の利用の

表 2　疎行列一覧

| # | Name | nnz(×10⁴) | N(×10⁴) | nnz/N |
|---|---|---|---|---|
| 1 | memchip | 1481.0 | 270.7 | 5.4 |
| 2 | apache1 | 54.2 | 8.0 | 6.4 |
| 3 | fv1 | 8.2 | 0.9 | 8.8 |
| 4 | Dubcova2 | 103.0 | 6.5 | 15.8 |
| 5 | nasa4704 | 10.4 | 0.4 | 22.2 |
| 6 | poisson3Db | 237.4 | 8.5 | 27.7 |
| 7 | bcsstk15 | 11.7 | 0.3 | 29.8 |
| 8 | bcsstd13 | 8.3 | 0.2 | 41.8 |
| 9 | bcsstk16 | 29.0 | 0.4 | 59.4 |
| 10 | raefsky3 | 148.8 | 2.1 | 70.2 |
| 11 | pdb1HYS | 434.4 | 3.6 | 119.3 |

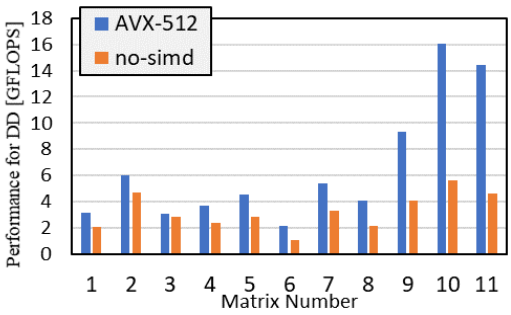


図 1 DD-SpMV の性能 (SuiteSparse)

有無を比較した DD-SpMV の性能を，図 1 に示す．図 1 は表 2 に対応した 1 行あたりの平均非零要素数順である．Number が大きいほど，データが連続して配置される可能性が高いため $x$ のメモリアクセスが良い．

1 行当たりの平均非零要素数が多い疎行列は AVX-512 を利用の有無で最大 3 倍の性能差がある．しかし，1 行当たりの平均非零要素数が少ない疎行列や，非零要素が連続して配置されていない疎行列は AVX-512 の効果が表れにくい．

### 3.2 データサイズによる SpMV の性能分析

データの連続性の演算性能に対する影響を分析するために，帯行列の帯幅を変化させて DD-SpMV を計測した．行列サイズ N は $10^5$ とした．AVX-512 の利用の有無の結果を図 2 に示す．

AVX-512 を利用した場合，帯幅が増えるにつれて，演算性能が向上していき利用しない場合と比較して 4 倍の演算性能となった．利用しない場合，性能は一定の値で頭打ちになる．これは演算器の性能がメモリから転送されるデータ量に対して処理が追い付かないことが原因と考えられる．したがって，非零要素の配置が連続することと 1 行当たりの平均非零要素数が，演算性能に依存していることが判明した．

次に，データサイズによる演算性能の影響を分析するために帯行列の行列サイズを変化させて DD-SpMV を計測した．1 行当たりの平均非零要素数は 32 とした．AVX-512 の利用の有無の結果を図 3 に示す．行列サイズが約 $6.2\times10^4$ までは L2 キャッシュに収まる．

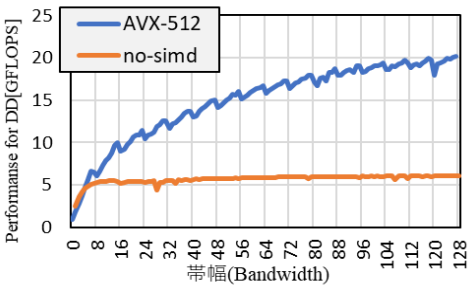


図 2 帯幅を変更した帯行列の DD-SpMV

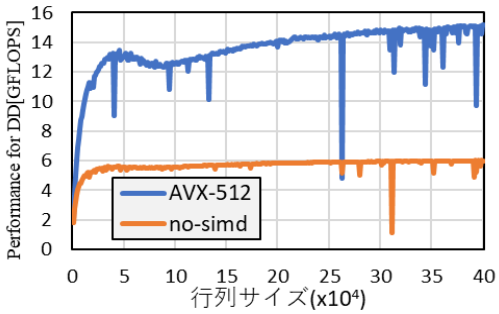


図 3 行列サイズを変更した帯行列の DD-SpMV

L2 キャッシュに収まるデータサイズの場合，並列化オーバーヘッドが大きく演算性能が低い．L2 キャッシュを超えるデータサイズでは，オーバーヘッドが隠蔽され演算性能が向上した．

## 4. おわりに

本論文では SIMD 拡張命令の AVX-512 を用いて，倍々精度の DD-SpMV を実装し，KNL の特性評価を行った．

KNL における DD-SpMV は，疎行列の非零要素の配置が連続すると AVX-512 の恩恵を受けて，性能が向上する．したがって，非零要素の配置が大きく性能に影響を与えると結論付けた．

今後の課題として反復解法への適応や，疎行列の構造に合わせた格納形式や環境変数を変更する必要があり，その指標を決める必要がある．

## 参考文献

[1] Hasegawa Hidehiko, Utilizing the quadruple-precision floating-point arithmetic operation for the Krylov Subspace Methods, The 8th SIAM Conference on Applied Linear Algebra, 2003.

[2] 菱沼利彰，藤井昭宏，田中輝雄，長谷川秀彦，AVX2 を用いた倍精度 BCRS 形式疎行列と倍々精度ベクトル積の高速化，情報処理学会論文誌コンピューティングシステム(ACS)，Vol. 7，No. 4，pp. 25-33，2014.

[3] 佐々木信一，菱沼利彰，藤井昭宏，田中輝雄，Many Integrated Core architecture における倍々精度疎行列ベクトル積，ハイパフォーマンスコンピューティング研究会，研究報告 2014-HPC-145, No.16, pp.1-7, 2014.

[4] How Intel® Xeon Phi™ Processors Benefit Machine Learning/Deep Learning Apps and Frameworks, https://software.intel.com/en-us/blogs/2016/06/20/how-xeon-phi-processors-benefit-machine-and-deep-learning-apps-frameworks （参照 2017-12-26）.

[5] Jim Jeffers, James Reinders, Avinash Sodani, Intel® Xeon Phi™ Processor High Performance Programming Knights Landing Edition, Morgan Kaufmann, 2016.

[6] Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, pp.57-65, 1994.

[7] SuiteSparse Matrix Collection, https://sparse.tamu.edu/, （参照 2017-12-26）.