

AVX を用いた倍々精度疎行列ベクトル積-転置行列-

菱沼利彰^{†1} 藤井昭宏^{†1} 田中輝雄^{†1} 長谷川秀彦^{†2}

1. はじめに

われわれは、AVX を用いて倍々精度ベクトル間の演算や、倍精度疎行列と倍々精度ベクトルの積の高速化を行ってきた[1]。倍々精度演算とは、倍精度演算を組み合わせることで4倍精度演算を実現する手法である[2]。AVX と SSE2 は SIMD 拡張命令であり[3]、SSE2 は倍精度 2 つ、AVX は倍精度 4 つのデータに対し SIMD 演算を実行できる。

倍精度疎行列 A と倍々精度ベクトル x の積 Ax において、 A を CRS 形式[4]で格納したとき、1 スレッドで、AVX の性能は約 6GFLOPS から 13GFLOPS となり、SSE2 の性能と比べて約 1.12 倍から 1.88 倍となった。また、性能は疎行列の平均非零要素数に大きく依存する。

反復法では、BiCG 法のように Ax と $A^T x$ の両方を必要とするものがあるため、CRS 形式で格納された A に対し AVX を用いて $A^T x$ の高速化を図る。

2. 実験

実験には Intel Core i7-2600K, Fedora16, Intel C/C++ Compiler 12.0.3 を用いた。コンパイルオプションは、AVX コードに対して“-O3 -xAVX -openmp -fp-model precise”，SSE2 コードに対して“-O3 -xSSE2 -openmp -fp-model precise”を用い、The University of Florida Sparse Matrix Collection (<http://www.cise.ufl.edu/research/sparse/matrices/>, フロリダコレクション)の疎行列を用いてテストを行った。

Ax と $A^T x$ は、倍精度加減算 8 回と倍精度乗算 25 回によって実現され、 x と解をストアする倍々精度ベクトル y へのアクセスが異なる。

図 1 に $A^T x$ の性能を示す。SSE2 と比較した AVX の性能

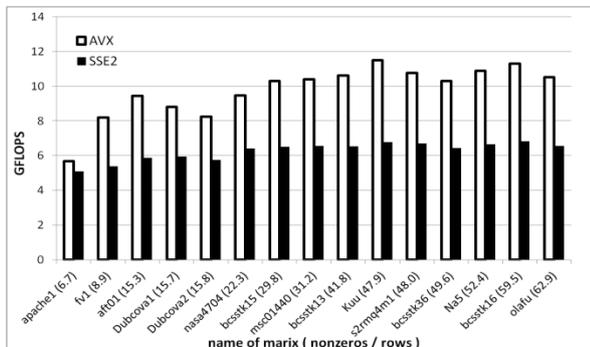


図 1 $A^T x$ の性能(1 スレッド)

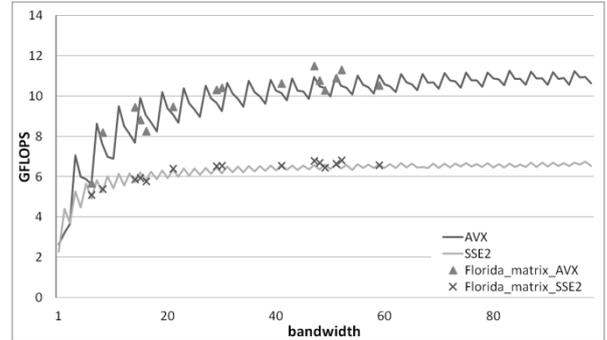


図 2 $A^T x$ の帯幅と性能の関係(1 スレッド)

は約 1.1 倍から 1.7 倍となった。倍々精度演算 Ax と $A^T x$ は加減算と乗算のバランスが悪く、AVX の理論ピーク性能である 27.2GFLOPS と、SSE2 の理論ピーク性能である 13.6GFLOPS の達成は不可能で、AVX は最大 18GFLOPS、SSE2 は最大 9GFLOPS となる。これに対し、AVX は約 32% から 64%、SSE2 は約 57% から 76% の性能となった。

$A^T x$ の性能は y に対するキャッシュミスが発生しやすく、平均非零要素数が 20 以上のとき AVX は Ax の性能の約 0.84 倍から 0.94 倍、SSE2 は約 0.97 倍から 0.99 倍になるが、平均非零要素数が 20 以下のとき、AVX は約 1.01 倍から 1.15 倍、SSE2 は約 1.02 倍から 1.06 倍となった。

図 2 に $N=10^5$ の帯行列の帯幅と性能の関係を示す。フロリダコレクションを同一グラフ上に示した。端数処理の影響により性能が階段状になるが、帯幅が 40 のとき、AVX は約 10.9GFLOPS となり理論ピーク性能の 60.1%、SSE2 は約 6.3GFLOPS となり、理論ピーク性能の 70% となった。

帯行列の性能と比較したフロリダコレクションの疎行列の性能は AVX は約 0.81 倍から 1.12 倍、SSE2 は約 0.94 倍から 1.02 倍となり、 Ax 同様に性能は平均非零要素数に大きく依存し、フロリダコレクションのような実際のアプリケーションで用いられる疎行列において、平均非零要素数から性能がある程度予測できると考えられる。

参考文献

- [1] 菱沼, 他: 反復法ライブラリ向け倍々精度演算の AVX を用いた高速化, HPC-135, No. 16 (2012)
- [2] Bailey, D, H.: A fortran-90 double-double library. <http://crd-legacy.lbl.gov/~dhbailey/mpdist/>
- [3] Intel® Architecture Instruction Set Extensions Programming Reference <http://software.intel.com/en-us/avx>
- [4] Barrett, R., et al.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM (1994)

^{†1} 工学院大学情報学部
Faculty of Informatics, Kogakuin University

^{†2} 筑波大学図書館情報メディア系
Faculty of Library, Information and Media Science, University of Tsukuba