

## Intel AVX 命令を用いた並列 FFT の実現と評価

Title: Implementation and evaluation of parallel FFT using Intel AVX instructions

高橋大介

Daisuke Takahashi

筑波大学システム情報系

Faculty of Engineering, Information and Systems, University of Tsukuba,  
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan

本発表では、Intel AVX 命令を用いて並列 FFT を実現し評価した結果について報告する。浮動小数点演算をより高速に処理するために、最近のプロセッサでは Intel の SSE/SSE2/SSE3/SSSE3/SSE4/AVX や、Motorola Power PC の AltiVec、そして SPARC64 VIIIfx の HPC-ACE など、Short Vector SIMD 命令を搭載しているものが多い。

しかし、これらの Short Vector SIMD 命令を使ったとしても、最近のプロセッサのデータ供給能力は、キャッシュに頼っているのが現状であり、メモリアクセスの最適化もあわせて行う必要がある。したがって、Short Vector SIMD 命令を搭載したプロセッサを用いて FFT を計算する際には、Short Vector SIMD 命令を有効に活用しつつ、かつキャッシュミスの回数を減らすということが性能を引き出す上での大きな鍵となる。

これまでに、Short Vector SIMD 命令を用いた FFT の実装[1, 2]がいくつか行われている。本研究では、Intel の Sandy Bridge マイクロアーキテクチャから導入された Short Vector SIMD 命令の 1 つである、AVX (Advanced Vector Extensions) 命令[3]を用いて、FFT カーネルのベクトル化を行った。さらに、キャッシュメモリを効果的に利用できるブロック six-step FFT アルゴリズム[4]と組み合わせると共に、OpenMP により並列化を行うことで、性能が向上することを示す。

実現した並列 FFT を Intel Xeon E3-1230 (3.2 GHz, 1 CPU, 4 コア) および Intel Core i5-2520M (2.5 GHz, 1 CPU, 2 コア) 上で性能評価を行った結果、 $2^{24}$  点倍精度複素数 FFT において、Intel Xeon E3-1230 では約 11.7 GFlops, Intel Core i5-2520M では約 5.8 GFlops の性能を得ることができた。

[1] M. Frigo and S. G. Johnson: The design and implementation of FFTW3, Proc. IEEE, Vol. 93, pp. 216—231 (2005).

[2] D. S. McFarlin, V. Arbatov, F. Franchetti, and M. Püschel: Automatic SIMD vectorization of fast Fourier transforms for the Larrabee and AVX instruction sets, Proc. 25th International Conference on Supercomputing (ICS'11), pp. 265—274 (2011).

[3] Intel Corporation: Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 1: Basic Architecture (2011).

[4] D. Takahashi: A blocking algorithm for FFT on cache-based processors, Proc. 9th International Conference on High Performance Computing and Network Europe (HPCN Europe 2001), Lecture Notes in Computer Science, No. 2110, pp. 551—554, Springer-Verlag (2001).