

メニーコア上での離散ウェーブレット変換の SIMD 命令を用いた階層的並列処理

Hierarchical Parallel Processing Using SIMD Instructions for Discrete Wavelet Transform on Manycore Processor

西倉 佑騎†
Yuki Nishikura

吉田 明正†
Akimasa Yoshida

1 はじめに

ベクトル演算ユニットを搭載した Intel Xeon Phi[1] は OpenMP[2] によるループ並列処理と AVX-512[3] による SIMD 並列処理を併用して演算を高速にすることが期待されている。

本稿では、画像圧縮等に用いられる 2 次元離散ウェーブレット変換プログラムを対象とし、画像データの行レベルの計算に OpenMP によるループ並列処理を適用し、さらに各行のデータとドベシ数列との内積演算に SIMD 並列処理を適用して、階層的な並列処理を実現する。本性能評価では、AVX-512 を搭載する Intel Xeon Phi のメニーコア (68 コア) を利用し、提案手法を実装した C プログラムにより並列実行したところ高い実効性能が達成され、提案手法の有効性が確認された。

2 離散ウェーブレット変換の階層的並列処理

2 次元離散ウェーブレット変換と SIMD 命令を用いた階層的並列処理手法について述べる。

2.1 Ingrid-Daubechies ウェーブレット変換

Ingrid-Daubechies ウェーブレット変換 [4] は、多重解像度解析を伴って画像圧縮に用いられるが、本稿では図 1(a) のようなレベル $j(j=1)$ の計算のみ扱う。画像データと p_k (数列長=16) と q_k (数列長=16) を使用し、式 (1) と式 (2) により求める。今回のプログラムでは図 1 のような $n \times n$ の画像 ($n=8192$) を入力とし、ウェーブレット変換によって、図 1(b) のような $n \times k(k=4096)$ のスケーリング係数 s 、 $n \times k$ のウェーブレット展開係数 w を求める。同様に s と w に対して垂直方向にウェーブレット変換を適用し、図 1(c) の $s1, w1h, w1v, w1d$ を求める。 $s1$ には $s0$ の低域成分、 $w1h$ には水平方向の高域成分、 $w1v$ には垂直方向の高域成分、 $w1d$ には水平・垂直方向の高域成分が現れる。

$$s_k^{(j)} = \sum_n \overline{p_{n-2k}} s_n^{(j-1)} \quad (1)$$

$$w_k^{(j)} = \sum_n \overline{q_{n-2k}} s_n^{(j-1)} \quad (2)$$

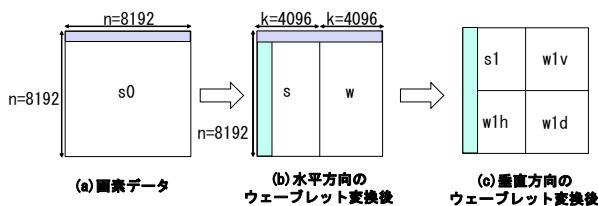


図 1 2 次元ウェーブレット変換の概念。

2.2 ウェーブレット変換の OpenMP によるループ並列処理

2 次元離散ウェーブレット変換プログラムにおいて図 1(a) の行単位で Intel Xeon Phi のメニーコアを用いて、OpenMP によるループ並列処理を行う。並列コードは指示文 `#pragma omp parallel for schedule(dynamic,64)` により実現する。ダイナミックスケジューリングのチャンクサイズ=64 としており、図 1(a) のような 8k 画像では 8192 行を 64 行単位で各コアにダイナミックスケジューリングしている。

2.3 ウェーブレット変換の SIMD (AVX-512) による SIMD 並列処理

Intel Xeon Phi プロセッサでは AVX-512(512bit) のベクトル演算ユニットが搭載されている。主な組み込み関数は表 1 の通りである。AVX-512 は 512bit 幅の ZMM レジスタを使用して演算を行う。

本節では、OpenMP により各コアに割り当てられた行データに対して水平方向のウェーブレット変換の過程、即ち図 1(a) の配列 $s0$ の各行を入力として、ドベシの数列 p_k に対する内積演算を行い、図 1(b) 左側のスケーリング係数 s を求める過程を説明する。

各行の画素データに対してドベシの数列 p_k (数列長=16) との内積を求める際に、AVX-512 の SIMD 命令を用いる。図 2(a) に AVX-512 コードを適用したものを示す。このコードによってスケーリング係数の 1 つ (この例では変数 $tall$) を求めることができる。AVX-512 では ZMM レジスタを用いるため、double 型データの場合 8 要素を同時に演算可能である。16 要素の内積 (図 2(b) の $vs0[0] \cdot vp[0]$ と $vs0[1] \cdot vp[1]$) の場合、8 要素単位の内積を $vt[0]$ 及び $vt[1]$ として求めて、それらの総和を $vtall$ に求め、 $vtall$ の 8 要素の和を $tall$ に求める。

これらの方法により、OpenMP によるループ並列処理と AVX-512 による SIMD 並列処理を階層的に行うことが可能になる。

表 1 AVX-512 の主な組み込み関数 (double 型)。

ロード*	<code>_mm512_load_pd(a)</code>
ストア	<code>_mm512_store_pd(a)</code>
加算	<code>_mm512_add_pd(_mm512d a, _mm512d b)</code>
乗算	<code>_mm512_mul_pd(_mm512d a, _mm512d b)</code>

3 Intel Xeon Phi 上での性能評価

本稿では、2 次元離散ウェーブレット変換プログラムを用いて性能評価を行う。関連研究として、GPU 上での離散ウェーブレット変換の階層的並列処理 [5] が行われているが、本研究のような SIMD 命令は用いられていなかった。

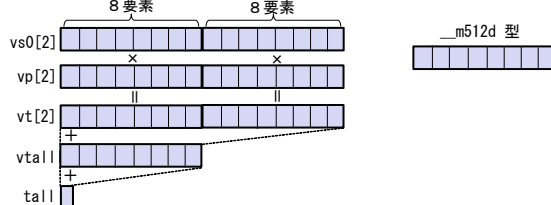
† 明治大学大学院先端数理科学研究科ネットワークデザイン専攻
Graduate School of Advanced Mathematical Sciences

```

__m512d vs0[2], vp[2], vt[2], vtall;
double at[8], tall;
:
for (i=0; i<2; i++) {
    vt[i] = _mm512_mul_pd(vs0[i], vp[i]);
    vtall = _mm512_add_pd(vtall, vt[i]);
}
_mm512_store_pd(at, vtall);
tall = at[0] + at[1] + at[2] + at[3]
      + at[4] + at[5] + at[6] + at[7];

```

(a) SIMD命令を用いたプログラム



(b) SIMD命令による計算手順

図 2 2次元ウェーブレット変換プログラムにおけるSIMD 並列処理。

3.1 性能評価環境

本性能評価では、画像圧縮の対象として画像データ (8k:8192*8192) を用いて、表 2 の Intel Xeon Phi サーバ上で実行した。画素値としては Y, Cb, Cr に変換された Y 成分のウェーブレット変換の処理時間を測定した。また OpenMP によるループ並列処理と AVX-512 による SIMD 並列処理の効果を確認するために最適化オプションは-O0 としている。

表 2 性能評価に用いる Intel Xeon Phi サーバの構成。

マシン	Intel Xeon Phi Knights Landing Server
プロセッサ	Intel Xeon Phi Processor 7250
CPU コア	68 コア (1.4GHz)
メモリ	48GB
OS	CentOS OS 7.4
処理系	icc18.0.1

3.2 階層的並列処理 (AVX-512 あり) による性能評価

本節では、OpenMP によるループ並列処理と AVX-512 による SIMD 並列処理を組み合わせた階層的並列処理による実行結果を示す。8k(8192*8192) の画像データに対して double 型と float 型の演算を適用した結果を図 3, 図 4 に示す。

double 型演算において、AVX-512 なし・64 スレッド実行では、実行時間は 1957[ms] となり逐次実行比で 43.8 倍であった。それに対して、AVX-512 あり・64 スレッド実行では、実行時間は 695[ms] となり逐次実行比で 123.5 倍であった。

AVX-512 の場合、float 型データであれば 16 要素を同時に扱えるため、精度が下がるが float 型演算でも性能評価を行った。AVX-512 なし 64 スレッド実行では、実行時間は 1764[ms] となり逐次実行比で 48.6 倍であった。それに対して、AVX-512 あり 64 スレッド実行では、実行時間は 480[ms] となり逐次実行比で 178.5 倍であった。

これらの結果から OpenMP による並列処理と AVX-512 による並列処理を組み合わせた階層的並列処理手法の有効性が確認できた。

4 おわりに

本稿では、2次元離散ウェーブレット変換プログラムに対して画像データの行レベルの計算に OpenMP による

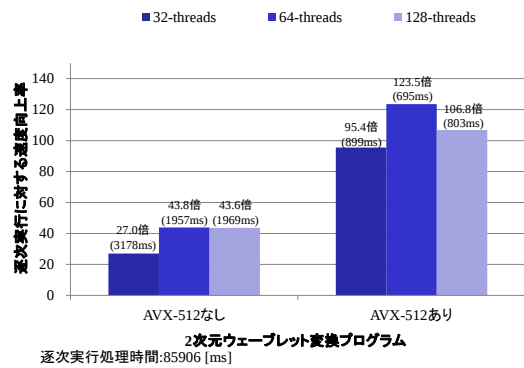


図 3 8k 画像における離散ウェーブレット変換の階層的並列処理 (double 型)。

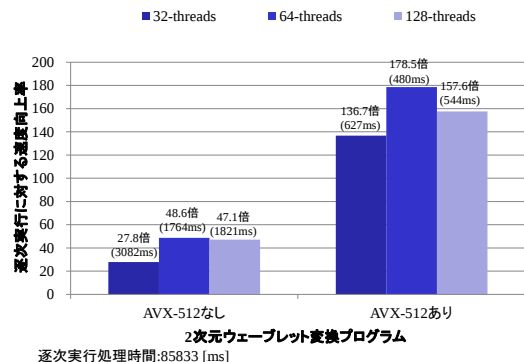


図 4 8k 画像における離散ウェーブレット変換の階層的並列処理 (float 型)。

ループ並列処理を適用し、さらに各行のデータとドベシ数列との内積演算に SIMD 並列処理 (AVX-512) を実現する階層的並列処理手法を提案した。Intel Xeon Phi 上で性能評価を行ったところ、ループ並列処理と SIMD 並列処理を組み合わせた階層的並列処理では、8k サイズの画像データにおいて 64 スレッド実行で 123.5 倍 (double 型) と 178.5 倍 (float 型) の速度向上が得られた。これらの結果から SIMD 命令 (AVX-512) を用いた階層的並列処理の有効性が確認された。

今後の課題として、大規模アプリケーションに本手法やタスク駆動型粗粒度並列処理 [6] を応用することが挙げられる。

参考文献

- [1] Intel Xeon Phi. <https://ark.intel.com/content/www/jp/ja/ark/products/94035/intel-xeon-phi-processor-7250-16gb-1-40-ghz-68-core.html>, 2019.
- [2] OpenMP. <https://www.openmp.org/>, 2018.
- [3] ジム・ジェファース, ジェームズ・レインダース, アビナッシュ・ソダーニ. インテル Xeon Phi プロセッサハイパフォーマンス・プログラミング Knights Landing エディション. カットシステム, 2017.
- [4] 中野宏毅, 山本鎮男, 吉田靖夫. ウェーブレットによる信号処理と画像処理. 共立出版, 1999.
- [5] 吉川一輝, 高平真由, 吉田明正. マルチ GPU 上での画像圧縮における離散ウェーブレット変換の階層的並列処理, 情報処理学会第 16 回情報科学技術フォーラム I-014, 2018.
- [6] A.Yoshida, A.Kamiyama, H.Oka. A Task-Driven Parallel Code Generation Scheme for Coarse Grain Parallelization on Android Platform. Journal of Information Processing, Vol.25, pp.426-437, 2017.