1H-06

SX-Aurora TSUBASA 上での離散ウェーブレット変換の階層的並列処理

Hierarchical Parallel Processing of Discrete Wavelet Transform on SX-Aurora TSUBASA

西倉　佑騎 †　　吉田 明正 †
Yuki Nishikura　　Akimasa Yoshida

1　はじめに

近年，CPU のメニーコア化に加えて，SIMD 命令やベクトル命令を使った高速化が期待されている．

SX-Aurora TSUBASA[1][2][3] はベクトルホストとベクトルエンジンを搭載したベクトルコンピュータである．ベクトルエンジンは高メモリバンド幅により，ベクトル命令を用いて最大 256 要素を一度に処理することができる．本稿では，画像圧縮に用いられる 2 次元離散ウェーブレット変換プログラムを対象とし，OpenMP によるループ並列処理とベクトル命令による SIMD 並列処理を階層的に組み合わせて並列処理を実現する．本手法では，ループアンローリングを適用することにより，ベクトル処理の性能を高めている．本性能評価では，SX-Aurora TSUBASA の 8 コアを利用して，16k 画像を対象とした離散ウェーブレット変換プログラムを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

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 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=20) と q_k(数列長=20) 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(a) のような $n \times n$ の画像 (n=16384) 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=8192) のスケーリング係数 s1x，$n \times k$ のウェーブレット展開係数 w1x を求める．同様に s1x と w1x に対して垂直方向にウェーブレット変換を適用し，図 1(e) 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．

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

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

3　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プログラムに OpenMP とベクトル命令を用いた，階層的並列処理手法について述べる．

3.1　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 SX-Aurora TSUBASA のマルチコア (8 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 により実現する．

†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

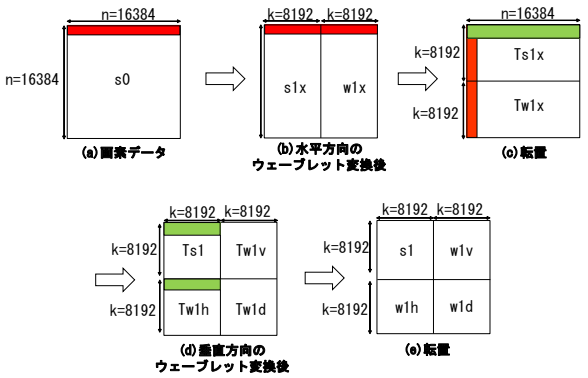

図 1　2 次元ウェーブレット変換の概念．

3.2　ベクトル命令による SIMD 並列処理

SX-Aurora TSUBASA はコンパイラオプションにより，ベクトル命令を利用できる．コンパイル時にベクトル化が可能だと判定した箇所を自動的にベクトル化を適用する．ベクトル命令が処理した演算要素数 (ベクトルの長さ) の平均を平均ベクトル長と呼ぶ．ベクトル命令を効果的に利用するには，ベクトル化した箇所のループ長をある程度の長さ以上であることや，1 つのベクトル命令で処理できる要素数が 256 なので，平均ベクトル長が 256 に近いことが重要である．

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，ドベシィの数列 p_k に対する積和演算を行い，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ィの数列 p_k（数列長=20）との積和を求める際に，ベクトル命令を用いる．図 2(a) のコードでコンパイルを行うと，3 重ループが存在するが，最内側のループに対してベクトル化が適用される．この場合，ベクトル化を適用した箇所のループ長が短いため，ベクトル化の効果が低い．次に図 2(b) のような，最内側のループは全イタレーションをループアンローリングすることで，ループ長と平均ベクトル長が大きくなるため，ベクトル化の効果が高まる．そして，図 2(c) のような，中間側のループは 2 イタレーションをループアンローリングすることで，VE の Last Level Cache(LLC) ヒット率高めることが可能である．これらの方法により，OpenMP によるループ並列処理とベクトル命令による SIMD 並列処理を階層的に行うことが可能になる．

4　SX-Aurora TSUBASA 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．

```
for(j=0; j<8192; j++) {
 for(k=0; k<4096; k++) {
  s1x[j*4096+k]=0.0;
  for(l=0; l<20; l++) {
   s1x[j*4096+k]=s0[j*8192+l+2*k]*p[l];
  }
 }
}
```

(a)ループアンローリング前のコード

```
for(j=0; j<8192; j++) {
 for(k=0; k<4096; k++) {
  s1x[j*4096+k]=0.0;

  s1x[j*4096+k]=s0[j*8192+0+2*k]*p[0]
               +s0[j*8192+1+2*k]*p[1]
                        |
               +s0[j*8192+19+2*k]*p[19];
 }
}
```

(b)ループアンローリング後のコード

```
for (j=0; j<8192; j++) {
 for (k=0; k<4096; k+=2) {
  s1x[j*4096+k  ]=0.0;
  s1x[j*4096+k+1]=0.0;

  s1x[j*4096+k  ]=s0[j*8192+0+2*k]*p[0]
                 +s0[j*8192+1+2*k]*p[1]
                          |
                 +s0[j*8192+19+2*k]*p[19];

  s1x[j*4096+k+1]=s0[j*8192+0+2*(k+1)]*p[0]
                 +s0[j*8192+1+2*(k+1)]*p[1]
                          |
                 +s0[j*8192+19+2*(k+1)]*p[19];
 }
}
```

(c)二重ループアンローリング後のコード

図 2　2 重ループアンローリングの概要.

4.1　SX-Aurora TSUBASA の構成

SX-Aurora TSUBASA は，ベクトルホスト (VH) とベクトルエンジン (VE) を搭載した，ベクトルコンピュータである．VH は x86/Linux ノードであり，主に OS の処理を行う．VE は 8 コアのマルチコア構成になっており，メモリが HBM2 であるため, 高いメモリバンド幅が特長である．これにより，ベクトル命令を用いて最大 256 要素を一度に処理することができる．

本稿で使用した VE Type 10C の理論演算性能 (倍精度) は 2.15TFLOPS，メモリ帯域は 0.75TB/s, 動作周波数は 1.4 GHz，8 つのベクトルプロセッサコアを持ち，LLC のサイズは 16MB である．SX-Aurora TSUBASA 向けのコンパイラは C/C++/Fortran に対応しており，最適化オプションの-O1 以上を指定することで自動ベクトル化が適用される．

表 1　ncc コンパイラのコンパイルオプション.

コンパイルオプション	意味
-O3	レベル 3 の最適化 (自動ベクトル化)
-fno-loop-unroll	ループアンローリングを 許可しない
-mno-vector	自動ベクトル化を 適用しない
-fopenmp	OpenMP 機能を使用する
-ftrace	ftrace 機能用の オブジェクトファイルを生成する

4.2　階層的並列処理による性能評価

本節では，OpenMP によるループ並列処理とベクトル命令による SIMD 並列処置を組み合わせた階層的並列処理による実行結果を示す．本性能評価では，画像圧縮の対象として画像データ (16k:16384*16384) を用いて，SX-Aurora TSUBASA 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 NEC コンパイラによる自動ベクトル化の効果を確認するために，使用したコンパイルオプションは表 1，離散ウェーブレット変換における LLC ヒット率を表 2 に示す．また，16k 画像データに対して提案手法を適用した結果を図 3 に示す.

2 重ループアンローリングを適用することで，LLC ヒット率が上昇して，更なる高速化が可能である．中間側のループアンローリングの展開数が 1 のときの LLC ヒット率は 29.3 %に対して，8 の時は 72.0 %であった．これにより，2 重ループアンローリングなし・ベクトル命令なし・8 コア実行では，実行時間 15220[ms] となり逐次実行比で 7.7 倍であった．次にベクトル命令を加えると，実行時間 11231[ms] となり逐次実行比で 10.4 倍である．最後に 2 重ループアンローリングあり・ベクトル命令あり・8 コア実行では，実行時間は 1752[ms] となり逐次実行比で 66.6 倍であった.

これらの結果から OpenMP による並列処理とベクトル命令による SIMD 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.

表 2　離散ウェーブレット変換における LLC ヒット率.

中間側のループ アンローリングの展開数	1	2	4	8
LLC ヒット率 [%]	29.3	31.7	46.9	72.0

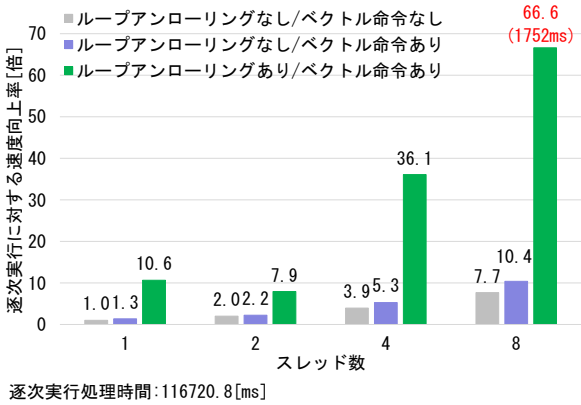

図 3　16k 画像データにおける離散ウェーブレット変換の階層的並列処理.

5　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 によるループ並列処理を適用し，さらに各行のデータとドベシィ数列との積和演算にベクトル命令による SIMD 並列処理を実現する階層的並列処理手法を提案した．SX-Aurora TSUBASA 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，16k 画像データにおいて，66.6 倍の速度向上が得られた．これは 2 重ループアンローリングにより，LLC ヒット率の上昇によるものと考えられる．これらの結果から，2 重ループアンローリングを用いた階層的並列処理の有効性が確認された.

参考文献

[1] SX-Aurora TSUBASA. https://www.hpc.nec/, 2020.

[2] 滝沢寛之. NEC SX-Aurora TSUBASA ではじめるベクトルプログラミング, 2020.

[3] 田處 雄大, 見神 広紀, 細見 岳生, 木村 啓二, 笠原 博徳. OSCAR 自動並列化コンパイラと NEC ベクトル化コンパイラの協調によるベクトル・パーソナルスパコン上での自動ベクトル並列化, 研究報告システム・アーキテクチャ（ARC）,Vol.2020-ARC-240,No.26,pp.1-6,2020.

[4] 中野宏毅, 山本鎭男, 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] OpenMP. https://www.openmp.org/, 2018.