

6S-3

8隣接プロセッサ・アレイによる
並列2-D FFTアルゴリズム

丹野州宣¹, 武田利浩¹, 堀口進²

山形大学

東北大学

1. はじめに

2次元高速フーリエ変換(2-D FFT)は絶対的な計算量が大きいので、高速処理の実現が強く望まれている。本論文では、直接法による2-D FFTを8隣接プロセッサ・アレイ上で実行する逐次転送法と一括転送法と呼ばれる2つの並列ハードウェア・アルゴリズムについて述べる。逐次転送法はプロセッサ間のデータ転送とバタフライ演算を繰り返すことにより2-D FFTを実行する。一方、一括転送法は計算の途中に1回だけビット逆順転送と呼ばれるデータの置換を行う以外はプロセッサ間でのデータ転送を生じない方法である。どちらのアルゴリズムも2次元バタフライ演算の性質と8隣接プロセッサ・アレイの持つ通信能力とを巧みに利用することにより、2-D FFTの高速処理を実現している。

2. 2-D FFTアルゴリズム(直接法)

$N=2^n$ として、 $N \times N$ 個の2次元複素入力データ列を $A_0(i; k)$ で表し、そのフーリエ係数を $G(u; v)$ ($0 \leq i, k, u, v \leq N-1$) とすれば、2次元離散フーリエ変換(2-D DFT)は

$$G(u; v) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} A_0(i; k) W_N^{iu} W_N^{kv} \quad (1)$$

で定義される。ただし、 $j = \sqrt{-1}$ として、

$$W_N = \exp(-2\pi j / N) \quad (2)$$

である。

この2-D DFTを高速処理する直接法による2-D FFTアルゴリズムは、次のように定義される2次元バタフライ演算を繰り返すことにより得られる。 $A_k(i; j)$ 、 $0 \leq i, j \leq N-1$ 、を第 k ($1 \leq k \leq n$) ステージの2次元バタフライ演算の結果とし、 i, j の2進数表現を

$$i = i_{n-1}2^{n-1} + i_{n-2}2^{n-2} + \dots + i_12 + i_0$$

$$j = j_{n-1}2^{n-1} + j_{n-2}2^{n-2} + \dots + j_12 + j_0$$

とすると、 $A_k(i, j)$ は

$$A_k(i, j) = A_k(i_{n-1}, \dots, i_{n-k}, \dots, i_0; j_{n-1}, \dots, j_{n-k}, \dots, j_0)$$

$$= A_{k-1}(i_{n-1}, \dots, 0, \dots, i_0; j_{n-1}, \dots, 0, \dots, j_0)$$

$$+ A_{k-1}(i_{n-1}, \dots, 0, \dots, i_0; j_{n-1}, \dots, 1, \dots, j_0) W_N^{B_k}$$

$$+ A_{k-1}(i_{n-1}, \dots, 1, \dots, i_0; j_{n-1}, \dots, 0, \dots, j_0) W_N^{C_k}$$

$$+ A_{k-1}(i_{n-1}, \dots, 1, \dots, i_0; j_{n-1}, \dots, 1, \dots, j_0) W_N^{D_k} \quad (3)$$

になる。ただし、

$$W_N = \exp(-2\pi j / N)$$

$$B_k = W_N^{i_{n-k+1}2^{n-2} + \dots + i_{n-1}2^{n-k}}$$

$$C_k = W_N^{j_{n-k+1}2^{n-2} + \dots + j_{n-1}2^{n-k}}$$

$$W_D = B_k(-1)^{i_{n-k}}$$

$$W_E = C_k(-1)^{j_{n-k}}$$

である⁽¹⁾。

3. 8隣接プロセッサ・アレイの構成

8隣接プロセッサ・アレイのPE(処理要素)は8本のリンクを持ち、このリンクを介して隣接PEと通信する。一般に、2次元アレイにおいてはアレイの端に位置するPE間を包絡(wraparound)で結合する場合がトラス状結合で、しない場合がセルラ状結合となる。 $P=2^{2m}$ 、 $m \geq 0$ とし、 $P = \sqrt{P} \times \sqrt{P}$ 個のPEからなる8隣接プロセッサ・アレイを仮定し、各PEをPE(行番号; 列番号)で表す。各PEのリンクは互いに独立に隣接するPEと半二重の双方向通信が可能であると仮定する。また、各PEはローカルメモリ

をもち、その容量は2次元データの数とPEの数で割った値に等しいとする。

4. 並列2-D FFTアルゴリズム

4.1 逐次転送法

逐次転送法はセルラ状8隣接プロセッサ・アレイ上に2次元バタフライ演算を直接インプリメントする方法である。したがって、PE間のデータ転送とバタフライ演算を繰り返すことにより2-D FFTが実行される。

4.1.1 8隣接プロセッサ・アレイ上の2次元バタフライ演算

図1(a)(最終ページ参照)に示すように8隣接プロセッサ・アレイ上のPE(x; y)とこのPEからrだけ離れた3つのPE、すなわちPE(x+r; y)、PE(x; y+r)、PE(x+r; y+r)を考える。また、各PEは図示のようにそれぞれ要素(f_0, f_1, f_2, f_3)を格納しているものとする。8隣接プロセッサ・アレイ上でのバタフライ演算を以下に定義する。

(1) PE(x; y)、PE(x+r; y)、PE(x; y+r)、PE(x+r; y+r)は同時にそれぞれの値 f_1, f_2, f_3 をPE(x; y)に送出する。隣接PE間におけるデータ転送時間を τ とすると、この転送に要する時間は $r \cdot \tau$ 時間となる。各転送は並列に、また、各PE同時に行われる。

(2) PE(x; y)は f_0, f_1, f_2, f_3 に対して式(3)で与えられるバタフライ演算を行う。この演算に要する時間を τ_0 とする。

(3) PE(x; y)はバタフライ演算の結果 F_0, F_1, F_2, F_3 を図1(b)に示すように返すものとする。この転送も各方向並列に、また各PE同時に行われ、 $r \cdot \tau_0$ の転送時間を要する。

(1)、(2)、(3)で述べた一連の操作を8隣接プロセッサ・アレイ上のインプレイス型2次元バタフライ演算と呼ぶ。

4.2.2 アルゴリズム

(a) $P=N^2$ の場合

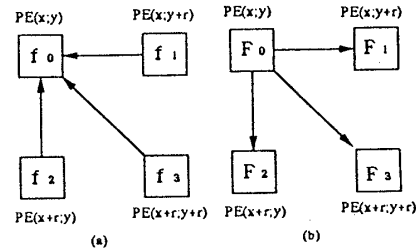


図1 2-Dインプレイスバタフライ演算

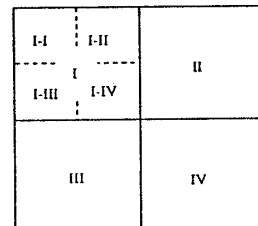


図2 プロセッサ・アレイの再帰的な4分割

Parallel 2-D FFT Algorithms on an Eight-Neighbor Mesh-Connected Processor Array
Kuninobu TANNO, Toshihiro TAKEDA, Susumu HORIGUCHI
1 YAMAGATA UNIV., 2 TOHOKU UNIV.

アルゴリズム1 $N \times N$ ($N=2^m$) 個の2-D FFTすべき信号系列 $A_i(j)$ を、 $P (=2^{2m}) = \sqrt{P} \times \sqrt{P}$ の8隣接プロセッサ・アレイのPE ($i; j$) に対応させて並べる。ただし、 $0 \leq i, j \leq n$ ($m=n$)。次いで、領域I (図2参照) にある $\sqrt{P} / 2 \times \sqrt{P} / 2$ ($P/4$) 個のPEで、 $r=N/2$ として並列に図1に示すインプレッス型2次元バタフライ演算を行う。以下、再帰的に各領域を4分割し、分割した各領域の領域IのPEで、 r を $1/2$ して並列にバタフライ演算を行う。この操作を再帰的に $\log_2 N$ 回 (第 n ステージまで) 実行することで2-D FFTが終了する。

(b) $N < P < N^2$ の場合

各PEに N^2/P 個の入力データを格納し、第 m ステージまではアルゴリズム1と同様にPE間でのデータ転送と2次元バタフライ演算を繰り返し、第 $(m+1) \sim$ 第 n ステージでは各PEに格納されているデータ同志でバタフライ演算を行う。

アルゴリズム2 $N \times N$ 個の2-D FFTすべき信号系列 $A_i(j) = A_0(i_{n-1}, \dots, i_{n-m}, \dots, i_0; j_{n-1}, \dots, j_{n-m}, \dots, j_0)$ を $P (=2^{2m}) = \sqrt{P} \times \sqrt{P}$ の8隣接プロセッサ・アレイ上のPE ($i_{n-1}, i_{n-2}, \dots, i_{n-m}; j_{n-1}, j_{n-2}, \dots, j_{n-m}$) に対応させて並べる。次いで、領域I (図2参照) にある $\sqrt{P} / 2 \times \sqrt{P} / 2$ ($P/4$) 個のPEで、 $r=2^m/2$ として並列に2次元バタフライ演算を行う。使用されるデータは r だけ離れた各PEから順序よく与えられるものとする。これを N^2/P 回繰り返すことにより第1ステージのバタフライ演算が終了する。以下、再帰的に各領域を4分割し、分割した各領域の領域I (領域I-Iなど) のPEで、 r を $1/2$ し、繰り返し回数は $1/4$ にして並列に2次元バタフライ演算を行う。この操作を再帰的に第 m ステージまで繰り返す。第 $(m+1) \sim$ 第 n ステージでは各PEに格納されているデータでバタフライ演算を行うことで2-D FFTが終了する。

4.2 一括転送法

一括転送法はPE間のデータ転送をまとめて行うことにより通信に要するオーバーヘッドをできるだけ少なくし、処理速度を上げようとするアルゴリズムである。

4.2.1 アルゴリズム

一括転送法による2-D FFTアルゴリズムは次の3段階、

- (1) 第1～第 $(n-m)$ ステージまで2-D FFTを行う、
 - (2) ビット逆順転送を行う、
 - (3) 第 $(n-m+1) \sim$ 第 n ステージの2-D FFTを行う、
- からなる。このアルゴリズムの特徴は(1)と(3)での2-D FFTは各PEに格納されたデータ間で行われ、データの転送は一括して(2)のビット逆順転送で行われることである。したがって、効率の良いビット逆順転送アルゴリズムが必要である。なお、各プロセッサには N^2/P 個のデータが格納されるが、初期データの配列はビット逆順で与える必要がある。

4.2.2 ビット逆順転送

第 $(n-m+1)$ ステージ以降の演算を同じPEに格納されているデータ同志の演算となるようにするためビット逆順転送と呼ばれるPE間のデータの入替が必要となる。これは各PEに格納されている N^2/P 個のデータを各PE宛に N^2/P 個づつデータを転送することを意味する。この転送に要する時間がこのアルゴリズムの処理速度を決定する。第 $(n-m)$ と第 $(n-m+1)$ ステージの間に生じるこの入替は、PE ($i_0, \dots, i_{n-1}; j_0, \dots, j_{n-1}$) に格納されているデータ $A_{n-m}(i_{n-1}, \dots, i_0; j_{n-1}, \dots, j_0)$ をPE ($i_{n-1}, \dots, i_{n-m}; j_{n-1}, \dots, j_{n-m}$) に転送することより行われる。

(a) トーラス状の場合

トーラス型では各PEは、データ入力、出力にそれぞれ4本のリンク (半二重の双方向通信) を用いることができ並列に $4P$ 個の単

位転送が可能である。各PEのリンクに0～7まで番号を付す。次のような転送アルゴリズムを提案する。

転送アルゴリズム

- (1) リンク0～3を送信リンクとし、リンク4～7を受信リンクとして転送距離が \sqrt{P} から1のデータまで順に転送を行う。このとき受信したデータがそのPE宛のものでないときは次の隣接した適当なPEに転送しなければならない。このようなデータを中継データと呼ぶ。中継データも含めて転送距離が長いデータから送るものとする。全てのデータの送受信を終了したら(2)の動作に移る。
- (2) リンク4～7を送信に、0～3を受信に用い(1)と同じ手順で送受信を行う。全ての送受信が終了したときビット逆順転送が完了する。

なお、この転送時間はビット逆順転送の下界に非常に近い値になることが示される。

(b) セルラ状の場合

セルラ状アレイについては(a)で提案したような8隣接プロセッサ・アレイの通信能力を生かす転送アルゴリズムが見つかっていない。したがって、セルラ状4隣接プロセッサ・アレイに提案されている文献(2)のアルゴリズムを採用する。

5. 2-D FFTアルゴリズムの評価

各アルゴリズムの処理時間 T_{tot} を算出し、その結果をまとめて示す。

(a) 逐次転送法

(1) $P=N^2$ の場合

$$T_{tot} = 2(N-1)\tau_r + \log_2 N \tau_b \quad (4)$$

(2) $N < P < N^2$ の場合

$$T_{tot} = \frac{N^2}{P} (\sqrt{P}-1)\tau_r + \log_2 P \tau_b + \frac{(\log_2 N - \log_2 P) N^2}{4P} \tau_b \quad (5)$$

(b) 一括転送法

$$T_{tot} = \frac{N^2 \log_2 N}{4P} \tau_b + \begin{cases} \frac{P+2}{12P\sqrt{P}} N^2 \tau_r \text{ (トラス状)} \\ \frac{N^2}{\sqrt{P}} \tau_r \text{ (セルラ状)} \end{cases} \quad (6)$$

6. むすび

2次元バタフライ演算を行うことで2-D FFTを計算する直接法を8隣接プロセッサ・アレイ上で実行する逐次転送法と一括転送法と呼ばれる2つの並列ハードウェア・アルゴリズムについて論じた。その結果、

- (1) 逐次転送法で $P=N^2$ としたときが最も高速で、処理時間は $O(N)$ となるが、PEの使用率は25%と低い、(2) 逐次転送方式で $N < P < N^2$ の場合は使用率は向上するが、処理時間は $O(N^2 \log_2 N)$ となる。(3) 一括転送方式でトラス状の場合も処理時間は逐次転送方式 ($N < P < N^2$) の場合と同じで $O(N^2 \log_2 N)$ となる。しかし、配線などのハードウェア量を考えると逐次転送法が有利と考えられる、などが明らかとなった。

今後、セルラ状アレイにおける効率のよいビット逆順転送法やデータの入出力などについても考察を進めたい。

文献

(1) J. S. Lim, "Two-Dimensional Signal and Image Processing," Prentice Hall (1990).
 (2) 国枝、伊藤、"メッシュ結合マルチプロセッサシステムにおける2次元FFTアルゴリズム"、電情通学論、Vol. J71-A, NO. 7, pp. 1424-1431 (1988).