

多次元FFTの並列処理

6T-5

吉原 郁夫¹⁾ 村松 晃¹⁾ 田中 輝雄²⁾
 (株)日立製作所システム開発研究所¹⁾、中央研究所²⁾

1. はじめに

並列計算機¹⁾²⁾³⁾による応用プログラムの高速処理を目的として、種々の数値シミュレーションで共通に使われる高速算法の並列化を検討している。本報で高速フーリエ変換(FFT)を、別報⁴⁾で共役勾配法(ICCG法)を報告する。

FFTは信号処理、画像処理、偏微分方程式などさまざまな分野で利用されている。取り扱う問題が複雑化、大規模化するにつれ、多次元FFTが重要になってきた。多くの応用に於いて多次元FFTは一次元FFTの反復適用で代替できるが、並列計算機で処理する場合、二次元問題は一次元問題にない新たな工夫を要する。しかし、三次元以上の問題は二次元の手法で対処できるので、二次元問題の検討が基本的である。

2. 二次元FFT

2.1 FFTとその処理時間

データ長 $N (= 2^n)$ のFFTは $\log N$ (以下対数の底は2)段のバタフライ演算からなり、各段では、一對の複素データ U 、 U' から一對の複素データ V 、 V' が生成される。

$$\begin{cases} \text{DO} \\ V = U + \omega * U' \\ V' = U - \omega * U' \end{cases} \quad (1)$$

ただし、 ω は捻り因子

(1)式のDOLoop1回当たりの命令数を a 、マシンサイクルを τ とすればFFTの計算時間は、

$$t_f = N \log N \ a \ \tau \quad (2)$$

と表せる。ただし、捻り因子作成の負荷、および外側DOLoop負荷等は省略。

2.2 二次元FFT

二次元FFTは次のように、2回の一次元FFTに帰着する。

$$y_{kn} = \sum_{km} \exp\left[-\frac{2\pi(k\ell + mn)}{N}\right] x_{km}$$

$$= \sum_m \exp\left[-\frac{2\pi mn}{N}\right] \left\{ \sum_k \exp\left[-\frac{2\pi k\ell}{N}\right] x_{km} \right\} \quad (3)$$

$N \times N$ の二次元データの処理には、 N 組のFFTが2回必要であり、シリアル処理での計算時間は $2Nt_f$ となる。

3. 二次元FFTの並列処理

$N \times N$ の二次元データを、 $M (= 2^m)$ 台の要素プロセッサ(PE)から成る並列計算機²⁾³⁾で処理する方法を考えて見よう。

一般に $M \ll N$ であり、1台のPEに多くのデータ点を割付ける。例えば、図1上のように各PEに行単位のデータを割付ける方法と、同図下のようにブロック単位に割付ける方法が考えられる。行単位割付けでX方向FFTを行なう場合、全データは自己PE内にあるから転送は生じない。しかし、Y方向FFTを行なう場合、バタフライ演算のうち $\log M$ 段は他PEにあるデータを参照せざるを得ず、転送が必要となる。ブロック割付けの場合は、Y方向FFT時の転送は減るが、同じ回数だけX方向の転送が生じ、結局総転送量は変わらない。即ち、転送負荷は同じである。以下、行単位割付けを例に処理方法及び処理時間の評価を行なう。

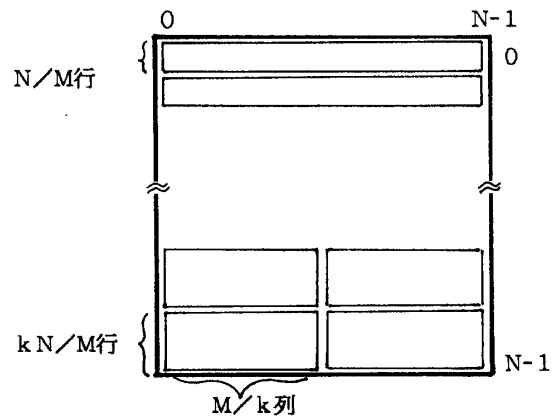


図1 データの割付け

3.1 データ転置方式

バタフライ演算時のデータ転送を避ける最も簡便な方法は、Y方向FFTを行なう前にデータを転置することである。これをコーナーターンと言う。

Parallel Processing of Multidimensional FFT
 Ikuo YOSHIHARA¹⁾, Akira MURAMATSU²⁾, Teruo TANAKA²⁾
¹⁾Systems Development Lab., ²⁾Central Res. Lab., Hitachi Ltd.

データ領域を図2に示す小領域(小行列)に分割する。PE間転送を用いれば、転置 $A_{j,t} \leftarrow A_{t,j}^T$ は並列度Mで実行可能である。具体的には、小行列をM個ずつのグループに分け、番号1~M-1に従って、次のDOループを順次実行すればよい。

```
DO I=1,N/M ; DO J=1,N/M
  A(N2-J,M2-I)=A(M1+I,N1+J) (4)
```

この演算は、ストアが他PEに対し行なわれる点が通常の代入文と異なる。転送するデータ量は $(N/M)^2 * (M-1)$ である。(4)式の1ループ当りの命令数をb、2ワード(複素数は実数2個相当)のデータ転送の命令数をcとすれば、転置に要する時間および全処理時間は、次の通りである。

$$t_1 = \frac{N^2}{M^2} (M-1) \max[b, c] \tau \quad (5)$$

$$\therefore T_1 = 2 N t_f / M + t_1$$

$$= \frac{N^2}{M} \left\{ 2 \log N a + \frac{M-1}{M} \max[b, c] \right\} \tau \quad (6)$$

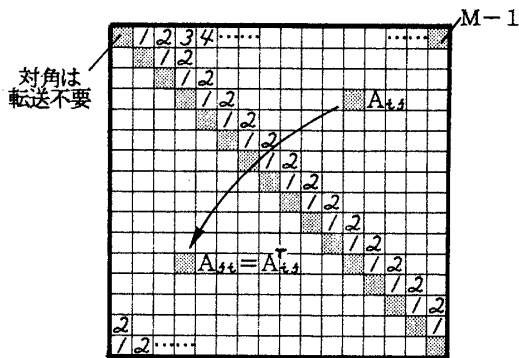


図2 データ転置の順序

3.2 データ固定方式

前節の方法は、マシンの並列度を最大限に引き出してはいるが、演算と転送が時間的オーバーラップなしに行なわれる。これに対しバタフライ演算時にデータを送れば総転送量は多くなるものの、転送は演算と同時に実行できる部分があるため、全処理時間は短くなる可能性がある。

データ割付けを固定したまま高速化を図るには、データ転送量を減らすことが重要である。このため、自PE内のデータだけを使い結果の一つだけを他PEに転送すればよいように配列のアドレッシングに工夫したプログラムを組む。このようにすれば、1バタフライ演算あたり2ワードの転送で済む。Y方向FFTと全処理に要する時間は次の通りである。

$$t_2 = \frac{N^2}{M} \{ (\log N - \log M) a + \log M \max[a + \delta, c] \} \tau \quad (7)$$

$$\therefore T_2 = N t_f / M + t_2$$

$$= \frac{N^2}{M} \{ 2 \log N a + \log M \max[\delta, c - a] \} \tau \quad (8)$$

ただし、 δ は受信のために増える命令数。

3.3 二方式の比較

T_1 , T_2 の長短を比較する。

$$T_1 - T_2 = \frac{N^2}{M} \left\{ \frac{M-1}{M} \max[b, c] - \log M \max[\delta, c - a] \right\} \tau \quad (9)$$

ここで、通常は $\delta \ll a$, $b \ll a$ であるから、転送時間が演算時間より短く ($c < a$)、PE台数が極端に多くなければ、データ固定方式の方が高速と云える。実際、別報のシミュレータ²⁾を用いた検討によれば、図3に示すように、データパラレル並列計算機¹⁾³⁾ではこの関係が成立し、データ固定方式の方が処理時間が短い。

一方、データ転送時間の方が長く ($c > a + \delta > a > b$)、しかもPE台数が多い場合は、データ転置方式の方が有利になることもある。

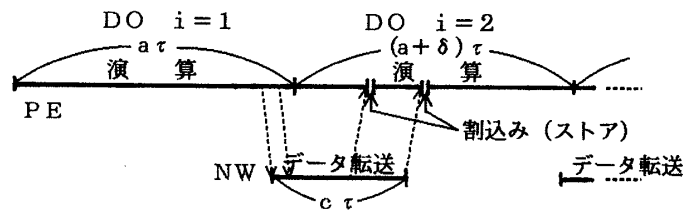


図3 演算と転送のオーバーラップ

4. おわりに

二次元FFTの並列処理方法について具体的に検討し、PE間データ転送速度の速い並列計算機においては、汎用機で必要とされたコーナーターンなしに効率よく処理できる見通しを得た。

多次元FFTの並列処理が可能となれば、高精細な画像処理、スペクトル法による偏微分方程式求解など、多くの応用が拓ける。

5. 参考文献

- 1) 村松、他：データパラレル計算モデルの提案、第36回全大、1988
- 2) 濱中、他：並列計算機H2Pのシステム構成、第38回全大、1989
- 3) 村松、他：分散型共有メモリを持つデータパラレル並列計算機のアーキテクチャ、第38回全大、1989
- 4) 前田、他：データパラレル計算機によるICCG法の並列処理、第38回全大、1989