

アレイ計算機におけるPE間結線構造とデータ転送回数について

成瀬 正 中野 治 雨宮 真人

(日本電信電話公社 武蔵野電気通信研究所)

1. はじめに

超高速計算を目的として多数の演算装置(Processing Element, PE)を結合させた並列処理計算機では一般にPE間のデータ転送回数が多くなり, そのためPE間の結線構成が性能及び実装の点から問題となる。特に, 適用領域を絞った専用計算機では応用問題に現れるデータ転送の形態に適合させて結線構成を設定することが性能/価格の点から重要である。

現在提案されている転送系は, PEを何段かのスイッチを介して結合した多段構成転送系(Multistage Network)とPEを直接結線した単段構成転送系(Single-stage Network)に区別される⁽¹⁾。

転送時間の点からみると, 多段構成転送系では, 全てのPE間での転送時間が平均化されるという特長を持つ反面, 特定のPE間での転送が多いような場合には, その転送遅れが問題となる。一方, 単段構成転送系では結線のないPE間での転送はいくつかのPEを中継して行われるため転送時間が中継PE数に依存するという欠点をもつが, データの転送形態と結線構成が適合した場合には直接転送の比重が高くなり高速転送が期待できる。

いくつかの基本的と思われる算法のデータ参照形態を調べてみると規則的な一定のパターンがあり, そのパターンは何種かに整理できる。したがって, このパターンに適合した結線構造をもつ単段構成転送系が実現できれば, このような算法に十分な効果が期待できる。しかし完全に適合させようとすると結線数が膨大となり実現上困難となるため, 現実には単純な結線で代用することになる。そのためには結線構造に對

する転送時間の評価を行うことが重要な問題となる。

本稿ではこの観点から, まずいくつかの基本的と思われる問題を整理し, 次に単段構成転送系の結線構造の評価を行い, 本稿で提案する間引き対数構造結線の有効性を示す。

2. データ参照形態

2.1 基本的問題のデータ参照形態

本節では総和問題のように n 個のデータが演算によって相互に関係づけられる問題を考える。このような問題を並列に実行する算法では演算すべきデータの参照形態に特有の型がある。そこで以下では, データの流れを示す図(以後データ流れ図という)を用いてこの型を明確にする。なお, データ流れ図では節点にそこで行われる演算を陽に記述する(データが素通りする場合は空白とする)。また, n は簡単のため 2^k とし, 対数の底は2とする。

1) 総和⁽²⁾, 最大値, 最小値

$$\sum_{i=0}^{n-1} x_i$$

$$\max_{0 \leq i \leq n-1} (x_0, \dots, x_i, \dots, x_{n-1})$$

$$\min_{0 \leq i \leq n-1} (x_0, \dots, x_i, \dots, x_{n-1})$$

これらは次のアルゴリズムで並列に計算できる。($x_i^{(0)} = x_i$, ($i=0, \dots, n-1$) とする。)

[アルゴリズムA I]

step 0. $l = 0$

step 1. $x_i^{(l+1)} = x_i^{(l)} \text{ OP } x_{(i+2^l)}^{(l)} \text{ mod } n$
 $i = 0, 1, \dots, n-1$

ここで, OP はそれぞれの演算に對應

する。

step 2. $l=l+1$, $l < \gamma$ ならば step 1へ。
 $l \geq \gamma$ ならば 終り。

図1にこのデータ流れ図を示す。

アルゴリズム I と図1から明らかのように、これはデータ数 n に対して $\log n$ 段のフェイズで並列に計算できる。データ参照形態の特徴は、

- i) 2^l のベキ乗 (2^l) 隔てたデータ間の参照だけである。
- ii) 一つのフェイズではデータ参照距離は一定である。ここで距離とは n を法として計算した添字の差をいう。
- iii) 2^l 隔てたデータ参照のフェイズは各 l ($0 \leq l \leq \gamma-1$) に対して1回づつ現れる。

である。

この特徴を持ったデータ参照形態を タイプ I とする。

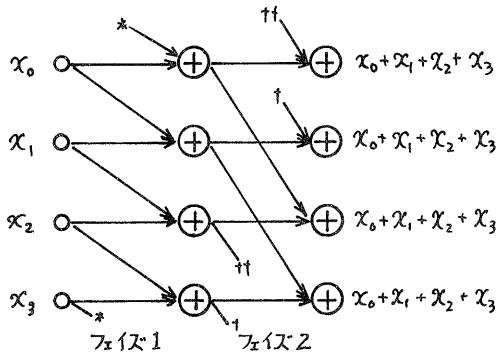


図1. 総和問題 ($n=4$)

2) ベクトルの内積⁽²⁾

ベクトル $x = (x_0, \dots, x_i, \dots, x_{n-1})$,
 $y = (y_0, \dots, y_i, \dots, y_{n-1})$ の内積は、

$$x \cdot y = \sum_{i=0}^{n-1} x_i \cdot y_i$$

で与えられる。

各 $x_i \cdot y_i$ が計算されていけば、これは総和問題に帰着する。 x_i と y_i を同一節点に与えておけば、積の計算で他の節点のデータ参照はない。したがって、

て、データ参照形態はタイプ I である。

3) 多項式⁽²⁾

$$\sum_{i=0}^{n-1} a_i \cdot x^i$$

これはベクトル $x = (1, x, \dots, x^{n-1})$,
 $a = (a_0, a_1, \dots, a_{n-1})$ の内積と考えることができる。

x はベクトル $(1, x, \dots, x)$ から次のように並列に求まる。

[アルゴリズム II]

step 0. $l=0$. $(x_0^{(0)}, x_1^{(0)}, \dots, x_{n-1}^{(0)}, \dots)$
 $x_{n-1}^{(0)} = (1, x, \dots, x, \dots, x)$

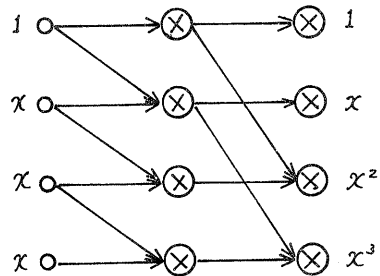
step 1. 各 i に対し次の計算をする。

$$i \geq 2^l \text{ ならば } x_i^{(l+1)} = x_i^{(l)} \times x_{i-2^l}^{(l)}$$

$$i < 2^l \text{ ならば } x_i^{(l+1)} = x_i^{(l)}$$

step 2. $l=l+1$, $l < \gamma$ ならば step 1へ
 $l \geq \gamma$ ならば 終り

図2にデータ流れ図を示す。フェイズ数は $\log n$ である。明らかにこのデータ参照形態もタイプ I である。



フェイズ1 フェイズ2
 図2. 多項式の計算 ($n=4$)
 (ベキを計算する部分)

4) ソーティング⁽³⁾

代表的な並列ソーティングアルゴリズムに odd-even merge sort がある。このアルゴリズムは文献(3)に詳しいのでここでは省略する。図3にデータ流れ図を示す。

この演算は $\log n (\log n + 1) / 2$ 段のフェイズで並列に実行できる。データ参照形態の特徴は、1) で述べた i), ii) の特徴と、

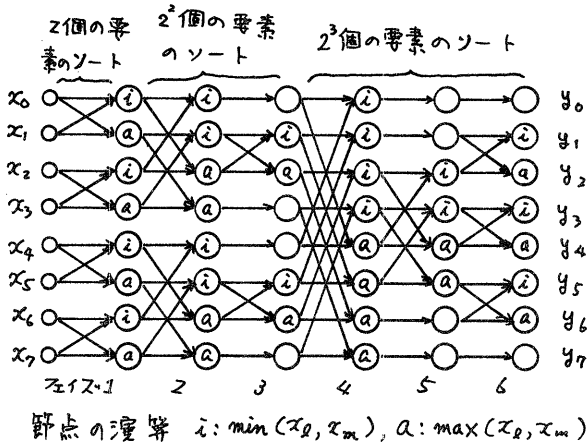


図3. 並列ソート (n=8)

iii) 2^l 隔てたデータ参照のサイズは各 l ($0 \leq l \leq r-1$) に対して $\log n - l + 1$ 回現れる。

である。

この特徴を持つデータ参照形態を タイプII とする。

5) FFT (Fast Fourier Transform) ⁽⁴⁾

FFT は (a) バタフライ演算と (b) 順序復元から構成される。

(a) バタフライ演算

バタフライ演算では次の式を各 l に対して順次計算する。

$$\left. \begin{aligned} x_i^{(l+1)} &= x_i^{(l)} + W^p x_j^{(l)} \\ x_j^{(l+1)} &= x_i^{(l)} - W^p x_j^{(l)} \end{aligned} \right\} l=0, 1, \dots, r-1$$

ここで、

$p = p(l, i)$ (詳細は文献(4)参照)

$$W = \exp(-j2\pi/n), \quad n = 2^r$$

$$\left. \begin{aligned} k \cdot 2^{r-l} \leq i < (k + \frac{1}{2}) 2^{r-l} \\ j = i + 2^{r-l+1} \end{aligned} \right\} k = 0, 1, \dots, 2^l - 1$$

このデータ流氷図を図4に示す。サイズ数は $\log n$ である。図4のデータ参照形態は 1) と同じ特徴を持つ。したがって、タイプIである。

(b) 順序復元

ここでは添字を2進数で考える (LSBをオのビット, MSBをオ $r-1$ ビットとする)。添字の小さい順に並んだデータ x_i , ($0 \leq i \leq 1 \dots 1$) をビット逆転順 (添字のビット順序を逆転すると添字の小さい順に並ぶよう) に並べかえる。

この並べかえは次のアルゴリズムで並列に実行できる。

[アルゴリズムIII]

step 0. $l = \lfloor r/2 \rfloor - 1$ ($\lfloor \cdot \rfloor$ は α の整数部分)

step 1. 各 i に対して次の計算をする。

i) i のオ ($r-1-l$) ビットが0かつオ l ビットが1ならば、 $x_j^{(l-1)} = x_i^{(l)}$ 。ただし、 $j = i + 2^{r-1-l} - 2^l$ 。

ii) i のオ ($r-1-l$) ビットが1かつオ l ビットが0ならば $x_j^{(l-1)} = x_i^{(l)}$ 。

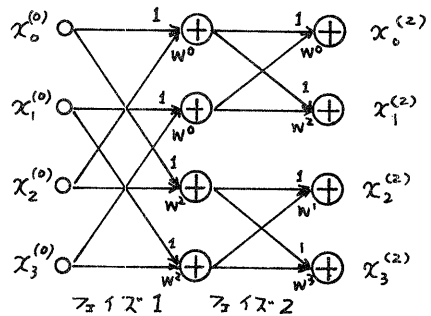


図4. バタフライ演算 (n=4)

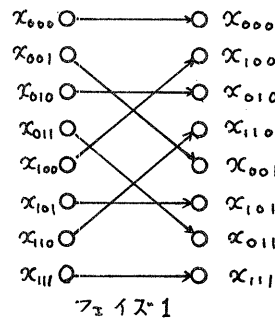


図5 順序復元 (n=8)

$$i \neq k, j = i - 2^{r-1-l} + 2^l.$$

iii) 上の 2 つのいずれでもなければ $x_i^{(l-1)} = x_i^{(l)}$.

step 2. $l = l - 1$, $l \geq 0$ ならば "step 1"へ。
 $l < 0$ ならば "終り".

step 1 の意味は, l 回目の繰返しにおいて添字の差の絶対値が $2^{r-1-l} - 2^l$ でかつオ ($r-1-l$) ビットとオ 1 ビットの値が異なるデータを入力かえることである。

データ流氷図を図 5 に示す。フェイズ数は $\lfloor \log n / 2 \rfloor$ である。データ参照形態の特徴は,

- i) $2^{l_1} - 2^{l_2}$ 隔てたデータ間の参照だけである。ただし, $l_1 - l_2 = r - 1 - 2l$
- ii) 一つのフェイズではデータ参照距離は一定である。
- iii) $2^{l_1} - 2^{l_2}$ 隔てた参照のフェイズは各 $l_1 - l_2 = r - 1 - 2l$ ($0 \leq l \leq \lfloor \log n / 2 \rfloor$) に対して 1 回づつ現れる。

である。

この特徴を持つデータ参照形態をタイプ III とする。

6) マトリクスの積

マトリクス $A = (a_{ij})$, $B = (b_{ij})$, ($0 \leq i, j \leq n-1$) の積 $C = (c_{ij})$ は,

$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} \cdot b_{kj}, \quad 0 \leq i, j \leq n-1$$

で与えられる。

各 $a_{ik} \cdot b_{kj}$ が計算されていれば n^2 組の Σ は並列に計算してよい。種はベクトルの内積と同様, 他の節点のデータを参照しないため, 以下では Σ について検討する。

n 組の $\sum_{j=0}^{n-1} d_{ij}$, ($j=0, \dots, n-1$) は次のアルゴリズムで並列に計算できる。

[アルゴリズム 4 IV]

加算ベクトル $(x_0^{(l)}, \dots, x_i^{(l)}, \dots, x_{n-1}^{(l)})$
被加算ベクトル $(y_0^{(l)}, \dots, y_i^{(l)}, \dots, y_{n-1}^{(l)})$
作業用マトリクス $(z_0^{(l)}(j), \dots, z_i^{(l)}(j), \dots, z_{n-1}^{(l)}(j))$

$$l = 0, \dots, r-1; j = 0, 1, \dots, n-1$$

とする。

step 0. $(z_0^{(0)}(j), \dots, z_i^{(0)}(j), \dots, z_{n-1}^{(0)}(j)) = (d_0, \dots, d_j, \dots, d_{n-1}), j = 0, \dots, n-1$

step 1. $l = 0$

step 2. $rep = n / 2^l$

step 3. $k=1$ から rep まで次のサステラ, γ (step 3.1 ~ step 3.3) を繰返す。

step 3.1 各 i に対し, i を 2 進数とみて,

i) オ 1 ビットが "0" ならば,

$$x_i^{(l)} = z_i^{(l)}(2k), y_i^{(l)} = z_i^{(l)}(2k-1)$$

ii) オ 1 ビットが "1" ならば

$$x_i^{(l)} = z_i^{(l)}(2k-1), y_i^{(l)} = z_i^{(l)}(2k)$$

step 3.2 $x_{(i-2^l) \bmod n} = x_i^{(l)}, i = 0, \dots, n-1$

step 3.3 $z_i^{(l+1)}(k) = x_i^{(l)} + y_i^{(l)}, i = 0, \dots, n-1$

step 4. $l = l + 1$, $l < r$ ならば "step 2" へ。
 $l \geq r$ ならば "終り".

n^2 組の Σ はこれを n 回繰返す。

データ流氷図を図 6 に示す。フェイズ数は $n-1$ である。データ参照の特徴は, 1) で述べた i), ii) の特徴と,

iii). 2^l 隔てたデータ参照のフェイズは各 l ($0 \leq l \leq r-1$) に対して $n/2^{l+1}$ 回現れる。

である。

この特徴を持つデータ参照形態をタイプ IV とする。

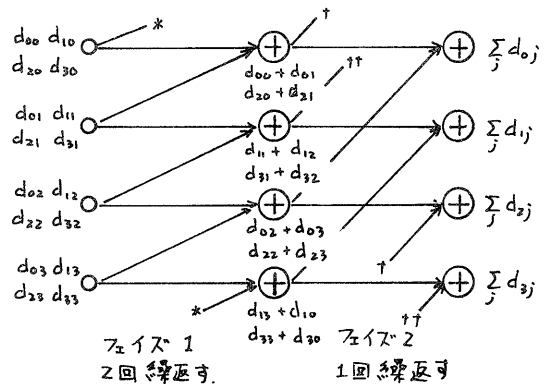


図 6. n 組の総和 ($n=4$)

以上、各種向題のデータ参照形態は転送回数評価の観点からはタイプI～タイプIVの4種類に分類される。

2.2 データ流れ図とPEのマッピング

データ流れ図に現れるアルゴリズムの並列性を反映したマッピングをする。具体的には、各フェイズの節点の演算を各PEで並列に実行するようにマッピングする。

このとき、データ参照に伴いPE間のデータ転送が生じるが、直接結線の無いPE間の中継転送による転送回数増加の評価が以下の論点である。

3. 評価対象転送系の結線構成

本稿では、単段構成転送系を評価対象とし、多段構成転送系は考えない。単段構成転送系の提案は多数あるが⁽⁴⁾、ここでは前章で述べたデータ参照形態に適合した転送系を考える。すなわち、均質な結線構成を料った完全結線、対数構造結線、向引き対数構造結線、隣接PE間結線の各転送系を対象とする。以下では次を前提として各転送系を説明する。

- i) PEはN台とし、 $N = 2^r$ とする。
- ii) PEは0からN-1まで番号付ける。
- iii) 直接結線のあるPE間では双方向のデータ転送ができる。(一時刻には一方向の転送しかできない。)
- iv) 隣接するPEとは PE_i と $PE_{(i+1) \bmod N}$ なるPEをいう。

v) PE間の距離とは、PE番号の差(Nを法として計算する)をいう。

(a) 完全結線

距離がN/2以下の任意のPE間を結線した結線構成(表1)。N/2より大きい距離の結線は、N/2より小さい距離の結線と表裏の関係にあるので不要

(b) 対数構造結線⁽⁵⁾

距離が 2^p ($p=0, \dots, r-1$)のPE間を結線した結線構成(表1)。

(c) 向引き対数構造結線

距離が 2^p ($p=j_0, \dots, j_{m-1}; j_0 < \dots < j_{m-1} < r, j_0=0$)のPE間を結線した結線構成(表1)。任意のPE間で転送可能とするためには $j_0=0$ 、すなわち隣接するPE間の結線は不可欠である。

(d) 隣接PE間結線

隣接するPE間を結線した結線構成(表1)。

各転送系のPEあたりの結線数(M)を表1に示す。向引き対数構造結線は向引きの仕方によ、r PEあたりの結線数が異なる。

対数構造結線と隣接PE間結線は向引き対数構造結線の特別な場合と考えることができる。

表1. 評価対象転送系の構造

結線構成	結線するPE	PEあたり結線数(M)
完全結線	$PE_i - PE_{(i+l) \bmod N}$ $l=1, \dots, N/2$ $i=0, \dots, N-1$	N/2
対数構造結線	$PE_i - PE_{(i+2^p) \bmod N}$ $p=1, \dots, r-1$ $i=0, \dots, N-1$	r
向引き対数構造結線	$PE_i - PE_{(i+2^p) \bmod N}$ $p=j_0, \dots, j_{m-1}; j_0 < \dots < j_{m-1} < r$ $i=0, \dots, N-1; j_0=0$	$1 \sim r$
隣接PE間結線	$PE_i - PE_{(i+1) \bmod N}$ $i=0, \dots, N-1$	1

4. 転送回数の評価

4.1 評価の前提

前述の各転送系に対し，2章で述べた各タイプの参照形態の転送回数を評価する。以下では直接結線のあるPE間の転送を1回と数える。

2章の検討から距離 2^l の転送と距離 $2^{l_1}-2^{l_2}$ の転送を考慮すればよく，以下ではこれをできる限り少ない(中継)転送回数で実行する転送手順を考える。

完全結線ではこれらの転送はその構造から1回でできる。その他の転送系では次の手順で転送する。転送系の結線構成を $(j_0, j_1, \dots, j_i, \dots, j_{M-1})$ で表す。ここで， $j_0 < j_1 < \dots < j_i < \dots < j_{M-1} < \nu$ ， $j_0 = 0$ であり， j_i は距離 2^{j_i} のPE間の結線を意味する。

(a) 距離 2^l の転送

$j_i \leq l$ なる最大の j_i 結線を利用する(図7)。これを j_m とすれば，転送回数は 2^{l-j_m} 回である。明らかにこの転送手順は距離 2^l の転送を最小実行回数で実行する。



数字はPE番号

図7. $(j_0, j_1) = (0, 1)$ の場合のPE₀からPE₄への転送。 j_1 結線を利用してPE₀→PE₂→PE₄と転送する。

(b) 距離 $2^{l_1}-2^{l_2}$ の転送

次の2通りの手順が考えられるが，ここでは前者の手順を採用する。

(b1) $2^{l_1}-2^{l_2} = 2^{l_2+k} + \dots + 2^{l_2+k} + \dots + 2^{l_1-1}$ であるから，距離 2^{l_2+k} の転送を $k=0$ から l_1-l_2-1 まで順次行う。

(b2) まず距離 2^{l_1} の転送をし，次いで逆方向に距離 2^{l_2} の転送をする。

(b1)の転送は次のように行う。

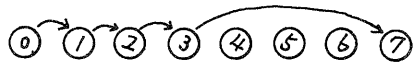
i) $l_2 \geq j_{M-1}$ ならば， j_{M-1} 結線のみを利用する。転送回数は $2^{l_1-j_{M-1}} - 2^{l_2-j_{M-1}}$ 回

ii) $l_1 \leq j_1$ ならば， j_0 結線のみを利用

する。転送回数は $2^{l_1}-2^{l_2}$ 回。

iii) i), ii)以外のときは， $l_1 > j_i$ なる最大の j_{i_1} と $l_2 \geq j_{i_2}$ なる最大の j_{i_2} が存在する。これらをそれぞれ j_{m_1}, j_{m_2} として， $j_{m_2}, j_{m_2+1}, \dots, j_{m_1}$ の各結線を利用して転送する。ただし，各々の 2^{l_2+k} の転送は $l_2+k \geq j_{m'}$ ($m_2 \leq m' \leq m_1$)なる最大の $j_{m'}$ 結線を用いる(図8)。転送回数は $2^{l_1-j_{m_1}} + 2^{j_{m_1}-j_{m_1-1}} + \dots + 2^{j_{m_2+1}-j_{m_2}} - (m_1-m_2) - 2^{l_2-j_{m_2}}$ 回

(b1), (b2)はいずれも距離 $2^{l_1}-2^{l_2}$ の転送を最小転送回数で実行する手順ではないが，転送制御の容易さからは(b1), (b2)のいずれかを採用するのがよいと考える。



数字はPE番号

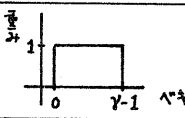
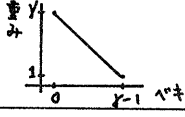
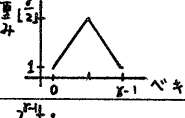
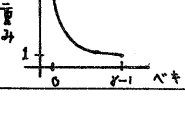
図8. $(j_0, j_1) = (0, 2)$ の場合のPE₀からPE₇への転送。 $2^3-2^0 = 2^2+2^1+2^0$ であるから，まず j_0 結線によりPE₀→PE₁→PE₂→PE₃と転送し(2^0+2^1)，次いで j_1 結線によりPE₃→PE₇と転送する(2^2)。

上述の転送手順に注意して各タイプのデータ転送形態をPE間転送の観点からとらえると，これらはいずれも距離 2^l のPE間転送で各べきの転送を何回実行するかということ，すなわち，各べきに実行回数の重みをつけた転送形態を調べることである。この観点から2章の問題を整理すると表2のようになる。

転送回数は尚引き初級構造結線に対してはこの重み w_i ($i=0, \dots, M-1$)と結線構成 (j_0, \dots, j_{M-1}) から次式によって定められる。

$$T = (w_0 + 2w_1 + \dots + 2^{j_1-1}w_{j_1-1}) + (w_{j_1} + 2w_{j_1+1} + \dots + 2^{j_2-j_1-1}w_{j_2-1}) + \dots + (w_{j_{M-1}} + 2w_{j_{M-1}+1} + \dots + 2^{\nu-1-j_{M-1}}w_{\nu-1})$$

表2. 転送の観点からみたデータ参照形態

データ参照形態	ベキの重み ($\gamma = \log n$)	出現する問題
タイプ I	 一様	総和, 最大値, 最小値, ベクトルの内積, など
タイプ II	 線型に減少	ソーティング
タイプ III	 三角形	ビット逆転順の整列
タイプ IV	 指数関数的に減少	マトリクスの乗算

完全結線についてはフェイズの数とする。ただし, マトリクスの乗算の場合のように, フェイズが繰返される場合は繰返しも含めたフェイズの数とする。

4.2 転送回数の評価データ

1) 転送系相互の比較 PEあたり結線数を少なくしていった場合に転送回数が増加する割合を調べる。

そのほかの転送系に対し, 各タイプの参照形態を完遂するに必要な転送回数を前節の式より求めた。結果を対数構造結線の転送回数を1とした相対転送回数として図9

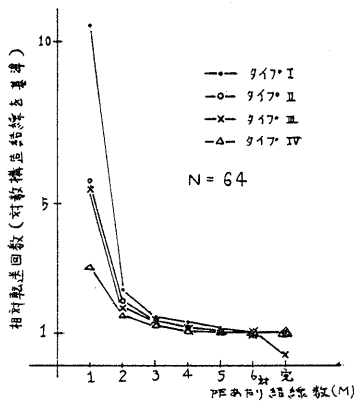


図9(a) 転送回数の比較 (N=64)

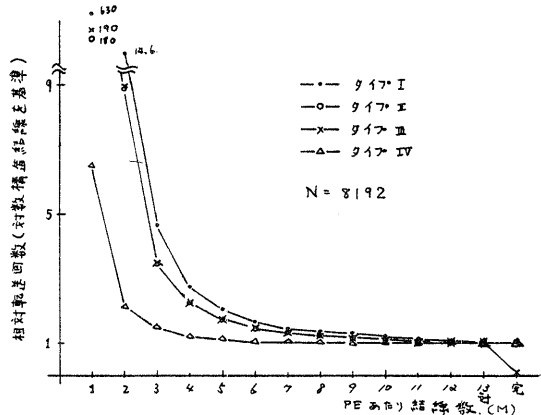


図9(c) 転送回数の比較 (N=8192)

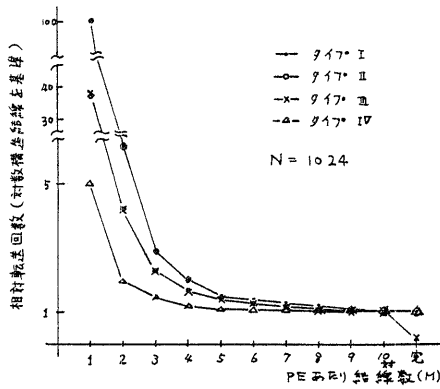


図9(b) 転送回数の比較 (N=1024)

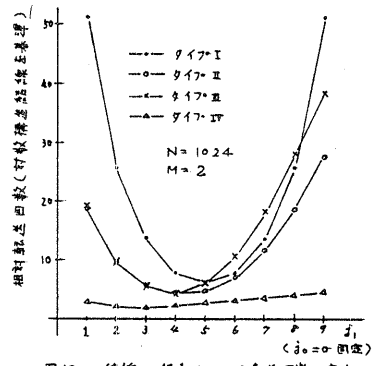


図10. 結線の組合せによる転送回数の変化

に示す。横軸は各転送系に対応し、数字はPEあたり結線数(M)を示す。ただし完は完全結線、対は対数構造結線を意味する。

向引き対数構造結線では結線の組合せが複数ある。 $N (=2^j)$ 台のPEに対しPEあたりの結線数をM本とすれば、その組合せは $j-1, C_{M-1}$ 通りである($j_0=0$ は固定されているからそれを除いて計算)。図は各Mに対し転送回数最小となる組合せの転送回数を示す。

2) 結線の組合せによる比較 向引き対数構造結線において、結線の組合せによる転送回数の変化を調べる。 $N=1024, M=2$ の場合について結線の組合せによる転送回数の変化を図10に示す。図は対数構造結線の転送回数を1とした相対転送回数である。

4.3 考察

4.2節の結果について考察する。

図9からMが小さくなるにつれて転送回数は増加する。 $N=1024$ の場合にはM=2とM=10の比がタイプI~タイプIVでそれぞれ6.2, 4.2, 4.2, 1.9である。すなわち、M=2まで向引いても転送回数は数倍の増加をおさめる。

図10から最小の転送回数をよえる結線(j_1)はタイプI~タイプIVでそれぞれ5, 4, 4, 3であり、各タイプで異なる。一方、結線を $j_1=5$ に固定した場合には、相対転送回数は6.2倍以内におさまる。

対数構造結線の転送回数に対し、その増加を5~6倍まで許せるなら、以上のことから $N=1024$ に対してはM=2($j_1=5$)でよいことがわかる。すなわち $N=8192$ に対してはM=3($j_1=4, j_2=9$)でよい。(PE数を8倍にしても結線数は1本増すだけでよい。)

すなわち、向引き対数構造結線はM=2程度まで向引いても2のべき乗

隔てたデータ参照形態を持つ問題に対して転送効率のすぐれた転送系ということが出来る。

5. おわりに

本稿では、並列アルゴリズムのデータ参照形態には共通性があることに着目し、こゝらを4種のタイプに分類した。次いで、種々の単独構成転送系に対し、4種のデータ参照形態の転送回数を調べた。こゝらのデータ参照形態には対数構造結線が最も適合するが、この実現は困難であるため、より簡単な結線構造の転送系で代用した際の転送回数増加を評価した。その結果、相当数の結線を向引いても向引き対数構造結線はその増加が少ない有効な転送系であることがわかった。

最後に、日壇御指導頂くオ一研究室山下敏一室長はじめ研究室諸氏に深謝する。

(参考文献)

- (1) H. J. Siegel, "Interconnection Network for SIMD Machines" Computer, Vol. 12, No. 6, pp. 57-65, JUN., 1979
- (2) 加藤, 苗村, "並列24理計算機, 超高速化へのアーキテクチャ" オーム社, 1978
- (3) D. E. Knuth, "The Art of Computer Programming, Vol. 3" pp. 220-235, Addison-Wesley, 1973
- (4) Brigham (宮川, 他訳), "高速フーリイ変換" 科学技術行出版社, 1978
- (5) Y. Okada, et al, "A Novel Multiprocessor Array" 2nd Symp. on Micro Archi., pp. 83-90, 1976