通信遅延を仮定した時の最適なリダクション計算

6E−8

野々村 洋†　栗野 俊一‡　深澤 良彰†

†早稲田大学理工学部　‡日本大学メディア科学研究室

1 はじめに

並列計算機においてリダクション計算を行なう場合、データ交換の時間（通信時間）を考慮しなければ、2分木状に進めた時が、最少のステップ数での計算となる。しかし、通信による遅延を考慮した場合、この方法ではデータ交換のための無駄な待ち時間が生じ、計算に要するステップ数は最少にならない。

本発表では、一定の通信時間を考慮した場合に、全てのプロセッサがデータ交換のための待ち時間なしで、かつ、最少のステップ数でリダクション計算を行うための並列計算機の最適なネットワークトポロジー、ならびにその上での処理割り当て法について述べる。

2 計算および計算機に関する仮定

計算機に対する仮定　対象となる計算機は必要十分な数のプロセッサを持ち、通信によってデータの交換を行なう疎結合マルチプロセッサ型の並列計算機であるものとする。ただし下のような条件を満たす。

- 隣接したプロセッサ間の通信
  - 通信遅延は常に一定
- プロセッサ (PE)
  - 対象となる計算に使われる演算は常に同一時間内で終了する
  - 同一の要素内で通信と転送は同時にできない

データ転送時間の定義　2つのプロセッサ間での通信は次のような順で行なわれるものとする。まず、送信側のプロセッサが処理を行ない、データを通信経路に流す。これが一定時間後、受信側のプロセッサに到達し、受信の処理が行なわれ、通信が完了する。データ転送時間とは、送信のための処理とデータの伝達にかかる時間を合わせたものと定義する。

計算の条件　次のような条件を全て満たすような基本演算からなる式のリダクションによる計算のアルゴリズムについて考える。

- 引数の数が同じ
- 結合律が成立する

本稿では、このような条件を満たすリダクション計算のみを対象とする。

3 計算のアルゴリズム

通信遅延を考慮しない場合、リダクション計算は2分木状に計算を進めた場合に最少のステップ数で計算が完了する。しかし、データ転送に常に一定(かつ0より大)の時間を要すると仮定すると、2分木状に計算を進めた場合は、最適とはならない。特にデータ転送の時間が計算時間に等しい場合で、ステップ数は遅延がない場合に比べて約2倍となる。(図1)

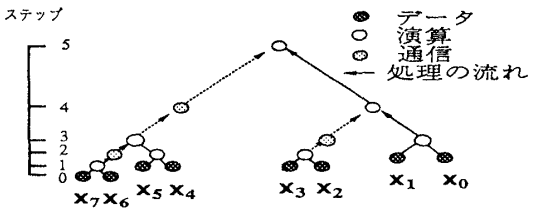

図 1: 通信遅延がある場合の2分木状の計算

一方、図2で示した計算木のように計算を進めた場合、各プロセッサは、データ伝達される直前まで計算を行なうことができ、データ転送による待ち時間がなくなる。このため、待ち時間による無駄がなくなり、一定時間の通信遅延を仮定した場合でも計算を最少のステップ数で進めることができる。

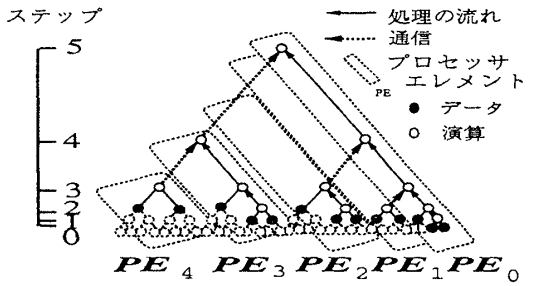

図 2: 通信遅延に対応した計算木

この通信遅延に対応した計算木は次のような手順により構成される。まず、通信遅延を考慮していない2分木状の計算木において、通信に対応した枝の部分に新たなノードを挿入し、通信に対応した枝は新たなノードを経由する2つの枝に分割する。さらに、各枝の高さをそろ

The optimum reduction calculation with communication delay
You Nonomura†, Shun-ichi Kurino‡, Yoshiaki Fukazawa †
† School of Science & Engineering, Waseda University
3-4-1 Ookubo, Shinjuku-ku, Tokyo 169, Japan
‡ Media Lab., Nihon University
1-5-2, Kanda Surugadai, Chiyoda-ku, Tokyo 101, Japan

えるため、最も低い枝に合わせるよう、ノードおよび枝を取り去る。このような操作を施すことによりステップ数最少となる計算木を構成することができる。この構成の様子を図3に示す。図2では、このようにしてできた計算木からさらに見やすいよう、遅延を示すノードを除いてある。

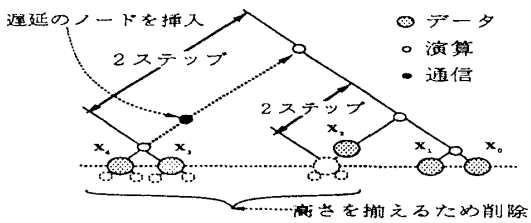

図 3: 計算木の構成法

この計算木をさらに一般化して、通信時間が m 、計算時間が n の場合についても構成することができる。

先の構成法では、通信遅延に対応するノードを挿入することにより、通信遅延を考慮した計算木を構成していた。一般化した場合も同様に、計算時間および通信遅延に対応するノードを、要する時間の数だけ挿入し、高さを揃えることにより計算木を構成することができる。また、計算に含まれる基本演算が l 項演算であった場合は、計算木を 2 分木から l 分木にすればよい。

このようにして、より一般的な場合での、無駄のない計算木を構成することができる。

4 フィボナッチ木

転送時間と計算時間が等しい場合の計算木を埋め込むための、並列計算機におけるネットワークトポロジーについて考えてみる。例として計算時間とデータ転送の時間が等しい場合の計算木を埋め込むネットワークを考える。高さ h の計算木に対応するネットワークトポロジー T_h は図4のようになる。これを高さ h のフィボナッチ木と呼ぶことにする

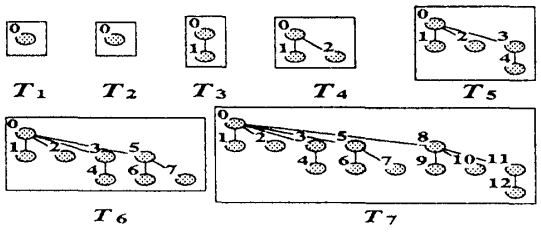

図 4: フィボナッチ木

フィボナッチ木 T_h は T_{h-1} と T_{h-2} の根に当たるノードをつなぎ、T_{h-1} の根であったノードを新たに T_h の根とする構成となっている。ただし、T_1 と T_2 は 1 つのノードのみからなる木である。先に図 2 に示した計算木は図 4 に示した T_5 の各ノード $t_0, t_1, \cdots t_4$ に、$PE_0 \rightarrow t_0, PE_1 \rightarrow t_1, \cdots PE_4 \rightarrow t_4$ のように埋め込むことができる。

計算木を一般化した場合、これに対応するネットワークトポロジーの方も一般化される。通信時間と計算時間の比が $m : n (m, n \in N)$ の場合に、$i(\geq 2)$ 項演算からなる計算を行なう場合、高さ h の計算木を埋め込むことのできるネットワークトポロジーを考える。これを T'_h とすると、これは T'_{h-m} の根であるノードと $i-1$ 個の $T'_{h-(m+n)}$ の根を結合したものとなる。ただし、$h \geq m$ であり、また $2m+n > h \geq m$ においてはノードが 1 つだけのグラフとなる。このグラフのノード数 N_h は次のような一般式で表される。

$$N_h = \begin{cases} N_{h-m} + (i-1)N_{h-(m+n)} & (h \geq 2m+n) \\ 1 & (2m+n > h \geq m) \\ 0 & (m > h > 0) \end{cases}$$

この式で示される数列に対応したフィボナッチ木を再帰的に構成することにより、フィボナッチ木の一般化を行なうことができる。

5 評価

転送時間と計算時間が等しいとき、計算を 2 分木状に進めた場合と、通信遅延を考慮した計算木を用いた場合の 2 通りに計算した場合のステップ数の比較を図5に示す。

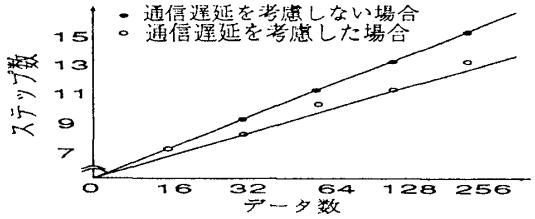

図 5: 2 つの計算法の比較

図5 よりわかるとおり、2 分木状の計算より、通信遅延を考慮した計算木を用いた方が、データ数が大きくなった場合で約 30% ほど高速になる。通信遅延がより大きい場合、この差はさらに大きくなる。

6 おわりに

以上の結果より次のことがいえる。まず一定時間の通信遅延、および計算時間が保証されているものとする。このとき最少のステップ数でリダクション計算を行なうための、データおよび処理の割り当て方法を計算木の形で示すことができる。また、この計算木を用いるための最適なネットワークトポロジーを得ることができる。

極端な例として、通信遅延が 0 の場合は計算木は 2 分木となり、通信遅延が無限大の場合、計算木は 1 次元のリスト構造になる。よって、この計算木は通信時間、および計算時間を考慮した計算木の一般化となっている。