数理解析研究所講究録
第 8[illegible]250 1993 年 250-255

バス結合型並列計算機における データ転送の最適アルゴリズム

京都大学工学部　岡部 寿男 (Yasuo OKABE)
京都大学工学部　津田 孝夫 (Takao TSUDA)

Abstract

Floyd モデルを拡張し、共有バスで結合されたマルチプロセッサをモデル化した新しい記憶階層モデルを提案する。いくつかの問題について、このモデル上でのデータ転送コストの下界を導出し、さらに行列積について下界を実現する最適アルゴリズムを示す。

1 はじめに

計算機における記憶階層の問題は、古くから理論的にも実際的にもさまざまな研究がなされている。特に、二階層記憶におけるデータ転送回数の理論上の下界に関する Floyd の研究 [1] 以来、さまざまな記憶階層のモデル化、およびその上でのデータ転送回数に関する研究がなされてきている [2]〜[5]。

本論文では、従来の二階層記憶のプロセッサを並列化した新しいモデルを提案し、その上でのデータ転送回数について議論する。本モデルは共有バスで結合されたマルチプロセッサをモデル化しており、現実の計算機システム、とくにネットワークで結合された分散システムに近く現実的であると考えられる。

提案するモデル上で、ソーティング、FFT、行列転置および行列積についてのデータ転送の下界を示す。さらに行列積および行列転置について、下界を実現する最適アルゴリズムが存在することを示す。

2 計算モデル

本稿で提案する計算モデルは、p 台のプロセッサおよび 1 台のディスク (ファイルサーバ) を 1 本の『バス』で接続したものである。各プロセッサは容量 M のローカルメモリをもつ。プロセッサ間、及びプロセッサとファイルサーバの間のデータ転送は B レコードごとの一括転送で行なわれる。以下、次のようなパラメータを用いる。

- N: 計算対象ファイル中のレコード数

- M: 各プロセッサの内部メモリに格納可能なレコードの最大数
- B: 1ブロックあたりのレコード数
- p: プロセッサ数

また $m = M/B$ とおく。

1回のデータ転送においてデータを送信することができるのは、ディスクまたはどれか1台のプロセッサだけである。バス上に流れるデータは、すべてのプロセッサおよびディスクが受信可能である。すなわちいわゆるブロードキャスト (またはマルチキャスト) を許す。

このモデルにおいては、計算時間を計算に必要なデータ転送の回数で計る。プロセッサは十分速いものと考え、内部メモリ上での計算に必要な時間は無視できるものとする。以下、本モデルを**バス結合並列二階層記憶モデル** (または単に並列二階層記憶モデル) とよぶ。これは、よく用いられる二階層記憶モデル [4],[2] をマルチプロセッサの場合に拡張したものとみなすことができる。

3 問題の定義

初期状態においては、各プロセッサの内部メモリの内容はすべて空であり、入力データはディスクの先頭番地から連続する Nレコードに格納されているものとする。

ソーティング

初期状態 N 個のレコードはディスクの先頭から格納されている。

計算 各プロセッサは、2つのレコードの大小比較、および交換の操作を行なう。

最終状態 N 個のレコードがディスクの先頭から小さいもの順にならんでいる。

FFT

N は2の冪乗とする。

初期状態 N 個のレコード $n_{i,0}$ $(0 \leq i \leq N-1)$ はディスクの先頭から格納されている。

計算 $n_{i,j}$ は $n_{i,j-1}$ と $n_{i\oplus 2^{j-1},j-1}$ から計算される $(1 \leq j \leq N)$。ここで$\oplus$ はそれぞれの数値の2進表現をビットごとの排他的論理和演算する演算子である。

目標状態 N 個のレコードが $n_{i,N}$ $(0 \leq i \leq N-1)$ がディスクの先頭から格納されている。

行列転置

$N = N_1 N_2$とする。

初期状態 $N_1 \times N_2$ 行列 $A = (a_{i,j})$ の要素である N 個のレコードはディスクの先頭から格納されている。

計算 各プロセッサは、2つのレコードの大小比較、および交換の操作を行なう。

最終状態 $N_2 \times N_1$ の転置行列 A^T がディスクの先頭から格納されている。

行列積

$N = 2N_1^2$とする。

初期状態 2つの $N_1 \times N_1$行列 A と B が列優先形式でディスクの先頭から格納されている。

計算 A の要素 $a_{I,K}$ と B の要素 $b_{K,J}$ とからその積 $c_{I,J}^{(K)}$ を求める計算1で求めた積(または2で求めた和)の和を求める計算[1]

目標状態 $N_1 \times N_1$行列 $C = AB$ が列優先形式でディスクの先頭から格納されている。

4 データ転送量の下界

プロセッサ数 p、各プロセッサの内部メモリ容量 m のバス結合並列二階層記憶モデルは、明らかに、内部メモリ容量 pMの二階層記憶モデルでシミュレートできる。二階層記憶モデルに関する結果 [2] から、バス結合並列二階層記憶モデルにおけるデータ転送回数の下界として以下が導かれる。

Theorem 1 バス結合並列二階層記憶モデルにおける

1. ソーティングおよび FFT に必要なデータ転送数の下界は

$$\Omega(\frac{N}{B}\frac{\log(N/B)}{\log(pM/B)})$$

である。

2. 行列転置に必要なデータ転送の下界は

$$\Omega(\frac{N}{B}(1+\frac{\log\min\{m,N_1,N_2,N/B\}}{\log(pM/B)}))$$

である。

[1] Strassen のアルゴリズム [6] のように、分配則を用いて計算量を減らすことは許さない。

3. 行列積に必要なデータ転送数の下界は

$$\Omega(\frac{N_1^3}{\min\{N_1, \sqrt{pM}\}B})$$

である。

5 最適アルゴリズム

行列転置については、文献 [2] のアルゴリズムがそのまま下界実現アルゴリズムとして実現できる。以下、行列積について、データ転送回数の下界を実現する最適アルゴリズムを示す。

2つの $k \times k$行列の積を計算するアルゴリズム 以下の分割を $2k^2 \leq pM$ となるまで繰り返す [2]。(pM はローカルメモリの総量)

1.

$$A = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}; B = \begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix}$$

と分割し、 $A_1 \sim A_4, B_1 \sim B_4$ を列優先でストアする。

2. 本アルゴリズムを再帰的に呼び出し、部分行列の積 $A_1B_1, A_2B_3, \ldots, A_3B_2, A_4B_4$ を計算。

3. $C_1 \sim C_4$ を計算。

$$C = AB = \begin{pmatrix} C_1 & C_2 \\ C_3 & C_4 \end{pmatrix} = \begin{pmatrix} A_1B_1 + A_2B_3 & A_1B_2 + A_2B_4 \\ A_3B_1 + A_4B_3 & A_3B_2 + A_4B_4 \end{pmatrix}$$

4. C を列優先でストアする。

(1), (3), (4) で必要なデータ転送回数は $O(k^2/B)$ である。$k \times k$ 行列の積の計算に必要なデータ転送回数を $T(k)$ とおくと、(2) より、

$$T(k) = 8T(k/2) + O(k^2/B)$$

が成立する。

次に $k \leq ..\sqrt{pM}/2$ のとき、

$$c_{I,J} = \sum_K a_{I,K} b_{K,J}$$

を以下のようにして計算する。

1. 第 j番目のプロセッサ $(1 \leq j \leq p)$ に

$$b_{*,J}, \quad J = (j-1)k/p+1, ..., (j-1)k/p+k/p$$

を転送。

2. $a_{*,K}, K = 1, ...k$ を順にブロードキャストし、各プロセッサで

$$c_{*,J} := c_{*,J} + a_{*,K} b_{K,J};$$

を計算

3. $c_{I,J}$ を順にストア。

各ステップは、それぞれ k^2/B のデータ転送で行なえる。すなわち

$$T(k) = 3k^2/B = O(pM/B)$$

よって一般の k については

$$T(k) = O(\frac{k^3}{\sqrt{pM}B})$$

以上、行列積については、4 章で示した下界を実現するアルゴリズムが存在することが示された。

6 おわりに

本論文では、二階層記憶モデルを拡張し、バス結合型並列計算機の新しいモデルを提案した。このモデル上でのソーティング、FFT、行列転置、行列積のデータ転送回数の下界を示した。さらに、行列積については、下界を実現する最適アルゴリズムを示した。

本モデルは、通常の並列計算のモデルと違い、プロセッサ数を増やしても、計算速度の加速はわずかである。プロセッサ数を p 倍にしても、行列積の場合で $\sqrt{p}$倍、行列転置、FFT などの場合には高々 $\log p$ 倍にしかならない。これは、ある意味で、バス結合型計算機の能力の限界をうまく表すものと考えることができる。近年 Ethernet 上に多数のワークステーションを接続して分散計算を行なうような試みがなされているが、本稿の結果は、そのようなシステムの性能見積りにも役立つ可能性がある。

今後の課題としては、まず、FFT、ソーティング等についての最適アルゴリズムの検討が必要である。また、ブロードキャストの機能を持たないモデルの能力についても興味がある。さらに、バス結合以外の結合網に対応するモデルについてはどのような結果が得られるかも考えてみたい。

参考文献

[1] R. W. Floyd: "Permuting information in idealized two-level storage," Complexity of Computer Calculations, R. Miller and J. Thatcher (Eds), Plenum, New York, 105–109 (1972).

[2] A. Aggarwal and J. S. Vitter, "The input/output complexity of sorting and related problems," Communications of the ACM (September 1988), 1116–1127.

[3] J. S. Vitter and E. A. M. Shriver, Algorithms for parallel memory I: two-level memories, " Technical Report No. CS–90–21, Department fo Computer Science, Brown University (Sept. 1990).

[4] J. Savage and J. S. Vitter: "Parallelism in space-time tradeoffs," VLSI: Algorithms and Architectures, P. Bertolazzi and F. Luccio, Eds, Elsevier Science Publishers B. V., 49–58 (1985).

[5] J. W. Hong and H. T. Kung: "I/O complexity: the red-bule pebble game," Proc. of the 13th Annual ACM Symposium of Theory of Computing, 326–333 (1981).

[6] V. Strassen: "Gaussian elimination is not optimal," Numerische Mathematik, **13**, 354–356 (1969).