アルゴリズム 34－5
（1993．8．19）

バス結合並列計算機モデルにおける データ転送の最適アルゴリズム

岡部 寿男　津田 孝夫
京都大学 工学部 情報工学教室

内部メモリを持つ複数台のプロセッサとファイルサーバが共有バスで結合された二階層記憶の新しい計算モデルを提案し、このモデル上でのデータ転送回数について論ずる。バスに対するブロードキャストが許されるか否かにより、モデルの能力は本質的に異なる。本稿ではその両方の場合について、ソーティング、FFT などのデータ転送回数の下界と下界を実現する最適アルゴリズムを示す。いずれも Aggarwal と Vitter の示した二階層記憶に関する結果をマルチプロセッサの場合に拡張したものである。

Optimal I/O Algorithms on Bus-Connected Parallel Machines

Yasuo OKABE and Takao TSUDA
Dept. of Information Science, Kyoto University
KYOTO 606-01, JAPAN

We propose a new computation model of two-level storage. It is comprised of processors with internal memory and a disk as a file server, whose are connected via a shared bus. Computation time is measured by the number of I/O operations. The power of the model essentially depends on whether *broadcasting* to the bus is allowed or not. We show tight lower and upper bounds of the number of I/O operations, for both of the above cases, required for sorting, permutation and FFT. These results are multi-processor extensions of the bounds on two-level storage provided by Aggarwal and Vitter.

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。

Floyd のモデルは、内部メモリ上に 2 個のレコードのみを置くことができるという単純なものであったが、現実の計算機に合わせ、内部メモリのサイズを増やし [5, 4]、ディスクとのデータ転送をブロック化したモデル [2] が提案され、これらのモデル上でのソーティング、FFT、行列転置などのデータ転送回数の下界、および下界を実現する最適アルゴリズムが示された。さらに並列にデータ転送が可能な複数のディスクが接続されたモデルについても同様の結果が示されている [3]。

本研究では、二階層記憶におけるプロセッサを並列化した新しい計算モデルを提案し、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。

このモデルでは、バスに対するブロードキャストを許すか否かで能力が異なり得る。我々はすでに、ブロードキャストを許すモデルにおいて行列転置および行列積におけるデータ転送回数の下界と最適アルゴリズムを示した [6]。本発表では、ソーティング、置換および FFT について、ブロードキャストを許す場合、許さない場合それぞれ、データ転送の下界と、下界を実現する最適アルゴリズムを示す。これらはいずれも Aggawal と Vitter の示した 1 プロセッサのモデルにおける結果 [2] を、マルチプロセッサの場合に拡張したものである。

ブロードキャストを許さないモデルでは、ソーティング、置換および FFT についてはプロセッサ数を増やしてもモデルの能力が向上しない。ブロードキャストを許すモデルでは、ソーティング、置換についてはマルチプロセッサ化によりモデルの能力は向上するが FFT においては、プロセッサ数をブロックサイズ B 以上に増やしても能力が向上しない。このモデルではマルチプロセッサにおける並列性と同時にデータのブロック転送における並列性についても考慮しており、最適アルゴリズムを考えることによって、問題自身のもつ並列化可能性を多面的に評価することができる。

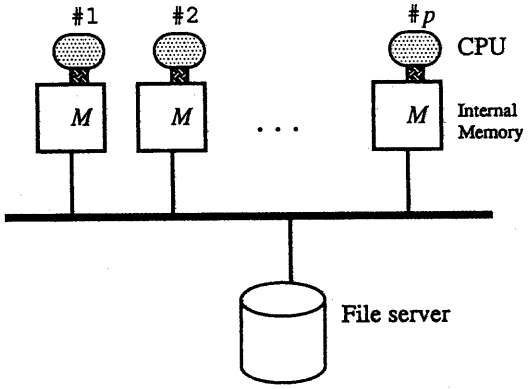

図 1: 並列二階層記憶モデル

以下、2 章では計算モデルについて定義する。3、4、5 章で、それぞれソーティング、置換、FFT についてデータ転送回数の下界の導出と、下界を実現する最適アルゴリズムを示す。

2 計算モデル

本稿で提案する計算モデルは、図 1 に示すように 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

各プロセッサの内部メモリおよびファイルサーバのディスクはブロック単位で一次元的にアドレス付けされているものとする。各レコードは少なくとも $\lfloor 1+\log_2 N \rfloor$-bit の大きさを持つと仮定する。

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

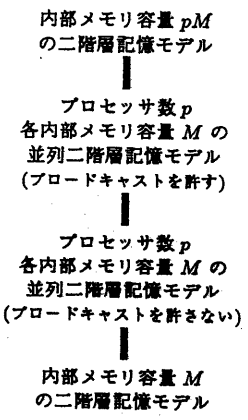

図 2: モデル間の能力の比較

である。ブロードキャストを許すモデルでは、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が可能である。ブロードキャストを許さないモデルでは、1回のデータ転送においてデータを受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。

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 (入出力) 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。

並列二階層記憶モデルは、よく用いられる二階層記憶モデル [2],[4] 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。一方プロセッサ数 p、各プロセッサの内部メモリ容量 M の並列二階層記憶モデルは、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。よってモデル能力の間には図2のような関係が成立する。これはちょうど、単一プロセッサ、分散メモリ型マルチプロセッサ、共有メモリ型プロセッサの間の関係に対応すると考えてもよい。

3 ソーティング

3.1 問題の定義

初期状態 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、N個の入力レコードはディスクの先頭番地から連続する N/B ブロックに格納されている。

計算 各プロセッサは、2つのレコードの大小比較、および内部メモリ上での移動の操作を行なう。

最終状態 N 個のレコードがディスクの先頭から連続する N/B ブロックに小さいものから順に格納されている。

3.2 データ転送回数の下界

プロセッサ数1の二階層記憶モデルにおけるソーティングのデータ転送回数については次の結果がある [2]。

命題 1 ブロックサイズ B、内部メモリサイズ M の二階層記憶モデルにおいて、N 個のレコードをソーティングするのに必要なデータ転送回数は、最悪、平均ともに

$$\Theta(\frac{N}{B}\frac{\log(N/B)}{\log(M/B)})$$

である。

プロセッサ数 p、各プロセッサの内部メモリ容量 M の並列二階層記憶モデルは、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。このことから、直ちに以下の下界が導かれる。

系 1 並列二階層記憶モデルにおけるソーティング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。

系1はブロードキャストの有無に関わらず成り立つが、ブロードキャストを許さないモデルにおいては、以下のようにしてさらに下界を改善することができる。議論を簡単化するために、モデルに次のような制約を加える。

- データ転送は、入力 (ディスクから内部メモリ) または出力 (内部メモリからディスク) のみである。プロセッサ間の直接通信は許さない。

- 入力レコードは、各ブロック内でソート済みである。

N 個の被ソートレコードに対し、その入力ブロックでの並び方は $\frac{N!}{B!^{N/B}}$ 通りある。よってこれらのレコード間の順序関係を比較と移動により完全に決定するためには、最悪の場合少なくとも

$$\Omega(\log_2 \frac{N!}{B!^{N/B}}) = \Omega(N \log(N/B))$$

回の比較が必要である。

データ転送コストが最小のアルゴリズムを一つ考える。このアルゴリズムに、

- 内部メモリに読み込まれたレコードに対しては比較演算により必ずそれらの間の相対順序を完全に決定する。
- 出力の際にはブロック上でレコードをソートしておく。

という制約を加えても最適性は失われない。

このアルゴリズムによるソーティング途中のある時点でのブロックの入力を考える。仮定により内部メモリ上の $M-B$ 個のレコードの相対順序は既知である。入力ブロック上の B 個のレコードについても同様である。ここで、高々

$$O(\log \binom{M}{B}) = O(B \log(M/B))$$

回の比較により長さ $M-B$ および B のそれぞれソートされた連を、長さ M のソートされた連にマージできることに注意する。$O(B \log(M/B))$ 回の比較でレコード間の相対順序を完全に決定できるのであるからそれ以上の比較は無駄であり、1 回の入力読み込みに対し、新たに行なえる有意な比較は $O(B \log(M/B))$ しかない。ブロックの出力の際には新たに行なうべき有意な比較はないので、ソートが完了するまでに少なくとも

$$\frac{\Omega(N \log(N/B))}{O(B \log(M/B))}) = \Omega(\frac{N}{B} \frac{\log(N/B)}{\log(M/B)})$$

回の入出力が行なわれることが必要である。以上まとめて

補題 1 ブロックサイズ B、プロセッサ数 p、内部メモリサイズ Mのブロードキャストなし並列二階層記憶モデルにおける N レコードのソーティングに必要なデータ転送数の下界は

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

である。

これは命題 1に示した二階層記憶モデルでの下界 [2] に一致する。なお以上は最悪データ転送回数についての議論であるが、$N!$ 通りに均質に分布した入力レコード並びに対する平均データ転送回数についても同じ下界が導ける。

3.3 最適アルゴリズム

補題 1に示すように、ブロードキャストなしモデルにおいては、データ転送回数の下界がプロセッサ数 p に依存しない。すなわち、二階層記憶モデルにおける最適アルゴリズムがこのモデルにおいても最適でありプロセッサを増やしても高々定数倍しか高速化されないことを意味する。

一方ブロードキャストモデルにおいては、p 台のプロセッサに分散配置された各 M の内部メモリを効率よく用いてサイズ pMの共有メモリの場合と同じ入出力回数でソーティングが行なえる可能性がある。実際、以下に示すとおりよく知られた分散ソートのアルゴリズムがこのモデル上で並列実行でき、補題 1 で示した下界が実現できる。

通常の二階層記憶モデルにおいては、内部ソートがデータ転送なしに行なえることが重要である。すなわち、内部メモリにちょうど収まる M レコードのソートは、レコードをディスクから内部メモリに入力しソート結果を書き戻すのに最低限必要な $2\lceil M/B \rceil$ 回の入出力でソートが行なえる。ブロードキャストつきモデルにおいても内部メモリに分散配置可能な pM レコードのソートが $O(pM/B)$ 回の入出力で行なえることを示す。

オンメモリソート

入力 $p(M-B)/2$ 個のレコードがファイルサーバ上の $\frac{p}{2}(\frac{M}{B}-1)$ 個のブロックに配置

出力 入力レコードをファイルサーバ上の連続する $\frac{p}{2}(\frac{M}{B}-1)$ 個のブロックにソートして配置

まず、入力ブロックを $(M/B-1)/2$ づつ p 台のプロセッサに分散配置する。各レコードに、それぞれ対応す

る $\lceil \log_2 pM \rceil$-bit のカウンタを用意する。次にプロセッサは順次、各レコードをブロック毎に順に全プロセッサにブロードキャストする。受けとったプロセッサでは、ブロードキャストされた各レコードと、内部メモリ上のレコードとを比較し、内部メモリ上のレコードの方が小さければ対応するカウンタに1を加える。比較においては、同一レコードに対する比較以外では必ず大小いずれかの結果をとるようにする[1]。

この操作のあとのカウンタの値は各レコードの入力レコード中での順位を表している。各レコードとその順位の組を一度すべてファイルサーバに書き戻し、それらを再度ブロードキャストする。i 番目のプロセッサ $(i = 0, ..., p)$ は、$\{(\frac{M}{B}-1)i/2, ..., (\frac{M}{B}-1)(i+1)/2-1\}$ 番目のデータをメモリ上に残す。最後に 1 番目のプロセッサより、順次データを書き戻す。

以上の操作により、$p(M-B)/2$ 個のレコードのソートが $O(pM/B)$ 回の入出力で行なえることが分かる。pM レコードのソートは、入力を4分割してそれぞれ上記アルゴリズムでソートし、それらを1プロセッサ上でマージすればよい。以下このソートをオンメモリソートとよぶことにする。

次に外部分散ソートのアルゴリズム [2] をブロードキャストつき並列二階層記憶モデル上で実行させることを考える。外部分散ソートのアルゴリズムは以下の通りである [2]。

$S = \sqrt{M/B}$ とおく。N レコードの S-近似分割要素 $b_1, ...b_S$ $(i_1 < i_2$ ならば $b_{i_1} < b_{i_2})$ を求め、レコードを S個のバケツに分割する。ここで $b_1, ...b_S$ が入力レコードの S-近似分割要素であるとは、$\forall i \in \{1, ..., S+1\}$ に対し半開区間 $(b_{i-1}, b_i]$ に含まれる入力レコードの個数を N_i とすると

$$\frac{1}{2}\frac{N}{S} \leq N_i \leq \frac{3}{2}\frac{N}{S}$$

が成立することである[2]。この分割操作を再帰的に繰り返し、1つのバケツのレコード数が pM 以下になったらオンメモリソートを行なう。

N レコードの S-近似分割要素は、後で示すように $O(N/B)$ で行なうことが可能である。これにより外部分散ソートの入出力回数は

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

である。

S-近似要素を求めるアルゴリズムの概略は以下の通りである。n レコード中の第 k番目の要素は $O(n/B)$ 時間で求めることができる。そこで、入力である N レコードを pM ごとにオンメモリソートし、それぞれから $S/4$ 番目ごとの要素を代表元として取り出す。これは $O(N/B)$ 回の入出力で可能である。得られた $4N/S$個のレコードのから第 Ni/S^2番目 $(i = 1, ..., S)$ の要素を取り出し b_i とする。必要な入出力回数は $O(S \cdot (N/S)/B) = O(N/B)$ であり、かつ b_i は求める性質を満たす。

定理 1 ブロックサイズ B、プロセッサ数 p、内部メモリサイズ Mの並列二階層記憶モデルにおいて、N 個のレコードをソーティングするのに必要なデータ転送回数は、最悪、平均ともに

$$\Theta(\frac{N}{B}\frac{\log(N/B)}{\log(pM/B)}) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許す場合}\end{pmatrix}$$

$$\Theta(\frac{N}{B}\frac{\log(N/B)}{\log(M/B)}) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許さない場合}\end{pmatrix}$$

である。

分散ソートでは、上述のようにマルチプロセッサ化による加速、すなわち分散メモリをあたかも共有メモリのように用いることが可能である。しかしこのようなマルチプロセッサ化による加速が常に可能である保証はない。実際もう一つの著名な外部ソートアルゴリズムであるマージソートについては、内部メモリ上でのマージ演算をオンメモリ上でおこなうことは自明でなく、マルチプロセッサ化が難しいのではないかと予想される。

4 置換

4.1 問題の定義

置換とは、ソーティングにおいて N 個のレコードの内容が $\{1, 2, \ldots, N\}$ の場合である。これはソーティングにおいてレコード間の順序が既知の場合と考えることもできる。

4.2 データ転送量の下界

まず二階層記憶モデルに関する結果 [2]

命題 2 ブロックサイズ B、内部メモリサイズ Mの二階層記憶モデルにおいて、N 個のレコードの置換に必

[1] キーの値が等しい場合は例えばレコードの入力時の位置を用いて比較することにすればよい

[2] 便宜上 $b_0 = -\infty$、$b_{S+1} = +\infty$ とおく

要なデータ転送回数は、最悪、平均ともに

$$\Theta(\min\{N, \frac{N}{B}\frac{\log(N/B)}{\log(M/B)}\})$$

より以下の下界が導かれる。

系 2 並列二階層記憶モデルにおける置換に必要なデータ転送数の下界は

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

である。

ブロードキャストを許さないモデルにおいては、さらにデータ転送回数の回数の下界を改善できる。議論を簡単化するために、モデルに次のような制約を加える

- 置換中において、各レコードは、内部メモリまたはディスク上の一カ所だけに存在する。すなわちデータ転送においては、転送元のデータは消去される。
- データ転送は、入力 (ディスクから内部メモリ) または出力 (内部メモリからディスク) のみである。プロセッサ間の直接通信は許さない。

以上の制約をつけても、ブロードキャストを許さないモデルにおいては置換に必要なデータ転送回数は高々定数倍にしか増加しないことが容易に示される (文献 [2] 補題 4.1 参照)。

初期状態では内部メモリおよびディスクの空き領域には nil が書かれているものとする。計算途中で実現されている置換とは、内部メモリおよびファイルサーバに一次元的アドレスを与えたときの (nil を無視した) レコードの並びのことである。

T 回の入出力によって実現できる置換の数を評価する。第 t 回目の入力においては、データを読み込むトラックの選択に $N/B+t-1$ の任意性、読み込みの対象となるプロセッサの選択に p の任意性がある。また、内部メモリのどのアドレスに読み込むかには $\binom{M}{B}$ の任意性がある。さらに、入力されるブロックが入力レコードの最初の読み込みである場合には、その並べ替えに $B!$ の任意性がある。よって、生成可能な置換の数は、第 t 回目の入力により、第 $t-1$ 回目までの入出力により生成可能な置換の数と比べて、入力レコードの最初の読み込みのときは

$$(N/B+t)B!p\binom{M}{B}$$

倍、それ以外の場合は

$$(N/B+t)p\binom{M}{B}$$

倍になる。第 t 回目の出力では、データをディスク上のどこへ書き込むかで $N/B+t$ の任意性、書きだしの対象となるプロセッサの選択に p の任意性がある。また、書き出すデータの内部メモリからの選択に $\binom{M}{B}$ の任意性がある。よって第 t 回目の出力により、生成可能な置換の数は

$$(N/B+t)p\binom{M}{B}$$

倍になる。T 回の入出力によって実現できる置換の数は高々 $B!^{N/B}((N/B+t)p\binom{M}{B})^T$。ここで、$t$ に対する自明な上界

$$(N/B+t) \le N(1+\log N)$$

を用いると、T回のデータ転送で生成可能な置換の最大数が $N!$ より大きくなる条件は

$$B!^{N/B}\left(N(1+\log N)p\binom{M}{B}\right)^T \ge N!$$

と書ける。スターリングの公式を用いると

$$T(\log N+\log p+B\log\frac{M}{B}) = \Omega(N\log\frac{N}{B})$$

より、

$$T = \Omega(\min\{N, \frac{N\log(N/B)}{B\log(M/B)}\})$$

を得る。これは命題 2 の下界に一致する。すなわちブロードキャストを許さなければ、ソーティングの場合と同様マルチプロセッサ化による高速化は高々定数倍である。

なお、置換はソーティングにおいてレコード間の順序が既知である場合と考えることができるから、以上の下界は比較以外の操作を許すソーティングにも適用できる。

4.3 最適アルゴリズム

置換はソーティングの特殊な場合であるから、3.3節のソーティングアルゴリズムは置換のアルゴリズムでもある。さらに、ブロードキャストつきモデルにおいて $N < \frac{N\log(N/B)}{B\log(pM/B)}$、ブロードキャストなしモデルにおいて $N < \frac{N\log(N/B)}{B\log(M/B)}$ の場合には、1 プロセッサのみを用いたデータ転送回数 N の自明なアルゴリズムを用いることにすれば、4.2節で示した下界が実現される。

定理 2 ブロックサイズ B、プロセッサ数 p、内部メモリサイズ M の並列二階層記憶モデルにおいて、N 個のレコードをソーティングするのに必要なデータ転送回数は、最悪、平均ともに

$$\Theta(\min(N, \frac{N}{B}\frac{\log(N/B)}{\log(pM/B)})) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許す場合}\end{pmatrix}$$

$$\Theta(\min(N, \frac{N}{B}\frac{\log(N/B)}{\log(M/B)})) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許さない場合}\end{pmatrix}$$

である。

5 FFT

5.1 問題の定義

N は 2 の冪乗とする。

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

5.2 データ転送回数の下界

FFT digraph は、3 段重ねにすることにより置換網 (permutation network) が実現できる [7]。よって置換網に対する下界は FFT に対しても下界となる。

まずブロードキャストを許さないモデルにおける置換網に必要なデータ転送回数の下界を導く。ある置換網とそれを実現する入出力アルゴリズムを考える。入出力の順序、および各ブロックのデータをどのプロセッサの内部メモリに読み込むかは入力データに依存せず固定であるから一回の入力により生成される置換の数の最大数は、入力レコードの最初の読み込みの場合$\binom{M}{B}$倍、それ以外の場合は $B!\binom{M}{B}$倍になる。出力については$\binom{M}{B}$倍である。よって T 回の入出力で生成されうる置換の最大数は $B!^{N/B}\binom{M}{B}^T$。これが $N!$以上となる条件から、

$$T = \Omega(\frac{N}{B}\frac{\log(N/B)}{\log(M/B))})$$

次に、ブロードキャストを許すモデルの場合を考える。一回の入力により B 個の データが p 台のプロセッサににそれぞれ $B_1, B_2, \ldots, B_p$ 個送られるとする。ここで $B_1 + B_2 + \ldots + B_p = B$ $(B_j \ge 0)$。生成される置換の数は、

$$\prod_{j=1,\ldots,p}\binom{M}{B_j}$$

これが最大値をとるのは $p \ge B$ のとき

$$\binom{M}{1}^B = M^B$$

$p < B$のときは

$$\binom{M}{\lfloor B/p\rfloor+1}^{B\bmod p}\binom{M}{\lfloor B/p\rfloor}^{p-B\bmod p} < \binom{M}{\lfloor B/p\rfloor+1}^p$$

よって入力レコードの最初の読み込みの以外の場合高々

$$\min\{\binom{M}{\lfloor B/p\rfloor+1}^p, M^B\}$$

倍、最初の読み込みの場合はそのさらに $B!$倍になる。よって T 回の入出力で生成されうる置換の最大数は $B!^{N/B}\min\{\binom{M}{\lfloor B/p\rfloor+1}^p, M^B\}^T$。これが $N!$以上となる条件から、

$$\begin{aligned} T &= \Omega(\frac{N}{B}\frac{\log(N/B)}{\log\min\{M, pM/B\}}) \\ &= \Omega(\frac{N}{B}\frac{\log(N/B)}{\log(pM/(B+p))}) \end{aligned}$$

補題 2 バス結合ブロードキャストなし並列二階層記憶モデルにおける FFT に必要なデータ転送数の下界は

$$\Omega(\frac{N}{B}\frac{\log(N/B)}{\log\frac{pM}{B+p}}) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許す場合}\end{pmatrix}$$

$$\Omega(\frac{N}{B}\frac{\log(N/B)}{\log(M/B)}) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許さない場合}\end{pmatrix}$$

5.3 最適アルゴリズム

ソーティングの場合と同様に、ブロードキャストを許すモデルでのオンメモリ FFT のアルゴリズムを考える。p, M いずれも 2 の冪であると仮定する。ノード数 pM の FFT digraph は図 3 のように p プロセッサ、$k = \frac{\log_2 pM}{\log_2 M}$ ステージに分解でき、各ステージ内ではプロセッサ間の通信は不要である。またステージ間の通信は

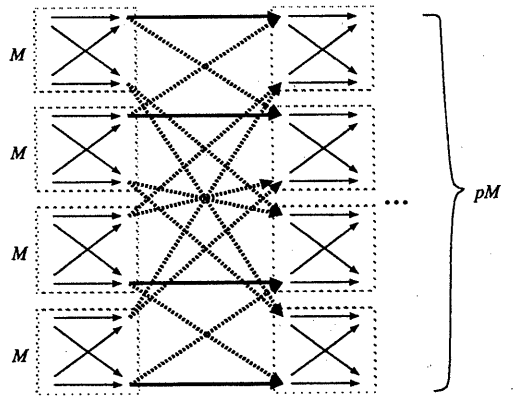

図 3: FFT digraph の ステージへの分解

pM/B 回のブロードキャストで行なえる。よってこの分解に沿った FFT に要するデータ転送の回数は kpM/B。

オンメモリ FFT アルゴリズムを用いると、二階層記憶モデルでの外部 FFT アルゴリズムがほぼそのままブロードキャストつき並列二階層記憶モデルでも実行でき、そのデータ転送回数は

$$O(\frac{N}{B}\frac{\log N}{\log M}\frac{\log \min\{pM, N/B\}}{\log(pM/B)})$$

これは整理すると補題 2のブロードキャストを許す場合の下界に一致する。すなわち

定理 3 ブロックサイズ B、プロセッサ数 p、内部メモリサイズ Mの並列二階層記憶モデルにおいて、N 点の FFT および置換網の実行に必要なデータ転送回数は、

$$\Theta(\frac{N}{B}\frac{\log(N/B)}{\log \frac{pM}{B+p}}) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許す場合}\end{pmatrix}$$

$$\Theta(\frac{N}{B}\frac{\log(N/B)}{\log(M/B)}) \quad \begin{pmatrix}\text{ブロードキャスト}\\ \text{を許さない場合}\end{pmatrix}$$

6 おわりに

本発表ではバス結合並列二階層記憶モデル上でのソーティング、置換、FFT について論じ、データ転送回数の下界と下界を実現する最適アルゴリズムを示した。その結果、ブロードキャストなしモデルとブロードキャストつきモデルでは能力が本質的に差があることが明らかになった。

ブロードキャストを許さないモデルでは、ソーティング、置換および FFT についてはプロセッサ数を増やしてもモデルの能力は向上しない。ブロードキャストを許すモデルにおいても、能力の向上は高々 $\log p$ 倍である。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しているものと考えることができる。

FFT においては、プロセッサ数をブロックサイズ B 以上に増やしても能力が向上しない。これは FFT の計算が FFT digraph に沿って行なわれることを条件にしているためである。すなわち、問題自身がデータ転送アルゴリズムを規定している一面があり、それ以上の並列性を引き出せないと考えられる。逆に、例えば問題を DFT の形で定義すればより少ないデータ転送回数でフーリエ変換が行なえる可能性もある。

今後の課題として、バス結合以外の結合網をもつモデルや、各プロセッサがディスクを持つようなモデルについても考えてみたい。

謝辞 日頃から御討論頂く本学國枝義敏助教授をはじめ津田研究室の諸氏に深謝します。なお本研究は一部文部省科学研究費補助金による。

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] 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).

[3] 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).

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

[5] 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).

[6] 岡部, 津田: 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, 計算機構とアルゴリズム研究集会報告集 (1993 冬の LA シンポジウム, Feb. 1993), 京都大学数理解析研究所講究録 No. 833, 250–255 (Apr. 1993).

[7] C. L. Wu and T. Y. Feng: "The universality of the shuffle-exchange network", IEEE Trans. Comput. C-30, 5, 324–332 (May 1981).