

# 多重バス結合並列プロセッサのための最適時間ソーティング アルゴリズム†

藤 田 聡<sup>††</sup> 山下 雅 史<sup>††</sup> 阿 江 忠<sup>††</sup>

本論文では、共有バス型並列プロセッサにおける共有バスの多重化の効果について理論的に評価する。共有バス数  $k$  が処理要素 (PE) 数  $n$  に依存して増加するとき、 $n$  要素のソートは  $\max(4n/k, C\sqrt{n \log n})$  時間で行うことができる。系が共有バスの他に線形アレイ結合を持つ場合、この手数はさらに  $\max(3n/k, C\sqrt{n \log n})$  にまで減少する。ここで  $C$  は定数である。これらの値は  $k \leq \sqrt{n/\log n}$  のときオーダ的に下界  $\Omega(n/k)$  と一致し最適となる。

## 1. はじめに

本論文では、共有バス型並列プロセッサにおける共有バスの多重化の効果について理論的に評価する。ここでは具体的な問題として、 $n$  台の処理要素 (PE) からなる並列プロセッサ上で  $n$  個の要素 (自然数) をソートする問題に注目する。ただし初期状態として、各 PE にひとつずつの要素が割り当てられているものとし、要素の入出力に要する時間は以下の議論では考慮しない。各 PE が線形に結合されたマシンモデル上でこの問題を解くには明らかに  $\Theta(n)$  時間を要するが、この系に共有バスを定数本付加したとしてもソートの手数はオーダ的には減少せず  $\Theta(n)$  のままであることが知られている<sup>1),2)</sup>。しかし共有バス数  $k$  が PE 数  $n$  に依存して増加する場合にこの手数がどの程度まで減少するかについては、これまでのところ明らかにされていない。

中野らは階層化バスを持つモデル上でのソートについて考察している<sup>3)</sup>。階層化バスモデルでは局所的な共有バスも存在するため、ネットワークの能力を単純にバス本数のみによって評価するのは公平でない。そこで彼らは、ネットワークを平面上に埋め込んだときの幅 (バス幅) を尺度としてモデルを評価している。文献 6) では、各 PE が 2 本以上のバスを選択できる場合、ソート時間が  $O(n/w + \log^2 n)$  となることが示されている。ここで  $w$  は埋め込み時のバス幅である。この結果は共有バス結合に局所性を導入している点で興味深い。階層構造がアルゴリズムに依存している

ためアーキテクチャとしての一般性は高くない。さらに文献 6) のアルゴリズムは階層化されたバスモデルのためのものであり、例えばすべてのバスが大域的であるような状況で  $o(n)$  の実行時間を達成するには、このアルゴリズムの単純な拡張では  $O(\log n)$  本の共有バスが必要となる。したがって文献 6) の結果は、大域バスの多重化が計算時間に与える影響を明らかにしているわけではない。

本論文で得られた結果は以下のように要約される： $k$  本の大域的な共有バスによって結合された並列プロセッサ上でのソートは  $\max(4n/k, C\sqrt{n \log n})$  時間で行うことができる。系が線形アレイを付加的に持つ場合、この手数はさらに  $\max(3n/k, C\sqrt{n \log n})$  にまで減少する。ここで  $C$  は定数である。したがって共有バス数  $k$  が PE 台数  $n$  に依存して増加するとき、線形アレイ結合の有無にかかわらず、 $n$  要素のソートに要する手数は  $o(n)$  となる。ここで  $o(n)$  はオーダ的に真に  $n$  より小さいことをあらわす。この結果は、系に付加される共有バス数が PE 数  $n$  によらず一定である場合と PE 数  $n$  に依存して増加する場合とでは、計算機の能力が真に異なることを意味している。また提案アルゴリズムは、 $k \leq \sqrt{n/\log n}$  のときオーダ的に下界  $\Omega(n/k)$  と一致し最適となる。

以下 2 章では、本稿で対象とする多重バス結合並列プロセッサモデル MBCP を定義する。3 章では共有バス数が PE 数  $n$  によらず一定である場合に関する Stout の結果を概説する。4 章では MBCP 上でのソーティングアルゴリズムを提案しその実行時間を評価する。このアルゴリズムは線形アレイ結合を持たない (同一本数の共有バスのみを有する) 並列プロセッサによって、定数倍の時間でシミュレートできる。最後に 5 章では今後に残された課題について述べる。

† An Optimal Sorting Algorithm for Parallel Processors with Multiple Busses by SATOSHI FUJITA, MASAFUMI YAMASHITA and TADASHI AE (Cluster II, Faculty of Engineering, Hiroshima University).

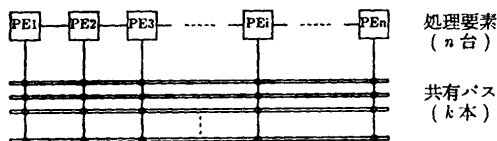
†† 広島大学工学部第二類

## 2. 多重バス結合並列プロセッサモデル

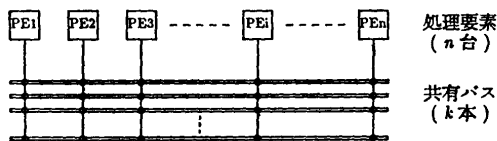
本論文で対象とするプロセッサは、 $k$ 本の共有バスと $n$ 台の PE からなる並列プロセッサである。議論をわかりやすくするため、以下ではまず最初に多重バスと線形アレイ結合を併せ持つモデルを考える（以下このモデルを MBCP と呼ぶ）。線形アレイ結合を持たない純粋な多重バス型並列プロセッサについては、MBCP 上の  $o(n)$  時間ソーティングアルゴリズムを具体的に示した後、4.3 節で言及する。MBCP と、MBCP から線形アレイを除去したモデルの概略図を図 1 に示す。MBCP は特殊な場合として 1 本の共有バスを付加された線形アレイプロセッサ<sup>3)</sup>を含む。なお各 PE は固有の識別子 (ID) を持ち、ID は線形アレイの左から順に昇順に与えられているものとする。

[MBCP における仮定]

- (1) 各 PE は定数個のレジスタを持つ同一の処理要素とする。各 PE は同期的に動作し、同一の共有バスに対する同時送信は許さない。
- (2) 各 PE は線形結合のほか、 $k$ 本のバスすべてにアクセス可能である。ただし 1 回の送受信に使用できるバス数はそれぞれ高々 1 本ずつであり、どのバスを使用するかは各 PE によって指定される。共有バスからの受信用には長さ 2 のバッファを設ける。
- (3) 実行時間は 1 つの要素が他の PE へ伝達される時間を 1 ステップとして評価し、PE 内部の



(a) 多重バスと線形アレイ結合を有するモデル MBCP  
(a) MBCP.



(b) MBCP から線形アレイ結合を除去したモデル  
(b) A model obtained by eliminating linear connections from MBCP.

図 1 多重バス型並列プロセッサモデル

Fig. 1 Model of parallel processors with multiple busses.

計算時間は考慮しない。1 ステップで送信できる情報量は、各 PE の識別子 ( $\log n$  ビット) と最大要素の持つ情報量との和の定数倍とする。また 1 ステップ内に各 PE は、隣接 PE との接続路 (ポート) を介した送受信と、共有バスへの送受信を同時に行えるものとする。ただしポートは全二重である。□

(1) はバス競合の排除に関する仮定である。また (3) は、1 回の通信に付随する計算が比較演算程度とはほぼ一定であることから、理論的な評価において妥当な仮定であると考えられる。なお以下の議論では、ソートすべき要素集合の中に同一の要素があった場合は、より小さい ID をもつ PE に割り当てられている要素の方が小さいとする。したがって 1 回の要素の転送では、要素の値と初期状態でその要素が割り当てられていた PE の ID を対にして送る必要がある。

## 3. 単一バス結合並列プロセッサ上でのソート

並列ソーティングに関しては、これまでも数多くの研究がなされている<sup>3)</sup>。なかでも  $d$  次元アレイ結合された並列プロセッサ上でのソート時間の下界を示した以下の定理は重要である<sup>3)</sup>。

[定理 1] (Stout 83)

PE 数  $n$ 、次元  $d$  のアレイ上でのソートには、付加される共有バスの本数が  $n$  によらず一定であれば、その有無にかかわらず  $\Omega(n^{1/d})$  時間要する。□

証明のアイデアは次のとおり。証明ではソートを  $n$  個の要素の置換とみなし、少なくとも  $\Omega(n^{1/d})$  単位時間を必要とするような置換を構成することによって、ソートに要する時間の下界を示している。この置換は、(1) 通信にアレイ結合を用いた場合には PE 間の距離だけの時間が必要であること、および (2) 共有バスを用いた場合には放送すべき要素数分の時間が必要となること、に注目して再帰的に構成されている。

アレイの次元を 1 に限定すると、定理 1 から、定数本の共有バスの線形アレイへの付加はソート時間のオーダーを  $\theta(n)$  から減少させないことがわかる。したがって次に問題となるのは、 $k$  が  $n$  に依存して増加するときソート手数のオーダーが  $n$  から真に減少するかどうかという問題である。以下ではこの問いに対して肯定的な解答が与えられる。

#### 4. 多重バス結合並列プロセッサ上のソート

MBCP において共有バス数の増加の効果を引き出すための方法のひとつは、共有バスの大域性と線形アレイ結合の局所性のバランスをうまくとることである。提案アルゴリズムでは線形アレイ結合の局所性をいかにするため、まずサイズ  $n$  の線形アレイをサイズ  $n/k$  の部分アレイに分割する（ここで  $k$  は共有バスの本数）。アルゴリズムでは、部分アレイをソートし各部分アレイの端にくるべき要素を検出したのち、要素を部分アレイ間で移動させるという方針をとる。部分アレイの端に来るべき要素は、中央値選択アルゴリズム<sup>1)</sup> およびその並列化アルゴリズム<sup>2)</sup> を MBCP 用に修正することで得られる（4.2 節参照）。

##### 4.1 ソーティングアルゴリズム

提案するソーティングアルゴリズムは以下の 4 フェーズよりなる。なお図 2 にこのアルゴリズムにそったソートの様子を例示する。要素集合の初期配置が図 2 (a) に示されている。

[Algorithm MBCP]

1. PE 集合を部分アレイに  $k$  等分する。部分アレイを順に  $A_1, \dots, A_k$  と呼ぶ。ここで  $A_i$  と  $A_{i+1}$

$A_1$	$A_2$	$A_3$
6 3 11	10 9 2	7 12 5

(a) 要素集合の初期配置

(a) Initial placement of elements.

3 6 11	2 9 10	5 7 12
$B_1$	$B_2$	$B_3$

(b) フェーズ 1 の結果

(b) Result of Phase 1.

第 1 ステップ

3	--	2	--	5	--
---	----	---	----	---	----

0	0	0	← $V(0)$
3	3	3	← $B(1)$

第 2 ステップ

3	--	2	--	5	--
---	----	---	----	---	----

3	0	0	← $V(1)$
2	2	2	← $B(2)$

2	3	0	← $V(2)$
---	---	---	----------

- (c) フェーズ 4 (ブロック  $B_1$  に関する部分のみ)
- (c) Phase 4 (For block  $B_1$  only).

図 2 アルゴリズム MBCP の説明

Fig. 2 Explanation of algorithm MBCP.

は隣接する部分アレイである。各部分アレイ上で要素集合を局所的にソートし、結果をその部分アレイ上のすべての PE に配布する ( $2\lceil n/k \rceil$  ステップ)。図 2 (b) では部分アレイ  $A_1, A_2, A_3$  上の要素がそれぞれソートされている。

2. 要素集合の  $j \times \lceil n/k \rceil$  ( $j=1, 2, \dots, k$ ) 番目の要素の値を求める ( $O(k \log n)$  ステップ)。要素集合の中で  $i$  番目に小さい要素を  $e_i$  (ただし  $e_0=0$ ) と呼び、要素集合の部分集合  $B_i$  ( $i=1, \dots, k$ ) を  $\{x | e_{(i-1)\lceil n/k \rceil} < x \leq e_{i\lceil n/k \rceil}\}$  と定義する。各  $B_i$  のことをブロックと呼ぶ（本論文では、要素集合の部分集合のことをブロック、PE 集合の部分集合のことを部分アレイと呼び区別する）。各  $e_{j\lceil n/k \rceil}$  の値は共有バスを用いて全 PE に対して放送される ( $k$  ステップ)。その結果各 PE は、自分がフェーズ 1 の終了時に保持していた要素がどのブロックに属するかを知ることができる。図 2 の例では、 $e_3=5, e_6=9, e_9=12$  が得られる。また図 2 (b) には各要素の属するブロックが付記されている。
3. 各 PE は自分の両隣の PE を調べることによって、その部分アレイの中で自分と同じブロックに属し、かつ自分より小さな（あるいは大きな）ID を持つ PE が存在するかどうかを調べる ( $O(1)$  ステップ)。

4. フェーズ 4 では実際に要素を部分アレイ間で転送する。このフェーズでは (4-a) と (4-b) の各ステップを同時に高々  $n/k+k$  回繰り返す（実行はブロックごと並列に行う）。フェーズ 4 では各 PE ごとに、 $V_i(t), F_i(t)$  という 2 種類の変数を用いる。 $V_i(t)$  は、第  $t$  ステップに  $PE_i$  によって保持されている値で、フェーズ 4 の開始時に 0 に初期化される。また  $F_i(t)$  は、第  $t$  ステップの  $PE_i$  の入力バッファの値であり、これは第  $t$  ステップで  $PE_i$  が注目している共有バスに流された値である。なおこのフェーズでは、ブロック  $B_i$  を部分アレイ  $A_i$  に対応させ ( $i=1, \dots, k$ )、各ブロックに共有バスを 1 本ずつ対応させておく。

##### (4-a) SEND

各ブロック  $B_i$  に対応付けられた部分アレイ  $A_i$  への転送は、小さい ID を持つ PE から順に行う。図 2 の例では、ブロック  $B_1$  に対応付けられた  $A_1$  への転送が  $PE_1, PE_4, PE_7$

の順に行われる。同一部分アレイ中で同一のブロックに属する PE のうち、自分が最小（あるいは最大）の ID を持っているかどうかはフェーズ3の結果より判定できる。各 PE は要素を送信する際、自分の ID と、自分が上記の意味で最大の ID を持つ PE であるかどうかを示すタグとを併せて送る。共有バス上にこのタグを持つ要素が流されると、その共有バスに要素を送信する権利は次の部分アレイに移される。次の部分アレイに対応する PE が存在しなければ、そのステップでは何も送信されず、送信権は自動的にさらにその次の部分アレイに移される。

#### (4-b) RECEIVE

以下の説明では簡単のため、すべての部分アレイ上に、現在注目しているブロックに属する PE が存在するものと仮定する（図2では  $B_1$  について考える）。そうでない場合も簡単なタイミングの修正により対処することができる。

各 PE ( $PE_i$ ) は第  $t$  ステップで  $V_i(t-1)$  の内容を  $PE_{i+1}$  へポートを介して転送する。このとき以下の計算の実行により、任意の  $t$ ,  $i$  において  $V_i(t) \geq V_{i-1}(t)$  となることを保証できる。

```

if ( $V_i(t-1) \geq F_i(t-1) > V_{i-1}(t-1)$ )
    then  $V_i(t) := F_i(t-1)$ 
if ( $F_i(t-1) > V_i(t-1)$ )
    then  $V_i(t) := V_i(t-1)$ 
if ( $V_i(t-1) \geq F_i(t-1)$ )
    then  $V_i(t) := V_{i-1}(t-1)$ 

```

フェーズ4の様子を図2(c)に示す。第1ステップでバスに流される要素 ( $F_i(1)$ ) は、 $PE_1$  に保持されていた3である。その結果、 $V_1(1)=3$ ,  $V_2(1)=V_3(1)=0$  となる。第2ステップでバスに流される要素 ( $F_i(2)$ ) は、 $PE_4$  に保持されていた2であり、これは  $F_i(1)$  より小さい。しかし上記の判定を各 PE 上で行うことにより、 $V_1(2)=2$ ,  $V_2(2)=3$ ,  $V_3(2)=0$  がそれぞれ得られる。つまりこのフェーズでは、ブロック間転送とブロック内のソートが同時実行されることになる。□

フェーズ1は、通常の奇偶置換ソート<sup>1)</sup>を行ったのち、共有バスを帰還路として用いたサイズ  $\lceil n/k \rceil$  のリング上でパイプライン的に部分ソート列を転送することにより、高々  $2\lceil n/k \rceil$  ステップで完了する。

フェーズ3に要するステップ数は、 $n$  や  $k$  によらず一定である。またフェーズ4の正当性も簡単な帰納法により容易に証明できる。したがって問題となるのは、フェーズ2の実行時間である。次節ではフェーズ2についてもう少し詳しく考察する。

#### 4.2 代表要素の検出

フェーズ2の開始時の状況は次のようになっている。

サイズ  $n/k$  の各部分アレイはソート済みであり、その結果は各部分アレイのすべての PE に配布されている。

ソート結果は各 PE 上にサイズ  $n/k$  の配列の形でたくわえておくことにする。フェーズ2の目的は  $k$  個の要素の値を求めることであるが、各 PE ( $PE_i$ ) にはアルゴリズムの実行中、自分が現在その内のどれを求めているのかを認識させておく。ここではその値をラベルと呼び、各 PE のラベルを処理要素の ID " $i$ " をパラメータとして  $label(i) \in \{n/k, 2n/k, \dots, n\}$  であらわす。また現在ラベル  $\alpha$  を持っている PE に割り当てられた最小要素の順位を  $ord(\alpha)$  で表す。この順位はそのラベルを持つ最小要素が更新される度にラベル  $\alpha$  を持つすべての PE に放送される。ラベル更新の様子を図3に示す。

以下の構成法では任意時刻において各ブロック上で同一の  $label(i)$  を持つ PE は必ず連続した位置に並んでいる（補題1参照）。また各共有バスはフェーズ4と同様、各ラベルの値に対して一意に対応付けておく。

#### [フェーズ2]

2-1. すべての PE について  $label(i) := n/k$  とし、 $ord(n/k) := 1$  と初期設定する。図2の例では、 $label(i) := 3$ ,  $ord(3) := 1$  である（図3(a)）。以

3	6	11	2	9	10	5	7	12
$label(i):$								
3	3	3	3	3	3	3	3	3
$ord(3) := 1$								

(a) 初期状態  
(a) Initial condition.

3	6	11	2	9	10	5	7	12
$label(i):$								
3	6	6	3	6	6	3	6	6
$ord(3) := 1, ord(6) := 4$								

(b) 1回目の繰返し直後の各 PE のラベル  
(b) Label of each PE after the first iteration.

図3 フェーズ2の説明

Fig. 3 Explanation of the second phase.

- 下の 2-2~2-5 を, 求めるべき  $k$  個の値すべてが得られるまで各ラベルごと並列に繰り返す.
- 2-2. 同一の部分アレイ  $A_i$  上に存在しかつ自分と同じラベル  $\alpha$  を持つ PE の集合  $S_{i,\alpha}$  を考える. また  $S_\alpha = \bigcup_{i=1}^k S_{i,\alpha}$  と定義する. 各集合  $S_{i,\alpha}$  に属する PE の持つ要素の中央値を, ラベル  $\alpha$  用の共有バスに対して ID の小さい PE から順に送り出し, フェーズ 4 と同様の手法を用いてオンラインでソートする (高々  $k$  ステップ). 図 2 の例では要素 6, 9, 7 がソートされる.
- 2-3. 中央値の中央値 ( $S_\alpha$  の仮中央値と呼ぶ) を求めた後, その値を  $S_\alpha$  の全 PE に放送する. 図 2 の例では仮中央値は 7 である.
- 2-4.  $S_\alpha$  の中でその仮中央値よりも小さい要素を持つ PE の個数  $num$  を計数する.  $num=0$  ならば, その仮中央値が全体で  $\alpha$  番目の値 (すなわち求めるべき値のひとつ) である. 図 2 の例では  $num=4$  である.
- 2-5. 区間 A [ $ord(\alpha), ord(\alpha)+num-1$ ] と区間 B [ $ord(\alpha)+num, ord(\alpha)+|S_\alpha|-1$ ] を考える. もし  $\alpha \in A$  ならば, 区間 B の要素をもつすべての PE のラベルをその区間にあるラベル候補の最小値 ( $\beta$  とする) に置き換え  $ord(\beta) := num$  とする. もし  $\alpha \in B$  ならば  $ord(\alpha) := num$  とし, 区間 A の要素をもつすべての PE のラベルをその区間にあるラベル候補の最小値 ( $\gamma$  とする) に置き換え  $ord(\gamma) := num$  とする. 図 2 の例では, 区間 A が [1, 4] 区間 B が [5, 9] であり,  $\alpha=3$  は区間 A に属している. したがって区間 B にある要素をもつ PE のラベルを, 区間 B に属するラベル候補の最小値 6 とする (図 3 (b)). 自分が区間 B に属するかどうかは, 区間 B で最も小さい要素が先程求めた仮中央値であることから判定できる. いずれの場合も, ラベル候補が存在しない場合はその区間の要素をもつ全 PE をその PE のラベルを空にすることによって無効化する.  $\square$

2-2~2-5 の実行時間に関して次の補題が成立する.

[補題 1]

フェーズ 2 のループ 1 回あたりに要する時間は  $O(k)$  である.

(証明) フェーズ 2 の開始時にサイズ  $n/k$  の各部分アレイはソートされており, その結果はその部分アレイ上のすべての PE に配布されている. さらに 2-5

で生成される要素集合はソート列上の連続した区間であるから, 任意の部分ソート列においても連続した区間に対応する. よって各部分アレイ上で同一のラベルを持つ PE は連続した位置に並んでおり, その要素の中央値は,  $O(1)$  時間で選択できる. つまり  $k$  個の中央値すべてをある部分アレイ上に集めるには  $k+O(1)$  時間あれば十分である. 中央値集合は  $k$  時間でソートできる. その結果 (仮中央値) は  $O(1)$  時間でそのラベルを持つ全 PE に対して放送可能である. 同様に仮中央値よりも小さい要素の総数も, 各 PE 上で定数時間で計算した結果を部分ブロックの代表値に転送させればよいから  $k+O(1)$  時間で計数できる. 最後に 2-5 の処理は各 PE 上で独立に  $O(1)$  時間で実行できるから, よってフェーズ 2 の 1 回のループに要する時間は  $O(k)$  である.  $\square$

繰返し回数に関しては以下の補題が成立する.

[補題 2]

フェーズ 2 の繰返し回数は, 高々  $\log n$  の定数倍である.

(証明) 証明は文献 2) と同様である. あるラベルを持つ PE の数は, フェーズ 2 のループを 1 回まわるとに少なくとも  $3/4$  以下に減少する (図 4 参照). したがって求めるべき要素集合がソート列上に等間隔に存在することに注意すれば, 高々  $\lceil \log_{4/3} k \rceil$  回の繰返しの後にすべてのラベルが系の上に出現することがわかる. 最後のラベルが出現したとき, 各ラベルあたりの PE 数は高々  $2n/k$  である. これを 1 にするにはループを高々  $\log_{4/3}(2n/k)$  回まわせばよいから, 結局フェーズ 2 の繰返し回数は高々  $\log_{4/3} 2n \approx 2.41 \log_2 n + O(1)$  となる.  $\square$

よって以下の定理が成立する.

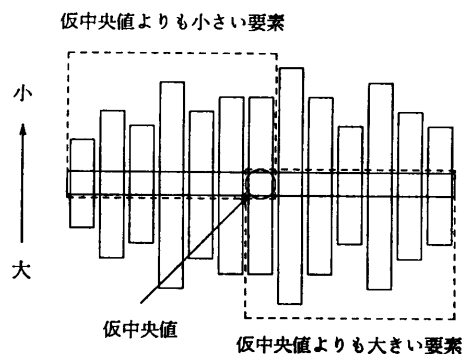


図 4 フェーズ 2 の正当性の説明

Fig. 4 Explanation of the correctness of the second phase.

## [定理 2]

$k$ 本の共有バスを持つ MBCP 上で  $j \times n/k$  ( $j=1, 2, \dots, k$ ) 番目に小さい  $k$ 個の要素を求めるのに要する時間は  $O(k \log n)$  である。□

この結果と上記の考察の結果から、MBCP 上でのソートに関して以下の定理が成り立つ。

## [定理 3]

$n$ 台の PE と  $k$ 本の共有バスを持つ MBCP 上での  $n$ 要素のソートは、高々  $\max(3n/k, C\sqrt{n \log n})$  時間で行うことができる。ここで  $C$  は定数である。□

共有バスを全く用いない純粋な線形アレイプロセッサモデルでの計算時間は  $n$  であるから<sup>1)</sup>、共有バスの効果は、( $n$ が十分大きければ) バス本数が 4 以上のとき確実にあらわれることになる。一方共有バスを有する線形アレイ上でのソート時間の下限はバス本数を  $k$  としたとき  $\Omega(n/k)$  であることが定理 1 の系として導かれるから、したがって提案したアルゴリズムは、 $k \leq \sqrt{n \log n}$  のときオーダ的に最適となる。

## 4.3 線形アレイ結合の除去

アルゴリズム MBCP において線形アレイ結合は次の 3通りの状況で用いられている(支配的ではない項に関連するフェーズ 2 は、ここでは考慮しない)。

(a) フェーズ 1 における部分アレイ上のソートおよび巡回シフト、

(b) フェーズ 3 における隣接 PE との通信、および

(c) フェーズ 4 におけるオンラインソート。

各部分アレイあたり 1本の共有バスが使用可能であることに注意すると、フェーズ 1 は共有バスを介した数え上げソートを用いて高々  $2n/k$  時間で完了する。次にフェーズ 3 の通信は、フェーズ 1 の結果各 PE がその部分アレイ上の局所的なソート列を知っていることに注意すれば、PE 内部の計算で置き換えることができる(現実には通信した方が高速になるが、このモデルでは PE 内部の処理時間を無視しているため、この通信を時間的な損失なしに除去できる)。最後にフェーズ 4 のソートは、部分アレイ間転送と部分アレイ内ソートを別々に実行することによって、線形アレイ結合を持たないモデル上で  $2n/k$  時間でシミュレートできる。

以上の考察から、MBCP から線形アレイ結合を除去したモデル上でのソートについて以下の系が導かれる。

## [系]

MBCP から線形アレイを除去したモデル上で、アルゴリズム MBCP は  $\max(4n/k, C\sqrt{n \log n})$  時間でシミュレートできる。□

## 5. おわりに

本論文では並列プロセッサにおける共有バスの多重化の効果について考察した。一般にソート問題は半群計算や中央値選択問題などとは異なり、共有バス付加の効果の少ないクラスに属する問題であると考えられている<sup>3)-5)</sup>。Stout によって示された定理(定理 1)はある意味でこのことを裏付ける結果となっている。しかし共有バス数を一定として PE 数  $n$  のみを増加させ、さらに定数倍の範囲内で  $\Theta(n)$  という形で実行時間を同一視するという Stout の評価方法は、少しでも高速な並列プロセッサを実現したいという立場からは片手落ちのように見える。現実的な局面としては、定数倍の高速化で十分な場合の方がむしろ多いようにも思われる。本論文の結果(定理 3)は、後者の観点から問題を捕え直し、共有バスの多重化の効果定数倍の範囲で肯定するものとなっている。

今後、本論文の結果をもとにして以下のような方向に議論を拡張していく予定である。

(1) 2次元アレイにおける共有バス付加の効果に関する考察。

(2) 共有バスの本数  $k$  が  $\sqrt{n \log n} < k < n \log n$  の範囲にあるときの最適アルゴリズムの開発。

## 参 考 文 献

- 1) Akl, S.G. (阿江ほか訳): 並列ソーティングアルゴリズム, 啓学出版 (1988).
- 2) Blum, M. et al.: Time Bounds for Selection, *J. Comput. Syst. Sci.*, Vol. 7, pp. 448-461 (1973).
- 3) Stout, Q.F.: Mesh-Connected Computers with Broadcasting, *IEEE Trans. Comput.*, Vol. C-32, No. 9, pp. 826-830 (Sept. 1983).
- 4) 梅尾ほか: グローバル・バスを利用した並列アルゴリズムについて, 情報処理学会アルゴリズム研究会, SIG-AL 4-7 (Nov. 1988).
- 5) Kumar, V.K.P. and Reisis, D.: Parallel Image Processing on Enhanced Arrays, *Proc. of ICPP '87*, pp. 909-912 (1987).
- 6) 中野ほか: バス付き 1次元格子上的並列ソーティングアルゴリズム, 信学論 D-I, Vol. J72-D-I, No. 9, pp. 631-641 (Sept. 1989).

(平成 2 年 7 月 24 日受付)

(平成 3 年 2 月 12 日採録)

**藤田 聡** (正会員)

昭和 60 年広島大学工学部第二類(電気系)卒業。平成 2 年同大学院博士課程修了。工学博士。現在広島大学工学部第二類助手。並列アルゴリズムの設計、分散システム等に興味をもつ。電子情報通信学会会員。

**山下 雅史** (正会員)

昭和 49 年京都大学工学部情報卒業。昭和 52 年同大学院修士課程修了。昭和 55 年名古屋大学大学院博士課程修了。工学博士。豊橋技術科学大学助手を経て、現在、広島大学工学部第二類助教授。この間、昭和 61 年より約 1 年間、カナダサイモンフレーザー大学客員助教授。マルチプロセッサの負荷分散問題、画像処理の基礎、組合せ問題の研究に従事。電子情報通信学会会員。

**阿江 忠** (正会員)

昭和 39 年東北大学工学部通信卒業。昭和 44 年同大学院博士課程修了。工学博士。東北大学助手、広島大学助教授を経て、昭和 57 年広島大学教授(工学部第二類計算機工学教育科目担当)となり現在に至る。この間、昭和 49 年より 1 年間、仏グルノーブル大学客員研究員。現在、主として、分散処理システム(マルチプロセッサおよび LAN)の設計、製作ならびに性能評価の研究に従事。電子情報通信学会、ACM、IEEE 各会員。