

並列処理向きシステムCubematについて

2N-2

洪 沢 進

群馬大学 工学部

1. まえがき

工業, 産業, 科学等の諸分野において, 数値, 非数値演算を高速に処理する実用的な要求が高い。(1), (2) このような要求に答える効率的な構成として, 多数の処理要素を用いた並列計算処理機構が, これまで数多く提案されてきた。それらの代表的な結合構成としては, 格子結合, 超立方体結合, 木結合, 多段相互結合ネットワーク等がある。

本報告では並列処理向きシステムCubematを提案し, その2次元レイアウトとそれを用いたアルゴリズムを与える。Cubematの各ノード間結合は, 超立方体(キューブ)結合と高速バス結合を基本にしている。提案している反射2進Cubematのレイアウトは, ノード数 $n$ が大きいと単純なレイアウト構成の約4/9の面積で実現できる。また, 反射2進Cubematのキューブ結合の一部は, すべてのノードに対して格子結合を形成している。このことは, 反射2進Cubematが超立方体アルゴリズムと格子結合アルゴリズムの双方を実行できることを示している。

また, 本報告ではCubematを用いたバイトニックソートの例を与えている。 $n$ ノードCubematが, 一度に $\sqrt{n}$ データを取り扱うことのできる主計算機のもとで動作するとき,  $n$ データ・バイトニックソートの本体は $O(\log^2 n)$ 時間, データ入出力は $O(\sqrt{n})$ 時間で実行できる。

2. Cubematの表現とレイアウト

2.1 双方向順次2進符号化によるCubemat

正整数 $m, r(r=2^m)$ に対して, 2次元平面 $xy$ 上に $n=r \times r$ 個の配列化したノード $e(I, J)(0 \leq I, J \leq r-1)$ を置く。ノード $e(I, J)$ の配列 $I, J$ をそれぞれ座標 $x, y$ の変化する方向にとる。配列 $I, J$ を次のように順次2進表現し, これらをそれぞれ $e^x, e^y$ とおく。

$$I = e^x = [e_{m-1} \dots e_1 e_0], \quad J = e^y = [e_{2m-1} \dots e_{m+1} e_m] \quad (1)$$

$$(e_i \in \{0, 1\}, 0 \leq i \leq 2m-1)$$

ノード $e(I, J)$ を $\langle e^x, e^y \rangle$ で表すとき,  $e(I, J)$ は次のように2進表現される。

$$e(I, J) = \langle e^x, e^y \rangle = [e_{2m-1} \dots e_1 e_0] \quad (2)$$

式(2)で表される2ノード $e(I, J)$ 間に キューブ結合を導入する。2ノードの2進表現のハミング距離が1のとき, キューブ結合はこれら2ノードを直接結合する。このとき, 結合線は $x$ または $y$ 方向のいずれか一方方向に引かれ, 両方向にまたがることはない。さらに, 各行及び列方向に高速バス結合を導入する。このバス結合は, 各ノードに情報を高速に伝送し, または多くのノードに一度に放送できることを目的とする。こうして構成されるノード結合を単純Cubematとよぶ。図1は,  $n=2^6(r=2^3, m=3)$ の単純Cubematの2次元レイアウトを示す。図中口はノードを表し, 実線はキ

ューブ結合, 点線はバス結合を表す。

レイアウトの仕方はThompsonのVLSIモデルに従う。レイアウトの評価において, 配線幅を単位長さとする。各ノードの内部は,  $(2m+2)$ 端子が相互に通信できるものであればどのような構成でもよいとする。 $(2m+2)$ 端子をノード領域の縁に置くと, ノードの一辺の長さは 高々 $O(m^c) = O(\log^c n)$  ( $c$ :定数)で実現できる。また, 各行, 列の配線領域の幅が $(2+2^2+\dots+2^{m-1})$ であることより, 単純Cubematの一辺の長さは

$$2^m \times [(2+2^2+\dots+2^{m-1}) + O(m^c)] = n + O(\sqrt{n} \log^c n) \quad (3)$$

これより,  $n$ ノード単純Cubematのレイアウト面積は次のようになる。

[補題1] ある正整数 $m$ に対して, 双方向順次2進符号化による $n=2^m \times 2^m$ ノードCubematのレイアウト面積 $A_m$ は, 配線幅を単位長さとするとき,

$$A_m = n^2 + O(n^{3/2} \log^c n) \quad (4) \quad \square$$

面積 $A_m$ は, 超立方体の2次元レイアウトの最適オーダーを満たしている。

2.2 双方向反射2進符号化によるCubemat

第2.1節と同様に, 2次元平面 $xy$ 上の配列 $I, J(0 \leq I, J \leq 2^m-1)$ をもつ $n=2^m \times 2^m$ 個のノードを考える。しかし, 配列 $I, J$ の反射2進符号をそれぞれ $u^x, u^y$ で表し, これらを次のように2進表現する。

$$u^x = [u_{m-1} \dots u_1 u_0], \quad u^y = [u_{2m-1} \dots u_{m+1} u_m] \quad (5)$$

$$(u_i \in \{0, 1\}, 0 \leq i \leq 2m-1)$$

このとき, 整数 $I$ の順次2進符号 $e^x$ と反射2進符号 $u^x$ の間には次の関係がある。(3), (4)

[補題2] 整数 $I$ の順次2進符号 $e^x$ と反射2進符号 $u^x$ をそれぞれ式(1), (5)の第1式のように表す。このとき,  $u^x$ の第 $j$ 桁 $u_j$ は $e^x$ の第 $j, j+1$ 桁 $e_j, e_{j+1}$ を用いて, 次のように表される。

$$u_j = \begin{cases} e_j \oplus e_{j+1} & (0 \leq j \leq m-2) \\ e_{m-1} & (j=m-1) \end{cases} \quad (6)$$

逆に $e^x$ の第 $j$ 桁 $e_j$ は $u^x$ を用いて次のように表される。

$$e_j = \sum_{i=j}^{m-1} u_i \quad (0 \leq j \leq m-1) \quad (7)$$

ここに, 記号 $\oplus, \sum$ はそれぞれ排他的論理和とmod2加算を表す。□

反射2進符号は, 巡回的に隣合う2進数対のハミング距離がつねに1であるという性質をもつ。配列 $I, J$ を双方向反射2進符号化した数 $u(I, J)$ を, 位置 $(I, J)$ にあるノードの名前とし, これを $\langle u^x, u^y \rangle$ で表せば,

$$u(I, J) = \langle u^x, u^y \rangle = [u_{2m-1} \dots u_1 u_0] \quad (8)$$

次に, 双方向反射2進符号化した任意の2ノード間にキューブ結合を定義する。また, 単純Cubematと同様に各行, 列にバス結合を導入する。このようにして構成されるノード結合を反射2進Cubematとよぶ。図2は,  $n=2^6(r=2^3, m=3)$ の反射2進Cubematの2次元レイアウトである。図の表し方は図1と同様である。

[定理1] ある正整数 $m$ に対して, 双方向反射2進符号

The Cubemat: a parallel processing system  
Susumu SHIBUSAWA  
Gunma University

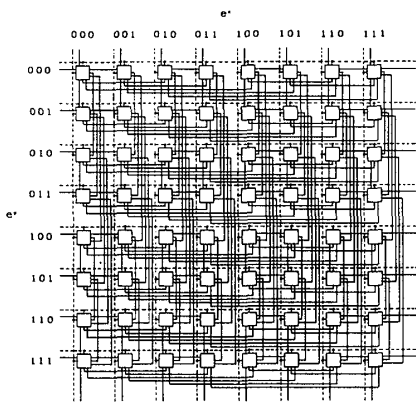


図1 単純Cubematの2次元レイアウト (n=2<sup>3</sup> × 2<sup>3</sup>)

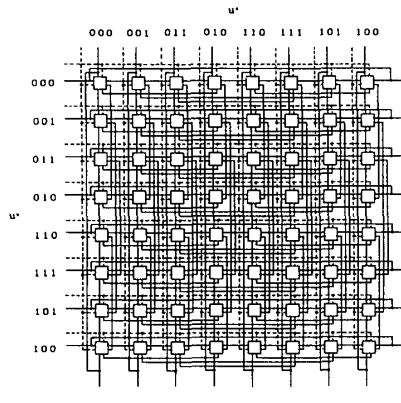


図2 反射2進Cubematの2次元レイアウト (n=2<sup>3</sup> × 2<sup>3</sup>)

```

1. procedure bitonic_sort;
   {input}
2. for J←0 to r-1 do {row}
3.   for I←0 to r-1 pardo {column}
4.     u←(I, J);
5.     send(a(u))
6.   odpar
7.   od
   {bitonic sort}
8. for i←0 to m-1 do
9.   for j←0 to i do
10.    for u←(0,0) to (r-1,r-1) pardo
11.      if i=0 then c00(u)←a(u)
12.        else if j=0 then c0j(u)←di-1,j-1(u)
13.          else cij(u)←di,j-1(u);
14.      comp(cij(ui,j))
15.    odpar
16.   od
17. od
   {output}
18. for J←0 to r-1 do
19.   for I←0 to r-1 pardo
20.     b(u)←dr-1,m-1(u);
21.   return(b(u))
22. odpar
23. od.
    
```

図3 Cubemat上でのバイトニックソート・アルゴリズム

化による $n=2^m \times 2^m$ ノードCubematのレイアウト面積 $A_r$ は、  

$$A_r = \begin{cases} 4n^2/9 + O(n^{3/2} \log^2 n) & (m: \text{偶数}) \\ 25n^2/36 + O(n^{3/2} \log^2 n) & (m: \text{奇数}) \end{cases} \quad (9)$$

ここに、 $c$ は定数を表す。

(略証) 反射2進Cubematのノードの一辺の長さは $O(m^2) = O(\log^2 n)$ であり、各行、列の配線領域の幅 $w$ は、  

$$w = \begin{cases} (2/3)\sqrt{n} + O(\log^2 m) & (m: \text{偶数}) \\ (5/6)\sqrt{n} + O(\log^2 n) & (m: \text{奇数}) \end{cases} \quad (10)$$

反射2進Cubematのレイアウトの一辺の長さは、各ノードの一辺の長さと同配線幅の和を $2^m = \sqrt{n}$ 倍して得られ、面積はその平方である。□

面積 $A_r$ は、超立方体の2次元レイアウトの下界のオーダーを満たしている。

[系]  $n=2^m \times 2^m$ ノードの単純Cubematと反射2進Cubematの2次元レイアウト面積 $A_s, A_r$ の間には、 $n$ が大きいと次の関係がある。

$$A_r \approx \begin{cases} 4A_s/9 & (m: \text{偶数}) \\ 25A_s/36 & (m: \text{奇数}) \end{cases} \quad (11) \quad \square$$

### 3. Cubemat上でのアルゴリズムの実行

#### 3.1 環境設定

Cubematと主計算機より成る構成上でアルゴリズムを実行する。主計算機はデータの入出力や整合等の役割を受け持ち、Cubematがアルゴリズム本体を実行する。各部の機能を次のように定める。

- (1) Cubemat内の各ノードは、簡単な処理能力と記憶をもつ。
- (2) データはキューブ結合またはバス結合を通して同期的に移動する。
- (3) 主計算機は、一度にある程度のデータ数を取り扱う能力を持つ。本研究では、一度に $\sqrt{n}=2^{m/2}$ 個のデータを取り扱うことができるとする。
- (4) 結合バスは、主計算機とCubemat間で一度にある程度のデータ数を送る能力を持つ。本研究では、一度に1(定数)データを送ることができるとする。

#### 3.2 バイトニックソート

$n$ データ系列 $\{a_0, a_1, \dots, a_{n-1}\}$ を、 $n$ ノード反射2進Cubemat上でバイトニックソート<sup>(5)</sup>する方法を示す。 $n$ ノードCubematのノード $u$  ( $0 \leq u \leq n-1$ )において、ソート前の任意のデータ系列を $a(u)$ 、ソート後のデータ系列を $b(u)$ とする。また、ある $i, j$  ( $0 \leq i \leq m-1, 0 \leq j \leq i$ )に対して、バ

イトニックソートの第 $i$ 段階のステップ $j$ の入出力系列をそれぞれ $c_{ij}(u), d_{ij}(u)$ とおく。このとき、バイトニックソートの第 $i$ 段階のステップ $j$ では、ノード $u$ の2進表現の第 $(i-j)$ 桁 $u_{i-j}$ の異なる隣接ノード間で、比較交換操作 $\text{comp}(c_{ij}(u_{i-j}))$ を行う。

[アルゴリズム1]  $n$ ノードCubemat上で $n$ データをバイトニックソートする操作を図3に示す。

入力: 任意の $n$ データ系列 $a(u)$  ( $0 \leq u \leq n-1$ )。

出力: ソートされた系列 $b(u)$ 。

図中記号for...do...odは順序的な操作、for...pardo...odparは並列的な操作を表す。命令 $\text{send}(a(u))$ と $\text{return}(b(u))$ は、それぞれ主計算機とCubemat間の行列化したデータ $a(u), b(u)$ の送信と返信を表す。行4はデータ $a(u)$ の行列化を意味する。バイトニックソートの中心は行11~14である。□

[定理2] アルゴリズム1を用いて、 $n$ ノードCubemat上で $n$ データをバイトニックソートする本体の実行時間は $O(\log^2 n)$ である。また、主計算機が一度に $\sqrt{n}$ データを取り扱うことができるとき、入出力も含めたアルゴリズム1の実行時間は $O(\sqrt{n})$ である。この評価は結合バスの能力に依存しない。□

#### 4. あとがき

Cubematは、超立方体と格子結合の他に多くの結合、動作特性を包含している。本報告で述べたバイトニックソート以外に、高速フーリエ変換(FFT)、行列演算等の動作も確認されているが、今後さらに多くのアルゴリズムの評価を予定している。

謝辞 御討論頂いた京都大学工学部矢島脩三教授に深謝致します。

- 文献 (1) J. D. Ullman: "Computational Aspects of VLSI", Computer Science Press(1984). (2) G. C. Fox: "The hypercube as a supercomputer", Proc. Second Int. Conf. Supercomputing, Vol. I, pp.186-193(1987). (3) 矢島脩三: "計算機の機能と構造", 岩波(昭57). (4) 渋谷進: "CCCネットワークのレイアウトの一評価", 信学論(D), J71-D, 7(1988). (5) S. G. Akl: "Parallel Sorting Algorithms", Academic Press(1985).