3次元メッシュモデルにおける無情報経路指定

1H－4

宮野 英次
九州大学大学院システム情報科学研究科

岩間 一雄
京都大学工学部

1 はじめに

3次元メッシュ計算機(3Dメッシュ)はN個のプロセッサを$N^{1/3}\times N^{1/3}\times N^{1/3}$の立方体格子点上に配置し，隣合うプロセッサと局所通信を行う並列モデルである．ラウティングとは，各プロセッサにそれぞれ1個ずつ入力として与えられたパケットを，指定された正しい行き先へと移動させる問題である．

本論文では，3Dメッシュ上でのラウティングについて考察し，$1.15\sqrt{N}+o(\sqrt{N})$時間で動作する最適無情報ラウティングアルゴリズムを示す．本アルゴリズムは次のような2つの条件を満たす．(i) すべてのパケットは最短経路を通る．(ii) すべてのパケットは高々3回しか動く方向を変えない(3屈曲条件)．2屈曲条件の元での無情報ラウティングには$\Omega(N^{2/3})$時間を必要とすることが既に知られている[IM97]．本稿の結果は，3屈曲条件へと条件を僅かに緩めるだけで，ラウティングを飛躍的に高速化することができることを示している．

2 無情報ラウティング

一言で言うと，ラウティングアルゴリズムはそれぞれのパケットの移動経路を決定することであると言える．パケットの経路が最初の場所(プロセッサの番号)と行き先のみによって完全に決まる時，アルゴリズムは無情報であると言う．また，すべての経路が高々b回しか向きを変えない時，アルゴリズムはb屈曲であると言う．

最大次数がdである任意のNプロセッサネットワーク上での無情報ラウティングの下限は$\Omega(\sqrt{N}/d)$であることが知られている[KKT91]．ハイパーキューブ網の場合には，その下限が同時に上限にもなっている[KKT91]．2Dメッシュの場合にも上限と下限が$2\sqrt{N}-2$時間で完全に一致する[Lei92]．しかし，3Dメッシュの場合には，一般的な$\Omega(\sqrt{N})$の下限，および次のような条件の元での$\Omega(N^{2/3})$の下限が知られているのみである[IM97]．

定理1 [IM97]．3Dメッシュ上での決定性，無情報，最短経路，2屈曲ラウティングアルゴリズムは$\Omega(N^{2/3})$時間を必要とする．

これらの条件を緩めた場合にも$\Omega(N^{2/3})$時間を必要とするのか，それとも$O(\sqrt{N})$時間の最適な無情報ラウティングアルゴリズムが存在するのかについては未解決のままであった．この未解決問題に対して，本稿では次のような結果を得ることができた．

定理2. 3Dメッシュ上で$1.15\sqrt{N}+o(\sqrt{N})$時間の決定性，無情報，最短経路，3屈曲ラウティングアルゴリズムが存在する（証明は次節）．

3 最適アルゴリズム

簡単のため，3Dメッシュのプロセッサ数をn^3とし，$O(n\sqrt{n})$時間ですべてのパケットを正しい場所へと移動する決定性無情報ラウティングアルゴリズムを示す．x_i平面$(1\le i\le n)$は$x=i$で定まる2次元平面を表す．y_j平面，z_k平面についても同様．x_iy_j線は$x=i$，$y=j$で定まるn個のプロセッサからなる1次元配列を意味する．y_jz_k線，z_kx_i線についても同様．

まず，アイデアを明確にするため，$2n\sqrt{n}+o(n\sqrt{n})$時間の簡単なアルゴリズムを示す．$n\times n\times n$の3Dメッシュは図1のように分割される．*x-zone*[1]は上から$\sqrt{n}$平面のプロセッサを表す．同様に，*x-zone*[2]は次の$\sqrt{n}$平面を表す．それぞれの平面は$\sqrt{n}\times\sqrt{n}$のブロックに分割される．z_k-*block*$[i,j]$はz_k平面上の上からi番目，左からj番目の$\sqrt{n}\times\sqrt{n}$の部分2次元メッシュを表す．それぞれのz_k-*block*$[i,j]$の上からj行目の$\sqrt{n}$個のプロセッサを中継プロセッサと呼ぶ．一つのz_kx_i線上の中継プロセッサの数は正確に$\sqrt{n}$個である．

アルゴリズム 3-bend

ステージ1. 行き先が*x-zone*[i]のプロセッサであるパケットを，同じ*x-zone*[i]内の中継プロセッサまでx軸方向に移動させる．

ステージ2. 中継プロセッサまで移動したパケットを正しいy座標まで，y軸方向に移動させる．

Oblivious Routing on Three-Dimensional Mesh Models
Eiji MIYANO (Kyushu University)
Kazuo IWAMA (Kyoto University)

ステージ**3**．すべてのパケットを正しい x 座標まで，x 軸方向に移動させる．

ステージ**4**．すべてのパケットを行き先まで，z 軸方向に移動させる．

補題1． アルゴリズム **3-bend** は高々 $2n\sqrt{n}+o(n\sqrt{n})$ ステップですべてのパケットを行き先へと移動する．

証明． (1) ステージ1において，すべてのパケットは遅れることなく移動できるので，高々 n ステップを必要とする．(2) ある一つの z_kx_i 線上の中継プロセッサの数は正確に $\sqrt{n}$ 個であり，それぞれの中継プロセッサは高々 n 個のパケットを保持するので，ステージ1終了後 z_kx_i 線上に集まったパケットの総数は $n\sqrt{n}$ 個である．最後に動いたパケットが正しい平面上のプロセッサに到着するのに n ステップを必要とするので，ステージ2は高々 $n\sqrt{n}+n$ ステップを必要とする．(3) ステージ2終了後，すべてのパケットは $\sqrt{n}\times n$ の部分 y_j 平面上に到着しているので，あるリンクを通るパケットの総数は高々 $n\sqrt{n}$ 個である．よって，最後のパケットが正しい線上のプロセッサに到着するのに高々 $n\sqrt{n}+\sqrt{n}$ ステップを必要とする．(4) $2n$ ステップですべてのパケットは正しい行き先へと移動することができる．以上より，高々 $2n\sqrt{n}+o(n\sqrt{n})$ ステップですべてのパケットを正しい行き先へと移動することができる． □

パケット全体を3つのグループに分け，各グループのパケットが別々の方向に移動することにより，さらに効率良くラウティングを行うことができる．次のようなアルゴリズム **Rev-3-bend** を考える．図2で "X" のグループのパケットは最初に x 軸方向，次に y 軸方向，x 軸方向，z 軸方向に移動を行う．"Y" または "Z" のパケットについては最初に y 軸または z 軸方向に移動し，各ステージで他のグループと異なる方向に移動することにより最終的な行き先へと移動する．ここで，ブロックは $\sqrt{n/3}\times\sqrt{n/3}$ の2次元メッシュへと変更する．

補題2． アルゴリズム **Rev-3-bend** は高々 $1.15n\sqrt{n}+o(n\sqrt{n})$ ステップですべてのパケットを行き先へと移動する．

証明． 補題1の証明とほぼ同様である．ただし，ステージ1で各パケットの保持するパケットの数が高々 $n/3$ であり，ある z_kx_i 線上の中継プロセッサの数は正確に $\sqrt{3n}$ 個である．以上からステージ2および3はそれぞれ高々 $n\sqrt{n/3}+o(\sqrt{n})$ ステップで十分である．$2/\sqrt{3}\approx 1.15$ より本補題が成り立つ． □

上述のアルゴリズムのステージ1では，行き先と同じ x-$zone$ 内の中継プロセッサまでパケットを移動させたために最短経路条件を満たしていない．しかし，一つ前の中継プロセッサで移動を止めることにより最短経路条件を満たすようにアルゴリズムを変更したとしても次のような補題を導くことができる．

補題3． 3D メッシュ上で $1.15\sqrt{N}+o(\sqrt{N})$ 時間の決定性，無情報，最短経路，3屈曲ラウティングアルゴリズムが存在する．

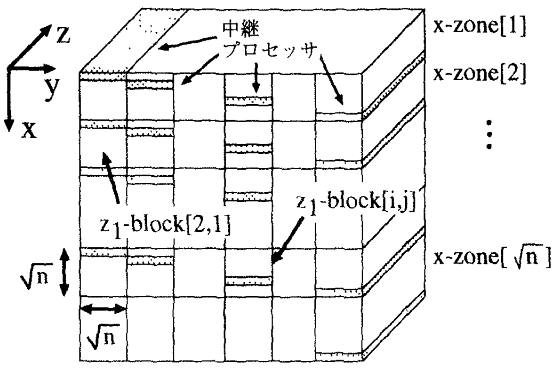

図 1: zone，block，中継プロセッサ

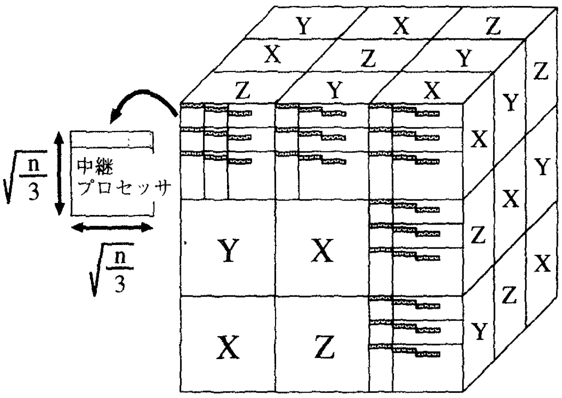

図 2: 3つのグループ

参考文献

[IM97] K. Iwama and E. Miyano, "Three-dimensional meshes are less powerful than two-dimensional one in oblivious routing," *ESA'97*, to appear.

[KKT91] C. Kaklamanis, D. Krizanc and A. Tsantilas, "Tight bounds for oblivious routing in the hypercube," *Math. Systems Theory 24* (1991) 223-232.

[Lei92] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes,* Morgan Kaufmann (1992).