計算機アーキテクチャ　95－11
(1992. 8. 19)

階層構造をもつ分散メモリ型超並列計算機ＭＡＮＤＡＬＡの設計

加納卓也　広田勝久　藤本茂訓　Andrew Flavell　高橋義造
徳島大学工学部知能情報工学科
E-mail: {taky, hiro, water, flavell, taka}@n30.is.tokushima-u.ac.jp

超並列計算機を実現するには、スケーラビリティ、物理的な階層構造、高速ノード間通信を考慮する必要がある。我々が提案したＭａｎｄａｌａは優れたスケーラビリティをもつ階層構造相互結合網であり、物理的な階層構造による制約を殆ど受けないため、超並列計算機向きである。また、20Mバイト／秒の高速通信を実現するために、相互結合網を通信専用のハードウェアであるルータによって構成し、その設計パラメータをネットワークシミュレータを用いて決定した。本稿では、超並列計算機網としてのＭａｎｄａｌａの有用性を述べるとともに、シミュレーションによって得られた数々の事実を報告する。

Design of Hierarchically Structured Distributed-Memory Massively Parallel Computer MANDALA

Takuya Kanoh　Katsuhisa Hirota　Shigenori Fujimoto　Andrew Flavell　Yoshizo Takahashi
Department of Information Science and Intelligent Systems,
Faculty of Engineering, Tokushima University
2-1 Minami-josanjima-cho, Tokushima 770, Japan
E-mail: {taky, hiro, water, flavell, taka}@n30.is.tokushima-u.ac.jp

For realizing massively parallel computers, one must consider scalability, physically hierarchical structure, and high-speed internode communication. We have proposed the Mandala network, which features hierarchical structure and excellent scalability. Moreover, since Mandala is hardly limited by the physically hierarchical structure, it is suitable for massively parallel computers. As we intend to implement the interconnection network with hardware-dedicated 20-Mbytes/sec routers, we have developed a simulator to set the design parameters of the network. Here, we describe the appeal of the Mandala network for massively parallel computers and several of its features extracted through simulation.

1.はじめに

近年、並列処理の研究が盛んに行われ、要素プロセッサが数千～数十万台規模の超並列計算機が注目されるようになったが、その現実的なアーキテクチャはいまだ開発されるにいたっていない。そこで、本研究では、ＭＩＭＤ構造をもつ分散メモリ型超並列計算機の現実的なアーキテクチャを確立し、その鋳型となるべきプロトタイプの開発を目標とする。

さて、数千台以上のプロセッサで構成される超並列計算機を実現するには、プロセッサ間の高速通信を可能にする高性能の相互結合網を開発する必要がある。本稿では、我々が昨年提案したMandala結合網や多進木のような階層構造をもつ相互結合網が超並列計算機に対して有効であることを述べ、また、シミュレータを用いることによって相互結合網の重要な構成要素であるルータの設計パラメタを決定した。そして最後に、現在開発中の小規模プロトタイプについて述べる。

2.超並列計算機の相互結合網

2.1 超並列計算機の相互結合網の要件

超並列計算機の相互結合網としては、以下の要件が挙げられる。この節では、これらの要件の実現法について検討する。

・高速通信の実現
・任意ＰＥ間通信の実現
・スケーラビリティ

2.1.1 高速通信の実現

高速に通信を行うには、以下のような方法がある。

（1）通信バンド幅を大きくする

バンド幅を大きくするには、ノード間の結線数（1チャネルあたりの信号のビット数）を増やす方法と、高速なデバイスを用いる方法がある。この２つを同時に満たす方法として、高密度集積化がある。ここで、結線数はノードの次数とトレードオフの関係があり、次数は通信距離に関係があることを考慮する必要がある。

（2）高速ルーティングアルゴリズムを用いる

送信先の競合がない場合は、回線交換が高速であるが、実際には競合するため、蓄積交換（パケット交換）を用いる。パケット交換の場合には、バーチャルカットスルー[1]、ワームホールルーティング[2]などのアルゴリズムを用いることができる。

（3）中継ノードを少なくする

通信距離が短くなるようなトポロジを用いる。ただし、ノード間通信の速度にばらつきがあるような場合は、これも考慮する必要がある。

（4）通信専用のハードウェアを用いる

（5）通信の障害を少なくする

多くの場合、リソースの占有が通信を妨げる原因となるため、これをおさえる必要がある。これには、ホットスポットをつくらないようにする。または、ホットスポットにバンド幅の大きい結合を用いる。

2.1.2 任意ＰＥ間通信の実現

任意のＰＥ間で通信を行うには、全てのＰＥを全てのＰＥと接続できるようにすればよい。これには、完全結合、クロスバ、多段網などの相互結合網を用いればよいが、超並列化には向かない。そこで、各ノードは中継する機構を用意する必要がある。

2.1.3 スケーラビリティ

スケーラビリティを得るには、Mandala結合網[3]、木などのような繰り返し構造を持つトポロジを用いればよい。それ以外のトポロジの場合は、各ノードの次数を必要以上に用意すればよいが、無駄を生じてしまう。

2.2 相互結合網における階層構造

超並列計算機を実現するためには、その物理的な構造をふまえたアーキテクチャを考えなければならない。超並列計算機は、コストパフォーマンス、フォールトトレランスなどの観点より、高密度集積化が大前提となる。つまり、１つのＶＬＳＩチップのなかに、１個あるいは複数個の要素プロセッサを詰め込む必要がある。そして、そのチップは基板上に実装され、それらがマザーボードによって接続される。ここに物理的な階層構造が存在していることに注意しなければならない。低密度、低並列のシステムでは無視できた物理的な階層構造が、高密度、高並列のシステムでは、その実現に大きく影響を及ぼすようになる。超並列計算機では、この階層構造を考慮した要素プロセッサのクラスタリングが重要課題となり、それに適した相互結合網の選択を行なわねばならない。

以上のことをふまえた超並列計算機の構成図を図２.１に示す。まず数個の要素プロセッサを１つのグループとしてまとめ、それをレベル１のクラスタとする。そして、そのクラスタ内の相互結合網をレベル１の相互結合網とする。次にレベル１のクラスタ数個をグループ化し、それをレベル２のクラスタ、そのクラスタ内の相互結合網をレベル２の相互結合網とする。以下同様にしてレベルｎの相互結合網によってレベルｎのクラスタを構成していく。このクラスタリングは物理構造に

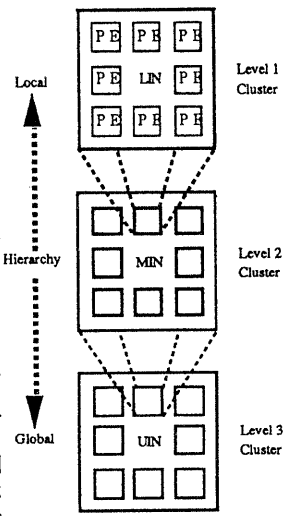

図２.１ 超並列計算機の構造

うまくマッピングできるように行なう必要がある。

図2.2は階層構造を考慮したシステムの例である。この例では、レベル1の相互結合網は完全結合、レベル2のそれはリングである。ここで1つのクラスタからさらに上位の相互結合網のために出すリンクの数を、そのクラスタの結合次数と呼ぶことにする。

図2.2の例では、レベル1のクラスタの結合次数は2である。超並列計算機を実現するためには、この結合次数があまり大きくならないようにしなければならない。結合次数が大きくなると、物理的な制約から、1つ1つのリンクのバンド幅が小さくなる恐れがあるからである。逆に、結合次数が小さいと、通信の集中が起こりやすくなるが、局所的な通信が多くなると問題は軽減される。

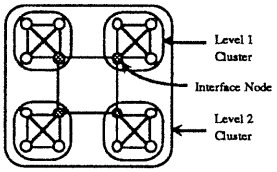

図2.2 階層構造を考慮したシステムの例
(完全結合/リング)

また、上位の相互結合網へのリンクを持つクラスタまたは要素プロセッサをインターフェイスノードと呼ぶ。このインターフェイスノードの選び方によっても、システム全体の特質が変化する。

2.3 クラスタ結合次数

ここでは、超並列計算機の相互結合網として考えられるいくつかの代表的な相互結合網のトポロジについて、要素プロセッサのクラスタリングを行なった場合の各クラスタの結合次数の観点から評価する。Nをシステム全体のノード数(要素プロセッサ数)、Cをクラスタ化係数とする。クラスタ化係数は、1つのクラスタ内に含まれる1レベル下位のクラスタの数を表す。ここではCは定数とする。

2.3.1 2進ハイパーキューブ

レベルLのクラスタの結合次数は、$(\log N-L\log C)C^{L}$で、ノード数、レベルが増えるにつれ大きくなり、超並列向きとはいえない。例えば、N=65,536、C=16の場合、レベル3のクラスタの結合次数は16,384となり、非実現的である。階層的に2進ハイパーキューブを用いた相互結合網(binary hypercube-based HIN)[4]では、logN-LlogCに改善される。

2.3.2 トーラス

レベルLのクラスタの結合次数は、$4C^{L/2}$で、レベルが増えるにつれて大きくなり、超並列向きとはいえない。例えば、N=65,536、C=16の場合、レベル3のクラスタの結合次数は256となり、これも非実現的である。

2.3.3 木

各クラスタの結合次数はレベル、ノード数などにかかわらず、常に1であり、超並列向きである。

2.3.4 Mandala結合網

我々が提案したMandala結合網は完全結合を再帰的に組み合わせたものである(図2.3)。これの各クラスタの結合次数は、レベル、ノード数などにかかわらず、常にCであり、超並列向きである。

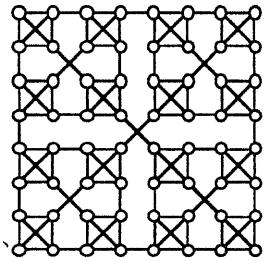
図2.3 Mandala
(C=4、L=3)

2.4 階層構造を考慮した平均通信時間

超並列計算機のように大規模なネットワークを構築する場合には、物理的な階層構造を考慮する必要がある。例えば、3cm角のシリコンチップ内部で通信を行った場合と、30cm角の多層プリント基板内で通信を行った場合では、通信時間は同じであるはずがない。一般に通信時間は物理的な通信距離にほぼ比例すると考えられるから、シリコンチップの内と外とでは、通信時間に約10倍の差があると考えられる(例えば200MBバイト/秒と20Mバイト/秒のように)。以下では、こういったことを考慮して、多少乱暴ではあるが、各階層間の通信時間の格差を10倍として、階層化が進んだ場合の各種相互結合網の平均通信時間(T)がどのように変化するかを考察してみた。なお、最下位階層でノード間の通信時間を1、各階層でのクラスタ数(クラスタ化係数:C)を16とした。

2.4.1 2進ハイパーキューブ

各階層でのトポロジも2進ハイパーキューブとなる。

レベル1 : $T=(\log C)/2=2$

レベル2 : $T=10(\log C)/2+(\log C)/2=11(\log C)/2=22$

レベル3 : $T=100(\log C)/2+10(\log C)/2+(\log C)/2$
$=111(\log C)/2=222$

レベル4 : $T=1000(\log C)/2+100(\log C)/2$
$+10(\log C)/2+(\log C)/2$
$=1111(\log C)/2=2222$

2.4.2 トーラス

各階層でのトポロジは$\sqrt{C}\times\sqrt{C}$のメッシュとした。全体ではトーラスとする。

レベル1 : $T=\sqrt{C}=2$

レベル2 : $T=10\sqrt{C}/2+(\sqrt{C}/2-1)(\sqrt{C}-1)+(\sqrt{C}-1)+3\sqrt{C}/8$
$=27.5$

レベル3 : $T=100\sqrt{C}/2+(10+\sqrt{C})(\sqrt{C}/2-1)(\sqrt{C}-1)$
$+(10+\sqrt{C}+1)((\sqrt{C}-1)+3\sqrt{C}/8)=309.5$

レベル4 : $T=1000\sqrt{C}/2$
$+(100+10\sqrt{C}+C)(\sqrt{C}/2-1)(\sqrt{C}-1)$
$+(100+10\sqrt{C}+10+\sqrt{C}+1)*((\sqrt{C}-1)+3\sqrt{C}/8)$
$=3165.5$

2.4.3 16進木

PEはリーフにのみ存在するものとした。結果は

2進ハイパーキューブと等しい。

レベル１：T=2

レベル２：T=10*2+2=22

レベル３：T=100*2+10*2+2=222

レベル４：T=1000*2+100*2+10*2+2=2222

2.4.4 Mandala結合網

クラスタサイズ１６のMandala結合網。結果は真値より小さくない近似値とした。

レベル１：T=(C-1)/C≒1

レベル２：T=10(C-1)/C+2(C-1)/C=12(C-1)/C= 11.25

レベル３：T=100(C-1)/C+2*10(C-1)/C+4(C-1)/C
=124(C-1)/C=116.25

レベル４：T=1000(C-1)/C+2*100(C-1)/C
+4*10(C-1)/C+8(C-1)/C
=1248(C-1)/C=1170

結果的にMandala結合網が最もよく、次いで、16進木と２進ハイパーキューブとなることがわかった。各階層間の通信時間の格差が３倍程度以上なら同様の結果が得られる。これは、最も通信時間が遅く、ボトルネックとなっている上位レベルの相互結合網をできるだけ通過しないような、いいかえれば、上位の相互結合網が密であるような相互結合網が有利であることを示している。無論、当然あるべき通信の局所性を考慮した場合は、下位の相互結合網が密であるものが有利となる。従って、各レベルの相互結合網が密であるMandala結合網、多進木などは、超並列計算機に非常に有利なトポロジであるといえる。

2.5 クラスタ結合次数・平均通信時間積

ノード間の結線数（ビット数）とノードの次数がトレードオフの関係にあることはすでに述べた。ここで、階層構造を考えた場合、各クラスタからその外部へと接続されているリンクの数、すなわち結合次数も、ノードの次数と同様に結線数とトレードオフの関係を持つ。一般に、結線数と結合次数の積はハードウェアの物理的な大きさからある程度の値に制限される。

このため、結線数と結合次数をバランスよく決めてやる必要がある。例えば、シリアルで１６チャネルの通信を行うか、４ビットで４チャネルの通信を行うか、１６ビットで１チャネルの通信を行うかを、うまく選ばなければならない。それでは、どう選べばよいのか。通信に局所性があれば、結線数を多くした方が、リソースの有効利用ができる。また、通信が疎らであれば、結線数を多くした方が、スループットが短くなり有利である。ただし、結合次数を小さくする場合は、インターフェイスノードをうまく選ぶ必要がある。また、結合次数が小さいと、フォールトトレラントが問題となるが、結線数の方でカバーすることも不可能ではないだろう。

また、ノード間の結線数は、ノード間通信の速度にほぼ比例するから、同様のハードウェアスペックを用いたとすると、結合次数は、ノード間通信時間にほぼ比例すると考えてよい。これを考慮して、２.４節「階層構造を考慮した平均通信時間」で得られた評価式の各項ごとに、それぞれのトポロジの各レベルでのクラスタの結合次数をかけると、以下のようになる。この積をクラスタ結合次数・平均通信時間積と呼び、この値が小さいほど平均的に通信が速く行なえることを示す。ここで、ノード数は65536（レベル数４）とする。

２進ハイパーキューブ：
2000*4*4096 + 200*8*256 + 20*12*16 + 2*16
= 33,181,472

トーラス：
2000*256 + (100*64 + 40*16 + 16)*3
+ (100*64 + 40*16 + 10*16 + 4 + 1)(3 + 3/2)
= 565,590.5

１６進木：
2,222 * 1 = 2,222

Mandala結合網：
1,170 * 16 = 18,720

これより、１６進木、Mandala結合網の順に高速通信が可能であることがわかる。

3.ネットワークシミュレーション

3.1 ネットワークシミュレーションの目的

現在、超並列計算機を構成するために数多くの相互結合網が考案されているが、汎用目的は当然のこと、特定目的についてもこれらの相互結合網の優劣を判断しがたい。これは我々が提案したMandala結合網についても同様のことがいえる。そこで我々はネットワークシミュレータ（DIENS: DIversified Evaluation Network Simulator）を開発し、Mandala結合網および他の相互結合網の動作状況を様々な条件下でシミュレートすることによって、実際に近い性能評価を行うことができるようにした。この章では、シミュレータの説明およびこれを用いたMandala結合網の評価を行なう。

3.2 ネットワークシミュレータの機能

今回、開発したネットワークシミュレータ(DIENS)では、多数のプロセッサを計算機上で仮想的に結合することによって相互結合網を構築し、これを用いてパケット交換方式によるプロセッサ間通信をシミュレートすることができる。特定条件下における１回のシミュレーションの結果はメッセージの平均通信時間およびパケットの衝突によってできる通信時間の遅延割合として得られる。これを様々な条件下で実行し得られた結果を総合的に評価することによって、個々の相互結合網についてその超並列計算機網への可能性を打診できる。ここでは、多種の相互結合網を様々な条件下でシミュレートできるように開発されたDIENSの機能を説明する。

3.2.1 相互結合網

DIENSは表３.１の相互結合網をシミュレートでき

る。またシミュレート可能な相互結合網の設定範囲もこの表に載せる。

相互結合網名	設定可能範囲		
Ｍａｎｄａｌａ結合網		レベル	クラスタサイズ
	可変範囲	１～５	３～１６
ハイパーキューブ		次元	基数
	可変範囲	１～８	２～１６
無向Ｋａｕｔｚ		桁数	基数
	可変範囲	１～７	２～　８
多進木		高さ	基数
	可変範囲	１～７	２～１５
トーラス		次元	１辺数（基数）
	可変範囲	１～８	２～１６

表３．１ シミュレートできる相互結合網とその設定範囲

３．２．２　バッファのブロック数とブロックサイズ

DIENSでシミュレートできるバッファのブロック数とブロックサイズを表３．２に示す。また、ブロック数やブロックサイズを自動的に変化させながらシミュレートできるようにした。

設定項目	設定範囲	設定値の単位
ブロック数	４，８，１６，３２，・・・・，１６ｋ	[word]
ブロックサイズ	１，２，４，８，・・・・，１２８	[block/buffer]

表３．２ シミュレートできるブロック数とブロックサイズの範囲
ただし1[word]は2[byte] である。

３．２．３　メッセージのサイズと生成速度

プロセッサ間通信をシミュレートさせるには、通信に使われるメッセージのサイズや生成速度（１秒間に１プロセッサが生成するメッセージ数）を設定できるようにしなければならない。DIENS は表３．３に示すような設定が可能である。またバッファのブロック数やブロックサイズと同様に、メッセージサイズとメッセージ生成速度を自動的に変化させながらシミュレートできる。

設定項目	設定範囲	設定値の単位
メッセージサイズ	１，４，１６，６４，・・・・，２５６ｋ	[word]
メッセージ生成速度	0.01, 0.1, 1, 10, ・・・・, 10万	[message/sec]

表３．３ シミュレートできるメッセージのサイズと生成速度
ただし1[word]は2[byte] である。

３．２．４　プロセッサ間通信の局所性

Mandala結合網やトーラス結合網などを用いて大規模な相互結合網を構成すると平均距離が非常に大きくなってしまう。従ってこのような場合、与えられた問題の中から通信の局所性を導き出して通信のオーバーヘッドを抑えることが非常に重要となる。実際の問題中ではこのような通信の局所性のため、平均通信距離やメッセージの平均通信時間は理論値との間にずれが生じる。よって、ネットワークを評価する上で通信の局所性も大いに考慮すべきであるが、これは理論的な特徴に反映させるのは難しい。が、その一方シミュレーションでならこれを通信所要時間等の特徴に反映させ易い。本シミュレータでは生成されたメッセージの送信先ノードへの距離の平均が”ある一定値”になるように送信先ノードを決定することによって、局所性のある通信をシミュレートすることができる。この”ある一定値”は理論上の平均通信距離より小さくかつ１以上の値である。

３．２．５　通信デッドロックの検出

閉ループをもつネットワークではデッドロックが生じる。従ってシミュレーションでも通信のデッドロックが発生するので、DIENSはこれを検出しデッドロック状況をユーザに知らせる機能を持つ。

３．２．６　その他の特徴

DIENSは次のような仮定を用いている。

・20Mバイト／秒のルータ間通信を行なう。
・１チャネルは16ビット幅である。

３．３　シミュレータによる通信衝突の評価

プロセッサ間通信では、プロセッサが発生する通信量が多すぎると相互結合網内でパケットが衝突するため通信所要時間に遅れが生じる。したがって我々はこの遅れをできるだけ減らすように相互結合網を設計しなければならない。そこでMandala結合網においてどの程度の通信所要時間に遅れが生じるかをDIENSによるシミュレーションによって調べる。

表３．４のシミュレーション条件Ａを用いてシミュレートを行った結果をグラフにしたものが図３．５である。この図は横軸にメッセージ生成速度（１つのプロセッサが１秒間に生成するメッセージ数）、縦軸に３．１式で計算される通信の衝突割合をとっている。

このグラフからレベル２クラスタサイズ４のMandala結合網は、サイズが８ｋバイトのメッセージを１秒間に１万回以上送信するとブロック数をいくら多くしても通信の衝突を減らすことはできない。

$$\text{通信の衝突割合} = \frac{\text{パケット衝突による通信の遅れ時間}}{\text{衝突無しの通信時間}} \times 100\,[\%]$$

・・・・（式３．１）

固定条件		
相互結合網	Ｍａｎｄａｌａ結合網	
	レベル２，クラスタサイズ４	
メッセージ	メッセージサイズ	８ｋバイト
バッファ	ブロックサイズ	２５６バイト
	ブロック数	４，３２
変化条件		
メッセージ	メッセージ生成速度	１秒間に１０～１０万個

表３．４　シミュレーション条件Ａ

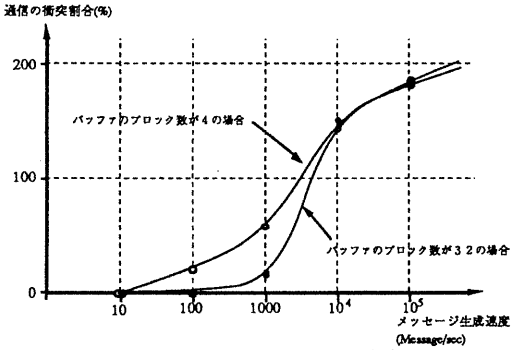

図３.５ メッセージ通信の衝突割合

３.４ デッドロック発生確率の評価

DIENSの通信デッドロックの検出機能を使えば、任意のネットワークに対してデッドロックの発生確率を求めることが出来る。そこでシミュレーション条件Ａにおいて各条件ごとに20回づつシミュレートし、それぞれについてデッドロック発生率を求めてグラフにしたものが図３.６である。

このグラフをみると図３.５のグラフと非常に似通っていることが分かる。すなわちこれはデッドロック発生確率が通信の衝突割合に比例して発生することを意味する。

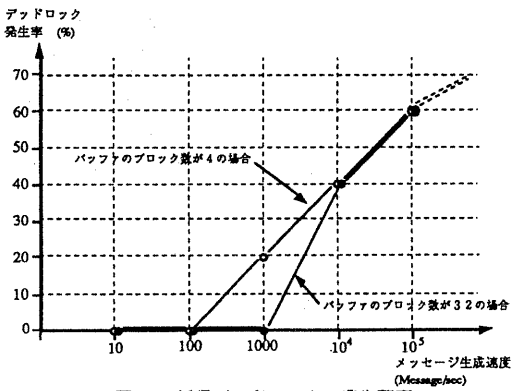

図３.６ 通信デッドロックの発生確率

３.５ メッセージサイズと通信時間の関係

図３.８は、シミュレーション条件Ｂを使って、各相互結合網におけるメッセージサイズと通信時間比の関係を表したグラフである。ただし通信時間比は式３.２で計算される。また、ブロックサイズについてはメッセージサイズごとに最適なものを選んだ。（メッセージサイズと最適なブロックサイズの関係については４章で述べる）

$$\text{通信時間比}=\frac{\text{各相互結合網の通信所要時間}}{\text{２進ハイパーキューブ網の通信所要時間}} \quad \cdots\cdots (式３.２)$$

このグラフから、メッセージサイズが小さくなるほどMandala結合網は他の相互結合網（特に２進ハイパーキューブや無向Kautz網）と比べ、相対的に通信時間が長くなるのがわかる。

固定条件		
相互結合網	Ｍａｎｄａｌａ結合網	レベル２ クラスクサイズ４
	ハイパーキューブ	２進 ８次元
	無向Ｋａｕｔｚ	６進 ３桁
	多進木	２進 高さ７
	トーラス	２次元 １辺数１６
メッセージ	メッセージ生成速度	１秒間に１０個
バッファ	ブロック数	４
変化条件		
メッセージ	メッセージサイズ	16～16kバイト
バッファ	ブロックサイズ	8～256バイト

表３.７ シミュレーション条件Ｂ

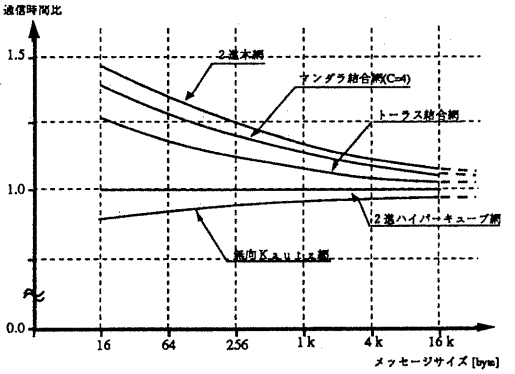

図３.８ ２進ハイパーキューブ網を基準とした各相互結合網のメッセージサイズと通信時間比の関係

4.シミュレーションによるルータの設計

プロセッサ間通信はバッファの構造に大きく影響を受けるため、できる限り高速な通信が可能となるように、我々は相互結合網のトポロジを決定するのと同様にバッファの設計にも気を配らなくてはならない。この節ではメッセージのサイズからバッファを構成するブロックの最適なサイズが求まることをシミュレーションの結果から導く。またこの結果を用いて、ルータで使用されるバッファの設計について考察する。

４.１ 最適なバッファ構造の決定

まず最初にメッセージサイズとメッセージ生成速度が決まっているとき、通信時間が最小となるような最適なバッファの構造（ブロックサイズとブロック数）が一意に決定することを図４.２を使って示す。この図は次のシミュレーション条件Ｄでシミュ

レートしその結果をグラフにしたものである。すなわちブロック数とブロックサイズを変化させていき、どの組合せのときが最も通信時間が短かったかを調べるためのグラフである。したがって横軸にブロックサイズ、縦軸にブロック数をとってあり、グラフ上の曲線は通信所要時間が等しくなるような等時間曲線を表わしている。このグラフをみると最適なバッファ構造はブロックサイズが２５６バイトでブロック数が２以上であることが分かる。

固定条件		
相互結合網	Ｍａｎｄａｌａ結合網	
	レベル２，クラスタサイズ４	
メッセージ	メッセージサイズ	８ｋバイト
	メッセージ生成速度	１秒間に１０個
変化条件		
バッファ	ブロックサイズ	８～４ｋバイト
	ブロック数	１～１２８

表４.１　シミュレーション条件Ｄ

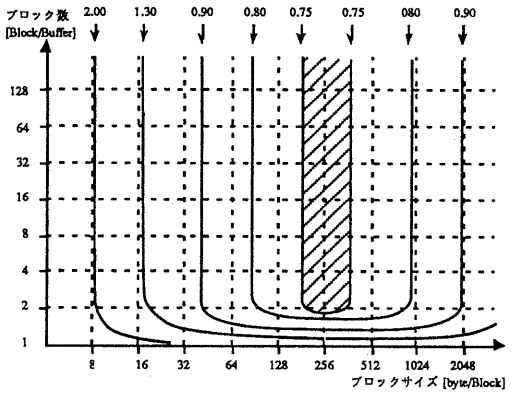

図４.２ ブロック数とブロックサイズを変化させたときの最適なバッファタイプの分布図。
（ただし等時間曲線の時間の単位はミリ秒）

さて、何故このようなグラフとなるのであろうか。まず最適なバッファ構造の範囲よりブロックサイズが小さくなると次の問題が生じる。

・メッセージのヘッダ部分の相対的な増加に伴うオーバーヘッドが生じる。
・パケットサイズ（＝ブロックサイズ）が小さくなるので逆にパケット数が増加し、これに伴うパケットの衝突が増加する。

また最適なバッファ構造の範囲よりブロックサイズが大きくなると次の問題が生じる。

・Store-and-Forwardの場合は、隣のルータに１パケット分送信し終わるまでその次のルータへ送信開始が出来ないので、パケットサイズが大きくなりすぎると中継経路中のルータによるパイプライン的な並列処理効果が得られなくなる。
・メッセージ生成時にこれをパケットに分解するとき、パケットサイズが大きいと最後のパケットが完全にデータで埋まる確率が下がる。

最後に最適なバッファ構造の範囲よりブロック数が少なくなると次の問題が生じる。

・ブロック数が少ないため１つのルータが１度にバッファリングできるパケット数が少なくなるため、パケットの衝突が頻繁に起こる。

したがって上記の問題が生じるため最適なバッファ構造が一意に求まるわけである。特に最適なブロックサイズは、メッセージのサイズおよび生成速度との間に何らかの関係があるのではないかと思われる。よって以下でこれについて考える。

４.２　ブロックサイズとメッセージ生成速度の関係

ここでは最適なブロックサイズとメッセージの生成速度との関係を調べる。図４.４は次のシミュレーション条件Ｅでシミュレートした結果のグラフ であり、グラフ中の曲線の意味は図４.２と同じで等時間曲線を表わす。いいかえれば、通信所要時間が最も短くなるような範囲の分布図である。

このグラフをみると、同じメッセージ生成速度中においては常にブロックサイズが２５６バイトのとき通信所要時間が最短となることが分かる。すなわち、通信所要時間が最短となるようなブロックサイズはメッセージ生成速度の影響を受けていない。よって消去法により、このような最適なブロックサイズはメッセージサイズにのみ影響を受けることが分かった。それでは最適なブロックサイズとメッセージサイズとはどのような関係にあるのであろうか。次にこのことについて考察する。

固定条件		
相互結合網	Ｍａｎｄａｌａ結合網	
	レベル２，クラスタサイズ４	
メッセージ	メッセージサイズ	８ｋバイト
バッファ	ブロック数	４
変化条件		
メッセージ	メッセージ生成速度	１秒間に0.01～1000個
バッファ	ブロックサイズ	８～４ｋバイト

表４.３　シミュレーション条件Ｅ

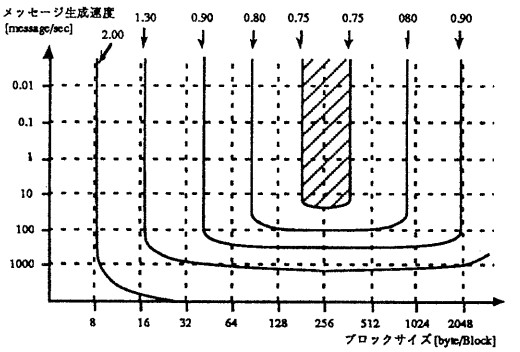

図４.４ メッセージ生成速度とブロックサイズを変化させたときの通信所要時間が最短となる範囲の分布図。
（ただし等時間曲線の時間の単位はミリ秒）

4.3　ブロックサイズとメッセージサイズの関係

表4.5は様々なメッセージサイズに対して図4.3のようなグラフを作成し、これらのグラフからそれぞれ最適なブロックサイズを求め表にしたものである。

この表をみるとメッセージサイズの増加に伴い、最適なブロックサイズも単調増加しているのが分かる。したがってこの表から最適なブロックサイズとメッセージサイズとの間の関係を近似式で表わすと式4.1のようになる。ただしこの式の右辺の分母Aは2～4の定数である。

メッセージサイズ	最適なブロックサイズ
16	8
64	16
256	64
1k	128
4k	128
16k	256

表4.5　最適なブロックサイズとメッセージサイズとの関係
（ただし表中の数字の単位はバイトである）

$$\text{最適なブロックサイズ} = A\sqrt{\text{メッセージサイズ}} \quad \cdots\cdots \quad (式4.1)$$

さて、ここで最適なブロックサイズとメッセージサイズの関係が明らかになったが、実はこのことから次のことが言える。

・ルータの設計において、ブロックサイズが可変であるようなバッファの設計を行えば、実際にメッセージを送信するときにそのメッセージサイズから式4.1を用いて最適なパケットサイズ（＝ブロックサイズ）を求めることができ、これによって最も効率のよい通信を行うことができる。

5.プロトタイプの開発

現在超並列計算機のアーキテクチャ検証のため、小規模のプロトタイプを開発している。このプロトタイプはそのままのアーキテクチャで超並列化できることを目標としている。

このプロトタイプは要素プロセッサ、ＮＩＵ(Network Interface Unit)、ルータからなる。要素プロセッサには２５ＭＨｚの３２ビットＲＩＳＣプロセッサＶＲ３６００（ＦＰＵ内臓）を使用し、メインメモリは４Ｍバイト、キャッシュは６４Ｋバイトで、約２０ＭＩＰＳの性能を持つ。ＮＩＵは、パケットの生成、組み立て、保留、管理などを行なう部分で、ＬＣＡ(Logic Cell Array)とメモリにより構成する。ルータは、クラスタサイズ４のMandala結合網およびパケット交換に対応した専用ハードウェアであり、ゲートアレイにより実現する。バーチャルカットスルールーティングおよびワームホールルーティングなどのテクニックを用いて、２０Ｍバイト／秒の速度でルーティングを行なう。

6.おわりに

超並列計算機を実現するには、物理的な階層構造を考慮する必要がある。そのため、相互結合網には、クラスタの結合次数が小さな定数となり、各階層の結合が密であるMandala結合網、多進木などのような階層構造をもつトポロジが適しているといえる。

またシミュレーションにより、バッファを構成するブロックの数は２～４個程度でよく、バッファサイズはメッセージサイズの平方根に比例することがわかった。これらの結果は今後より深い検討を行なって超並列計算機ＭＡＮＤＡＬＡのプロトタイプ設計に役立てる。

参考文献

[1] Kermani P., Kleinrock L. : Virtual Cut-Through: A New Computer Communication Switching Technique, Computer Networks 3(4), pp.267-286 (1979)

[2] Seitz C. et al. : Wormhole Chip Project Report (1985)

[3] Andrew Flavell, Takuya Kanoh, Yoshizo Takahashi : Mandala: An Interconnection Network For A Scalable Massively Parallel Computer, 情報処理学会第43回全国大会講演論文集（分冊6）,4Q-13 (1991)

[4] Sivarama P.Dandamudi, Derek L.Eager: Hierarchical Interconnection Networks for Multicomputer Systems, IEEE Transactions on Computers, Vol.39, No.6, pp.786-797 (1990)