Vol.2011-HPC-130 No.48
2011/7/28

高帯域と低遅延を両立するFat Tree結線方式の提案

中島耕太[†1]　成瀬彰[†1]
住元真司[†1]　久門耕一[†1]

本稿では，全対全通信時の経路競合を避けつつ平均ホップ数を削減するFat Treeの結線方法を提案する．Fat Treeは，Fully Bisectional Bandwidthを保証するネットワークであり，高い帯域を実現できる．一方，ネットワークの規模が大きくなるとノード間の平均ホップ数が増加するため，遅延が大きくなる．そこで，全対全通信時におけるシフト通信パターンの規則性に着目し，全対全通信時の経路競合を抑止しつつ，少ないホップ数で到達可能なノード数を増やすように結線する．これにより，標準的なFat Treeが持つ高い帯域を維持しつつ，ノード間の平均ホップ数を削減する．

提案する結線では，標準的なFat Treeと比較して，19-ary 3-tree構成において，3ホップ以内で到達可能なノードを18.1倍に増加させた．また，平均ホップ数を約36.7%削減できることを確認した．

A Proposal of Fat Tree Wiring Method to Realize High Bandwidth and Low Latency

Kohta NAKASHIMA,[†1] Akira NARUSE,[†1]
Shinji SUMIMOTO[†1] and Kouichi KUMON [†1]

This paper describes a proposal of Fat Tree wiring method in order to realize high bandwidth and low latency. Fat Tree is a network to guarantee Fully Bisectional Bandwidth and to realize high bandwidth. However, increase of network size causes increase of average hops number and network latency. In order to resolve this problem, we focus the regularity of shift all-to-all communication patterns and routing method in Fat Tree, and change network wiring which satisfies both routing congestion avoidance and increase of the number of nodes which can reach in smaller hops number. The proposal network wiring method keeps high bandwidth in regular Fat Tree and reduces average hops number between nodes.

The proposal method boosts 18.1 times larger the number of nodes within three hops than regular Fat Tree, and reduced 36.7% average hops numbers.

1. はじめに

近年，HPC分野において，多数の計算サーバを高速なネットワークで接続するクラスタシステムが広く用いられている．クラスタネットワークは，並列計算処理に必要な通信に用いられるため，高い帯域と低い遅延が求められる．これを実現するため，特に大規模なクラスタシステムにおいては，InfiniBand[1)]によるFat Tree接続が広く用いられている．

Fat Treeは，Fully Bisectional Bandwidthを保証するネットワークであり，高い帯域を実現できる．特に全対全通信において広く用いられているシフト通信パターン[2)]では，それぞれの通信フェーズにおいて，経路競合が生じないように転送先が決定される．このため，全対全通信においても，高い帯域を実現できる．

一方，標準的なFat Treeネットワークには，ネットワーク規模の増加に応じてノード間の平均ホップ数が増加し，遅延が増大する問題がある．

そこで，本稿では，全対全通信時におけるシフト通信パターンの規則性に着目し，全対全通信時の経路競合を抑止しつつ，少ないホップ数で到達可能なノード数を増やすように結線する．これにより，標準的なFat Treeが持つ高い帯域を維持しつつ，ノード間の平均ホップ数を削減する．

提案する結線では，標準的なFat Treeと比較して，19-ary 3-tree構成において，3ホップ以内で到達可能なノードを18.1倍に増加させた．また，平均ホップ数を約36.7%削減できることを確認した．これにより，平均ホップ数の削減による遅延削減の見込みを得た．

2. 標準的なFat Treeと平均ホップ数

標準的なFat Treeは，図1に示すような構成のネットワークである．最上段を除く各スイッチにおける接続リンク本数は，上側と下側で等しいため，下側から上側へ，あるいは上側から下側へ向かう帯域を十分確保できるネットワークである．このため，高い帯域を必要とするクラスタネットワークで広く採用されている．

標準的なFat Treeは，ネットワーク規模が増加すると，スイッチ段数の増加と各段のスイッチにおける上側/下側との接続リンク数である次数の増加により，ノード間の平均ホッ

†1 (株) 富士通研究所
Fujitsu Laboratories Ltd.

Vol.2011-HPC-130 No.48
2011/7/28

表 1　n-ary 3-tree

次数 (n)	ノード数	ホップ数 k で到達可能なノード数			平均ホップ数
		$k=1$	$k=3$	$k=5$	
4	64	3	12	48	4.36
8	512	7	56	448	4.72
12	1,728	11	132	1,584	4.82
16	4,096	15	240	3,840	4.87

表 2　直結環境とスイッチ経由環境での遅延時間の比較 (μs)

直結環境	スイッチ経由環境	差分
1.16	1.27	0.11

表 3　測定環境

サーバ	Fujitsu RX200S5
CPU	Xeon X5570 2.93GHz
HCA	Mellanox ConnectX2
IB-SW	Mellanox MIS5025Q
ベンチマーク	perftest 1.00
経由スイッチ数	1

プ数が増加する．但し，現在主流の構成は 3 段までの構成であるので，本稿では段数は 3 段に限定し，次数の増加の影響について議論する．

表 1 は，3 段構成 Fat-Tree(n-ary 3-tree) における次数 (n) を変化させた場合のネットワーク全体のノード数，ホップ数 k で到達可能なノード数，ノード間の平均ホップ数の関係を示している．次数が増加すると，ネットワーク全体のノード数は次数の 3 乗に比例する．3 ホップ以下で到達可能なノード数は次数の 2 乗にほぼ比例して増加する．したがって，ノード間の平均ホップ数は増加する．このように，大規模なネットワークでは平均ホップ数が増大する．

平均ホップ数の増大は遅延の増大を引き起こす．表 3 の環境における InfiniBand における HCA 直結環境とスイッチ経由環境での遅延時間を表 2 に示す．表 2 からスイッチ 1 つあたりの遅延は約 0.11μs であり，経由するスイッチが 1 つ増えると，遅延が 9.5%増加することがわかる．特に，今後，Sandy Bridge アーキテクチャのように CPU に直接 PCI Express コントローラが実装されるようになると，サーバ内の遅延時間は大きく減少すると期待できる．このような場合，全体の遅延時間に占めるスイッチ遅延の割合がさらに増加する可能性が高い．したがって，平均ホップ数をできるだけ削減し，スイッチ遅延を削減する必要がある．

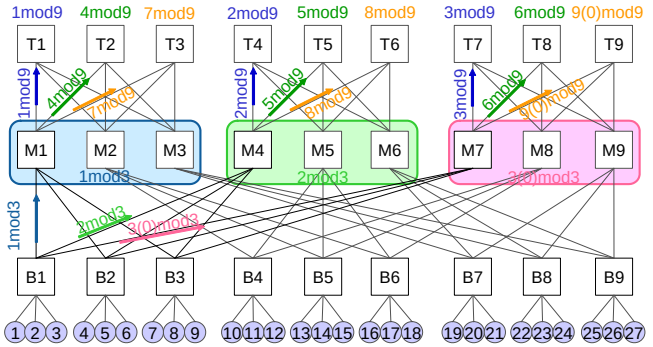

図 1　Fat Tree における標準的なルーティング

3. 全対全通信における経路競合回避

3.1 ルーティング

InfiniBand では，各スイッチに転送先ノード番号に対応する出力ポート番号の表を設定することでルーティングを実現している．すなわち，静的なルーティングである．InfiniBand における Fat Tree における標準的なルーティングでは，各ノードへの経路を決める際，経由する上段側のスイッチを順番に選択することで負荷分散を実現している[2]．図 1 のような 3-ary 3-tree の場合は，1 段目から 2 段目へ向かう経路は 3 通りある．まず，これを順番に選択する．すなわち，ノード 1 へ向かう経路は，B2～B3 からは M1 を，B4～B6 からは M2 を，B7～B9 からは M3 を選択し，ノード 2 へ向かう経路は，B2～B3 からは M4 を，B4～B6 からは M5 を，B7～B9 からは M6 を選択する．したがって，したがって，転送先ノード番号が $1mod3$ となるパケット[*1]は M1～M3 に，転送先ノード番号が $2mod3$ となるパケットは M4～M6 に，転送先ノード番号が $3(0)mod3$ となるパケットは M7～M9 へ転送される．

次に，2 段目から 3 段目へ向かう 3 通りの経路を順番に選択する．例えば，1 段目から M1～M3 にはノード 1,4,7,10,13,16,19,22,25 へ向かうパケットが到達する．これを順番に振り分けるので，転送先ノード番号が 1,10,19($1mod9$) となるパケットは T1 に，転送先ノー

⋆1 本稿では,「転送先ノード番号を N で割った時の余りが k となる」ことを「転送先ノード番号が $kmodN$ となる」と表現する．また転送先ノード番号以外の番号や数字についても同様の表現を用いる．

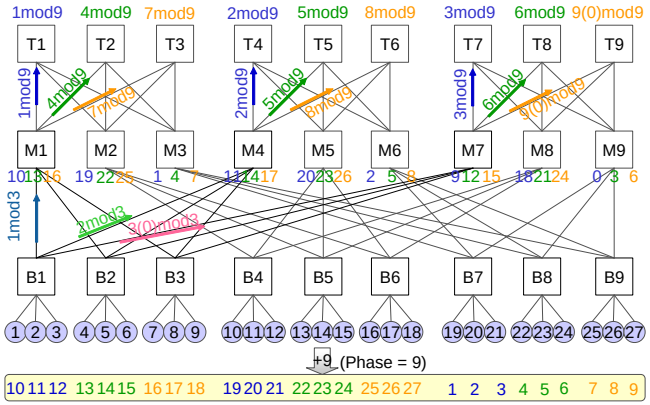

図 2 全対全通信時の経路競合回避

ド番号が 4,13,22($4mod9$) となるパケットは T2 に，転送先ノード番号が 7,16,25($7mod9$) となるパケットは T3 へ転送される．T4〜T9 も同様に図 1 の各スイッチの上に示している転送先ノード番号の経路が転送される．

3.2 シフト通信パターンによる経路競合回避

シフト通信パターンは，全対全通信通信時において広く用いられている通信パターンである．全対全通信では，全てのノードがそれぞれ全てのノードに対してメッセージを送受する．ノード数が N，各ノードの番号が p とすると，全対全通信は，N 回の通信フェーズから構成され，i 番目の通信フェーズでは，自身のノード番号から i 番先のノードへパケットを転送する．したがって，転送相手のノード番号は $(i+p)\%N$ である．

図 2 に 9 番目の通信フェーズの場合を事例として示す．9 番目の通信フェーズでは，転送先ノードは自身から 9 番先のノードになる．図 2 のように，1 段目のスイッチでは，配下のノードから転送されるパケットの転送先ノード番号は，$1mod3$，$2mod3$，$3(0)mod3$ となるため，必ず別々の 2 段目スイッチへ転送される．2 段目スイッチでは，M1〜M3 においては，1 段目のスイッチから受け取るパケットの転送先ノード番号は $1mod9$，$4mod9$，$7mod9$ となるため，必ず別々の 3 段目スイッチへ転送される．M4〜9 についても同様である．図 2 では，9 番目の通信フェーズを事例として取り上げたが，どのフェーズにおいても，各スイッチが受け取るパケットの宛先の関係は同じである．

このように，どの通信フェーズにおいても，各段のスイッチにおいて経路競合を回避できるため，標準的な Fat Tree ルーティングでは，全対全通信において高いスループットを実現できる．

4. 高帯域と低遅延を両立する Fat Tree 結線とルーティング

4.1 結線方式

4.1.1 基本方式

平均ホップ数を削減するためには，2 段目で折り返して到達する 3 ホップで到達可能なノード数を増加させる必要がある．標準的な Fat Tree では，1 段目の複数のスイッチが 2 段目のスイッチを介して接続されている．このような 1 段目の複数スイッチ同士の接続関係を近接接続と以降表現する．図 1 の例では，B1 と B2，B1 と B3，B2 と B3 は近接接続である．また，B1 と近接接続されるスイッチは B2，B3 のみである．

n-ary 3-tree の場合，ある 1 段目のスイッチを起点とすると，起点から 2 段目の n 個のスイッチと接続される．この n 個の 2 段目のスイッチがそれぞれ別々の 1 段目のスイッチと接続される場合，2 段目のスイッチは起点を除いて最大で $n-1$ 個の 1 段目のスイッチと新たに接続できる．したがって，起点のスイッチから，自身を含めて最大 $n(n-1)+1$ 個のスイッチと近接接続可能である．したがって，近接接続可能なスイッチを増やし，平均ホップ数を削減するために，できるだけ 2 段目のスイッチが別々の 1 段目のスイッチと接続するように結線する．

次に，全対全通信における経路競合回避について考察する．3.2 節で議論したように，各段のスイッチにおいてどの通信フェーズでも下側から到着したパケットを別々の上段側へ転送できるようにすれば経路競合は回避できる．すなわち，3-ary 3-tree においては，2 段目スイッチでは，M1〜M3 のいずれにおいても 1 段目のスイッチから受け取るパケットの転送先ノード番号が $1mod9$，$4mod9$，$7mod9$ となるように接続される必要がある．M4〜M6 や M7〜M9 においても同様である．

これを実現するためには，2 段目のスイッチは，1 段目のスイッチ番号がそれぞれ $1mod3$，$2mod3$，$3(0)mod3$ となる 3 つのスイッチと接続されており，かつ，1 段目のスイッチは M1〜M3，M4〜M6，M7〜M9 の 3 つのスイッチ群とそれぞれ 1 つずつ接続されていれば良い．この理由について説明する．上記条件を満たす場合における 1 段目のスイッチから 2 段目のスイッチへ転送されるパケットの転送先ノード番号の関係を表 4–6 に示す．フェーズ番号が $0mod9$ の場合，表 4 に示すように 1 段目のスイッチ番号が $1mod3$，$2mod3$，$3(0)mod3$ となる 3 つのスイッチから，M1〜M3 へ転送されるパケットの転送先ノード番号は，$1mod9$，

表 4　フェーズ番号が $0mod9$ となる場合のパケット転送先の関係

送出元スイッチ番号	2 段目が受け取るパケットの転送先ノード番号		
(1 段目)	M1～M3	M4～M6	M7～M9
$1mod3$	$1mod9$	$2mod9$	$3mod9$
$2mod3$	$4mod9$	$5mod9$	$6mod9$
$3mod3$	$7mod9$	$8mod9$	$9mod9$

表 5　フェーズ番号が $1mod9$ と合同の場合のパケット転送先の関係

送出元スイッチ番号	2 段目が受け取るパケットの転送先ノード番号		
(1 段目)	M1～M3	M4～M6	M7～M9
$1mod3$	$4mod9$	$2mod9$	$3mod9$
$2mod3$	$7mod9$	$5mod9$	$6mod9$
$3mod3$	$1mod9$	$8mod9$	$9mod9$

表 6　フェーズ番号が $2mod9$ と合同の場合のパケット転送先の関係

送出元スイッチ番号	2 段目が受け取るパケットの転送先ノード番号		
(1 段目)	M1～M3	M4～M6	M7～M9
$1mod3$	$4mod9$	$5mod9$	$3mod9$
$2mod3$	$7mod9$	$8mod9$	$6mod9$
$3mod3$	$1mod9$	$2mod9$	$9mod9$

$4mod9$，$7mod9$ であり，別々の 3 段目のスイッチへ振り分けられる．M3～M6，M7～M9 についても，同様である．フェーズ番号が $1mod9$，$2mod9$ の場合も表 5，6 に示すように 2 段目のスイッチへ転送されるパケットは，別々の 3 段目のスイッチへ振り分けられる．それ以外のフェーズ番号の場合も同様である．このように，どのフェーズにおいても，経路競合を避けることができる．

そこで，図 3 のような結線を考える．スイッチ番号が $1mod3$ のものを青，スイッチ番号が $2mod3$ のものを緑，スイッチ番号が $3(0)mod3$ のものを橙で示している．さらに，B1～B3 を Group1，B4～B6 を Group2，B7～B9 を Group3 と分類する．そして，2 段目のスイッチは，M1 については Group1 の 3 つのスイッチを 1 つずつ，M4 については Group1 からスイッチ番号が $1mod3$ であるスイッチ (青)，Group2 からスイッチ番号が $2mod3$ であるスイッチ (緑)，Group3 からスイッチ番号が $3mod3$ であるスイッチ (橙) となるように Group 番号を+1 しながら選択し，接続する．さらに，M7 については Group1 からスイッチ番号が $1mod3$ であるスイッチ (青)，Group3 からスイッチ番号が $2mod3$ であるスイッチ (緑)，Group1 からスイッチ番号が $3mod3$ であるスイッチ (橙) となるように Group 番号を+2 しながら選択し，接続する．このように接続すると，2 段目のスイッチは全てスイッチ番号が $1mod3$，$2mod3$，$3(0)mod3$ となる 1 段目のスイッチと 1 つずつ接続される．したがって，全対全通信時の経路競合を回避できる．さらに，B1 から B2，B3 だけでなく，B5，B6，B8，B9 にも近接接続するようになる．

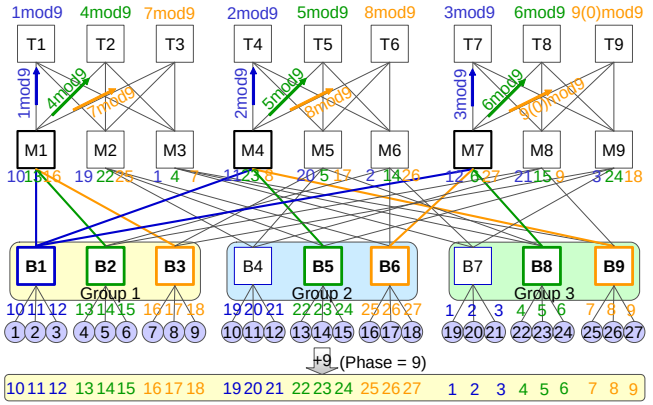

図 3　近接接続スイッチを増やした構成 (3-ary)

同様に，図 4 のような 4-ary tree について考える．B1～B4 を Group1，B5～B8 を Group2，B9～B11 を Group3，B12～B16 を Group4 と分類する．そして，2 段目のスイッチは，M1 については Group1 の 4 つのスイッチを 1 つずつ，M5 については，Group1,2,3,4 の順にスイッチ番号がそれぞれ $1mod4$，$2mod4$，$3mod4$，$4mod4$ となるスイッチを選択する．すなわち，Group 番号を+1 しながら各色を順番に選択する．M9 については，Group1,3,1,3 の順にスイッチ番号がそれぞれ $1mod4$，$2mod4$，$3mod4$，$4mod4$ となるスイッチを選択する．すなわち，Group 番号を+2 しながら各色を順番に選択する．M13 については，Group1,4,3,2 からスイッチ番号がそれぞれ $1mod4$，$2mod4$，$3mod4$，$4mod4$ となるスイッチを選択する．すなわち，Group 番号を+3 しながら各色を順番に選択する．このように接続すると，B1 から B2～B4 だけでなく，B6，B8，B10～B12，B14，B16 にも近接接続するようになる．

本手法を用いると，2 段目のスイッチでの経路競合を避けると共に，標準的な Fat Tree 接続と比較して，3 ホップで接続可能なノード数を増やすことができる．特に，3-ary 3-tree の場合からわかるように，n-ary 3-tree において，n が素数である場合，近接接続されるスイッチ数は最大の $n(n-1)+1$ となる．

4.1.2　相互直交ラテン方格による改良

次数が素数でない場合は，図 4 に示すように，M1 と M9 が共に B1 と B3 に接続されて

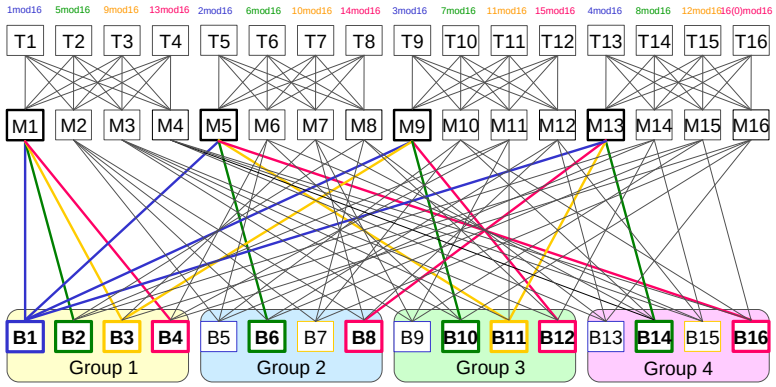

図 4 近接接続スイッチを増やした構成 (4-ary)

いる．このように，複数の 1 段目のスイッチが共通する複数の 2 段目のスイッチと接続されているため，近接接続されるスイッチ数は最大とならない．

そこで，4-ary 3-tree を例に，さらに近接接続できるスイッチ数を改善できないか考える．図 5 は図 4 の結線における 2 段目と 1 段目の接続関係を示している．例えば，M1 の行に並ぶ 4 つのスイッチ B1〜B4 は，これらが M1 と接続関係にあることを示している．図 6 は，図 5 の各スイッチ番号を Group 番号に変更したものである．

図 5 より，○と△の印で示した 1 段目のスイッチは，それぞれ共通の 2 段目のスイッチと接続されていることが分かる．○の印で示したスイッチの重なりを解消するためには，M9 の行に B3 が出現しないようにする必要がある．これを図 6 で考えると，M9 の行での Group 番号の重なりがないようにすれば良い．即ち，M9〜M12 の 4×4 の方格が，縦方向/横方向ともに同じ数が存在しない並びとなるラテン方格を構成すればよいことが分かる．次に△の印で示したスイッチの重なりを考える．図 6 において，M5〜M8 の方格と M13〜M16 の方格は共にラテン方格である．しかし，△の印のように，スイッチの重なりが生じており，互いに直交でないことが分かる．したがって，M5〜M8，M9〜M12，M13〜M16 のそれぞれの方格が相互に直交するラテン方格であれば，重なりが解消される．

相互直交ラテン方格は広く研究されており，一辺の長さを n とすると，n が素数の冪乗であるとき，$n-1$ 個の相互直交ラテン方格が存在し，それぞれを求められることが知られている[3),4)]．それ以外の n に対する相互直交ラテン方格については，最大でも $n-1$ 個以下であることは証明されているものの，最大数存在するかどうかについては現時点では明らかにされていない．

図 5 2 段目と 1 段目の接続関係

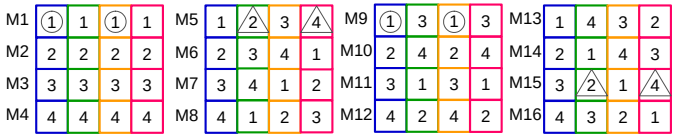

図 6 2 段目と 1 段目の Group 番号の関係

図 7 相互直交ラテン方格による Group 番号の関係

図 8 相互直交ラテン方格による 2 段目と 1 段目の接続関係

そこで，本稿では，n が素数の冪乗である場合について，$n-1$ 個の相互直交ラテン方格から最適な結線を得る方法について議論する．図 7 における，M5〜M8，M9〜M12，M13〜M16 の方格は相互に直交するラテン方格である．そして，各方格の数字がそれぞれ Group 番号であるとして，図 8 の接続表を生成する．この接続表から得られる結線は図 9 のようになり，B1 から B2〜B4，B6〜B8，B10〜B12 へは近接接続となる．この構成では，どの 1 段目のスイッチでも近接接続するスイッチ数は自身を含めて 13 となり最大となる．

4.2 ルーティング方式

メッセージ長が短い場合は帯域よりも遅延が性能に大きく影響を与える．逆に，メッセー

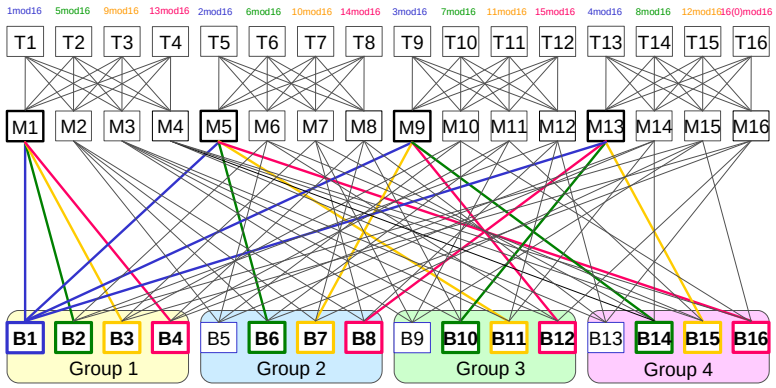

図 9　相互直交ラテン方格による接続構成

ジ長が長い場合は，遅延よりも帯域が性能に大きく影響を与える．したがって，メッセージ長に応じて，帯域を優先するか，遅延を優先するかを選択し，適切な経路に切り替える必要がある．帯域を優先する場合は，経路競合を避けるために，従来通りの経路を用いる．遅延を優先する場合は，ホップ数が少ない経路を選択する．

例えば，図 3 において，ノード 1 からノード 13 への経路を考える．帯域を優先する場合，ノード 13 への経路はスイッチ番号「13」が $1mod3$，$4mod9$ であるので，B1 → M1 → T2 → M2 → B5 を選択する．逆に遅延を優先する場合は，最短経路である B1 → M4 → B5 選択する．

InfiniBand では，同一の転送先アドレスに対する経路は 1 つしか設定できないので，1 つのノードにアドレスを 2 つ割り当てて，帯域優先用の経路と遅延優先用の経路をそれぞれ設定する．経路の切り替えは，転送先アドレスを選択することで実現する．

5. 評　　価

5.1 評 価 項 目

評価では，標準的な Fat Tree(標準 FT)，4.1.1 節で説明した基本方式 (提案方式 (基本))，4.1.2 節で説明したラテン方格方式 (提案方式 (ラテン方格)) において，n-ary 3-tree における近接接続スイッチ数，近接接続可能なスイッチの割合，平均ホップ数を算出し，比較する．提案方式 (ラテン方格) は，次数 (n) が素数の冪乗の場合に適用可能であるが，次数が素数であるとき，提案方式 (基本) と提案方式 (ラテン方格) は同じ結線が得られる．そこで評価

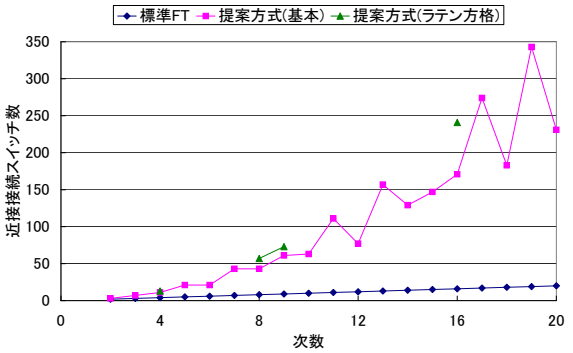

図 10　近接接続スイッチ数

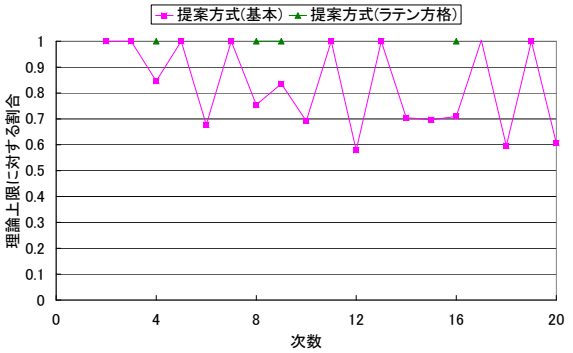

図 11　理論上限との比率

では，提案方式 (ラテン方格) については，次数が $p^k(k \geq 2, p$ は素数) の場合のみを評価対象とした．

n-ary 3-tree では，1 段目，2 段目にはそれぞれ $2n$ ポートスイッチを，3 段目には n ポートスイッチを用い，最大構成の場合について評価する．なお，$2 \leq n \leq 20$ の範囲について評価を行った．

5.2 近接接続スイッチ数

図 10 に n-ary 3-tree における近接接続スイッチ数を示す．標準 FT では，次数に比例して近接接続スイッチ数が増加するのに対し，提案方式 (基本) では，次数が素数である場合

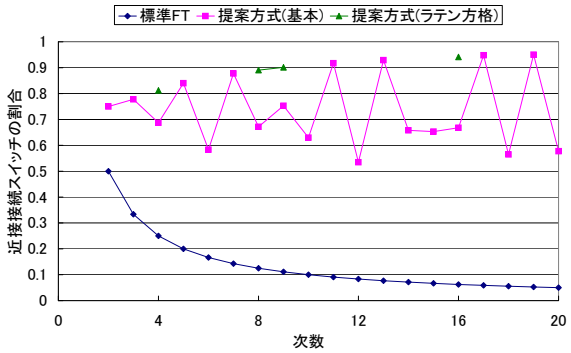

図 12　近接接続スイッチの割合

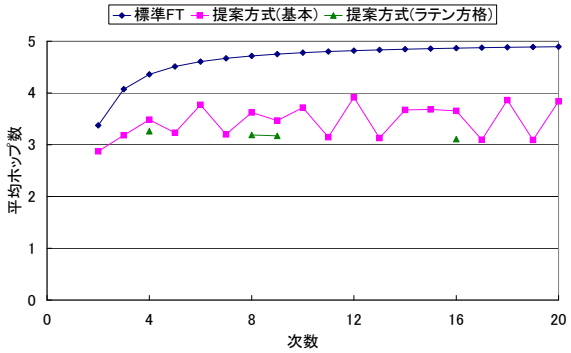

図 13　平均ホップ数

に関しては，理論上の近接接続スイッチ数の上限である $n(n-1)+1$ となり，大幅に増やすことができる．この結果，19-ary 3-tree 構成において，3 ホップ以内で到達可能なノードを 18.1 倍に増加させた．次数が素数でない場合に関しても，標準 FT と比較すると大幅に改善される．また，提案方式 (ラテン方格) を用いると，次数が素数の冪乗である場合でも，理論上限となり，改善できる．

近接接続スイッチ数の理論上限を 1 とした場合の割合を図 11 に示す．次数 (n) によって，上限に対する割合はまちまちであるが，最悪の場合 $(n=12)$ でも，上限値の 57%であるため，概ね $O(n^2)$ に従うと考えられる．

5.3　近接接続可能なスイッチの割合

図 12 に n-ary 3-tree における近接接続可能なスイッチの割合を示す．近接接続可能なスイッチの割合は，近接接続スイッチ数を，全ての 1 段目のスイッチの数で割って求める．標準 FT では次数の増加に伴い近接接続可能なスイッチの割合が大幅に低下する．これに対し，提案方式 (基本) では，少なくとも次数が素数となる場合は，次数の増加に伴い，近接接続可能なスイッチの割合を向上させている．また，提案方式 (ラテン方格) によって，さらに次数が素数の冪乗の場合も改善している．次数が素数および素数の冪乗でない場合についても，標準 FT と比較すると，大幅に改善できていることが分かる．

5.4　平均ホップ数

図 13 に n-ary 3-tree における平均ホップ数を示す．ここでホップ数とは，ノード-ノード間で到達するために経由するスイッチ数である．標準 FT では，次数が増加すると，ほとんどのスイッチに対して 3 段目のスイッチを経由して到達することになるので，平均ホップ数は 5 に近付く．一方，提案方式では，近接接続可能なスイッチが多く，平均ホップ数は改善される．最善の場合は次数が 19 の場合であり，平均ホップ数を 36.7%削減できることが分かった．

6. 関 連 研 究

文献[3] では，Fat Tree においてホップ数を削減する接続方式について議論されている．文献[3] では，図 14 のような結線を提案しており，本稿と同様にできるだけ最短ホップ数で到達可能となる構成をラテン方格を用いて実現している．一方で，経路競合については考慮されていない．例えば，B1 の配下にあるノード 1〜3 から B2 の配下にあるノード 4〜6 へ通信すると，経由できるスイッチが M1 しかないため，経路競合が発生する．

文献[5] では，図 15 のような文献[3] で提案されたラテン方格 Fat Tree と標準的な Fat Tree を組み合わせた構成を提案している．しかし，文献[3] と同様の問題があり，全対全通信時に，経路競合が発生する．

最短ホップ数で到達可能なノード数をできるだけ増やしつつ，全対全通信における経路競合まで考慮したネットワーク構成に関する研究事例は，我々の知る限りない．

7. お わ り に

本稿では，全対全通信時の経路競合を避けつつ平均ホップ数を削減する Fat Tree の結線方法を提案した．提案する結線では，標準的な Fat Tree と同様に全対全通信において広く

Vol.2011-HPC-130 No.48
2011/7/28

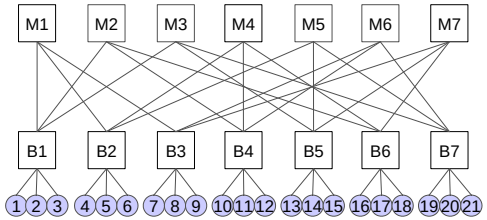

図 14 文献[3] の提案方式

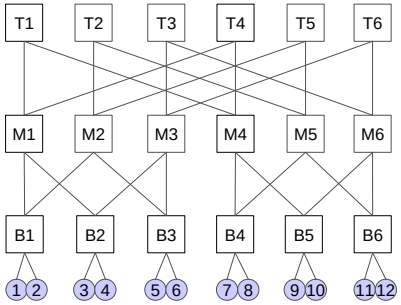

図 15 文献[5] の提案方式

用いられるシフト通信パターンにおいて，各通信フェーズでの経路競合を避けることができる．その上で，標準的な Fat Tree と比較して少ないホップ数で到達可能なノード数を増やすことができる．評価の結果，標準的な Fat Tree と比較して，19-ary 3-tree 構成において，3 ホップ以内で到達可能なノードを 18.1 倍に増やせることを確認した．また，平均ホップ数を約 36.7%削減できることを確認した．特に次数が素数あるいは素数の冪乗である場合，3 ホップ以内で到達可能なノードを理論上限とすることができ，効果が大きいことを示した．また，次数がそれ以外の数の場合でも，標準的な Fat Tree と比較して 3 ホップ以内で到達可能なノードを大幅に増やせることを示した．これにより，平均ホップ数の削減による遅延削減の見込みを得た．

今後の課題として，次数が素数あるいは素数の冪上でない場合の改善方法の検討や，実環境への適用と評価がある．

参 考 文 献

1) InfiniBand Architecture Specification Release 1.2, InfiniBand Trade Association, http://www.infinibandta.org.
2) E. Zahavi, G. Johnson, D.J. Kerbyson, and M. Lang : “ Optimized Infiniband Fat-tree routing for shift all-to-all communication patterns,” In Proceedings of the International Supercomputing Conference 2007 (ISC07), 2007.
3) M. Valerio, L. E. Moser and P. M. Mellliar-Smith: “Using Fat-Trees to Maximize the Number of Processors in a Massivly Parallel Computer,” In Proceedings of the 1993 International Conference on Parallel and Distributed Systems, pp. 128-134.
4) William J. Gilibert, W. Keith Nicholson: “ Modern Algebra with Applications Second Edition,” Willey-Interscience, 2003.
5) 岩本 邦生，高橋 義造 : Fat-Tree 型相互結合網の設計，情報処理学会全国大会講演論文集 第 49 回平成 6 年後期 (6)，59-60，(1994)．