通信量バランスの良いデッドロック回避ルーティング手法の提案とクラスタネットワークにおける評価

中島耕太[†] 成瀬彰[†]
住元真司[†] 久門耕一[†]

本論文では，通信量バランスの良いデッドロック回避ルーティング手法であるターン追加法を提案する．本手法は，ターン禁止法の一種であり，スイッチの入力ポートと出力ポートの組であるターンの使用を部分的に禁止してデッドロックを回避する手法である．全ターンを禁止した状態を初期状態とし，通信量の大きいターンから順に許可判定を行い，そのターンを使用してもデッドロックが生じない場合は当該ターンを許可する．ターン単位で禁止/許可を判別するため，既存手法と比較するとネットワークの一部分や一部のスイッチに禁止ターンが偏りやすくなる傾向は低くなる．このため，通信量バランスの良いルーティングを実現しやすい．

本手法をランダムネットワークと Fat Tree ベースのクラスタネットワークに適用し，評価した．ランダムネットワークでは，Up*/Down*法と比較してスループットを最大 2.05 倍改善し，TP 法と比較してほぼ同性能であることを確認した．また，クラスタネットワークでは，8192 ノード構成の Fat Tree を 2 つ接続した場合，Fat Tree を接続する経路において，TP 法と比較して，スループットを最大 4.77 倍改善できることを確認した．

Proposal of a Deadlock Avoidance Routing Method to Improve Traffic Balance and Its Evaluation in Cluster Networks

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

This paper describes a proposal of turn addition method that is a deadlock avoidance routing method to improve traffic balance. A turn is defined as a pair of input-output ports in a switch. The turn addition method avoids deadlock by prohibited turns which break turn loops. In initial state, all turns in a network are prohibited, then each turn are distinguished to allowed or prohibited from heaviest traffic order. In the distinction, if the turn does not cause deadlock by previous allowed turns, it marked to allowed. In this method, prohibited turns tend to be not intensively deployed a few switches, then network traffic balance are improved.

We apply the turn addition method to routing in random network and in cluster system network based on fat trees. In the evaluation result in random network, the turn addition method generate better traffic balance routing than Up*/down* method, and it can achieve 2.05 times higher throughput. The routing performance by turn addition method is similar to one by TP method. In cluster system network which consists of two connected 8192 node fat trees, the performance of the link which connect between two fat trees is improved by 4.77 times higher.

1. はじめに

近年，PC サーバに代表されるコモディティ部品から構成される計算機を高速ネットワークで接続したクラスタシステムが広く用いられている．特に HPC 分野において，クラスタシステムは現在の主流の構成である．

クラスタシステムでは，サーバ間接続用ネットワークとして InfiniBand[1] や Myrinet といった高速インタコネクトが採用されている．特に InfiniBand は，2010 年 11 月時点で Top500 中 213 システムで採用されており，大規模クラスタにおいてはデファクトスタンダードとなっている．また接続トポロジーは Fat Tree が広く採用されている．Fat Tree は，メッシュやトーラスといったトポロジーと比較すると高コストとなりやすいものの，高帯域かつ比較的均質のとれた構成であり，性能と管理の容易性において優れている．

† （株）富士通研究所
Fujitsu Laboratories Ltd.

ⓒ 2011 Information Processing Society of Japan

このため，今後もクラスタネットワークにおいて広く採用されると考えられる．

一方，最近では，多様なアプリケーションに対応するため，複数の種類の計算サーバ群を相互に接続する事例[2),3)] も登場している．また，目的別にサーバ群を複数にわけ，複数 Fat Tree を接続したこの構成は，全ての計算サーバを 1 つの Fat Tree で接続するよりネットワーク構築コストが削減できるため，今後利用が進むと考えられる．

標準的な Fat Tree 構成であれば，標準的なルーティング手法を用いれば，そのトポロジーの性質上，通信量バランスの良いデッドロックを回避する高性能なネットワークが実現できる．しかし，複数 Fat Tree を接続した構成は，Fat Tree の標準的なルーティング手法は適用できない．トポロジーに依存しないデッドロック回避ルーティング手法を用いる必要がある．

トポロジーに依存しないデッドロック回避ルーティング手法として，Up*/Down*法と Turn-Prohibition 法 (TP 法) が知られている．これらの既存手法を複数 Fat Tree を接続した構成に適用すると，多くの構成で通信量バランスは悪くなる．一部の限られた構成において，既存手法でも通信量バランスが良い場合もあるが，この限られた構成は，必ずしもコスト，物理的制約，保守性において優れているとは限らない．したがって，より幅広い構成においても，十分に通信量バランスが良くなるルーティング手法が必要である．

そこで，本論文では，デッドロック回避ルーティング手法であるターン追加法を提案する．本手法は，ターン禁止法の一種であり，スイッチの入力ポートと出力ポートの組であるターンの使用を部分的に禁止してデッドロックを回避する手法である．全ターンを禁止した状態を初期状態とし，通信量の大きいターンから順に許可判定を行い，そのターンを使用してもデッドロックが生じない場合は当該ターンを許可する．ターン単位で禁止/許可を判別するため，既存手法と比較するとネットワークの一部分や一部のスイッチに禁止ターンが偏りやすくなる傾向は低くなる．このため，既存手法では通信量バランスが悪くなるネットワークでも，通信量バランスの良いルーティングを実現しやすい．

本手法をランダムネットワークと Fat Tree をベースとしたクラスタネットワークに適用し，評価した．ランダムネットワークによる評価では，Up*/Down* 法と比較してスループットを最大 2.05 倍改善し，TP 法と比較してほぼ同程度のスループットを実現できることを確認した．また，クラスタネットワークでの評価では，2 つの 8192 ノード構成の Fat Tree を接続した場合において，Fat Tree を接続する経路において，TP 法と比較して，スループットを 4.77 倍改善できることを確認した．

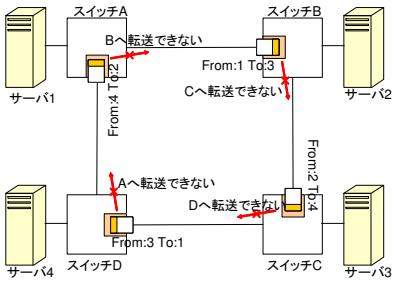

図 1　デッドロックの発生

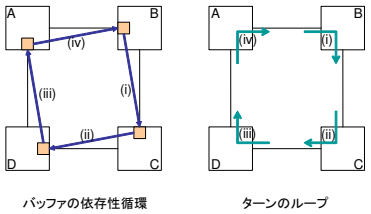

図 2　バッファの依存性循環とターンのループ

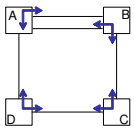

図 3　チャネル追加法

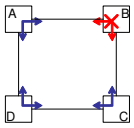

図 4　ターン禁止法

2. 既存のデッドロック回避手法

2.1 デッドロックの発生と回避条件

ループを含むトポロジーを持つネットワークでは，デッドロックが発生する可能性があることが広く知られている．例えば，**図 1** のようなリング状のネットワークにおいて，各サーバがそれぞれ対角線上のサーバへ同時に時計回りの経路で送信する場合を考える．この場合，まず，時計回り方向に隣接するスイッチの入力バッファにパケットが到着する．入力バッファの容量が 1 パケット分だとすると，スイッチはさらに次のスイッチへ転送しようとするものの，次のスイッチの入力バッファには空きがなく，転送することができない．図 1 の状況では，パケットが破棄されない限り，全てのパケットの転送ができなくなる．これがデッドロックである．したがって，デッドロックが発生する可能性があるルーティングは回避しなければならない．

デッドロックを回避するためには，**図 2** に示すパケット転送における入力バッファから次の入力バッファへの依存性の循環をなくせばよい．スイッチにおける入力ポートから出力ポートへの転送をターンと定義すると，このバッファの依存性循環とターンのループは同値である．したがって，ターンのループを排除すればデッドロックは回避できる．

ⓒ 2011 Information Processing Society of Japan

2.2 デッドロック回避手法

デッドロックを回避するルーティング方式はこれまで広く研究されている．基本的には以下の 2 つの方式の組み合わせによりデッドロックを回避している．

- チャネル追加法
- ターン禁止法

チャネル追加法[4)~6)] では，**図 3** のようにチャネルを追加することにより，ターンのループを防ぐ．図 3 では，物理的にリンクを加えているが，実際には，入力バッファを複数持たせる仮想チャネルにより図 3 の構造を実現できる．ルーティングの自由度は高いが，入力バッファを複数持たせる必要があり，多くのハードウェア資源を必要とする．特にクラスタネットワークで広く使用されている InfiniBand において仮想チャネルを利用するためには，MPI 実装に代表される通信ミドルウェアが対応する必要があり，実際の適用は難しい．

ターン禁止法[7)~9)] では，**図 4** のようにルーティングに使用するターンの一部を禁止することでターンのループを防ぐ．ルーティングの自由度は低下するが，追加のハードウェア資源は不要である．そこで本稿では，チャネル追加法は使用せず，ターン禁止法によってのみデッドロック回避を行う．

2.3 ターン禁止法によるデッドロック回避

既存のターン禁止法のうち，トポロジーに依存しない手法として Up*/Down*法[7)] と Turn-Prohibition 法[8)](TP 法) がある．

2.3.1 Up*/Down*法

Up*/Down*法[7)] は最も広く使用されている手法の 1 つである．Up*/Down*法では，まず，ネットワーク上の 1 つのスイッチを頂点とし，各スイッチと頂点との距離を算出する．そして各リンクにおいて頂点へ近づく方向を Up 方向と定める．頂点との距離が同一である場合はノード番号が若い方向を Up 方向とする．そして，Down 方向の入力から Up 方向への出力となるターンを禁止することによりターンのループを回避している．この手法は，単純な操作で禁止ターンを算出できるため幅広く用いられている．しかし，頂点と反対側に禁止ターンが偏りやすいという欠点がある．

2.3.2 Turn-Prohibition 法 (TP 法)

Turn-Prohibition 法[8)](TP 法) では，まず，ネットワーク上の 1 つのスイッチを選択する．そして選択したスイッチが形成する全てのターンを禁止ターンに設定し，そのスイッチと接続されるリンクを取り除く．この操作を全てのスイッチが取り除かれるまで繰り返す．文献[8)] によると，Up*/Down*法よりも通信量分散の良いルーティングが得られる．しかし，初期に選択されるスイッチに禁止ターンが偏りやすいという欠点がある．

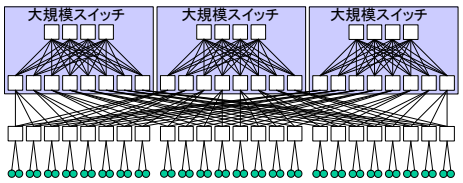

図 5 大規模スイッチによる見かけ上 2 段の Fat Tree 構成例

3. クラスタネットワークとルーティング

本章では，クラスタネットワークとして広く用いられている単一 Fat Tree 構成と，複数 Fat Tree を接続した構成について議論する．ここでは，複数 Fat Tree を接続した構成の例として，最も簡単な 2 つの Fat Tree を接続した構成について議論する．

3.1 単一 Fat Tree 構成

3.1.1 構　　成

近年大規模クラスタネットワークでは，InfiniBand による Fat Tree 構成が最も広く採用されている．InfiniBand において数千ノード規模の Fat Tree を構成するには，上段側スイッチとして Voltaire 社製 Grid Director 4700 に代表される数百ポート規模の大規模スイッチを採用する例が多い．この構成は見かけ上 2 段構成であるが，実際には，大規模スイッチの内部は複数の数十ポート規模の小規模スイッチによる 2 段の Fat Tree 構成である．したがって，**図 5** のように見かけ上 2 段の Fat Tree 構成も，実際には 3 段の構成である．

全て小規模スイッチを用いて 3 段構成を実現することも可能であるが，一般に，配線の煩雑性，物理的な配線の制約，保守性の観点から，大規模スイッチを用いた構成が主流である．

3.1.2 ルーティング

単一 Fat Tree 構成において Up*/Down*法，TP 法による禁止ターンを設定すると，最善の場合は**図 6** と**図 7** のようになる．図 6 で示す各スイッチの数字は，「0」で示すスイッチを頂点としたときのホップ数を示している．したがって，各リンクにおいて若い番号のスイッチへ向かう方向が Up 方向である．TP 法では図 7 に示すローマ数字 ((i)–(v)) の順にスイッチを選択し，取り除く．同じ番号の順序は任意である．

いずれの場合も Fat Tree においては，基幹となる通信は，(a) → (b) → (c) → (d) → (e) のような下段側から上段側への方向，下段側から下段側への方向，上段側から下段側への方向のターンが使用される．(d) → (e) → (f) のような上段側から上段側へのターンは使用されない．禁止ターンは全て上段側から上段側へのターンであるため，通信量バランスのよいルーティングが実現できる．

ⓒ 2011 Information Processing Society of Japan

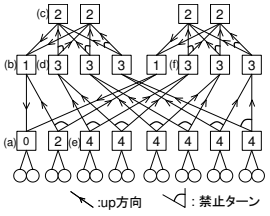

図 6 Up*/Down*法による禁止ターン設定

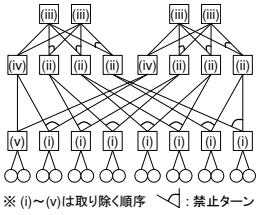

図 7 TP 法による禁止ターン設定

3.2 2 つの Fat Tree を接続した構成

3.2.1 構　　成

3 段構成の Fat Tree を接続する場合，接続箇所として，論理的には以下の 3 箇所が考えられる．

- 上段スイッチ (**図 8**，**図 9**，**図 10**)
- 中段スイッチ (**図 11**，**図 12**，**図 13**)
- 下段スイッチ (**図 14**，**図 15**)

しかし，実際にクラスタネットワークとして構築する場合，上段スイッチ及び下段スイッチを接続箇所とする構成には制限がある．

3.1.1 項で述べた理由により，多くのシステムでは，上段/中段スイッチとして大規模スイッチが用いられている．この場合，上段スイッチから外部へ出力されるポートが存在しないため，上段スイッチを接続箇所とする構成は採用できない．

また，下段スイッチの影響範囲は配下のエンドノードに閉じているため，比較的頻繁に電源切断可能であるという前提で保守される場合が多い．特に近年普及しているブレードサーバでは，ブレード筐体内に下段スイッチが格納されるため，この傾向が強い．さらに下段スイッチは，収容するエンドノード付近に配置され，比較的広範囲に設置される場合が多い．このため，下段スイッチを接続箇所とすると，接続距離が長くなり，管理しづらくなる．このように下段スイッチを接続する構成は，保守性や物理的制限の制約を受ける．

なお，中段スイッチを接続箇所とする構成については，保守性や物理的制約といった制限は少ない．

3.2.2 既存手法によるルーティング

2 つの 3 段構成の Fat Tree を接続する環境では，基本的には各 Fat Tree 内部のサーバ間が密に通信することを前提としている．したがって，第一に各 Fat Tree 内部の通信量バランスが良くなくてはならない．その上で，可能な限り Fat Tree 間接続の通信量バランスが良いことが求められる．これを考慮し，既存手法である Up*/Down*法と TP 法による各構成の禁止ターン設定について考察する．

(1) 上段スイッチで Fat Tree を接続した構成

Up*/Down*法の場合，図 8 の例のように禁止ターンが設定される．各 Fat Tree 内部において基幹となるターンが禁止されず，Fat Tree を接続する経路も禁止されないため，通信量バランスのよい経路が得られる．しかし，一部結線が障害等で使用できない場合は，図 9 のようになり，グレーのスイッチに隣接するリンクの Up 方向が変更される．このため，Fat Tree 内部において基幹となるターンが一部禁止され，通信量バランスが著しく低下する．

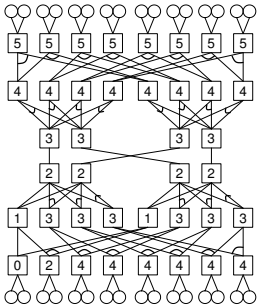

図 8 上段スイッチ接続 (Up*/Down*法)

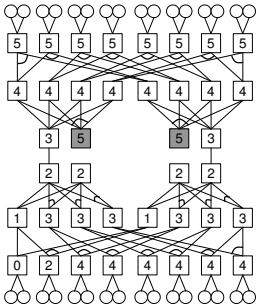

図 9 一部結線が欠けた構成

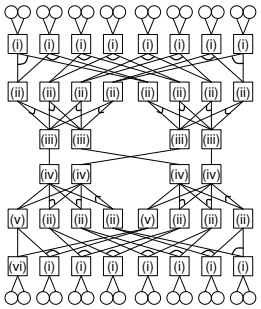

図 10 上段スイッチ接続 (TP 法)

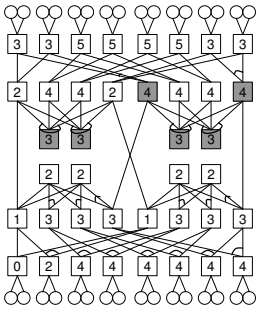

図 11 中段スイッチ接続 (Up*/Down*)

TP 法の場合，図 10 のような禁止ターンの設定となり，通信量バランスのよい経路が得られる．また，TP 法では，一部結線が故障した場合でも，通信量バランスのよい経路が得られる．

(2) 中段スイッチで Fat Tree を接続した構成

Up*/Down*法の場合，図 11 の例のように禁止ターンが設定される．グレーのスイッチにおいて Fat Tree 内部の基幹となるターンが一部禁止されてしまい，通信量バランスが著しく低下する．

TP 法の場合，図 12 のようになる．それぞれの Fat Tree については通信量バランスが良い．一方で，Fat Tree 間の接続経路に着目すると，グレーのスイッチにおいて，上段側から接続経路へのターンが禁止される．このため，太線で示す接続経路がほとんど使用できなくなるので，通信量バランスが悪くなる．

中段スイッチ同士を接続した構成でも，図 13 のような構成であれば，Up*/Down*法により通信量バランスのよいルーティングが実現できる．但し，コストの観点からこの構成は採用できない場合も多い．なお，この構成で TP 法を用いた場合でも，図 12 の例と同

ⓒ 2011 Information Processing Society of Japan

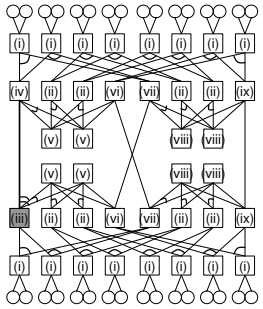

図 12 中段スイッチ接続 (TP)

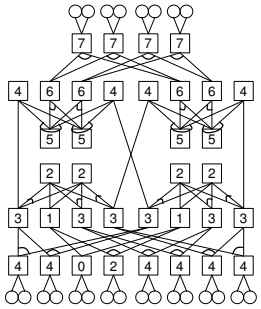

図 13 一部を除去

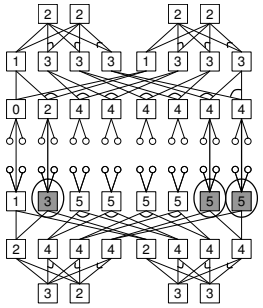

図 14 下段スイッチ接続 (Up*/Down*)

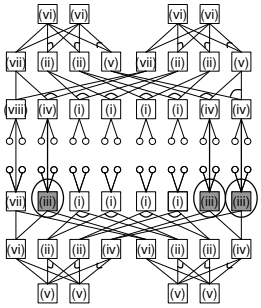

図 15 下段スイッチ接続 (TP)

様に Fat Tree 間の接続経路の通信量バランスは悪い．

(3) 下段スイッチで Fat Tree を接続した構成

Up*/Down*法，TP 法のいずれの場合も，図 14，図 15 のような禁止ターンの設定となり，各 Fat Tree の基幹となるターンが禁止されないため通信量バランスが良い．一方で，グレーで示したスイッチを経由するターンが全て禁止されるため，太線の経路をほとんど利用できず，Fat Tree 間の接続経路の通信量バランスは悪い．

3.2.3 2 つの Fat Tree を接続した構成の問題点

これまでの議論をまとめると**表 1** のようになる．物理的制約や保守性の観点を考えると，中段接続が望ましいと言える．但し，構築環境によって大きく基準が異なるため，環境によっては，必ずしも中段接続が最善とは限らない．ルーティングの観点で考えると，既存手法では，一部の構成では通信量バランスが良いが，多くの構成では通信量バランスが悪いため，選択できる構成が限定される．

このように，従来手法は構成に対して制限が厳しく，総合的に良いクラスタネットワークの設計を行うことは非常に難しい．より柔軟なネットワークを設計のためには，どのような構成でも，十分に通信量バランスが良くなるルーティング手法が必要である．

表 1 接続箇所の違いによる比較

接続箇所	上段接続	中段接続	下段接続
物理的制約や保守性	×	○	×
通信量バランス			
Up*/Down*法	×	×	△
TP 法	○	△	△

4. ターン追加法

これまでの議論から，以下を満たすデッドロック回避ルーティング方式の実現が課題である．

(1) 複雑なトポロジーのサポート
(2) 良い通信量バランス

そこで，この課題を解決するデッドロック回避ルーティング手法であるターン追加法を提案する．ターン追加法は，ターン禁止法の一種であり，一部のターンを禁止することでデッドロックを回避する．本章では，まず 4.1 節で，トポロジーに依存せず，通信量バランスの良いルーティングを得るための禁止ターンの決定方法について説明する．次に，4.2 節で，本手法によって禁止ターンを設定しても，到達経路の存在が保証されることを示す．

4.1 禁止ターン決定方法

4.1.1 概　　要

ターン追加法は，ネットワーク上のターンを順に追加し，当該ターンの追加によるループ発生の有無により，当該ターンの禁止/許可を判別する手法である．

スイッチ単位でなく，ターンを単位として取り扱うため，Up*/Down*法や TP 法と比較するとネットワークの一部分や一部のスイッチに禁止ターンが偏りやすくなる傾向は低くなる．

また，ネットワークの通信は均質ではない．例えば，2 つの Fat Tree を接続した構成では，Fat Tree 内部の通信量は大きく，Fat Tree 間の通信量は少ない．そこで，通信量の大きいターンから順番に許可/禁止の判別を行うことでできるだけルーティングに与える影響の少ないターンが禁止ターンとなるようにする．

このように禁止ターンを決定することで，複雑なトポロジーにも対応でき，通信量バランスが良く，構成に対する柔軟性の高いルーティングの実現を図る．以降，具体的なターン通信量の算出方法とターン追加による禁止/許可の判別方法について説明する．

4.1.2 ターン通信量の算出

ターン通信量は次のようにして算出する．

(i) サーバ間通信量の定義
(ii) 禁止ターンなし時の暫定ルーティング算出
(iii) ターン通信量の算出

まず，サーバ間の通信量をあらかじめ決定する．この通信量とは単位時間あたりの転送データ量であり，スループットと同等の概念である．例えば，並列計算を実施する Fat Tree 内部の計算サーバ間には大きな

ⓒ 2011 Information Processing Society of Japan

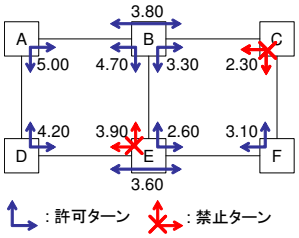

図 16 ターンの追加による禁止/許可の判別

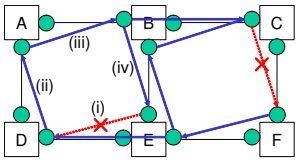

図 17 バッファ依存性ループの探索

通信量を設定し，Fat Tree 間の通信量は小さい値を設定する．

次に，ルーティング対象のネットワークにおいて，禁止ターンが存在しないと仮定した場合の暫定的な最善のルーティングを算出する．例えば，クラスタネットワークの場合，Fat Tree 内部は，転送先毎に経由するルートスイッチを順番に割り当てる標準的なルーティングを行う．Fat Tree 間は，各リンクの通信量が均等になるように通信ペア毎に 1 つずつ経路を決定する．この場合，最善な経路が複数ある場合はどれか 1 つを選択する．

そして，そのルーティングにおいて全ての通信ペアについて通過するターンを特定し，通過するターンに対し，設定した通信量を積算する．

4.1.3 ターンの追加による禁止/許可の判別

通信量が大きいターンから順に許可/禁止の判別を行う．但し，クラスタネットワークのような対称性のあるネットワークの場合，同一の通信量を持つターンが多く存在する．この場合，経路の分散を考慮し，同一スイッチに許可/禁止ターンが集中しないようにスイッチ毎に 1 つずつターンを選択する．さらにスイッチのなかでは特定の上位スイッチへ向かうリンクへの許可/禁止ターンが偏らないようにターンの反対側の端点となる上位スイッチを順番に切り替えて選択する．対称性のないネットワークの場合，同一の通信量を持つターンの順序はランダムにする．双方向のターンをペアとして取り扱うため，ソートの基準となるターン通信量は，双方向の合計値を用いる．

具体的なターンの判別処理を**図 18** に示す．`turns` で示す配列 (3 行目) には，ネットワーク上の全てのターンが格納されており，判別する順番にソートされている．当該ターンを追加し (7 行目)，当該ターンの追加によりループが形成される場合は当該ターンを禁止ターンとし，ループが形成されない場合は許可ターンと判別する．

この操作は，双方向のターンをペアとして取り扱う．すなわち，片方向のターンを禁止する場合は逆方向も禁止し (11 行目)，許可する場合は逆方向も許可する (15 行目)．なお，逆方向の許可/禁止をここで行うため，`turns` で示される以降の配列には，既に許可/禁止が決定されているものが含まれる．このため，4 行目では，既に決定済みのターンをスキップするようにしている．

ループ形成有無の判定 (`checkloop`) について説明する．連続する許可ターンのつながりは，図 2 に示したようにバッファの依存性の形式で表現できる．例えば**図 16** の場合，**図 17** に示すグラフで表現される☆．そこで，このグラフを追加したターンを起点に探索し，自身に戻る場合にはループ有り，自身に戻らない場合はループ無しと判定する．なお，図 18 では深さ優先探索を用いているが，幅優先探索でも構わない．

ネットワークのターン数を n とした場合の計算量について議論する．ループ形成有無の判定処理 (`checkloop`) は，最悪の場合でも，その時点で許可されている全てのターンの数に比例するため，計算量は $O(n)$ となる．この計算をターンの数だけ行うため判別処理 (`distinct`) の計算量は $O(n^2)$ である．

4.1.4 具体的な事例

図 16 のようなネットワークの事例について具体的に説明する．図 16 における双方向の矢印はそれぞれターンを示している．また，ターンに記されている数字は双方向合計のターン通信量である．

はじめに，最も通信量が大きいターン DAB を許可ターン群に追加する．ループは生じないので，許可ターンと判別する．同じ要領で通信量の順にターン ABE と EDA を追加する．いずれの追加でも，ループは生じないので許可ターンと判別する．次にターン BED を追加すると，既に許可ターンとなっているターン EDA，DAB，ABE とループを形成するため，禁止ターンと判別する．さらに，ターン ABC，DEF，EBC，CFE，FEB の順に追加する．いずれの追加でも，ループは生じないので許可ターンと判別する．そして，ターン BCF を追加すると，既に許可ターンとなっているターン FEB，EBC，BCF とループを形成するため，禁止ターンと判別する．

このよう，通信量が大きい順にターンを追加し，ループが生じるターンを禁止ターンに決定する．このように禁止ターンを決定することで，できるだけ通信量が少ないターンを禁止ターンに指定できる．

最終的に，得られた禁止ターンに基づき，デッドロック回避ルーティングを算出する．基本的には暫定ルーティング算出と同様の操作を行うが，禁止ターンを経由しないようにする．

☆ このグラフでは左回り方向のターンのみ記載している．

ⓒ 2011 Information Processing Society of Japan

```
 1: distinct() {
 2:     for(i=0;i<turnnum;i++){
 3:         t = turns[i];
 4:         if(decided(t)){
 5:             continue;
 6:         }
 7:         addturn(t);
 8:         if(checkloop(t) == LOOP_DETECT){
 9:             delturn(t);
10:             prohibit(t);
11:             prohibit(reverse(t));
12:         }else{
13:             addturn(reverse(t));
14:             allow(t);
15:             allow(reverse(t));
16:         }
17:     }
18: }
19:
20: addturn(t) {
21:     next = getnextturn(t);
22:     t->next[t->cnt++] = next;
23: }
24:
25: delturn(t) {
26:     t->next[--(t->cnt)] = NULL;
27: }
28:
29: checkloop(t) {
30:    push(t);
31:    while(cur = pop(t)){
32:        for(i=0;i<cur->cnt;i++){
33:            if(cur->next[i] == t){
34:                return LOOP_DETECT;
35:            }
36:            push(cur->next[i]);
37:        }
38:    }
39:    return OK;
40: }
```

図 18　禁止ターン判別処理の疑似コード

最終的に得られる禁止ターンや経路は，はじめに設定した通信量に大きく左右される．クラスタネットワークの場合，各ノードの通信特性が比較的明確であり，予測しやすいが，より一般的なネットワークや業務サーバに適用する場合は，通信量の見積もり予測が外れる場合もある．このような場合，非効率な経路となるため，必要に応じて，より実情に近い通信量を見積もり，再度禁止ターン箇所を決定しなおす必要がある．

4.2　到達性保証

4.2.1　概　　要

ターン追加法では，ターンが禁止ターンに分類される場合には，必ずそのターンを追加することによってループを形成する連続する許可ターンが存在する．双方向のターンをペアで取り扱っているため，禁止ターンによって到達できなくなった経路は必ず連続する許可ターンを利用して逆向きに迂回できる．以降，詳細について説明する．

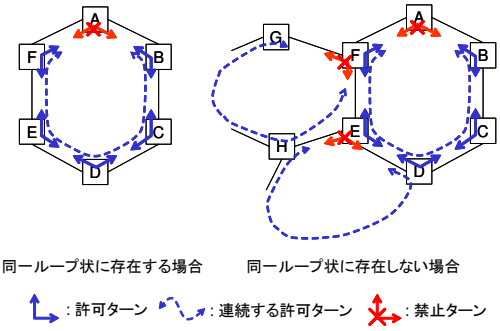

図 19　到達性保証

4.2.2　同一ループ上に存在するスイッチ間

図 19 左のような対象ループ状に存在するスイッチ間の到達保証性を考える．ターン ABC，BCD，CDE，DEF，EFA が許可ターンである場合，ターン FAB は禁止ターンとなる．スイッチ A と B の間に，反時計回りの連続する許可ターンが存在するため，いずれのスイッチ間も到達可能である．

4.2.3　同一ループ上に存在しないスイッチ間

次に図 19 右のような場合について考える．具体的にはループ外のスイッチ G からスイッチ A〜F への到達保証性を考える．ターン GFE が許可ターンであれば，明らかに G から A〜F へは到達可能である．

次にターン GFE が禁止ターンである場合を考える．ターン GFE が禁止ターンである場合は，ターン GFE を追加することでループを形成する連続する許可ターンが存在する必要がある．したがって，その経路を経由して G から E へは到達可能である．

仮に，ターン HED が許可ターンであれば，G から A〜F へは到達可能となる．ターン HED が禁止ターンである場合は，先ほどと同様の理由により，ターン HED を追加することでループを形成する連続する許可ターンが存在し，これにより，D へ到達可能である．

この議論から，途中に禁止ターンが存在する経路であっても，その禁止ターンに対応する連続する許可ターンを用いて G から A〜F へは到達可能であることがわかる．このように，ターン追加法では，到達性が保証される．

5.　評　　価

ターン追加法の性能を評価するため，ランダムネットワークと 2 つの Fat Tree を接続したクラスタネットワークによる評価を行った．仮定した通信量による算出シミュレーションにより評価する．ターン追加法 (add) の他，比較のため Up*/Down 法 (updown)，TP 法 (tp) について評価を行った．

ⓒ 2011 Information Processing Society of Japan

5.1 評価方法

評価指標として文献[8]と同様にボトルネック箇所のリンク通信量から算出するネットワークスループットを用いる．

ターン追加法の場合，4.1 節に示す手順でルーティングを算出する．Up*/Down*法の場合，各スイッチをそれぞれ頂点とした場合の禁止ターンを算出し，禁止ターンの通信量の合計値が最小となる頂点とした場合に基づきルーティングを算出する．TP 法の場合，各スイッチにおけるターン通信量の合計値を算出し，合計値が最も小さいスイッチから順に選択し，取り除く操作を行い禁止ターンを得る．これに基づきルーティングを算出する．

なお，ターン通信量は，ランダムネットワークにおいては，全てのサーバ間が均等に通信すると仮定して算出する．クラスタネットワークにおいては，各 Fat Tree 内のサーバ間の通信量を 1 とし，Fat Tree 間の通信量を 1/100 として算出する．

具体的には以下の手順で算出する．

(1) 各リンクの通信量の算出
(2) ネットワークスループットの算出

以降，詳細を説明する．

5.1.1 各リンクの通信量の算出

すべてのサーバ間において，得られたデッドロック回避ルーティングに基づき経由するリンクを特定し，経由するリンクにサーバ間の通信量を積算する．ランダムネットワークにおいては，全てのサーバ間が他のサーバに通信量 1.00 で均等に通信すると仮定する．クラスタネットワークにおいては，Fat Tree 内部と Fat Tree 間に分けて評価する．Fat Tree 内部については，Fat Tree 内における自身以外のサーバに対し通信量 1.00 で均等に通信すると仮定する．Fat Tree 間については，Fat Tree 間を接続経路の本数を p，片側 Fat Tree のサーバ数を n とした場合，各サーバが相手側のサーバへ通信量 p/n で均等に通信すると仮定する．

5.1.2 ネットワークスループットの算出

ネットワークスループットは，ボトルネック箇所であるリンク通信量に基づいて算出する．各リンクの通信量が最大となる箇所をボトルネック箇所とし，ボトルネック箇所のリンク容量を通信量で割る．

具体的な算出方法を以下に示す．5.1.1 項で述べたように各サーバが 1.00 の通信量を送出すると仮定する．各リンクの通信容量が 1.00 であり，ボトルネック箇所の通信量が M である場合，実際に各サーバが送出できる通信量は $1.00/M$ である．そこで，この値をネットワークスループットとする．クラスタネットワークにおける評価では，Fat Tree 内部と Fat Tree 間の各評価において，完全に均等に通信量分散した場合が最大値となり，ネットワークスループットは 1.00 となる．

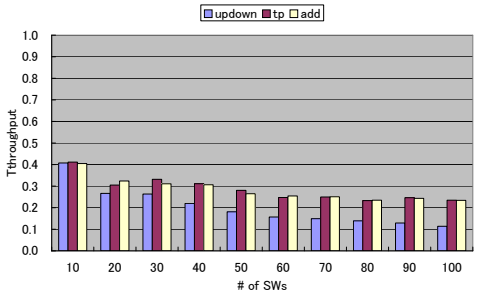

図 20 ランダムネットワークのスループット

5.2 ランダムネットワークによる評価

スイッチ数を 10 から 100 まで 10 個刻みで変化させた各場合においてそれぞれ 10 種類のランダムネットワークにおける評価を行った．各スイッチにおけるスイッチ-スイッチ間接続用ポート数を 10 とし，スイッチ-スイッチ間はランダムに 2 つのスイッチのポートを選択し接続することで，ランダムネットワークを生成する．また，各スイッチにはそれぞれ 10 台サーバを接続する．

各ランダムネットワークにおいてネットワークスループットを算出した．各スイッチ数における 10 種類のネットワークの平均を**図 20** に示す．Up*/Down* 法と比較すると，スイッチ数が 20 以上の場合において高性能であり，ネットワーク規模が大きいほどその差は大きくなる．スイッチ数 100 の場合において，スループットは平均 2.08 倍に改善される．また，TP 法と比較すると，全体としてほぼ同程度の性能であることがわかる．

5.3 クラスタネットワークによる評価

2 つの Fat Tree を接続した構成における評価を行った．接続箇所の違いによる評価とネットワーク規模の違いによる評価を実施する．

なお，単体の Fat Tree 構成については，ターン追加法，Up*/Down*法，TP 法のいずれにおいても同様の禁止ターンの設定となり，通信量バランスは一致するため，評価の対象としない．

5.3.1 接続箇所による違い

接続箇所を上段スイッチ (図 8)，中段スイッチ (図 11)，下段スイッチ (図 14) とした場合の 3 段 Fat Tree を 2 つ接続したクラスタネットワークにおいて，Fat Tree 内部と Fat Tree 間のネットワークスループットを評価した．

Fat Tree 内部及び Fat Tree 間接続のネットワークスループットの結果を**図 21** に示す．Fat Tree 内部については，ターン追加法と TP 法では，いずれの接続でも，基幹となるターンが制限されないため，FBB(Full Bisectional Bandwidth) 帯域を実現できる．一方，Up*/Down*法では，上段/中段スイッチを接続する場合，基幹となるターンが一部禁止されるた

ⓒ 2011 Information Processing Society of Japan

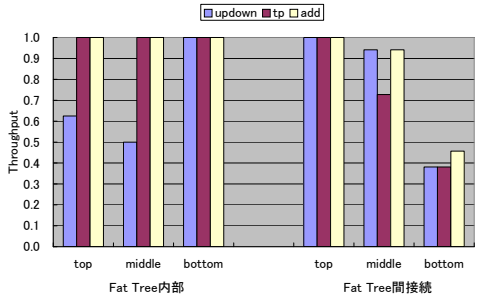

図 21 接続箇所の違いによるスループットの比較

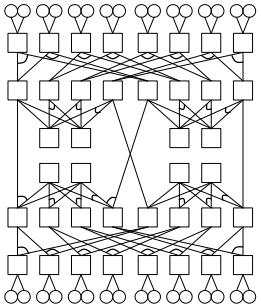

図 22 中段スイッチ接続 (ターン追加法)

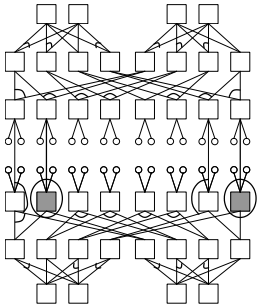

図 23 下段スイッチ接続 (ターン追加法)

め，性能が悪化する．したがって，上段/中段スイッチを接続する構成では，Up*/Down*法の適用は困難である．

Fat Tree 間接続については，上段スイッチを接続する場合，どの手法でも Fat Tree 間の通信経路に禁止ターンが存在しないため，高性能が得られる．また，中段スイッチを接続する場合は，ターン追加法では，**図 22** のような禁止ターンが得られる．接続経路の禁止ターンが複数のスイッチに分散するため，通信量を分散させることができ，高性能を実現できる．下段スイッチを接続する場合は，**図 23** のような禁止ターンが得られる．接続経路の禁止ターンの一部が複数のスイッチに分散するため，比較的高い性能を実現できる．

この結果から，既存手法は，構成によっては通信量バランスが悪くなる場合があるのに対し，ターン追加法は，いずれの構成においても通信量バランスが良く，クラスタネットワークの結線パターンに対して柔軟に適用できることが分かる．

なお，下段スイッチを接続する場合は，どの手法においても，Fat Tree 間接続経路の性能は低い．これは，当該トポロジーでは，いかなる手法でも，接続経路と Fat Tree 間の経路で経由するターンが全て禁止されるスイッチが生じてしまうためである．

5.3.2 ネットワーク規模による違い

ネットワーク規模の違いによる評価を行った．ここでは，配線コスト，物理的制約，保守性といった制約が少ない中段スイッチを接続した構成について評価する．

表 2 Fat Tree 間接続リンクへの禁止ターン割合

ネットワーク規模	禁止ターンの割合 (%)	Fat Tree 間接続リンク数	ノード数
16	6.25	4	32
128	15.23	16	256
1024	19.78	64	2048
8192	23.43	256	8192

16 ノード，128 ノード，1024 ノード，8192 ノード構成の 3 段 Fat Tree を 2 つ接続したクラスタネットワークにおいて，Fat Tree 内部と Fat Tree 間のネットワークスループットを評価した．

Fat Tree 内部及び Fat Tree 間接続のスループットの結果を**図 24** に示す．Fat Tree 内部については，ターン追加法と TP 法では，基幹となるターンが制限されないため，FBB 帯域を実現できる．一方，Up*/Down*法では，基幹となるターンが制限されるため，規模の増加に伴い，劇的に性能が悪化する．

Fat Tree 間接続については，Up*/Down*法では，Fat Tree 間の通信経路に禁止ターンが存在しないため，高いスループットを実現できる．

ターン追加法では，16 ノード構成と比較して 128 ノード構成では性能が低下し，1024 ノード，8192 ノード構成で性能が向上する．これは，**表 2** に示す禁止ターンの割合，Fat Tree 間接続リンク数，ノード数が関連する．接続リンクとの間に形成される禁止ターンの割合は，ネットワーク規模に従い増加する．特にこの割合は 16 ノード構成と比較して 128 ノード構成では 2.43 倍と高い割合で増加する．これにより，128 ノード構成では性能が低下すると考えられる．一方で，ネットワーク規模に従い Fat Tree 間接続リンク数とノード数は，規模の増加に伴い，4 倍ずつ増加する．接続リンク数が多いと経路の選択肢が増加するため，スループットを向上させる．また，転送先ノード数が増加するため，より細かい粒度で経路を振り分けることができる．これに対して，禁止ターンの割合は増加するものの，128 ノード構成と 1024 ノード構成，1024 ノード構成と 8192 ノード構成を比較すると増加の割合はそれぞれ 1.30 倍と 1.23 倍の増加であり，増加の割合は低い．これらの要因により，1024 ノード，8192 ノード構成では，性能が向上したと考えられる．このように，ターン追加法では，禁止ターンによる制限を受けるもの，各経路に通信量を分散させることができる．

一方，TP 法では，禁止ターンが一部のスイッチに集中する．特に図 12 に示すような上段スイッチとのなすターンがすべて禁止される箇所が生じる．このため，ほとんど使用できない経路が生じ，スループットは規模の増加に伴い悪化する．

この結果，ターン追加法では，Fat Tree 内部におい

ⓒ 2011 Information Processing Society of Japan

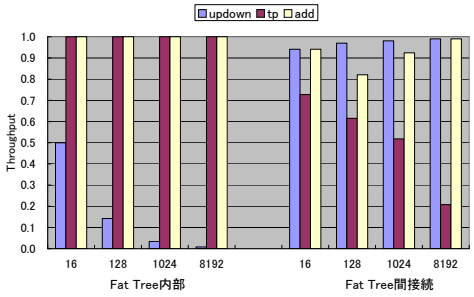

図 24 規模の違いによるスループットの比較

ても Fat Tree 間の経路においても通信量分散の良い経路が得られることがわかった．特に大規模構成では，効果が大きく，8192 ノード構成において，TP 法と比較して，Fat Tree 間接続のスループットを最大 4.77 倍改善できることがわかった．

6. 関 連 研 究

デッドロック回避ルーティング手法は広く研究されている．2 次元 Turn モデル[10] は，Up*/Down*法における Up/Down 方向を 2 次元に拡張したターン禁止法の一種である．本手法は，トポロジーに依存せず，双方向のターンを別々に取り扱うことができるため，メッシュやトーラスといったネットワークへも適用可能である．しかし，本論文で取り扱うような Fat Tree をベースとしたクラスタネットワークでは，Up*/Down* 法と同様の禁止ターンの設定となるため，効果は低い．

他にも，トポロジーを限定したターン禁止法[9] やチャネル追加法[4)~6)] の研究は広く行われているが，Fat Tree をベースとした構成に適用可能であり，仮想チャネルを使用しない手法は，我々が知る限りない．

7. お わ り に

本稿では，通信量バランスの良いデッドロック回避ルーティング手法であるターン追加法を提案した．

ランダムネットワークによる評価では，Up*/Down* 法と比較してスループットを最大 2.05 倍改善し，TP 法と比較してほぼ同程度のスループットを実現できることを確認した．また，2 つの Fat Tree を接続したクラスタネットワークでの評価では，いずれの接続方法においても，Fat Tree 内部及び Fat Tree 間において Up*/Down*法と TP 法と比較して最も通信量の分散するルーティングが得られることを確認した．さらに，8192 ノード構成において，TP 法と比較して，Fat Tree を接続する経路についてスループットを最大 4.77 倍改善できることを確認した．これにより，ターン追加法は様々なクラスタネットワークの結線パターンに対して柔軟に適用でき，既存手法より通信量分散の良いルーティングが得られる手法であることを示した．

本手法により，より柔軟なクラスタネットワークの設計が可能になる．すなわち，コスト，物理的制約，保守性といったルーティング以外の要素も重視した設計が可能になる．このようにクラスタネットワークの設計に柔軟性を与える点が，提案手法の利点である．

なお，本論文では，ターン追加法をクラスタネットワークにて評価したが，ターン追加法は，クラスタネットワークにのみ特化した手法ではない．一般的なネットワークにも適用可能であり，Fat Tree ベースのトポロジーであれば，高い効果が得られると考えられる．

今後の課題として，実環境に対するターン追加法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) 日本原子力研究開発機構, http://www.jaea.go.jp/
3) 「日本原子力研究開発機構様の新スーパーコンピュータシステムが稼動」, 富士通株式会社, プレスリリース, http://pr.fujitsu.com/jp/news/2010/03/1.html (2010).
4) J. Duato : “A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks,” IEEE Transaction on Parallel and Distributed Systems, Vol.4, No.12. (1993).
5) T. Skeie, et al.: “LASH-TOR: A Generic Transition-Oriented Routing Algorithm,” Proceedings of the Tenth International Parallel and Distributed Processing Symposium (IPDPS’04).
6) O. Lysne, et al.: “Layered Routing in Irregular Networks,” IEEE Transactions on Parallel and Distributed Systems, Vol.17, No.1 (2006).
7) Schroeder, M.D., et al.: “Autonet: A High-Speed, Self-Configuring Local Area Network Using Point-to-Point Links,” IEEE Journal on selected areas in communications. Vol.9, No.8 (1991).
8) D. Starobinski, et al.: “Application of Network Calculs to General Topologies Using Turn-Prohibition,” IEE/ACM Transactions on Networking, Vol.11, No.3 (2003).
9) C. J. Glass and L. L. NI: “The Turn Model for Adaptive Routing,” Journal of the Assocation for Computing Machinery, Vol.41, No.5, pp 874–902 (1994).
10) 上樂 明也, 鯉渕 道紘, 天野 英晴 : 2 次元 Turn モデルに基づくイレギュラーネットワーク向けルーティングアルゴリズムの設計と評価, 情報処理学会論文誌: コンピューティングシステム, Vol. 44, No. SIG 11 (ACS 3), pp 157–168 (2003).

ⓒ 2011 Information Processing Society of Japan