Vol.2017-HPC-160 No.6
2017/7/26

スイッチ数を削減するネットワークトポロジーの評価

清水 俊宏[1]　中島耕太[1]

概要：近年のクラスタシステムは 1000 ノードを超える大規模なものが広がっており，1000〜10000 台のサーバを接続して並列処理するシステムが登場している．PC クラスタを接続する際のトポロジーとしては Fat-Tree が広く採用されれているが，必要となるスイッチ台数が多いため，コストが高くなるとう問題点がある．この問題を解決するために，我々はすでに Fat-Tree に対してスイッチ数を削減する多層 Fullmesh やラテン方陣 Fat-Tree といったトポロジー構造の活用を提案し，そのトポロジー構造上での通信手法，特に最も高負荷な集合通信である All-to-all 通信の効率的な転送方式を提案している．本稿ではこれらの通信手法の実環境での評価・分析について論ずる．評価には NAS Parallel Benchmarks(NPB) を用い，NPB のうち処理中で All-to-all 通信が用いられるフーリエ変換 (FT) の処理性能を評価した．

キーワード：トポロジー，多層 full-mesh，All-to-all 通信，経路競合回避，NAS Parallel Benchmarks

1. はじめに

近年の科学技術計算や産業用とのシミュレーション計算は，大規模化・高速化が求められており，並列計算システムの大規模化・高速化が急務となっている．クラスタシステムの大規模化の大規模化のためには，接続可能なサーバ台数を増やして並列処理することが重要である．並列化した場合の性能は 1 台での性能に比べるとサーバ間通信分の処理時間がかかるため，この処理時間を削減する機構が必要となってくる．その一つのアプローチとしてサーバ同士をどのようにつなぐか，そのトポロジー構造に着目して性能向上を図る方法がある．同一のリンク上で同じ方向に複数の通信が通過した場合，経路競合が起こり通信に遅延が生じるため性能低下の原因となる．よって，経路競合の少ないトポロジー構造及び通信手段が必要となる．また，サーバ間の中継にはスイッチを用いるが，スイッチはサーバと比べて高価であるため，より少ないスイッチで多くのサーバを接続することが望ましい．

現在のクラスタシステムのトポロジー構造は Fat-Tree が広く採用されている．Fat-Tree は All-to-all 通信という非常に負荷の高い通信処理を競合なく実行できる．All-to-all 通信とは，各サーバがすべてのサーバにデータを送る通信である．Fat-Tree にはサーバ間の経路に冗長性があるため，高性能だる反面，段数を固定したときに接続可能なサーバ数が少ないという問題がある．したがって，より多くのサーバをつなぐには段数を高くしなくてはならず，1 スイッチあたりのサーバ台数が増えてしまうため効率が悪い．そこで，Fat-Tree よりもさらに少ないスイッチ数でより多くのサーバを接続できるトポロジー構造が求められている．

多層 full-mesh[1] やラテン方陣 Fat-Tree[2] は従来の Fat-Tree より多くのサーバを接続することを目的として提案されたトポロジー構造である．ラテン方陣 Fat-Tree は従来の Fat-Tree や多層 full-mesh に比べてスイッチのコスト面での効率は良いが，その分サーバ間の結合が疎になっているため，一般には高い性能を達成することが困難である．しかし，通信順序を工夫することにより，すでに我々はこれらのトポロジー上で All-to-all 通信を行うことが可能であることを示し，その手法を考案・提案・評価した．[3], [4], [1] その結果，我々の手法が InfiniBand の理論的な最適値をほぼ達成していることが確認された．

しかしながら，上記の結果はいずれも Intel MPI Benchmarks(IMB) を用いた All-to-all 通信単体のものであり，我々の手法を実アプリへの適用については試みられてこなかった．本稿では多層 fullmesh における All-to-all 通信を NAS Parallel のフーリエ変換処理 (FT) に適用し，その性能を評価する．FT では行列の転置で All-to-all 通信が行われているため，その部分での通信コストの改善で性能の向上が期待できる．そこで，本稿では FFT によるベンチマークを実際に動かしてその改善度合いがどの程度かを検証する．

[1] (株) 富士通研究所

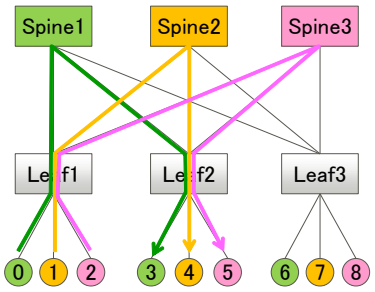

図 1 Fat-Tree によるトポロジー構造

図 2 シフト通信パターンの例

2. FatTree の課題

論文 [4] で述べられているように，多くのクラスタシステムでは InfiniBand[6] による図 1 のような Fat-Tree によるトポロジー構造が採用されている．

Fat-Tree には以下のような特徴がある．

(1) サーバ間の経路が多く冗長性があるため，通信負荷の高い All-to-all 通信でも高スループットで通信できる
(2) 通信を経由・転送するために多くのスイッチを必要とするためコストが高い

All-to-all 通信の性能を向上させることで，All-to-all 通信を内部で複数回実行する高速フーリエ変換 (FFT) などの処理を高速かつ並列に処理することができる [7]．

図 1 の上段のスイッチを spine スイッチ，中段のスイッチを leaf スイッチと呼ぶ．本論文で扱うトポロジー構造はスイッチのポートを同数ずつ上位向けと下位向けの 2 つの面に分け，片面のポート数を次数と呼ぶ．すなわち，次数はポート数の半分である．図 1 のスイッチの次数は 3 である．

なお，図 1 では各 spine スイッチの片面のみにトポロジーを構築しており，ポートが半分余っている．そこで，spine スイッチの反対側にも同様の構造を構築することで，サーバ台数を 2 倍にすることが可能である．

Fat-Tree においてはシフト通信パターンを採用することで，競合のない All-to-all 通信が可能である．シフト通信パターンとは，N をサーバ台数として，i 番目のフェーズでサーバ番号 S のサーバは $(S+i)\%N$ に送信する，という通信パターンである [8][9]．例えば，図 1 のように Fat-Tree のすべてのサーバを用いたシフト通信パターンでの All-to-all 通信は図 2 のようになる．図 2 において，第 3 フェーズでサーバ 0,1,2 はそれぞれサーバ 3,4,5 に送信することを表しており，図 1 の経路で送信することで，競合を回避できる．

また，leaf スイッチ単位でサーバ群を選択してジョブを割り当てることで，部分的なジョブにおける All-to-all 通信も可能であり，これによってジョブサイズに応じたコストに見合った運用も可能である．

しかしながら，前述の通り Fat-Tree は通信性能がよいものの大規模化には不向きである．Fat-Tree において，用いるスイッチの次数を変えずにより多くのサーバをつなぎたい場合，図 3 のように段数を増やすことで，対応しなくてはならない．図 3 は次数 3 の 2 段 Fat-Tree を 3 段に変更する例である．3.3 節で論じるように，Fat-Tree で段数を増やすと多くのサーバが接続できる反面，ネットワークコストやレイテンシも増大する．例えば，図 3 では 1 スイッチあたりのノード数が 2 から 1.2 に低下してしまう．近年ではクラスタシステムは大規模化の傾向にあるため，トポロジー構造を Fat-Tree で組んでしまうと多くのスイッチが必要となりコストがかかる．

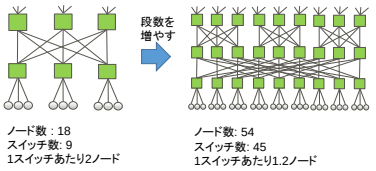

図 3 FatTree の問題点

3. 低コストで大規模化可能なトポロジー

前節における Fat-Tree の課題を解決するトポロジー構造としてラテン方陣 Fat-Tree[2] や多層 fullmesh[1] が提案されている．ラテン方陣 Fat-Tree は多層 fullmesh に比べて，さらに大規模なクラスタを構築するのに適しているが，All-to-all 通信が可能なノード数の制約が多く，とりわけ本稿で扱う 2 のべき乗の台数での All-to-all 通信に不向きであるため，本稿では多層 fullmesh を用いることを考える．

3.1 多層 fullmesh トポロジーの構造

多層 fullmesh トポロジーは図 4 にトポロジー [15] をベースに拡張したトポロジーである．各層は fullmesh 構造であり，各 fullmesh の leaf スイッチ間のリンクを切断して spine スイッチを挟んだ構造となっている．leaf および spine スイッチのポート数を $2d$ としたとき，同一スイッチで接続可能なノード数は $d^2(d+1)$ 台 $(O(d^3))$ となる．同じポート数のスイッチを用いて Fat-Tree を組んだ場合の接続可能台数は最大で $2d^2$ 台 $(O(d^2))$ となるため，多層 fullmesh

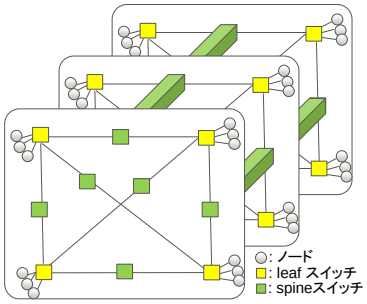

図 4 多層 fullmesh トポロジー構造

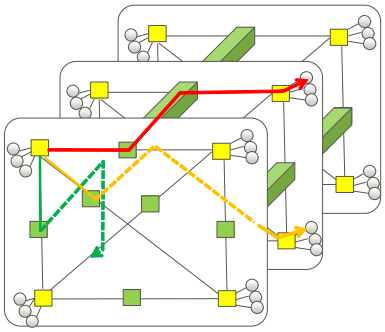

図 5 異なる位置の leaf への送信

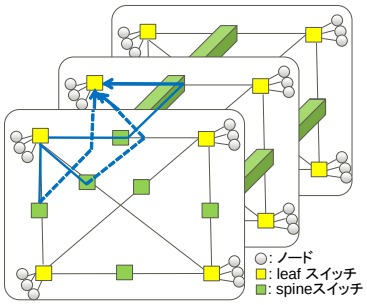

図 6 同一の位置の leaf への送信

は Fat-Tree に比べて桁違いに多くのサーバを接続することができるといえる．

3.2 多層 fullmesh トポロジーでの競合のない All-to-all 通信

fullmesh トポロジーでの競合のない All-to-all 通信についてはすでに知られている [15] ため，同一の層の間での通信は可能である．

異なる層への通信については，多層 fullmesh において spine スイッチを経由して別の層に移動できるという性質を利用する．例えば，第 1 層のノードから第 2 層のノードへの通信は図 5 に示す異なる位置の leaf への送信と，図 6 に示す同一の位置の leaf への送信で実現される．このとき，同時に第 2 層から第 3 層，第 3 層から第 4 層の通信を同様に行うことができ，これによって All-to-all 通信の 1 フェーズを構成する．

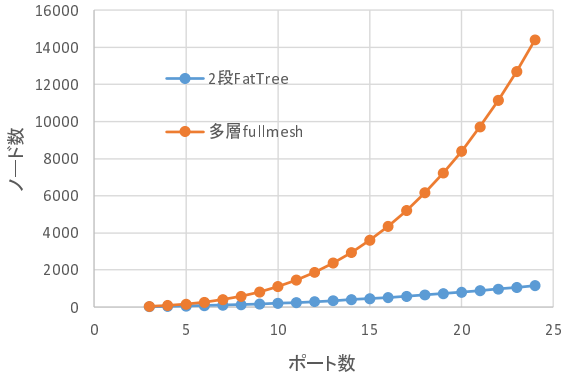

図 7 次数ごとの接続可能サーバ数

3.3 多層 fullmesh と従来の Fat-Tree におけるネットワークコストの比較

本節では，机上計算によって多層 fullmesh と従来の Fat-Tree のネットワークコスト (必要なスイッチ数とリンク数) を比較し，多層 fullmesh のコスト面での有用性を示す．比較には 2 段 Fat-Tree, 3 段 Fat-Tree, 多層 fullmesh を用いる．次数 d のスイッチを用いた場合，2 段 Fat-Tree の接続可能サーバ数は $2d^2$, 必要となるスイッチ数は $3d$ であり，3 段 Fat-Tree の場合は接続可能サーバ数は $2d^3$, 必要となるスイッチ数は $5d^2$，多層 fullmesh の場合は接続可能サーバ数は $d^2(d+1)$, 必要となるスイッチ数は $3/2d(d+1)$ となる．

次数ごとに接続可能なサーバ数をプロットした結果を図 7 に示す．多層 fullmesh では 2 段 Fat-Tree に比べて格段に多くのサーバが接続可能であることがわかる．

以上により，多層 fullmesh はレイテンシを維持しつつ，多くのサーバを少ないスイッチで接続可能なトポロジー構造である．本稿ではこの多層 fullmesh をアプリに応用して性能評価・分析する．

4. 評価

本章では NAS Parallel Benchmarks による実験を行い，評価および分析を行う．まず，NPB の概要および NPB を用いた評価方法説明し，次に評価結果の紹介と分析を行う．

4.1 NAS Parallel Benchmarks の概要

Nas Parallel Benchmarks(NPB)[11] は NASA Advanced Supercomputing division(NAS) が開発している並列計算のためのベンチマークセットであり，8 つのセットからなる．本稿ではこれらのうち，All-to-all 通信を用いているフーリエ変換 (FT) 用ベンチマークのみ評価を行うものとする．FT のベンチマークでは巨大行列の転置を行っており，その処理で All-to-all 通信が用いられている．FT のベンチマークにおいてはノード数は 2 のべき乗に固定されているため本稿の評価もノード数が 2 のべき乗のノード数が

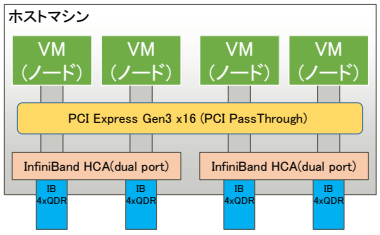

図 8 サーバの構成

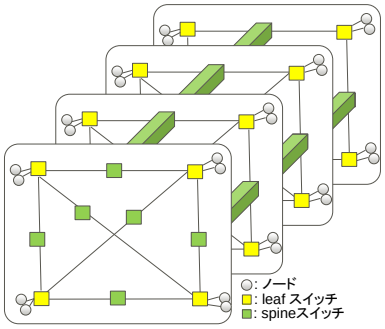

図 9 実験に用いるトポロジー構造

扱いやすくなるように構築した．

4.2 評価方法

図 9 に示すように実機を用いて多層 fullmesh のトポロジー構造を構築する．各層は 2 ノードのサーバが接続された 4 台の leaf スイッチからなる．これを 4 層構築し，それぞれの leaf スイッチ間を共通の spine スイッチで接続する．よって，サーバ数は合計 32 ノードとなる．実機サーバは 8 台あり，1 台につき 4 ノードの VM と 2 枚の Dual-Port の HCA(合計 4 ポート) を搭載している．図 8 に示すように，各実機サーバの 2 つの HCA にある 4 ポートは SR-IOV によってそれぞれ別々のポートとして認識させ，それをサーバ上の各 VM に PCI パススルーで接続した．これによって，各ノードは物理 HCA32 台に接続された場合と同様の動作をする．

なお，スイッチのポート利用数が leaf スイッチで 5 ポート，spine スイッチで 8 ポートとなり，leaf と spine で異なるスイッチ構成ではあるが，上記の手法で同様に競合のない All-to-all 通信が可能である．

NPB では FT のサイズ B, C, D について評価した．NPB の都合上用いるノード数は 2 のべき乗に限定した，8 台, 16 台, 32 台となる．それぞれ，図 9 の 1 層，2 層，4 層 (全体) を用いて実現する．

実験に用いるサーバ，ソフトウェア，InfiniBand の諸元を表 1，表 2，表 3 に示す．なお，InfiniBand のスイッチおよびケーブルは Mellanox 社製のものを用いた．[10]

表 1 サーバの諸元

サーバ	Fujitsu PRIMERGY RX200S8
CPU	Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz
コア数	24 コア
メモリ	1600MHz, 132GB

表 2 ソフトウェアの諸元

OS	Cent OS 7.2(VM), 6.4(物理マシン)
Linux Kernel	3.10.0-327.el7.x86_64
仮想マシンソフト	Kernel-based Virtual Machine(KVM)
MPI	OpenMPI 1.10.0
ベンチマーク	NAS Parallel Benchmarks (NPB)

表 3 InfiniBand の諸元

Spine スイッチ	SX6005 (FDR X4 12 ポート)
Leaf スイッチ	IS5022 (QDR X4 8 ポート)
HCA	Connect X3 (Dual Port FDR)

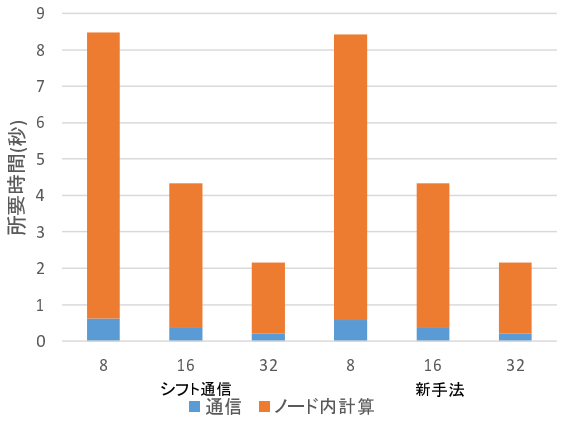

図 10 サイズ B の結果

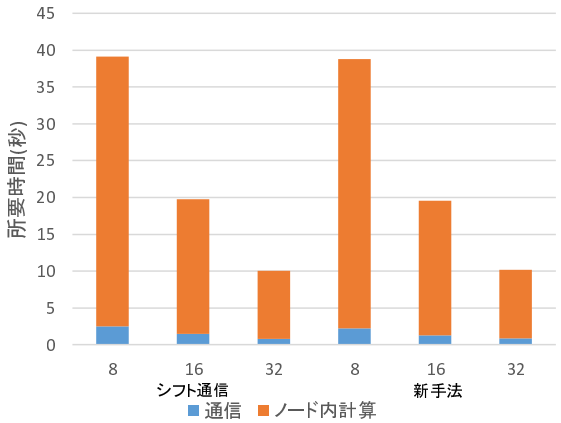

図 11 サイズ C の結果

4.3 結果と考察

サイズ B による計測結果を図 10 に，サイズ C による計測結果を図 11 に，サイズ D による計測結果を図 12 に示す．この結果，通信性能のみでは 11.4%, 全体性能では 1%改善したが効果が小さい．

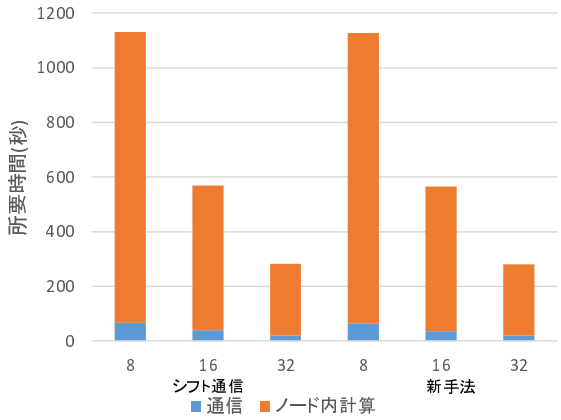

図 12 サイズ D の結果

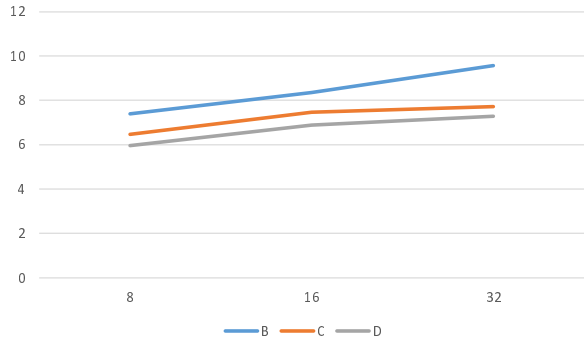

図 13 全体性能に占める通信の比率

そこで，全体性能に占める通信の比率 (%) を図 13 に示す．ノード数が増えるほど通信の比率が上がることとなり，大規模環境では通信の改善が効果的となることも期待される．また，今回評価に用いた NPB は 1 プロセスにつき 1 スレッドであるが，マルチスレッド化すれば，通信の性能がボトルネックとなる．

NPB においてスループット (MiB/s) を計算した結果，図 15 のようになった．この結果は理論限界 (事前に測定した PingPong 通信のスループット 3717Mib/s) よりも小さく，性能が低下していることがわかる．この原因を調査するための比較対象として Intel MPI Benchmarks(IMB) を用いて一定量 (2GiB) のデータを 8, 16, 32 台に分散させて All-to-all 通信を行いスループットを計測したところ，図 14 のようになった．この結果は [5] と同様に新手法の性能がシフト通信よりも良く，理論限界に近い最適値であることを示している．NPB での All-to-all 通信の傾向と IMB の結果は異なっており，アプリでの評価では性能低下が起こっているものと考えられる．

この低下について原因を探るため，1 層の fullmesh 上の 8 ノードについて以下の 3 構成でのスループットを計算した．

(1) 物理マシンで 8 ノード
(2) 8 台の物理マシン上にそれぞれ 1 台の VM を配置した 8 ノード
(3) 2 台の物理マシン上にそれぞれ 4 ノードずつ VM を配置した 8 ノード

結果を図 16 に示す．3. のように各実機上に 4VM を配置させた場合にのみ起こることがわかった．また，この VM 構成は [5] でも同様のものであるものの，このような現象は発生しなかったため，アプリ込みの評価を行ったことも影響していると考えられる．

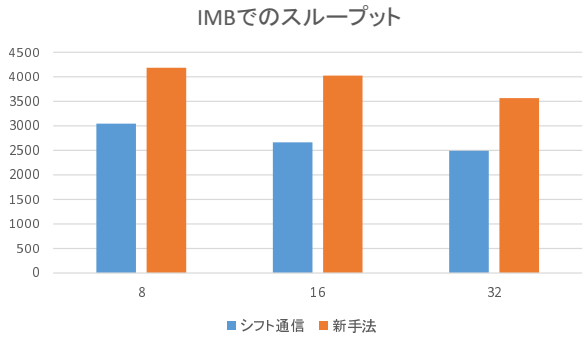

図 14 IMB での結果

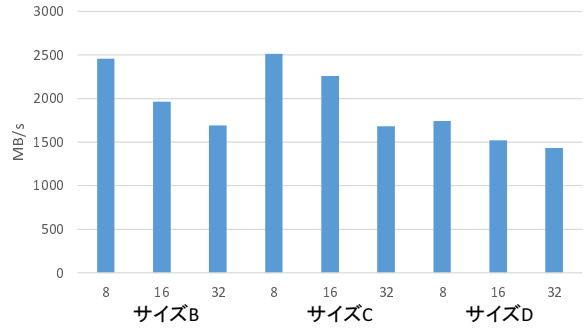

図 15 スループットの比較

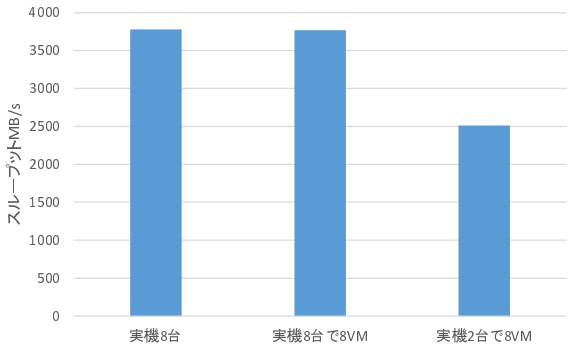

図 16 8 台でのスループット

5. 関連研究

論文 [16] は並列化した FFT の性能を Clos と Mesh ネットワークで比較したものである．1 次元，2 次元，3 次元の FFT に対して評価を行い，Clos ネットワークは Mesh

ネットワークに比べて最大で 50%の性能を達成した．しかしながら，All-to-all 通信に着目した改良ではなく，また対象とするトポロジーも我々が用いている多層 fullmesh ではない．

また，論文 [17] は All-to-all 通信の頻度を減らした FFT のアルゴリズムを提案している．この手法は転置の際に必要な All-to-all 通信の回数を従来の 6 回から 2 回に削減することで，高速化を図っている．All-to-all 通信の性能を向上させるという我々のアプローチとは異なるものの，連携してさらなる高速化が期待される．

6. おわりに

本稿では，低コストで大規模化可能な多層 full-mesh トポロジーを用いたクラスタ環境上で NAS Parallel Benchmarks のフーリエ変換を実行し，結果をまとめた．その結果，通信性能のみでは最大で 11.4%の高速化を達成した．しかしながら，全体性能では 1%程度にとどまっており，改善の余地がある．また，スループットの値でも Intel MPI Benchmarks による基本性能においては InfiniBand QDR の性能における理論限界を達成しているが，期待通りのものではなかった．

以下の課題が残されている．現状の NPB は単一ノード内での並列化が十分に行われていないため，ネットワークコストに比べてノード内処理のオーバーヘッドが大きく，All-to-all 通信 (転置処理) での高速化の効果を十分に発揮できていない．したがって，この部分の高速化・並列化が要求される．また，1 つの物理マシンに多くの VM を立ち上げると Frotran で書いた OpenMPI・InfiniBand の性能が低下する現象が起こっており，それを改善する必要がある．また，本稿は主に Static ルーティングによるものであるが，Dragonfly[13], [14] などの Dynamic ルーティングによるトポロジー上でも実験・比較することも重要である．

参考文献

[1] 中島 耕太, 三輪 真弘: スイッチ台数の削減と高い All-to-all 通信性能を両立する多層 Fullmesh トポロジーの提案, 情報処理学会研研究報告 2014-HPC-145(12), pp.1–7, (2014).
[2] M. Valerio, L. E. Moser and P. M. Melliar-Smith: *Recursively Scalable Fat-Trees as Interconnection Networks*, Proceedings of the 13th IEEE International Phoenix Conference on Computers and Communications, pp. 40–46, (1994).
[3] 清水 俊宏，中島 耕太: 接続台数を最大化するラテン方陣 Fat-Tree における All-to-all 通信での経路競合回避, 情報処理学会研研究報告 2015-HPC-151(3), pp.1–8, (2015).
[4] 清水俊宏，中島耕太: 低コストで大規模化可能なラテン方陣 Fat-Tree における All-to-all 通信の高速化の実現，ハイパフォーマンスコンピューティングと計算科学シンポジウム論文集 2016, pp. 77-87, (2016).
[5] T.Shimizu, M.Masahiro, K.Nakashima: SC'16 Research Posters: Acceleration of All-to-All Communication on Multi-Layer Full Mesh, Low-Cost Connectable Network Topology, The International Conference for High Performance Computing (SC'16), (2016).
[6] InfiniBand Architecture Specification Release 1.3, InfiniBand Trade Association, http://www.infinibandta.org.
[7] D.Takahashi: *Implementation of Parallel 1-D FFT on GPU Clusters*, IEEE 16th International Conference on Computational Science and Engineering, pp.174–180, (2013).
[8] C.Gomez, F. Gilabert, M.E. Gomez, P. Lopez and J. Duato: *Deterministic versus Adaptive Routing in Fat-trees*, In Proceedings of the 2007 IEEE International Parallel and Distributed Processing Symposium (IPDPS07), pp.1–8, (2007).
[9] E. Zahavi, G. Johnson, D. J. Kerbyson, and M. Lang: *Optimized Infiniband Fat-Tree Routing for Shift All-to-all Communication Patterns* , Concurrency Computation Practice and Experience 22 (2), pp.217–231, (2009).
[10] Mellanox, http://www.mellanox.com
[11] NAS Parallel Benchmarks, https://www.nas.nasa.gov/publications/npb.html
[12] OpenMPI, http://www.open-mpi.org
[13] J. Kim, W.J. Dally, S. Scott and D. Abts: *Technology-Driven, Highly-Scalable Dragonfly Topology*, 35th International Symposium on Computer Architecture (ISCA '08), pp.77–88, (2008).
[14] N. Jain, A. Bhatele, N. Xiang, N. J. Wright and L. V. Kale: *Maximizing Throughput on a Dragonfly Network*, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp.336–347, (2014).
[15] E. Totoni and L. Kale: *ACM SRC Poster: Optimizing All-to-All Algorithm for PERCS Network Using Simulation*, International Conference for High Performance Computing, Networking, Storage and Analysis (SC'11), (2011).
[16] Kettimuthu, Rajkumar, and Sankara Muthukrishnan: *A performance study of parallel FFT in clos and mesh networks.*, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA. (2005).
[17] Tsukahara, Hiroshi, et al. "Implementation of low communication frequency 3D FFT algorithm for ultra-large-scale micromagnetics simulation." Computer Physics Communications 207 (2016): 217-220.