VFREC-Net：ドライバ制御による tagged-VLAN を用いた PC クラスタ向けマルチパスネットワーク

三浦信一[†] 岡本高幸[†] 朴泰祐[†]
佐藤三久[†] 高橋大介[†]

VLAN ルーティング法は，コモディティネットワークである Ethernet において，安価な Layer-2 switch 間の接続に柔軟性を持たせ HPC クラスタ向けの高性能なネットワークを構築できる．しかしこれを実現する既存方法は，いくつかの問題により大規模化には適さなかった．我々はこれらの問題点を解決するために，tagged-VLAN を直接制御可能な Linux 用ネットワークデバイスドライバを用意し，これを利用した高バンド幅 Tree Network の構築を支援するシステム，VFREC-Net を開発した．Linux 上で動作する我々の仮想ネットワークデバイスドライバは，IEEE 802.1q フレームを直接制御できるため，フラットな IP アドレス空間と，ユーザ透過なシステムを提供できる．これにより既存の MPI ライブラリにいっさいの変更を加えることなく，NPB のようなアプリケーションが動作可能となった．また，本システムを用いた評価では，既存の Ethernet を用いたツリーネットワークと比較して，高いバイセクションバンド幅を得られ，NPB では最大で 1.85 倍の性能向上を確認できた．結論として VFREC-Net に基づく 2 段の Fat Tree Network 構成において，全 node を大きな単一 switch でフラットに結合した場合と同等の性能がすべてのベンチマークで得られることが分かった．

VFREC-Net: Multi-path Network for PC Clusters Based on Tagged-VLAN Technology with Driver Control

Shin'ichi Miura,[†] Takayuki Okamoto,[†] Taisuke Boku,[†]
Mitsuhisa Sato[†] and Daisuke Takahashi[†]

Gigabit Ethernet has a very high performance/price ratio and is applicable to make a relatively small size of HPC cluster, as the interconnection network. When we increase its size, however, we need to introduce multiple switches, and the links between the switches make a performance bottleneck. VLAN-based routing method is an excellent technique to utilize multiple links between intermediate switches on a cluster with Ethernet although its implementation method is not sophisticated so far. We have developed a special driver for Linux operating system to handle this problem and enable to apply this technique to a real large scale cluster. In this paper, we describe the design and implementation of this driver as well as its performance evaluation on basic bisection bandwidth and NPB. Through the evaluation we confirmed that our method can enhance the bisection bandwidth on VLAN-based Fat Tree. Moreover, on NPB kernels, we confirmed the parallel processing performance on 2-stage Fat Tree based on VFREC-Net system achieves 1.85 times higher performance than ordinary single link. Our solution achieves almost the same performance with a flat network by a large single-stage switch.

1. はじめに

現在，一般の PC クラスタの多くは，node 間を接続するネットワークとして Ethernet を採用している．特に Gigabit Ethernet（以後，GbE）はそのコストパフォーマンスの高さから多くのクラスタ環境で使用されている．GbE 用 NIC の価格は非常に安く，加えて Layer-2 用の GbE switch はある程度の port 数以下であればその port 単価が NIC のそれを下回るほどの低価格である．しかし GbE を大規模な HPC クラスタに用いる場合，大きなネットワークボトルネックが生じる．一般的にコストパフォーマンスの良い Layer-2 GbE switch は，24 port 程度の比較的小規模なものになる．クラスタの規模が 24 node 以下では問題ないが，クラスタが大規模化した場合，複数台の安価な switch

† 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba

を tree 構造等で相互に結合しなければならない．しかし，switch 間を流れるデータトラフィックが性能ボトルネックを引き起こすため，クラスタの性能を node 数に合わせて向上させるためには，その部分のバンド幅をできる限り増強する必要がある．node-switch 間の接続に GbE を用いる場合，switch-switch 間の接続には GbE よりも高速なリンク，たとえば 10 Gigabit Ethernet（10GbE）等を用いるべきである．しかし，10GbE をサポートする switch は GbE のそれよりも極端に高価であり，種類も制限されている．コストパフォーマンスのために GbE を利用している以上，上位リンクでの 10GbE の利用はそのメリットがきわめて乏しくなる．そのような理由により，GbE で構成される HPC クラスタでは，上位リンクの接続にも GbE を用いることが多く，クラスタの性能に制約が生じやすい．そのために Ethernet を用いた HPC クラスタは大規模化には適していなかった．

大規模な HPC クラスタでは汎用ネットワークである Ethernet よりも，高価であるが高性能な専用ネットワークである，Myrinet[7] や InfiniBand[8] 等の SAN（System Area Network）が多く用いられている．SAN では先ほどの問題を解決するために，switch 間の接続を多重化し，それらを同時に利用する trunk 技術を用いることができる．この trunk 技術は LACP[1] 等の形で Ethernet にも存在する．Ethernet における LACP（Link Aggregation Control Protocol）は，switch 間に 2〜8 本程度の並行リンクを用意し，それらを同時に利用することが可能である．しかし，この trunk 結合をする 2 台の switch 間には他の switch を挟むことができず，あくまで 2 台の switch 間の平行結線を行うものである．したがって，たとえば Fat Tree 構造のように上位層ほど多数のリンクを利用する構成を利用する場合，switch の port 数のうち多数を LACP 用に利用し，接続 node あるいは接続 switch 数の制約を生み，結局大規模化には対応できない．LACP に用いるリンク数を減らせばバンド幅制約を生じ，性能が抑えられてしまう．

これらの問題点を解決するための有効な方法の 1 つとして，VLAN ルーティング法[9] が提案されている．VLAN ルーティング法を用いると，既存の VLAN 技術[2] を使用し switch 間の接続に柔軟性を持たせることが可能になる．しかし，文献 9) で提案された VLAN ルーティング法は，これを HPC クラスタに用いることが原理的に可能であることを示しているが，実現方法や拡張性等の点で様々な問題を含んでいる．本稿では，既存の VLAN ルーティング法の問題点を示し，それを解決するための方法を示す．

2. VLAN ルーティング法

2.1 既存研究

Ethernet の switch 間に複数のパスを形成できない理由は，それによりネットワーク中に loop が形成され，その間にブロードキャストストームが発生するからである．Ethernet パケットそのものに，switch 間の経路を任意に決定する規格が盛り込まれていないため，loop のあるネットワークでは，一意のネットワークルートを決定できない．この問題を解決する VLAN ルーティング法と呼ばれる VLAN 技術を用いたルーティング方法が提案されている[9)〜11)]．VLAN ルーティング法は IEEE 802.1q[2] で規格化された tagged-VLAN 技術を応用し，物理的に loop のあるネットワークを論理的に木構造のみからなるネットワークに分割する．

しかし文献 9) に基づく既存研究では，これらの環境を構築するために，Linux 上の標準的な VLAN 実装を用いていた[2),3)]．この技術のみを用いて Linux で VLAN ID の操作を行うためには，各 VLAN ID に対応した仮想ネットワークデバイスを用意し，それぞれに独立した IP アドレスとサブネットを割り当て，OS のルーティングアルゴリズムを介することが必要である．**図 1** のようなネットワークで node A が node C に対して通信する場合，VLAN ID = 1 の経路を使用したい場合は 192.168.1.3 を，VLAN ID = 2 の経路を使用したい場合は 192.168.2.3 をそれぞれアクセスしなければならない．このような環境を用いて，たとえば MPI を用いた並列処理を行う場合，switch 間に用意されたパスを偏りなく使用するために，送受信する node のペアごとに VLAN ID，すなわち IP アドレスを変える必要がある．node A - node C は VLAN ID = 1 を利用するが，node B - node C では VLAN ID = 2 を利用しなければならず，MPI のすべての node でそれぞれ異なる設定ファイルを用意する必要があり，管理は複雑になる．加えて，クラスタ規模が

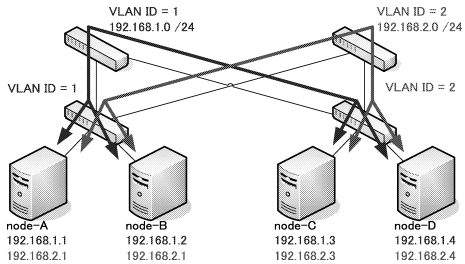

図 1 複数の経路を持つネットワーク
Fig. 1 Multipath network.

大きくなり，使用する VLAN ID が増大した場合，仮想ネットワークインタフェースが増えることでその管理が複雑となり，また OS が行うルーティング処理のために，通信にオーバヘッドが発生する恐れがある．したがって，この従来手法を大規模 HPC クラスタに本格的に適用するのは難しい．

2.2 提案手法

既存手法の問題点は，VLAN ID の制御[3] が IP におけるサブネット単位での大雑把な制御になり，送信元・受信先に応じた細かい制御ができないことである．

我々はこれを回避するために，IP アドレスベースでの VLAN ID 制御ではなく，MAC アドレスベースでの VLAN ID の制御を行うことを提案する．ただし，現在の Linux 上 IEEE 802.1q ドライバ実装には，MAC アドレスを基にした VLAN ID を決定する手段がない．そのため，我々は MAC アドレスに基づき VLAN ID を決定するネットワークデバイスドライバを，Linux 用 IEEE 802.1q ドライバを拡張することによって実現する．ルーティングのために IP アドレスを用いず MAC アドレスを用いた理由は，ネットワークデバイスドライバが担当するパケットの処理が MAC アドレスを処理するレイヤと近接しているため，送受信先の MAC アドレス情報の取得が簡単であること，加えて，IP によらない Ethernet を用いた通信でも処理可能であるからである．

VLAN ID の制御をネットワークデバイスドライバ上で行うことで，ユーザは通常の Socket API を使用でき，通常の IP を用いた通信が可能になる．既存手法では，ユーザは利用する VLAN ID ごとにサブネットを選択する作業が必要であるが，本実装ではユーザから見たネットワークはフラットになり，管理が容易になる．一般のユーザが意識せずに標準の TCP/IP を用いた通信が可能になるため，既存のネットワークプログラムが変更なく使える．また，現在 PC クラスタで多く用いられている MPICH[4] や LAM[5] 等の MPI ライブラリも Ethernet 上での実装は TCP/IP を使用しているため，これらにいっさいの変更を加えずに使えることは大きなメリットになる．

3. VFREC-Net

我々は先に示した既存の VLAN ルーティング法を拡張した新しいシステムを実装する．このシステムを VFREC-Net（VLAN-based Flexible, Reliable and Expandable Commodity Network）と名づける．このシステムは，実際に MAC アドレスに基づき tagged-VLAN の制御を行う VFREC-Net ドライバ，およびそれを制御するドライバコントローラで成り立つ．我々はこの VFREC-Net システムを，Linux で提供されている VLAN ドライバ[3] を基に実装した．

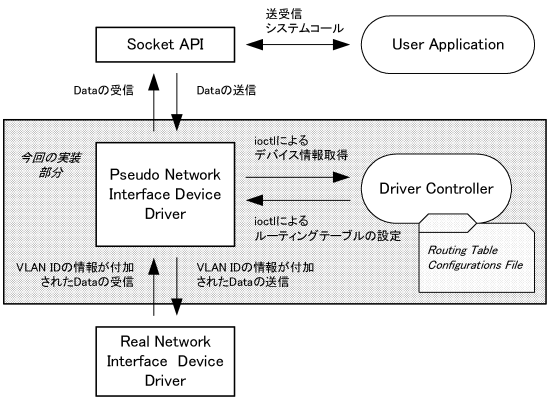

図 2 VFREC-Net ドライバの実装
Fig. 2 Implementation of VFREC-Net driver.

3.1 デバイスドライバの役割

VFREC-Net ドライバの実装を**図 2** に示す．VFREC-Net ドライバは，socket API と本来のネットワークデバイスドライバの間に位置する．VFREC-Net ドライバは，socket API からの要求を本来のネットワークデバイスの代わりに処理し，本来のデバイスに伝える．本ドライバの主な処理は以下のとおりである．

送信

- Socket API より送信を指示されたデータを解析し，VLAN ID を決定する．
- 送信データに VLAN ID を加え，ネットワークデバイスに送信要求を行う．

受信

- IEEE 802.1q のフレームをデバイスが受信した場合に，このデバイスドライバへと渡すように指示する．
- 受信データから VLAN tag を除去し，socket API に渡す．

送信時に VLAN ID を決定する方法は，socket API から渡された送信データに設定されている MAC アドレスに基づく．この MAC アドレスとドライバに設定されているルーティングテーブルを比較することで VLAN ID を決定する．

3.2 ルーティング方法とその設定

本実装では静的なルーティング処理を前提とする．静的ルーティングにおいてルーティングテーブルを決定する場合，一般に送信元アドレスで VLAN ID を決定する Source Routing 法，送信先アドレスに応じ

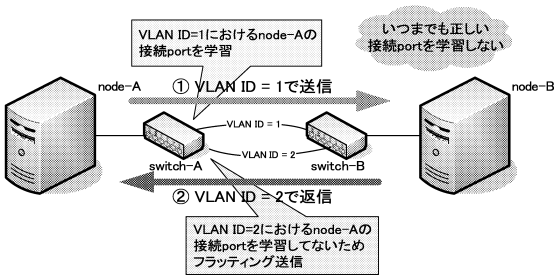

図 3 Switch の MAC アドレス学習アルゴリズムの問題点
Fig. 3 Problems of MAC address learning algorithm of switch.

て VLAN ID を決定する Destination Routing 法，そして，送信元と送信先のアドレスを比較したうえで VLAN ID を決定する Source-Destination Routing 法の 3 種類の方法があげられる．しかし，switch を介した通信の場合，switch の MAC アドレス学習メカニズムを考慮する必要がある．一般的な IEEE 802.1q 対応の Layer-2 switch では，ある node 間でパケットの送受信経路（VLAN ID）が異なる場合，MAC アドレスの学習ができない特性がある．すなわち，MAC アドレス学習は VLAN ID ごとに独立に行われ，これが異なるとそれ以前の学習結果が反映されない．たとえば，node A と node B 間で通信を行う場合，A から B への通信経路の VLAN ID と B から A へのそれが異なる場合，中継する switch はいつまでも相手 node の MAC アドレスとそこに到達するための port の関係を学習できない（**図 3**）．MAC アドレスを学習できない場合，その switch ではいつまでもパケットをフラッティング送信し，ネットワーク全体の性能を低下させる．したがって，Source Routing，Destination Routing はともに，送受信する相手と自分の関係を考慮にいれずに VLAN ID を決定する手法のためにこの問題が生じる．以上の理由により，VFREC-Net では Source-Destination Routing を採用し，必ず node 間で決められた同一の VLAN ID を利用する．

ドライバへのルーティングテーブルの設定は，ドライバの初期化時に外部のドライバコントロールツールから *ioctl()* を通じて行う．コントロールアプリケーションは，ルーティングテーブルが書かれたコンフィグレーションファイルを読み込み，ドライバが動作している node に必要なルーティングテーブルを再構築したうえでドライバに設定する．コンフィグレーションファイルは，本ネットワークに所属するすべての node で共有することが可能である．

共有される設定ファイルには，全 node に関する通信組合せの VLAN ID テーブルを記述する必要がある．このため node 数が n の場合，この組合せは $O(n^2)$ となり，クラスタの規模が大きくなるにつれ，設定ファイルサイズは大きくなり複雑になりやすい．これを回避するために，我々は各 node に優先度を与えることでルーティングテーブルを決定する．設定ファイルには，MAC アドレスとその場合に用いる VLAN ID，そして優先度を記述する．自分の MAC アドレスよりも送信先の MAC アドレスの優先度が高い場合は，相手の MAC アドレスに設定されている VLAN ID を用いて通信を行う．優先度が自分の MAC アドレスよりも低い場合や，通信の初めの段階で送信 MAC アドレスが決定していない場合におきる ARP 等のメッセージでは，自分に割り当てられている VLAN ID で送信を行う．このような仕組みを利用することで設定ファイルサイズは $O(n)$ となり，設定が簡便になる．現在標準的に用いている優先度は「node 番号（0〜n-1）の低い方を優先する」という単純なものである．これにより，n^2 通りの送受信 node の組合せにおいて，結果的に全通信で使用される VLAN ID を等分配できる．

3.3 ネットワークトポロジ

過去にも並列計算機用に様々なネットワークが提案されているが，既存研究[9] では，VLAN ルーティング法を用いることで，今まで専用並列計算機でのみ実現できたネットワークを，VLAN ルーティング法を用いて実現できることを示している．しかし一般的には，すべての HPC アプリケーションに柔軟に対応できる Tree 形のネットワークトポロジが最適である．そこで，本実装では Tree 構成を拡張した Fat Tree 構成を念頭に置く．文献 9) では VLAN ルーティング法に基づいて構築された Fat Tree を VBFT（VLAN-Based Fat Tree)☆と名づけており，今回我々はこれをモデルとして VFREC-Net の実装と評価を行う（**図 4**）．

3.4 Layer-2 switch の VLAN 設定

VFREC-Net は各 node における NIC 上の VLAN ID 制御と，これを適切にルーティングするネットワーク上の多段 switch の VLAN ルーティングから構成される．したがって，すべての switch 上で適切な VLAN 構成をあらかじめ設定する必要がある．この作業は必ずしも容易ではなく，各 node を接続する port と上位リンクに対応する port の関係を考慮し，1 つずつ VLAN を設定しなければならない．また，1 つの node から来るパケットは複数の VLAN ID のどれかを持

☆ このような Fat Tree 構成のネットワークは今までにも大型計算機や SAN でも使用されてきたが，Ethernet では VLAN ルーティング法を利用することで利用可能になり，それを区別するため VBFT と呼ぶことにする．

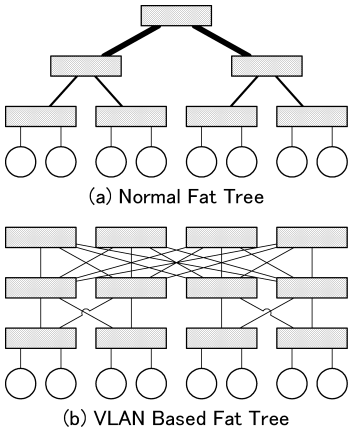

図 4 一般的の Fat Tree と，VLAN ルーティング法で可能になる Fat Tree 構成の VBFT（VLAN Based Fat Tree）
Fig. 4 Normal Fat Tree & VLAN Based Fat Tree (VBFT).

つため，switch 上の VLAN 構成も重複した port を持つ．

現在の VFREC-Net は switch の設定をマニュアル操作に頼り，今回の実装でもすべての switch を人手で VLAN 設定した．しかし，VLAN 構成を可能とする switch のほとんどは，その設定を何らかの configuration file で残すことができ，これを VFREC-Net の構成に合わせて自動的に構築することは不可能ではない．この点は今後の課題である（7 章参照）．

4. 評　　価

実装したデバイスドライバの性能を評価する．評価環境として，**表 1** に示す node の構成で合計 16 node のクラスタを構築した．比較するネットワークは**図 5** に示す 7 種類である．Tree および VBFT では，node と直接接続する switch を複数台用意する．以後説明のために node に直接接続する下部の switch の台数を n と表現する．また，VBFT では上位の switch も複数台あるため，これを m と表現する．(a) Flat では，単独の switch にすべての node を接続する．(b), (c) の Tree では n 台の switch 間を 1 台の上位 switch を介して接続し，通常の Ethernet ドライバを利用をする．(d)〜(g) で示す VBFT では n 台の switch をそれぞれ m 台の switch に接続したうえで，開発した VFREC-Net ドライバを用いる．実験ネットワーク (b)，(d) および (e) では $n = 2$ となり，(c)，(f) および (g) では $n = 4$ となる．(d)，(f) では $m = 2$，(e), (g) では $m = 4$ となる．以後，7 種類のネットワーク構成を図 5 に基づき，Flat，Tree，VBFT(n, m) と表現する．VFREC-Net ドライバを用いる場合のルーティングテーブルの優先度付けは，node 番号が若い順に優先度が高いことにし，それぞれに割り当てられる VLAN ID は，1 から m までの ID をサイクリックに node 番号の若い順から割り当てる．これらの環境を利用し，基本性能評価として通信遅延時間，スループット，バイセクションバンド幅を計測，その後，一般的なベンチマークとして NAS Parallel Benchmarks を評価する．

表 1 評価環境
Table 1 Evaluation environment.

node	
CPU	Intel Xeon 3.0 GHz EM64T 1-way
Memory	DDR2/400 1.0 Gigabytes
NIC	Intel 82541EI Gigabit Ethernet (PCI-X 64 bit/133 MHz)
OS	Linux Kernel 2.6.14
MPI	LAM ver.7.1.1
Compiler	GCC ver.4.0

switch
DELL PowerConnect 5224 Gigabit Ethernet 24 Port

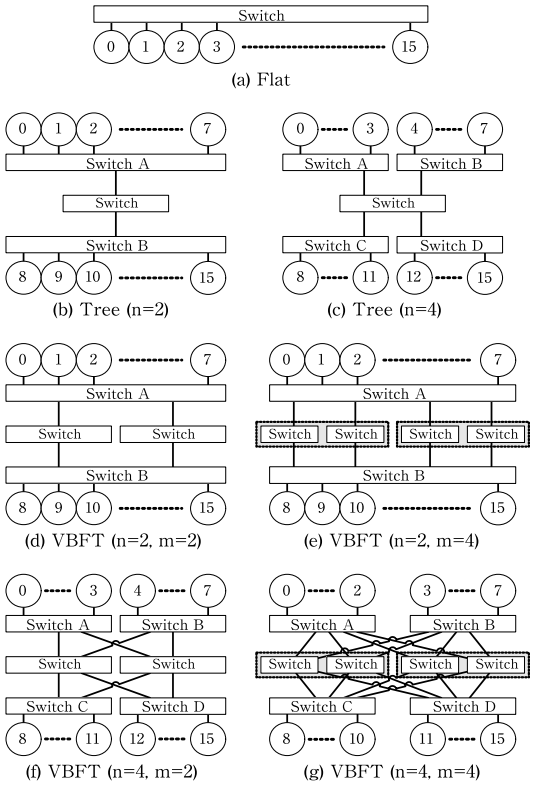

図 5 実験ネットワーク
Fig. 5 Network for experiment.

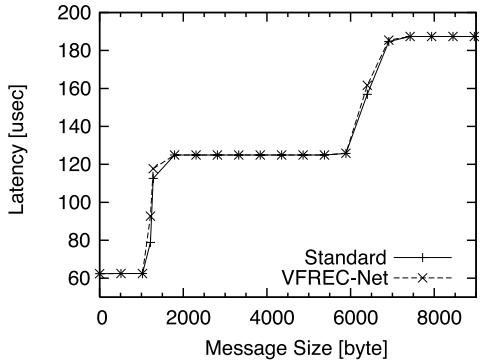

図 6 UDP/IP を用いた場合の片方向通信遅延時間
Fig. 6 Latency of UDP/IP.

4.1 基本性能評価

4.1.1 単一リンクでの通信遅延時間

開発したデバイスドライバは，socket API から渡された送信データを解析し MAC アドレスから VLAN ID を決定した後に，本来のデバイスドライバで送信する．そのため通常のデバイスドライバを用いる場合と比較して，オーバヘッドが付加される．本実験は，このオーバヘッドを計測するために，UDP/IP を用いた場合の通信遅延時間について評価する．評価では Flat ネットワークを用いて，標準的な Ethernet ドライバおよび VFREC-Net ドライバを用いた場合の 2 種類の環境を利用する．これらの環境でメッセージサイズを変えながら，node 0–1 間で 1,000 回の ping-pong を行うときの平均の片道時間を計測する．結果を**図 6** に示す．

結果が示すように，使用するドライバによる性能差はない．VLAN にともなう大きな処理は，送受信データへの tag 操作である．HPC クラスタで多く用いられる比較的高性能な NIC には，IEEE 802.1q tag 制御の専用のハードウェアが装備されている．本実装ではこの機能を利用することでオーバヘッドは小さくなる．このような機能がない NIC を用いた場合，通信遅延時間は大きくなる可能性がある．

我々が開発した VFREC-Net ドライバは，従来の VLAN ルーティング手法で用いられていた Linux 用 VLAN ドライバ（8021q[3]）を拡張したものである．我々のドライバを用いた場合と標準的な Ethernet ドライバを用いた場合で通信遅延時間にほとんど差異がないことから，VFREC-Net ドライバは従来のドライバと比較してオーバヘッドは生じてないと結論できる．

4.1.2 単一リンクでのスループット

次に最大スループットを計測する．計測には Iperf[6] ver.1.7.0 を用いる．ここで比較として，(a) Flat，(b) Tree（$n = 2$）および (d) VBFT（$n = 2$，$m = 2$）のそれぞれ node 0–8 の間のスループットを計測する．Iperf で GbE の性能を最大限に引き出すために TCP Window Size を 128 Kbyte に設定する．参考までに (a) Flat に接続した場合で，VFREC-Net ドライバを用いた評価も行う．結果を**表 2** に示す．

表 2 Iperf を用いた 2 点間における最大スループット
Table 2 Throughput between a pair of nodes.

環境	Throughput
(a), node-0 & node-8	940 Mbps
(a), node-0 & node-8（ドライバ使用）	938 Mbps
(b), node-0 & node-8	940 Mbps
(d), node-0 & node-8（ドライバ使用）	938 Mbps

結果のように，スループットは標準的な Ethernet ドライバと，今回用意した VFREC-Net ドライバの性能に大きな差はない．本ドライバの実装で 2 Mbps ほど性能が落ちているのは，それぞれのパケットに IEEE 802.1q tag（4 Bytes）を付加しているためであり，デバイスドライバの処理自体のオーバヘッドによるものではない．

4.1.3 バイセクションバンド幅

これまでの評価は，1 組の node ペア間での通信において，実装したデバイスドライバが通常のデバイスドライバと比較した場合の性能低下が生じないことを確認するものであった．次に VBFT 結合において，実装したドライバが正しく動作することを確認し，その場合のネットワーク性能を評価する．図 5 のネットワーク構成において，switch A（または A&B）に接続された node 0–7 から，switch B（または C&D）に接続された node 8–15 へパケットを一斉送信（送信 node の番号を i とすると，受信 node の番号は $(i + 8)$ となる）し，そのとき得られたバンド幅を計測する．受信側で観測されるバンド幅の合計が片方向のバイセクションバンド幅になる．ここでは MPI を用いて評価する．結果を**図 7** に示す．

結果のように，switch 間の接続が 1 link のみである標準的な Tree 構成のネットワークでは，バイセクションバンド幅が制限される．その値は (b) では約 120 MBytes/sec，(c) では 240 MBytes/sec となる．一方で我々の開発したドライバを用い，マルチリンクを活用した VBFT では，switch 間に用意された link 数によりバイセクションバンド幅が増加し，node の台数に対して十分な switch 間接続（m）が用意された場合には，node が Flat に接続された状態と同様のバイセクションバンド幅を示している．この結果より，我々の開発した VFREC-Net ドライバが有効に

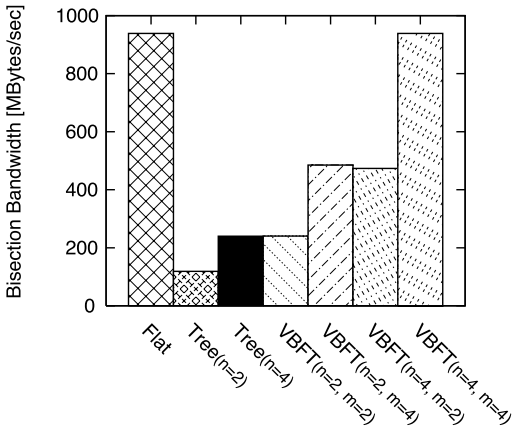

図 7 バイセクションバンド幅
Fig. 7 Bi-section bandwidth.

機能していることが確認できた．

以上の結果から単一ストリームの通信遅延時間およびスループットには VFREC-Net デバイスドライバを用いる場合と，そうでない場合では大きな性能差はない．それらに加えて VFREC-Net ドライバを用いることでバイセクションバンド幅が改善されるため，このドライバと VBFT の構成を組み合わせることで，一般的な Tree のネットワークと比較して高い性能を示すことが期待できる．

4.2 NAS Parallel Benchmarks

最後に，実際のアプリケーションを想定して NAS Parallel Benchmarks（NPB）を評価する．本評価には，NPB ver.3.2，CLASS = B，PROCS = 16 を用いる．ベンチマークとして，EP，FT，IS，MG，CG の 5 カーネルを対象とし，Flat との相対性能を評価する．**図 8** に評価結果を示す．

評価結果 EP，FT，IS

まず EP では計算中に通信はほとんど発生しないため，計算にネットワーク性能を必要としない．そのため，どの種類のネットワークを使った場合においても，性能は変化することはない．FT，IS ベンチマークではともに，バイセクションバンド幅の改善の効果が，クラスタの性能増大に寄与している．通常の Tree 構成では，Flat の性能に対して 6 割程度の性能に抑えられるのに対して，VBFT 構成では，上位 switch 数の増大とともに Flat の性能に近づいている．特に IS の場合，本来最高性能を示すと考えられる Flat 構成よりも VBFT（$n = 4$，$m = 4$）構成が高い性能を示している（Tree（$n = 4$）の 1.85 倍，Flat の 1.14 倍）．この理由として，node に接続する switch が増えることで，switch のトータルのバンド幅が flat のときよりも高くなり，その影響が VBFT 構成をとるときのオーバヘッドを上回るのが原因だと考えられる．

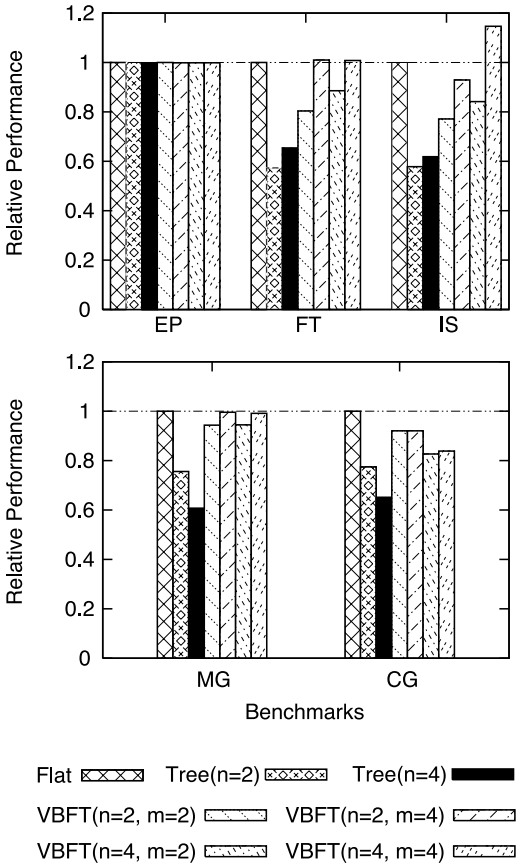

図 8 NAS Parallel Benchmarks 結果
Fig. 8 NAS Parallel Benchmark result.

評価結果 MG，CG

FT，IS では，バイセクションバンド幅の改善が，そのままベンチマーク性能に反映されていた．しかし MG，CG ではバイセクションバンド幅の改善が性能向上に結び付いていない．MG では Tree 構成で $n = 2 \rightarrow 4$ と変化した際，バイセクションバンド幅が増えているにもかかわらず，性能は低下している．これは node を複数の switch に分割することで計算の通信アルゴリズム上，switch 間をまたぐ通信が増加したことが原因である．しかし VBFT 構成にすることでその問題が解決し，$m = 4$ の場合 Flat の性能に近づく結果となる．CG の場合でも Tree 構成で $n = 2 \rightarrow 4$ と変化した場合に性能が低下しているのは MG と同様である．しかしながら CG では，今までの他のベンチマークと異なる特性を示している．すなわち，VBFT 構成時において $n = 2 \rightarrow 4$ になった場合に性能が低下し，次に，下部 switch どうしを接

続する経路（上位 switch 数 m）が $2 \rightarrow 4$ と増えているにもかかわらず性能向上は頭打ちになっている．この結果は，現在適用している優先度付け VLAN ID 決定手法が CG の通信アルゴリズムと相性が悪く，適切に switch 間の経路を分散できないためと推測できる．次にこの推測に基づき，CG の最適なルーティングを検討する．

5. Kernel-CG におけるルーティングの最適化

CG の結果で性能向上が得られない理由は，不適当なルーティングアルゴリズムを用いたためである．そこで，適切なルーティングを行うことにより負荷分散を適正化することが可能かどうかを実験的に確かめてみる．ここで問題となる通信部分はベクトル転置通信である．この通信は**図 9** に示すような通信パターンになる．VFREC-Net で VBFT 構成をとった場合に使用された通信経路（VLAN ID）の割当てを**表 3** に示す．これから，下部 switch 数（n）が 2 である場合，switch 間接続数（m）が $2 \rightarrow 4$ と増加した場合でも，switch 間にまたがる通信では，同一の経路を利用することになり，負荷が分散しないことが分かる．また $n = 2 \rightarrow 4$ と増やした場合 $n = 2$ と比較して通信経路の偏りが大きいことが分かる．このような環境で性能向上を得るためには，表 3 の最適化例に示すような経路を選択するルーティングテーブルを VFREC-Net 側に設定する必要がある．

これらの検討を基に上位リンクの負荷を CG に最適に分散するルーティングテーブル設定したクラスタ環境でのベンチマーク結果を**図 10** に示す．reference data は通常の Flat および Tree の場合を，non-optimized は図 8 に示した最適化前の VBFT を，optimized はここで行った最適化による VBFT の結果を示している．結果が示すように，上位リンク数が増加（$m = 2 \rightarrow 4$）した場合では，古いルーティングテーブル決定手法では速度向上が得られないが，ルーティング方法を最適化した場合，性能が向上することを確認できた．また下部 switch 数を増加させた場合（$n = 2 \rightarrow 4$）においてもバイセクションバンド幅に応じた性能向上を確認できた．

このように一般の Ethernet で用いられる通常の Tree 構造と比較して，VFREC-Net を用いた VBFT では高い性能を示し，また Tree だけでなく Flat なネットワークと比較した場合においても，同等の性能を示している．現在の VFREC-Net 実装では，CG において速度向上を得られないが，その問題点はルーティングテーブル決定手法にあり，それを改良することで CG においても速度向上を期待できる．

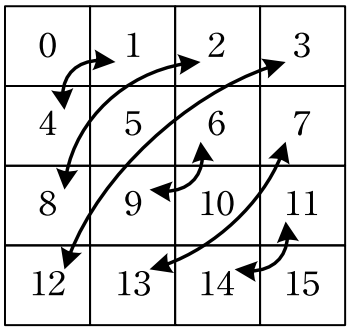

図 9 CG のベクトル転置計算で発生する通信
Fig. 9 Matrix transport communication on Kernel-CG.

表 3 CG のベクトル転置計算で発生する通信の VLAN ID 割当（* $n = 2$ の場合は同一 switch 内での通信になり，性能への影響は少ない）
Table 3 VLAN ID mapping on Kernel-CG.

組合せ	$m = 2$	$m = 4$	最適化例（$m = 4$）
1-4	2(*)	2(*)	1(*)
2-8	1	3	2
3-12	2	4	3
6-9	1	3	4
7-13	2	4	1
11-14	2(*)	2(*)	2(*)

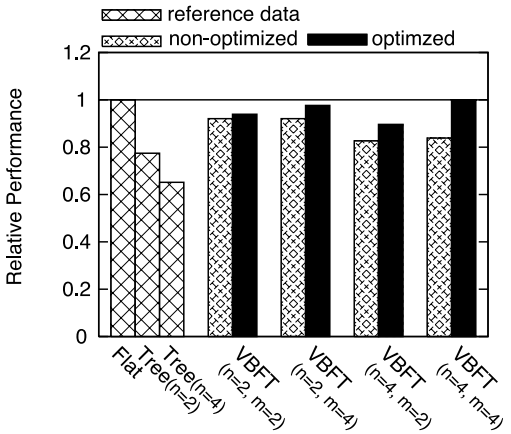

図 10 通信最適化による CG の性能
Fig. 10 Performance result of Kernel-CG with communication optimization.

6. 大規模化への検討

本稿で評価したクラスタは比較的小規模な構成であるが，本システムは数百から数千の node 数までのクラスタを想定している．ここで，クラスタの規模が大きくなる場合の問題点について述べる．

6.1 拡張性

本実装はルーティングに VLAN ID を用いる．そのため，システムの拡張の度合いは switch が対応できる VLAN ID 数に依存する．理論的に VLAN では最大 4095 の VLAN ID を使用できる．しかし，比較的安価な switch でサポートしている VLAN ID は 256 程度である．しかしながら，256 程度の VLAN ID が用いることが可能であるならば数千台規模のクラスタには十分対応できる．

例として，比較的安価な 24 port の switch を用いて 4096 node のクラスタを VBFT を用いて構成する場合を考える．1 台の switch の全 24 port のうち 16 port を下位からの接続に，8 port を上位への接続に用いるものとする．この場合，最大 3 段の接続で全 node を接続できる．通常の Tree では，直接 node と接続する最下位 switch は $4096/16 = 256$ 台，2 段目の switch は $256/16 = 16$ 台，最上位の 3 段目は $16/16 = 1$ 台の switch で接続可能である．VBFT の接続では，最下位の switch 以外を水平方向に展開する．switch に用意した上位への接続 8 port をすべて利用する場合，2 段目の switch は $16 \times 8 = 128$ 台，3 段目の switch は $1 \times 8 \times 8 = 64$ 台用意する必要がある．VBFT では，どの最上位 switch を経路として使用するのかを VLAN ID を用いて制御するため，最上位の switch の数が最大で用いる VLAN ID の数になり，この場合 64 ID となる．そのため，現在 switch でサポートしている 256 ID で不足することはない．

この設定方法では，接続段数が増えるにともなって相対的にバイセクションバンド幅が低下する．バイセクションバンド幅を維持しながらシステムサイズを拡張するためには，switch に用意された 24 port を可能な限り多く上位への接続に用いればよい．これは単純に switch 数の増加，すなわちコストの問題につながり，システム全体に占めるネットワークのコストの割合により調節可能である．また，最近では 48 port 程度の switch も比較的安価になりつつあることから，段数を抑えつつ，上位階層へのリンク数を増やす，または下位階層への接続 node または switch 数を増やすことがより容易になる．したがって，さらなる大規模化が可能であり，理論的には数万 node までの拡張も可能である．

6.2 耐故障性

本システムではルーティングの指標として MAC アドレスを用いるため，故障した node を代替 node に置き換えた場合，その代替 node に設定されている MAC アドレスを基に，他の node すべてに新規のルーティングテーブルを設定する必要がある．現在の実装ではルーティングテーブルを各 node でローカルファイルという形で保持しているため，代替 node を用意した場合でも，これを更新するためにいくぶんの手間とシステム停止時間を必要とする．この問題を解決するために，NIS またはその他の手段を用いてルーティングテーブルをすべての node で共有する方法を実装する予定である．本システムの各 node では，実際にクラスタの通信を行うネットワーク以外に NIS，NFS といった一般的なクラスタ管理を行うネットワークを別に用意することを想定している．新規の node に交換した場合でも，このような管理ネットワークを用いることでスムーズに新規のルーティングテーブルを各 node に設定し，故障によるシステム全体の停止時間を小さくすることを可能にする予定である．

7. 今後の計画

7.1 ルーティングテーブルの最適化手法の検討

我々の開発したデバイスドライバは，ルーティングテーブルを静的に決定している．しかしながら，HPC 計算の通信パターンは一定ではなく，このルーティングテーブルが最適であると限らない．現実として NPB の CG では，それにより性能向上が制限されている．クラスタの規模が小さい場合は，それらの通信パターンによりルーティングテーブルを再設定しても問題ないが，本システムのターゲットは比較的大規模なクラスタである．現状の優先度付けルーティングアルゴリズムを見直し，より多くのアプリケーションにも対応するように考えなければならない．我々はこの問題を解決するために，動的なルーティングテーブルの再構成を考えている．我々の手法ではルーティング方法をアプリケーション実行中に変更することも原理的に可能である．したがって，アプリケーションの特性に従ったアダプティブなルーティング制御を行うことを検討している．このため，ユーザにルーティングテーブルを変更する特別な API を提供し，計算途中でも通信のパターンに応じてダイナミックにルーティングテーブルを変更できるシステムを提供する．また定期的に通信ログを取得することで，各経路に最適に負荷分散されるようにシステム側が自動的にルーティングテーブルを変更する方法や，アプリケーションの予備的実行に基づくプロファイリングを用いた制御等についても考える．

7.2 VLAN の環境を管理するシステムの開発

VFREC-Netでは，tagged-VLAN を用いて，switch 間に複数の経路を提供することが可能であるが，その

ネットワーク構成は非常に複雑であり，管理や導入時に必要な負担が大きい．そこで，トータルにネットワークを管理するシステムに拡張する．たとえばVFREC-Netの初期導入時の各switchの設定は非常に複雑になりやすい．そこであらかじめ主要なswitchの設定ファイルを自動的に生成するスクリプトをユーザに提供する手法を検討する．

8. おわりに

本稿では，既存のVLANルーティング法の持つ問題点を解決し，ユーザ透過なネットワーク環境を構築するために，VLANルーティング法を実現するデバイスドライバを中心とした，対価格性能比の高いEthernet用クラスタ向けネットワークであるVFREC-Netを開発した．その結果，switch間を1 linkで接続する既存のTreeネットワークと比較した場合，ネットワークのバイセクションバンド幅が大幅に改善され，クラスタシステム全体の性能を大きく向上させることが可能になった．本システムは原理的に，VLAN routingに基づくあらゆるネットワーク構成に対応可能であり，数万nodeからなる大規模クラスタにも適用可能な拡張性を持つ．既存の研究では一般的なLAMやMPICHといったMPIシステムを使う場合，各ホストでMPIのホストファイルの内容をそれぞれのnodeに合わせて別々に用意する必要がある．本システムではMACアドレスベースでVLAN IDを制御するデバイスドライバを用意し，Source-Destinationルーティングテーブルをすべてのnodeで共有することが可能になるため，クラスタの管理を簡略化することが可能になった．また本システムで提供するデバイスドライバは，OSまたはユーザから見た場合，単一のEthernetデバイスそのものであるため，フラットなIPネットワークを構築でき，既存のTCP・UDP/IPのプログラムに手を加えることなく使用できる．そのため，HPCの分野に限らず幅広い分野での応用も期待できる．

謝辞 本研究を行うにあたり，貴重な助言・提言をいただいたCREST「メガスケールクラスタ研究チーム」のメンバに深く感謝します．本研究の一部は，科学技術振興事業団「戦略的創造研究推進事業（CREST）—情報社会を支える新しい高性能情報処理技術—『超低電力技術によるディペンダブルメガスケールコンピューティング』」および文部科学省科学研究費補助（基盤研究（C）17500031）による．

参考文献

1) IEEE 802.3ad.
http://www.ieee802.org/1/pages/802.1ad.html
2) IEEE 802.1q.
http://www.ieee802.org/1/pages/802.1Q.html
3) 802.1Q VLAN implementation for Linux.
http://scry.wanfear.com/ greear/vlan.html
4) MPICH.
http://www-unix.mcs.anl.gov/mpi/mpich/
5) LAM.
http://www.mpi.nd.edu/lam/
6) Iperf Benchmark.
http://dast.nlanr.net/Projects/Iperf/
7) Myrinet.
http://www.myri.com/myrinet/
8) Infiniband.
http://www.infinibandta.org/
9) 工藤知宏，松田元彦，手塚宏史，児玉祐悦，建部修見，関口智嗣：VLANを用いた複数パスを持つクラスタ向きL2 Ethernetネットワーク，*IPSJ Transaction*, Vol.45, No.SIG 6 (ACS 6), pp.35–43 (2004).
10) 大塚智宏，鯉渕道紘，上樂明也，工藤知宏，天野英晴：VLANを用いたマルチパスEthernetにおける経路構築法，IPSJ研究報告2005-ARC-164, pp.115–120 (Aug. 2005).
11) Otsuka, T., Koibuchi, M., Jouraku, A. and Amano, H.: Vlan-based minimal paths in pc cluster with ethernet on mesh and torus, *Proc. International Conference on Parallel Processing* (*ICPP'05*), pp.567–576 (2005).

(平成18年1月27日受付)
(平成18年5月 5 日採録)

三浦 信一（学生会員）
昭和54年生．平成14年千歳科学技術大学光科学部光応用システム学科卒業．平成16年筑波大学大学院理工学研究科修士課程修了．現在，筑波大学大学院システム情報工学研究科在学中．クラスタコンピューティング等に関する研究に従事．

岡本 高幸（学生会員）
昭和58年生．平成18年筑波大学第三学群情報学類卒業．現在，同大学大学院システム情報工学研究科在学中．クラスタ用ネットワークに関する研究に従事．

朴　泰祐（正会員）

昭和 36 年生．昭和 59 年慶應義塾大学工学部電気工学科卒業．平成 2 年同大学大学院理工学研究科電気工学専攻後期博士課程修了．工学博士．昭和 63 年慶應義塾大学理工学部物理学科助手．平成 4 年筑波大学電子・情報工学系講師，平成 7 年同助教授，平成 16 年同大学大学院システム情報工学系助教授，平成 17 年同教授，現在に至る．超並列計算機アーキテクチャ，ハイパフォーマンスコンピューティング，クラスタコンピューティング，グリッドに関する研究に従事．平成 14 年度および平成 15 年度情報処理学会論文賞受賞．日本応用数理学会，IEEE CS 各会員．

佐藤　三久（正会員）

昭和 34 年生．昭和 57 年東京大学理学部情報科学科卒業．昭和 61 年同大学大学院理学系研究科博士課程中退．同年新技術事業団後藤磁束量子情報プロジェクトに参加．平成 3 年通産省電子技術総合研究所入所．平成 8 年新情報処理開発機構並列分散システムパフォーマンス研究室室長．平成 13 年より，筑波大学システム情報工学研究科教授．同大学計算科学研究センター勤務．理学博士．並列処理アーキテクチャ，言語およびコンパイラ，計算機性能評価技術，グリッドコンピューティング等の研究に従事．IEEE，日本応用数理学会各会員．

高橋　大介（正会員）

昭和 45 年生．平成 3 年呉工業高等専門学校電気工学科卒業．平成 5 年豊橋技術科学大学工学部情報工学課程卒業．平成 7 年同大学大学院工学研究科情報工学専攻修士課程修了．平成 9 年東京大学大学院理学系研究科情報科学専攻博士課程中退．同年同大学大型計算機センター助手．平成 11 年同大学情報基盤センター助手．平成 12 年埼玉大学大学院理工学研究科助手．平成 13 年筑波大学電子・情報工学系講師．平成 16 年筑波大学大学院システム情報工学研究科講師．博士（理学）．並列数値計算アルゴリズムに関する研究に従事．平成 10 年度情報処理学会山下記念研究賞，平成 10 年度，平成 15 年度情報処理学会論文賞各受賞．日本応用数理学会，ACM，IEEE，SIAM 各会員．