2005－HPC－104（3）
2005／10／7

tagged-VLANに基づくPCクラスタ向け高バンド幅ツリーネットワークの開発

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，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，いくつかの問題点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，それを利用した高バンド幅ツリーネットワークを構築した．

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．これによって既存のMPIライブラリに一切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した．また，本システムを用いた予備評価では，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確認できた．

High-bandwidth Tree Network for PC Clusters based on Tagged-VLAN Technology

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

Gigabit Ethernet has a very high performance/price ratio and 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 a good 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 preliminary performance evaluation. Through the evaluation we confirmed our method can enhance the bisection bandwidth on VLAN-based Fat Tree.

1. はじめに

現在，一般のクラスタの多くでは，node間を接続するネットワークとしてEthernetが採用されている．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使用されている．GbE用NICの価格は非常に安くなっており，加えてLayer-2用のGbE switchはある程度のポート数以下であればそのポート単価がNICのそれを下回るほどの低価格となっている．しかしこのGbEは，HPCクラスタのネットワークに用いる場合，大規模化において問題が生じる．一般的にコストパフォーマンスのよいGbE switchは，24 port程度の比較的小規模なswitchである．クラスタの規模が24 node以下で比較的小さい場合はよいが，クラスタが大規模化した場合，いくつかの安価なswitchをtree構造等で相互に結合しなくてはいけない．クラスタの性能をnode数に合わせて向上させるためには，switch間結合のバンド幅をできる限り増強しなければ，その部分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．node-switch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，すなわちEthernetであれば10 Gigabit Ethernet (10GbE) を用いるべきである．し

† 筑波大学大学院 システム情報工学研究科 コンピュータサイエンス専攻
Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba
†† 筑波大学 第三学群 情報学類
College of Information Science, Third Cluster of College, University of Tsukuba

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為に GbE を利用している為，それを使用することは意味が無い．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 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向かなかった．

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，高価であるが高速な専用のネットワークである，Myrinet[3] や Infiniband[4] といった 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 結合を行う 2 つの switch の間に他の 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．従って，例えば 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造を作ろうとすると，switch のポート数のうち多数を LACP 用に利用してしまい，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[5] 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 技術を使用し switch 間の接続に柔軟性を持たせることが可能である．しかし，文献[5] 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はいるが，実現方法や拡張性等の点で様々な問題を含んでいる．本研究では，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 間に複数のパスを容易に持てないのは，それによってネットワーク中に 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 間の経路を任意の決定する規格が盛り込まれてないため，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することが出来ない．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[5)~8)]．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フラットなネットワークに分割する．

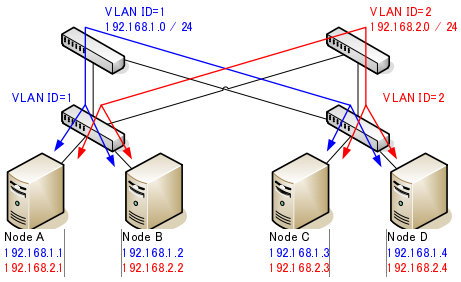

図 1 経路が異なるネットワーク

しかし文献[5] に基づく既存研究では，これらの環境を構築するめに，Linux 上の標準的な VLAN 実装を用いていた[2]．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 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くてはいけない．このような環境を用いて，例えば 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は，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 を利用しなくてはならず，MPI のすべての node で設定ファイルを共有できず，管理は複雑になる．加えて，クラスタ規模が大きくなり，使用する VLAN ID が増大し，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．従って，この従来手法を大規模 HPC クラスタに本格的に用いるのは難しい．

2.2 提 案 手 法

既存手法の問題な点は，現在の VLAN ID の制御が IP におけるサブネット単位での大雑把な制御になり，送受信先に応じた細かい制御ができないためである．

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 の実装は，MAC アドレスを基にした VLAN ID を決定する手段を持っていない．そのため，我々は MAC アドレスを基に VLAN ID を決定するネットワークデバイスドライバを用意する．ルーティングの為に 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

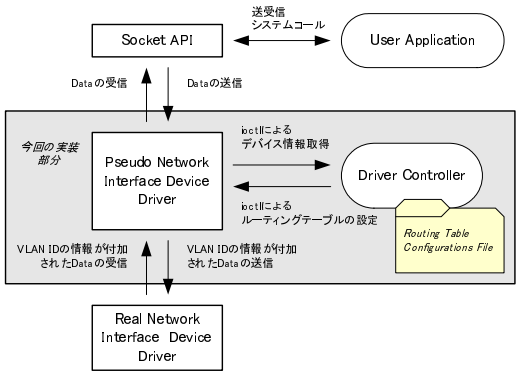

図 2 本実装のドライバと OS 等との関係

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と，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．既存手法でユーザは，利用する VLAN ID ごとにサブネットを選択するという作業が必要であったが，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，管理が楽になる．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無く使える．また，現在 PC クラスタで多く用いられている MPICH や LAM といった MPI ライブラリも TCP/IP を使用しているので，これらに変更を加えずに使えることは大きなメリットになる．

3. 実　　装

我々はこのような環境を実現するデバイスドライバを，Linux で提供されている VLAN ドライバを基に実装した．そのデバイスドライバの実装を図 2 に示す．

3.1 デバイスドライバの役割

用意したデバイスドライバは，socket API と送受信を行う本来のネットワークデバイスドライバの間に位置する．本実装のドライバは，socket API からの要求を本来のネットワークデバイスに変わって処理し，我々の意図する処理を加えた上で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について解析し，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

- IEEE802.1q のバケットを受信した時に，このデバイスドライバへ渡すように指示する．
- 受信データから 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．MAC アドレスは，socket API からネットワークデバイスドライバに送信データが渡される際に設定されている．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

ルーティングテーブルを決定するための，ルーティング方法について述べる．本ドライバは，送信元によって VLAN ID を決定する Source Routing 法，送信先に応じて VLAN ID を決定する Destination Routing 法，そして，自身と送信先を比較した上で VLAN ID を決定する Source-Destination Routing 法の 3 種類の方法が使用できる．しかし，switch を介在した通信の場合，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 L2-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 毎に独立に行われ，これが異なるとそれ以前の学習結果が反映されない．例えば，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 はいつまでたっても相手 node の MAC アドレスとそこに到達するためのポートの関係を学習できない．switch では MAC アドレスを学習できない場合，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．従って，Source Routing, Destination Routing は共に，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．Source-Destination Routing を採用し，必ず node 間で決められた同一の VLAN ID を利用することが必要である．

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アプリケーションから *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生成した上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，すべての node における通信の組合せによる VLAN ID 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 $O(n^2)$ となってしまうため，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

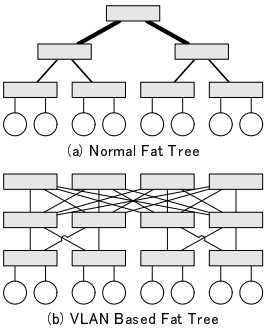

図 3 一般的な Fat Tree の概念図と，VLAN ルーティング法で可能になる Fat Tree 構成の VBFT(VLAN Based Fat Tree)

表 1 評価環境

CPU	AMD Athlon MP Processor 1800+ 8 node
Memory	1.5 Gigabytes
NIC	Intel PRO/1000MT (PCI-X 64bit/66MHz)
OS	Linux Kernel 2.6.11
MPI	LAM ver.7.1.1
Compiler	GCC ver.4.0
Switch	DELL PowerConnnect 5224 Gigabit Ethernet 24 Port

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，もしくは不明な場合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サイズは $O(n)$ となり，設定が簡便になる．

3.3 ネットワークトポロジ

既存研究[5] では，VLAN ルーティング法を用いることで様々なネットワークを実現できる事を示している．現状では，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 構成を念頭に入れる．VLAN ルーティング法で Fat Tree 環境を実現するために，VBFT(VLAN Based Fat Tree)☆が提案・評価されている[5]．今回これを選択した (図 3).

4. 評　　価

ここでは実装したデバイスドライバの性能評価を行う．評価環境は表 1 に示すとおりで，比較するネットワークは図 4 に示す 3 種類である． これらの環境を

☆ このような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った．

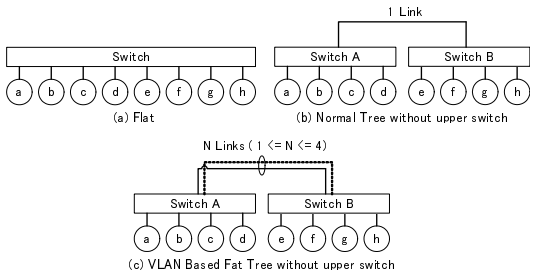

図 4 実験に使用した三種類のネットワーク
- **(a) Flat**　単独の switch にすべての node を接続
- **(b) Tree**　switch 間を 1 本の link で接続し，通常の Ethernet ドライバを利用
- **(c) VBFT**　switch 間に n 本の link で接続し，開発したネットワークドライバを利用
- *　上位 switch の接続は省略し，switch 間を直接接続
- *　実装ドライバを用いる場合のルーティングテーブルの優先度付けは，node 番号が若い順に優先度が高いことにし，それぞれに割り当てられる VLAN ID は，1-n 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．

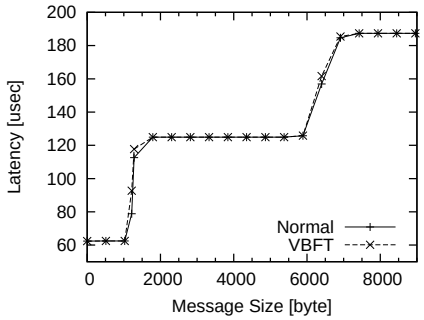

図 5 UDP/IP を用いた場合の片方向通信遅延時間

もちいて，通信遅延時間, スループット, バイセクションバンド幅, NAS Parallel Benchmarks について評価を行った．以後，3 つのネットワーク構成を図 4 に基づき，Flat, Tree, VBFT と表現する．

4.1 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 を決定したのちに，本来のデバイスドライバで送信するため，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 と VBFT において，メッセージサイズを変えながら，node a-b 間で 1000 回の ping-pong を行ったときの平均の片道時間を計測した．結果を図 5 に示す．

結果のように，使用するドライバによる性能差は無い．VLAN に伴う大きな処理は，送受信データへの tag 操作である．しかし，今回用いたような 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あり，この機

表 2 Iperf を用いた 2 点間における最大スループット

環境	Throughput
(a), node-a & node-b	898 Mbps
(a), node-a & node-b, (ドライバ使用)	896 Mbps
(b), node-a & node-e	898 Mbps
(c), node-a & node-e, (ドライバ使用)	896 Mbps

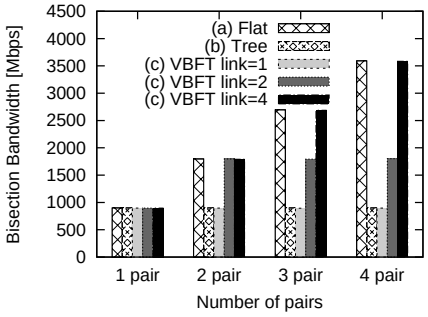

図 6 バイセクションバンド幅

能を利用することでオーバヘッドは小さくなる．このような機能が無い NIC を用いた場合，通信遅延時間は大きくなると思われる．

4.2 スループット

それぞれの環境において，スループットを計測する．計測には Iperf ver.2.0.2 を用いた．Iperf で GbE の性能を最大限に引き出すために TCP Window Size を 128Kbyte とした．参考までに Flat に接続した場合において，実装したドライバを用いた評価も行った．結果を表 2 に示す．

結果のように，スループットは標準的な Ethernet のデバイスドライバと，今回実装したドライバの性能は大きな差がない．本ドライバの実装で 2Mbps 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 (4byte) 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の為ではない．

4.3 バイセクションバンド幅

これまでの評価は，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を確認するものだった．ここで，実装したドライバが我々の意図したとおり実装され，それによって性能が向上するかどうか評価する．図 4 のネットワーク構成において，switch A に接続された node a-d から，switch B に接続された node e-h に向かってパケットを一斉送信し，そのときのバンド幅を計測する．受信側で観測されたバンド幅の合計がバイセクションバンド幅になる．この評価には MPI を用いた．結果を図 6 に示す．

結果のように，switch 間の接続が 1 link しか用意できない Tree では，バイセクションバンド幅が最大 1Gbps 程度に制限される．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 数によってバイセクションバンド幅が順調に増加し，node の台数に対して十分な link 数が用意された場合は，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，我々の開発したドライバが有効であることを示している．

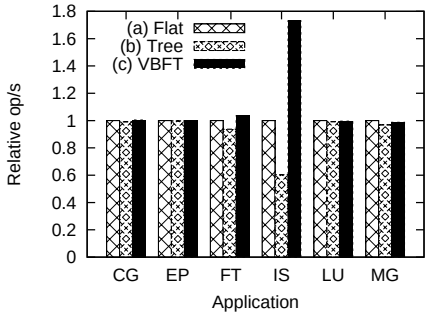

図 7 NAS Parallel Benchmarks 結果

4.4 NAS Parallel Benchmarks

最後に，NAS Parrallel Benchmarks(NPB) を評価する．本評価には，NPB ver.3.2, CLASS=B, PROCS=8 を用いた．VBFT においては上位リンクは 4 link 使用した．評価するアプリケーションは，CG, EP, FT, IS, LU, MG である．評価結果を図 7 に示す．

グラフには，Flat なネットワークを 1 とした場合の相対性能を示している．まず，全体的に見て VBFT の性能が Tree に比べ良くなっているが，その差は僅かであり，図 6 に示したバイセクションバンド幅の増大の具合に比べ小さい．これは，今回の実験では node 数の制約により，ベンチマークで要求される switch 間通信容量がそれほど大きくなく，Tree に対する VBFT のアドバンテージがそれほど大きくなかったためと考えられる．この点については今後実験クラスタの規模を大きくし，実証していく必要がある．いくつかの結果で，理論的には最大性能を示すはずである Flat が，VBFT より性能が悪い場合がある．本評価結果は現在解析不足であり，今後これらの原因を究明していく予定である．しかし，VBFT が通常の Tree のネットワークと比較して性能が良い傾向を確認できた．

5. 今後の計画

MPI を用いたベンチマークによる再評価

今回評価した NPB では，結果にいくつかの疑問が生じている．“本ドライバを用いる場合” と “通常のネットワークデバイスドライバを利用する場合” では，通信のタイミングが変わり，計算性能に影響を及ぼしている可能性がある．このことから，より詳しく NPB 内部の通信パターンを検証し，それに適したルーティングテーブルの設定を行った上で再評価したい．

また，現在のルーティングテーブルの設定が，本ベンチマークや実アプリケーションに適しているとは限

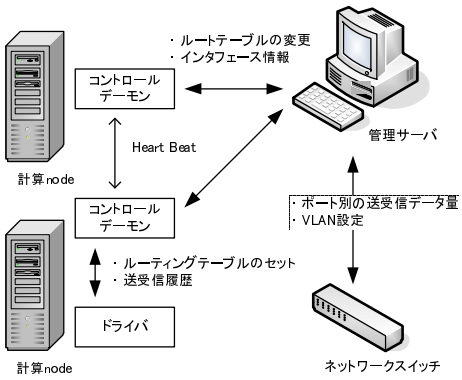

図 8 VLAN 環境を統合管理するアプリケーションのイメージ図

らない．今後はルーティングテーブルの評価も行う．

VLAN の環境を管理するシステムの開発

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．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って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．そのため，ルーティングテーブルを一括的に管理し，動的に変更することが可能なシステムに拡張する (図 8)．ルーティングテーブルの情報は一括してサーバが保持し，各 node がデバイスドライバの初期化時にルーティングテーブルをサーバから受取る仕様にする．ユーザが通信パターンによってルーティングテーブルを変更したいときは，サーバ上のテーブルを変更するだけで，各 node に自動的に再設定することを可能にさせたい．また管理サーバが定期的にネットワーク switch に対して SNMP などを通じてトラフィックを定期的に検査することで，通信に偏りがある場合は，動的にルーティングテーブルを変更するようなことも考慮に入れたい．それら加えて，各 node 上で実行される管理デーモンが，他の node とハートビートなどを行い，switch 間接続の不具合を検知させ，ルーティングテーブルの変更を管理サーバに要求するような実装も考えている．このように VLAN を用いることで複雑になるネットワークを管理するシステムを実装していく．

6. ま と め

本研究では，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増大し，クラスタシステム全体の性能向上が見込むことができる．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ったが，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ったため，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 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・提言を頂いた CREST「メガスケールクラスタ研究チーム」のメンバーに深く感謝します．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) Myrinet. *http://www.myri.com/myrinet/*.
4) Infiniband. *http://www.infinibandta.org/*.
5) 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, May 2004.
6) Tomohiro Kudoh, Hiroshi Tezuka, Motohiko Matsuda Yuetsu Kodama, Osamu Tatebe, Satoshi Sekiguchi. Vlan-based Routing: Multi-path L2 Ethernet Network for HPC Clusters. *Proceedings of 2004 IEEE International Conference on Cluster Computing (Cluster'2004)*, 2004.
7) 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.
8) Tomohiro Otsuka, Michihiro Koibuchi Akiya Jouraku, Hideharu Amano. Vlan-based minimal paths in pc cluster with ethernet on mesh and torus. *in Proceedings of the International Conference on Parallel Processing (ICPP'05)*, pp.567-576, 2005.