情報処理学会論文誌 Vol.55 No.1 484–493 (Jan. 2014)

迂回路付き完全網PCクラスタの試作と評価

福永 隆文[1,a)]

受付日 2013年3月14日, 採録日 2013年10月9日

概要：マルチコア・プロセッサの登場により各 PC の計算能力は飛躍的に向上し，PC クラスタのノードとしてマルチコア PC が幅広く利用されている．並列処理においてマルチコア PC を用いれば 1 台のノードで実行できるプロセス数を向上させることができるが，それにともない各ノード間の通信量は増大することになる．安価な Ethernet を用いたクラスタ環境では，ますます通信ボトルネックが生じやすくなる．本論文ではそのボトルネック緩和のためマルチポート Ethernet NIC を用いて完全網 PC クラスタを試作し，並列性能を評価した．本システムは直接接続された経路に加え迂回路を利用した通信を行うことでノード間バンド幅を向上させることができ，直接経路利用時にはスイッチを経由しないためレイテンシを向上させることができる．NPB ベンチマークを用いた評価では FT，LU，MG，CG，BT，SP において IEEE802.3ad，ラウンドロビン分散を上回る結果を示した．

キーワード：PC クラスタ，IEEE802.3ad，マルチパスコミュニケーション，Bonding テクニック

Fully Connected PC Cluster with Indirect Routes

Takafumi Fukunaga[1,a)]

Received: March 14, 2013, Accepted: October 9, 2013

Abstract: Multi-core processor PCs are widely used as cluster nodes for its high cost-effectiveness. The number of processes of being able to run concurrently on a node increases in direct relation to the number of cores. However, the communication bottleneck appears easily due to increasing traffic. This paper proposes the fully connected PC cluster with indirect routes. This system improves the bandwidth by using multi paths concurrently and latency when using direct routes. The results of FT, LU, MG, CG, BT, and SP in NPB benchmarks on proposed system are better than those on IEEE802.3ad and round-robin systems.

Keywords: PC cluster, IEEE802.3ad, multi-path communication, Bonding technique

1. はじめに

近年，マルチコア・プロセッサの低価格化にともない，PC クラスタの各ノードが実装するコア数はますます増加している．この飛躍的な計算能力の向上にともなって増加するノード間通信量を処理できるネットワークの構築が問題となる．クラスタ向け高速ネットワークとして Myrinet [1], Infiniband [2], Quadrics Network（QsNet）[3], RHiNET [4] などがある．それらはハードの性能を引き出すために専用のプロトコルを用いている．たとえば Myrinet の PM/Myrinet [5], InfiniBand の PM/InfiniBand [6]，QsNet の Elan3lib [7]，RHiNET の PM/RHiNET [8] があげられる．これらの高速ネットワークは専用ライブラリでハードの性能を引き出すことにより Ethernet に比べて遅延が大きく改善している．しかしながら，Ethernet は長年広く使われており，ビジネス向けとしてもこれまで多くの実績がある．現在，各ベンダからデータセンタ向け 10 G Ethernet 製品が発売され，40 GbE，100 GbE も開発が進むなど，Ethernet の普及はこれからも期待される．今回，広く用いられている Ethernet を利用した PC クラスタに焦点を当て性能向上の研究を行った．

本論文では 1 台に実装できるコア数の飛躍的な増加とマルチポート Ethernet NIC の登場により現実味を帯びて

[1] 熊本県立技術短期大学校
Kumamoto Prefectural College of Technology, Kikuchi, Kumamoto 869–1102, Japan
[a)] t-fukunaga@kumamoto-pct.ac.jp

きた完全網 PC クラスタを試作・評価する．提案する完全網では直経路のみでなくすべての迂回路を用い，通信負荷分散を行う．提案方式は複数の Ethernet NIC を実装したノード上で NIC ドライバの修正とモジュールのロードだけで簡単に導入できる．低価格で広く用いられている Gigabit Ethernet（GbE）環境および 10 G Ethernet 環境での利用を目的としている．機器の制限で今回は Gigabit Ethernet 環境で構築したが，10 GbE 環境でも構築可能と考えている．提案方式はカーネルにはいっさいの修正を行わず，動的ロード・アンロードが可能なモジュールを組み込むだけで実現できる．複数の通信経路への通信負荷分散は TCP ストリーム単位で行われる．個々のストリームのバンド幅向上が目的ではなく，総ストリームの合計バンド幅の向上が目的となる．また，転送は NIC ドライバで行うためレイヤ 2 内での処理となる．上位ネットワークとの接続およびブロードキャスト用にスイッチを 1 台用いた．ブロードキャストはすべてのポートから出力することで実現可能であることを確認したが，スイッチを用いた方が効果的である．

2. 関連技術

提案方式と同様に既存の Ethernet 環境で複数の回線を用いて通信負荷分散を行う方法として IEEE802.3ad（LACP）がある．LACP 対応のスイッチを用いる必要がある．多くはスイッチ間のバンド幅向上および冗長化のため用いられるが，サーバに LACP 対応のドライバを組み込めばサーバ・スイッチ間で利用することもできる．各 Ethernet ポート（通信線）への分散は MAC アドレス，IP アドレスなどのヘッダ情報をハッシュキーとしたハッシュ計算で行われる．したがって通信環境ごとにその効果は違ったものとなる．複数の回線を利用したとしても均等分散は偶然に頼らざるをえず，利用されない回線も存在する．また，ハッシュを用いず，パケットをラウンドロビンで複数の回線に順次振り分けるラウンドロビン分散も知られている．ラウンドロビン分散は送信側で複数の回線に均等に分散することができる．しかしながら，各受信ポートへ到着するパケットは不連続であるため頻繁な out-of-order 処理が発生する．今回，これら関連技術とバンド幅，遅延，並列処理性能を比較した．

VLAN を用いて L2 スイッチ間に複数パスを設定し，Fat Tree，メッシュ，完全網などを構成する方式 [10] が提案されている．この方式はパス数に比例してスイッチ間バンド幅を向上させることができるが，ノードが利用できるバンド幅は 1 つの NIC の性能に制限される（ハイパークロスバ構成では 2 または 3）．今回の提案方式は任意の 2 ノード間トラフィックを完全網に接続しているすべての NIC に分散することができ，ノードが利用できるバンド幅が大きく向上する点が異なる．また，方式 [10] 同様，提案方式は容易にメッシュ，トーラスなど他のトポロジに対応可能と考えている．トーラスへの拡張方法については 6 章で述べる．

大規模クラスタ PACS-CS [11] 用のネットワークとして Ethernet によるによる 3 次元ハイパークロスバ網を実現し，ノード間の直接通信のほか，中継ノードを経由した間接通信を行う専用通信ライブラリ PM/Ethernet-HTB [12] が開発されている．この方式では独自プロトコルによりゼローコピー通信や alloc_skb 処理の高速化などの最適化を行っている．提案方式は専用ライブラリを使用しないため，既存のアプリケーションがそのまま利用でき，提案システムの導入を意識することなく開発を行うことができる．また，PACS-CS では 3 次元ハイパークロスバ網のために複数のスイッチを用いるが，提案方式はブロードキャスト以外の通信にスイッチを用いない点も異なる．

大規模システムのインターコネクトとして利用されている PERCS [13] や Dragonfly [14] も直経路と迂回路を切り替えて通信が可能である．これらはラウンドロビンやランダム方式による経路選択，中継ノードの指定，負荷が小さい経路の動的選択などの方法で複数通信路から経路を選択できる．また，経路選択にハードウェア処理が取り入れられているため高速である．主にスーパコンピュータ内の計算ノードのインターコネクトとして利用されている．提案システムは既存のパソコンを使って構築できる．

3. 提案方式の概要

迂回路付き完全網 PC クラスタの実現には 2 つの機能が必要となる．まずは各ストリームを直経路および迂回路のいずれに割り当てるかの判断を行う機能が必要となる．次に迂回路が選択された場合に中継点となるノードにて転送を行う機能が必要となる．この 2 つの機能に分けて説明する．

3.1 ポート割当て機能

図 1 はパケットに出力 Ethernet ポートを割り当てる流れである．Eport は Ethernet ポートを表す．パケット S および T はともにノード C 宛のパケットであり，それぞれ異なる高バンド幅ストリームに属すると仮定する．最初に，各パケットは後述する基準で高バンド幅ストリームに属するパケットであるかどうかの判定を受ける．高バンド幅でない場合，通信路をほとんど消費しないと判断し，遅延性能が良い直経路に割り当てる．なお，データを持たない ACK パケットは無条件に直経路に割り当てる．ACK の遅延を回避するためである．高バンド幅と判定されたパケットは直経路・迂回路のいずれかに割り当てられるが，以前割り当てられた情報は Output Eport ハッシュテーブルに格納されているので，ハッシュ関数で検索を行う．ヒットすれば示された番号の Ethernet ポートに割り当てられる．図ではパケット S（①）がすでに登録済み（②）なの

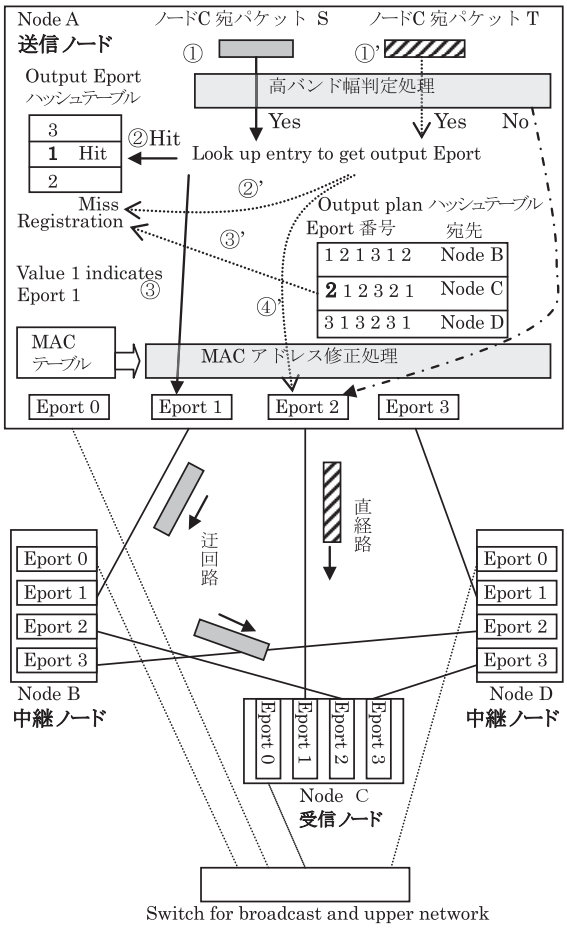


図 1 パケットへのポート割当て方法

Fig. 1 Method for allocating Ethernet ports to streams.

で，記憶された番号 “1” に従って Eport 1（③）に割り当てられている．一方，高バンド幅ストリームのパケット T（①’）は Output Eport ハッシュテーブルに登録されていない（②’）（高バンド幅と認定されてから最初のパケット）ため，Output plan ハッシュテーブルの宛先ノード C のエントリ内配列の最初の要素 “2” を取り出し，割り当てる Eport 番号（Eport 2）とする（④’）．割当て情報はパケット T に続く同一ストリームのパケットで利用するため Output Eport ハッシュテーブルに登録される（③’）．結果的にパケット S が属するストリームはノード B を経由する迂回路，パケット T が属するストリームは直経路で送られることになる．

なお，Output plan ハッシュテーブルに記述してある出力 Eport 番号配列の要素は 10 個あり，高バンド幅ストリームの発生順にラウンドロビンで割り当てる．このテーブルは事前に手動で用意するが，出力ポートの指定は通信経路の指定につながる．たとえば，図 1 でノード C への送信時，ノード A の出力ポート Eport 1，Eport 2，Eport 3 はノード B 経由迂回路，直経路，ノード D 経由迂回路に対応する．10 要素に含まれる直経路の割合を変えることで直経路の利用頻度と性能の関係を評価できる．ランダム関数

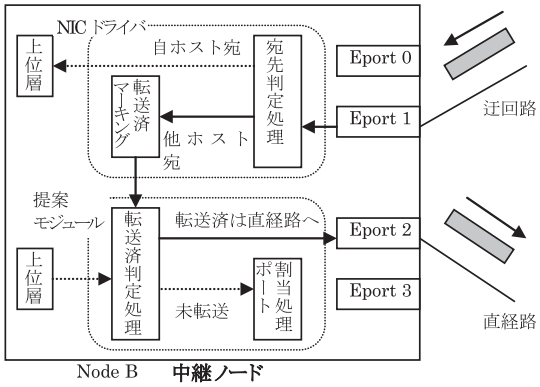


図 2 転送機能

Fig. 2 Transfer function.

を用いたテーブルを用いない動的な割当て比率の変更も可能であるが，予備実験の結果，短いスパンでは比率が正確に反映できず，直経路の出現間隔も均等ではないことが分かった．ベンチマークを用いた並列性能評価も静的なテーブルを用いた場合に及ばなかった．静的テーブルでは，割当て比率を正確に記述できるだけでなく，直経路の出現間隔をほぼ均等にできる．たとえば，割当て比率を 3 : 7 にしたい場合，記述式であれば直：迂回：迂回：直：迂回：迂回：直：迂回：迂回：迂回のようにほぼ均等に直経路を配置できる．

送信パケットの Ethernet ヘッダ内宛先 MAC アドレスは ARP の仕組みにより最終目的ノードの Eport 0 の MAC アドレスとなっている．そのため送信直前に宛先ノードまたは中継ノードの受信 Ethernet ポートに応じて宛先 MAC アドレスを書き換える仕組みを組み込んだ（MAC アドレス修正処理）．また，同時にヘッダ内送信元 MAC アドレスに割り当てた出力ポートのアドレスを書き込んだ．図の下方のスイッチは上位ネットワーク接続用である．

3.2 転送機能

当システムでは各ノードが受信したパケットは自ホスト宛とは限らない．他ホスト宛の中継パケットであれば速やかに最終目的ノード宛に転送する機能が必要となる．この転送時，中継ノード数を 1 に制限しているため，再度中継ノードに送信しない仕組みも必要となる．図 1 の中継ノードであるノード B を例として**図 2** に中継ノードでの処理を示す．実線矢印が転送時の流れとなる．中継ノードで受信されたパケットに対してヘッダ解析により，自ホスト宛か他ホスト宛かの判定が行われる（宛先判定処理）．自ホスト宛の場合は上位層に渡され，他ホスト宛の場合は転送が 1 度行われたことを示すマーキングが行われた後，送信関連関数をコールし，パケットを渡す．送信関連関数には提案モジュールが含まれる．提案モジュール内で転送済みマーキングの確認が行われ，転送済みの場合は直経路ポートへ割り当てられる．これにより転送は 1 回に限られる．

転送 1 回で受信側のすべてのポートを活用できるため 2 回の転送は必要ない．マーキングが施されていないパケットは前節のポート割当て処理が行われる．

4. 実装

迂回路付き完全網 PC クラスタは NIC ドライバの修正および Linux に標準で実装されている Bonding モジュールの修正を行い，Linux 上に実装した．Bonding モジュールは通信負荷分散と冗長化のために使われ，IEEE802.3ad（LACP）やラウンドロビン送信を実装している．前章と同様にポート割当て機能と転送機能に分けて説明する．

4.1 ポート割当て機能

Bonding モジュールはロードされると擬似インタフェースを作成する．擬似インタフェースはマスタと呼ばれ，実際のポートのインタフェースはスレーブと呼ばれる．見かけ上送信はマスタを使って送信されるが，実際は Bonding モジュールがプロトコルスタックからいったんパケットを受け取り，送信モードに従って選択したポートのドライバを呼び出し送信する．今回，Output plan テーブルに従い，ストリームを特定の迂回路または直経路に割り当てる仕組みをこのモジュール内のラウンドロビン送信関数を修正することで実現した．

Output plan ハッシュテーブルは宛先ノードごとにストリームに割り当てる Eport 番号を事前に計画するテーブルである．宛先ノードのマスタの MAC アドレスをハッシュのキーとしている．テーブルを**表 1** に示す．10 個の出力 Eport 番号配列に示された番号を出力ポートとして発生したストリームに順次（ラウンドロビン）割り当てていく．10 個の中に現れる直経路の数を変化させることにより，直経路対迂回路の利用比率を変えることができる．なお，この配列の 1 つ目の要素は直経路を示す役割も兼ねている．

いったん Output plan テーブルからストリームのパケットに対して特定の Eport が割り当てられると，その割当て情報は Output Eport ハッシュテーブルに記憶され，同じストリームの後続パケットの Eport 割当てに用いられる．ハッシュの計算には宛先および送信元 TCP ポート番号，宛先 MAC アドレス，宛先 IP アドレスを用いた．ストリームの終了を示す Fin パケットでテーブルの登録情報は削除されるため，古い情報が残ることはない．

表 1 Output plan ハッシュテーブル
Table 1 Output plan hash table.

	説明
出力 Eport 番号配列（要素数 10）	同一宛先へのストリームに対して，Eport 番号をラウンドロビンで割り当てる．ただし，1 つ目の要素は直経路の番号とする．
インデックス	上記配列内の次に利用する要素を指す．

各ストリームへの Eport 割当てのフローを**図 3** に示す．(1) は事前にクラスタ内ノードの MAC アドレスをハッシュキーとして Output plan ハッシュテーブルを作成しているので，エントリが存在しない宛先 MAC アドレスはブロードキャスト，他ネットワーク宛などになるのでスイッチへ送信を行う．

(2) においてデフォルトの経路として直経路を Output plan テーブルの出力 Eport 番号配列の 1 つ目の要素から取得する．

(3) の “IP データグラム全長 > 52” はデータ部を含むパケットを選択するためである．ACK のみの TCP データグラムは IP ヘッダが 20 バイト，TCP ヘッダが 32 バイト（タイムスタンプオプション 10 バイト，NOP(1)，NOP(1) 含む）のため 52 バイト長となる．データを含むパケットはこの値より大きい IP データグラム長を持つ．同じく (3) の “中継パケットではない” の判定には IP ヘッダの TTL を用いた．転送機能により中継パケットはあらかじめ TTL が初期値より減じられているため（TTL = TTL 初期値）が “中継パケットではない” の判定条件となる（次節，転送機能参照）．(3) の条件を 1 つでも満たさない場合はデフォルトで設定した直経路から出力することになる．

(6) の高バンド幅ストリームの判定には次式を用いる．

$$write\_seq - snd\_una > threshold \quad (1)$$

各変数は Linux カーネルの送信バッファで用いられるメンバである（**図 4** 参照）．式を満たす場合，高バンド幅ストリームと判定する．write_seq はアプリケーションから送信バッファに渡されたデータの最終番地を保持し，snd_una は受信側からの ACK を待っている先頭番地である．それ

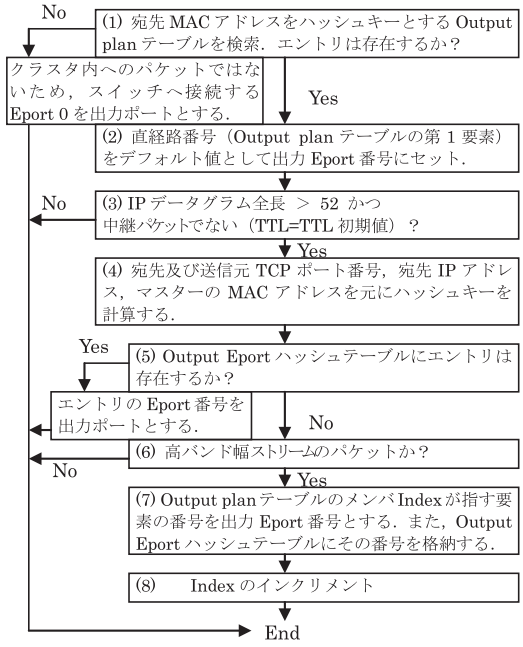


図 3 ストリームへの Eport（Ethernet port）割当てフロー
Fig. 3 Ethernet port allocation flow.

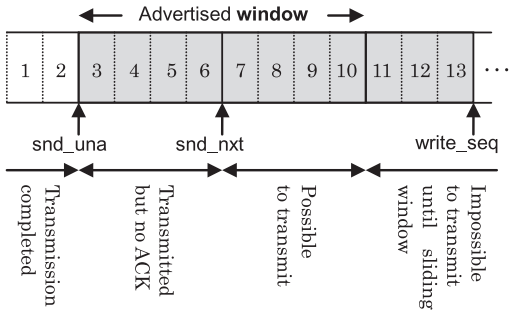


図 4 送信バッファ内メンバ
Fig. 4 Members in send buffer.

らのメンバの間にある snd_nxt はこれから送信するデータの位置を示している．したがって，snd_una から snd_nxt までは “送信したが，受信が確認できていないデータ量”，snd_nxt から write_seq までは “アプリケーションは送信関数を実行したがバッファ内にとどまり送信されていないデータ量” をそれぞれ表す．それらの和は，“アプリケーションは送信を実行したが受信側に届いたことが確認できていないデータ量” となる．この量が式 (1) の左辺で求められる．この値の大きさは “送信が迅速に完了していない程度” を示すが，今回，高バンド幅の指標として用いた．1 対 1 通信においては閾値を 7,000 とした場合約 250 Mbps 以上が判定式により選択された．今回，閾値は 7,000 とした．

送信する Eport が割り当てられた後は送信関連関数に引き渡す前に宛先および送信元 MAC アドレスを正しく変更する必要がある．宛先 MAC アドレスは割り当てられた出力 Eport と接続されている受信側 Eport のアドレスに書き換える必要がある．書き換える前の値は宛先ノードのマスタの MAC アドレスとなっている．宛先 MAC アドレスの書き換えには事前に用意した MAC テーブルを利用する．このテーブルは出力 Eport ごとに対向する（接続している）相手ノードの Eport の MAC アドレスが格納されている．よって，決定した出力 Eport 番号をインデックスとして MAC テーブルのエントリを取得し，受信 Eport の MAC アドレスを取得できる．送信元 MAC アドレスは出力 Eport の MAC アドレスに書き換える．書き換える前は自ノードのマスタの MAC アドレスになっている．

4.2 転送機能

転送機能は NIC ドライバに実装した．本環境のすべての NIC は Intel PRO/1000 ドライバを用いるので e1000_main.c を修正した．転送処理の対象はユニキャストでありながら自ノード宛ではないパケットとした．そのため下記条件を受信ノードで判定し，すべてを満たす宛先 IP アドレスを持つパケットを転送対象とした．なお，ネットマスクは 255.255.255.0 であり，転送はレイヤ 2 で行うため PC クラスタ全体は同一ネットワークとなる．

① 上位 3 バイトが自ノードの IP アドレスと同じである．
② 下位 1 バイトが自ノードの IP アドレスと異なる．
③ 下位 1 バイトが 255 でない．

上記の条件を満たし中継パケットと判定された Ethernet フレームの MAC アドレスは送信ノードで自ノードの受信 Eport の MAC アドレスに変更されているので本来の宛先 MAC アドレスに変更する必要がある．そこで ARP キャッシュを IP ヘッダ内宛先 IP アドレスで検索し，MAC アドレスを取得し書き換えた．この MAC アドレスは最終宛先ノードのマスタの MAC アドレスであるが，この値が本来の MAC アドレスである．また，送信元 MAC アドレスは自ノードのマスタの MAC アドレスに書き換えた．これらの書き換えにより送信関数では中継パケット，初めて送信するパケット間で MAC アドレスの扱いを区別する必要がなく，送信時のコードを簡略化できる．

中継パケットについては自ノード内の送信関数に渡すことになるが，送信関数内で中継パケットであることが認識されなければならない．中継数を 1 と制限しているため中継パケットは直経路に割り当てるからである．そのため，前述した条件で中継パケットと判定されたパケットの IP ヘッダ内 TTL を 1 デクリメントし，ポート割当て機能に中継パケットであることを知らせることとした．TTL は生存期間（ルーティング回数）を制限するために用いられ，ルーティング処理が行われるたびにデクリメントされる値である．今回の転送はレイヤ 2 で行われ，ルーティングは行わないため，デクリメントされず，値はつねに初期値（今回の環境では 64）のままである．今回，この変化しない TTL の値を中継パケットのサインとして利用した．最後に送信関数 dev_queue_xmit() 関数を呼び出し，パケットを引き渡す．後続のコードで今回修正を行った Bonding モジュールが実行されポート割当てが行われた後，送信される．

提案方式はブロードキャストを除きノード間のみの通信となるため，スイッチの制限を考慮せず NIC が対応する範囲内で Jumbo フレームを活用できる．Jumbo フレーム受信処理は MTU 1500 時に受信データを展開する領域にヘッダ（Ethernet，IP，TCP）のみ展開し，後続のデータは複数の page（今回 4,096 バイト）に保存する．そのため MTU が 1,500 の場合と 1,500 を超える場合は異なる関数が用いられるが提案方式は Jumbo フレーム処理コードにも実装されている．

5. 評価

この章ではバンド幅，遅延，並列処理性能を評価し，関連技術である IEEE802.3ad（LACP）およびラウンドロビン分散との比較を行う．完全網だけであれば各ノードのポートを異なるネットワークとし，ルーティングを利用することで実現できるため，この形態との比較も行った．測定環境を**表 2** に示す．PC クラスタは 8 台のノードから構

表 2 測定環境
Table 2 Experimental environment.

Hardware	Quad-Core Opteron 2.4GHz × 2, 16 Gbytes memory, Intel Pro/1000 PT Dual Port Sever Adapter (2 ports) × 2, Intel Pro/1000 PT Quad Port Server Adapter (4 ports), SUPERMICRO H8DAE-2 motherboard
Switch	NETGEAR GSM7248R (Layer 2, GbE × 48 ports, LACP support), NETGEAR GS748TP (Layer 2, GbE × 48 ports, LACP support), NETGEAR JGS516 (Layer 2, GbE × 16 ports, for broadcast)
OS	Linux 2.6.24

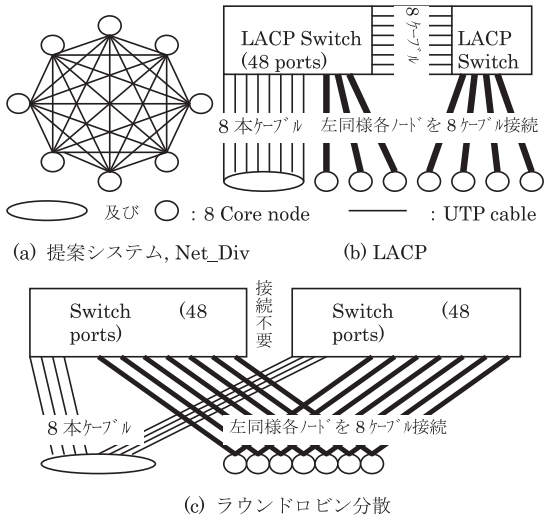


図 5 比較システム
Fig. 5 Evaluated systems.

成され，各ノードは Quad-Core を 2 個搭載しているため 8 コアの実装となる．Gigabit Ethernet ポートは 2 ポートアダプタを 2 枚，4 ポートアダプタを 1 枚用いるため合計 8 ポートとなる．Dual port adapter 比較対象システムを下記に示す．

① 提案システム．直経路割合 1，3，5，7，10 割について評価する（D1，D3，D5，D7，D10 と表記）
② 各 Eport をネットワーク分割し，ルーティングを利用して完全網を実現した既存システム（Net_Div と表記）
③ Bonding モジュールの IEEE802.3ad（LACP）機能を利用した既存システム（LACP と表記）
④ Bonding モジュールのラウンドロビン分散機能を利用した既存システム（Round と表記）

図 5 に各システムの接続形態を示す．提案システムの上位ネットワーク接続用スイッチは省略している．提案システムはスイッチに接続したポートを含むと合計 8 つの Ethernet ポートを使用するため，他手法も同数のポートを利用するシステムとした．LACP ではスイッチ間も 8 本のケーブルを用い，LACP で接続している．分散のためのハッシュキーとして MAC アドレスを用いた．今回用いた GS748TP が MAC アドレスハッシュキーにしか対応していないためである．LACP は比較的高機能なスイッチに実装されているがハッシュキーの変更が可能なスイッチはさらに限定される．ハッシュキーの変更による性能への影響については次回の課題としたい．LACP と Round で用いる L2 Gigabit スイッチ 2 台は機器の制限により型番が異なる GSM7248R および GS748TP を使用したが，これらのスイッチで接続した 2 ノード間バンド幅はともに 941 Mbps であり，ハードウェアの限界能力に近い性能を持つ．遅延はそれぞれ 37.4 μ 秒，35.5 μ 秒であり性能差は 5%にとどまるため，型番の違うスイッチを用いることによる性能評価への影響は少ないと考える．ともに Netperf-1.2.7 を用いて測定した．ブロードキャスト用には NETGEAR JGS516 ノンインテリジェントスイッチを利用したが，ARP などで用いるだけであり当スイッチの性能は全体の性能にほとんど影響を与えない．なお，MTU はデフォルト値の 1,500 を用いた．MTU が 1,500 を超える Jumbo フレームを提案方式，LACP に適用した結果についても並列性能の節で述べる．

5.1 バンド幅

1 方向，双方向，1 対 N，N 対 1，N 対 N のバンド幅を Netperf-1.2.7 を用いて測定した．マルチコアノードでは複数プロセスを同時実行するため，測定ツールのプロセス数を 1，2，4，6，8 と変化させて測定した．1p，2p，4p，6p，8p と表記する．測定結果はそれらの合計値である．**図 6** と**図 7** に 1 方向バンド幅，双方向バンド幅を示す．プロセス数が大きくなるほど D1，D3，D5 がこの順に良い性能を示しているが，迂回路を利用できるためである．直経路を含め 7 つある通信経路を均等に利用できる D1 が最も良い性能を示した．Round はプロセス数が 1（1p）の場合に 1 方向バンド幅が 2 Gbps 近くを示しているが，単一ストリームの各パケットを複数の経路に分散（パケットごと分散）するため，ストリームのバンド幅が向上する．プロセス数が増加しても合計バンド幅が向上しない理由は各ノードで利用する 8 つのポートがすべて同じ MAC アドレスを持つためである．スイッチは最近その MAC アドレスを学習したポートからのみ出力するのでスイッチと受信側ノード間ではパケット分散ができない．そのためプロセス数が増えてもバンド幅は向上しない．

図 8，**図 9**，**図 10** に 1 対 N，N 対 1，N 対 N 通信バンド幅を示す．1 対 N 通信では Round が 8 ポート GbE のハードウェア性能（8 Gbps）に近い結果を引き出している．前述したように単一の受信ノードの複数ポートへのパケット分散ができないが，1 対 N 通信では受信ノードが複数になるため自然に受信側分散が行われ，送信側の分散のみが問題となるためである．送信側でのパケット分散は完全に均等分散となるため高い性能を示した．Client-Server システムのサーバ側分散に向いているといえる．ただ，グラフ上は Round に劣っている Net_Div，提案方式も 1 対 N 通

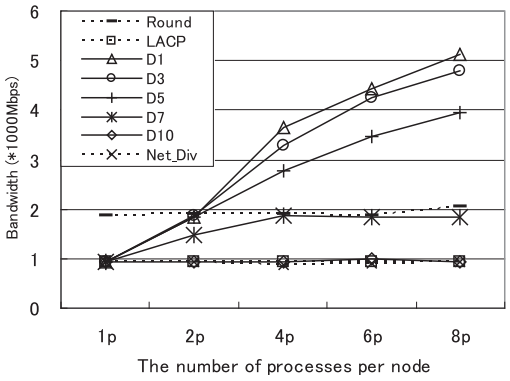


図 6 片方向バンド幅
Fig. 6 Unidirectional bandwidth.

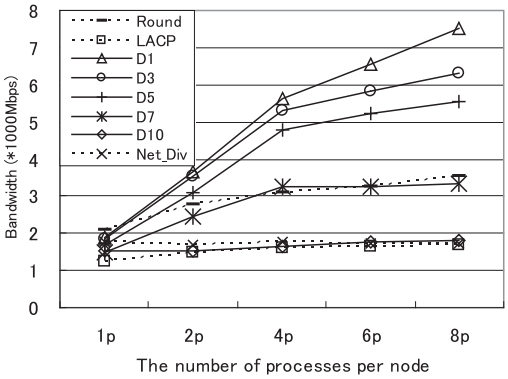


図 7 双方向バンド幅
Fig. 7 Bidirectional bandwidth.

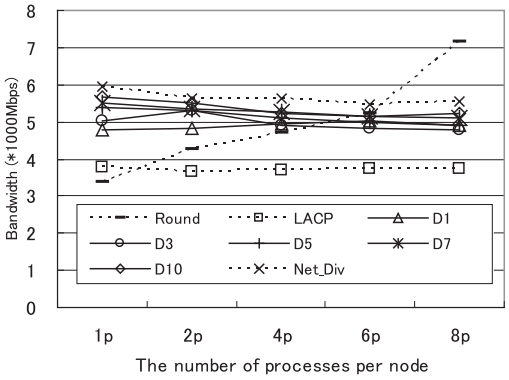


図 8 1 対 N バンド幅
Fig. 8 One-to-many bandwidth.

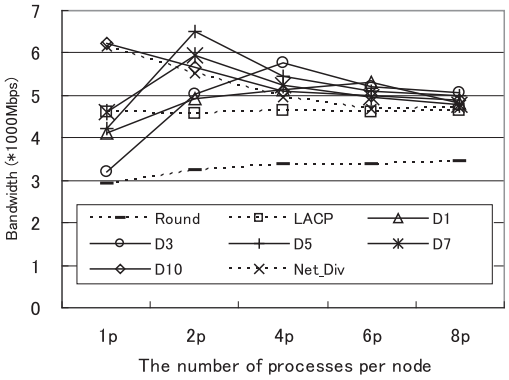


図 9 N 対 1 バンド幅
Fig. 9 Many-to-one bandwidth.

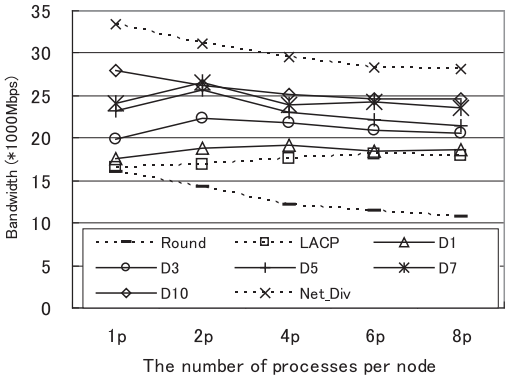


図 10 N 対 N バンド幅
Fig. 10 Many-to-many bandwidth.

信では 7 ポートのハードウェア性能（7 Gbps）の 75%以上の性能を達成した．Round は頻繁な Out-of-order を引き起こすため 1 対 N 8p での CPU 使用率は 48%に達した．Net_Div，提案方式が 30%以下であるので速度向上を考慮しても CPU 負荷が高いことが分かる．

N 対 1 通信では同じ動き（直経路のみを用いる）をする Net_Div と D10 が良い性能を示した．受信側となる 1 つのノードの 7 つのポートを送信側の 7 つのノードが 1 つずつ利用するためほぼ均等に 7 ポートへ負荷分散されるためである．プロセス数が大きくなると 1 つの経路に複数のストリームが混在するため，競合，オーバヘッドが大きくなりわずかに性能が下がった．Round は受信側の負荷分散が行われないため他より性能が大きく劣る結果となった．

N 対 N 通信ではルーティングを利用した完全網が良い結果を示した．同じ動作をする提案方式の D10 の 1.19 倍（1p 時）となった．ポート割当て機能，転送機能（転送はしないが判定等コード実行）のためのコード追加だけでなく，Bonding モジュール自身のオーバヘッドが原因と考えられる．CPU 使用率は Net_Div の 21%に対し，D10 は 28%であった．

LACP は全体的に低い結果となったが，これはどのポートを使うのかの判定に MAC アドレスや IP アドレスなどのネットワークパラメータを用いるため，均等な分散は偶然に頼らざるをえないという事情による．実際，今回のスイッチ間 LACP では 8 本のケーブルのうち 4 本しか使われていなかった．また，ノード接続用ケーブル 8 本のうち 4 本しか使わないノードもあった．

5.2 遅延

図 11 は遅延を Netperf-1.2.7 を用いて測定した結果である．LACP（1 SW）と LACP（2 SW）は同じスイッチに接続されたノード間および異なるスイッチに接続されたノード間の遅延を表す．後者はスイッチ間の遅延が含まれ

る．提案方式は直経路のみを用いるD10の結果である．提案方式D10とNet_Divを用いた完全網がスイッチの遅延を回避できるため最も良い性能を示した．迂回路の遅延を中継数を増加させながら計測したところ，77.1 μ秒（中継数1），105.5 μ秒（中継数2），132.0 μ秒（中継数3）となった．中継数に比例して遅延が増加している．提案方式では中継数を1に制限した．中継数1で受信側ノードの7つのEportをすべて活用できるため，制限のデメリットはない．

5.3 並列性能

提案方式はMPICHに限らずすべてのTCP/IP通信に利用できるが，並列処理の測定ツールとしてはMPICHを用いたベンチマークNAS Parallel Processing Benchmarks（NPB）が広く用いられるためNPB3.3で評価した．NPBにはFT，LU，MG，CG，IS，BT，SP，EPという8つのベンチマークがあるが，EPはあまり通信を行わないためEPを除く7つのベンチマークで評価した．**図 12** にシリアル実行に対する速度向上度を示す．1ノードでの実行プロセス数1p〜8pのうち，最も性能が向上した場合を表示した．横軸の表記はベンチマーク名＋プロセス数となっている．比較のため，1ポートでの性能を併記した．なお，1ポート時はスイッチ1台を用いたため，スイッチ間ボトルネックは存在しない．ISを除く6つのベンチマークでD3が良い性能を示した．特にLU，FTにて他より優れた性能を示した．ISは全対全通信を繰り返し，全処理時間の9割以上が通信に費やされるベンチマークであり，低遅延を必要とするためNet_Divがわずかに良い性能を示した．

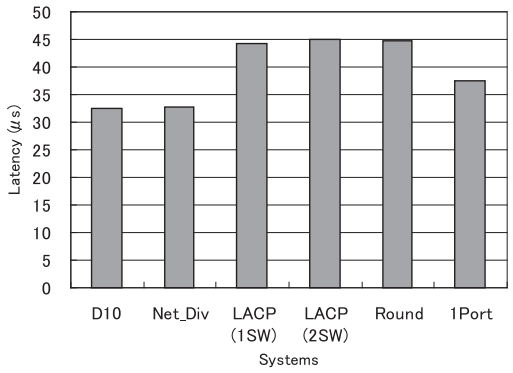


図 11 遅延
Fig. 11 Latency.

LUは8ノードの各ノードで1プロセス（1p）を実行した場合，すべてのシステムにおいてほぼ同程度の性能を示し，いずれも割り当てられたコアの使用率は90%に達した．また，この場合トータル8プロセスとなるが，外部通信をともなわない1ノード8コアで8プロセスを実行した場合もほぼ同じ並列性能を示した．これらの結果からLUは8プロセス分割時点ではCPUボトルネックであることが分かる．また，通信に用いるストリーム数は各ノード8プロセス実行時でも10程度であり相対的に少なく，その分各ストリームが送受信するパケット数は大きい．4プロセス（4p），トータル32プロセスの時点でボトルネックが通信にシフトし，1対1バンド幅性能が勝る提案方式が良い性能を示した．各ノードで8プロセス実行する8pよりも4プロセス実行する4pが良い性能を示したのは，8pの送受信数は4pの1.7倍に増加したためである．FTは単位時間あたりの通信量が最も大きく，またストリーム数も8pで約350と最も多い．多くのストリームを複数の回線に分散できる提案方式が良い性能を示した．SP，BTは最も送受信量が多く，通信負荷が高いため提案方式が良い性能を示した．CGはバタフライ通信を行うが，バタフライ通信は複数のステージに分けて通信を行いAll-reduce通信を行う[9]．プロセス数が増加すると通信ステージ数が増加するため4pが8pより良い結果を示した．

ストリームのパケットを複数の経路へ分散させるRoundは単一ストリームのバンド幅を向上できるため，大量のパケットを少ないストリーム数で通信を行うLUで効果を示したが提案方式には及ばなかった．Roundは同一ストリームのパケットが受信側の複数のEthernetポートに不規則に分散して到達するため，out-of-orderを送信側に知らせるSACKフレームが受信側から大量に送られる．FTで全パケットに占めるSACK割合は29.5%に達した．LACPは

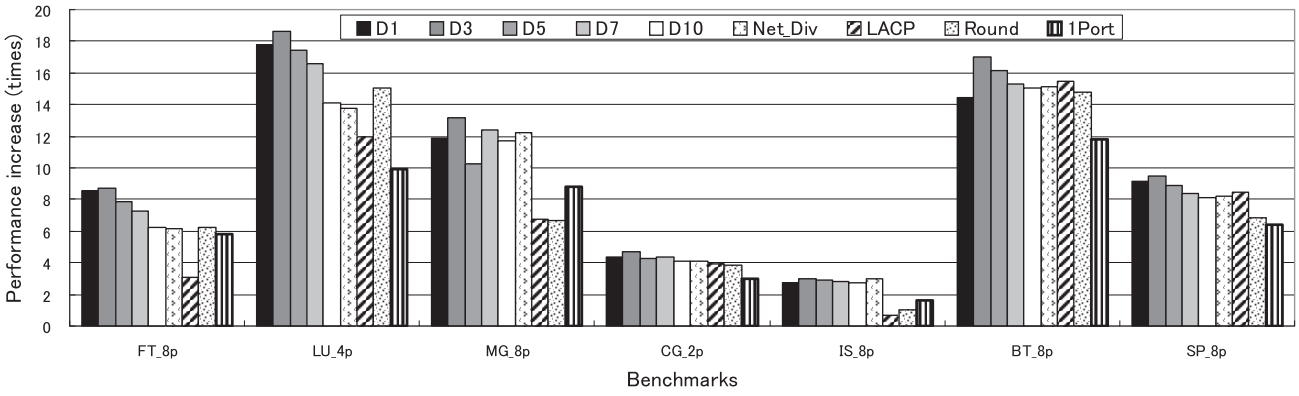


図 12 並列処理性能
Fig. 12 Parallel processing performance evaluation using NPB.

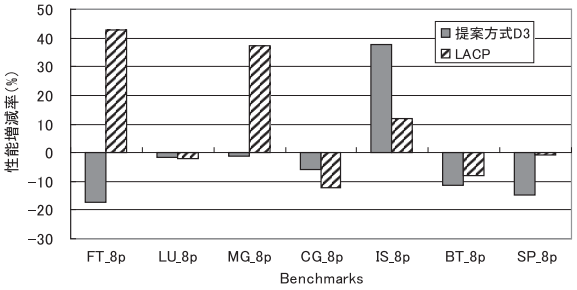


図 13 Jumbo フレーム適用時の性能増減率

Fig. 13 Rate of performance increase when applying Jumbo frame.

0.35%，提案方式と Net_Dev はほとんど発生しなかった．この大量の SACK の発生は通信リソースと CPU パワーを浪費し，性能へ悪影響を及ぼす．

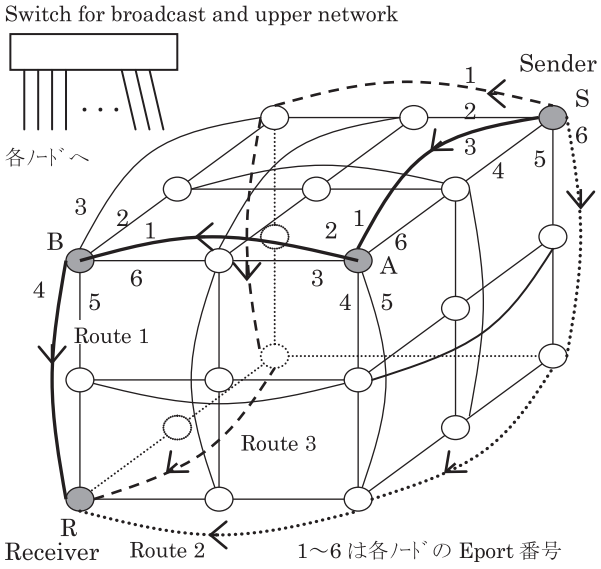
LACP は各ノードで 8 本のすべての回線を用いることができなかったこと，スイッチ間も 4 本のケーブルしか用いることができずスイッチ間ボトルネックを起こしたことにより全体的に低い性能となった．特に単位時間あたりの通信量が多い FT，IS では 1 ポートよりも遅い結果となった．

図 13 に MTU を 9,216 とし，Jumbo フレームを利用した場合の提案方式，LACP の速度向上度を示す．プロセス数 64 で比較した．9,216 はスイッチ，NIC の上限値である．パケット通信量が大きく，処理時間の 9 割以上が通信に費やされている IS はともに性能が向上した．LACP では FT，MG で大きな効果が確認できるが，Jumbo フレーム利用によるバンド幅向上（片方向通信で 986.9 Mbps）がスイッチ間ボトルネックや同一スイッチ内ノード間通信を改善できたためと考えられる．しかし，デフォルト（MTU 1,500）で提案方式に比べ 35%（FT），50%（MG）の性能にとどまっており，Jumbo フレームを利用しても提案方式には及ばなかった．その他のベンチマークの性能低下の原因として，Jumbo フレームの長い帯域占有時間がパケット間の競合による遅延を引き起こしたことが考えられる．

6. 他トポロジへの拡張

提案方式はわずかなコード修正と Output Plan ハッシュテーブルの工夫でメッシュやトーラスなど他のトポロジにも対応が可能である．この章では 3 次元トーラスを用いて，拡張方針を述べる．**図 14** はノード S からノード R へのパケット転送ルート例およびテーブル例を示している．完全網ではテーブルエントリ内の出力 Eport 番号配列の先頭要素は直経路を示したが，トーラスでは直経路または最小中継ノード数を持つ経路の 1 つ（以後，最短経路）を示すことになる．

送信ノードでは完全網の場合と同じく，通信負荷分散のためすべての Eport を通信に利用するが，直経路または最短経路を優先する．ノード S の R 用エントリでは最短経路 Route 1，Route 2，Route 3 への出力 Eport 番号 1，3，6



Route 1 が採用されるケースのテーブル例
（配列の最初の要素は直経路または最短経路）
- ノード S 内 Output Plan Hash Table の宛先ノード R 用エントリ
  Eport 番号 1,**3**,6,2,4,5,1,3,6,1・・・ index 1
- ノード A 内 Output Plan Hash Table の宛先ノード R 用エントリ
  Eport 番号 **2**,5,3,4,2,5,1,6,2,5・・・
- ノード B 内 Output Plan Hash Table の宛先ノード R 用エントリ
  Eport 番号 **4**,5,6,4,2,3,4,5,6,4・・・
  （1 はノード A 間でループとなるため記述しない）

図 14 提案方式の 3 次元トーラスへの拡張

Fig. 14 Proposal system extension to three-dimensional torus.

を優先的に利用している．優先利用する Eport が複数存在するためテーブルエントリ数を 10 から拡張する必要がある．Eport 番号 2，4，5 が選択された場合，最小でも中継ノードが 3 となってしまい，遅延は増加するが，トータルバンド幅は向上する．図中のノード S は index に従い Eport 番号 3 を送信に用いている．中継ノード A では転送済みマーキング（TTL < 64：4.2 節参照）が行われているため index に基づく経路選択は行わず，R 用エントリ内最初の要素である直経路または最短経路を用いて送信する．図では 2 が最短経路として記述してあるため Eport 2 から出力する．この動作は完全網が 2 度目の中継ノードへの転送を行わず，直経路を用いる動作と基本的に同じである．中継ノード B の動作は中継ノード A と同様であり，その結果パケットは Eport 4 を通じて目的ノード R に到達する．

ただし，テーブルの作成時，ループが発生しないように注意しなければならない．ノード B の R 用エントリの出力 Eport 番号配列に 1（Eport 1）を記述すると B が送信ノード，R が受信ノードとなった場合に無駄なループが発生する．B で index に従い Eport 1 が選択された場合 A に送信されるが，A の最初の要素が 2 のため最短ルートとして Eport 2 から出力し B に戻りループとなる．B は最初の要素 4 を次に用いるため Eport 4 から出力し無限ループは

発生しないが，ループを避けるため配列要素に 1 は含めない．他のノードでも同様である．また，最短経路（最初の要素）に記述ミスがあると無限ループが発生する恐れがある．たとえばノード B の R 用エントリの最初の配列要素に最短経路ではない 1 を記述してしまうと，ノード A から転送されたパケットを再び A に戻してしまうため無限ループが発生する．TTL などを用いた無限ループ検出機能が必要となる．

ARP ブロードキャストに対しては完全網と同じく各ノードの Eport 0 に接続したスイッチを用いて行う．転送コストが少ない ACK の送受信にも同じスイッチを利用する．

今回の完全網の実験から中継ノード数上限 1 では複数経路を用いたバンド幅向上が並列処理に効果的であることが分かった．図 14 の 3 次元トーラスは直径（ノード間距離で最大のもの）が 3 であるため，最短ルートでも中継数が 2 となる通信が発生する．この経路の利用を抑えること，たとえば近隣ノード間の通信を中心としたプログラム制作などの工夫が必要となる．

7. おわりに

スケーラビリティの低さからノードを完全網で接続することは現実的ではなかったが，近年，コア数の急増，マルチポート NIC の登場により 1,000 コアを超える完全網クラスタ構築が可能となった．試作した完全網 PC クラスタを評価した結果，迂回路対直経路の割合を 7 : 3 とした場合が最も良い性能を示し，FT，LU，MG，CG，IS，BT，SP，EP のベンチマークにおいて LACP に対する向上度は 2.79，1.56，1.96，1.18，4.37，1.10，1.13 となった．

当システムはブロードキャスト以外の通信に対してスイッチを用いないため，ノードの CPU 資源をフォワーディング処理に費やすことになる．しかしながら PC のコア技術は飛躍的に向上しており，コストあたりの CPU パワーは増加し続けている．通信処理へのノード CPU パワー提供はコア技術の向上により十分サポートされると考えている．また，スケーリングも問題となるが，上位ネットワーク接続用スイッチを用いて階層化することで大規模システムに対応する．頻繁な通信を行うプロセス群を単一の完全網 PC クラスタ内に配置し，比較的疎な通信はスイッチを経由して通信を行う．この構成による大規模システムの検証はこれからの課題である．

参考文献

[1] Boden, N.J., Cohen, D., Felderman, R.E., Kulawik, A.E., Seitz, C.L., Seizovic, J.N. and Su, W.-K.: Myrinet: A Gigabit-per-second Local Area Network, *IEEE Micro*, pp.29–36 (Feb. 1995).

[2] InfiniBand^{TM}Architecture Specification, available from ⟨http://www.infinibandta.org⟩, InfiniBand Trade Association (2004).

[3] Petrini, F., Feng, W.-C., Hoisie, A., Coll, S. and Frachtenberg, E.: The Quadrics Network: High-Performance Clustering Technology, *IEEE Micro*, Vol.22, No.1, pp.46–57 (2002).

[4] Watanabe, K., Otsuka, T., Tsuchiya, J., Nishi, H., Yamamoto, J., Tanabe, N., Kudoh, T. and Amano, H.: Martini: A Network Interface Controller Chip for High-Performance Computing with Distributed PCs, *IEEE Trans. Parallel and Distributed Systems*, Vol.18, No.9, pp.1282–1295 (2007).

[5] Tezuka, H., Hori, A. and Ishikawa, Y.: PM: A high-performance communication library for multi-user parallel environments, Technical Report TR-96015, Real World Computing Partnership (1996).

[6] Sumimoto, S., Naruse, A., Kumon, K., Hosoe, K. and Shimizu, T.: PM/InfiniBand-FJ: A High Performance Communication Facility Using InfiniBand for Large Scale PC Clusters, *Proc. 7th International Conference on High Performance Computing and Grid in Asia Pacific Region* (*HPCAsia'04*), pp.104–113 (2004).

[7] Petrini, F., Feng, W.-C., Hoisie, A., Coll, S. and Frachtenberg, E.: The Quadrics Network: High-Performance Clustering Technology, *IEEE Micro*, Vol.22, No.1, pp.46–57 (2002).

[8] Otsuka, T., Watanabe, K., Tsuchiya, J., Harada, H., Yamamoto, J., Nishi, H., Kudoh, T. and Amano, H.: Performance Evaluation of a Prototype of RHiNET-2: A Network-Based Distributed Parallel Computing System, *Proc. IASTED Int'l Multi-Conf. Applied Informatics* (*AI '03*), pp.738–743 (Feb. 2003).

[9] Pacheco, P.S.: *Parallel Programming with MPI*, Morgan Kaufmann Publishers, Inc. (2002).

[10] 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，情報処理学会論文誌：コンピューティングシステム，Vol.45, No.SIG 6, pp.35–44 (2004).

[11] 朴　泰祐，佐藤三久，宇川　彰：計算科学のための超並列クラスタ PACS-CS の概要，情報処理学会研究報告 2005-HPC-103，pp.133–138 (2005).

[12] 住元真司，大江和一，久門耕一，朴　泰祐，佐藤三久，宇川　彰：複数 Gigabit Ethernet を用いた PACS-CS のための高性能通信機構の設計と評価，情報処理学会論文誌，Vol.47, No.SIG 12, pp.25–34 (2006).

[13] Arimilli, B., Arimilli, R., Chung, V., Clark, S., Denzel, W., Drerup, B., Hoefler, T., Joyner, J., Lewis, J., Li, J., Ni, N. and Rajamony, R.: The PERCS High-Performance Interconnect, *Proc. 2010 18th IEEE Symposium on High Performance Interconnects*, pp.75–82 (2010).

[14] Kim, J., Dally, W., Scott, S. and Abts, D.: Technology-Driven, Highly-Scalable Dragonfly Topology, *ISCA*, pp.77–88 (2008).

福永 隆文 （正会員）

1961 年生．1986 年宮崎大学卒業．2009 年熊本大学大学院自然科学研究科システム情報科学専攻博士課程修了．博士（工学）．現在，熊本県立技術短期大学校情報システム技術科教授．並列分散処理の研究に従事．