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

住元真司[†] 大江和一[†] 久門耕一[†]
朴泰祐[††] 佐藤三久[††] 宇川彰[††]

本論文では，PACS-CS システムのための高性能通信機構 PM/Ethernet-HXB の設計と評価について述べる．PACS-CS の計算ノードは Gigabit Ethernet を計算ネットワークとして 1 次元に 2 系統ずつ用いた 3 次元 Hyper Crossbar 結合を備える．PM/Ethernet-HXB の設計では，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も並行して行うため，複数 Ethernet を用いた通信処理コストを極限まで下げることが重要な課題である．この課題を解決するため，通信バッファ間での Zero-Copy 通信を実現する超軽量通信プロトコルを開発した．PM/Ethernet-HXB を SCore クラスタシステムソフトウェア上に実装，評価した．低レベル通信評価の結果，Gigabit Ethernet 9 本時に，片方向で 1065 MB/s の通信バンド幅性能，3 次元隣接通信の片方向で 741 MB/s（理論性能の 98.8%），双方向で 1401 MB/s（理論性能の 93.4%）と高い通信バンド幅性能を達成している．3 次元結合における MPI アプリケーション性能劣化は，NAS 並列ベンチマーク Class C，8 ノードにおいて 1 次元結合と 3 次元（$2 \times 2 \times 2$）結合の性能差で IS で 13%，CG で 7%と小さくおさえられており，既存の Gigabit Ethernet ハードウェアを用いて高いアプリケーション性能を実現している．

The Design and Evaluation of High Performance Communication Facility Using Multiple Gigabit Ethernet for PACS-CS System

Shinji Sumimoto,[†] Kazuichi Ooe,[†] Kouichi Kumon,[†]
Taisuke Boku,[††] Mitsuhisa Sato[††] and Akira Ukawa[††]

This paper discusses the design and evaluation of high performance communication facility called PM/Ethernet-HXB for PACS-CS system. A computing node of the PACS-CS uses 3-dimensional hyper crossbar connections, which have two Gigabit Ethernet for one dimensional connection. In the PM/Ethernet-HXB design, communication protocol overhead must be minimized on multiple Ethernet devices, because the PACS-CS requires not only direct communications between nodes but also communication using routing nodes. To minimize the communication protocol overhead, a light weight communication protocol using Zero-copy communication between communication buffers of nodes has been developed. We have implemented the PM/Ethernet-HXB on SCore cluster system software, and evaluated its communication and application performance. The PM/Ethernet-HXB achieves 1065 MB/s of uni-directional communication bandwidth using nine Gigabit Ethernet links, 741 MB/s (98.8% of theorical performance) of uni-directional, 1401 MB/s (93.4% of theorical performance) of bi-directional bandwidth on the 3-dimensional hyper crossbar connections. The performance differences using NAS parallel benchmarks between 3-dimensional hyper crossbar ($2 \times 2 \times 2$) and 1-dimensional network (8 nodes) are minimal, that of IS Class C is 13%, and that of CG Class C is 7%. Therefore, PM/Ethernet-HXB realizes higher application performance using existing Gigabit Ethernet hardware.

1. はじめに

計算科学における大規模シミュレーション需要は拡大の一途である．分野は多岐にわたり，物性科学，バイオインフォマティクス，生命科学などの工学応用に加え，素粒子，宇宙などの基礎科学においても，1 PFLOPS クラスの高い計算性能が必要となりつつある．

筑波大学計算科学研究センターでは，これまで大規模シミュレーションの実効性能を高めるために，CP-PACS [1] をはじめ，専用の並列計算機を開発し利用してきた．614 Gflops の計算性能を持つ CP-PACS [1] は，1996 年より利用され，素粒子物理学，物性物理学

† 富士通研究所
FUJITSU LABORATORIES
†† 筑波大学
University of Tsukuba

における大規模計算を実行している．しかし，より高い計算性能への必要性から，2006 年 6 月導入を目標に CP-PACS の 20 倍以上の性能を持つ，PACS-CS [2),9)] を開発することになった．

PACS-CS は，PC のプロセッサを用い CP-PACS で採用した 3 次元 Hyper Crossbar 結合を複数の Gigabit Ethernet ネットワークで実現する新しいコンセプトの超並列クラスタシステムである．PACS-CS は，InfiniBand などの専用クラスタインターコネクトを採用する代わりに Gigabit Ethernet を 1 方向に 2 本使い 3 次元接続することにより，計算ノードあたり片方向で 750 MB/s（双方向 1,500 MB/s）の通信バンド幅を持つネットワークを備える．PACS-CS の性能を引き出すには 3 次元 Hyper Crossbar ネットワーク性能を最大限に引き出す通信機構が必要である．

本論文では，PACS-CS 上で高い通信性能を実現する高性能通信機構 PM/Ethernet-HXB の設計と評価について述べる．PM/Ethernet-HXB の設計では，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も並行して行うため，複数 Ethernet を用いた通信処理コストを極限まで下げることが重要な課題である．この課題を解決するため，通信バッファ間での Zero-Copy 通信を実現する超軽量通信プロトコルを開発した．

PM/Ethernet-HXB を SCore クラスタシステムソフトウェア上に実装，評価した．低レベル通信評価の結果，Gigabit Ethernet 9 本時に，片方向で 1065 MB/s の通信バンド幅性能，3 次元隣接通信の片方向で 741 MB/s，双方向で 1401 MB/s と高い通信バンド幅性能を達成している．また，8 ノード PC クラスタでの MPI アプリケーションでも高い実行性能を達成している．

本論文の構成は，2 章で PACS-CS の概要と通信機構の設計課題を述べ，3 章でこれを実現する通信機構 PM/Ethernet-HXB の設計を議論する．4 章で PM-Ethernet-HXB の実装，5 章で評価する．6 章に関連研究について触れ，7 章でまとめる．

2. PACS-CS の概要と通信機構の設計課題

2.1 PACS-CS の概要

PACS-CS [2)] は，筑波大学で開発進行中の PC クラスタシステムで，3 次元 Hyper Crossbar ネットワークを持つ．システムの概要を**表 1** に示す．

PACS-CS の特徴は，計算処理と通信処理のメモリバンド幅バランスを考慮した単一プロセッサノードと 3 次元 Hyper Crossbar 結合（16 × 16 × 10）を採用

表 1 PACS-CS システムの概要
Table 1 PACS-CS System specification overview.

ノード計算機	Intel LV Xeon 2.8 GHz（EM64T） （Intel E7520, 2 GB DDR2 SDRAM 4 × 64 bit 133 MHz PCI-X Bus）
計算ネットワーク	3 次元 Hyper Crossbar Gigabit Ethernet（E1000）× 6 jumbo frame 利用 日立電線製 Switch 利用
管理 IO ネットワーク	Gigabit Ethernet x2
ホスト OS クラスタ OS	Linux Fedora Core 3 for x86_64 SCore5.8.3 [3)]
ノード数	2,560（16 × 16 × 10）

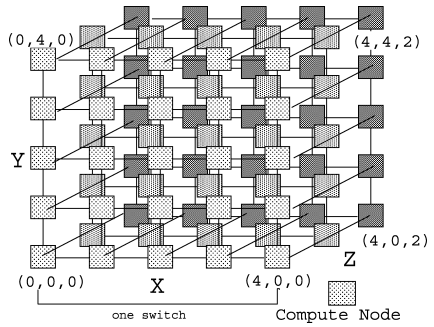


図 1 3 次元 Hyper Crossbar 結合例（5 × 5 × 3）
Fig. 1 Connection example of 3D Hyper Crossbar network (5 × 5 × 3).

している点である．**図 1** は 3 次元 Hyper Crossbar 結合（5 × 5 × 3）の結合例である．図 1 において，各次元に連なる 1 本の線が 1 台のスイッチに相当し，3 次元 Hyper Crossbar 結合では，この (X, Y, Z) の各次元につながるノードがそれぞれ 1 台のスイッチで結合される．

3 次元 Hyper Crossbar 結合におけるノード間通信は，たとえば図 1 中において，座標 (0, 0, 0) のノードから，座標 (1-4, 0, 0)，座標 (0, 1-4, 0)，座標 (0, 0, 1-2) のノードへは Ethernet スイッチ経由で直接通信が可能であるが，たとえば座標 (4, 0, 2) や (4, 4, 2) のノードへは直接通信できない．座標 (0, 0, 0) のノードから座標 (4, 0, 2) のノードへは最短 1 回中継ノード☆を経由した通信が必要である．同様に座標 (4, 4, 2) のノードでは最短 2 回中継ノード☆☆を経由した通信が必要である．

実行される主要なアプリケーションは，素粒子物理学における QCD 計算であり，CP-PACS の時代から 3 次元 Hyper Crossbar 結合向けに開発されてきたも

☆ Ex 座標 (4, 0, 0) が中継ノード．
☆☆ Ex 座標 (4, 0, 0) と座標 (4, 0, 2) が中継ノード．

のである．これらのアプリケーションは多次元メッシュ上に接続されたノード間でデータを交換しながら計算を行うため，3 次元の隣接双方向通信を多用している．通信機構の開発では，この 3 次元の隣接双方向通信をハードウェア性能限界まで引き出すとともにルーティングをともなう通信を高速に行うことが重要となる．

2.2 PACS-CS 上の通信機構の課題

PACS-CS 上の通信機構の課題は

- 複数の NIC を用いた高性能通信
- 次元間のルーティング処理（最大 2 段）

を低通信処理オーバヘッドで実現することにある．

特に PACS-CS では，1 ノードに計算用ネットワークとして，Zero-Copy 通信など特殊な機能を持たない Gigabit Ethernet を 6 系統採用している．1 ノードが扱う総通信バンド幅は片方向 750 MB/s（双方向 1,500 MB/s）と InfiniBand や Myrinet などのクラスタ専用インターコネクトと同等レベルである．また，通信処理については，クラスタ専用インターコネクトがネットワークインタフェース（NIC）上で処理するのと異なり，Ethernet ではホストプロセッサで処理する必要がある．このため，通信処理がアプリケーションに対して外乱となって実行性能に影響を与える場合が予想される．したがって通信プロトコル処理は，極限まで無駄なく低オーバヘッドである必要がある．

2.3 複数 NIC を用いる通信処理の要件

表 2 に，NIC 数ごとの片方向通信を行った場合の各パケットサイズごとの到着時間間隔を示す．本表は理論的な通信性能をもとに計算したものである．表 2 は同様に，1 パケットあたりの送信，および受信処理を表記載の時間以内に処理しないと最大通信バンド幅が得られないことを示している．

表 2 より，Giga × 6 で 8 KB メッセージで 750 MB/s を実現するには，1 パケットあたり 10.9 μs 以内に通信処理を行う必要がある．双方向通信の場合は，最大通信バンド幅を実現するために，半分の 5.5 μs 以内に通信処理を行わなければならない．

この 5.5 μs が通信機構の実現目標になるが，この時間の中には Ethernet デバイスドライバ，Linux OS の処理など通信プロトコル以外の処理も含まれるため，通信処理は極限まで無駄を排除しなければならない．

表 2 NIC 数とパケットサイズごとの到着時間間隔（μs）
Table 2 Number of NICs vs. message arrival period (μs).

	1500B	4096B	8192B
Giga × 1	12.0	32.8	65.5
Giga × 2	6.0	16.4	32.8
Giga × 4	3.0	8.2	16.4
Giga × 6	2.0	5.5	10.9
Giga × 8	1.5	4.1	8.2
Giga × 10	1.2	3.3	6.6

3. PM/Ethernet-HXB の設計

本章では，PACS-CS 上のために開発された通信機構 PM/Ethernet-HXB の設計について述べる．

3.1 設 計 方 針

PM/Ethernet-HXB の設計方針として，以下の 4 つを考慮する．

- PM/Ethernet [4),5)] をベースにし，これを高速化するアプローチを採用
- Kernel コードの変更なし．デバイスドライバの組み込みだけで実現
- Ethernet ドライバを改造．しかし改造は最低限
- マルチプロセッサシステムでも動作可能

PM/Ethernet では，TCP/IP プロトコル処理を解析した結果に基づき，オーバヘッドが大きい信頼性確保のためのプロトコル処理，ハードウェア・ソフトウェア割込みのオーバヘッドを排除し高い通信性能を実現している．このため，残るオーバヘッドを排除しなければならない．

残る大きなオーバヘッドとして受信側のプロセッサコピーがあり，この削減手法として Zero Copy 通信の実現がある．PM/Ethernet では送信側の Zero Copy 通信は実現しているが，受信側は実現していない．コピー処理分のオーバヘッドを削減可能である．これ以外は，細かな最適化処理の積み重ねにより通信オーバヘッドを削減する．

3.2 Zero Copy 通信の実現

既存の Gigabit Ethernet NIC と Linux を用いた通信では，Linux の通信用のバッファである skbuf を用いた通信となる．Ethernet を用いた高性能通信において，送信時の Zero-Copy は PM/Ethernet [4),5)] ですでに実現されているため，受信時の Zero-Copy の実現について議論する．

パケット受信時には，Linux の枠組みを使う限り skbuf に最初にパケットが格納される．このパケットを，ユーザプロセスがコピーなしに参照可能とするためには，受信パケットが格納された skbuf をユーザプロセスの仮想アドレス空間に map すれば参照可能になる．

通常，受信パケットを格納する skbuf は Ethernet デバイスドライバで受信ハードウェアに割り当てられ，パケット受信処理後に Linux カーネルに返還され再利用される．返還時には別目的で利用される可能性があるため，受信後にユーザプロセスに map し，通信処

理後には，map を解除しなければならない．

しかし，参考文献 6) によると，この map に要するコストは 4 KB ページあたり 0.064 μs で 9,000 バイトの jumbo frame の map に 0.192 μs，map 解除にも同じコストが必要で，計 0.384 μs かかる計算になりコストが非常に大きい．

この問題を解決するため，Ethernet デバイスドライバが受信用に割り当てる skbuf はすべて再利用し，この受信用の skbuf のすべてをユーザプロセスにあらかじめ map しておく方式を採用することにした（**受信 skbuf pre-map 方式**）．この受信 skbuf pre-map 方式では，あらかじめ受信用の skbuf に ID を与えておき，ユーザプロセスではその ID に応じたアドレスに該当の skbuf を map しておく．

メッセージ受信時，デバイスドライバの受信処理で skbuf から ID を解釈し，その ID をユーザプロセスに通知する．ユーザプロセスはその ID から ID に応じたアドレスを参照することにより，受信パケットをコピーすることなく参照可能になる．

この方式は，通信プロトコル処理を大きく削減することが可能であるが，全体の skbuf をあらかじめ map するため，skbuf の量に応じたユーザプロセスの仮想アドレス空間が必要である．

3.3 通信プロトコルの軽量化

複数のネットワークにパケットを分散させた場合の通信において，最も問題となるのは，送信順にパケットが届かないため，受信側でパケットを送信順に並べかえる処理（パケット Ordering）が必要な点である．

複数の Ethernet を用いた高性能通信機構として PM/Ethernet Network Trunking [7]（以下，PM/Ethernet NT）がある．PM/Ethernet NT の実装では，パケット Ordering に skbuf の持っている構造体リンクを用いてパケットの順にキュー構造で実装し，このキューへのアクセスのために test and set 関数による排他制御を用いている．この排他制御は，1 つのパケットの出し入れに 2 回行われることになる．

しかし，この排他制御のコストを Linux 2.6.14.4 Xeon 3.6 GHz の計算機で測定したところ，1 回あたり 0.07 μs，2 回で 0.14 μs と 5.5 μs の 2.5%の定常コストがかかることが分かった．このため，実装する軽量通信プロトコルの設計では，定常的に発生する排他制御処理を排除する．

パケット Ordering 処理において排他制御処理を排除するために，パケットの Sequence 番号の特性を利用した．Sequence 番号が同じパケットが同時に届かない性質を利用して，ノードの宛先ごとに一定数の配列を設け，配列の値が 0 の場合は空き，0 以外の場合は受信済みとして排他制御を不要にしている．なお，0 以外の値としては skbuf を識別できる値を代入する．この方式は，排他制御処理が不要，かつ，パケットの並べかえが不要な点において高速であるが，相手先ごとに一定数の配列が必要であるため，メモリ資源を多く必要とする．

これ以外に，たとえば次に述べる最適化を行い通信プロトコル処理を削減する．

- alloc_skb（free_skb）処理の高速化
- システムコール呼び出し回数の削減
- システムコール実行時の引数コピーの削減

3.4 ルーティング処理の高速化

ルーティングアルゴリズムは CP-PACS のアルゴリズムを採用している [1),8)]．このルーティングアルゴリズムは，各軸の転送順を，たとえば，最初に X 軸方向，次に Y 軸，最後に Z 軸と一定にする方式である．

この処理を高速に実行するために，各計算ノードを 3 次元座標で管理するとともに高速にルーティング処理を実施するために，受信用の skbuf をそのまま送信用の skbuf として再利用し，ヘッダだけを変更して転送する方式とする．

なお，PACS-CS では，Ethernet デバイスに十分な送受信リソースがあり，かつ万が一送受信のための通信リソースが枯渇した場合にはパケット廃棄で再送を行うため，ルーティング処理におけるデッドロックは発生しない．

4. 実　　装

図 2 に，PM/Ethernet-HXB のソフトウェア構成を示す．PM/Ethernet-HXB は，SCore の通信機構である PMv2 [10),11)] の 1 つの通信デバイスとして実装されている．OS は Linux(2.6.14.4) である．

PM/Ethernet-HXB は，PMv2 Library の API を実装している PM/Ethernet-HXB User Library と 3

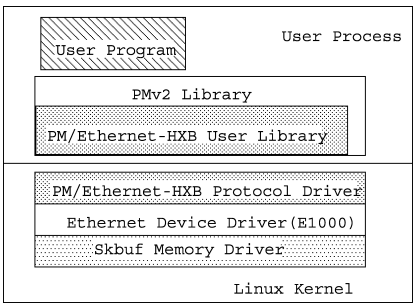


図 2 PM/Ethernet-HXB の構成
Fig. 2 PM/Ethernet-HXB architecture.

つのデバイスドライバで構成されている．各デバイスドライバの概要は次のとおりである．

Skbuf Memory Driver： skbuf を扱うデバイスドライバである．デバイスドライバロード時に一定量の skbuf を確保し，再利用する．

Ethernet Device Driver（E1000）： E1000 のデバイスドライバを変更して利用している．変更箇所は，関数置き換え用ヘッダの include 文挿入と，デバイスドライバ名の変更のみ．変更内容は，Skbuf Memory デバイスドライバを使ったバッファ割当て関数名とメッセージ処理関数（netif_rx など）の関数名の置き換えである．

PM/Ethernet-HXB Protocol Driver： 送受信処理，ルーティング処理などを実装している．

全体的にドライバからライブラリまで処理を簡潔化し最適化して実装している．Linux カーネルへの変更はしない方針であるため，PM/Ethernet で実現した Interrupt Reaping 機構は実装していない．また，3.3 節で述べた，宛先ごとのメッセージ受信のための配列は現状，1 ノードあたり，int タイプの配列を 32 割り当てている．このため，2,560 ノードで，320 KB 消費している．また，3.2 節で述べた，Skbuf Memory ドライバで確保する skbuf の量は 12 KB のサイズを 4,096 個（48 MB）確保している．

Skbuf Memory デバイスドライバにおいて，alloc_skb（free_skb）処理の高速化を行った．Linux 2.6.14.4，Xeon 3.6 GHz 上での，alloc_skb+free_skb の処理コストは，オリジナルで 0.22 μs であったが，最適化を行い，0.11 μs と高速化している．

5. 評　　価

本章では，実現した PM/Ethernet-HXB の通信性能とアプリケーション性能を示し，設計により高い性能が得られているかを評価する．

PACS-CS システムは，まだ実機が完成していないので可能な限り条件をあわせた 8 ノードクラスタ環境を構築し測定を行った．**表 3** に評価環境を示す．Ethernet スイッチ機種の違いとプロセッサ周波数以外は同じ構成とである．Ethernet スイッチについてはどちらも同じチップセットを用いて作られており，通信性能評価のうえ，同等性能であることを確認している．

評価は，低レベル通信性能，MPI レベルの通信性能，そしてアプリケーションとして NAS 並列ベンチマークの実行性能を 1 次元と多次元環境で測定し実施する．通信性能評価では，PM/Ethernet-HXB の通信性能の限界を見極めるために，PACS-CS の持つリンク数以上の 9 リンクまでの性能評価を行う．Dual E1000 NIC は E7520 の 4 つの PCI-X バスにそれぞれ 1 枚ずつ搭載しているため．9 リンクの評価の場合，PCI-X バスの 1 つに E1000 が 3 つ接続される．

表 3 8 ノード性能評価クラスタ環境
Table 3 Measurement environments of 8 node cluster.

ノード計算機（8 ノード）	Intel Xeon 3.6 GHz（1 MB cache）（Intel E7520，2 GB DDR2 SDRAM，4 × 64 bit 133 MHz PCI-X Bus）
Ethernet（1 Gbps）	Intel Dual E1000 NIC × 4，+ E1000 × 1 Foundry FWSX 448 GigE Switch
OS	Fedora Core 3 for x86_64,（2.6.14.4 Uni-Processor kernel）SCore5.8.3

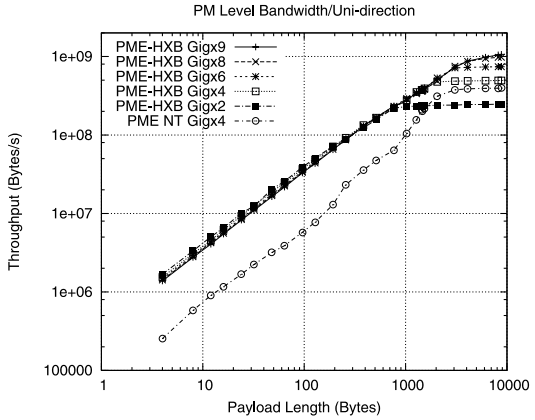


図 3 低レベルの通信バンド幅性能：1 次元片方向
Fig. 3 Low level communication bandwidth performance: 1D unidirection.

アプリケーション評価で用いたコンパイラは Intel コンパイラ（Version 9）である．なお，比較として低レベル通信では PM/Ethernet NT，アプリケーション性能では TCP/IP 上の MPI との結果もあわせて示す．これ以降 PM/Ethernet-HXB は，図表中 PME-HXB，PM/Ethernet NT は，図表中 PME NT と示す．

5.1 低レベル通信バンド幅性能評価：1 次元片方向

本節では 1 次元の低レベル通信性能を測定，パケットあたりの処理時間を算出し，2.3 節で述べた要件を満たしているかを示す．

図 3 に，Gigabit Ethernet リンク数を 9 まで増加させた場合の 1 次元片方向の低レベル通信バンド幅性能を示す．比較としてリンク数が 4 の場合の PM/Ethernet NT の結果も載せている．図 3 より，PM/Ethernet-HXB の性能はメッセージサイズが 768 バイト以下ではほぼ同じであることが分かる．また，メッセージサイズ 768 バイト以下の結果は PM/Ethernet NT の結

表 4 低レベルの通信バンド幅性能と処理遅延
Table 4 Low level communication bandwidth performance and processing latency.

	通信バンド幅性能	処理遅延
PME-HXB (Gigx2)	248 MB/s (99%)	2.38 μs
PME-HXB (Gigx4)	494 MB/s (99%)	2.58 μs
PME-HXB (Gigx6)	741 MB/s (99%)	2.71 μs
PME-HXB (Gigx8)	981 MB/s (98%)	2.81 μs
PME-HXB (Gigx9)	1065 MB/s (94%)	2.90 μs
PME NT (Gigx4)	393 MB/s (78%)	15.7 μs

() 内は理論性能に対する割合

果に比べて 4〜6 倍高い結果である☆.

表 4 に，図 3 の結果における最大通信性能と転送時のメッセージ処理遅延を示す．転送時のメッセージ処理遅延は，4 バイトの転送バンド幅値から算出した．

表 4 の通信バンド幅性能の結果より，PM/Ethernet-HXB はリンク数が 9 まで理論性能に対して 94%以上の高い通信効率を達成しており，リンク数が 9 において 1,065 MB/s と InfiniBand 4x の理論性能を超える性能を実現している．これは，PM/Ethernet-HXB が Zero-Copy 通信を実現している結果である．リンク数増加による通信処理の増加分は 9 リンクまで 1 リンクあたり 0.07 μs と小さく抑えられている．これに対し，PM/Ethernet NT では，リンク数 4 本で 393 MB/s（76%）と，すでに頭打ちである．これは，通信プロトコル処理オーバヘッドのため（処理遅延 15.7 μs + Copy 遅延 5 μs）である．

また，表 4 の処理遅延結果より，PM/Ethernet-HXB はリンク数 9 本のときにおいてさえも 2.9 μs の通信処理遅延を実現している．プロセッサ周波数考慮の場合 3.7 μs で，目標の 5.5 μs 以下を実現している．

5.2 低レベル通信バンド幅性能評価：多次元隣接

本節では，3 次元 Hyper Crossbar 結合上での通信性能を評価する．3 次元 Hyper Crossbar 結合では，同時に 3 方向に通信可能である．このため次元数に応じた通信先のノード数を準備し，次の 3 つの典型的な通信パターンを測定し評価する．

送信： 1 つのノードが各次元のノードに対して送信
受信： 1 つのノードに対して各次元のノードが送信
双方向： 1 つのノードが各次元に対して送受信し，各次元のノードも同様に送受信

表 5 に測定結果を示す．表 5 のすべての結果において送信，受信とも理論性能の 98.7%以上の通信性能を実現しており，双方向でも理論性能の 93.4%以上の通信性能を実現している．特に 3 次元通信時に送信，受信では，741 MB/s 以上，双方向では 1401 MB/s の通信性能を実現しており，PACS-CS で多用される 3 次元隣接通信においても高い通信性能を実現している．

☆ ex：4 バイトメッセージ時，PM/Ethernet NT の 0.25 MB/s に比べて，PM/Ethernet-HXB は 1.5 MB/s.

表 5 低レベルの通信バンド幅性能：PME-HXB N 次元
Table 5 Low level communication bandwidth performance: PME-HXB Nth dimension.

MB/s	送信	受信	双方向
1 次元	247.6 (99.0%)	247.5 (99.0%)	481.1 (96.2%)
2 次元	493.6 (98.7%)	494.9 (99.0%)	951.3 (95.1%)
3 次元	741.3 (98.8%)	742.3 (99.0%)	1401.3 (93.4%)

1 次元：2 ノード通信，2 次元：3 ノード通信，3 次元：4 ノード通信
() 内は理論性能に対する割合

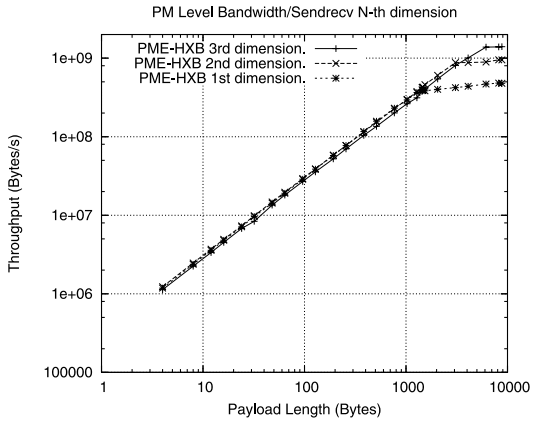


図 4 低レベルの通信バンド幅性能：N 次元送受信
Fig. 4 Low level communication bandwidth performance: Nth dimension, sendRecv.

図 4 に，1 次元から 3 次元までの送受信性能の結果を示す．図 4 より次元数増加による通信処理性能の劣化は小さく，次元数増加で最大バンド幅が次元数の分だけ増加することが分かる．さらに 5.1 節の 9 リンクまでの性能とあわせて，PM/Ethernet-HXB の通信性能は，片方向，双方向ともに素直な特性を示している．

5.3 低レベルルーティング通信性能評価

本節では，3 次元 Hyper Crossbar 結合のルーティング処理挿入時の低レベル通信性能を評価する．

表 6 に，リンク数 2 本と 4 本で 1 次元から次元数を増加させた場合の通信バンド幅性能と，リンク数 2 本で次元数を増加させた場合の 1/2 ラウンドトリップ時間（RTT）の測定結果を示す．表 6 より，リンク数 2 本，4 本の場合とも最大通信バンド幅性能は理論性能の 97%以上を実現していることが分かる．また，1/2 RTT の測定結果より，ルーティングが 1 段増えるごとに 12 μs 程度増える結果となった．このう

表 6 ルーティング処理挿入時の低レベル通信バンド幅性能と 1/2 RTT：PME-HXB
Table 6 Low level communication bandwidth performance and 1/2 communication round trip time with routing processing: PME-HXB.

	Giga × 2 バンド幅	Giga × 4 バンド幅	Giga × 2 1/2 RTT
1 次元	248 MB/s	494 MB/s	14.9 (9.8) μs
2 次元	248 MB/s	489 MB/s	27.5 (17.3) μs
3 次元	248 MB/s	-	39.3 (24.0) μs

注）1/2 RTT はスイッチ 1 段あたりの遅延 5.1 μs を含む
() 内はスイッチの遅延を除いた結果

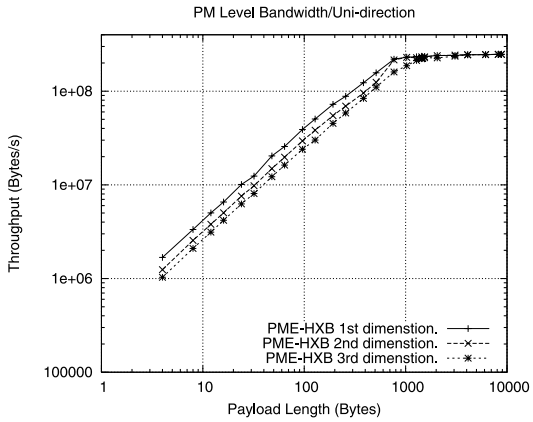


図 5 低レベルの片方向通信バンド幅性能：N 次元ルーティング
Fig. 5 Unidirectional low level commnication bandwidth performance: Nth dimension with routing.

ち Ethernet スイッチの遅延が 5.1μs あるので，その差 6.9 μs がメッセージルーティング 1 段あたりのオーバヘッドといえる．

図 5 に，リンク数 2 本時の次元数を変化させた場合の片方向の通信バンド幅性能を示す．図 5 の結果より，ルーティング処理が入ることで，メッセージサイズが 512 バイト以下で一定のオーバヘッドが生じていることが分かる．このオーバヘッドは，4 バイトメッセージ時にルーティング 1 段で 26%，ルーティング 2 段で 39%である．このためショートメッセージを多用するアプリケーションの場合にルーティングが入る通信を多用すると性能劣化する場合がある．しかし，PACS-CS 上の主要アプリケーションである QCD，DFT は，ともにルーティング処理をともなう通信を極力使わない実装となっているため大きな問題とはならない．

5.4 MPI 通信性能

本節では，YAMPII2 [12] を用い 1 次元の MPI レベルの通信性能を評価する．YAMPII2 は eager と rendezvous プロトコルを用いており，本 MPI 通信性能評価においては MPI の非同期通信を用いた rendezvous プロトコルを用いて，1 MB 程度のメッセージ長で通信性能測定を行った．**表 7** に測定結果を示す．

表 7 MPI レベルの通信バンド幅性能：PME-HXB
Table 7 MPI level communication bandwidth performance: PME-HXB.

	YAMPII2 片方向	YAMPII2 双方向
Gigx2	244 MB/s (98%)	490 MB/s (98%)
Gigx4	494 MB/s (99%)	896 MB/s (90%)
Gigx6	739 MB/s (98%)	937 MB/s (62%)
Gigx8	915 MB/s (92%)	932 MB/s (47%)
Gigx9	890 MB/s (79%)	921 MB/s (40%)

() 内は理論性能に対する割合

表 7 より，リンク数 8 本までは，理論性能の 90%以上を実現しているが，9 本では通信性能が落ちていることが分かる．この性能劣化は，MPI 処理時間の増加と PM/Ethernet-HXB のリンク数増加による通信処理時間の増加が相互に影響し発生している．片方向通信での性能劣化は，リンクごとの通信処理時間の違い（表 4 より，6 リンク時 2.71 μs，8 リンク時 2.81 μs と 9 リンク時 2.90 μs）により，プロトコル処理での ACK 送信にタイミング的な遅れが生じたためである．また，双方向通信での性能劣化は，リンク数増加による PM/Ethernet-HXB の処理時間の増加によるもので，この時間に MPI 処理時間（6 μs）が加わると，6 リンクの場合で 8.71 μs（通信バンド幅 941 MB/s），8 リンク 8.81 μs（通信バンド幅 929 MB/s），9 リンク 8.90 μs（通信バンド幅 920 MB/s）とリンク数増加の微小な時間増加が性能劣化に関係している．これは，6 リンクにおいても微小なオーバヘッドが加わると性能劣化する可能性を示唆しており，安定したアプリケーション性能の実現のためには，さらなる最適化が必要である．

5.5 アプリケーション性能評価：1 次元

本節では，アプリケーションとして NAS 並列ベンチマークを用い 1 次元でのリンク数を 6 本まで変化させた場合の性能差を評価する．測定に利用した MPI は YAMPII2 [12] である．**図 6**，**図 7**，**図 8** に，それぞれ IS，CG，LU のクラス C の結果を示す．比較として YAMPII2/TCP と MPICH-1.2.5/p4 の結果も示す．

表 8 に性能比較をまとめる．表 8 より，通信バンド幅性能に依存する IS，CG ではリンク数増加によるアプリケーション性能向上が見込めることが分かる．

5.6 アプリケーション性能評価：1 次元 vs 3 次元

本節では，ルーティング挿入時のアプリケーション性能への影響を評価するために，8 ノードクラスタでの 1 次元（リンク数 2 本と 6 本）と 3 次元（2×2×2，

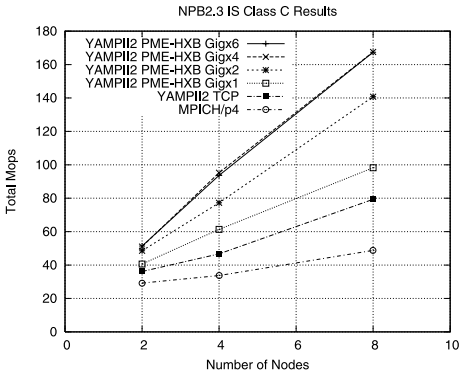


図 6 アプリケーション実行性能：IS クラス C
Fig. 6 Application performance results: IS CLASS C.

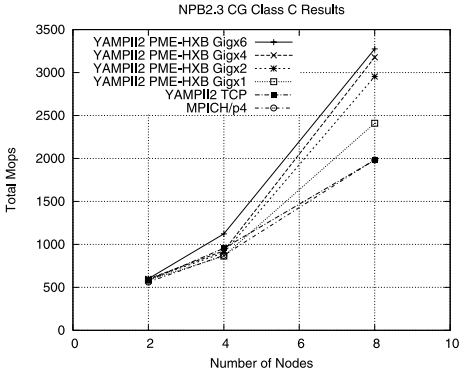


図 7 アプリケーション実行性能：CG クラス C
Fig. 7 Application performance results: CG CLASS C.

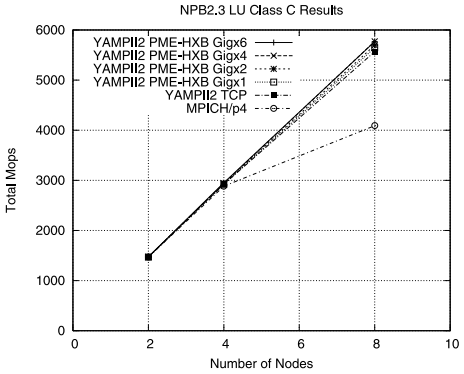


図 8 アプリケーション実行性能：LU クラス C
Fig. 8 Application performance results: LU CLASS C.

表 8 NAS 並列ベンチマークの 6 リンク性能との比較 PME-HXB（8 ノード：1 次元）
Table 8 Comparison of NAS parallel benchmark performance results with using 6 link: PME-HXB (8 nodes: 1D).

	性能向上	対 1 リンク	対 YAMPII2/TCP
IS-C	4 リンクまで	170%	211%
CG-C	4 リンクまで	135%	165%
LU-C	2 リンクまで	108%	103%

表 9 NAS 並列ベンチマーク Class C の性能比較 PME-HXB（8 ノード：1 次元 vs 3 次元）
Table 9 Performance comparison of NAS parallel benchmark results class C: PME-HXB (8 nodes: 1D vs. 3D).

	1 次元 (Gig × 2)	1 次元 (Gig × 6)	3 次元 (Gig × 2)
IS	140.8 (95%)	167.4 (113%)	147.6 MOPS (100%)
CG	2954.0 (97%)	3275.1 (107%)	3059.7 MOPS (100%)
LU	5696.8 (99%)	5769.4 (101%)	5739.4 MOPS (100%)

() 内は 3 次元の結果を 100%とした場合の割合

リンク数 2 本 × 3 次元 = 6 本）の性能比較を行う．本測定では，実行バイナリは 1 次元，3 次元ともに同じバイナリを用い，3 次元用に特殊なものは使っていない．測定に利用した MPI は YAMPII2[12] である．**表 9** にこの測定結果を示す．

表 9 より総リンク数が同じ 6 本の場合の 1 次元と 3 次元の結果を比較すると，性能差は IS で 13%，CG で 7%である一方，LU で 1%と小さいことが分かる．特に IS の差は全対全通信を行うため 13%あるが，図 6 の YAMPII/TCP の結果に比べると 86%高い．ソフトウェアルーティング 2 段の挿入時でも性能劣化が小さいからである．

リンク数 2 本の 1 次元と 3 次元の結果を比べた場合には，CG と IS は，3 次元の場合の方が 3〜5%性能が高い．これは，CG，IS では通信バンド幅が性能に与える影響が大きく，ルーティング遅延の影響より総リンク数増加の効果が大きいことを意味する．

6. 関連研究

複数の NIC をサポートしたものに，PM/Ethernet Network Trunking[7]，PM/Ethernet-kRMA[13]，Channel Bonding[14] があるが，多次元 Hyper Crossbar を想定したルーティング機構は持っていない．Hyper Crossbar 結合を効率良く実現可能なものに VLAN TAG 利用の結合方式[15] があるが，複数ネットワークを扱うには Channel Bonding との併用が必要である．

Hyper Crossbar 結合をハードウェアでサポートした専用ネットワークとして SCI[16]，CP-PACS[1] などがあるが，PM/Ethernet-HXB は既存の Ethernet を用いソフトウェアのみで実現している点が異なる．

7. 結　論

本論文では，PACS-CS システムのための高性能通信機構 PM/Ethernet-HXB の設計と評価について述

べた．PM/Ethernet-HXB の設計では，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も並行して行うため，複数 Ethernet を用いた通信処理コストを極限まで下げることが重要な課題である．この課題を解決するため，通信バッファ間での Zero-Copy 通信を実現する超軽量通信プロトコルを開発した．

PM/Ethernet-HXB を SCore クラスタシステムソフトウェア上に実装，評価した．低レベル通信評価の結果，Gigabit Ethernet 9 本時に，片方向で 1065 MB/s の通信バンド幅性能，3 次元隣接通信の片方向で 741 MB/s（理論性能の 98.8%），双方向で 1401 MB/s（理論性能の 93.4%）と高い通信バンド幅性能を達成している．また，MPI 通信性能でも，8 リンクで 915 MB/s を実現している．3 次元結合における MPI アプリケーション性能劣化は，NAS 並列ベンチマーク Class C，8 ノードにおいて 1 次元結合と 3 次元（$2 \times 2 \times 2$）結合の性能差で IS で 13%，CG で 7%と小さくおさえられており，既存の Gigabit Ethernet ハードウェアを用いて高いアプリケーション性能を実現している．

なお，PM/Ethernet-HXB は PACS-CS 向けに開発したものであるが，一般の PC クラスタにおいても複数の Gigabit Ethernet を搭載することで容易に適用することが可能である．

今後は，MPI を含めた通信の高速化を進めるとともに，PM/Ethernet-HXB を PACS-CS 上で稼働させ，大規模アプリケーション評価を行う予定である．

参 考 文 献

1) Boku, T., Itakura, K., Nakamura, H. and Nakazawa, K.: CP-PACS: A massively parallel processor for large scale scientific calculations, *International Conference on Supercomputing'97*, pp.108–115, ACM (July 1997).
2) 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 05-HPC-103（SWoPP'2005）(Aug. 2005).
3) SCore Cluster System Software. http://www.pccluster.org/
4) 住元真司, 堀 敦史, 手塚宏史, 原田 浩, 高橋俊行, 石川 裕：既存 OS の枠組を用いたクラスタシステム向け高速通信機構の提案，情報処理学会論文誌，Vol.41, No.6, pp.1688–1696 (2000).
5) Sumimoto, S., Tezuka, H., Hori, A., Harada, H., Takahashi, T. and Ishikawa, Y.: High Performance Communication using a Commodity Network for Cluster Systems, *9th International Symposium on High Performance Distributed Computing* (*HPDC-9*), pp.139–146, IEEE (Aug. 2000).
6) 住元真司, 佐藤 充, 中島耕太, 久門耕一, 石川 裕：10 Gb Ethernet を用いた高性能通信機構の設計，情報処理学会研究報告 04-HPC-099（SWoPP'2004）(Aug. 2004).
7) 住元真司, 堀 敦史, 原田 浩, 石川 裕：複数 Ethernet を束ねる Network Trunking 機構の提案と 1,024 プロセッサ PC クラスタ上での性能評価，*HPCS2002*, 情報処理学会 (Jan. 2002).
8) 朴 泰祐, 板倉憲一, 曽根 猛, 三島 健, 中澤喜三郎, 中村 宏：ハイパクロスバ・ネットワークにおける転送性能向上のための手法とその評価，情報処理学会論文誌，Vol.36, No.7, pp.1610–1618 (1995).
9) Boku, T., Sato, M., Ukawa, A., Takahashi, D., Sumimoto, S., Kumon, K., Moriyama, T. and Shimizu, M.: PACS-CS: A large-scale bandwidth-aware PC cluster for scientific computations, *Proc. CCGrid2006*, pp.233–240 (2006).
10) 住元真司, 堀 敦史, 手塚宏史, 原田 浩, 高橋俊行, 石川 裕：高速通信機構 PM2 の設計と評価，情報処理学会論文誌：ハイパフォーマンスコンピューティングシステム，Vol.41, No.SIG 5 (HPS-1), pp.80–90 (2000).
11) Takahashi, T., Sumimoto, S., Hori, A., Harada, H. and Ishikawa, Y.: PM2: A High Performance Communication Middleware for Heterogeneous Network Environments, *Supercomputing 2000*, *IEEE and ACM SIGARCH* (Nov. 2000), Published by CD-ROM (Nov. 2000).
12) 石川 裕：YAMPII もう一つの MPI 実装，情報処理学会研究報告 04-HPC-099（SWoPP'2004），pp.115–120 (Aug. 2004).
13) Sumimoto, S. and Kumon, K.: PM/Ehernet-kRMA: A High Performance Remote Memory Access Facility Using Multiple Gigabit Ethernet Cards, *3rd International Symposium on Cluster Computing and the Grid*, pp.326–334, IEEE (May 2003).
14) Sterling, T., Savarese, D., Becker, D.J., Fryxell, B. and Olson, K.: Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation, *Proc. 4th IEEE Symposim on High Performance Distributed Computing* (*HPDC-95*) (Aug. 1995).
15) 工藤知宏, 松田元彦, 手塚宏史, 清水敏行, 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 (ACS 6), pp.35–44 (2004).
16) The Local Area Memory Port, Local Area

MultiProcessor, Scalable Coherent Interface, and Serial Express Users, Developers and Manufacturers Association.
http://www.scizzl.com/

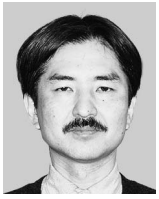
(平成 18 年 1 月 27 日受付)
(平成 18 年 5 月 9 日採録)

住元　真司（正会員）

1986 年同志社大学工学部電子工学科卒業．同年富士通（株）入社．（株）富士通研究所にて並列オペレーティングシステム，並列分散システムソフトウェアの研究開発に従事．1997 年より新情報処理開発機構に出向．コモディティネットワークを用いた高速通信機構の研究開発，RWCP SCore2，SCore3 クラスタ等大規模 PC クラスタ開発に従事．2002 年より（株）富士通研究所にて高速通信機構の研究開発，理研スーパーコンバインドクラスタ等大規模 PC クラスタ，UHPC システムの開発等に従事，並列分散システムのアーキテクチャ，システムソフトウェア等に興味を持つ．平成 12 年度情報処理学会論文賞受賞，工学博士（慶應義塾大学大学院理工学研究科）．

大江　和一（正会員）

1988 年九州大学工学部情報工学科卒業．同年富士通（株）に入社．現在，（株）富士通研究所に勤務．ストレージシステムの研究開発を経て，現在 PACS-CS 通信部の研究開発に従事．



久門　耕一（正会員）

1979 年東京大学電気工学科卒業．1981 年同大学大学院電子工学専門課程修士課程修了．1984 年同課程博士課程中退．同年（株）富士通研究所入社．現在，同社 IT コア研究所に所属．CPU，メモリ，並列計算機アーキテクチャに関する研究に従事．GCC，Linux カーネル等の改良にも興味を持つ．日本ソフトウェア科学会会員．

朴　泰祐（正会員）

1984 年慶應義塾大学工学部電気工学科卒業．1990 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．1988 年慶應義塾大学理工学部物理学科助手．1992 年筑波大学電子・情報工学系講師，1995 年同助教授，2004 年同大学大学院システム情報工学系助教授，2005 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．2002 年度および 2003 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）

1959 年生．1982 年東京大学理学部情報科学科卒業．1986 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．1991 年通産省電子技術総合研究所入所．1996 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．2001 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．

宇川　彰

筑波大学教授．数理物質科学研究科物理学専攻．計算科学研究センター長．東京大学卒業（1972 年）．理学博士（1977 年）．コーネル大学，CERN，プリンストン大学，東京大学原子核研究所を経て，1985 年より筑波大学に勤務．専門は，格子上の場の理論による素粒子物理学の理論的研究と計算科学のための並列計算機の開発．CP-PACS および PACS-CS の開発・製作に従事．仁科記念賞（1994 年）．