DIMM スロット搭載型ネットワークインタフェース DIMMnet-1 とその高バンド幅通信機構 BOTF

田邊 昇[†,☆] 山本淳二[†,☆☆] 濱田芳博[††]
中條拓伯[††] 工藤知宏[†] 天野英晴[†††]

我々は DIMM スロット搭載型ネットワークインタフェース DIMMnet-1 を開発中である．DIMMnet-1 は BOTF という高バンド幅通信機構を装備している．これにより，1 GHz の Pentium3 ベース PC 上での同時送受信継続バンド幅として各方向 1,022 MB/s が実現可能で，66 MB/s を理論的限界とする標準的な PCI バスに搭載されたネットワークインタフェースの 10 倍以上の性能向上が得られることを示す．さらに 1.33 GHz の Athlon ベース PC 上での実験評価により，DIMMnet を DDR-DIMM に対応させることにより，同時送受信継続バンド幅を各方向 1,882 MB/s にまで改善できることを示す．

BOTF: A High Bandwidth Communication Mechanism of DIMMnet-1 Network Interface Plugged into a DIMM Slot

Noboru Tanabe,[†,☆] Junji Yamamoto,[†,☆☆] Yoshihiro Hamada,[††]
Hironori Nakajo,[††] Tomohiro Kudoh[†] and Hideharu Amano[†††]

A high performance network interface architecture for PC clusters in order to overcome the limitation of PCI bus and communication overhead is presented. We propose a network interface plugged into a DIMM slot with BOTF sending mechanisms. A prototype called DIMMnet-1 and its implementation are also presented. DIMMnet-1's simultaneous sending and receiving bandwidth evaluated by measuring data copying performance on 1 GHz Pentium3 based PC is 1,022 MB/s for each dirrections. This performance is more than ten times faster than NIC plagged into standard PCI bus whose thoretical limit is 66 MB/s. The effect of supporting DDR slot on the future DIMMnet is evaluated on Athlon 1.33 GHz based PC. The result shows 1,882 MB/s for each directions can be realized.

1. はじめに

近年，高性能 PC を多数用いて並列処理を行う，いわゆる PC クラスタが注目されている．高性能な PC クラスタ用に Myrinet [1]，SCI-PCI [2]等の高速ネットワークインタフェース（NIC）が各種[3),4)]開発されており，これらはいずれも PCI バスに接続される．光インタコネクションの持つ大きなバンド幅を有効に活用するには従来の PCI バスではバンド幅および遅延ともに力不足である．

一方，Infiniband [8]が次世代のサーバ向け入出力の規格として提案され，製品が開発されつつある．しかし，最も価格性能比においてメリットのあるエンドユーザ用の量産 PC に，Infiniband が普及するかどうか不透明である．GigaE PM2 [5]を用いるなどしてすべてをコモディティ部品で構築するシステムよりも十分優れた性能を実現しつつ，価格性能比を最大にする PC クラスタを構築するためには，Infiniband 等とは別のアプローチも検討に値する．

このような背景から我々は，従来のように PCI バス等の入出力バスではなく，メモリスロットに搭載されるタイプの NIC を検討してきた．このようなクラスの NIC を MEMOnet [16),18)]と名付けた．MEMOnet は安価な PC 上で，PCI バスのバンド幅や遅延時間の限界を超越した NIC を実現可能とすると思われ

† 新情報処理開発機構
Real World Computing Partnership
†† 東京農工大学
Tokyo University of Agriculture and Technology
††† 慶應義塾大学
Keio University
☆ 現在，株式会社東芝，研究開発センター
Presently with Corporate Research and Development Center, Toshiba, Corp.
☆☆ 現在，株式会社日立製作所
Presently with Hitachi, Ltd.

る．我々は MEMOnet のプロトタイプとして DIMM（Dual inline memory module）スロットに搭載される DIMMnet-1 を開発中である．

本論文ではまず，バンド幅の観点から従来型 NIC の問題点を列挙する．これらが改善された NIC のプロトタイプである DIMMnet-1 の概要を紹介し，DIMMnet-1 においてはこれらがどのように解決されているかを解説する．次に DIMMnet-1 の同時送受信継続バンド幅を予測する．最後に，DDR-DIMM に対応した場合の効果について述べる．

2. 従来型 NIC のバンド幅に関する課題

現在流通している製品レベルの PC クラスタ用 NIC は PCI バスベースであり，下記に示す種々のバンド幅に関する課題をかかえている．

2.1 ピークバンド幅不足

SCI-PCI[2)]や RHiNET[6)]のように 10 GBps 程度の通信リンクは光インタフェースを用いたり，比較的短距離であれば Synfinity[7)]のように先進的な電気的インタフェースを用いれば可能である．しかし，PCI バスは 133 MB/s，64 bit66 MHzPCI でも 532 MB/s にすぎず，これらの通信リンクを効率的に動作させるためにはバンド幅が足りない．安価なコンシューマ用 PC には 64 bit66 MHzPCI や PCIX 等のサーバ用の I/O バスが現状では搭載されておらず，将来的にも搭載されるか不透明である．よって当面は安価なコンシューマ用 PC では 133 MB/s という制約を受ける．

2.2 同時送受信継続バンド幅不足

PCI ボトルネックは，**図 1** に示されるような「送信と受信を同時に行う」という，SPMD 型アプリケーションの実行において頻繁に発生する状況で最も顕著になる．図 1 で明らかなように，いかにメモリバンド幅が高バンド幅になろうとも，いかに通信リンクが高速になろうとも，NIC が PCI バスに搭載される以上，PCI バスのたかだか半分の 66 MB/s で律速されてしまう．実際には PCI バスが半二重通信路であるため方向切替えが頻発し，そのオーバヘッドによってピークの半分のバンド幅も出すことが困難である．

2.3 主記憶アクセスの競合

PCI バスからの DMA アクセスも，CPU からの主記憶アクセスも同じ主記憶へのアクセスとなるので，競合が発生する．通常の PC の主記憶のバンド幅は 800 MB/s〜1 GB/s 程度で，そこに光リンク等からの 1 GB/s ものデータを受信しようとすると，それだけでバンド幅が足りなくなり，ホスト CPU の処理速度低下を引き起こす．

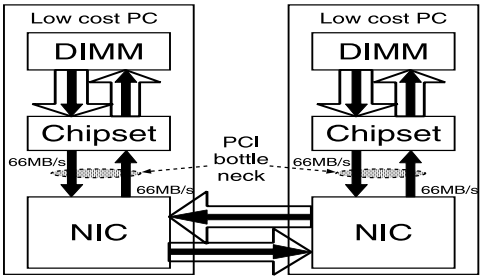


図 1 PCI 型 NIC における同時送受信
Fig. 1 Simultaneous send & receive on PCI based NIC.

2.4 各種 PCI カードとの競合

イーサカード，SCSI カード，ビデオカード，ビデオキャプチャカード等の PCI カードとも PCI バスバンド幅や，主記憶バンド幅を分け合わなければならない．たとえば，ビデオキャプチャカードと Myrinet でフルサイズ，フルレートの動画転送（約 35 MB/s）を主記憶経由で行おうとすると，PCI のバンド幅を使いきる．

2.5 NIC メモリへのアクセス集中

代表的な従来型 NIC である Myrinet において，送信と受信を並行して行わせるならば NIC 上のメモリ（NIC メモリ）をアクセスする DMA 転送が 4 種類（主記憶 ⇔ NIC メモリ，NIC メモリ ⇔ 通信リンク）と，NIC 上のプロセッサからの命令フェッチ，データフェッチがすべて 1 カ所の NIC メモリ（Myrinet の場合は小容量高速 SRAM）に集中する．このため，通信リンクを 8 倍のバンド幅に引き上げたら，Myrinet のような構成では NIC メモリに必要なバンド幅を確保することが困難である．

特に通信リンクのバンド幅が I/O バスのバンド幅より大きい場合は，通信リンクから入力されるパケットを NIC メモリを介さず主記憶上に受けようとすると NIC あるいはネットワークに存在する小容量バッファを溢れさせてかえってパフォーマンスダウンにつながることが予想される．このため，通信は必ず NIC メモリ経由とならざるをえず，このことは NIC メモリのバンド幅不足に拍車をかける．

2.6 細粒度通信におけるバンド幅不足

CPU のレジスタにすべて保持することのできる程度の少量のデータを，低オーバヘッドで送受信することができるならば，計算処理フェーズと通信フェーズを完全に分けることなく，通信を起動後，相手にデータが届くまでの間に次の計算処理をすることを効率的に行える．いい換えると，1 つの通信方式における低遅延と高バンド幅の両立が重要である．ところが，従

来の NIC では通信オーバヘッドが大きく，このような細粒度な通信を行うとかえって効率が悪くなる．通常，uncached 属性の領域にマップされる DMA コントローラへの制御用データの受け渡しに時間がかかる．そのうえ，通信の完了通知もポーリングではなく DMA で行わないと PCI バス上での競合により送受信のバンド幅低下を引き起こすので，遅延が犠牲となり，細粒度通信におけるバンド幅が不足する．

2.7 高遅延ゆえの通信時期の集中

従来の NIC では通信起動時間が短くても数 μs 程度かかるので，メッセージのサイズを大きくして通信回数を減らすためにある程度同じ行き先のデータが溜まるまで計算を行ってから通信を行わざるをえない．よって SPMD タイプのアプリケーションでは通信の集中するフェーズと，通信が発生しないフェーズがくっきり分かれがちとなる．このため実質的なバンド幅を低めてしまう．さらに，結果をまずメモリに溜め込み，パッキングしてから送り，受信側でもアンパッキングが必要なので，不必要な主記憶バンド幅消費や，パッキング・アンパッキングオーバヘッドにより実質通信バンド幅の低下を引き起こす．

2.8 I/O バスの標準化の弊害

通常のパーソナルユースにおいては，CPU の著しい速度向上に沿った形で増強されるメモリバスへの要求と異なり，I/O バスへのバンド幅に対する要求は比較的小さい．このため，PCI バスが長い間 PC の I/O バスのデファクトスタンダードとして君臨したまま，なかなか進歩しなかった．一方，PC クラスタ用 NIC は CPU 速度に応じた高いバンド幅が必要である．このため，NIC が PCI バス上に搭載される限り，NIC がホストとの間でやりとりするためのバンド幅とホスト CPU の速度の間のバランスを維持することが困難である．

3. DIMMnet-1 プロトタイプ

我々は 32 bit33 MHzPCI バススロットしか搭載されていない低コストなパソコン上で，PCI 型 NIC の十倍以上高速な 1 GB/s 程度の同時送受信継続バンド幅を実現できるならば画期的であると考え，それを目標とした．そのために後述する MEMOnet や Block On-the-fly（BOTF）送信[17),19)]等のアーキテクチャを考案し，その有効性を実証すべく，DIMMnet のプロトタイプ DIMMnet-1 を開発した．本章ではその概要を述べる．

3.1 DIMMnet-1 の概要

DIMMnet-1 は，PC100 または PC133 仕様の DIMM スロットに装着するネットワークインタフェースである．DIMMnet-1 の主な仕様を**表 1** に，その基本構造を**図 2** に示す．後述する Martini LSI は低遅延の FET バススイッチにより 2 個の SO-DIMM（ノート型 PC で用いられる汎用部品）を切り替えてホスト CPU とネットワーク間で共有される大容量な分散共有メモリを構成し，リンクインタフェースとデータの送受信をする．DIMM スロットの信号をじかに入力する DIMM 型 NIC 制御ポートを有する．メモリバス側のインタフェースは日本電子機械工業会規格の「プロセッサ搭載メモリ・モジュール（PEMM）動作仕様標準」[9)]に準拠した．PEMM 規格準拠のチップセットやマザーボードは現状では存在しないので，PEMM 準拠モード以外にも，PEMM で追加された 2 つの信号（メモリへのアクセスを待たせる信号と割込み信号）がなくても動作するモードの 2 つのモードを有する．

表 1 DIMMnet-1 の主な仕様
Table 1 Basic specifications of DIMMnet-1.

ホストとのインタフェース	DIMM および PEMM
NIC メモリ（大容量共有メモリ）	PC133，SO-DIMM2 枚
搭載可能 SO-DIMM 容量	64 MB〜1 GB
低遅延共有メモリ（LLCM）容量	128 KB（Martini LSI 上）
命令 SRAM 容量	128 KB（Martini LSI 上）
データ SRAM 容量	128 KB（Martini LSI 上）
オンチップ CPU	R3000 風 32 bitRISC
通信リンクバンド幅	各方向 8 GBps（全二重）
NIC メモリバンド幅	1024 MB/s（ホスト側） 1024 MB/s（network 側）
最短送信時 NIC 遅延時間	105 ns（DIMM〜リンク）
最短受信時 NIC 遅延時間	90 ns（リンク〜LLCM）
NIC-LSI のテクノロジ	0.14 μmCMOS

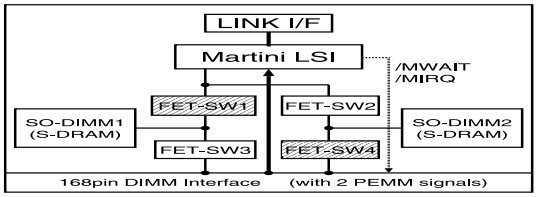


図 2 DIMMnet-1 の基本構造
Fig. 2 Basic structure of DIMMnet-1.

3.2 DIMMnet-1 の基板とスイッチ

DIMMnet-1 は**表 2** に示される大別して 3 種類のスイッチおよび DIMMnet-1 どうしが接続可能である．それぞれのスイッチに合わせたインタフェースを搭載する 3 種類の基板を想定しており，現在までに電気版と光版各 1 種類の計 2 種類の DIMMnet-1 基板が作成された．

電気版プロトタイプである DIMMnet-1/e（**図 3**）の基板は電気版の RHiNET-2/SW および OIP（Optical

表 2 DIMMnet-1 に接続可能なスイッチの仕様
Table 2 Specification of switches for DIMMnet-1.

スイッチ	RHiNET2 [10]	RHiNET3 [11]	OIP-SW [12]
光 port	8 (or 0)	8	15 (or 16)
電気 port	0 (or 8)	0	1 (or 0)
I/O ピン	800 MBps×9	1250 MBps×8	250 MBps×9
バンド幅	8 GBps	10 GBps	2.5 GBps
距離（光）	100 m	1 km	100 m
距離（電気）	3 m	-	5 m
再送制御	N/A	OK	N/A
Table routing	OK	OK	N/A
Source routing	N/A	OK	OK
開発元	RWCP & 日立	RWCP & 日立	NEC & RWCP

図 3 DIMMnet-1/e
Fig. 3 DIMMnet-1/e.

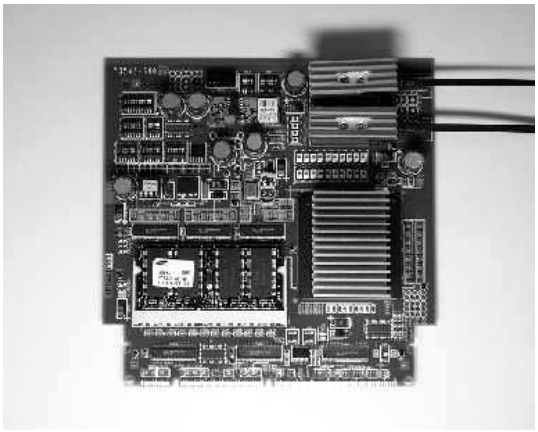


図 4 DIMMnet-1/o2
Fig. 4 DIMMnet-1/o2.

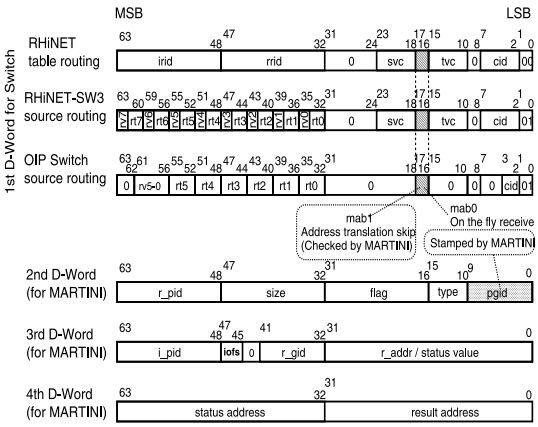


図 5 DIMMnet-1 のヘッダフォーマット
Fig. 5 Packet format of DIMMnet-1.

IP）スイッチ[12]と LVDS（Low Voltage Differential Signaling）レベルの電気信号を用いたケーブル接続により接続可能である．

光版プロトタイプである DIMMnet-1/o2（**図 4**）の基板は，RHiNET-2 用の光インタフェースを搭載しており，光版の RHiNET-2/SW との接続が可能である．

3.3 Martini LSI

Martini LSI[20]は，PCI バスベースの RHiNET-2/NI と DIMM スロットベースの DIMMnet-1 の機能を 1 チップで実現する NIC 制御チップである．リモート DMA による通信のほかに，PIO（Programmed I/O）通信の一種として低遅延性を最優先させた Atomic On-the-fly（AOTF）通信や低遅延性と高バンド幅と柔軟性を両立する Block On-the-fly（BOTF）通信の 3 種類の送信方式をサポートしている．低遅延と高バンド幅が要求される単純なデータ転送はハードウェアのみによりサポートし，メッセージ交換の SEND/RECEIVE，ロック，バリア，同期通信等の機能はチップ内に実装されたコアプロセッサにより実現する．モジュール（Martini LSI 内で独立制御されているブロック）単位のパイプライン化と代行機能[21]により，コアプロセッサは，ハードウェアの一部を動作させながら，処理に介入することが可能であり，柔軟なソフトウェア/ハードウェア処理分担が可能となっている．

3.4 パケットヘッダフォーマット

DIMMnet-1 の 32 bit アドレス版パケットのヘッダフォーマットを**図 5** に示す．DIMMnet-1 のパケットヘッダフォーマットは RHiNET のパケットヘッダフォーマットと同一である．このほかに 64 bit アドレスに対応したフォーマットも用意されている．なお，送信側で BOTF を用い，受信側でコアプロセッサによるソフトを介した受信を用いることにより，先頭の 10 バイト以降の領域も受信側アプリケーションで利

表 3 PM-D API の概要
Table 3 Summary of PM-D API.

NIC メモリ領域	_pmAlloc	割当て
	_pmFree	解放
NIC メモリバンク	_pMBank	状態を得る
	_pmBLock	ロック
	_pmBRelease	ロックの解除
AOTF	_pmOpenA	書き込み口生成
	_pmMapA	ヘッダシードの書き込み口へのマッピングと HTLB（Header TLB）への登録
	_pmUnmapA	書き込み口のアンマッピングと HTLB からの削除
	_pmCloseA	書き込み口の閉鎖
BOTF	_pmOpenB	書き込み口生成
	_pmCreateB	ヘッダシードの生成
	_pmCloseB	書き込み口の閉鎖

用できる柔軟性を有している．

3.5 API

DIMMnet-1 における Martini LSI 固有の機能（AOTF 送信，BOTF 送信，バンク切替え，NIC メモリ領域）を利用するための API である PM-D API の主な API を**表 3** に示す．PM-D は PM API[13]への機能拡張を意図して設計された．DMA 通信等その他の機能はおおむね RHiNET 用 API に準拠し，それらの上に PM API や MPI 等の上位レイヤを構築する予定である．

4. 提案アーキテクチャとその実装

4.1 MEMOnet

4.1.1 MEMOnet の定義と分類

MEMOnet とは筆者らによって 1999 年 8 月より提唱されている NIC のクラスである．従来のように PCI バス等の I/O バスに接続されるのではなく，「主記憶が搭載されるメモリスロットに接続される NIC」と定義されている．

MEMOnet には基盤となる汎用パソコンの主記憶の実装仕様で大きく分けて 3 つ（SIMM，DIMM，RIMM）が考えられる．SIMM（Single inline memory module）形式のものを SIMMnet，DIMM（Dual inline memory module）形式のものを DIMMnet，RIMM（Rambus inline memory module）形式のものを RIMMnet と呼んでいる．MINI[14]は SIMMnet の一例，DIMMnet-1 は DIMMnet の一例である．

DIMM の中には EDO（Enhanced data output）-DRAM を用いたもの，SDR（Single Data Rate）-SDRAM を用いたもの，DDR（Double Data Rate）-SDRAM を用いたものなど種々のものが存在するが，DIMMnet もこれらに応じてに細分化できる．最初のプロトタイプである DIMMnet-1 は，現在最も普及している SDR-SDRAM ベースの DIMM スロットに搭載される DIMMnet である．

4.1.2 MEMOnet における 2 ポート メモリ構造

MEMOnet はホスト側からは主記憶と同様なメモリとして見えている必要があり，ネットワーク側からもアクセス可能なメモリとして見えている必要がある．つまり，MEMOnet は本質的に 2 ポート メモリの機能を内在することになる．

ネットワークのバンド幅が主記憶バンド幅に匹敵する高さを持つ場合は，NIC メモリのホスト側バンド幅とネットワーク側バンド幅は独立に提供されることが望ましい．MINI[14]や，さらに古くは Prodigy のホストインタフェース[15]のように画像用 2 ポート メモリを用いて NIC メモリの独立 2 ポート化を実現した例もあるが，低コストでより高いバンド幅を実現するために DIMMnet-1 ではダブルバッファ的制御がなされる 2 枚の SDRAM ベースの SO-DIMM により構成される．このため，SRAM で実装された Myrinet の NIC メモリと比べて大容量化が容易であり，バンクを切り替えた瞬間から，その直前まで受信領域だった NIC メモリの領域が，ネットワークから隔離されたホストの主記憶に変化する．このため，NIC メモリから主記憶への受信データ書き戻しが必要だった Myrinet 等のように 2.5 節において述べた NIC メモリへのアクセス集中の問題が緩和される．

さらに DIMMnet-1 では，Martini LSI 上に受信用に低遅延共有メモリ（LLCM）という SRAM ベースの 2 ポート メモリを 128 KB 有しており，こちらはバンク切替えなしにホストとネットワークの双方からアクセスが可能である．LLCM も NIC メモリへのアクセス集中の問題を緩和する．

4.2 Block オンザフライ（BOTF）送信

Block On-the-fly（BOTF）送信は，後述するプロテクション刻印ウィンドウに対してユーザモードで連続して書き込まれる一連のデータに，プロテクション情報を付加して，ネットワークに送信する低遅延で高バンド幅な通信である．

ほとんどパケットそのものに近い状態でホスト CPU から NIC のハードウェアに渡されるので，NIC の回路が簡素で済み，かつ少ないクロック数でネットワークにパケットを出力できる．

さらに，この送信は NIC メモリへのアクセスをともなわない実現が可能なために，2.5 節において述べたメモリ NIC へのアクセス集中の問題が一段と緩和する．

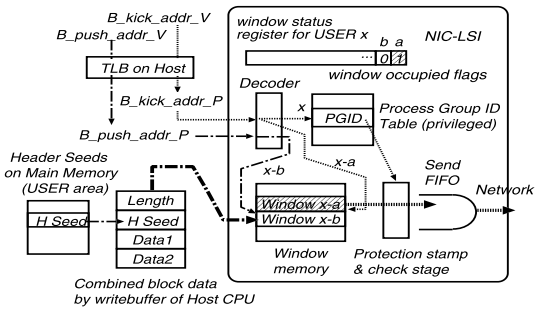


図 6 Block オンザフライ（BOTF）送信
Fig. 6 Block on-the-fly sending.

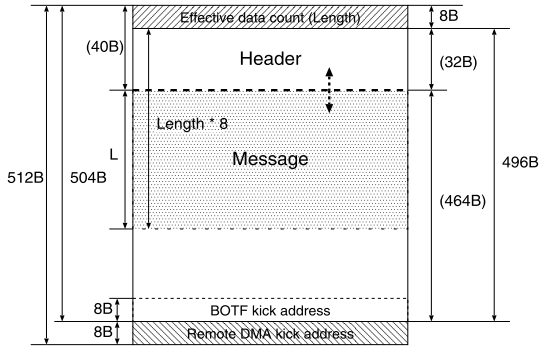


図 7 ウィンドウ内のマッピング
Fig. 7 Mapping in a window for BOTF.

BOTF 送信機能は DIMMnet-1 のみならず，Martini LSI を用いた PCI バスベースの NIC である RHiNET-2/NI や RHiNET-3/NI でも利用可能である．

4.2.1 プロテクション刻印ウィンドウ

BOTF 送信におけるパケット生成メカニズムを**図 6** に示す．図 6 中の window memory と protection stamp & check stage から構成されるプロテクション刻印ウィンドウは，ユーザモードでコピーされる複数ダブルワードにわたる一連の書き込みデータにプロテクション情報を付加して，パケットに変換する機能を有するメモリである．

プロテクション刻印ウィンドウは，複数の領域に分割され，異なるウィンドウはそれぞれが物理アドレス上で異なるページにマップされている．こうして_pmOpenB() のような特権モードで実行される初期化関数の中で，この物理ページは他のプロセスから隔離されるように仮想空間にマップすることが可能となる．DIMMnet-1 の場合は Martini LSI 上に**図 7** で示されるような 512 バイトのウィンドウが 64 個実装される．ウィンドウの一部はリモート DMA 送信の際の通信制御情報の通知にも利用され，最後の 8 バイトは BOTF では用いることができない．

ウィンドウの物理アドレスとプロセスグループ ID（PGID）と呼ばれるプロテクション情報の対応関係を保存する PGID テーブル（図 6 の Process Group ID Table）も 64 エントリ有する．

ステータスレジスタ（図 6 の window status register）はウィンドウへの上書き可否状態を示すビット（図 6 の window occupied flag）を有する．上書き可能状態であるウィンドウ（図 6 の window x-b）上に，ホスト CPU より望ましくは連続的に仮想アドレス（図 6 の B_push_addr_V）をインクリメントさせてパケットのイメージ（図 6 の Length，H-Seed，Data1，Data2）を書き込む．するとホスト CPU の TLB によって物理アドレス（図 6 の B_push_addr_P）に変換される．さらに，ホスト CPU のライトバッファによってバーストアクセスにまとめあげられて，目的のウィンドウに高速にコピーされる．

このウィンドウ上に書き込まれたデータから BOTF 送信を起動することを BOTF キックと呼び，DIMMnet-1 の場合はウィンドウの最後から 2 番目のダブルワードに対応するアドレス（図 7 の BOTF kick address）に何かを書き込むことによって指示される．

よって，もし，ホスト CPU がデータをウィンドウに途中までコピーしただけで，BOTF キックがなされる前にホスト CPU 上でプロセススイッチが行われても，途中までコピーされたデータはユーザごとに異なる場所に保存されているので，切り換わったプロセスによってそのデータが破壊され，複数プロセスからのパケットが混ざってしまうことが回避される．

BOTF キックが検知されると，該当するウィンドウへの上書き可否状態を示すビット（図 6 の window occupied flag の b）が上書き不可能状態に変化する．さらにウィンドウ上に書かれたヘッダシードと呼ばれるパケットヘッダ生成用のデータの PGID フィールド（図 5 の pgid）に，ウィンドウに対応した PGID が刻印される．BOTF ではヘッダシードがユーザモードで書き込まれるため，プロテクション上の理由から，ウィンドウ上に書かれたヘッダシード上のフラグ（図 5 の mab1）が，物理アドレス指定になっていないことをチェックする．

こうしてプロテクション情報が付加されて完成したヘッダとウィンドウ上で後続するデータをウィンドウの先頭ダブルワードに書き込まれたダブルワード数（図 6 の Length）だけパケットとしてネットワークに送信する．DIMMnet-1 においては 1 回の BOTF キックによりヘッダ込みで 496 バイトまでのデータを運ぶパケットを生成することが可能である．

パケットのネットワークへの送信が終了するとステータスレジスタの該当するウィンドウへの上書き可否状態を示すビットが上書き可能状態に変化する．

4.2.2 ウィンドウへのバンド幅維持対策

BOTF 送信は細粒度通信の遅延時間の低下だけでなく，高バンド幅を実現することも目的に設計された通信方式である．BOTF 送信はホスト CPU によるコピーに基づく一種の PIO（Programmed I/O）である．通常，PIO は DMA に比べ低バンド幅な通信方式とされてきた．

しかし，最近の CPU はライトバッファの装備や CPU 動作周波数の高速化にともない，PIO による連続アドレスへの書き込みが大変高速になってきている．CPU の動作周波数向上の凄まじい速さに対する I/O の進歩の鈍さから，従来のように PIO は低速で DMA は高速という常識が必ずしも成り立たないケースがありうるようになってきた．ただし，何ら工夫しなくても PIO 性能が大幅に向上するわけではなく，ライトバッファやキャッシュ等の CPU のアーキテクチャをふまえたうえで，それらを制御する命令を駆使する等の主にソフトウェア的な対策を行うことによって高いバンド幅を維持できる．具体的には，以下の対策が有効であると考えており，その効果については 5.1.2 項で述べる．

- **ダブルバッファリング**

ウィンドウはコンテクストスイッチによる複数プロセスからのデータの錯綜を防止するために FIFO ではなく，アドレッシング可能なメモリで構成されている．このため，BOTF キックを実行後にウィンドウ上のデータがパケット化されてネットワークに出力し終わるまでは，別のパケットのためのデータを同一ウィンドウに書くべきではない．よって，ウィンドウへの書き込みフェーズと，ほぼそれと同様な時間を消費するネットワークへの出力フェーズが存在するために，1 つのプロセスが 1 つのウィンドウにしか書き込めない場合は実質的にはバンド幅が半減してしまう．そこで，1 つのプロセスが利用できるウィンドウは少なくとも 2 つ以上にして，CPU が書き込んでいるフェーズのウィンドウと，BOTF 送信制御部がパケット化してネットワークに送信しているフェーズのウィンドウが別になるようにして用いることで，バンド幅の半減を防止できる．

- **並列ステータス確認**

さらに，ウィンドウが送信終了状態になり，上書きが可能になったかどうかを確認してからでないと，書き込みフェーズに移るべきではないので，ホスト CPU からのステータスレジスタのチェックが必要になるが，このチェックには非キャッシュ領域へのリードに必要な時間が必要なので，これが実効バンド幅を低下させる要因となる．そこで，望ましくは 2 個のウィンドウをフル稼働の状態で用いるのではなく，それ以上のウィンドウを 1 つのプロセスが利用できるようにすることを推奨する．ステータスを 1 回読むときに並列して複数のステータスビットを読み，先行してチェックしたウィンドウが遊休状態であったならば，直前に書いていたウィンドウが書き終わったら即座に，遊休状態であることが確認できているウィンドウに書き込むという制御を行うことによって，実効バンド幅を高めることが可能である．DIMMnet-1 では 64 個すべてのウィンドウのステータスが 1 回のリードアクセスで確認できるようになっているので，空きウィンドウがなくなるまでステータスチェックをしないことによってステータスチェックのオーバヘッドを減らせる．

- **Write combining**

通常 I/O デバイスはホスト CPU から見て uncached（UC）属性にすることが多いが，ウィンドウを write combining（WC）属性の領域として設定することが望ましい．write combining 属性の領域へのアクセスはキャッシングされず，ホスト CPU のライトバッファによってバーストアクセス化される．これは Intel の Pentium Pro 以降の IA32 アーキテクチャに定義されている領域の属性であり，MTRR と呼ばれるレジスタを操作することによって特定の領域を write combining 属性に設定することができる．Linux でもカーネルのオプションを適切に設定することで，その操作ができるようになっている．DIMMnet-1 ではウィンドウを write combining 属性に設定可能にするために，uncached 属性に設定されるべきステータスレジスタ等のその他の内部資源とは異なるページに物理アドレスが入るようにしている．

- **64 bit，128 bit レジスタ**

Athlon でも利用可能な MMX 命令で用いる 64 bit 幅のレジスタや，Pentium3 の場合は SSE 命令セットで追加された 128 bit 幅のレジスタを PIO の際のデータ転送に用いることによって PIO 転送性能を向上させることができる．

- **ループアンローリング**

NIC への PIO 転送を行う際のデータ保持にレジスタを 1 個だけ使って，単純なループでブロックデータを NIC にコピーするよりも，複数のレジスタを用いてそのループをアンローリングすることによって，バンド幅をいっそう向上することができる．

• **プリフェッチ**

送信するべきデータがキャッシュにはなく，主記憶上にある場合は，実効バンド幅を向上させるためには，Intel の SSE 命令セットや AMD の 3D Now 命令セットで追加された PREFETCHNTA 命令等を用いて，主記憶へのリード遅延時間を隠蔽するようにすべきである．その際，プリフェッチのサイズと時期は主記憶のリード遅延を考慮して，適切に選択されるべきである．なぜなら，必要以上に大量のデータをプリフェッチしてしまうとソフトウェアパイプライニング時のプロローグ部分のオーバヘッドが大きくなるとともに，後半でプリフェッチしたデータが前半でプリフェッチしたデータを追い出しかねないためである．書き出す前にプリフェッチが終わっているだけの最小限なサイズと時期でプリフェッチをかけることが望まれる．

• **オンキャッシュ転送**

送信すべきデータがキャッシュ上にない場合は，まずは主記憶上からレジスタに読み出し，次にそれをウィンドウ上に書き出すことになる．この場合，主記憶からの読み出しも，ウィンドウへの書き出しもホスト CPU の FSB（Front side bus）やメモリバスのバンド幅を取り合う形になるので，継続的なバンド幅は上限で FSB のバンド幅の半分しか出すことができないことになる．一方，並列処理を行っている状況では，計算処理をデータに対して行い，その結果を必要とするノードに送るという状況がごく自然に起こりうる．BOTF 送信では送信オーバヘッドは従来よりも少ないので，通信回数を増加させてでも，キャッシュ上にそのデータが残っているうちに送信をしてしまった方がよいケースが想定できる．もし，キャッシュ上に送るべきデータがあるならば，FSB バンド幅の半分を使って CPU 内部に取り込む必要がないため，FSB のほぼすべてのバンド幅を使ってウィンドウにデータを書き出すことが可能となる．この方式では通信時期が分散されることになるので，ネットワークの輻輳を低減し，システム全体としての実効バンド幅はさらに向上するものと思われる．

4.2.3 API とその動作

BOTF による送信における API とその動作を**図 8** に示す．BOTF 送信に際しては，以下の手順で API を用いて起動される．

(1) _pmOpenB() によりウィンドウやステータスレジスタを，ユーザモードでアクセスできるように仮想アドレス B_win_addr_V および B_stat_addr_V と対応付ける．同時に上記仮想アドレスから一意に決まるウィンドウに対応する Martini LSI 上の PGID テーブルのエントリにプロテクション情報（PGID）を設定する．

(2) ヘッダの第 1 ダブルワードである routing header や，r_pid や i_pid 等のヘッダフィールドを熟知していないユーザは _pmCreateB() により B_header_addr_V から始まるユーザ空間上の領域にヘッダシードを生成する．

(3) ステータスレジスタ（B_win_addr_V）をリードしてこれから書き込もうとしているウィンドウが使用中状態でないことを確認する．

(4) 空き状態であれば，ヘッダシードと送信したいデータをウィンドウ上にコピーする．

(5) 最後にウィンドウの BOTF キックアドレス（先頭から 496 バイト目）に書き込みを行う．

(6) BOTF 送信部により，ウィンドウ上のパケットイメージにユーザに対応するプロテクション情報が刻印され，ネットワークに送信される．送信が終了すると対応するウィンドウのステータスが空き状態に更新される．

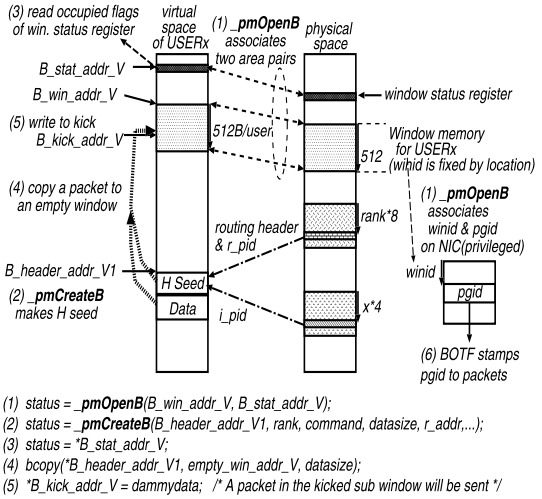


図 8 BOTF 送信のための API の動作
Fig. 8 Behavior of API for BOTF sending.

なお，同様な通信を繰り返す場合は上記の (1)，(2) において下準備した結果を再利用すればよい．

4.2.4 BOTF の用途

BOTF 送信には低遅延，高バンド幅，情報の再利用性，生成できるヘッダの柔軟性等種々の特徴がある．これらを生かす用途としては以下のようなものが考えられる．

• **細粒度な一貫性維持**

ソフトウェア分散共有メモリの実装においてはページフォルトを利用してページレベルでの一貫

性維持を行うタイプ[22]と，コードを事前に解析してソフト的に作られるキャッシュの一貫性維持を行うコードを挿入するタイプ[23]がある．前者では一貫性維持操作が指示されるまでの更新履歴である DIFF を用いて転送量を抑える実装が主流であるが，その場合，ページの汚れ具合が少ないと転送データが短くなり，低遅延な BOTF の効果が期待できる．後者の場合では，より通信の粒度は細かくなるので，BOTF による低遅延通信の効果がいっそう期待できる．

- **細粒度なメッセージのマルチキャスト**
  同一のデータを複数のノードに転送したい場合，BOTF を用いればウィンドウに書かれたパケットイメージを宛先だけ変更してキックすることにより，パケットヘッダや送信データの再利用が行われる．ステータスチェックのオーバヘッドを考慮して適切に用いることにより，一から BOTF を起動するよりもさらに低オーバヘッドで実現できる．
- **低遅延な PULL 要求パケット生成**
  リモートノード上のメモリを読み出す PULL 要求パケットは，ボディを持たないきわめて短いパケットである．その実装においてはたとえば，Martini LSI のハードウェアで実装された PULL プリミティブを用いるよりも，BOTF によってソフト的に生成する方が高速であることが予想される．
- **ソフト的拡張プロトコル用パケット生成**
  Martini LSI では，PUSH と PULL の 2 種類の操作のみ，すべてをハードウェアによって実行するように設計されているが，その他のメッセージ交換やロック，バリア等の実装はリモート側でコアプロセッサに割込みをかけて，そのハンドラによってソフト的に処理される．これらは上記 PULL 要求パケットのようにきわめて短いと同時に，パケットフォーマットに対する自由度が要求される．BOTF の場合は低オーバヘッドであると同時に，ルーティング用ヘッダとプロテクション情報（PGID）以外は新しい拡張プロトコル設計者に自由に使わせることができるだけの自由度がある．
- **ブロック分割時の境界ラインの転送**
  たとえば，二次元配列を行および列方向にブロック分割して各ノードに担当させる場合，隣接するブロック間でブロックの境界部のデータを二重に持って更新しあうことになる．その際，比較的短い連続アドレスに格納されたデータを隣接ブロックのメモリにコピーする必要があるが，そのような短いデータのパケットを BOTF は効率的に送信することができる．

5. 予測性能

5.1 同時送受信継続バンド幅

DIMMnet-1 における BOTF 送信を用いた同時送受信継続バンド幅を考察する．本論文では同時送受信継続バンド幅とは「2 つのノード間で互いにパケットを送りつつ受けるという動作を継続的に行った場合の 2 ノード間でやりとりされる情報のバンド幅（片方向分で表示）」とする．DIMMnet-1 の場合は前述のようにパケットヘッダが柔軟であり，可変長である．そこで本章では，まず 1 GHz の Pentium3 ベースパソコン上の PC133 型 DIMM スロットに搭載された NIC の BOTF 送信方式そのものの限界を与えるヘッダ込みの同時送受信継続バンド幅を実験に基づいて考察する．次に，図 5 に示した RHiNET 互換の 32 ビットアドレシングモードの受信側ですべてをハード処理できる通信を行うケースでのヘッダ（32 バイト）を除いたデータのバンド幅について予測する．

5.1.1 同時送受信継続バンド幅の近似

DIMMnet-1 におけるヘッダ込みの同時送受信継続バンド幅は送信側のホスト CPU によるコピー性能によって近似できる．その理由を以下に列挙する．

- **ウィンドウへの書き込みと吐き出しの並列動作**
  BOTF 送信を行う際，BOTF キックが実行されるまではウィンドウ上のデータは送信されない．ウィンドウ上に未送信データが残っている状態では，そこに次の通信用のデータを上書きしてしまうと先行するパケットが破壊されてしまうので，正しく動作しない．ここで，各プロセスがウィンドウを 1 つしか占有できないとすると，ウィンドウへの書き込みと，ウィンドウからの吐き出しがシリアライズされてしまい，バンド幅が半減する．しかし，DIMMnet-1 では各プロセスがウィンドウを 2 つ以上占有できるように運用することで，これを防止できる．よって，データをウィンドウへ書き込んだときとほぼ同様のバンド幅でネットワークへの吐き出しを行うことが可能である．
- **同時送受信時のデータ経路**
  DIMMnet-1 における同時送受信は**図 9** に示すように，送信側で BOTF 送信することで，TLB 参照機能を有するパイプライン化された受信部と送信側の経路は互いにバンド幅を奪い合わないように 1 GB/s のデータ経路を構成できる．しかも，

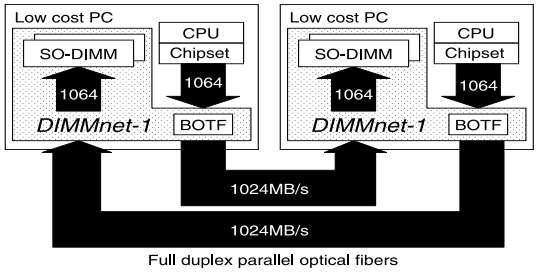


図 9 DIMMnet-1 における同時送受信
Fig. 9 Simultaneous send & receive on DIMMnet-1.

通過するすべての経路で 1 GB/s クラスのバンド幅を有し，前述のようにウィンドウにおけるバンド幅低下も回避できるためハードウェア上のバンド幅のボトルネックがない．よって，送信側でのバンド幅は，同時送受信時の受信側でのバンド幅と等しいことになる．

- **ステータス確認遅延の隠蔽**
  実効バンド幅を制約する要因としては，ウィンドウのステータスフラグの検査のためのソフトウェアオーバヘッドが考えられる．DIMMnet-1 では 1 回のメモリアクセスで 64 個のウィンドウの状態を読むことでができる．このため，1 つのプロセスが複数のウィンドウを用いている場合，1 回のメモリアクセスで複数の空きウィンドウを発見できる可能性があり，ステータスチェックの回数を減らし，実効バンド幅を高めることができる．BOTF での最大バンド幅は 64 個のウィンドウを 1 つのプロセスに使わせる運用をした場合に実現でき，64 回に 1 回のステータスチェック後に連続して 64 個のウィンドウにコピーできる．1 回のステータスチェックには uncached 領域のメモリリード遅延がかかり，これが 1 GHz の Pentium3 と VIA Pro266 の組合せで 109 ns であった．504 バイトの書き込み（約 500 ns 強）を 64 回行うごとに 1 回 109 ns のオーバヘッドが加わるが，0.3%のロスでほぼ無視できる大きさと考えられる．

以上から，500 バイト程度のデータのウィンドウへのコピーバンド幅は，DIMMnet-1 で BOTF を用い，ウィンドウをたくさん使う運用形態で長いデータを 2 ノード間で同時送受信する際のヘッダ込みの同時送受信継続バンド幅の近似値として採用できる．この値は，ヘッダ長をきわめて短く設定した場合の同時送受信継続バンド幅の近似値としても採用できる．

5.1.2 ウィンドウへのコピーバンド幅

4.2.2 項において述べたウィンドウへのコピーバンド幅向上対策を行った場合の 512 バイト書き込みのバンド幅測定結果を**表 4** に示す．表 4 中の UC は uncached 属性，WC は write combining 属性を表す．本実験は内部周波数 1 GHz，FSB 周波数 133 MHz の Pentium3 と VIA 社 Apollo Pro266 チップセットと PC133 ベース SDR-DIMM を用いた PC 上で，OS としては Linux を用いて測定した．

表 4 ウィンドウへの 512 バイトコピーバンド幅
Table 4 Bandwidth for copying 512 Bytes.

	ソース	属性	命令	loop	他の技法	MB/s
1	32b	UC	-	-	-	67
2	reg.	WC	-	-	-	320
3	64b		MMX	-	-	549
4	reg.			2 回展開	-	705
5	main				-	233
6	mem.				prefetch	332
7	cache				on cache	705
8	128b		SSE	-	-	964
9	reg.			2 回展開	-	**1022**
10	main				-	283
11	mem.				prefetch	407
12	cache				on cache	**1004**

ここで，1 → 2 は WC 属性の効果，2 → 3 は MMX 命令（64 bit レジスタの使用）の効果，2 → 8 は SSE 命令（128 bit レジスタの使用）の効果，3 → 4 および 8 → 9 はループアンローリングの効果，5 → 6 および 10 → 11 はプリフェッチの効果，6 → 7 および 11 → 12 はオンキャッシュ転送の効果が判別でき，4.2.2 項の write combining 以降の項目で述べたソフト的なウィンドウへのコピーバンド幅向上対策はすべてにおいて効果があったことが分かる．特に送信すべきデータがすべて CPU のレジスタ上にある場合は，1022 MB/s というきわめて高いバンド幅を実現できることが分かった．

この速度は，ヘッダ分のロスを考慮しても，標準的な 32 bit33 MHzPCI 環境における従来の NIC のみならず，現時点で最新の PC クラスタ向け NIC 製品である Myrinet2000 の単方向通信バンド幅 245 MB/s や，SCI PCI-64/66 の 200 MB/s と比べて圧倒的に高速である．さらに SCI PCI-64/66 が同時送受信時の合計で単方向の倍には程遠い 304 MB/s にしかならないのに対し，DIMMnet-1 の場合は送受信を同時に実行しても上記のほぼ倍の合計バンド幅が得られる点で，これらの 64 bit66 MHz 版 PCI ベースの製品に比べてきわめて高速であるといえる．

送信すべきデータが CPU 側になく，メモリ上にある場合は大きなバンド幅低下が発生する．これはメモリからのリードと，メモリ上への書き出しが同時に発生してバンド幅が半減し，さらにバス方向切換オーバ

ヘッドも加わって性能低下しているものと思われる．

しかし，送信すべきデータがキャッシュ上にあるうちに BOTF 送信をかけてしまえば，ソースデータがレジスタ上にある場合とほぼ同等レベルの高いバンド幅（1004 MB/s）を出すことができることが分かる．

上記から，DIMMnet-1 を用いたシステム用の並列プログラム設計者およびコンパイラ設計者への指針としては「従来のように送信を先延ばししてメッセージ長を伸ばし，送信回数を減らすことを優先するよりも，送信すべきデータがキャッシュに残っている早いタイミングで BOTF によりこまめにデータを送信してしまうことが性能向上の鍵となりうる」ということがいえる．

5.1.3 メッセージ長と利用ウィンドウ数の影響

実際のパケットにはヘッダが含まれるので，ユーザが送信したいメッセージのほかにヘッダを送信するためにバンド幅が消費される．このため，パケット 1 個あたりのメッセージ長が短くなるほどヘッダ転送にかかるオーバヘッドの影響が出る．

また，ユーザあたり利用可能なウィンドウメモリの数が少なくなるほど，ウィンドウのステータスフラグのチェックの頻度が上がるため実効バンド幅が低下する．

上記の 2 つの効果を加味した DIMM 周波数 133 MHz で受信側をすべてハードで処理するタイプの通信を行う場合の DIMMnet-1 の同時送受信継続バンド幅の値は次の式で近似できる．ただし，ウィンドウの数が 1 個になった場合はウィンドウへの書き込みと吐き出しが逐次実行になるので下式の半分の値となる．

$$Bandwidth = \frac{L \times W}{0.109 + 0.038 + \frac{W \times (L+40)}{1022}}$$

ここで L はパケットあたりのメッセージ長，W は 1 ユーザが利用可能なウィンドウ数である．ウィンドウのステータスフラグのチェックのオーバヘッドは実測結果 0.109 μs を用いる．BOTF キックによる 1 ワードの書き込みにともなうバースト転送間のギャップとしてはチップセット上でバースト長 4 が固定されているものとして 5 クロックサイクル分の 0.038 μs を用いる．図 7 で示すように BOTF の転送サイズ指定（8 バイト）とヘッダ（32 バイト）とメッセージ（L バイト）の合計 $L+40$ バイトをホストのソフトにより約 1022 MB/s でウィンドウにコピーできるという近似を行った．実際のバンド幅は 8 バイトで割り切れないメッセージ長の場合はこの式で表される曲線より若干下方向に外れることになる．

上記の式をプロットすると**図 10** のようになる．特にメッセージサイズが短い領域では多くのウィンドウを用いてステータスフラグのチェックのオーバヘッドを薄める効果がバンド幅に大きく出ることが分かる．予測されたバンド幅は 50 バイト以下の短いメッセージにおいても，Myrinet2000 等の最新型の NIC 製品が 10 KB を超えるような長いメッセージでしか近づくことができない PCI バス型 NIC の同時送受信継続実行時の理論的上限バンド幅を大幅に超える値を実現できることが分かる．

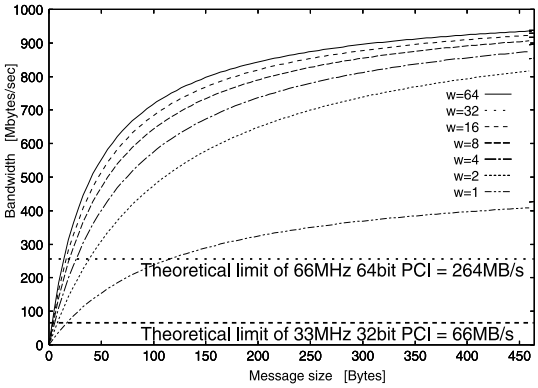


図 10 DIMMnet-1 の BOTF による同時送受信継続バンド幅
Fig. 10 Sustained simultaneous send & receive bandwidth on DIMMnet-1.

5.2 DDR-DIMM への対応

長い間進歩が止まっていた PCI 等の I/O バスと異なり，CPU の性能向上を追うように PC のメモリスロットは急速な性能向上を続けている．DDR-SDRAM ベースの DIMM やそのマザーボードも，個人ユーザが十分購入できるレベルに価格が低下している．さらに DDR-SDRAM ベースの DIMM スロットに対応する Pentium4 用チップセットも開発が予定されている．よって今後は DDR-SDRAM ベースの DIMM スロット搭載の PC が低価格 PC においても主流になってくると思われる．

このような状況を背景に，将来，もし DIMMnet を DDR-SDRAM にも対応させるならば，利用可能なコストパフォーマンスの良い PC が増加するだけでなく，DIMMnet そのものの性能向上も期待できる．

内部周波数 1.33 GHz，FSB 周波数 266 MHz 換算（133 MHz の DDR）の Athlon と ALi 社 MAGiK1 チップセットと PC2100 ベース DDR-DIMM を用いた PC 上で，64 bit レジスタ上のデータを MMX 命令セットにより 64 KB 分 Write Combining 属性のメモリ領域にコピーするバンド幅として 1882 MB/s を観測した．このことは DDR-DIMM に対応した DIMMnet をバンド幅ボトルネックを回避した DIMMnet-1

と同様の構成で作るならば，送信 1,882 MB/s と受信 1,882 MB/s を同時に行える可能性を示している．さらに，SSE 命令セットが利用できる Athlon MP を用いるならば，さらなる性能向上の可能性があると思われる．

一方，Pentium3 ベースの PC では DDR-DIMM を用いたとしても，ほとんど性能向上が見られなかった．その理由は FSB が 133 MHz の SDR 転送であるために，バンド幅が FSB でボトルネックになるためであると思われる．Pentium3 の FSB バンド幅の上限は 1,064 MB/s であるのに対し，Pentium4 では 3,200 MB/s にまで改善され，FSB ネックが解消されている．現時点では実験ができないが，DDR-DIMM に対応した Pentium4 ベースの PC が利用可能になれば，Intel の CPU を用いた PC 上でも大幅なバンド幅向上が DIMMnet の DDR 対応により期待できる．

6. ま と め

バンド幅の観点から従来型 NIC の問題点を明らかにし，その対策として MEMOnet と BOTF を提案し，DIMMnet-1 プロトタイプにおいてはどのように実装され，解決されているかを述べた．DIMMnet-1 では BOTF 送信を用いた場合，従来の NIC が苦手としていた同時送受信継続実行において，1 GHz の Pentium3 のシステム上でヘッダ込みのバンド幅として各方向 1,022 MB/s，RHiNET 互換の 32 バイトヘッダを適用したヘッダ損失やステータスチェックオーバヘッドも考慮したバンド幅として各方向 937 MB/s という 32 bit33 MHzPCI バス上で用いられる NIC の 10 倍以上の高速化が得られることが予測された．さらに DDR-DIMM に対応するならば 1.33 GHz の Athlon により 1,882 MB/s にまで改善できる見通しを示した．今後は，BOTF よりも低遅延性を重視して設計された通信機構である AOTF の評価や，DIMMnet-1 の実機上での評価と，ソフトウェア環境の整備を進める予定である．

謝辞 新情報処理開発機構の西氏，慶應義塾大学の土屋氏，渡辺氏，(株) 日立 IT の今城氏，上嶋氏，金野氏，寺川氏，慶光院氏，岩田氏，山本氏，柏原氏，大杉氏をはじめ Martini LSI および DIMMnet-1 の開発に携わったすべての方々に感謝いたします．

また，本研究は新情報処理開発機構が推進した RWC（Real World Computing）プロジェクトの並列分散コンピューティング技術研究の一環として行われた．

参 考 文 献

1) Myricom Corp. http://www.myri.com/
2) Dolphin Corp.: PCI-SCI Adapter Card D320/D321 Functional Overview, Part No.: D1950-10299 (1999.11).
3) Fillo and Gillett: Architecture and Implementation of MEMORY CHANNEL 2, *Digital Technical Journal*, Vol.9 (1) (1997).
4) Emulex Corp.: cLAN Hardware Installation Guide, Part No.: CLAN-D001-001
5) 住元，堀，手塚，原田，高橋，石川：GigaE PM II：Gigabit Ethernet による高速通信ライブラリの設計，情報処理学会計算機アーキテクチャ研究会，Vol.99, No.67, pp.61–66 (1999).
6) Kudoh, Nishimura, Yamamoto, Nishi, Tatebe and Amano: RHiNET: A network for high performance parallel processing using locally distributed computers, *IWIA'99* (1999.11).
7) 田村，後藤，Sastry： 高速信号伝送技術：Synfinity II, *FUJITSU*, Vol.50, No.4, pp.235–241 (1999).
8) InfiniBand Trade Association. http://www.infinibandta.org/
9) 日本電子機械工業会：日本電子機械工業会規格：プロセッサ搭載メモリ・モジュール（PEMM）動作仕様標準，EIAJ ED-5514 (1998.7).
10) 西，多昌，西村，山本，工藤，天野：LASN 用 8Gbps/port 8x8 One-chip スイッチ：RHiNET-2/SW，2000 年記念並列処理シンポジウム（JSPP2000），pp.173–180 (2000.5).
11) 西，上野，多昌，稲沢，西村，工藤，天野：LASN 用 10Gbps/port 8x8 ネットワークスイッチ：RHiNET-3/SW，情報処理学会計算機アーキテクチャ研究会，Vol.2000，No.110，pp.13–18 (2000)
12) Yoshikawa and Matsuoka: Optical Interconnections for Parallel and Distributed Computing, *Proc. IEEE*, Vol.88, No.6, pp.849–855 (2000).
13) Tezuka, Hori, Ishikawa and Sato: PM: An Operating System Coordinated High Performance Communication Library, *High Performance Computing and Networking'97* (1997).
14) Minnich, Burns and Hady: The Memory Integrated Network Interface, *IEEE Micro*, Vol.15, No.1 (1995.2).
15) 田邊：マルチプロセッサシステム，公開特許公報，特願平 2-157491（出願 1990.6），特開平 4-48371（公開 1992.2）．
16) 田邊，山本，工藤：メモリスロットに搭載されるネットワークインタフェース MEMnet，情報処理学会計算機アーキテクチャ研究会，Vol.99, No.67, pp.73–78 (1999).

17) 田邊，山本，工藤： メモリスロット搭載型ネットワークインタフェース DIMMnet-1 における細粒度通信機構，情報処理学会計算機アーキテクチャ研究会，Vol.2000, No.23, pp.65–70 (2000).
18) Tanabe, Yamamoto, Nishi, Kudoh, Hamada, Nakajo and Amano: MEMOnet: Network interface plugged into a memory slot, *IEEE International Conference on Cluster Computing* (*CLUSTER2000*), pp.17–26 (2000.11).
19) Tanabe, Yamamoto, Nishi, Kudoh, Hamada, Nakajo and Amano: On-the-fly Sending: A Low Latency High Bandwidth Message Transfer Mechanism, *5th International Symposium on Parallel Architectures, Algorithms, and Networks* (*I-SPAN2000*), pp.186–193 (2000.12).
20) 山本，田邊，西，土屋，渡辺，今城，上嶋，金野，寺川，慶光院，工藤，天野：高速性と柔軟性を併せ持つネットワークインタフェース用チップ：Martini，情報処理学会計算機アーキテクチャ研究会，Vol.2000, No.110, pp.19–24 (2000).
21) 天野，山本，渡邊，土屋，金子，工藤：クラスタコンピュータ用ネットワークインタフェースチップ Martini における代行処理機構，電子情報通信学会技術報告 CPSY2001-54 (2001.10).
22) Keleher, P., Cox, A.L. and Zwaenepoel, W.: Lazy consistency for software distributed shared memory, *Proc. 19th ISCA*, pp.13–21 (1992).
23) Scales, D.J., Gharachorloo, K. and Thekkath, C.A.: Shasta: A Low Overhead, Software-Only Approach for Supporting Fine-Grain Shared Memory, *ASPLOS'96* (1996.10).

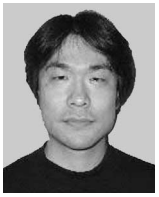
(平成 13 年 9 月 7 日受付)
(平成 14 年 2 月 13 日採録)

田邊　昇（正会員）

1985 年横浜国立大学工学部卒業．1987 年同大学大学院工学研究科修了．同年（株）東芝に入社．1998 年より 2001 年まで新情報処理開発機構つくば研究センターに出向．並列処理，並列アーキテクチャに関する研究に従事．現在，（株）東芝研究開発センター勤務．工学博士．電子情報通信学会会員．

山本　淳二（正会員）

1991 年慶應義塾大学理工学部卒業．1997 年同大学大学院理工学研究科博士課程単位取得退学．同年新情報処理開発機構入社．2002 年より（株）日立製作所・研究開発本部に勤務．並列処理・ネットワークに関する研究に従事．博士（工学）．



濱田　芳博

2001 年東京農工大学工学部卒業．現在，同大学大学院工学研究科（前期課程）在学中．電子情報工学専攻．

中條　拓伯（正会員）

1961 年生まれ．1987 年神戸大学大学院工学研究科修了電子工学専攻．1989 年神戸大学工学部情報知能工学科助手を経て，現在，東京農工大学工学部情報コミュニケーション工学科助教授．1998 年より 1 年間イリノイ大学スーパコンピューティング研究開発センター（CSRD）にて客員助教授．プロセッサアーキテクチャ，分散共有メモリ，クラスタコンピューティングに関する研究に従事．電子情報通信学会，IEEE-CS 各会員．博士（工学）．

工藤　知宏（正会員）

1991 年慶應義塾大学大学院理工学研究科博士課程単位取得退学．東京工科大学講師・助教授を経て，1997 年より新情報処理開発機構並列分散システムアーキテクチャつくば研究室室長．工学博士．並列処理，通信アーキテクチャに関する研究に従事．

天野　英晴（正会員）

1986 年慶應義塾大学大学院理工学研究科修了．工学博士．現在，同大学情報工学科教授．計算機アーキテクチャの研究に従事．