計算機アーキテクチャ 137−12
ハイパフォーマンス
コンピューティング 80−12
（2000. 3. 2）

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

田邊 昇　　山本 淳二　　工藤 知宏

新情報処理開発機構つくば研究センタ

筆者らはネットワークインタフェース(NIC)をメモリスロットに搭載するというアプローチ（MEMOnet アーキテクチャ）に基づき、高性能な PC クラスタ用 NIC のプロトタイプ DIMMnet-1 を開発中である。DIMM スロットを有する安価なパソコン上で、片方向あたり 8Gbps という通信バンド幅と、8 バイトデータ送信時にソフトウェアオーバーヘッド 1 命令サイクル、ハードウェア遅延 45ns という極めて高速なメッセージ生成が可能である。本報告では従来型 NIC の問題点をバンド幅と遅延時間の観点から整理し、DIMMnet-1 においてはどのようにこれらを克復しているかを解説する。次に DIMMnet-1 における細粒度通信アーキテクチャとその通信遅延性能に関するを予測を述べる。

The Finegrain Communication Facilities on DIMMnet-1 Network Interface Inserted in a Memory Slot

Noboru Tanabe　　Junji Yamamoto　　Tomohiro Kudoh

Real World Computing Partnership

The finegrain communication architecture on DIMMnet-1 network interface based on MEMOnet is described. MEMOnet is the network interface (NIC) architecture that the NIC inserted in a memory slot. The DIMMnet-1's link bandwidth is 8Gbps per direction. The overhead to generate a message is only 1 CPU cycle and the hardware delay is only 45ns. The excellent performance is available for cheap personal computers with DIMM slots. We also present the many problems of the old fashioned NICs. These problems become obvious if the 8Gbps optical link is introduced to the NIC with the same old fashioned architecture. Most of all these problems are newly solved on the DIMMnet-1.

1 はじめに

近年、高性能 PC を多数用いて並列処理を行なういわゆるクラスタコンピューティングが注目されている。これらには、Myrinet[1] に代表されるシステムエリアの低遅延ネットワークを用いたものと、100BaseT や Gigabit Ethernet などの汎用の高速ネットワークを用いたもの [2] がある。これらはいずれも PCI バスに接続されるタイプのネットワークインタフェース (NIC) を用いている。

筆者らは RHiNET-1 [3] という 1Gbps の光インタコネクションを用いた、PCI バスに接続されるタイプの NIC を開発し、RHiNET-2 [4] という 8Gbps の光インタコネクション [5] を用いた、64bit66MHzPCI バスに接続されるタイプの NIC を開発中である。

しかし、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、PCIX[6]、InfiniBand[7] といった次世代のサーバー向け入出力の規格が提案されつつある。よってサーバー機に関してはこのボトルネックは改善されつつあるが、最も価格性能比においてメリットのあるエンドユーザー用 PC 向けに、これらの入出力バスが普及するかどうか不透明である。

このような問題意識にたち我々は、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ネットワークインタフェースを検討している。このようなクラスのネットワークインタフェースを MEMOnet[8] 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ネットワークインタフェースを実現可能とすると思われる。

そのようなクラスに属する NIC の研究としては、1990 年代中期、MINI [9] という複数の SIMM ソケットに搭載されるエクステンダを介したボード上に画像用のデュアルポート DRAM を用いた ATM 用 NIC があった。一方、DIMM を拡張した新しいメモリモジュールが近年提案され、規格化が進められた。それが日本の工業規格である EIAJ や国際規格である JEDEC で認可された PEMM(Processor Enhanced Memory Module)[10] である。我々はこの PEMM 規格を基盤に、MEMOnet のプロトタイプとして DIMM スロットに搭載される DIMMnet-1 を開発中である。

光インタコネクションの有効活用から出発した経緯から、筆者ら MEMOnet を提唱した際に、バンド幅に関する評価 [8] を示した。しかし、MEMOnet の利点はバンド幅向上はもとより、通信遅延の低下に貢献すると考えている。MEMOnet により、極めて低遅延かつ高バンド幅のネットワークを実現できれば、Shasta[11] のようなアプリケーションの逐次バイナリコードからプライベートでないロード・ストアに対して一貫性管理コードを挿入するような細粒度で高頻度なメッセージ通信を発生するアプローチを、より効果的にする可能性を秘めている。

本報告ではまず、従来型 NIC の問題点をバンド幅と遅延時間の観点から整理し、DIMMnet-1 のアーキテクチャにおいてはどのようにこれらを克復しているかを解説する。次に DIMMnet-1 の遅延時間に関する改善に焦点をあて、細粒度通信機構の構成と動作を紹介し、通信遅延時間がどの程度になるかという見通しを報告する。

2 従来型 NIC の問題点

本章では、8Gbps の光通信リンクを活用する DIMMnet-1 で解決する課題を明らかにすべく、従来の NIC の問題点を明らかにする。それらは大別して、PCI バス利用に起因するもの (P:PCI)、資源の不足に起因するもの (C:Capacity)、アーキテクチャ上の工夫の不足によるもの (A:Architecture) の 3 つの要因があり、それらが絡み合っているものもある。以下、各節のタイトルに付した記号 (P,C,A) は関連する上記の要因を示す。

2.1 バンド幅の不足

2.1.1 ピークバンド幅不足 (P)

光モジュールを使えば 8Gbps(1GB/s) のバンド幅を実装することは可能である。一方、PCI バスは 133MB/s、

64bit66MHzPCIでも532MB/sにすぎない。安価なコンシューマ用PCには64bit66MHzPCIやPCIX等のサーバー用の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133MB/sという制約を受ける。

2.1.2 主記憶アクセスの競合 (P)

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800MB/s〜1GB/s程度で、そこに光通信による1GB/sを流し込むとそれだけでバンド幅が足らなくなる。

2.1.3 各種PCIカードとの競合 (P)

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。例えば、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35MB/s)を主記憶経由でやろうとすると、PCIのバンド幅を使いきる。

2.1.4 NICメモリへのアクセス集中 (P,C,A)

Myrinetで言えば送信と受信を並行して行わせるならばNIC上の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、データフェッチが全て一箇所のNICメモリ（小容量高速SRAM）に集中するので、通信リンクを8倍のバンド幅に引き上げたら、NICメモリでバンド幅を確保することが困難である。

特に通信リンクが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得ず、このことはNICメモリのバンド幅不足に拍車をかける。さらなるアーキテクチャ上の工夫により送信時のDMAを排除できるが、そこまでやっているNICは見当たらない。

2.1.5 高遅延ゆえの通信時期の集中 (P,C,A)

従来は通信起動遅延時間が短くても数μs程度かかるので、計算しながら即通信するのではなく、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得ない。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。この実質的なバンド幅を低めてしまう。さらに、メモリに溜め込んでから送るので不必要な主記憶バンド幅消費を引き起こすこともある。

2.2 大きな遅延時間

2.2.1 PCIバスの遅い周波数 (P)

PCIバスはFSBや主記憶のバスと比べて通常1/4程度の周波数で動いている。よってPCIバス上で1クロックで済む処理でも、FSBや主記憶のバスにおける1クロックの4倍程度の時間がかかる。

2.2.2 バス調停の遅延 (P)

PCIバスには複数のバスマスタが接続される可能性があるのでバスの調停にかかる遅延がかかる。この遅延は主記憶バスの1/4程度の周波数で行われる事象である。

2.2.3 NICメモリの容量不足 (C)

従来のNIC上のメモリは外付けSRAMによる高速性と引き換えに小容量であったため、主記憶上に通信に必要な情報を一時的または永続的に置かざるを得ないことがある。このため、DMAのセットアップ時間やバス調停時間のオーバーヘッドが通信のたびにかかるようになる。

2.2.4 ディスクリプタの動きの激しさ (P,C)

ディスクリプタを主記憶上に置かざるを得ない構成のNICを用いる場合、NICと主記憶とCPUの間をディスクリプタが行き交い、その度に調停フェーズも入る。

また、SRAMで実装されているためにNICメモリの容量が少ないMyrinetなどの場合、NICメモリ上に配置されるVIを保持する領域の容量には制約が大きいので、ノード数の大きなクラスタにVIA[12]を載せるにはNICメモリ上ではなく主記憶上にVIを配置せざるを得なくなる。その場合も同様の遅延が生じる。

2.2.5 パケット廃棄対応の複雑なプロトコル (A)

パケットがネットワーク中で廃棄されることが前提になっていることが多いために、プロトコルが複雑になり、それらを大方ソフトで行っているため遅延時間がかかる。

2.2.6 低速プロセッサ上での複雑な処理 (A,P)

ホストのCPUへの割り込みやNIC上資源へのCPUからのアクセスに伴うオーバーヘッド（ハードによってはこれがカーネルモードへの移行が強要される）を避けるために、近年のホストCPUと比較して10分の1以下の周波数で動くNIC上のプロセッサで複雑なプロトコル処理を行わせているために、大きな遅延時間がかかる。

2.2.7 イベント通知オーバーヘッド (P,A)

PCにおける割込みオーバーヘッドは極めて大きい。それを避けるためにPCIバス上のNICへのポーリングをする場合があるが、メモリを1ワード読むための時間より、PCIバスを1ワード読むための時間は大きく、バスの調停も引き起こす。つまり、PCIバス上の資源は安易にポーリングを行える場所ではない。

さらにMyrinetの場合はNIC上のレジスタが同一ページにマップされるため、マルチユーザー環境では事実上ユーザーモードでのNICレジスタへのポーリングはできない。よってNICメモリ上に設けたフラグをポーリングするか、DMAにより1ワードのDMAで主記憶上のフラグを更新しなければならない。PCIバス上のNICメモリへのポーリングがバンド幅に悪影響を与えるため、例えばMyrinet上のPM2.0ではDMAで主記憶上のフラグを更新するようになった。これでバンド幅は向上するが、遅延時間はDMAの増加で1μs程度増加させてしまう。

2.2.8 ヘッダー情報再利用性の乏しさ (A)

パケットとなるべきデータの格納アドレスや、宛先関連のパケットのヘッダーの元になる情報を含むディスクリプタは部分的にも再利用されることはなく、別のパケットを生成するたびにNICと主記憶とCPUの間を行き交う。

2.2.9 ディスクリプタの冗長ビット (A)

ディスクリプタは通常ドライバのソフトやNICのDMAが扱いやすいようなデータ構造を持つ場合があり、冗長なビットを多数含んでいると共に、パケットのヘッダーを生成するにもNICが数クロックを費やして毎回変換を行っている。例えば1ワードや1ダブルワードといった粒度の細かい通信を行おうとする場合はディスクリプタの方が本来送るべき情報より多くの遅延を生む。

2.2.10 冗長なDMA転送 (P,C)

高バンド幅のNICは前述の理由で、NIC上のメモリに一度データを受け、改めて主記憶にDMA転送する必要があった。よって、送受信で1回ずつの冗長なDMA転送が存在した。アーキテクチャ上の工夫により送信時のDMAを排除できるが、排除していないのが実状である。

3 MEMOnetとDIMMnet-1

3.1 MEMOnetとは

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Myrinet [1]や

RHiNET [3] [4] 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三つ(SIMM, DIMM, RIMM)が考えられる。SIMM形式のものをSIMMnet、DIMM形式のものをDIMMnet、RIMM (Rambus仕様) 形式のものをRIMMnetと呼んでいる。MINI[9]はSIMMnetの一例といういことになる。

我々はMEMOnetの有効性を実証すべく、DIMMnetのプロトタイプDIMMnet-1を現在開発中である。

3.2 DIMMnet-1の概要

DIMMnet-1は、PC100またはPC133仕様のDIMMスロットにささるネットワークインタフェースである。その基本構造を図1に、主な仕様を表1に示す。

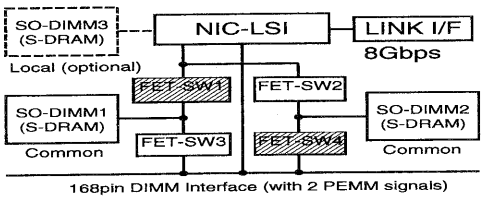

図 1: DIMMnet-1の基本構造

NIC-LSIは低遅延のFETバススイッチにより2バンクのSO-DIMM (ノート型PCで用いられる汎用部品) 間を切り替えつつ、RHiNET-2互換の光リンクインタフェースとデータの送受信をする。DIMMスロットの信号はじかにNIC-LSIに入力されるNIC制御ポートを有する。

SO-DIMMのシリアルプレゼンスディテクトデータを読んで、それに対応する適切なシリアルプレゼンスディテクトデータをDIMMスロットに出力する。さらにオプショナルでローカルなSO-DIMMを1バンク持つ。

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」[10]に準拠させる。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(バンクメモリへのアクセスを待たせる信号と割込み信号)が無くても動作するモードの二つのモードを有する。

表 1: DIMMnet-1の主な仕様

インタフェース	PC133ベースのPEMM
通信リンク	12ch光接続モジュール MDS4212A,MDR4212A
NICメモリ(共有)	PC133 SO-DIMM×2
NICメモリ(ローカル)	PC133 SO-DIMM×1
NIC-LSIのテクノロジ	0.18 μm CMOS
通信リンクバンド幅	各方向8Gbps(全二重)
NICメモリバンド幅	2128MB/s(ローカルメモリ使用時3192MB/s)
送信時NIC遅延時間	45ns
受信時NIC遅延時間	45ns
メモリアクセス遅延増	0クロック

4 DIMMnet-1における改善点

本章では、2章に列挙した問題点に対するDIMMnet-1における改善を2章に列挙した順に述べる。

4.1 バンド幅の不足への対応

4.1.1 メモリスロットで1GB/s

メモリスロットをCPUとのデータ受け渡し口にしており、そのバンド幅は光通信リンクの片方向分(1GB/s)を有する。これはPCIバスの8倍、64bit66.6MHzPCIバスの約2倍の能力を有し、1Gbps程度の通信リンクをPCIバスで支えているMyrinetやRHiNET-1と比較して、8倍の能力になった通信リンクを有するDIMMnet-1はバランスを維持している。

4.1.2 ダブルバッファでバンド幅倍増

DIMMnet-1では主記憶のパスにNICが搭載されるMEMOnetアーキテクチャをとる。NIC上に大容量のSO-DIMMが最低2枚搭載され、ある瞬間ではその片方がCPUの方を向いており、もう片方がNICの方を向いているように低遅延FETバススイッチが制御される。つまりNICメモリがダブルバッファの構成を取るので、見かけ上合計で2GB/sのメモリバンド幅があるように見え、CPUとNICの同時並列アクセスが実現される。

4.1.3 PCIカードとはバンド幅を非共有

DIMMnet-1はPCIバスとは十分に隔離された場所をアクセスするので、他のイーサカード、SCSIカード、ビデオカード、ビデオキャプチャカード等のPCIカードとはNICがバンド幅を共有しない。

4.1.4 内蔵メモリ、バイパス等によるアクセス分散

DIMMnet-1は、RHiNET/SWに用いられたASICまたはその後継シリーズを使用して実装される。このため内蔵メモリも比較的大きなものを実装でき、内蔵CPUのキャッシュや、通信リンクとCPUとのインタフェースなどに使用されるFIFOも内蔵される。それらのバッファが溢れるか、NICメモリへのアクセスは通信のソースまたはディスティネーションがNICメモリではない限り発生しないので、NICメモリへのアクセス要求は軽減される。さらに、On the fly送信においては通信リンクとCPUとのインタフェースにあるFIFOバッファ間がバイパスされ、NICメモリは経由しないのでNICメモリへのアクセス要求は発生しない。

4.1.5 低遅延化による通信時期の分散

DIMMnet-1では後述する様々な対策により、通信遅延時間を大幅に短縮する。従来のメッセージ発生が数マイクロ秒以上かかっていたのに対し、DIMMnet-1では最短で45ns + α (αはチップセット遅延)でメッセージを発生させることができる。その結果、通信粒度を粗くするために通信時期が集中してしまうことを防止・軽減でき、実効的な通信バンド幅は高くなったように見える。

4.2 大きな遅延時間への対応

4.2.1 メモリバスの高い周波数

DIMMnet-1では主記憶のパスにNICが搭載されるMEMOnetアーキテクチャをとるので、安価なPC上でもPCIバスと比べて4倍程度の周波数で動作可能なNICを構築できる。DIMMスロットでは100MHz ~133MHz程度が現在の主流だが、その程度の周波数はFPGAでは実装困難ながらASICによれば実現可能な領域である。

4.2.2 ダブルバッファによる調停の削減

DIMMnet-1では主記憶のパスにFETバススイッチを介して2つのメモリバンクが接続せれており、NICはある瞬間は2つあるメモリバンクのうち少なくとも一つを排他的にアクセス可能な状態にあり、別バンクのNICメモリや、別スロットにある通常のDIMMへのCPUからのアクセスと競合しない。NICのDMAによる送受信が必ずしもCPUとの調停を必要としないことを意味する。さらに他のPCIデバイスとの主記憶の競合に関しては一切ない。よって従来のPCIバスによるNICの場合はあらゆるDMAアクセスがCPUや他のPCIバス上のマスタデバイスとの調停を必要としていたのに対して、バス調停の遅延の総和を縮めることができる。しかもこの遅延は主記憶バスというPCIバスの4倍程度の周波数で行われる事象によるものである。

4.2.3 内蔵メモリと大容量NICメモリ

DIMMnet-1 では比較的潤沢な内蔵メモリを持ったASIC により実装され、基本的には内蔵メモリが溢れたり、デスティネーションが外部メモリである場合に限り外部メモリへのアクセスが発生する。このため、外部メモリへの速度的要求は Myrinet 等の従来型 NIC と比較して小さく、ノート PC 用の安価な汎用部品である SO-DIMM による外部メモリによって NIC メモリを構成することができる。高々4MB 程度しか無い Myrinet 等と比較して、安価に大容量の NIC メモリを持つことができ、NIC メモリの容量不足に起因する望ましくない使い方に伴う DMA やバス調停時間のオーバーヘッドが回避できる。

4.2.4 NIC上に配置されるディスクリプタリスト

DIMMnet-1 では主記憶と同等の遅延時間でアクセスが可能な大容量の NIC メモリを搭載するために、従来の NIC で用いられていた DMA 転送モードにおけるディスクリプタを主記憶上に保持したり、通信前に CPU が作成し NIC に転送する必要は無く、NIC メモリ上に配置することができる。このため CPU はディスクリプタリストへのトップポインタのみを NIC にセットしてやるだけで良く、NIC と主記憶と CPU の間をディスクリプタが行き交い、その度に調停時間や DMA 設定時間、転送遅延時間がかかるといった従来の NIC で発生していた遅延時間を排除できる。またノード数の大きなクラスタに VIA を載せる際に NIC メモリ上に VI を配置できるので、従来の NIC で発生していた遅延時間を排除できる。

4.2.5 選択的信頼性維持ポリシー

DIMMnet-1 ではパケットそのものが消失することは bit 化け等のハード的なエラーが起こった時のみで、ethernet や ATM のように正常な動作をしつつパケットが廃棄されることはない。DIMMnet-1 で用いられる光モジュールの bit error rate は 10^{-20}であり高い信頼性を持っている。ただし台数や設置状況によってはハードの信頼性を十分高いと見れる場合と見れない場合があると考えられる。

さらに DIMMnet-1 の場合はある瞬間では NIC が確実に占有できる容量もバンド幅も大きな領域（NIC メモリの片方のバンク）を持っており、受信の際のメモリアクセスの優先度を高くすることにより受信パケットの欠落を従来の NIC より低く押さえることが可能である。

一方、上位ソフトウェアレイヤによる対応（チェックポインティングなど）を仮定する動向も一部で存在する。例えば VIA1.0 [12] のように信頼性のある通信は必須事項ではない。RWCP で開発された SCore-D においてもコンシステントチェックポインティング [13] が実現されている。

よって DIMMnet-1 ではハードで通信の信頼性を保証するモードと、保証しない代わりに高速なモードを切り替え可能とする。信頼性を保証しないモードでは、ハード的にエラーは検出可能とすることで上位ソフトによる対応を促す。一種の同期手段としても用いられるプログラマから見える ACK 付きの通信と、ACK なしの通信を明示的に指示可能とし、適宜に ACK 付きの通信を追加したり間引くことで信頼性と性能の妥協点を調整可能とする。

信頼性を保証するモードでは NIC がパケットにシーケンス番号を付け、通信ログを NIC メモリ上に取り、NACK が返ってきた場合、NACK を引き起こしたシーケンス番号までさかのぼって再送を行う。受信バッファオーバーフローを起こした場合は NACK が送信元に返され、受信側は同一ノードからのそのシーケンス番号以降のパケットを廃棄する。通信ログを残す際に余計なメモリアクセスが増加するが、ローカルメモリを設置することにより、通信ログ保存に伴うオーバーヘッドを軽減可能とする。

4.2.6 高速プロセッサ上での簡素な処理

DIMMnet-1 では、Myrinet の LANai プロセッサのように PCI バスのクロックまたはその半分の周波数 (33.3MHz) で動作する低速な CPU ではなく、メモリバスのクロック（100～133MHz）で動作する NIC-LSI に内蔵される CPU によって処理される。その処理内容は、パケットそのものが消失することは十分頻度が低くチェックポインティング等の上位ソフトウェアで救うべき事象として扱う比較的簡素なプロトコル処理である。

4.2.7 制御ポート等による軽いポーリング

DIMMnet-1 では、DIMM スロット上に FET バススイッチを介して接続される NIC メモリと、DIMM スロットに直に接続される NIC 制御ポートを有する。このポートは CPU からしかアクセスされないので調停も、バンク切り替えの煩わしさも無く、一部の特権レジスタを除く内部のレジスタや内部メモリをユーザーモードでアンキャッシャブルな主記憶と同様の比較的小さなオーバーヘッドでアクセス可能とする。ポーリングによるアクセスと NIC の送受信が全て PCI バスを通過する従来の NIC と異なり、独立の経路でアクセスがされるのでポーリング NIC の送受信のバンド幅低下といった問題を誘発しない。よって、頻繁にポーリングすることによって割込みオーバーヘッドを排除することが可能である。

NIC メモリ上にあるワードをポーリングするケースでは、その時点で CPU 側から見えているバンクへのポーリングは上記のレジスタへのポーリングと同様にアンキャッシャブルな主記憶へのポーリングコストと同等である。その時点で CPU 側から見えていないバンクへのポーリングは、NIC 制御ポートを介したバンク切り替え要求後、NIC 制御レジスタのポーリングを行ってバンクが切り替わったことを確認するというオーバーヘッドが追加される。これとて数マイクロ秒を要する割込みオーバーヘッドに比べればはるかに低オーバーヘッドである。

4.2.8 ヘッダーバッファとディスクリプタリスト

DIMMnet-1 では、後述するシングルモード On the fly 送信においては、あらかじめヘッダーバッファと NIC メモリ上にパケットヘッダーの種となるデータ（ヘッダーシード）を登録しておき、そこに保存されている同一ページ内へのワードまたはダブルワード送信は、既に登録済みのパケットヘッダー情報が再利用される。

DMA 転送モードにおけるディスクリプタは NIC メモリ上にリスト形式で配置する。このため CPU はディスクリプタリストへのトップポインタのみを NIC にセットすれば良く、その再利用が可能である。例えば、ループの繰返しごとに同じ通信を行うのであれば、ループ内に存在する通信のディスクリプタのリストを NIC メモリ上に保持しておくことにより、一連の通信のディスクリプタは再利用され、通信起動オーバーヘッドは大幅に短縮される。

4.2.9 冗長ビットを排除したヘッダーシード

DIMMnet-1 では、後述するシングルモード On the fly 送信においては、あらかじめヘッダーバッファと NIC メモリ上にパケットヘッダーの種となるデータ（ヘッダーシード）を登録する。これは最初の登録を行う API の関数の中で生成され、プロセスグループ ID とオフセット値を追加するだけでヘッダーになる冗長ビットをそぎ落としたヘッダーに限りなく近い状態のデータであり、実際の通信発生時に未圧縮の状態では NIC に渡されない。

4.2.10 主記憶的NICメモリとOn the fly送信

DIMMnet-1 では、従来の NIC メモリよりも低オーバーヘッドかつ高バンド幅で、桁違いに大容量である性質を生かし、NIC メモリをメッセージの終点とし、ユーザープロセスから直接、NIC 上のメモリに届いたデータをアクセスする。DMA 転送の回数は DMA 送信モードでは送信時に NIC メモリから通信リンクへの DMA1 回と、受信時に通信リンクから NIC メモリへの DMA1 回のみであり、送受信で 1 回ずつの DMA 転送を削減している。

さらに、On the fly 送信においては NIC メモリは経由しないので NIC メモリからの DMA は完全に排除される。

5 DIMMnet-1 の細粒度通信機構

5.1 細粒度通信用 API の概要

DIMMnet-1 では RWCP で開発され MPICH-PM や SCore などのベースになっている PM API に、表 2に一部を列挙した DIMMnet-1 の細粒度通信用の拡張を施した PM-D(PM DIMMnet extention) API を当面提供する予定である。

表 2: PM-D API の概要

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
Block mode On the fly 通信	_pmOpenB	書き込み口生成
	_pmCreateH	ヘッダシードの生成
	_pmKickB	起動アドレスの生成
	_pmCloseB	書き込み口の閉鎖
Single mode On the fly 通信	_pmOpenS	書き込み口生成
	_pmMapS	ヘッダシードのマッピングとヘッダバッファへの登録
	_pmUnmapS	ヘッダシードのアンマッピングとヘッダバッファからの削除
	_pmCloseS	書き込み口の閉鎖

5.2 シングルモード On the fly 通信

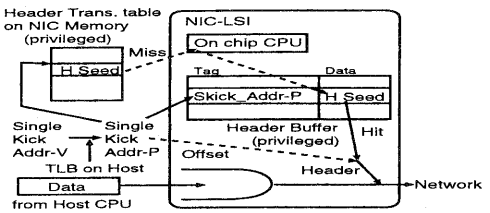

図 2: シングルモード On the fly 通信

シングルモード On the fly 通信におけるパケット生成メカニズムを図 2 に示す。シングルモード On the fly 通信に際しては、あらかじめ _pmOpenS() により起動用の仮想アドレス skick_addr を確保し、_pmMapS() によりヘッダーバッファに skick_addr をタグとして連想されるヘッダーシードをセットしておく。送信すべきデータがレジスタ上に存在すれば、CPU がレジスタ上にあるデータを仮想アドレス skick_addr に書き込むというわずか 1 命令でメッセージ送信を起動できる。

ヘッダーシードは送信すべきパケットのヘッダーから、リモートアドレス部の下位が削除されたものである。これが登録されるヘッダーバッファやヘッダー変換テーブルはユーザーモードからは直接は触れることのできない場所に配置され、_pmMapS() の内部でカーネルモードに移行して、カーネル空間にマップされる NIC レジスタ経由でセットされるので、プロテクションをつかさどるプロセスグループ ID(PGID) をユーザーが勝手に書き換えたりできない。よって、この送信モードに限り、_pmMapS() の内部でカーネルモードからリモートアドレスを物理アドレスで登録でき、受信時のリモートにおけるアドレス変換オーバーヘッドを削除可能である。

なお、このモードで発生するパケットは、単純なリモートライトを行うだけではない。リモートノード remote_node のアドレス remote_addr（pv_flag で仮想アドレスか物理アドレスかを指定）に command を転送し、ステータスを仮想アドレス stat_addr に、結果を仮想アドレス result_addr に返す。command には ACK なしリモートライト、ACK 付きリモートライト、ACK なしリモート間接ライト、ACK 付きリモート間接ライト、リモートリード、返り値なしホットメッセージ、返り値付きホットメッセージがある。ホットメッセージとは、リモートノードの NIC プロセッサでのハンドラーの仮想アドレスを指定した一種のリモートプロシージャコールである。

5.3 ブロックモード On the fly 通信

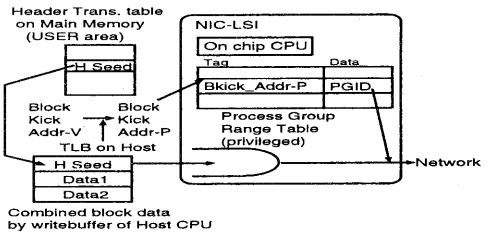

図 3: ブロックモード On the fly 通信

ブロックモード On the fly 通信におけるパケット生成メカニズムを図 3 に示す。シングルモード On the fly 通信があらかじめ NIC 側にヘッダーシードを設定しておくのに対し、ブロックモード On the fly 通信は _pmCreateH() で生成したヘッダーシードをユーザー空間に保持し、これをブロックモード On the fly 送信用の FIFO 入口 bpush_addr（write combining 属性の領域）に書き込み、送りたいデータをそれに後続して書き込み、最後に bkick_addr（uncachable 属性の領域）にダミーを書いて先行する bpush_addr に書き込まれたデータをパケットにしてネットワークに出力する。

こうして CPU のライトバッファによって書き込まれた細かいデータはライトバッファの容量（通常、キャッシュのラインサイズ）までブロック化されてバーストライトが発生し、最後の bpush_addr への書き込みがライトバッファに残ったデータを全て NIC のブロックモード On the fly 送信用の FIFO に吐き出す。

その際、ヘッダーシードがユーザー空間にあるからといって事前に PGID を書き換えても bpush_addr に対応する物理アドレスを頼りに NIC が正しい PGID に付け替えるので他人に成りすますことはできない。

その他、送信データが 8 バイト以上になっても構わないことと、物理アドレスでのリモートアドレス指定ができないことを除けば、概ねシングルモード On the fly 送信のパケットが実行する動作は同じである。

6 性能見積もり

6.1 最短送信時 NIC 遅延時間

シングルモード On the fly 送信の際に、ヘッダーバッファがヒット場合が最短送信遅延時間を達成する。その際の各クロックサイクルでの動作は以下のようになり、NIC が搭載されるメモリスロット上に最初の信号が発生してから 6 クロック（133MHz 動作時に 45ns）で通信リンクインタフェースへの出力が始まる。

1. SDRAM の 8 バイトライト動作第一サイクル目。
2. 上記の第二サイクル目。
3. メモリスロット上のアドレスとデータを入力 FIFO に取り込む。
4. 取り込んだアドレスの上位ビットからヘッダーバッファを検索する。
5. ヒットして得られたヘッダーシードに取り込んだアドレスの下位をはめ込みヘッダーを完成させ、送信 FIFO にヘッダーを突っ込む。
6. 送信 FIFO からヘッダーを読み出し、リンクに送出。

なお、実際の送信に際しては、CPU がレジスタ上にある double のデータを skick_addr に書き込む 1 命令を実

行してからメモリスロット上に信号が現れるまで数クロック（チップセットに依存）の遅延が加算される。

CPUからみればあらかじめヘッダーバッファにヘッダーシードをセットしておけばわずか1命令でメッセージ送信が起動できる点で究極的に高速な送信を実現している。

6.2 最短受信時NIC遅延時間

物理アドレス指定の8バイトのリモートライトの際に、NICの方を向いているバンクに書き込む場合に、最短送信遅延時間を達成する。その際の各クロックサイクルでの動作は以下のようになり、通信リンクインタフェース上に最初の信号が発生してから6クロック（133MHz動作時に45ns）でNICメモリへの出力が始まる。

1. 通信リンクから受信FIFOにヘッダー1を取り込む。
2. 受信FIFOからヘッダー1を取り出す。
3. 受信FIFOからヘッダー2を取り出し、CRCを計算。ヘッダー中のコマンドを検査し、物理アドレス指定8バイトリモートライトであることを知る。
4. 受信FIFOからデータを取り出し、CRCを計算。
5. 受信FIFOからCRCを取り出し、チェック後、受信データとアドレスをライトFIFOに突っ込む。
6. アドレスから書き込みバンクをチェックし、アクティブであることを知り、NICメモリへの書き始める。

なお、仮想アドレス指定の場合は第六クロックサイクルでNIC内TLBによる物理アドレスへの変換が入るために最短でもう1クロック時間がかかる。アクティブではないバンクに受信する場合は、バンクがCPUからロックされていない場合は、もう1クロック時間がかかる。

6.3 最短リモートライト遅延時間

チップセット遅延時間と最短送信遅延時間と最短受信遅延時間の合計が2ノード間直結時の最短リモートライト遅延時間となる。チップセット遅延時間を多目に見積もって10クロック(75ns)とすると、合計で165nsとなる。

表3に各NICでのプロセス間単方向通信遅延を示す。

表3: 各NICでのプロセス間単方向通信遅延

DIMMnet-1	**165ns**	1倍	本論文
Memory channel 2	2.2μs	**13**倍	文献[14]
PM on Myrinet	7.5μs	**45**倍	文献[15]
GigaE PM II	44.6μs	**270**倍	文献[2]

7 おわりに

従来型NICの数多くの問題点をバンド幅と遅延時間の観点から列挙し、現在開発中のDIMMnet-1がいかにしてこれらの問題を克復しているかを示した。I/Oバスではなくメモリスロットに NICを配置するという斬新なアプローチに起因する様々なメリットを享受できるようDIMMnet-1では様々な工夫をしている。

今回示した細粒度通信機構によれば、CPUの命令セットアーキテクチャから根本的に変えない限り実装テクノロジやソフトが進歩してもなかなか短縮できなかった遅延時間を、Alphaベースのシステムで用いられているMemory channel 2の**13**倍、高性能なクラスタ用NICとして各所で活用されているMyrinet上のPMの**45**倍、汎用Gigabit Ethernetで構成するGigaE PMの**270**倍という圧倒的な遅延時間短縮の見通しを得た点で画期的である。

DIMMnet-1は新たなハード性能のバランス点を与え、従来のバランス点の上に成り立っていた並列ソフト技術の再構築を促すような技術であると思われる。DIMMnet-1の設計思想に基づくNIC製品が流布するようになれば、例えば逐次バイナリコードをそのまま並列化してしまうShastaのような魅力的なソフトウェア技術を、性能面で真に使える技術に押し上げ、普及させていく原動力となるのかもしれない。そのような評価は今後の課題である。

今後は、今回触れなかったホットメッセージ等を実行する主体であるNIC内CPU等の一部ハードウェアを共有する形で同一LSI内に実装されるRHiNET-2との両立を模索しつつNIC内CPU回りの設計を進め、2001年春の完成を目指し、DIMMnet-1の開発を進める予定である。

謝辞

バンクメモリへのコンパイラ対応の経験を御教授いただいた早稲田大の笠原博徳教授、MEMnetという名称についてコメントをいただいた東大の松本尚助手、従来型NICの問題点を御議論いただいた新情報処理開発機構の手塚宏史氏、住元真司氏に感謝致します。

参考文献

[1] Myricom corp. http://www.myri.com/

[2] 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", 情報処理学会研究報告 99-ARC-134 (SWoPP'99), pp.61-66 (1999.8)

[3] 山本, 建部, 横山, 土屋, 宮脇, 清水, 天野, 工藤 "高性能並列計算用ネットワーク RHiNET-1の実装と評価", 情報処理学会研究報告 HOKKE2000 (2000.3)

[4] Tomohiro Kudoh, Shinji Nishimura, Junji Yamamoto, Hiroaki Nishi, Osamu Tatebe and Hideharu Amano "RHiNET: A network for high performance parallel processing using locally distributed computers", IWIA'99 (1999.11)

[5] S. Nishimura, T. Kudoh, H. Nishi, N. Matsudaira, K. Harasawa, S. Akutsu, K. Tasho, H. Amano "A network switch using optical interconnection for high performance parallel computing using PCs", Parallel Interconnect'99 (1999.10)

[6] "Computer Makers Propose New PCI Design", http://www.techweb.com/wire/story/TWB19980904S0008

[7] InfiniBand Trade Association, http://www.sysio.org/

[8] 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" 情報処理学会研究報告 99-ARC-134 (SWoPP'99), pp.73-78 (1999.8)

[9] Ron Minnich, Dan Burns and Frank Hady "The Memory Integrated Network Interface" IEEE Micro, Vol. 15, No. 1, (1995.2)

[10] 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)

[11] D. J. Scales, K. Gharachorloo, and C. A. Thekkath "Shasta: A Low Overhead, Software-Only Approach for Supporting Fine-Grain Shared Memory", ASPLOS'96 (1996.10)

[12] Intel corp. "Intel VI Architecture Developer's Guide V1.0", ftp://download.intel.com/design/servers/vi/intel.pdf

[13] 西岡, 堀, 手塚, 石川 "クラスタにおけるコンシステントチェックポイントの実現", 並列処理シンポジウム JSPP'99, pp.207-214 (1999.6)

[14] M. Fillo and R. B. Gillett "Architecture and Implementation of MEMORY CHANNEL 2 ", Digital Technical Journal, Vol.9(1), (1997)

[15] H. Tezuka, A. Hori, Y. Ishikawa, and M. Sato "PM: An Operating System Coordinated High Performance Communication Library", High Performance Computing and Networking'97 (1997)