計算機アーキテクチャ 134−13
（1999. 8. 3）

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

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

新しいネットワークインタフェース (NIC) のクラスとして MEMnet を提案する。MEMnet は 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 NIC である。MEMnet 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性能を持つ NIC を実現可能とする。そ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。本報告では DIMMnet-1 のアーキテクチャの概要を述べ、NIC 上メモリを write combining 属性に設定することを提案する。次に DIMMnet-1 の性能を左右する NIC 上メモリへのアクセス速度の評価を報告する。低オーバーヘッドな On the fly 送信モードで重要になる書き込みについては旧来の P6 アーキテクチャ命令のみで PCI バスの限界を越えるバンド幅 (189MB/s) を実現できた。

MEMnet : Network interface attached on memory slot

Noboru Tanabe† Junji Yamamoto† Tomohiro Kudoh†

†Real World Computing Partnership

We defined a new network interface class called MEMnet which is network interface attached on memory slot. MEMnet provides large-bandwidth low-latency communication which is impossible to realize on PCI bus for the cheap PCs with high performance. We are developing a prototype of MEMnet called DIMMnet-1 which is attached on DIMM slot. In this report, we present the overview of DIMMnet-1 and propose using write combining attribute which has no cachability for the memory on MEMnet. We also present its memory access performance. We could observe high writing bandwidth (189 MB/s) which is impossible to realize on PCI bus from MPU to the memory area with write combining attribute accessed by the conventional P6 instruction set.

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レイテンシネットワークを用いたもの [2] 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 [3][4] 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を用いている。

筆者の所属する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においても RHiNET [5] [6] という、1Gbps〜8Gbps 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ネットワークインタフェースを開発している。

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レイテンシともに力不足である。一方、PCIX[7] や NGIO[8] 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の量産 PC 向けに、これらの入出力バスが厳しい価格制約を満たした上で普及するかどうか不透明である。全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、PCIX 等とは別のアプローチも検討に値する。

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 **MEMnet** と名付ける。**MEMnet** 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(Memory Integrated Network Interface)[14] という複数の SIMM ソケットに刺さ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。しかし SIMM は二枚一組で動作するために、どうしても複雑な形状とならざるを得ず、同定度の性能レベルにあったデファクトスタンダードの地位を確立した PCI バスの普及に伴い淘汰されてしまったと考えられる。

これに対し DIMM は SIMM と異なり一枚で動作できるので **MEMnet** として用いる際の実装形態が簡素になる。一方、この DIMM のメリットに着目した 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)[9] [10] [11] [12] [13] である。

我々はこの PEMM 規格を基盤に、**MEMnet** のプロトタイプとして AT 互換パソコンの DIMM スロットに搭載される **DIMMnet-1** を開発中である。**DIMMnet-1** は、本来 DSP の PC への効率的利用を意図して設計されて来たこの規格をネットワークインタフェースに応用するものである。

本報告では **DIMMnet-1** のアーキテクチャの概要を述べ、その性能を左右するメモリスロット上に置かれたネットワークインタフェース上メモリへのアクセス速度の評価を報告する。

2 MEMnet と PEMM

MEMnet は従来の Myrinet [1] や **RHiNET** [5] [6] 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 (Network Interface Card)**」と定義する。

MEMnet 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** と呼ぶことにする。MINI[14] は SIMMnet の一例といういことになる。

我々は淘汰された MINI の教訓を生かしつつ **DIMMnet** のプロトタイプ **DIMMnet-1** を開発する。その際、DIMM の国際規格の JEDEC 規格に準拠させるとともに、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 EIAJ ED-5514 の「プロセッサ搭載メモリ・モジュール (PEMM) 動作仕様標準」[9] に準拠させる。

なお PEMM はアイ・オー・データ機器社が WORLD PC EXPO98 において参考出品した DSP ボード [10] で実演された技術である。その技術の出どころは Texas Instsruments 社つくば研究開発センターであり、国際規格の JEDEC にも既に認定されている。PEMM に関する情報は LoomShuttle 社 [12] の web にまとめられており、TI 社製の PEMM のマニュアル [13] もそこからダウンロード可能である。今後、対応するマザーボードが出て来る可能性が高まって来ており、日経 BP の取材によれば台湾のチップセットメーカーが興味を示していることが報告 [11] されている。

3 DIMMnet-1 の概要

3.1 概略構造

図 1に **DIMMnet-1** の概略ブロック図を示す。**DIMMnet-1** は通常より背の高い DIMM 基板にノートパソコン等で用いられる小型の DIMM(各 64-128MB 程度の SO-DIMM) が二枚搭載され、これらと全体を制御する NIC-LSI と DIMM インタフェースとの間を FET スイッチを介して切替える。図 1の 4 箇所のスイッチのうち斜線を施した FET-SW1,4 が同時に ON/OFF する。FET-SW2,3 は基本的にはこれらと ON/OFF が反転した動作をする。

構造面で言えば従来の PCI バスに接続される NIC とは、MPU と NIC の両側からバンクがかち合わない限り同時に高速にアクセスされる大容量のメモリを持っているところが異なる。従来にないこの特質をうまく NIC の高性能化に活かせるかどうかが鍵となる。

なお、当面は PEMM 規格準拠のマザーボードが入手できないことが予想され、割り込み信号を DIMM スロットからチップセットには供給できない。NIC からの割り込みがないと実現できない機能については、別の PCI カードからチップセット側に割り込み信号を供給可能とする。

また、1〜8Gbps の性能を実現する光リンクモジュールのサイズや重量が DIMM スロットと整合しない場合は、LVDS (Low Voltage Differential Signaling) チャネルリンクシリアライザ／デシリアライザ [16] 等を用いて、上記割り込み信号用 PCI カード上に NIC〜光モジュール間の信号を接続する予定である。

3.2 グローバルメモリモデル

ユーザーから見えるメモリは、以下の領域から構成される。

1. 通常のメモリ (ノード内の自プロセスが読み書き可能かつキャッシングとページングがなされる領域)
2. NIC 上メモリ (ユーザー空間にマップされた NIC 上のキャッシングとページングがされない領域で、P6 系

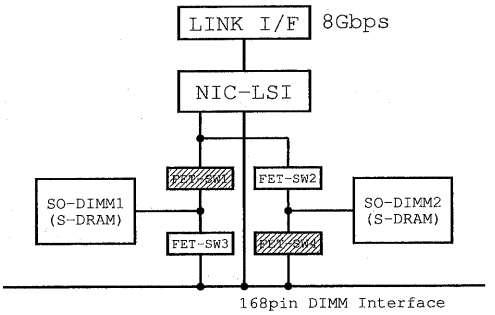

図 1: DIMMnet-1 の概略ブロック図

MPU を用いたシステムにおいては write combining 属性の領域に設定されることを推奨する)

3. リモートメモリ (リモートノード上の同一並列プロセスグループ ID を持つ通信領域であり、上記 2 の領域が用いられ、ユーザーからは送信先 RANK(論理 PE 番号) と通信領域 ID と領域内オフセットを指定してアクセスされる。なお、通信領域 ID は 1 個の RANK が複数持つことを可能とする)
4. NIC 通信コマンド (ユーザー空間にマップされた通信起動アドレスで uncachable 属性の領域に設定される)
5. NIC レジスタ (ユーザー空間にマップされた一部の NIC 制御用レジスタ)

DIMMnet-1 は MPU やチップセットから見たときはハード的にはあくまでもメモリとして存在しているので、上記 1 の領域に対してはバスマスタになって DMA を行ったりはしない。よって、NIC によるデータの送受信は全て NIC 上メモリに対して行われる。キャッシュによる再利用効果を生かしたい場合などのように、そのデータを 1 の領域に移動する必要があるときは、ユーザープログラムがメモリ間コピーを行うものとする。

3.3 プロテクションとアドレス変換

プロテクションは並列プロセスグループ ID(PGID) によって実現される。プロセスはローカルリソースマネージャー (LRM) にグローバルメモリを要求して必要な容量のピンダウンされた領域を確保し、通信領域 ID(WIN-ID) を発行してもらう。各プロセスがグローバルメモリへのアクセスに参加する際は、グローバルリソースマネージャー (GRM) に参加を問い合わせる。GRM は PGID の発行を行い、ノード ID とプロセス ID を PGID と論理 PE 番号 (RANK) に対応付ける。GRM は、全ての PGID に対する RANK とノード ID とプロセス ID の対応表を管理し、各プロセスからの問い合わせ (同一の PGID を持つプロセス ID とそのノード ID と RANK の対応表を要求する等) に応答する。

各ノードでは LRM により自ノード内に存在する全ての PGID と、それに対応するノード内の RANK と、それに対応する WIN-ID から、それに対応する領域の先頭の物理アドレスを引くことができる表 (GRWT) を設定する。対応表 1 は NIC 上メモリに格納される。次に NIC の中のテーブルに送信を起動するための物理アドレスの範囲から PGID を引くことができる表 (PGRT) を設定する。引続き、その物理アドレスをユーザ空間にマップし、MPU が取り扱うプロセスページテーブルに登録される。以上は全てカーネルモードで実行される。

こうして NIC は、MPU の TLB 経由で所定の物理アドレスが発生されたことをもって、PGRT を引いてそのア

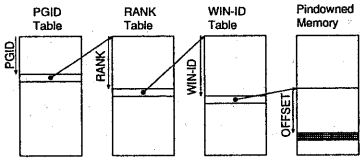

図 2: GRWT による受信時アドレス変換

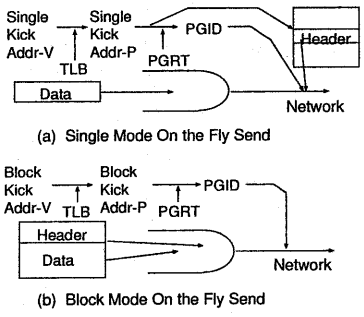

図 3: On the Fly 送信

クセスを行った PGID を決定し、それをリモートアクセスを行うパケットヘッダーにハード的に取り付ける。リモートアクセスを行うパケットを受信した際には図 2に示すようにその PGID と RANK と WIN-ID によって、NIC は GRWT を引き、オフセットを加えて物理アドレスを得て、アクセスを実行する。

3.4 on-the-fly リモートアクセス機構

on-the-fly リモートアクセス機構は NIC 上の空間に書き込まれるデータやアドレスをキャプチャーしてパケットを生成し細粒度通信を行う機構である。

on-the-fly リモートアクセスには図 3に示すように word (32bit) および double word (64bit) のデータにヘッダー情報を付加しつつ送信するシングルモード送信と、double word 以上のブロックデータをパケットに変換して送信するブロックモード送信の二つの送信モードを提供する。

シングルモード送信を行うためには、事前にその領域のアドレスへのアクセスが発生したらいかなるリモートアクセスパケットを発生させるかを定義しておく必要がある。その際、必要であればパケットヘッダーの元となるデータのための配列を NIC 上のメモリ上に物理アドレス上で連続した領域を確保し、値を設定しておく。この際、論理 PE 番号 (MPI でいうところの RANK) はネットワークがハンドリング可能なノードアドレスに変換されて設定される。

ブロックモード送信では、1double word 以上のデータブロックの送信に対応する。送信を指示する所定のアドレスをアクセスするまでパケットヘッダーを含む送信データとして後続する書き込みアクセスを NIC がキャプチャし、並列プロセスグループ ID を NIC が付加してパケットに変換してネットワークに送信する。

3.5 リモート間接書き込み機構

3.5.1 リモート間接書き込みとは

リモート間接書き込み機構とは新情報処理開発機構超並列東芝研究室で筆者らが設計を進めていた超並列計算機 TS/1 [17] [18] [19] で提供していたメッセージ交換支援機構の一つである。

リモート間接書き込みは、送信側からは受信コマンドのアドレスと書き込みデータのペアを送り、受信側は受信側のメモリの指定アドレスに記述された受信コマンドに従って受け取ったデータを書き込む仕組みである。受信コマンドはチェイン可能なので受信側でのかなり複雑なアンパックが可能である。

そのバリエーションとして、後述するリモート FIFO 書き込み機構やホットメッセージ機構が考えられ、これらとハードウェアを共通利用できるものと思われるので、**DIMMnet** においてもこのリモート間接書き込みを採用する。

3.5.2 リモート間接書き込みによるメッセージ交換

送信側が受信側のどこに書き込んで良いか判らない場合はメッセージ交換を使わざるを得ない。もちろん MPI などで記述する場合は常にメッセージ交換を使うことなる。受信時は通常、receive を行う関数の中で対応する key を持つメッセージをシステムバッファから検索し、receive を行う関数の中で指定されるユーザー領域にコピーされる。

MEMnet の場合は受信バッファを NIC 上メモリに大量に置くことができるので、受信側では key 毎（または key の一部分を共通とするメッセージ集合毎）にバッファを用意しておくことは容易である。よって key によるメッセージ検索の速度向上が期待できる。

リモート間接書き込みでメッセージ交換を実現する場合、受信コマンドは receive を行う関数の中でセットする。(上書きする場合は受信コマンドのイネーブルフラグをセットするだけである）メッセージ到着時に受信コマンドが設定されていない場合は、CPU 側に割込みを発生させることで対処できる。

これに対し、受信前に受信時に行う処理を記述した受信コマンドが設定されていると、この検索やコピーが不要になる。こうなるようにプログラマにはなるべく早いタイミングで receive を行う関数を実行して受信コマンドを設定することを推奨する。

3.6 リモート FIFO 書き込み機構

メッセージ交換の実装 [20] や、VIA [21] の実装においてリモートの FIFO への書き込みのサポートは便利であると考えられる。特に **MEMnet** においては NIC が大量のメモリに高速にアクセスできるので、通信相手ごとに受信バッファを持つ VIA を実装する際には大規模なクラスタを効率よく実現できると思われる。

リモート FIFO 書き込みは、リモート間接書き込みにおける、受信コマンドの受信アドレス自動設定等によって実現可能である。すなわち FIFO の初期化の際に、FIFO 用受信コマンドとしてバッファ先頭アドレス、サイズ、トップポインタアドレス、ボトムポインタアドレスを設定し、受信はボトムポインタアドレスから行い、アドレスのラップアラウンドと、ボトムポインタのトップポインタへの到達（FIFO オーバーフロー）の検出および割込み発生を FIFO 用受信コマンドを解釈する NIC 内ハードウェアが行う。

受信コマンドは NIC 上のメモリに置かれるので、**MEMnet** の場合はそれが大量にあるのでメモリ容量が許す範囲で、非常に多数のリモートから書き込み可能な FIFO を設置できる。

3.7 ホットメッセージ機構

3.7.1 ホットメッセージ機構とは

台数が比較的多い分散システムにおいてはバリア同期の高速化が重要である。[22] ホットメッセージ機構とはバリア同期の高速化を主目的としたもので、リモート間接書き込みの一バリエーションとして、あるリモート領域にあ

るコマンドを指定したキックフラグの立ったリモート間接書き込みパケットを受信すると、これをトリガーとして、コマンドに記述された処理を行う機構である。ホットメッセージが起動する処理を実行する主体はホスト MPU ではなく NIC である。

3.7.2 ホットメッセージ機構によるマルチキャスト

コマンドリストに複数のメッセージ送信コマンドのみがリンクされていると、マルチキャストが発生する。これは以下のバリア同期の通知過程においても用いられる。ホットメッセージによるマルチキャストのメリットとしては、コマンドの先頭をキックするところまででメッセージの使命は終わっているのでアクノリッジが早く出せること、長大な送信先リストをメッセージの中に載せて運ばなくても良いことである。ただし、そのアクノリッジは全ての送信先で受信が完了したことを意味するものではなく、保管されている再送に備えた送信中メッセージを破棄して良いという意味にすぎない。

3.7.3 ホットメッセージ機構によるバリア同期

コマンドリストにデクリメント&テストと1個のメッセージ送信コマンドと初期値復元コマンドがリンクされていると、デクリメントされた結果ゼロになると1個のメッセージが発生した後初期値を復元するので、トリー状に構成された計数バリア同期の収集過程において用いられる。コマンドリストに複数のメッセージ送信コマンドのみがリンクされていると、マルチキャストが発生する。これはバリア同期の通知過程等において用いられる。

マルチキャストされるデータの書き込み先をリモートのメモリ上に設け、バリア同期ポイントに突入した後でできる処理を行った後にメモリ上のフラグをポーリングすることで、ファジーバリアを実装することができる。さらにコマンド列や同期フラグのインスタンスを複製しておけば、マルチカラーの同期も実装可能である。

4 MEMnet の問題点と解決法

現状の PC 用チップセットや MPU では、**MEMnet** や PEMM のようなハードウェアの存在は想定されていないため必ずしも十分な性能が得られるとは限らない。現状の汎用 PC チップセット (440BX 等) および Intel P6 系 MPU(Pentium Pro, Celeron, Pentium II, Pentium III) を用いる上で **MEMnet** が抱えていると思われる問題点には以下のようなものが上げられる。

1. Intel P6 系 MPU は命令からはキャッシュ全体のフラッシュ(一斉無効化) をすることはできても、ライン毎やページ属性毎にはキャッシュのインバリデート (無効化) ができない。さらに、既存の PC 用チップセットはメモリスロット側からは PCI バスのように DMA データに対応するキャッシュラインをインバリデートできない。このため **MEMnet** 上のメモリに受信された受信データをキャッシュ領域に配置することができない。しかし、uncachable 領域に受信データが配置されることになると、MPU から **MEMnet** 上のメモリに受信された受信データを利用する際にはキャッシュを用いた際に発生するライン単位 (32 バイト) のバーストアクセストランザクションが利用できない。ゆえに、工夫をしないと悲劇的な性能劣化を引き起こすことが予想される。
2. メモリスロット側からは PEMM 規格対応のチップセットでない限り割り込みがかけられない。

特に上記の第一の問題点は **MEMnet** の存在価値を揺るがす程に根本的にして深刻なものであり、十分な対策と評価が必要である。上記の第一の問題点を解決するために考えられる方策としては以下のようなものが上げられる。

1. **Intel P6 系 MPU 全般に通用する方策：Write combining の活用**
   Write combining とは、Intel 製 MPU では Pentium Pro において初めて実装された機能であり、元々はビデオフレームバッファの効率的な実装を目的に導入されたものである。Write combining は MPU からキャッシュされない領域に対する書き込みの際に、複数の細切れな書き込みを書き込みバッファにおいて遅延させてまとめ上げ、メモリに対してバーストアクセス要求を発生させるアーキテクチャである。

   Intel Architecture Software Developer's Manual volume.3[23] Chapter.9 によると、P6 系 MPU では MTRR(Memory Type Range Register) に 4 種キャッシュ取扱属性 (WB:write back , WC:write combining , WT:write through , UC:uncachable) を設定することができる。この MTRR によって write combining 属性に設定された領域は MPU からはキャッシュされず、かつ、32 バイトの書き込みバッファによって複数の細切れな書き込みをまとめ上げ、メモリに対してバーストアクセス要求を発生させ、かつ、読み出しについては speculative な読み出しが有効になっている。

   この Write combining の機能は非キャッシュ領域に対する読み書きに対する性能向上に大いに貢献すると思われ、Pentium Pro, Celeron, Pentium II においては特にこの機能を **MEMnet** 上のメモリ領域に対して使うことを提案する。

   ただし非キャッシュ領域に対する読み書きに対して、この対策によりどの程度の性能が得られるかは実際に評価する必要があり、本報告ではこの点について実験を行っている。詳細は後述する。
2. **Pentium III 以降にのみ通用する方策：Streaming SIMD 命令の活用**
   Intel Architecture Software Developer's Manual volume.2[24] によると Pentium III においては 128bit 幅のレジスタセットが新たに導入され、P6 系 MPU の命令セットに Streaming SIMD 命令が追加された。その中には 32 バイト以上のデータブロックを事前に事前にロードすることを可能にする PREFETCH 命令、Write combining buffer の意図的な書き込み遅延指示をする SFENCE 命令、4 ワード (128bit) の転送に対応した MOVE 命令など、Pentium III 以前の P6 系 MPU には得られない効率的なバーストデータ転送を可能にすると思われる命令が存在する。ただし今回の実験に用いた C コンパイラ (egcs) がまだ Streaming SIMD 命令に対応していないため、これらの効果を見極める性能評価の実施は今後の課題である。

5 予備性能評価

本章では、4 章で示した第一の問題点がどの程度であり、その解決策として Write combining 機能の適用がどの程度有効なのかを予測するために、主記憶上の領域や PCI バス上のメモリに対し、各種キャッシュ取扱属性を設定し、リードおよびライトのバンド幅を測定する。

本章における実験の測定環境を表 1に示す。

以上の実験環境において主記憶上の領域および PCI バス上の領域に対し、各種キャッシュ取扱属性 (WB:write back , WC:write combining , WT:write through , UC:uncachable) を設定してリードおよびライトのバンド幅をメモリ～MPU 間データ転送バンド幅を Celeron333MHz のシステムの場合と Pentium III のシステムの場合について測定した結果を表 2, 表 3に示す。

測定に用いたプログラムは C 言語で記述したドライバモジュールと計測プログラムからなる。ドライバモジュー

表 1: 測定環境

MPU	Celeron	Pentium III
MPU 内部クロック	333MHz	500MHz
FSB クロック	66.6MHz	100MHz
チップセット	440BX AGP	
主記憶	DIMM 128MB (PC100 SDRAM CL=2)	
PCI バス上メモリ	Myrinet M2M-PCI32C (SRAM 1MB)	
OS	Red hat 6.0 (カーネルは Linux 2.2.10 CONFIG_MTRR を設定)	
C コンパイラ	egcs-2.91.66	

ルを OS 起動後に insmod し、計測プログラムの中でユーザー空間上に malloc() で大量に確保したメモリ領域のうち、計測プログラム自体からは大きくアドレスが外れる場所に読み書き用の 1 ページ (4KB) のデータ領域を設定する。その領域を shell から/proc/mtrr 経由で P6 系 MPU の MTRR(Memory Type Range Resister) の値を書き換えて、所定の領域の属性を変更する。その後、計測プログラムの中で word(32bit) 単位で 1 万回読み書きし、バンド幅を測定した。PCI バス上のメモリとしては PCI バス上の高性能 NIC の代表である Myrinet 上の SRAM を MPU からほぼ同様に読み書きしバンド幅を測定した。

表 2: メモリ～MPU 間データ転送バンド幅 (Celeron 333MHz)

場所	属性	write	Read
DIMM	WB	130MB/s	354MB/s
DIMM	WT	31MB/s	354MB/s
DIMM	WC	130MB/s	26MB/s
DIMM	UC	44MB/s	20MB/s
PCI	WC	101MB/s	5MB/s
PCI	UC	33MB/s	5MB/s

表 3: メモリ～MPU 間データ転送バンド幅 (Pentium III 500MHz)

場所	属性	write	Read
DIMM	WB	187MB/s	591MB/s
DIMM	WT	47MB/s	594MB/s
DIMM	WC	189MB/s	39MB/s
DIMM	UC	66MB/s	30MB/s
PCI	WC	88MB/s	5MB/s
PCI	UC	50MB/s	5MB/s

以上の結果から、既存の P6 系 MPU (Celeron, Pentium III) と既存の PC 用チップセット (440BX) でも MPU からの書き込みに関しては、write combining 属性に設定された非キャッシュ領域が write back 属性に設定されたキャッシュ領域と同等の転送バンド幅が得られることが判る。これは、P6 系 MPU に内蔵される write combining バッファが write back 属性に設定されたキャッシュ領域のライン書き戻し時のバーストアクセストランザクションと同等のトランザクションをメモリに対して発生させることができていることを裏付けている。

一方、DIMM の write combining 属性に設定された非キャッシュ領域へのリードは uncachable 属性に設定された非キャッシュ領域へのリードより 25～30%程度しか高速化しておらず、書き込み時の 130MB/s(Celeron),189MB/s(Pentium III) と比較すると 1/5 程度の性能しか出ていない。つまり write combining 属性に設定された非キャッシュ領域に対する読み出しに対し、ほとんど効果的なプリフェッチを行っていないということが判った。Intel Architecture Software Developer's Manual[23] によると、write combining 属性に設定された領域に対する読み出しは speculative な読み出しを許していることになっているが、本測定環境で本テストプログラムが発生する命令列においては、時々二つの 32bit アクセスを一つの 64bit アクセスにまとめ上げる程度のことしかできていないと予想される。

MPU により一度だけしか用いらない受信データを扱う際には、PCI バスからの DMA により受信した write back 属性に設定されたキャッシュ領域上のデータは全て無効化されるので、ミスヒットしたラインのリフィル時のバンド幅で MPU 側に読めることになる。この値はおそらく write combining 属性に設定された非キャッシュ領域の読み出しバンド幅よりも比較的高いと思われる。なぜなら、キャッシュのリフィルはメモリ側にラインを write back した後にラインをメモリから読み出すことになり、write back 属性の書き込みバンド幅の半分弱のバンド幅が期待できるからである。つまり、再利用のないデータに付いての MPU 側からの利用時の実効バンド幅は少なくとも今回測定に用いた Celelon や Pentium III では PCI バスからの DMA により受信する場合の半分程度に性能低下してしまうと考えられる。

しかし、write combining 属性に設定された領域に対する読み出しが 32 バイトのバーストアクセストランザクションを発生可能なように MPU を改善できれば、おそらく 3 倍程度の性能向上を見込むことができ、PCI バスからの DMA により受信する場合に対し優位に立つことが可能だと思われる。たとえ MPU が対応しなくても、筆者の提案した先読みバッファ[25] や Cray T3E に採用されている Stream buffer[26] をチップセット内に入れるなどすればチップセットで解決可能とも思われる。

一方、PCI バス上のメモリを読み書きする場合は、write combining 属性ならば 101MB/s(Celeron), 88MB/s (Pentium III の場合 FSB が 100MHz の方が 66.6MHz よりも遅くなるが、これは 440BX チップセット内部での同期化の問題ではないかと思われる) と書き込み性能は比較的良好である。しかし、読み出し性能が 5MB/s と圧倒的に遅い。つまり細粒度な送信に有利な状況を提供する DMA を用いないプログラムによる転送時も、書き込み先を write combining 属性に設定しておけば良好であるが、受信については DMA でキャッシュ領域上に書き込みを行わず、プログラムによる転送時を行うと、許容しがたいバンド幅しか得られないことがわかる。

6 おわりに

本報告では **MEMnet** を提案し、その DIMM 上での一実現である DIMMnet-1 の概要について述べた。さらに、**MEMnet** を既存の P6 系 MPU と既存のチップセットで用いる際に問題になる非キャッシュ領域に対する MPU のアクセス性能の性能測定を行った。

write combining 属性の非キャッシュ領域を用いることにより、現状でも送信動作時には PCI バスの限界 (133MB/s) を越える十分なバンド幅 (Pentium III 500MHz において 189MB/s) が得られることがわかった。**MEMnet** は性能向上のテンポが緩い入出力バスの呪縛を断ち切ることにより、性能向上の著しい Direct Rambus 等の高バンド幅メモリや更なる MPU の高速化の恩恵を直接享受できる。

しかし、今回の予備評価から Pentium II までの P6 系 MPU のアーキテクチャに追加する新たな対応がないかぎり、**MEMnet** 上のメモリに受信されたデータを MPU やキャッシュ効果のある通常の主記憶上に移動する際には、大きなバンド幅の低下が発生してしまうという問題点が明らかになった。ただし今回の測定では利用した C コンパイラの限界から Pentium III において追加された

streaming SIMD 命令を一切使っていない。具体的には128bit 幅のレジスタを用いたメモリアクセスや、キャッシュを汚さないデータプリフェッチの命令を駆使することにより改善できる可能性を一切試していない。

次世代のコンシューマ用MPUにおいては**MEMnet**を有効足らしめるwrite combining属性の領域に対する読み出し時の積極的なバーストアクセスを伴うプリフェッチ機能を採り入れることを強く念願したい。既にPentium IIIに非再利用型のデータに対するキャッシュを汚さないプリフェッチの機能が組み込まれている。これを活用することで今回明らかになった読み出し時のバンド幅の低下を改善できる可能性が残されており、それを明らかにするのは今後の課題となる。今回の予備性能評価から、おそらく**MEMnet**が本当に意味を持ち始めるのはstreaming SIMD 命令が多くの次世代のコンシューマ用PCで利用可能となる数年後のことであり、少なくとも現在ではないことが判ったということになろう。

なお、我々は8Gbpsクラスの光インタコネクションを用いたRHiNET-2も開発中であり、2000年秋の稼働を目指している。DIMMnet-1を構築する論理回路は、そのRHiNET-2を構築するLSI内部に同居する形で実装される予定である。

謝辞

バンド幅に対する問題意識を強めていただいた東芝の浅野滋博氏、Linuxでwrite combining領域の設定が可能なことを教えていただいた新情報処理開発機構の手塚 宏史氏、PEMMの資料を提供しているLoomshuttle社を紹介していただいたテキサスインスツルメンツ社の鎌田晴海氏に感謝致します。特にPCIバスについてご教授いただいた上、実験システム作成にご尽力いただいたシナジェテックの清水 敏行氏に深く感謝致します。

参考文献

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

[2] 手塚, 堀, O'Caroll, 石川"RWC PC Cluster IIの構築と性能評価", *ARC*, No. 128-5, pp. 25-30 (1998)

[3] 益口 , 建部 , 関口 , 長嶋 , 佐藤"アルファワークステーションクラスタetlwizの性能評価", *ARC*, No. 128-11, pp.61-66, (1998)

[4] 住元, 堀, 手塚, 原田, 高橋, 石川 "Gigabit Ethernetを用いた高速通信ライブラリの設計と評価", 並列処理シンポジウムJSPP'99 pp.63-70 (1999)

[5] 工藤, 山本, 建部, 佐藤, 西, 天野, 石川 "PC間ネットワークによる共有アドレス空間を持つ並列処理システム", *ARC*, No. 132-21, pages 121-126 (1999.3)

[6] 山本, 工藤, 宮脇, 坂, 清水, 天野 "コモディティPCを用いた並列処理のための通信機構について", *ARC*, No. 132-20, pages 115-120 (1999.3)

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

[8] "Intel Introduces Next Generation Input/Output for Computing Servers", http://developer.intel.com/pressroom/archive/releases/sp111198.htm

[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] "メモリモジュールに装着できるDSP「PEMM」", http://www.watch.impress.co.jp/pc/docs/article/981005/expo98_c.htm

[11] "DSP搭載のメモリモジュールが登場、家庭用PCの性能を数倍向上", http://biztech.nikkeibp.co.jp/wcs/show/leaf?CID=onair/biztech/pc/49539

[12] Loom Shuttle com. "PEMM Information and Links", http://www.loomshuttle.com/PEMM.html

[13] Texas Instruments Inc. "Processor Enhanced Memory Module (PEMM) Hardware Reference Guide", http://www.loomshuttle.com/PEMMhrg.pdf

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

[15] "Direct Rambus" http://www.rambus.com/html/direct_rambus__.html

[16] National Semiconductor "高速インタフェースLVDS 5.4Gビット/秒の超高速データが10mまでシールド銅線で伝送可能", 日経エレクトロニクスno.745, pp.128-129 (1999.6)

[17] 田邊, 菅野, 鈴木, 小柳："マルチパラダイム超並列テラフロップスマシンTS/1の構想", 情報処理学会技術報告, (SWoPP 鞆の浦'93), ARC-101-6, pp,41-48 (1993.8)

[18] 鈴木, 田邊, 菅野, 小柳 : "超並列 Teraflops マシンTS/1 ～ 分散共有メモリアーキテクチャ ～", 情報処理学会, 第48回全国大会論文集, Vol.6, pp.53-54 (1994.3)

[19] 田邊："超並列テラフロップスマシンTS/1における並列処理～プロセッサ間チェイニングとその応用～", 情報処理学会論文誌,Vol.36,No.3(1995.3)

[20] 森本, 松本, 平木 "メモリベース通信を用いた高速MPIの実装方式", 並列処理シンポジウムJSPP'98論文集, pp. 191-198, (1998.6)

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

[22] 原田, 手塚, 堀, 住元, 高橋, 石川 "Myrinetを用いた分散共有メモリシステムの評価", 98-HPC-73, pp.73-78 (1998)

[23] Intel corp. "Intel Architecture Software Developer's Manual Volume.3 : System Programming Guide", http://developer.intel.com/design/pentiumii/manuals/243192.htm

[24] Intel corp. "Intel Architecture Software Developer's Manual Volume.2 : Instruction Set Reference Manual", http://developer.intel.com/design/pentiumii/manuals/243191.htm

[25] Noboru Tanabe "Memory Access Control Device with Prefetch and Read out Block Length Control Functions", United State Patent No.5,752,272 (出願 1993.3, 登録 1998.3)

[26] S. Palacharla, R. E. Kessler "Evaluating Stream Buffers as a Secondary Cache Replacement", Proc. 21st Int'l Symp. on Computer Architecture, pp.24-33 (1994.4)