社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－ARC－169 (18)
2006／8／1

DIMMnet-3ネットワークインタフェースにおけるMPI支援機能

田邊 昇[†] 北村 聡[††] 宮部保雄[††]
宮代具隆[††] 天野英晴[††]
羅 徴哲[†††] 中條拓伯[†††]

本報告では現在開発中の高機能ネットワークインタフェース DIMMnet-3 における MPI の高速化支援機能について述べる．これらの方式群の一部を DDR DIMM slot に装着される DIMMnet-2 上に実装することでその動作確認を行った．MPI 実装時の構成要素の DIMMnet-2 実機上での遅延およびバンド幅を評価した．マッチングを要する短いメッセージ受信は LHS を用いると 60 バイト以下の場合約 3μs の遅延が短縮された．ソフト処理に対し，ベクトルコマンドによればコピー性能で約 2 倍，スキャター転送，ギャザー転送では 6.8 倍の性能向上を観測した．これらは DIMMnet 上の MPI における中程度のメッセージ長を有する通信のバンド幅の向上や，派生データタイプ通信の性能向上に寄与する．

Support Functions for MPI on DIMMnet-3 Network Interface

NOBORU TANABE,[†] AKIRA KITAMURA,[††] YASUO MIYABE,[††] TOMOTAKA MIYASHIRO,[††]
HIDEHARU AMANO,[††] ZHENGZHE LUO[†††] and HIRONORI NAKAJO [†††]

In this report, support functions for MPI on DIMMnet-3 network interface which is under development. Some of them are implemented and validated on a DIMMnet-2 which is operating on DDR DIMM slot. The performance of some parts of MPI system are evaluated on a real board. Acceleration ratio of burst copy is about 2. Those of gather or scatter transfer are 6.8 compared from software on a host. These will accelerate bandwidth of medium grain message and derived datatype communications of MPI for DIMMnet.

1. はじめに

MPI(Message Passing Interface) は並列プログラム記述のために広く用いられており，デファクトスタンダードの地位を確立している．短めのメッセージ用に多用される Eager プロトコルは受信側の状態に構わず送信するので，受信側が想定した順序と異なる順序でメッセージが届くと著しい性能低下が生じる．この状況における性能とメモリ使用量の間にはトレードオフがある．

さらに，派生データタイプによる通信の高速化も重要である．この種の通信は不連続な領域へのメモリアクセスを発生するため，性能は連続転送に比べて著しく低かった．近年，アルゴンヌ国立研究所のグループの研究[1)2)]では，派生データタイプのアクセスパターンによって最適なパッキングのアルゴリズムを選択することで，その性能の改善が図られた．しかし，全てをソフトウェアで処理する方式なのでオーバーヘッドが大きい上，一部のアクセスパターンでは著しい性能低下がある．

一方，Ohio 州立大学による Infiniband の Gather/Scater 機能付 RDMA を用いた MPI の実装[3)4)]がなされ，上記におけるホスト CPU 上のソフト処理を NIC 上の CPU 上で走るファームウェア処理にオフロードしている．しかし，NIC 上の CPU はホスト CPU より一桁周波数が低いため，究極的な低遅延を実現できてはいないと考えられる．

本研究は以上の状況を鑑み，筆者らは NIC 上のハードウェアでサポートすることでホスト CPU 負荷も少なく大幅な MPI の高速化を実現することを目指している．この研究を筆者らが開発したネットワークインタフェース (NIC) である DIMMnet-2 を軸に行ってきた．しかし，ハード機構の実装が不完全な状態での MPI-2 の実装[10)]は，短いメッセージまでホストを介したベクトル命令との組み合わせで実装するという無理な実装が強いられた結果，DIMMnet-2 の MPI への適合性を疑問視する考え方を浮上させた．そこで，特に NIC の性能指標として取り上げられることが多い「最短 MPI 通信遅延時間」の短縮について，DIMMnet-2 および DIMMnet-3 は本来どのような設計になっているのかを明らかにする必要性が高まった．

本報告ではまず MPI の高速化に向けた課題を列挙する．次に稼働中の DIMMnet-2 や開発中の DIMMnet-3[6)7)] の概要を紹介し，その上に実装される MPI サポートハードウェアについて述べる．これらのハードウェアは MPI の Eager プロトコルや派生データタイプ通信の効率的な実装を可能にすることを確認すべく行った，提案方式のいくつかについて DIMMnet-2 実機上での性能評価について述べる．

† (株) 東芝，研究開発センター
Corporate Research and Developpment Center, Toshiba
†† 慶應義塾大学
Keio University
††† 東京農工大学
Tokyo University of Agriculture and Technology

2. MPIの高速化に向けた課題

本章では DIMMnet-3 における MPI のハードウェア支援を考察するに当たり，既存の NIC やソフトのみによる実装において残されている性能上の課題について述べる．特に，これらの中には最短 MPI 通信遅延時間のような NIC の基本的な通信性能の宣伝用カタログ値として使われる Hero Data のみならず，アプリケーションを実行させる際において顕著に現れてくる利用者本意の課題を中心に述べる．

2.1 短いメッセージの遅延の短縮

現在知られている多くの MPI の実装ではメッセージが短い場合は Eager プロトコル，長い場合は Rendezvous プロトコルによって実装されている．長いメッセージにおけるバンド幅は，Rendezvous プロトコルに起因するメッセージあたりにかかる初期オーバーヘッドが，効率の良いデータ転送時間によって薄められるため，比較的ハードウェアの性能を十分に出せているケースが多い．これに対し，Eager プロトコルで送られる短いメッセージは送信側主導で行われるが，受信側での処理が重く，ある程度以上の低遅延化が難しかった．

アプリケーションによってはノード数の増加に伴い，平均メッセージ長が短くなり，相対的に通信が処理時間に占める比率が上がることによってスケーラビリティに問題が生じる．

また大域通信の性能が支配的なアプリケーションでは，短いメッセージの遅延が全体の性能やスケーラビリティを左右する．

MPI における通信遅延の短縮においては単にデータの移動にかかる時間だけでなく，送信元の RANK やコミュニケータやキーによるマッチングという，通常ソフトウェアまたはファームウェアによって実現されている部分まで含めた遅延時間短縮を考慮する必要がある．

つまり，データ本体のみならず，マッチングに必要なエンベロープと呼ばれるマッチング用データを含めて，マッチングを行うべき適切な主体 (ホスト CPU または NIC のオンチップ CPU または NIC のハード) に低遅延で伝達する必要がある．

2.2 スケーラビリティ確保と高性能の両立

MPI においては Eager プロトコルで送信されたメッセージが受信側に到着した際に，受信側でまだこれに対応する受信用領域を指定する関数が実行されていなかった場合は，一旦 MPI のシステムバッファにバッファリングされる．

従来，このバッファは遠隔書き込み系の 1sided 通信でデータ転送が行われている場合は送信元ごとに確保しなければならず，スケーラビリティに問題があった．

もしくは全ての受信データを一つの共通バッファにバッファリングして受信関数の実行時に共通バッファから所望のマッチングが得られるメッセージを検索する方式の場合は，確保すべきバッファ領域を節約できるかわりに，検索に時間がかかり，遅延時間が短くならないという問題があった．

2.3 送信側と受信側がかみ合わない場合の遅延の短縮

MPI では送信側と受信側の RANK やコミュニケータやキーが一致する送信側の関数と受信側の関数の組どおしの間でデータ転送がなされる．一方，一般に並列プログラムでは複数のノードからあるノードに届くメッセージの順序を低オーバーヘッドで保証することは困難である．

このため，受信側のプログラムが期待した順序で送信側からのメッセージが届かず，前述の共通バッファ上で本来速やかに処理したかった短いメッセージが，他のノードからの長いメッセージの後にバッファリングされることにより，著しく非効率的な遅延が発生し，アプリケーションの性能を低下させることがある．

同一の組の RANK 間でのメッセージの間には先入れ先出しの順序性が確保される必要があるが，異なる RANK の組の間のメッセージの順序性は保証する必要が無いので，異なる RANK からの長い受信メッセージによって制御系に用いられがちな短いメッセージの受信が遅延させられないように工夫されることが望ましい．

2.4 MPI 処理系による CPU 浪費とキャッシュ汚染の低減

MPI のシステム受信バッファに入ったメッセージを最終的には受信関数によって指定された位置にコピーする必要がある．ホスト CPU によるソフトによるコピーを行った場合は，CPU 時間を消費させてしまうとともに，CPU のキャッシュをそのデータ転送によって汚染させてしまい，せっかく計算処理が速やかにできるようにキャッシュ上にあったデータが追い出されてしまう可能性が高まる．

Eager プロトコルと Rendezvous プロトコルの切り替えが起きるメッセージ長は MPI の実装やネットワークの環境によって異なるが，数 KB 以上にも及ぶことがあり，受信後に速やかに計算に利用されない場合は，Eager プロトコルによる受信時のキャッシュの汚染による計算時間の低下の問題が発生する可能性がある．

2.5 派生データタイプ通信の高速化

MPI の古い実装では派生データタイプ通信の性能が極端に悪かったこともあり，これまでは MPI の派生データタイプ通信を利用したアプリケーションはあまり多くなかった．しかし，MPICH2 のように派生データタイプ通信の性能がやや改善されてくると，明示的なパッキングやアンパッキングから開放される上に若干の高速化が達成されることから，派生データタイプ通信を利用したアプリケーションが今後増加してくる可能性があると思われる．しかし，派生データタイプ通信には本質的に不連続アクセスが伴い，その処理はキャッシュベースの CPU には向いていない．MPICH2 において派生データタイプ通信の性能が改善されてきたとはいえ，ソフトによる最適化がうまくいく場合のみに少しだけうまくいくレベルであるため，ソフトの限界を超えた高速化を達成できるハードを併用した高速化が望まれる．

3. DIMMnet

本章では前記の課題を解決する後述の提案方式の予備評価を行うプラットフォームとして本論文で用いた DIMMnet-2 と，提案方式を全て実装することを目標に開発中の DIMMnet-3 の 2 つのハードウェアプロトタイプの概要について紹介する．

3.1 DIMMnet-2

DIMMnet-2 は PC のシングルチャネル DDR(PC1600) スロットに装着可能なベクトル型メモリアクセス機能付き PC クラスタ用ネットワークインタフェース兼用メモリモジュールのプロトタイプである．

DIMMnet-2 は機能試作検証用のモデルであり，コストも度外視で性能追求型でもない．FPGA として XILINX 社製 Virtex-II Pro を 1 個と 128MB の SO-DIMM を 2 枚搭載しており，FPGA は現在 100MHz で動作している．性能面を評価する場合は商用の NIC は ASIC でこの 2 倍程度の周波数で動作させている点を考慮して，本プロトタイプの性能を割り増して見る必要がある．さらに Infiniband 4X の市販スイッチに接続することで PC クラスタ用ネットワークインタフェースとしても機能している．

DIMMnet-2 のベクトル転送コマンドにはローカルに作用するコマンドと，リモートノードに作用するコマンドに大別され，大半のコマンドにはローカル用とリモート用の 2 種類が用意されている．

コマンドの中には DIMMnet-1 において初めて実装された BOTF(Block On-The-Fly) 送信コマンドもある．BOTF はプロテクション情報以外の位置には自由なヘッダーフォーマットを有する柔軟性が高いパケットを低遅延でネットワークに送出する機能で，主に短いパケットの生成に用いられる．後述する本論文の通信実験にはこの BOTF も用いられる．

DDR の高速メモリバスに対応するために，DIMMnet-2 では DIMMnet-1 とは異なり，FPGA 上にベクトルレジスタとして機能するリード用とライト用の Window メモリを搭載し，SO-DIMM またはリモートの SO-DIMM との間のベクトル型のデータ転送命令を備えている．つまり，ホストからは直接仮想空間に SO-DIMM 領域をマップして読み出すことが BIOS のタイミング調整範囲を超えるためできない．

この点が DIMMnet-2 を最近米国を中心に流行している UPC や CAF(Co Array FORTRAN) などの 1sided 通信主体で実装される PGAS(Partitioned Global Address Space) モデルではなく，MPI のようなメッセージ交換モデルでの利用時，例えば HOKKE'06 の発表での MPI2 の実装 [10] において，性能低下要因として問題になっていた．

さらに，ネットワークから流入するパケットが LLCM というホストから低遅延でアクセスできるオンチップメモリに受信できないという不具合を抱えていたことも MPI の高遅延化につながってしまった．

しかし，最近の機能拡張により，後述する LHS 機能や VCOPY 機能が実装され，MPI による短いメッセージ通信の高速化や，ハードによる SO-DIMM 間のコピー，ギャザー転送，スキャター転送などが DIMMnet-2 の実機上で可能となった．本報告ではこれらの新機能を中心に評価を行う．

3.2 DIMMnet-3

DIMMnet-3 は DIMMnet-2 におけるいくつかの実用上の問題点を解決することを目指し，動作周波数，コスト，サイズ，メモリ容量，信頼性の面で，より実用に近いレベルのプロトタイプであり，その基本構想は SWoPP'05 において発表した．

DIMMnet-3 ではチップ分割による部品コスト低下，大幅なオンボードメモリ容量の増強，ECC による信頼性向上をはかっている．さらに，より新型で多様なホストへの適用性の強化をはかっており，DDR2 ベースのデュアルチャネル型主記憶を有するパーソナルコンピュータ (PC) と東芝 Cell リファレンスセット (CRS) への装着を可能とすることを目標に開発中である．

図 1 に示すような XILINX 社の現在の主流の FPGA である Virtex4FX を 1 個搭載した PC 用の DDR2 スロットに装着される子基板が 2005 年度に試作された．この子基板は 200MHz の DDR2(DDR2-400)CAS レイテンシ 5 のメモリとして PC から認識および連続アクセス可能で，装着した状態で Linux OS が立ち上がることを確認できている．

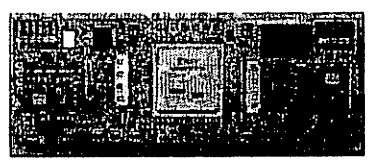

図 1 PC 用 DIMMnet-3 子基板

さらに，2006 年度に XILINX 社の FPGA である Virtex4FX を 1 個搭載した親基板を設計中である．PC 上にこれらを装着する場合は図 2 に示すように，最大 2 枚の DDR2 スロット上の子基板と，PCI スロット上に装着される親基板の間が，高速シリアルリンクである RocketIO でケーブル接続される．

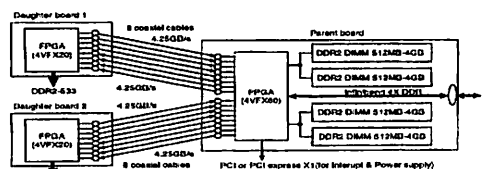

図 2 PC 用 DIMMnet-3 における基板間接続

親基板は 1U サイズの筐体に収納可能なサイズで実現され，最大 32GB のメモリを装着可能とする．単なるネットワークインタフェースとしてみた場合はこのメモリ容量は異常に大きいが，PC あたりの搭載メモリ容量と不連続アクセスを改善し

た高機能メモリとしてデータベース応用や HPC 応用にも利用できるような設計となっている．

CRS に上記親基板を装着する場合は図 3 に示すように，CRS 上の DDR2 ベースの SO-DIMM ソケットから，パラレルのまま短いケーブルで親基板まで Super Companion Chip の DDR2 信号を導くことでホストと親基板の間の接続を行う予定である．

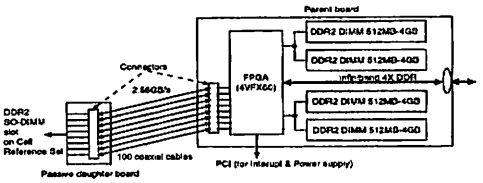

図 3 Cell リファレンスセット用 DIMMnet-3 における基板間接続

4. 提案方式

本章では DIMMnet-3 における MPI のハードウェア支援として提案または紹介する方式について述べる．ここで提案している方式は DIMMnet-3 への実装を念頭には入れているが，ベクトル化と関係ないものは必ずしも DIMMnet の基本構成を前提とする方式ではない．

4.1 遠隔間接書き込み IPUSH

SWoPP'99[8] において筆者はメッセージ交換の低遅延高バンド幅化を行うために，遠隔間接書き込みおよび遠隔 FIFO 書き込みを提案した．DIMMnet-2 においては既にこの機構は IPUSH という名称のハードウェアによって，より洗練された形態の実装がなされた．MPI の 2sided 通信が送信側から見ると 1sided 通信で行われ，それが IPUSH ではハードの機能で実現されているため，1sided な RDMA 通信とほぼ同等の遅延時間およびバンド幅で，受信側が指定した場所に遠隔書き込みを行うことができる．

IPUSH では受信側の NIC 上のテーブルへの設定を調整することにより，通信頻度の高い相手からの通信は独立した受信バッファに受信し，それ以外の相手からの通信は共通の受信バッファに受信することもでき，これによって受信バッファ領域の大幅な削減と低遅延化の両方が達成される．IPUSH の詳細は SACSIS'06[9] および論文誌 ACS-15 に掲載されているので参照されたい．

4.2 有限長メッセージ頭部分別 LHS

MPI における短いメッセージの遅延短縮と，受信側のシステムバッファ検索コストの短縮によるメッセージ遅延短縮のために，有限長メッセージ頭部分別 (LHS : Limited-length Head Separation) を提案する．

有限長メッセージ頭部分別 LHS の基本構造を図 4 に，LHS は「受信側に到着したメッセージの長さが事前に指定された長さ以下の場合は低遅延な高速バッファ(LH バッファ) に保存し，それを超える長さのメッセージを受信した場合は，後半部へのポインタ，または後半部へのポインタと前半部を LH バッファに保存するとともに，後半部を大容量バッファに保存する受信方式」である．

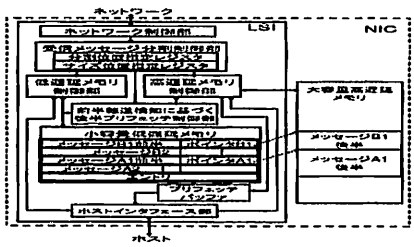

図 4 有限長メッセージ頭部分別 LHS の基本構造

LHS による送信側と受信側がかみ合わない場合の遅延の短縮の様子を図 5 に示す．メッセージ B1,B2,A1,A2 という順番で受信側に届き，MPI のシステムバッファにバッファリングされたパケットが，従来は全て高遅延大容量メモリから読み出され退避されていたものが，LHS を用いた場合は全てのエンベロープ部の転送が低遅延な LH バッファから行われるので大幅に高速化する．

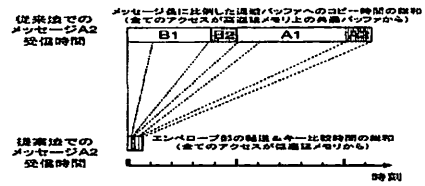

図 5 LHS による送信側と受信側がかみ合わない場合の遅延の短縮の様子

ここで大容量バッファは主記憶上のピンダウン領域に確保しても良いし，DIMMnet-2 の SO-DIMM 領域のようなネットワークインタフェース上のオフチップメモリに確保しても良い．よって，原理的にはこの方式は Myrinet などのファームウェアで実装することも可能である．高速バッファは主にネットワークコントローラ LSI 上のオンチップメモリを想定し，大容量バッファより低遅延であるが容量は少ない．

現状では DIMMnet-2 上に LHS の基本機能の実装が完了した現状の実装では閾値より長いメッセージの場合はポインタのみを LH バッファに格納する方式を実装している．DIMMnet-3 においても LHS は実装される予定である．

DIMMnet-2 における LHS を用いない場合の受信側ノードでの処理は以下のようになっていた．

(1)SO-DIMM からホストへのエンベロープのベクトルロードコマンド VL による読み出し
(2) ホストでエンベロープのマッチング
(3)SO-DIMM からホストへのボディのベクトルロードコマンド VL による読み出し
(4)MPI_Receive() で指定された領域にコピー

これに対して LHS を用いた場合の受信側ノードでの処理はエンベロープ込みで LH バッファの 1 エントリ以下のサイズの場合は以下のようになる．

(1)LH バッファからホストへのエントリ (エンベロープ+ボディ) の PIO による読み出し
(2) ホストでエンベロープのマッチング
(3)MPI_Receive() で指定された領域にコピー

このようにホストからは直接見えずベクトルロードコマンドを使わなければホストから見える位置にデータが出てこない構成になっている DIMMnet-2 の場合は短いメッセージの MPI 受信遅延時間が大幅に短縮されることが期待できる．

4.3 頭部取得による後半プリフェッチ HTP

中程度の長さを持つメッセージの遅延短縮とバンド幅向上のため，有限長メッセージ頭部分別と組み合わせて，頭部取得によるメッセージ後半プリフェッチ (HTP : Head-transfering Triggered Prefetching) を提案する．

図 4 には HTP を行うための監視部とプリフェッチバッファが付いた構成を示している．本方式は，前記の LHS において LH バッファへのホストからのアクセスを監視し，その際にホストに LH バッファから後半部へのポインタ付のエントリが転送されたことを契機として，ホストからの要求に先立ち，後半部を大容量バッファからプリフェッチバッファにプリフェッチを開始することである．

現状では DIMMnet-2 上には HTP の機能は実装されていないが DIMMnet-3 においては実装される予定である．

4.4 ベクトル化データコピー VCOPY

メッセージ交換におけるノード内のデータコピーは通常ソフトウェアによって行われる．しかし，ソフトウェアによるコピーは CPU 時間の浪費，主記憶バンド幅の浪費により，結果として処理と通信の並行実効の阻害を伴うとともに，CPU 内部のキャッシュの汚染による副作用も伴う．

そこで，上記のような問題の解決手段として，ベクトル化データコピー (VCOPY : Vector Copy) を提案する．本方式は MPI などのメッセージ交換におけるバッファがベクトル転送命令によってアクセス可能なメモリ領域にある場合，ハードウェアで実装されたベクトル転送命令によってバッファ間のデータコピーを行う方式である．

これによって，ホストが介在しないことや，主記憶バンド幅以上のバンド幅をハードが用いることが可能なことなどによるコピー自体の高速化に伴う通信遅延の低下が実現されるとともに，メッセージのバッファ間コピーを行う際の CPU 時間の浪費や，主記憶バンド幅の浪費を排除し，その間の CPU による処理の並行実行可能性を高める．

例えば DIMMnet-2 や DIMMnet-3 のようにホストの主記憶と DIMMnet 上の大容量オンボードメモリが全く別のバン

ド幅でアクセス可能な構成においては，VCOPY によって受信メッセージのバッファ間コピー中に，ホスト CPU は CPU 時間や主記憶バンド幅をコピー作業により奪われることなく，主記憶上にあるデータを使った処理を並行実行することができる．

現状では DIMMnet-2 上に VCOPY の基本機能の実装が完了した．DIMMnet-3 においても VCOPY は実装される予定である．

4.5 ベクトル化派生データタイプ通信 VDDC

不連続アクセスを伴う派生データタイプ通信の高速化のために，ベクトル化派生データタイプ通信 (VDDC : Vectorized Derived Datatype Communication) を提案する．本方式は MPI の派生データタイプ通信における不連続アクセスパターンをハードウェアで実現されるベクトル命令の列に変換し，これによって派生データタイプ通信を高速化するものである．図 6 にローカルでギャザーしたデータをリモートの MPI バッファに書き込み，それをリモートでスキャターするような派生データタイプ通信を行う場合の流れを示す．

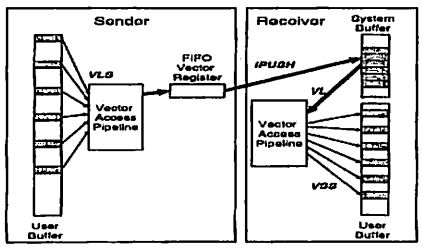

図 6 ベクトル化派生データタイプ通信

本機能は既に DIMMnet-2 に実装済みのベクトルコマンドをソフトウェア的に組み合わせることによって実現することができる．現状の DIMMnet-2 ではベクトル転送命令において通常のベクトルレジスタの他に，内部の FIFO ベクトルレジスタをオペランドとして指定することが可能である．これらの FIFO ベクトルレジスタを介してチェイニングされるベクトル転送命令によって柔軟なギャザー処理，スキャター処理を高速に実行することができる．配列の転置を行う際に等間隔ベクトルロードコマンド VLS と連続ベクトルストアコマンド VS を FIFO でチェイニングする例については HPCS'06 の論文 [13] において提案済みである．DIMMnet-3 においても同様のチェイニング機構を実装し，VDDC に基づく MPI が実装される予定である．

4.6 二相メッセージ交換 TPMP

MPI における中程度以上のサイズを有するメッセージの遅延短縮とバンド幅向上を目的に，二相メッセージ交換 (TPMP : Two Phased Message Passing) を提案する．図 7 に TPMP の動作の概念図を示す．本方式はある程度以上の長さをもつメッセージを送信側で二分割し，送信側のデータへのポインタを有するエンベロープを含む前半を前述の LHS などを用いて Eager プロトコルで受信させつつ，後半を Rendezvous プロトコル流に遠隔読み出しにより受信側からユーザー受信バッファに取り込むことで，前半の処理と後半の処理をオーバーラップさせ，遅延短縮とバンド幅向上を実現する．

本方式はハードウェア機構としては IPUSH や LHS があることが望ましいが，最低限，遠隔読み出しまたは遠隔書き込みがあれば実現できる．よって現状の DIMMnet-2 においてソフトウェア的に実装が可能であるが現状では未実装であり，今後 DIMMnet-3 用の MPI において実装される予定である。

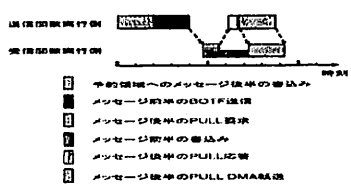

図 7 二相メッセージ交換 TPMP の動作の概念図

5. 性能評価

本章では，前記の提案方式のうち IPUSH, LHS, VCOPY, VDDC の四つに関して予備的な評価を行うことにより，これらの MPI への効果を考察する．

5.1 評価環境

本章の実験において用いられた評価環境を以下に示す．

- CPU : Pentium4 2.6GHz, L2=512KByte
- Chipset : VIA VT8751A
- Memory : PC-1600 DDR-SDRAM 512MB x1
- OS : RedHat8.0(kernel 2.4.27)
- Compiler : gcc3.3.5(Compile option:-Wall)
- Network:InfiniBand Swtich(Voltaire ISR6000)
- cable : 2m

5.2 IPUSH

5.2.1 測定方法

前述の環境でスイッチを介して 2 台のノードを接続し，以下の 4 項目に関して Ping-Pong でバンド幅を測定した．なお，(3),(4),(5),(6) については 2048Byte 以上のデータコピーに関しては Prefetch Window を 4 枚使用した．

(1) PUSH による遠隔 SO-DIMM 間転送
(2) IPUSH による遠隔 SO-DIMM 間転送
(3) (1) ＋主記憶へのデータのコピー
(4) (2) ＋主記憶へのデータのコピー
(5) (3) においてコピー先のサイズを固定した場合
(6) (4) においてコピー先のサイズを固定した場合

5.2.2 結　　果

結果を図 8 に示す．(1) と (2),(3) と (4),(5) と (6) はそれぞれほとんど性能の差がない．つまりマッチングの無い単純な 2sided 通信が IPUSH により 1sided 通信である PUSH とほぼ同等の性能で実行できている．主記憶へのコピーを行った場合，最大バンド幅が IPUSH でが 236.8MB/s，PUSH で 237.4MB/s となった．ただ，転送サイズを大きくしていくと，最終的には 170MB/s 程度で落ち着いてしまう．この値は SO-DIMM 間転送のバンド幅 (最大で約 680MB/s) に比べるとかなり落ちている．主記憶にコピーする領域のキャッシュ属性は Write Back としており，(5)(6) でこの領域のサイズを固定して測定を行った結果，128KB〜4MB の範囲でバンド幅は低下しなかった．

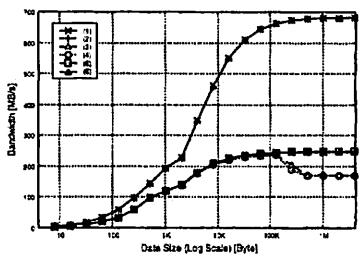

図 8 PUSH および IPUSH のピンポンバンド幅

5.2.3 考　　察

(5)(6) の結果より，主記憶へのコピーをする際に，メッセージサイズがキャッシュのサイズと同等レベル以上に大きくなるとキャッシュの汚染が進み，リプレースに伴って性能低下が起きていることが判明した．受信先がホストの主記憶であった場合は現状の DIMMnet-2 ではホストの主記憶に DMA する機能はないため，メッセージサイズが大きい場合は上記のような問題点が残ることがわかる．しかし，特定の配列のアロケーション位置を SO-DIMM 側に指定できる言語との組み合わせで用いられる拡張版の MPI で SO-DIMM 側に受信する場合は，VCOPY によるホストを介在しないコピーの効果が期待でき，上記の問題は発生しない．

5.3 LHS

5.3.1 測定方法

前述の環境でスイッチを介して 2 台のノードを接続し，以下の 2 項目に関して送信側を BOTF を用いて Ping-Pong で遅延時間を測定した．

(1)LHS を用いない場合の MPI を模擬した通信
送信側が BOTF で LHS を起動しない従来の IPUSH を起

動するヘッダーを有するメッセージを送信し, SO-DIMM 上のバッファに取り込み, ホストからその完了フラグをポーリングして, SO-DIMM 上のバッファからエンベロープをベクトルロードコマンド VL を用いて Prefetch Window 経由でホストが読み取り, そこに記載された長さで再度 VL を用いて Prefetch Window 経由でメッセージのボディ部をホストが読み取り, 主記憶に書き込む.

(2)LHS を用いる場合の MPI を模擬した通信

送信側が BOTF で LHS および IPUSH を起動するヘッダーを有するメッセージを送信し, LH バッファからホストへのエントリ (エンベロープ+ボディ) の Programmed I/O による読み出し, メッセージのボディ部を主記憶に書き込む.

5.3.2 結　　果

結果を図 9 に示す. LHS を用いると LH バッファよりも小さなボディ部を持つメッセージでは VL コマンドの起動が排除されるため, 60 バイト付近を境に約 650ns の遅延の削減が観測された. これは LH バッファから読み出すのか, SO-DIMM から読み出すのかによる差である. さらに, (2) の遅延時間は (1) の遅延時間よりも 2 回分の VL コマンドの起動が排除されるため, LH バッファよりも小さなボディ部を持つメッセージでは約 3μs, それ以上でも約 2μs の遅延の削減が観測された.

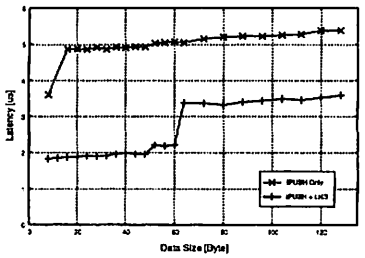

図 9　LHS による通信遅延時間の変化

5.3.3 考　　察

LHS は DIMMnet の MPI 実装において最短 MPI 通信遅延の短縮をもたらすことが確実と考えられる. 一方, メッセージ到着順序が受信側の想定する順序とは異なった場合でも, エンベロープの受信はアクセス遅延が長い SO-DIMM からではなく, 遅延が短い LH バッファから読み出されるため, 遅延の悪化を抑制できると考えられるが, その点の確認実験は今後の課題である.

5.4 VCOPY

5.4.1 測 定 方 法

前述の環境で 1 台のノードを用い, 以下の 3 項目に関して連続データのコピーのバンド幅を測定した.

(1) ソフトによるホストの主記憶間コピー

(2)Window 間移動で部分的にホストを介した SO-DIMM 間コピー

(3)VCOPY(VL → FIFO → VS の chaining) を用いた SO-DIMM 間コピー

5.4.2 結　　果

結果を図 10 に示す. chainig がない状態 (2) ではホスト処理 (1) にやや劣っていたが (3) においてはこれらを大幅に改善し最大 522MB/s のコピーがホストによるデータ移動なしに実行された.

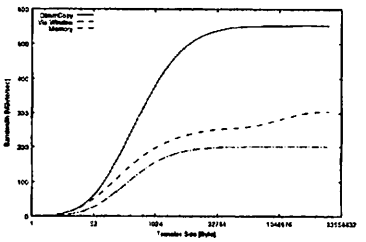

図 10　VCOPY による SO-DIMM 間コピーバンド幅の変化

5.4.3 考　　察

DIMMnet-2 の現状の実装状態においては SO-DIMM へのバンド幅は 3.2GB/s あるので 1.6GB/s のコピー性能が期待できるのに対して, 連続アクセスであるにもかかわらず, 最大 522MB/s という結果は実装状態に改善すべきものが残っていることが示唆される結果を得たと言える. 現時点で判明している現状の実装状態の問題点としては, 不連続アクセス性能を重視した設計であるため, DDR SDRAM へのバースト長を最小の 2(1 チャネルあたり 16 バイト) に設定され, 同じ Row, Bank に対してアクセスする場合 1clock おきにしか Column アドレスを投入できない点, Write Unit と Prefetch Unit が個別に DIMM I/F へのアクセスを制御してため, 切り替え時にコマンドの空きが出来る点が挙げられる.

起動状態にあるコマンドが連続系アクセスのみの場合は SO-DIMM I/F がバースト長を 4 または 8 に切り替えるよう改造すれば, 連続コピーでのバンド幅半減は回避できると考えられる. また, DIMMnet-3 では少なくとも周波数は 2 倍になるのでホストからのベクトルコマンド起動時間以外はさらに概ね 2 倍の性能を出すことが予想される.

さらに, このバンド幅でのコピーが SO-DIMM が最終受信場所に指定されている場合にはホストの CPU 時間もメモリバンド幅も消費せず, ホスト CPU のキャッシュや TLB の汚染も起こさずに実現可能である. この効果に対する評価は今後の課題である.

5.5 VDDC

5.5.1 測 定 方 法

前述の環境で 1 台のノードを用い, 以下の 2 項目に関して不連続データの SO-DIMM 間ギャザー転送および SO-DIMM 間スキャター転送のバンド幅を測定した. 全てデータのタイプは 8 バイトで, ストライドは 1024 または 1032 と変化させた.

(1) ソフトによるホストの主記憶間 gather 転送

(2)Window 間移動で部分的にホストを介した SO-DIMM 間 gather 転送

(3)VCOPY(VLS → FIFO → VS の chaining) を用いた SO-DIMM 間 gather 転送

(4) ソフトによるホストの主記憶間 scatter 転送

(5)Window 間移動で部分的にホストを介した SO-DIMM 間 scatter 転送

(6)VCOPY(VL → FIFO → VSS の chaining) を用いた SO-DIMM 間 scatter 転送

5.5.2 結　　果

(1)〜(3) の結果を図 11 に, (4)〜(6) の結果を図 12 示す. ソフトによる主記憶間 gather 転送や scatter 転送はキャッシュライン中の有効データ比率が少なく, ストライドが大きい場合は TLB ミス頻度も高くなるので, 著しい性能低下が発生する. これに対して等間隔アクセス系のベクトルコマンドを用いた VCOPY を用いる方法では, 約 6.8 倍性能が向上する. このうち chaining は 2 倍程度の効果を示した.

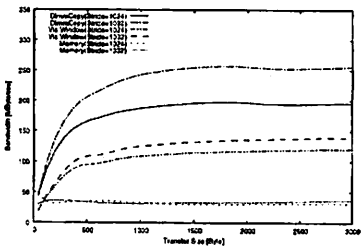

図 11　gather 転送バンド幅への VLS,VS コマンドおよび chaining の効果

5.5.3 考　　察

派生データタイプ通信の送信側あるいは受信側で等間隔の配置を指定している派生データタイプを有する通信を行う場合に, 上記の VCOPY による gather 転送や scatter 転送が有効に機能すると考えられる. ただし, 性能の絶対値としては現状の実装ではハードウェアを用いている割にはあまり高いものではない. その一つの原因としてはベクトル型スーパーコンピュータの主記憶のように多数のバンクが DIMMnet-2 上には実装できておらず, 結果として不連続データへのアクセスバンド幅が十分に上がっていない点などが考えられる.

6. 関 連 研 究

短いメッセージの遅延時間の短縮については, QsNET-II や

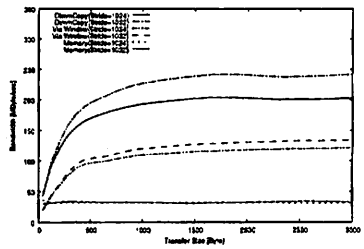

図 12 scatter 転送バンド幅への VL,VSS コマンドおよび chaining の効果

Infinipath といった商用 PC クラスタ用 NIC においてはいくつかの試みがなされている．QsNET-II の場合は STEN と呼ばれる短いパケット送信専用の送信機構を準備している．この点は DIMMnet-1 や DIMMnet-2 における AOTF や BOTF といった短いパケット送信専用の送信機構を準備する方針と類似している．しかし，本報告で提案している LHS や HTP のような受信側の高速化については，Thread Processor と呼ばれるオンチップ CPU 上のファームウェアにマッチング処理をオフロードしている点以外に特に目立った機構を持っているという報告がない．

Infinipath[12] は内部構造の詳細が不明であるが，オンチッププロセッサや DMA を排除していると言われており，ホスト CPU から MPI の機能本体をオフロードしない点は DIMMnet-2 や DIMMnet-3 と同様である．DIMM スロットではなく Hypertransport の専用スロットである HTX に装着されることで他の NIC よりも遅延時間の短縮を図っている．しかし，オンボードの大容量メモリは持たずに，届いたデータは即座にホストに退避しないとネットワーク側の輻輳を誘発すると思われ，大容量のオンボードメモリを搭載する DIMMnet-2 や DIMMnet-3 とは異なる．このようなアプローチはマルチコア CPU 時代にはやや優勢になってくるものの，通信専用に常時，受信データを主記憶に退避するスレッドが必要になってくるので，CPU 負荷を DIMMnet 以上に上げるとともに，キャッシュの汚染が激しくなると思われる．

派生データタイプによる通信の高速化は，近年，ANL のグループの研究 [1][2] では，派生データタイプのアクセスパターンによって最適なパッキングのアルゴリズムを選択することで，その性能の改善が図られた．しかし，全てをソフトウェアで処理する方式なのでオーバーヘッドが大きい上，一部のアクセスパターンでは著しい性能低下が残っている．

一方，Ohio 州立大学による Infiniband の Gather/Scater 機能付 RDMA を用いた MPI の実装 [3][4] がなされ，上記におけるホスト CPU 上のソフト処理を NIC 上の CPU 上で走るファームウェア処理にオフロードしている．しかし，NIC 上の CPU はホスト CPU より一桁周波数が低い上，ストライドあたりのデータの塊のサイズが小さい場合の NIC 上の CPU のキャッシュミス多発の問題の解決にはなっていないものと思われ，究極的な低遅延を実現できてはいないと考えられる．

MPI の実装ではないが，Nieplocha らは ARMCI[5] という等間隔アクセスパターンを定義可能な通信用 API を定義しており，Infiniband で結合されるクラスタ上に実装して効果を得ている．ただし，このアプローチは高機能な動作を定義できる API によって関数の起動回数を削減することによって実現されており，ストライドあたりのデータの塊のサイズが小さい場合のキャッシュミス多発の問題の解決にはなっていないものと思われる．

NIC 上のハードウェアで高速化をはかる試みとしては，富士通による UZURA[6] 上での行列転置に特化したコーナーターナー的ハードウェアを導入した例が報告されている．この方式は PCI-X バス上のデータ転送を行列転置用に限定して最適化をしているものであり，派生データタイプによる通信のごく一部にしか適用できず，応用範囲が狭い．

7. ま と め

本報告では現在開発中の高機能ネットワークインタフェース DIMMnet-3 における MPI の高速化支援機能について述べた．既に提案済みの IPUSH に加え，有限長メッセージ頭部分別 (LHS)，頭部取得によるメッセージ後半プリフェッチ (HTP)，ベクトル化データコピー (VCOPY)，ベクトル化派生データタイプ通信 (VDDC)，二相メッセージ交換 (TPMP) のコンセプトを提案した．

ここで紹介または提案される方式群の一部 (IPUSH, LHS, VCOPY) を DDR DIMM slot に装着される DIMMnet-2 プロトタイプ上に実装することでその動作確認を行った．MPI を実装する際の構成要素の DIMMnet-2 実機上での遅延およびバンド幅の評価結果を示した．マッチングを要する短い 1 メッセージの高速受信機構 LHS を用いると 60 バイト以下のメッセージの場合は約 3μs の遅延短縮が得られた．ソフト処理に対し，ベクトルコマンドによればコピー性能で約 2 倍，スキャター転送，ギャザー転送では 6.8 倍の性能向上を観測した．これらは DIMMnet 上の MPI における中程度のメッセージ長を有する通信のバンド幅の向上や，派生データタイプ 1 通信の性能向上に寄与する．

メッセージ到着順序が受信側の想定する順序とは異なった場合の LHS の効果の評価は今後の課題である．DIMMnet-3 の実機の開発を進め，DIMMnet-2 上で現段階では試作できていない HTP や TPMP の実装・評価や，これらの機能を DIMMnet-3 に移植するとともに，それらの機能を用いた MPI の実装を行い，有効性を示すことが今後の課題として挙げられる．

謝辞 本研究は総務省戦略的情報通信研究開発推進制度 (SCOPE) の一環として行われたものである．DIMMnet-2 および 3 の開発に関する議論にご参加いただいている慶應義塾大学の西牧教授，渡辺氏，大塚氏，伊沢氏，東京農工大学の並木助教授，浜田氏，荒木氏，木立氏，森氏，金井氏，池田氏，立命館大学の国枝教授，妻氏，森山氏，高柳氏，福田氏，原岡氏，和歌山大学の齋藤講師，京都大学の上原助教授，日立 IT 社の上垣氏，今城氏，岩田氏，森山氏に感謝いたします．

参 考 文 献

1) R. Ross, N. Miller, and W. Gropp : "Implementing fast and reusable datatype processing", In Proceedings of the 10th EuroPVM /MPI Conference, pp.404-413 (Sep. 2003)
2) S. Byna, W. Gropp, X. Sun, and R. Thakur : "Improving the performance of mpi derived datatypes by optimizing memory-access cost", IEEE International Conference on Cluster Computing (CLUSTER2003), pp.412-419 (Dec. 2003)
3) J. Wu, D. K. Panda, and P. Wyckoff : "High Performance Implementation of MPI Derived Datatype Communication over InfiniBand", 18th International Parallel and Distributed Processing Symposium (2004)
4) S. P. Kini, J. Liu, J. Wu, P. Wyckoff, and D. K. Panda : "Fast and Scalable Barrier using RDMA and Multicast Mechanisms for InfiniBand-Based Clusters", Euro PVM/MPI Conference, (Sep. 2003)
5) J.Nieplocha, V.Tipparaju, M.Krishnan, D.K.Panda : "High Performance Remote Memory Access Communication : The ARMCI Approach", International Journal of High Performance Computing Applications, Vol.20, No.2, pp.233-253
6) 田邊, 羅, 濱田, 中條, 北村, 宮代, 宮部, 天野 : "DIMM スロット装着型デバイス DIMMnet-2 の改良方針", 情報処理学会計算機アーキテクチャ研究会, 2005-ARC-164, pp.127-132 (Aug. 2005)
7) 田邊, 羅, 並木, 中條, 天野 : "メモリ周りに制約を有する MPU におけるプリフェッチ機能付メモリモジュールの意義", 情報処理学会計算機アーキテクチャ研究会, 2006-ARC-167, pp.13-18 (Feb. 2006)
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" 情報処理学会計算機アーキテクチャ研究会, Vol. 99, No. 67, pp. 73-78, (1999.8)
9) 北村, 宮部, 中條, 田邊, 天野 : "メッセージパッシングモデルを支援するパケット受信機構の DIMMnet-2 への実装と評価", 先進的計算基盤システムシンポジウム SACSIS2006, pp.359-366 (May 2006)
10) 荒木, 森, 金井, 田邊, 天野, 並木, 中條 : "DIMMnet-2 における通信ライブラリ MPI-2 の実現", 情報処理学会計算機アーキテクチャ研究会, 2006-ARC-167, pp.49-54 (Feb. 2006)
11) J. Beecroft, D. Addison, D.Hewson, M. McLaren, D. Roweth, F. Petrini and J. Nieplocha "QsNET II : Defining HighoPerformance Network Design", IEEE MICRO, Vol.25, No.4, pp.34-47 (Jul. 2005)
12) D.W.Doerfler "An Analysis of the Pathscale Inc. Infiniband Host Channel Adapter, Infinipath", SANDIA REPORT SAND2005-5199 (Aug. 2005)
13) 田邊, 羅, 中條, 稲崎, 安積, 土肥, 北村, 天野 : "プリフェッチ機能を有するメモリモジュールによる等間隔アクセスの高速化", ハイパフォーマンスコンピューティングと計算科学シンポジウム (HPCS2006), pp.55-62 (Jan. 2006)
14) 中島, 佐藤, 後藤, 住元, 久門, 石川 : "配列転置データ転送を高速化する 10Gb Ethernet インタフェースカードの設計", 先進的計算基盤システムシンポジウム SACSIS2006, pp.127-134 (May 2006)