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

2007－ARC－174 (24)
2007／8／2

メッセージ頭部の格納場所切替によるメッセージ交換の高速化

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

PCI Expressやトランキングの採用により，近年のMPIの通信バンド幅の向上は著しいのに対し，通信遅延の短縮は難しい．本論文ではメッセージ頭部の格納場所切替によりメッセージ交換を高速化する通信機構であるLHS(Limited-length Head Separation)の性能とハードウェア量を評価する．LHSはDIMMスロットに装着されるNICであるDIMMnet以外のNICでも適用可能だが，NIC上に大容量で高遅延なメモリを有するDIMMnetにおいては特に効果が大きいことが予想される．DDR DIMMスロットに装着されるDIMMnet-2の実機上でLHSの動作確認と効果の評価を行った．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unexpected messages queue内のメッセージ検索速度は，LHSを使う場合は使わない場合と比較して43倍高速化した．この絶対性能は低遅延なMPIのオフローディングで有名な商用NICであるQsNET-IIの6.8倍に当たる．一方，ハードウェア量は上記の検索の加速器であるALPUに比べて大幅に少なく，LHSはより大規模なシステムに適していることも判った．

Accelerating Message Passing with Switching Location to Store Heads of Messages

Noboru Tanabe,[†] Akira Kitamura,[††] Yasuo Miyabe,[††] Tomotaka Miyashiro,[††]
Hideharu Amano[††] and Hironori Nakajo [†††]

PCI Express and network trunking make the increase of bandwidth for MPI significant. On the other hand, decreasing latency for message passing is difficult. In this paper, support function named LHS(Limited-length Head Separation) is evaluated. This accelerates message passing by means of switching location to hold head parts of messages. Though this can be adopted on DIMMnet and the other NICs, the effect on DIMMnet with large capacity and long latency may be bigger. This is validated and evaluated on a DIMMnet-2 which is operating on a DDR DIMM slot. Searching speed of unexpected messages queue when the order of message reception is different from receiver's expectation is accelerated in 43 times with LHS. This absolute performance is 6.8 times higher than QsNET-II which is famous as a low latency comercial NIC with MPI off-loading. Hardware cost of LHS is significantly lower than that of ALPU which is a hardware accelerator for searching written above. Therefore, LHS is better suited for larger parallel systems.

1. はじめに

メッセージ交換のAPI(Application Program Interface)であるMPI(Message Passing Interface)はPCクラスタや並列計算機上での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．NIC(Network Interface Card)上の複数のポートまたは複数のNICを用いたトランキングの技術やPCI Expressの採用[1]により，長いメッセージのバンド幅の向上は比較的容易になってきているのに対し，短いメッセージの遅延時間の短縮は困難である．

MPIの代表的な実装においてはEagerプロトコルとRendezvousプロトコルの二種類が使い分けられる．短めのメッセージ用に用いられる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，unexpected messages queueと呼ばれるバッファに一旦保持される．その後，受信側がコールした受信関数の中でunexpected messages queue内のメッセージの検索が発生するため著しい性能低下が生じる場合がある．Brightwell等の研究[2]によると，NAS Parallel Benchmarks(NPB)のFTやISなどにおいてはノード数に近い多くのメッセージがunexpected messages queueに滞留するため，その中から受信キーが一致するメッセージを検索する時間の増加により通信遅延が増加する．

Underwoodらの研究[3]では，標準的なMPI性能のベンチマークにおいては測定されていない上記のような状況の特性を測定するベンチマークを提案し，各種NICにおけるunexpected messages queue内のメッセージの検索遅延が測定されている．例えばその論文[3]で報告されているMyrinet(LanaiX)上におけるGMを用いたMPI実装では，上記の検索をNICにオフロードせず，ホストCPU上で行ない検索遅延増加が抑制されている．しかし，この方式では最大バンド幅とならび重視される性能指標である最短遅延時間が6μ秒へと損なわれてしまっており，トータルとしてはあまり望ましくない．

一方，QsNET-II[4]においては短いメッセージの遅延時間を短縮するためにNIC上のオンチップCPUのファームウェアにMPIのマッチング処理をオフロードしている．しかしNIC上のCPUはホスト上のCPUに比べ大幅に非力となるため，QsNET-II流のファームウェアオフロード方式には限界がある．マルチコア化を背景に急速にホストCPU能力が向上しているトレンドからは必ずしもベストではない．例えばQsNET-IIでは1個の滞留メッセージあたり100nsのペナルティが生じる．1000ノードクラス以上の大規模並列システム上で滞留メッセージが多数発生する局面では，前述のMyrinetの例のように検索をオフロードしない方式より，遅延時間が大幅に劣化する．

一方，Underwoodらのその後の研究[5]によると，NIC上のCPUのファームウェアと連携して動作するALPUと呼ばれるハードウェアによるMPIのキュー操作の高速化機構の提案と評価が行われた．ALPUはレジスタと比較器を主体とする論理ゲートで実現されたセルを待機可能メッセージ数に対応した個数並べるため，キューへの挿入や検索自体は大変高速である．しかし，大量のハードウェアを必要とするため比較的少数(256個)のセルしかNICのLSI上に搭載することが難しい．このため，ノード数がALPUのセル数より大きなシステムではALPUのセル数を超えた時点で，ソフトウェアによる例外処理が必要になり，大幅な性能低下が発生することが報告されている．4コアのCOTS CPUが市販されているが，64ソケット程度の並列システムでALPUは限界点に達してしまう．マルチコア化が進行し，コア数の多いシステムの構築が容易になってきている流れから，これは問題である．

本研究は以上の状況を鑑み，最短通信遅延の短縮を目指しつつ，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場

† (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

合について，unexpected messages queue 上の滞留に伴う遅延増加を抑制し，ハードウェア量の増大を抑制してノード数が大きな大規模システムにも適用可能な受信方式を確立することを目的としている．

筆者らは上記の目的のために，LHS(Limited-length Head Separation) を提案 [6] した．これにより短いメッセージの最短通信遅延が長くなるのではなく，大幅に短くなる効果を DIMMnet-2 の実機上での実験 [6][7] により示した．一方，これまでの研究では unexpected messages queue 検索時間短縮への LHS の効果や，ハードウェア量の評価は行われていなかった．

本報告では，第 2 章でメッセージ交換の高速化に向けた課題を列挙する．第 3 章でメッセージ交換サポートハードウェアである LHS を解説する．第 4 章では LHS を評価するために利用した DIMMnet-2 の概要とその LHS 実装前の問題点や，DIMMnet-2 上での LHS の実装について述べる．第 5 章では unexpected messages queue 検索時間短縮への LHS の効果やハードウェア量の評価について述べ，第 6 章でまとめる．

2. メッセージ交換の高速化に向けた課題

本章では通信遅延時間の短縮とスケーラビリティを両立できるメッセージ交換のハードウェア支援を考察するに当たり，既存の NIC やソフトウェア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．

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，受信キー (rank，tag，communicator) 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，受信キーを含むエンベロープを適切なマッチング処理主体 (ホスト CPU または NIC のオンチップ CPU または NIC のハードワイヤドロジック) に低遅延で伝達する必要がある．

2.2 送信側と受信側がかみ合わない場合の遅延の短縮

MPI では受信キーが一致する送信側の関数と受信側の関数の組同士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メッセージが届かないことが往々にして起きる．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である unexpected messages queue にバッファリングされる．

この時に，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や多数のメッセージの後にバッファリングされることにより，著しく大きな遅延が発生し，アプリケーション性能に甚大な影響が出ることがある．

同一の rank ペア間でのメッセージ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することが望ましい．

2.3 高性能とスケーラビリティの両立

従来，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そこ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が少ないかわりに，検索に時間がかかり，性能に問題があった．

一方，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，他の先行メッセージが前にバッファリングされることが少なくなるため検索性能は前者より良いが，このバッファを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．

また，前述の ALPU はセル数が足りている場合は検索性能は大変良いが，セルは比較器とレジスタを主体としたランダムロジックで構成されるためハードウェア量が大きい．このため，1000 ノードに対応する分量を実装するのは困難である．もし ALPU から滞留メッセージが溢れた場合には極端な性能低下が起きるため，スケーラビリティに問題があった．

3. 提 案 方 式

3.1 有限長メッセージ頭部分別 LHS

筆者等は 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 [6] した．**図 1** に LHS の基本コンセプトを図示する．

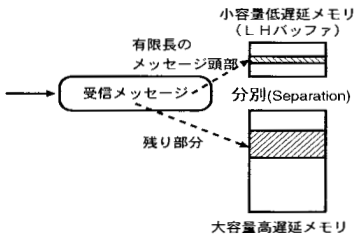

図 1 有限長メッセージ頭部分別 LHS の基本コンセプト

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，受信メッセージへのポインタ，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．

有限長メッセージ頭部分別 LHS の基本構造を**図 2** に示し，これを元にその動作を説明する．LHS は大容量高遅延メモリと小容量低遅延メモリを使い分ける．典型的にはこれらは IPUSH[11] を用いるなどして FIFO 状に制御されることが望ましい．この例では，両者は NIC 上にあり，NIC は LSI と大容量高遅延メモリから構成される．

ただし大容量高遅延メモリは NIC 上にあっても良いし，ホストの主記憶上のページング対象から外されたピンダウン領域に確保しても良い．小容量低遅延メモリは典型的には LSI に内蔵される高速 SRAM を想定しているが，LSI 外部に実装された高速 SRAM でもよいし，ホストの主記憶上のピンダウン領域に確保しても部分的な効果が得られる．

図 2 の例での LSI は，受信メッセージ分割制御部，低遅延メモリ制御部，小容量低遅延メモリ，高遅延メモリ制御部，ホストインタフェース部，ネットワーク制御部を備えている．フロー制御や再送制御などはネットワーク制御部が行い，受信メッセージ分割制御部には誤りのないデータが受け渡されるものとする．

受信メッセージ分割制御部は，例えば分割位置指定レジスタで設定する位置よりも前にあるメッセージ前半部を，低遅延メモリ制御部に依頼して小容量低遅延メモリに格納する．

メッセージのサイズはメッセージの前半部にあるものとし，

例えばサイズ位置指定レジスタで決定される所定の位置にあるメッセージサイズを受信メッセージ分割制御部は読み取る．ここで小容量低遅延メモリ上のエントリサイズより小さいメッセージ (図 2 中のメッセージ A2 および B2) の場合は全体が小容量低遅延メモリ上の固定長エントリに格納される．エントリサイズはメッセージの前半部を処理する主体（ホスト CPU または NIC 上の CPU）のキャッシュラインサイズと，扱うヘッダー部のサイズと，低遅延通信を促進したいペイロード部のサイズの兼ね合いから決定する．典型的には Pentium4 で MPI を高速化する場合 64 バイトである．

一方，エントリサイズより大きいメッセージ (図 2 中のメッセージ A1 および B1) の場合は，所定の位置よりも後にあるメッセージ後半部を高遅延メモリ制御部に依頼して大容量高遅延メモリに格納する．その際，メッセージ後半部を格納した大容量高遅延メモリにおけるアドレス情報（ポインタ）をメッセージ前半部とともに小容量低遅延メモリに格納しておく．

ホストはホストインタフェースを介して低遅延メモリ制御部，高遅延メモリ制御部を制御し，小容量低遅延メモリや大容量高遅延メモリにアクセスする．ホストインタフェースは PCI express 等の汎用 I/O でも，DIMMnet-2 のようなプリフェッチバッファを介したメモリバスインタフェースでも良い．

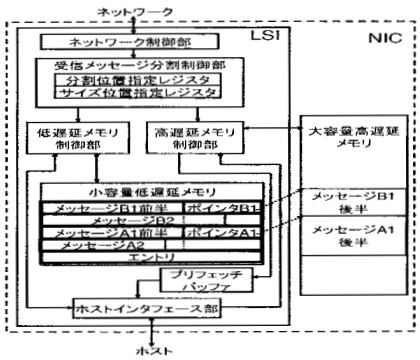

図 2　LHS の基本構造

なお，LHS と混同しやすい既存の方式として TCP Offload Engine(TOE)[8] や Intel I/O Acceleration Technology[9] でも採用されている Header splitting がある．しかし以下に説明するように LHS と Header splitting とは似て非なるものである．

Header splitting とはヘッダー (主に TCP ヘッダー) とそれ以外を切り分け，それらを並列パスで処理できるようにすることである．つまり高速化原理は [並列処理の促進] である．これに対して LHS は「低遅延メモリと高遅延メモリを使い分ける点」や「ヘッダーだけではなく所定の長さ以下のペイロードも含めて前半部として分離できる点」が Header splitting には無い新しい高速化原理を有する方式である．

3.2　LHS によるメッセージ交換の高速化原理

筆者等は LHS を MPI の unexpected receive queue 等のメッセージ交換用バッファに適用することを提案している．LHS によるメッセージ交換の高速化は以下の 3 つの原理から効果を期待することができる．なお，本報告の主題は 3 番目の高速化原理の評価であり，残り 2 つは別報告 [6][7] で評価済みである．

3.2.1　1 個のエンベロープ取得時間の短縮

MPI のメッセージにはエンベロープと呼ばれる制御情報部が含まれる．MPI の通信では常に，エンベロープに含まれる受信キーが一致するメッセージを受信時に選択する必要がある．このエンベロープは通常メッセージの先頭部に付加されている．よって前述の所定の位置を適切に設定することで，LHS が前半部として切り分けた部分にこのエンベロープが納まり，エンベロープは低遅延メモリ (LH バッファ) に誘導される．よって 1 個のエンベロープを取得する時間が短縮するため，最短通信遅延が短縮する．

なお，例えば QsNET-II のようにオンチップのキャッシュ(一種の低遅延メモリ) とオフチップの DRAM(一種の高遅延メモリ) を有する NIC もある．この構成で LHS を適用して，エンベロープを含む部分と含まない部分でキャッシング (低遅延メモリに書き込み) するか否かを切り替えるならば，容量が大きくなりがちなペイロード部によるキャッシュの汚染と性能劣化を回避できる．

3.2.2　短いペイロード部取得時間の短縮

LHS は Header splitting と同様の動作ができるとともに，メッセージ交換において最も遅延時間を短縮すべき短いメッセージのために，所定の長さに入る範囲でペイロード部を含むように前半部を切り分けることができる．こうすることによりエンベロープをエンベロープ解析主体（例えばホスト CPU）が取り込んだ時に，同一キャッシュライン上に短いペイロード部を取り込むことができる．

このような原理により，ペイロード部を取り込むためのメモリアクセス遅延が隠蔽できるので，短いメッセージの最短通信遅延がさらに短縮する．

3.2.3　複数のエンベロープ取得時間の短縮

LHS による送信側と受信側がかみ合わない場合の受信側における遅延の短縮の様子を**図 3** に示す．A2 を受信する関数の実行前にメッセージが B1,B2,A1,A2 という順で届くと，MPI では unexpected messages queue に到着順にバッファリングされる．これらのパケットが，従来は全て高遅延大容量メモリから読み出され，B1,B2,A1 と退避されていた．一方，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ため大幅に高速化する．

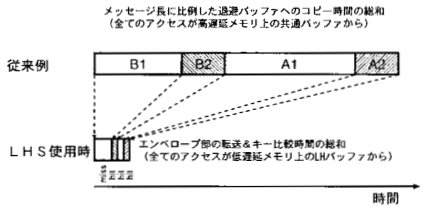

図 3　LHS による送信側と受信側がかみ合わない場合の遅延の短縮の様子

この際，LHS は複数のエンベロープが同一または連続したキャッシュライン上に到着順に整然と並んでいる状態を作り出す．ホストで用いられる近年の汎用 CPU には連続したキャッシュラインを自動的にプリフェッチするハードウェアプリフェチ機能や，命令から制御するソフトウェアプリフェッチの機能があり，LHS によってこれらが有効に機能するようになる．図 3 の例では，B1 を取得する 1 回目の LH バッファアクセスはミスヒットによるメモリアクセス遅延がかかるが，B2,A1,A2 と連続して複数のエンベロープを取り込みに行く際にはキャッシュにヒットする．

このようにして LHS はノード数が多いシステムで大量に unexpected messages queue にメッセージが滞留している状況下では，メッセージ 1 個あたりの検索に要する時間がキャッシュアクセス時間で決定され，大幅に高速化する．

4.　DIMMnet-2 とその上の LHS

本章では提案方式の評価を行うプラットフォームとして本報告で用いた DIMMnet-2 の概要と，MPI の高速化における DIMMnet-2 に固有の課題と，DIMMnet-2 上への LHS の実装について述べる．

4.1　改良前の 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．**図 4** に改良前の DIMMnet-2 の主要ロジックを示す．

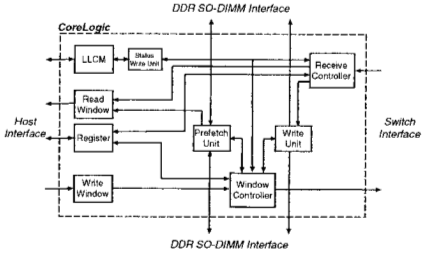

図 4 改良前の DIMMnet-2 の主要ロジック

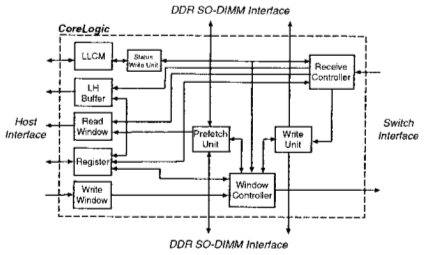

図 5 改良後の DIMMnet-2 の主要ロジック

DIMMnet-2 は FPGA ベースの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をしたモデルでもない．FPGA として XILINX 社製 Virtex-II Pro を 1 個と，256MB の SO-DIMM(Small Outline Dual In-line Memory Module) を 2 枚搭載しており，FPGA は現在 100MHz で動作している．

商用の NIC 等は ASIC を用いることでこの 2〜5 倍程度の周波数で動作させており，例えば QsNET-II の ASIC は 200MHz で動作しており，ALPU の論文[5] 上でのシミュレーション評価における周波数設定は 500MHz になっている．一方，本報告における評価は 100MHz で動作する FPGA ベースの DIMMnet-2 プロトタイプ実機上での性能評価を示している．よって，後述の評価の章における議論のように他 NIC との絶対性能比較をする場合は，FPGA ベースの DIMMnet-2 プロトタイプ上での性能より，ASIC ベースの実装における性能は高くできることを考慮する必要がある．

さらに DIMMnet-2 はベクトル型のプリフェッチ機能を有するメモリモジュールとして動作するとともに，Infiniband 4X の市販スイッチに接続することが可能な 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する．

SDR ベースのメモリスロットに装着される DIMMnet-1 とは異なり，DDR ベースの高速メモリスロットに対応する場合，BIOS 設定によるタイミング調整範囲を超えるため，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困難である．そこで DIMMnet-2 では 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．ベクトル型のデータ転送命令を利用した SO-DIMM 領域のアクセスは，通常の主記憶と同様のアクセスに比べるとアクセス遅延時間が長くなってしまう．この点は MPI のようなメッセージ交換モデルで利用する際には，例えば初期段階での MPI2 の実装[10] においては，SO-DIMM アクセス遅延時間は性能低下要因として問題になっていた．

4.2 DIMMnet-2 における LHS

DIMMnet-2 において LHS は前記の MPI 遅延時間の短縮と，ノード数の多い並列システムにも適用可能な unexpected messages queue の滞留に伴う遅延短縮の二つの目的を達成するためのハードウェア機構として位置づけられる．

PCI バス等の汎用 I/O 上に装着される NIC の場合は，LHS の大容量バッファは主記憶上のスワップ対象から除外登録された領域 (ピンダウン領域) に確保しても良い．一方，DIMMnet-2 ではネットワークコントローラ FPGA からアクセスしやすい位置にある大容量メモリである DIMMnet-2 ボード上の SO-DIMM 領域に確保する．一方，小容量高速バッファ(LH バッファ) は主にネットワークコントローラ FPGA 上のブロックメモリを用いており，大容量バッファより低遅延であるが容量は少ない．

図 4 に LHS によって改良後の DIMMnet-2 の主要ロジックを示す．ホストからアクセスしやすい位置に配置された小容量高速バッファである LH バッファのブロックが**図 5** と比較して追加されている．LH バッファはホストの仮想空間にマップされており，SO-DIMM 領域と異なりホストからベクトルコマンドを実行しなくても直接アクセスできる．

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 MPI 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_Irecv() 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(Programed IO) 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_Irecv() で指定された領域にコピー

このようにホストからは SO-DIMM 領域が直接見えず，大き目の遅延がある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 SO-DIMM 領域上のデータが出てこない構成になっている DIMMnet-2 の場合は，LHS により短いメッセージの MPI 受信遅延時間やバッファ検索時間が大幅に短縮されることが期待できる．

一方，LH バッファのキャッシュ属性は Writeback 等のキャッシュ可能領域にマップする．この場合，アドレスの再利用が行われる前にはキャッシュラインをフラッシュする必要があるが，LH バッファへのアクセスはほぼアドレス昇順連続アクセスになるためホストプロセッサのハードウェアプリフェッチ機能が有効に働き，キャッシュのヒット率が高くなる．このため，大量の先着メッセージが滞留した LH バッファをホストが検索しに行く場合，滞留メッセージのエンベロープあたりのエントリアクセス時間は，ホストの主記憶へのアクセス遅延ではなく，キャッシュへのアクセス遅延によって決定される．よって，DIMMnet-2 における LHS による検索の加速率は，マップ領域と SO-DIMM 領域への単発的なアクセス遅延 (100ns 程度: 数百 ns 程度) の比率よりも高くなることが予想される．

5. 性能評価

本章では，LHS に関して DIMMnet-2 プロトタイプの実機上に実装を行い，実験評価を行うことにより，LHS の unexpected messages queue 検索時間短縮への効果とハードウェア量について考察する．

5.1 Unexpected メッセージ検索時間短縮化効果

LHS によれば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，unexpected messages queue からのエンベロープの読み出しはアクセス遅延が短い LH バッファから読み出される等の高速化要因がある．このため，同 queue からのメッセージ検索時間を短縮化できると考えられる．本節では，その効果の評価について述べる．

5.1.1 評価環境

本節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)

5.1.2 測定方法

LHS を用い，エンベロープの格納場所が LH バッファとなる場合は，あらかじめ LH バッファの X 番目のエンベロープが格納される位置に特定のデータを書き込んでおき，それ以外の領域には 0 を書き込んだ状態にする．特定のデータをバッファの頭から順番に読み出し比較する，という処理を繰り返して，目的のデータが見つかるまでの時間を測定した．比較は，目的のデータ以外は rank の比較のみを行い (この時点で違うエンベロープと判断し，次のエンベロープを読み出す)，目的のデータの場合のみ，rank, tag, communicator の比較を行っている．

エンベロープの格納場所が SO-DIMM の場合は，16Byte のエンベロープ (目的のデータ以外は 0) を必要回数書き込み，頭から順番に 16Byte のベクトル読み出し (VL) 実行と比較を X 回繰り返している．読みだし部分以外の処理はエンベロープの格納場所が LH バッファの場合と同じである．

5.1.3 結　　果

評価結果を**図 6** に示す．横軸は unexpected messages queue に溜まったメッセージの個数に相当し，unexpected messages queue の先頭から MPI_Irecv( ) 関数の中で指定されたエンベロープを有するメッセージまでの深さである．1 メッセージあたりの検索時間は，LHS を用いない場合は SO-DIMM へのベクトルロードアクセス時間に若干の計算時間を加えた時間がかかり，その時間は平均 634ns であった．一方，LHS を使った場合は平均 14.8ns かかる．LHS を使った場合と使わない場合のグラフの傾きの差が示すように，LHS を用いることにより unexpected messages queue の検索時間は約 43 倍加速された．

また，絶対性能の比較を行うため 1 メッセージあたり 0.1μ 秒という文献 [3] 上での QsNET-II の Tport を用いた場合の unexpected messages queue 検索速度を元に QsNET-II の値を図 6 に併記した．DIMMnet-2 は FPGA 実装のため半分の周波数であり測定環境のホスト CPU も遅めであるにも関わらず，QsNET-II より unexpected messages queue 検索速度が 6.8 倍高速であった．

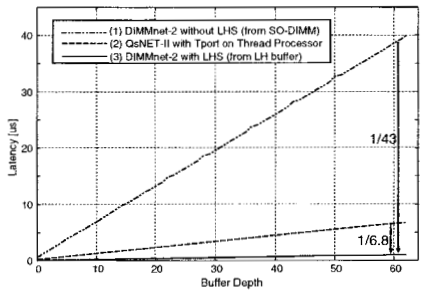

図 6　LHS による unexpected messages queue 検索時間への効果

5.1.4 考　　察

DIMMnet-2 においてはベクトルロードコマンドを用いた SO-DIMM アクセス遅延時間が大きいことが一因として効いており，LHS を用いることにより長い unexpected messages queue の検索時間が著しく高速化した．一方，DIMMnet-2 のベクトルロードコマンドを用いたメモリアクセス遅延時間は 600ns 程度であり，主記憶アクセス遅延は 100ns 程度であることが LHS を用いる場合のバッファの深さが 0 の時の測定値からわかるので，DIMMnet-2 の特殊なメモリ構成に起因する加速率は高々6 倍程度であり 43 倍という加速率はそれだけでは説明できないことは明らかである．

残りの 7 倍程度の加速率向上は DIMMnet-2 の特殊なメモリ構成に起因しない別の要因で達成されているはずである．具体的には，キャッシュがヒットしていないとメッセージあたり主記憶アクセス遅延 100ns 程度がかかってしまうはずであり，平均 14.8ns という計測結果はキャッシュがヒットしていることを示していると考えられる．DIMMnet-2 の今回の実験におけるエントリサイズは 64 バイト，エンベロープサイズは 16 バイトである．Pentium4 のキャッシュラインサイズは 64 バイト，隣接ラインのプリフェッチ設定の効果も入れると実質的に 128 バイトのキャッシュラインサイズである．これらのサイズを考慮すると，最初のエントリをアクセスした際に次のエンベロープがキャッシュ上に取り込まれ，次回はキャッシュがヒットをすることでハードウェアプリフェッチが起動し，後続のキャッシュラインも次々とプリフェッチされてくる．このように，LHS によってキャッシュラインサイズ程度のオフセットでアドレス昇順にエンベロープ並ぶために，ホスト CPU のハードウェアプリフェッチが有効に機能するという原理で，残りの 7 倍の加速が達成されているものと考えられる．

上記のように LHS により unexpected messages queue の検索性能と大容量メモリのアクセス遅延の分離ができ，unexpected messages queue の検索性能は主にホスト CPU のキャッシュアクセス性能，すなわちメモリ周辺の周波数に依存する．

DIMMnet-2 の後継機である DIMMnet-3 はメモリバスの高周波化とメモリの大容量化に伴い，ホストインタフェース部と大容量メモリ部が別基板となる．その結果，ホストから大容量メモリのアクセス遅延は延びる傾向にあるが，検索時間はホスト CPU のメモリ周辺の周波数が向上することから，本実験環境より一層の高速化が達成できると考えられる．

次に，PCI バス等の汎用 I/O に装着されホストの主記憶に DMA を行うことができる NIC 上に LHS を実装し，LH バッファを主記憶上のキャッシュ可能属性を有するピンダウン領域に確保し，検索を DIMMnet-2 同様にホスト CPU が行う場合を考える．この場合は，前述のキャッシュが効きやすくなる効果が同様に期待できるので，7 倍程度の加速率が期待できる．

一方，PCI-X バスに装着される NIC の QsNET-II[4] は，MPI のキュー操作を NIC 上の CPU(Thread Processor) が実行する．受信メッセージはキャッシュを介して DRAM 上の queue に受信される．ここで，QsNET-II の unexpected messages queue は Thread Processor が管理するリストである．1 メッセージあたり 100ns という検索時間は 1 回の DRAM アクセス遅延と大体一致していることから，それは毎回 DRAM アクセス遅延がかかるようなキャッシュが効き難いデータ構造になっており，Thread Processor が高性能化しても検索性能が変化しないと考えられる．事実，旧世代の QsNET から QsNET-II への進化においてオンチップ CPU の性能は大幅に向上しているが，検索性能はほとんど変化がない [3]．

これに対して DRAM アクセス遅延を検索性能から分離した DIMMnet-2 は QsNET-II と比較して 6.8 倍の高速である．この結果はオンボード DRAM 上の queue 検索をオンチップ CPU が行う QsNET 型の NIC においても，プリフェッチ機能と LHS を併用すれば DIMMnet-2 固有でない部分の効果で 7 倍程度の加速率が得られる可能性があることを意味する．

5.2 ハードウェア量

LH バッファは ALPU[5] のような論理回路ではなく単純なメモリであるため，ノード数の大きな大規模クラスタシステムにおける長い unexpected messages queue に ALPU よりも簡単に対応することができると考えられる．この点を明らかにすべく本節では LHS のハードウェア量の定量的な評価を行う．

5.2.1 測定方法

LHS のハードウェア量を評価するために下記に示す 4 つの構成について，論理合成を行った．論理合成に用いたツールは ISE 8.2i SP3 で，Synthesize のオプションで面積よりも速度を優先して，合成している．対象デバイスは Virtex-II Pro XC2VP70-7FF1517 として論理合成した．LHS のエントリは全て 64 バイトとした．

(1)　エントリ数 64 個の LH バッファ(4KB) を 1 個持つもの
(2)　(1) のエントリ数を 2 倍 (128 個=8KB) に変更したもの
(3)　(1) のエントリ数を 4 倍 (256 個=16KB) に変更したもの
(4)　(1) のエントリ数を 8 倍 (512 個=32KB) に変更したもの

5.2.2 結　　果

表 1 に LHS と ALPU のハードウェア量の比較を示す．ALPU の値は文献 [5] から引用した．どちらも FPGA として Virtex-II Pro を用いているので LUT の数，フリップフロップ (FF) の数は特に補正しなくてもそのまま平等に比較できる．ALPU は

表 1 LHS と ALPU のハードウェア量の比較

対応メッセージ数	ALPU のロジック部 Slice 数 (使用率)	LHS のロジック部 Slice 数 (使用率)	LHS のメモリ部 BRAM 数 (使用率)
64	不明	116(0.4%)	4/328(1%)
128	5,215(16%)	129(0.4%)	4/328(1%)
256	10,350(31%)	137(0.4%)	8/328(2%)
512	不明	148(0.4%)	16/328(4%)

ブロックメモリは用いないが LHS はブロックメモリを用いる．ALPU はセルが MPI のエンベロープ 1 個に対応し，LHS はエントリが MPI のエンベロープ 1 個とデータ部 56 バイト分に対応する．よってこれらの個数がそれぞれのオンチップでエンベロープを保持可能なメッセージ数となる．

表 1 において LHS のメモリ部のブロック RAM 数が構成 (1) と (2) の場合で同一になっているが，Virtex-II Pro の場合，ブロック RAM のビット幅と深さに制約があるためこのような結果となっている．

表 1 から明らかなように，LHS に必要なロジック部は 128 エントリの場合で ALPU の約 1/40，256 エントリの場合で ALPU の約 1/76 にすぎず，エントリ数を増やしても ALPU のように大幅に増加することはない．ただし記憶部分として ALPU のようにフリップフロップを用いるのではなくブロックメモリを用いている．このためブロックメモリを LHS はエントリ数に比例して消費するが，ALPU と同等の規模では FPGA 内のブロックメモリの 1〜2%しか消費しておらず，LHS によるブロックメモリの消費量はクリティカルではないと考える．

5.2.3 考　　察

Underwood らの論文における ALPU ではエンベロープの比較部 42bit，ポインタ 16bit の合計 58bit をセル上に記憶できるようにしているだけなので，前述のような LHS における 56 バイトまでのデータを有するメッセージ受信を加速する効果は無い．これに対して上記のハード量の見積もりは LHS のエントリは全て 64 バイトとしているので，ALPU と同等のエンベロープの検索加速に対応できる使用メモリ量は表 1 中の値の約 1/8 程度に相当する．

使用する FPGA のサイズを小さくしてコストを抑制したり，対応ノード数を多くしたい場合は，エントリを構成しているハードウェア量を小さくすることに主眼をおき，LHS のエントリを構成しているメモリ部分を節約するような実装を行うことが可能である．

一方，ALPU は既に 256 エントリ程度が限界であり，それを超えるノード数のシステム上で NPB の FT や IS などを実行すれば，ノード数に近い多くのメッセージが unexpected messages queue に滞留するため，ソフト的な例外処理に伴って顕著な性能低下が発生すると考えられる．

6. ま と め

本論文では MPI 等のメッセージ交換の高速化支援機能である有限長メッセージ頭部分別 (LHS) について提案し，DDR DIMM スロットに装着される 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．

性能評価としては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 の効果の評価を行った．LHS を使う場合，使わない場合と比較して unexpected messages queue の中のメッセージ検索において 43 倍の高速化がなされることを確認できた．内訳は DIMMnet-2 固有のメモリ構成に起因する効果が約 6 倍で，残りの約 7 倍がホストのキャッシュがヒットしやすくなった効果である．絶対性能は低遅延な MPI のオフローディングで有名な商用 NIC である QsNET-II の 6.8 倍に当たり，後者の効果による加速率と概ね一致する．

一方，ハードウェア量は上記の検索のアクセラレータである ALPU に比べて LHS に必要なロジック部は 128 エントリの場合で ALPU の約 1/40，256 エントリの場合で ALPU の約 1/76 にすぎない．LHS の場合は代わりにブロックメモリを消費するがその分量も FPGA 内蔵のメモリの数%で収まる程度に少なく，LHS はより大規模な並列システムに適していることも判った．

今後は，後継機である DIMMnet-3 の実機の開発を進め，本論文では報告できていない通信機構や MPI の実装・評価を進め，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きた慶應義塾大学の西准教授，渡辺氏，大塚氏，伊沢氏，東京農工大学の並木教授，羅氏，池田氏，太田氏，浜田氏，荒木氏，木立氏，森氏，金井氏，立命館大学の国枝教授，表氏，森山氏，高柳氏，種田氏，藤岡氏，名古屋工業大学の齋藤准教授，和歌山大学の中平氏，笠松氏，京都大学の上原准教授，日立 JTE 社の上嶋氏，今城氏，岩田氏に感謝いたします．

参 考 文 献

1) J. Liu, A. Mamidala, A. Vishnu, D. K. Panda : "Evaluating Infiniband Performance with PCI Express", IEEE MICRO, Vol.25, No.1, pp.20-29 (Jan. 2005)
2) R. Brightwell and K. D. Underwood : "An Analysis of NIC Resource Usage for Offloading MPI", 18th International Parallel and Distributed Processing Symposium (IPDPS'04) (2004)
3) K. D. Underwood and R. Brightwell : "The Impact of MPI Queue Usage on Message Latency", International Conference on Parallel Processing (ICPP'04) pp.152-160 (2004)
4) J. Beecroft, D. Addison, D.Hewson, M. McLaren, D. Roweth, F. Petrini and J. Nieplocha "QsNET II : Defining High Performance Network Design", IEEE MICRO, Vol.25, No.4, pp.34-47 (Jul. 2005)
5) K. D. Underwood, K. S. Hemmert, A. Rodrigues, R. Murphy and R. Brightwell : "A Hardware Acceleration Unit for MPI Queue Processing", 19th International Parallel and Distributed Processing Symposium (IPDPS'05) (2005)
6) 田邊, 北村, 宮部, 宮代, 天野, 羅, 中條 : "DIMMnet-3 ネットワークインタフェースにおける MPI 支援機能", 情報処理学会計算機アーキテクチャ研究会, 2006-ARC-169, pp.103-108 (Aug. 2006)
7) 田邊, 北村, 宮部, 宮代, 天野, 羅, 中條 : "主記憶以外に大容量メモリを有するメモリ/ネットワークアーキテクチャ", 情報処理学会計算機アーキテクチャ研究会, 2007-ARC-172, pp.157-162 (Mar. 2007)
8) Chelsio Communications : "Time for TOE - The Benefits of 10 Gbps TCP Offload", A Chelsio Communications White Paper http://www.chelsio.com/solutions/pdf/Chelsio_TOE_Value_Prop.pdf
9) K. Lauritzen, T. Sawicki, T. Stachura and C. E. Wilson : "Intel(R) I/O Acceleration Technology Improves Network Performance, Reliability and Efficiently", Technology@Intel Magazine (Mar. 2005)
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) 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", 情報処理学会論文誌コンピューティングシステム, Vol.47, No.SIG12(ACS15), pp.59-73 (Sep. 2006)