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

2008－ARC－177 (31)
2008－HPC－114 (31)
2008／3／6

ハードウェアを用いたメッセージ交換システムのスケーラビリティ改善

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

NPB FT のようにノード数の数十%にあたるメッセージが平均してメッセージバッファに溜まるアプリケーションが存在する．MPI のオフロードにより低遅延通信を実現するネットワークインタフェースを用いた大規模並列システムにおいては，MPI のメッセージバッファ検索時間が通信時間を劇的に増加させ，この種のアプリケーションのスケーラビリティを制約する．これまで筆者らは通信遅延やメッセージバッファ検索遅延を短縮する通信機構である LHS(Limited-length Head Separation) を提案し、メッセージバッファにメッセージが全く無い場合や少数しか溜まっていない場合の評価を DIMMnet-2 の上で行ない、部分的な有効性を示してきた。本論文ではメッセージバッファにメッセージが千個溜まるような数千ノード以上の大規模並列計算機への適用を想定し、LHS によるメッセージバッファ検索性能とハードウェア量のスケーラビリティを評価する．さらに，そのスケーラビリティ性能向上に向けたホストインタフェースの種類や周波数変更や各種プリフェッチ手法の効果を DIMMnet-3 ベースの環境でも評価する．

Improvement of Scalability of Message Passing System with Hardware Acceleration

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

Such as NPB FT, there exists an application in which messages of several tens of percentages of the number of nodes in average are accumlated in a message buffer. By offloading of MPI, in a massively parallel system with network interface which realizes low latency communication, retlieving a message buffer of MPI brings significant growth of communication time, thus it restricts scalability of such kind of applications. So far we have proposed a communication mechanism, LHS (Limited-length Head Separation) which reduces communication latency or latency of retlieving a message buffer. Effects of the mechanism has been shown partially by evaluating the cases in which there is no message in a message buffer or which a small number of messages are kept in it using DIMMnet-2. In this paper, supposing application for a massively parallel system with over thousands nodes which can hold thousands of messages, we have evaluated performance of retlieving a message buffer and scalability of an amount of hardware. Moreover, we have evaluated types, changes of frequency of host interface and effects of some kinds of prefetching for higher performance of scalability under environment of DIMMnet-3.

1. はじめに

近年では実行性能 1PFLOPS 以上，ピーク性能 10PFLOPS の次世代スーパーコンピュータ [1] 等，数万ノード以上の規模が不可避と考えられる実システムの開発競争が日米を中心に繰り広げられている．このような規模の超並列マシン上ではスケーラビリティが性能上最も重要な設計項目である．

超並列マシンのアプリケーションの大半はメッセージ交換の API(Application Program Interface) のデファクトスタンダードである MPI(Message Passing Interface) を用いて書かれている．この膨大なアプリケーション資産を極力そのままノード数が多いマシン上で高速化したいというニーズは大きい．

MPI は並列マシン間のポータビリティ確保のために広く用いられているため，様々な並列環境で動作する．このため HPF や GA(Global Array) など並列プログラムの生産性向上を目指したいくつかの API の処理系は MPI へのトランスレータとして実装されている．そのような API のユーザは MPI でプログラムを書いていないため MPI レベルでの問題発生に対して身に覚えがない．例えば GA で記述された NWChem が細粒度メッセージを大量に生成するスケーラビリティに乏しい MPI プログラムに変換されるという問題が起きる．このような場合，アプリケーションプログラマによる最適化が困難である．このように，システム提供側からの MPI のスケーラビリティ向上のサポートは High Productivity Computing のトレンドからも重要性が増している．

NPB FT のようにノード数の数十%にあたるメッセージが平均して MPI のメッセージバッファに溜まるアプリケーションが存在する [2]．このようなスケーラビリティ問題を起こしがちな All to All 型メッセージ通信のバンド幅は，結合網の二分割 (Bi-section) バンド幅で決まるとされている．しかし，ノード数が 1000 を越えるような超並列マシン上では実効的な二分割バンド幅は必ずしも結合網断面を横切る通信リンクのバンド幅の総和で決まるのではない．例えば QsNET-II[3] のような低遅延 NIC(Network Interface Card) 上では，通信 1 回当たり数十〜百 μ 秒にも達するメッセージバッファ検索時間 [4] で決定されるようになる．よって，メッセージバッファ検索のスケーラビリティが高い高速化が望まれる．

このような状況を背景に，Cray XT 系の超並列マシンを Cray 社と共同開発してきた米国 Sandia 国立研究所の研究 [5] で，ALPU と呼ばれるハードウェアによる方式が提案された．しかし，ハードウェアで対応可能な滞留メッセージ数がハードウェア量の問題で 256 程度に制約されるため，スケーラビリティに問題がある．

筆者ら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合について，メッセージバッファ上の滞留に伴うメッセージ検索時間増加を抑制し，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研究を進めている．

† (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

上記の目的の達成のため，筆者らは LHS(Limited-length Head Separation) を提案し、滞留しない時の通信遅延の短縮[7]や，滞留が少ない時のメッセージバッファ検索の高速化[8][9]の評価を DIMMnet-2 プロトタイプ[6] 上で行なってきた．

本論文は大量にメッセージが滞留した時のメッセージバッファ検索の LHS による高速化と，ハードウェア量の増大の LHS による抑制の二点に絞って評価を行う．さらにホストインタフェースを DDR から DDR2 にグレードアップさせた DIMMnet-3 を想定した環境においても、ホストインタフェースの種類や周波数やプリフェッチ手法が検索性能に与える影響に関して評価を行う．

本論文では，第 2 章でメッセージバッファ検索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 章ではメッセージバッファ検索時間短縮への LHS の効果やハードウェア量の評価について述べ，第 5 章でまとめる．

2. 解決すべき課題

本章ではメッセージバッファ検索の高速化を考察するに当たり，既存の手法において残されている課題について述べる．

2.1 メッセージバッファ検索時間の短縮

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．

QsNET-II[3] ではメッセージバッファ検索などの MPI の処理を NIC 上のファームウェアにオフロードしている．メッセージバッファ検索においては 1 個の滞留メッセージあたり 100ns のペナルティが生じる．この値は滞留メッセージ数が少ない局面では低遅延通信を実現する良好な値である．一方，Brightwell 等の研究[2] に報告されているように，NAS Parallel Benchmarks(NPB) の FT ではノード数の数十%にあたるメッセージが平均して unexpected messages queue に滞留する．このような，大規模並列システム上で滞留メッセージが多数 (例えば 1000 個) 発生する局面では，以下の検索をオフロードしない方式より，QsNET-II では遅延時間が大幅に劣化するという問題がある[4]．

QsNET から QsNET-II には NIC 上の CPU が高性能化しているにもかかわらず，滞留メッセージあたり 100ns のペナルティという検索性能は向上していない[4]．このことは CPU 性能ではなく，検索の際に参照されるデータのアクセス時間が問題となっている可能性がある．

2.2 短いメッセージの遅延の短縮

Underwood らの研究[4] 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，滞留メッセージ数に比例して増加する検索遅延増加が抑制されている．しかし，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 QsNET-II の 2μ 秒程度と比べ，この方式では 6μ 秒へと損なわれてしまっている．

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の増加が全体の性能やスケーラビリティに問題が生じる．

このように，従来から存在する単純に NIC へのオフロードを行わない方式は，他の重要な基本性能の劣化を伴うため，トータルとしてはあまり望ましくない．

2.3 スケーラビリティの確保

Underwood らのその後の研究[5] によると，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，最短通信遅延も良好である．

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，大幅な性能低下の発生が報告されている．

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．

ピーク性能 10PFLOPS の次世代スーパーコンピュータ[1] 等，数万ノード以上の規模が不可避と考えられるシステムも開発が始まっており，高度なスケーラビリティの確保が望まれる．このようなシステム上で例えば NPB FT と同様な処理を行なうアプリケーションを実行させる場合は，利用ノード数の増加に応じてメッセージ滞留が増加することが予想され，極端な性能低下を防止・軽減する方法が望まれる．

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 等のメッセージ交換システムソフトウェアにおける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[7] した．**図 1** に LHS の基本コンセプトを図示する．

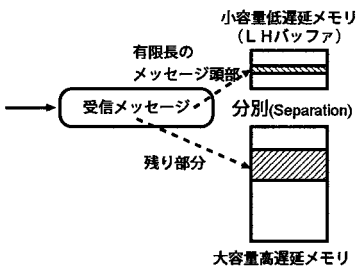

図 1 有限長メッセージ頭部分別 LHS の基本コンセプト

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，FIFO 状制御付きの遠隔間接書き込みを高速化するハードウェアである IPUSH[10] 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のトランスポート層以下の処理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 のようなメモリバスインタフェースでも良い．

3.2 LHS によるメッセージ交換の高速化原理

筆者らは 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する．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．

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．

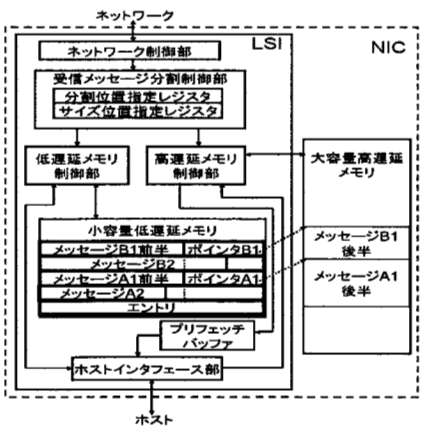

図 2 LHS の基本構造

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固定長の情報である．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．

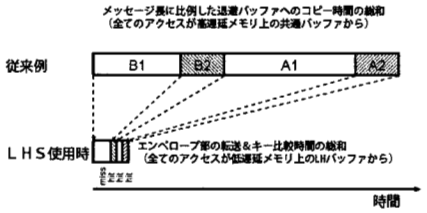

図 3 LHS による送信側と受信側がかみ合わない場合の遅延の短縮の様子

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. 性能評価

本章では，DIMMnet-2 ベースと DIMMnet-3 ベースの 2 つの環境上で実験評価を行うことにより，LHS の unexpected messages queue 検索時間への効果とハードウェア量のスケー

ラビリティについて考察する.

4.1 Unexpected メッセージ検索時間のスケーラビリティ

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．従来の研究では DIMMnet-2 ベースの環境で 64 メッセージまでの滞留に対する短縮化効果を示すことができたが，それをもってスケーラブルであると結論付けるには無理があった．本節では，1000 メッセージが滞留するような状況下でのその効果の評価について述べる．そのスケーラビリティ向上に向けて，DIMMnet-3 ベースの環境でホストインタフェース種類の変更や，周波数の変化や，プリフェッチ方式による効果の変化についても評価を行なう．

4.1.1 評価環境

本節の実験において用いられた評価環境を表 1 に示す．DIMMnet-3 ベースの環境については現状の DIMMnet-3 のプロトタイプで使われている FPGA の動作周波数 (200MHz) 以上の周波数特性を得るために，DIMMnet-3 そのものはアクセスさせずに主記憶上に確保した領域を LH バッファとしてアクセスさせて測定を行なった．

4.1.2 測定方法

DIMMnet-2 上または主記憶上に構成した 1024 エントリ構成の LH バッファを用い，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．

DIMMnet-3 想定環境における評価においては DIMM の設定を PC-3200(200MHz) および PC5300(333MHz) として測定した．

LH バッファへのアクセス性能の高速化のために，ホスト CPU からの各種プリフェッチ方法における性能の評価を行った．今回の評価で用いるプリフェッチ機能は 2 つある．一つはハードウェアプリフェッチ，もう一つはソフトウェアプリフェッチである．今回の計測では，以下の表 2 に示す 6 つのプリフェッチの組み合わせで計測を行った．

4.1.3 結　　果

各種 NIC における unexpected messages queue の検索時間を図 4 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．上記の測定方法で記載された方法で得られた 1 メッセージあたりの検索時間は，DIMMnet-2 において LHS を使った場合は平均 44.4 ns，DIMMnet-3 において LHS を使った場合は平均 29.0ns かかった．

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 [4] 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 4 に併記した．ALPU の値は文献 [5] のシミュレーションの結果のグラフから引用した．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 2.3 倍高速であった．ALPU のシミュレーションは周波数 500MHz と DIMMnet-2 の 5 倍の周波数における値であるので、ALPU の速度はかなり高めに見積もられているはずであるが，深さ 1000 の時で 1.5 倍の速度に過ぎない。DIMMnet-3 の実機は ALPU のシミュレーション上の周波数の 2/5 に過ぎないが，深さ 1000 の時でほぼ同等の速度が得られる．利用したマザーボードで利用可能な DIMM の最高周波数である 333MHz で動作する場合を想定した測定環境上では若干性能が改善し，500MHz 動作を想定した ALPU のセル数が 128 の時に深さ 600 程度，セル数が 256 の時に深さ 800 程度で DIMMnet-3 は ALPU より時間がかからなくなる．つまり，LHS は ALPU より低い周波数でも性能面でスケーラビリティが高い．

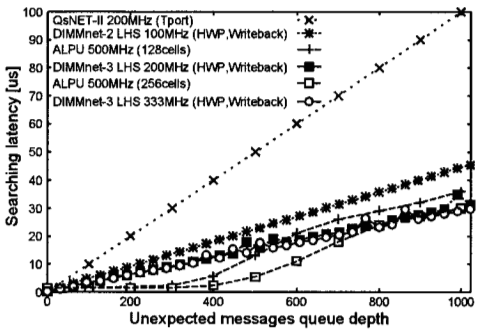

図 4　各 NIC における unexpected messages queue 検索時間

ホスト CPU からの各種プリフェッチ方法における LHS による unexpected messages queue 検索時間の測定結果のうち，DIMMnet-2 上での結果を図 5 に，DIMMnet-3 想定環境 (200MHz) での結果を図 6 に示す．

図 5 に示されるように，DIMMnet-2 においてはプリフェッチ手法によって性能に差が現れる。これは LHS を用いてもメモリバンド幅に余裕が無いためと考えられる．

図 6 に示されるように，200MHz の DIMMnet-3 想定環境においては，キャッシュを有効にするかしないかでは大きな差が見られるものの，ほぼ常にキャッシュがヒットする最高の状態を CLFLUSH 命令をコメントアウトして強制的に毎回ヒットさせた場合のグラフと，キャッシュを有効にして動作させている場合のグラフはほぼ重なっている．つまり，LHS の導入によって，ほぼ常にキャッシュがヒットする状態ができあがっていることがわかる．図には表示していないが，このような状態ではハードウェア任せではなく，あえて命令を追加しているソフトウェアプリフェッチを行なうと若干性能が低下していた．

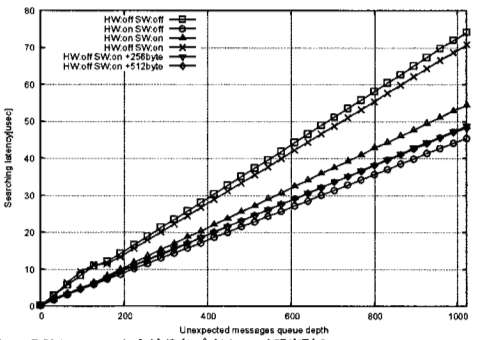

図 5　DIMMnet-2 における各プリフェッチ設定別の unexpected messages queue 検索時間

表 1　評価環境

環境	DIMMnet-2	DIMMnet-3 装着想定
CPU	Pentium4 2.6GHz	PentiumD 840 3.2GHz (シングルコア動作)
L2 cache	512Kbyte	2Mbyte
Chipset	VIA 8751A	Intel 955X
Memory	PC-1600 DDR SDRAM 512MB x1	PC-5300 DDR2 SDRAM 512MB x2 DIMM の設定を PC-3200(200MHz)、PC5300(333MHz) として測定
OS	RedHat8.0 (Kernel 2.4.27)	openSUSE (Kernel 2.6.22.5)
Compiler	GCC 3.3.5	GCC 4.2.1
Compile option	-Wall	-Wall

表 2　プリフェッチ方法の組み合わせ

ハードウェア	ソフトウェア	備考
off	off	両方ともプリフェッチが無効
on	off	
on	on	
off	on	エンベロープ比較を行なう文の直前に 128 バイトプリフェッチインライン関数を挿入
off	on +256byte	プリフェッチ先を探索中のアドレスから 256 バイト先に設定
off	on +512byte	プリフェッチ先を探索中のアドレスから 512 バイト先に設定

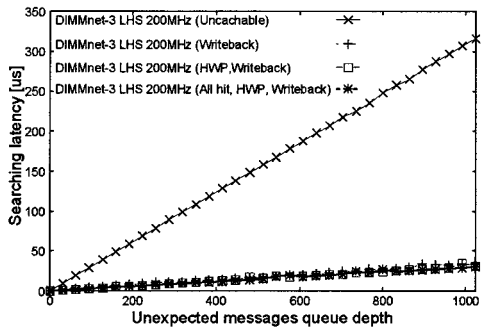

図 6　DIMMnet-3 想定環境における各プリフェッチ設定別の unexpected messages queue 検索時間

4.1.4 考　　察

先行研究 [9] における考察の中で DIMMnet-3 における検索時間はホスト CPU のメモリ周辺の周波数が向上することから，DIMMnet-2 より一層の高速化が達成できることが予想されていた．本実験ではそのことが測定によって実証されたと言える．

100MHz を DDR2 化して 200MHz にした場合の効果は大きかった．しかし，200MHz で動作する DIMMnet-3 想定環境において LHS を用いた場合は，メモリネックではなく CPU 処理ネックになっているので，これ以上の性能向上はメモリの周波数を上げてもあまり達成できず，最適化が強力と言われる Intel コンパイラの使用やアセンブリ言語でのプログラミングなど，エンベロープ比較処理の最適化やホスト CPU の強化による性能向上の可能性の追求の方が有益であると考えられる．

一方，LHS と ALPU では性能のスケーラビリティに関して，測定範囲では遅延の絶対値が 30 μ 秒程度と大きい上に両者はあまり大きな差が出ているとは言えず，さらなる改良が必要である．ここで，ALPU では 1 本の比較機能の付いたハードウェアによるキューにメッセージを格納していくので先着メッセージを全部探索しなければならない．これに対し LHS の場合は，先着メッセージを複数の LH バッファに振り分けて，LH バッファ1 個あたりに滞留する深さを減少させ，同じ LH バッファに割り振られた一部の先着メッセージだけ探索するように実装することが可能である．よって，実際のアプリケーション実行時に期待できる平均的な探索時間は，先着メッセージが 1000 個あるような状況では例えば 8〜16 個程度の LH バッファに振り分けられる実装の LHS によれば，ALPU より 1 桁程度の検索時間短縮が可能であると考えられる．

4.2 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．

4.2.1 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) に変更したもの
(3)　(1) のエントリ数を 4 倍 (256 個) に変更したもの
(4)　(1) のエントリ数を 8 倍 (512 個) に変更したもの
(5)　(1) のエントリ数を 16 倍 (1024 個) に変更したもの

4.2.2 結　　果

表 3 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 3 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 3 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

表 3 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%)
1024	不明	152(0.4%)	32/328(8%)

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．

4.2.3 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 3 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．

5. 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) の Unexpected Message Queue に 1024 個までのメッセージが滞留した場合の効果の評価を行った．

性能のスケーラビリティに関しては，LHS を用いた DIMMnet-3 は同じ周波数で動作する QsNET-II より常に 3.4 倍の検索速度を実現できる．メッセージバッファ検索のアクセラレータである ALPU において現実的なハードウェア量の投入を行い DIMMnet-3 の 1.5 倍の周波数で動作させた場合と比較しても，1000 のメッセージが滞留する大規模システムでは DIMMnet-3 は ALPU より高い性能が得られた．このように LHS は性能上も ALPU より高いスケーラビリティがあることがわかった．ただし DDR2 化や周波数向上による LHS の検索性能向上には限度があり，メモリバンド幅ネックが解消されるレベルに達すると鈍ることもわかった．

LHS を用いて MPI のエンベロープをホスト CPU に転送する場合のプリフェッチ方式については，DIMMnet-2 および DIMMnet-3 の両方の環境で，ハードウェアプリフェッチのみを用いることが性能を最も引き出せることを確認した．

一方，ハードウェア量は ALPU は 256 エントリ程度までしか FPGA 上に実装できないのに対して，LHS は DIMMnet-2 の FPGA 上にも 1024 個の滞留メッセージを保持できるものをロジック部は全体の 0.4%，メモリ部は 8%の消費だけで無理なく実装することができた．このように LHS はハードウェア量の点で，ALPU より大幅に高いスケーラビリティがあることがわかった．

このように LHS は性能とハードウェア量の両面から ALPU より高いスケーラビリティを有するものの，1000 のメッセージが滞留する状態での通信遅延は 30 μ 秒に近いため，アプリケーションのスケーラビリティ確保の観点からはあと一桁程度の性能向上がなされることが望ましい．今後は，複数の LH バッファに適切に滞留メッセージを振り分けることにより，1 回の受信の際における LH バッファへの平均アクセス回数を大幅に減らすことにより，アプリケーションのスケーラビリティ確保に向けた更なる性能向上を行なう予定である．

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，浜田氏，荒木氏，木立氏，森氏，金井氏，金氏，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) 渡辺:“プロジェクト進捗報告”, 次世代スーパーコンピューティングシンポジウム 2007〜ペタスケール・システムの利用に向けて〜, p.5-11 (2007-10).
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) 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)
4) 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)
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) 北村, 濱田, 宮部, 伊澤, 宮代, 田邊, 中條, 天野 : “DIMMnet-2 ネットワークインタフェースコントローラの設計と実装”, 情報処理学会論文誌コンピューティングシステム, Vol.46, No.SIG12(ACS11), pp.13-26 (Aug. 2005)
7) 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)
8) 田邊, 北村, 宮部, 宮代, 天野, 中條 : “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”, 情報処理学会計算機アーキテクチャ研究会 (SWoPP'07), 2007-ARC-174, pp.139-144 (Aug. 2007)
9) 田邊, 北村, 宮部, 宮代, 天野, 中條 : “メッセージ頭部の格納場所切替によるメッセージバッファ検索の高速化”, コンピュータシステムシンポジウム (ComSys'07), pp.125-132 (Nov. 2007)
10) 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)