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

2007－ARC－172 (27)
2007－HPC－109 (27)
2007／3／2

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

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[†††]

HPC 用途以外にも Google 型サーバーなども視野に入れた高性能コンピュータシステムの構築方法として，本報告では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を提案する．従来の主記憶の問題点を考察し，PC や東芝 Cell リファレンスセットのメモリシステムおよび I/O システムの改善を行う手段の一実現例として DIMMnet-3 を紹介する．小さな制御データと大きな応答データを混在して扱う際の遅延増加を抑制することを目的とした LHC や LHS と名づけた通信機構を提案・紹介し，それらの通信遅延性能に関する評価結果を示す．

Network/Memory Architecture with Large Memory in addition to Main Memory

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

In this report, we propose the memory/network architecture with large extended memory as a way to make high performance computer system not only for HPC but also for Google-like servers. At first, the problems on conventional main memory are mentioned. We introduce DIMMnet-3 for the solution to enhance memory system and I/O system of PC and Toshiba Cell Reference Set. Communication mechanisms named LHC and LHS are proposed or introduced. These are architecture for reducing latency for mixed messages with small controlling data and large acknowledge data. The latency evaluation of them is shown.

1. はじめに

PC クラスタは HPC(High Performance Computing) 用途を中心に高性能なコンピュータシステムへのニーズを支えてきた．ところがネットワーク文化および携帯電話に代表される小型情報ネットワーク機器の普及に伴い，近年の高性能なコンピュータシステムへの社会的ニーズとしては，検索エンジン Google[1] に代表されるネットワークサービスの高性能化と高機能化への期待と要求が高まってきている．このようなサービスを行うサーバー機器 (以降，Google 型サーバーと呼ぶ) では，クライアント側機器には保持しきれない大容量の情報を抱え込み，世界中のクライアント機器からの要望に沿った解析や加工や検索や更新を行った上で，適切な応答を許容可能な時間内に，社会的ニーズに応じたスループットで返す必要がある．

そのような Google 型サーバーの市場規模そのものは，携帯電話などのクライアント側機器の市場規模に比べるとあまり大きくはなくても，社会的なニーズは大きい．このため，Google のサーバーも歴史的には低価格な COTS(Commercial Off The Self) を用いたクラスタの技術によって構築されてきた [1]．しかし，ネットワーク社会の進展に伴うニーズの高まりへの積極的な対応を進めた結果，その大規模化に伴う電力問題が顕在化してきている．初期開発コストと消費電力を節約しつつ，大容量・高性能・高機能を達成しうる技術が重要性を増している．

一方，Google 型サーバーを構成している COTS の CPU の性能向上は目覚しく，近年ではマルチコア化に伴って，その傾向は一層強まってきている．これに対して，メモリや I/O の性能向上はこれに追いつくのが年々困難になってきている．

近年の LSI 開発費用は，90nm のプロセスではマスク 1 枚 1 億円とも言われるレベルまで高騰しており，新規マスクの生成を必要とする新規 LSI の開発をできるだけ回避する必要性が高まってきている．つまり，既存の CPU や既存のブリッジチップそのものを改造することなく，メモリや I/O における問題点へ対処できるソリューションが年々重要になってきている．

本報告では，第 2 章で従来の主記憶の問題点について述べる．第 3 章ではそれらの解決を行うための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と，その基本コンセプトを支援するための通信機構について提案する．第 4 章では，その具体例として Memory/Network Enhancer として開発を行っている DIMMnet-3 の概要を DIMMnet-2 と対比させつつ述べる．第 5 章では，前述のような改造を施した DIMMnet-3 が想定しているアプリケーションについて述べる．第 6 章では，大容量の主記憶以外のメモリにネットワーク側からのリクエストに応じて応答を返すべきデータを保持し，低遅延高バンド幅で応答を返すという処理をサポートする通信機構の評価を述べる．

† (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. 従来の COTS の主記憶に起因する問題点

本章では，HPC 用途や Google 型サーバー向けの高性能コンピュータシステム (COTS クラスタ) を構築する際の構成要素における，主記憶に関連した問題点を列挙する．チップ内外の通信も含め，あるがままの主記憶をメインに使うことには様々な状況下で以下に示すような多様な問題点がある．COTS としては Intel アーキテクチャの PC だけでなく，民生品における量産規模と高い演算性能を背景に近年有望視されつつある Cell Broadband Engine(CBE)[2][3][4] ベースのシステムも本章の対象範囲とする．

2.1 不連続アクセスへの不適合

不連続アクセスを主体とするアプリケーションにおいては，マルチコア化に伴ってただでさえバランスを欠きがちになっているメモリへの負荷が増大するため，著しい性能低下が発生してしまう．不連続アクセスへの不適合の主な要因として，以下の 4 つを列挙できる．

2.1.1 キャッシュアーキテクチャの不適合

COTS の CPU ではキャッシュアーキテクチャが採用されている．大きなデータセットに対する不連続アクセスを伴うアプリケーションにおいては，キャッシュライン単位でのアクセスに伴って，実際の計算に必要なデータ分量の 16 倍～32 倍の空間をアクセスしなければならないことがある．例えば実質 128 バイトのラインサイズを有する Pentium4 などの CPU を用いて主記憶データベースフルスキャンを行う場合，そのうちの 4 バイトしか CPU により使用されるデータが含まれないことがある．この場合，キャッシュラインやメモリバンド幅はその倍率だけ浪費されるため，高頻度でキャッシュミスを発生させる．同時に主記憶へのアクセスが増加してしまい，メモリバスのバンド幅が浪費される．こうして不連続アクセスを主体とするアプリケーションでは連続アクセスを主体とするアプリケーションと比べて激しい性能低下が発生する．なお，ラインサイズを小さくして程度は PC ほど大きくないが，この問題はキャッシュを有するベクトルプロセッサである Cray X1[5] においても存在する．

2.1.2 チップ内ネットワークの不適合

マルチコア化にともなってチップ内でのデータ転送に必要なバンド幅も高まってくるため，CBE のようにチップ内に非常に広いバンド幅を持つチップ内ネットワークを有する CPU も製品化されている．CBE の演算能力の大部分を担う 8 個の SPE と呼ばれるコアにはキャッシュアーキテクチャは採用されていないが，上記のチップ内ネットワークの性質上，短いデータのアクセス時にはチップ内ネットワーク上において転送効率が低下してしまう [4]．例えば CBE の場合は 128 バイトのデータをアクセスする場合も 1 バイトのデータをアクセスする場合もほぼ同じ時間がかかってしまう．このため前述のキャッシュアーキテクチャの不整合の問題と同様な問題が発生する．

2.1.3 仮想記憶アーキテクチャの不適合

COTS の CPU では TLB を用いた仮想記憶アーキテクチャが採用されている．不連続アクセスを伴うアプリケーションにおいては，多次元の大きな配列を各方向にアクセスする場合のように直前にアクセスしたアドレスから大きく離れたアドレスを頻繁にアクセスしなければならないことがある．この場合，アクセス間のアドレスオフセットがページサイズに対して十分に小さくないと，高頻度で TLB ミスやページフォルトが発生する．こうして不連続アクセスを主体とするアプリケーションでは連続アクセスを主体とするアプリケーションと比べて激しい性能低下が発生する．

2.1.4 DRAM アーキテクチャの不適合

従来の主記憶に用いられる同期 DRAM はキャッシュベースの CPU を前提として，キャッシュライン単位のバーストアクセスを想定し，そのようなアクセスが高速化されるように設計されている．このため，等間隔アクセスやランダムなアクセスを従来の主記憶に出した場合，同期 DRAM はその内部構造上の理由から，単体では高いスループットを提供できない．しかし，ベクトル型スーパーコンピュータの主記憶のように DRAM のバンク数を増やしてインターリーブドメモリを構成するならば，この問題を緩和することができる．しかし，大半の PC においてはバンク数も多くなく，ライン単位でのアクセスとなるため，問題が温存されている．

2.2 糊しろ領域 (HALO) 間の通信への不適合

メッセージ交換 API におけるデファクトスタンダードである MPI には派生データ型通信 [9][10][11] という不連続な領域間の通信を容易に記述できる．この機能は例えば領域分割時の糊しろ領域 (HALO) 間の通信のような不連続なアクセスを引き起こす．Global Array (GA) やその実装ライブラリである ARMCI[12] を用いる場合も同様である．このため，不連続アクセスへの不適合性を示す従来の主記憶を通信の送信元または送信先に用いていると，激しい性能低下を引き起こす．

2.3 通信時の I/O バス遅延への不適合

通常の I/O バス上に装着されるネットワークインタフェースを介して主記憶上のデータを送信，または主記憶上に受信する際には，必ず I/O バス上での調停などのオーバーヘッドが入ってしまい，特に細粒度の不連続アクセスを伴う場合には，通信効率が上がらないという問題点がある．例えば，通信制御語リストを多数リンクして NIC(Network Interface Card) 上のファームウェアによって実現される Infiniband の Gather/Scatter 通信においても，個々のデータの塊が小さい場合には性能向上があまり得られない [11]．この問題は I/O バスを PCI から PCI Express に変更したとしても，ほとんど状況が変わらない．

2.4 通信時のピンダウンオーバーヘッドへの不適合

RDMA(Remote DMA) は近年の高速通信を支える重要なハードウェア機構であるが，受信側での書き込み先が高頻度に再利用されない限り，オンライントランザクションのようにどこがアクセスされるのかについて再利用性があまり無い場合には，ピンダウンキャッシュ [6] と呼ばれるキャッシュがうまく機能せず，オンライントランザクション処理の性能低下をもたらすなどの問題がある．

2.5 容量拡張性の制約への不適合

マザーボード設計時に決められた最大容量以上のメモリを積むことができない．メモリ容量が重要なアプリケーションにおいて，マルチコア化による演算能力の向上に見合った記憶容量を確保できない．システム全体で保持すべきメモリ容量が大きい場合，必要な台数が増加して消費電力が高くなることもある．この問題は特に安価なマザーボードを用いて作成される Google 型サーバーにおいて顕著である．

3. 提案コンセプト

3.1 拡張大容量メモリを用いるシステムアーキテクチャ

DIMMnet が登場するまでの PC の主記憶は，チップセットメーカーが決めた仕様に準拠した通常のメモリモジュールを全スロットにできるだけ多くの容量のものを装着して用いる以外の改善方法は無かった．基本的にはキャッシュライン単位で発生するバーストアクセスを効率的に処理できることに最適化された既定容量を上限とする主記憶を，ハード的にはあるがままの状態で使うしかなかった．

よって，上記と相性が悪いアプリケーションでは，FFT のように可能な場合はキャッシュaware なアルゴリズム [13] に熟練したプログラマが全面的に書き換えるか，CG のようにその効果が期待できない場合は性能低下を受け容れるしかなかった．

前述のような問題点の解決策として「拡張大容量メモリを用いるシステムアーキテクチャ」を提案する．提案するコンセプトは「本来のマザーボードが想定する主記憶の容量を超える容量やアクセス性能やネットワーク機能を有する拡張大容量メモリを装着し，メモリシステムや I/O システムを改善したコンピュータシステムを構築すること」である．**図 1** に提案する基本コンセプトの概念図を示す．

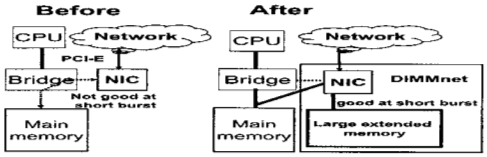

図 1 提案する基本コンセプト

元々，大容量なメモリを必要とするサーバーのコストの大半はメモリモジュールによって占められる．一方，消費電力の大半は CPU によって消費される．従来の主記憶で用いられるものと同等のメモリモジュールを上記拡張メモリ上で使用できるようにすることによって，コスト的には大きなインパクトをもたらさずに，メモリ容量がシステムの機能や性能を制約するアプリケーションにおいて，より高消費電力な CPU のシステム全体における使用個数を下げることができる．

複数の CPU ソケットを有するマザーボードは販売台数が少ないために高価であるとともに，特に AMD 社の CPU では CPU にメモリが直結されるため，メモリ容量を増やすためには CPU 台数を増やす必要がある．これに対して提案コンセプトは，メモリ容量を増やすために CPU 台数の増加を必要としない．

その結果，提案コンセプトを用いる場合は，CPU ソケットが 1 個しかない安価なマザーボードでも，CPU 単体の性能はマルチコア化によって急速に向上しているので，処理性能とメモリ容量の比率を崩さずに，1 筐体あたりのメモリ容量を維持したまま CPU 台数を減らすことができる．つまり，削減される CPU のコストやマザーボードのコストの合計が拡張大容量メモリを実現するための付加ハードウェアのコストと同等程度にすることができれば，初期コストはそのままで，消費電力や設置面積の削減に伴う維持費の節約を実現することができる．

さらに提案コンセプトを用いる場合は，本来の主記憶とは性質の異なったメモリシステムを増設することができる．例えば，ベクトル型スーパーコンピュータのように多数のメモリバンクをインターリーブして，大容量で，不連続アクセスへのスループットを高めたメモリシステムを拡張メモリとして実現することもできる．このようなシステムでは，CPU の著しい進歩をシステムに取り入れつつ，全体としてはベクトル型スーパーコンピュータに近い性質を有する高性能なコンピュータシステムを安価に実現することが可能になる．

3.2 拡張大容量メモリを用いる通信アーキテクチャ

上記提案コンセプトは，通信という観点を中心に言い換えると「主記憶よりも外部ネットワークからアクセスしやすい場所に拡張された大容量かつアクセス性能が改善されたメモリを用いることで，通信性能を向上すること」である．

通信に用いられるデータには大別して以下の 2 種類に分類できる．

(1) 制御用のデータ

この種のデータはホスト CPU に低遅延で伝達され，ホスト CPU から低遅延でネットワークに送出されるべき，比較的短い長さのデータである．通信経路制御などに用いられるハードよりの制御データや，MPI のエンベロープなどの MPI 処理系で用いられる制御データもこのカテゴリーに分類される．

(2) メッセージ本体

この種のデータは大容量な場合はサーバー上では従来 HDD 上に配置され，制御用のデータのやり取りに伴って必要が生じた場合に読み出されたり，応答データとしてメッセージに組み立てられる．データのアクセスパターンの規則性や再利用性が無い場合はピンダウンキャッシュがミスヒットするため通信遅延が更に増加する．

通信に用いられるデータの上記の性質上の違いに鑑み，それぞれを上記提案コンセプトを適用したシステム上で効率的に扱うために，送信側の通信アーキテクチャと受信側の通信アーキテクチャを提案する．

3.2.1 制御データ用送信側通信アーキテクチャ

ホスト CPU 側で生成される短いデータを低遅延でネットワークに送出する性質を有する通信手段を用いる．具体的には AOTF[14] や BOTF[15] がこの用途には適する．

3.2.2 制御データ用受信側通信アーキテクチャ

短いデータを低遅延でネットワークから取り込み，ホスト CPU 側に低遅延で伝達する性質を有する通信手段を用いる．具体的には LHS[23] がこの用途には適する．LHS を用いた受信では短いメッセージは小容量のオンチップメモリ上に受信される．

3.2.3 メッセージ本体用送信側通信アーキテクチャ

ホスト CPU 側で生成される短い制御用のデータと大容量拡張メモリ上に保持されたデータ本体を用いて一個のメッセージとし，これを低遅延高バンド幅で送ることができる通信手段を用いる．具体的には以下に示す送信側通信機構 LHC を提案する．

Limited Head Conbining(LHC)

小容量のオンチップメモリ上に保持されたメッセージ前半部と，大容量のオフチップメモリ上に保持されたメッセージ後半部を組み合わせて一つのメッセージとしてネットワークに送出する送信方式．**図 2** に LHC と LHS の基本コンセプトの概念図を示す．LHS[23] が受信側の挙動をメッセージの前半部と後半部に分けて切り替えるのに対して，LHC はその概念を送信側に拡張したものである．

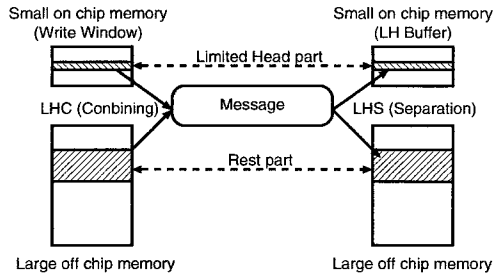

図 2 LHC と LHS の基本コンセプト

3.2.4 メッセージ本体用受信側通信アーキテクチャ

短い制御用のデータ部分 (メッセージ前半) を低遅延でネットワークから取り込み，ホスト CPU 側に低遅延で伝達するとともに，メッセージ本体部分 (メッセージ後半) を大容量のオフチップメモリ上に受信する通信手段を用いる．具体的には LHS[23] がこの用途には適する．

4. DIMMnet-3 の概要

4.1 改善ポイント

DIMMnet-2 は基本コンセプトの機能確認用のプロトタイプであり，性能や容量やサイズや信頼性やコストなどにいくつかの課題を残していた．DIMMnet-3 では，DIMMnet-2 において確立してきた基本コンセプトを踏襲しつつ，これらの課題をいくつか解決する．DIMMnet-2[22] と比較して DIMMnet-3 において改善されるポイントを以下に示す．

4.1.1 搭載メモリの大容量化と高信頼化

DIMMnet-3 はオンボードメモリ上に，これまで主記憶上で保持されてきた大半のデータの本体を保持し，仮想記憶を介して主記憶をオンボードメモリのキャッシュとして利用可能とする．このために，DIMMnet-3 の on-board memory を DIMMnet2 よりも数十倍大容量にする．大容量のメモリを安定して動作させられるようにするため，我々はソフトエラー対策も必須と考え，ECC 付きのメモリを許容するように作成する．

4.1.2 不連続アクセススループットの向上

DIMMnet-3 は DIMMnet-2 と同様のいくつかのバリエーションを有するベクトルコマンドにより不連続データのロードストアを行う機能を持つ．従来の DIMMnet-2 では DRAM チップ内の複数バンク動作の未実装や，メモリチャネル数の不足などの理由から十分な不連続アクセススループットが現状では実装していないため，バックエンドメモリシステムのスループット不足が不連続アクセスの連続化による本来の効果 [16][17] を薄めてしまっていた．[18][19] 一方，DIMMnet-3 では上記のメモリ容量拡張に伴い，メモリバンク数は大幅に増加している．このため，これらの多数のメモリバンクを同時にアクティブにして並行して動作させることで，不連続アクセススループットを現状の DIMMnet-2 よりは大幅にベクトル型スーパーコンピュータの主記憶に近づけることができると考えている．

4.1.3 MPI 高速化支援

DIMMnet-2 は FPGA ベースで構築されていることもあり，その実験の後期においては DIMMnet-3 に搭載する予定であった機能の一部も追加して機能確認を行ってきた．その機能の中には MPI の高速化をする機能群がある．その機能群について詳細を記述するスペースがないため，概略のみ列挙する．

(1)IPUSH[22]
送信側では最終書き込みアドレスを指定せず，ハードウェアによりリモートメモリ上に形成された FIFO の一つに高速にメッセージを書き込む．

(2)LHC
大容量オフチップメモリに格納された大きなデータに，小容量オンチップメモリ (Write Window) 上に格納されたエンベロープ部分を付加し，少ないホスト負荷かつ低遅延で送信する．

(3)LHS[23]
メッセージ前半を小容量オンチップメモリ (LH バッファ) に格納し，溢れたメッセージ後半を大容量オフチップメモリに格納することで短いメッセージやメッセージのエンベロープ部分を低遅延でホストに伝達する．

(4)HTP[23]
LHS と組で用いられ，より積極的なプリフェッチを行うことで遅延時間の短縮と実効バンド幅の向上をはかる．

(5)VCOPY[23]
ベクトルコマンドを用いて搭載メモリ間のコピーをホストが行うことに伴うオーバーヘッドを軽減する．

(6)VDDC[23]
ベクトルコマンドを用いて搭載メモリ上に不連続に配置されたデータ間のデータ転送を高速化する．

4.1.4 様々なホストへの対応

DIMMnet-2 は特に高さにおいて大きな物理的なサイズを持っている．ゆえに，大規模な PC クラスタ向けに多用される 1U サーバには装着できなかった．基板分割を行うことにより，DIMMnet-3 は 1U サーバに装着できるように設計される．さらに，PC だけでなく東芝 Cell リファレンスセット [2][3] にも DIMMnet-3 を装着できるように設計される [21]．

4.1.5 汎用の支持構造への対応

DIMMnet-2 は DIMM スロットに非常に大きな基板を無理やり装着していた．このため，コネクタや接点への無理な力がかからないように筐体に合わせて特注する基板支持構造物を DIMMnet-2 は必要とした．一方 DIMMnet-3 の場合は，本来の DIMM のサイズより若干大きな軽量の基板が DIMM スロットに装着されるので，斜めざしが可能な DIMM スロットを有する 1U サーバーにも装着できる．基板を分割しているため，将来的には FB-DIMM や PCI Express などのホストインタフェースを有する子基板を少ないコストと手間で開発することができる．さらに親基板側は PCI バスへの装着とともに 3.5 インチ HDD ベイへの装着も可能とする．これにより，大型の PCI カードが 1 枚しか入らない多くの 1U サーバにおいても，複数の親基板を装着することにより，大容量の半導体拡張メモリを構成できる．

4.1.6 Si ディスクドライバによる高速仮想主記憶

DIMMnet-2 上には既に Windows 上で動作する Si ディスクドライバを開発済みである．ローカルおよびリモート側の DIMMnet-2 上に搭載されたメモリ上に確保された領域をソフト的には HDD として見せかけることに成功している．さらに，その仮想的な HDD 上にスワップファイルを確保することにより，ホスト上に本来装着できる限界より大きな容量の半導体メモリがアプリケーションからは主記憶として実装されているかのように動作させることができる．DIMMnet-3 ではこの機能を Linux にも移植する．

4.2 基本仕様上の改善状況

DIMMnet-2 と DIMMnet-3 のホストインタフェースの違い，装着可能なメモリモジュールの違い，およびその他の基本仕様上の違いをそれぞれ表 1，表 2，表 3 に示す．DIMMnet-3 は Frontend と Backend の 2 枚の基板に分割されていて，その間が基本的には複数の高速シリアルインタフェース (Rocket IO) と同軸ケーブルを用いて接続されている．このため，Frontend 基板を交換することにより，様々な種類のホストインタフェースに容易に対応することができる．なお，東芝 Cell Reference Set(CRS) 向けのホストインタフェースの Frontend 基板のみ，FPGA は搭載せず，ケーブルの引き出しのみを行う．

共通の Backend ボードの外形をどのように切断して金具を取り付けるかを変えることによって，DIMMnet-3 の形状は以下の 3 種類が選択できるように設計されている．

(1) All-in-one type : DIMMnet-2 と同様の All in one 型の大きな DIMM の形状
(2) PCI type : PCI バスに装着される PCI ボードの形状
(3) HDD type : 3.5 インチ HDD と同様の形状

ただし All in one 型の場合は，最大 2 枚の大型 DIMM を増設できるライザーカードが装着されるべきカードエッジをホスト I/F として使用するので，搭載メモリ容量は大幅に減る．

表 1 DIMMnet-2(D2) と DIMMnet-3(D3) のホストインタフェース (CRS : 東芝 Cell リファレンスセット)

	Base	Buffer	ECC	Speed
D2	DDR-DIMM	unbuf.	No	100MHz
D3	DDR2-DIMM	unbuf.	No	266MHz
D3	DDR2-DIMM	reg.	Yes	266MHz
D3	DDR2-SODIMM on CRS			166MHz
D3	FB-DIMM			?
D3	PCI Express			4X/8X/16X

表 2 DIMMnet-2(D2) と DIMMnet-3(D3) の受容可能なメモリモジュール

	Speed	Base	Buffer	ECC	Capacity
D2	100MHz	DDR-DIMM	unbuf.	No	256MB
D3	266MHz	DDR2-SODIMM	unbuf.	No	1GB
D3	266MHz	DDR2-SODIMM	reg.	Yes	2GB
D3	266MHz	DDR2-DIMM	unbuf.	No	1GB
D3	266MHz	DDR2-DIMM	reg.	Yes	1-8GB

表 3 DIMMnet-2 と DIMMnet-3 の基本仕様 (IB : Infiniband)

	DIMMnet-2	DIMMnet-3
Mem. capacity	512MB	28GB
# of Mem. module	2	5
Mem. bandwidth	3.2GB/s	12.8GB/s
Network I/F	IB 4X	IB 4X DDR
Network BW	1GB/s/dir.	2GB/s/dir.
All in one DIMM	yes	yes
Frontend + PCI	N.A.	yes
Frontend + HDD	N.A.	yes

5. アプリケーション

5.1 既存チップの改善

東芝 Cell リファレンスセットでは，I/O チップとして Super Companion Chip(SCC) を搭載している．SCC には PCI Express (x4) のインターフェースがあるが，これ以上のバンド幅で外部装置とやりとりするインターフェースは搭載していない．SCC を改造するためには非常に大きなコストがかかるが，DIMMnet-3 を SCC の DDR2 SO-DIMM スロットに装着することによって 1GB/s 以上のバンド幅に少ない開発コストで対応できるものとすることができる．また，Cell の主記憶が XDR DRAM という特殊なメモリをポイント to ポイント接続する必要があるために，実装可能な主記憶容量が一般的な PC の主記憶より小さくなってしまう．これはメモリ容量が重要な意味を持つ技術計算やデータベース応用においては大きな問題である．DIMMnet-3 は Cell ベースのシステムであっても，多額な開発コストを払うことなしに数十 GB クラスの半導体メモリを実装可能としている．**図 3** に IEEE MICRO 誌上で発表された Cell Broadband Engine のバンド幅のグラフ上に DIMMnet-3 が期待される効果を示す．ここで示されるように，Cell の内部バスはそのパケットサイズである 128 バイト単位以上でアクセスしないと効率的に動作しないが，DIMMnet-3 を装着することにより，4 バイトや 8 バイトといった細粒度の不連続アクセスが発生するようなアプリケーションに対しても，512 バイト単位以上でアクセスすることができるようになるので，高い性能を維持できるものと考えられる．

5.2 高速メモリの容量拡張

Oracle 社の TimesTen のような主記憶データベースは，検索対象のデータベースの全データが主記憶上に載ることが必須となっている．このため，大容量の主記憶を有する高価なプラットフォーム上でしか大規模で実用的なシステムを構築することができなかった．これに対して，DIMMnet-3 は前述の通り，性能的にも仮想的に主記憶が大きくなった状態を実現できるため，より安価なプラットフォーム上でも大規模で高速な主記憶データベースを実現することができる．

有限要素法などを用いた構造計算などの前処理で大規模で細かいメッシュを生成する必要があるが，このアプリケーションでも大容量の主記憶が必要になる．並列処理は必要でないのに大容量の主記憶を使うために高価な SMP システムを購入して実施しているところが多いと言われている．しかし，DIMMnet-3 を用いれば前述の主記憶データベース同様に，アプリケーションプログラムの変更なしに，安価なプラットフォーム上で実行することが可能となる．

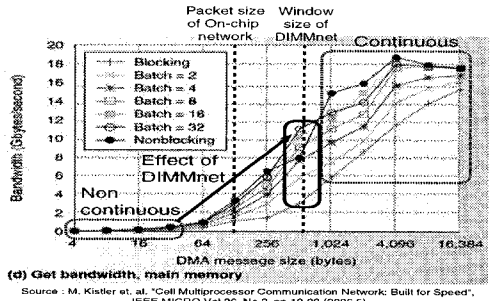

図 3 Cell Broadband Engine のバンド幅に対する DIMMnet-3 の効果

5.3 メモリネックのサーバにおける電力削減

Google に代表される Web 検索エンジンはそのレスポンスの高速化を実現するために検索対象の大半を半導体メモリ上に配置する必要がある．このため，世界中の Web 情報を検索対象として，世界中の多数の利用者からの検索要求に対応するために，数万台規模の安価な PC から構築される PC クラスタが用いられている．このため，電力や，サーバーを配備する建物や空調など，維持費が膨大なものになっている．これに対し，**図 4** に示されるように DIMMnet-3 を用いれば，ホストの OS を巻き込むことも抑制しつつ，I/O バスのトランザクションも最小化しつつ，ネットワーク側からアクセス可能な大容量の半導体メモリを，少ないノード数で実装することができる．このため，結果的には一桁程度の台数削減と維持費削減が達成できるものと考えられる．

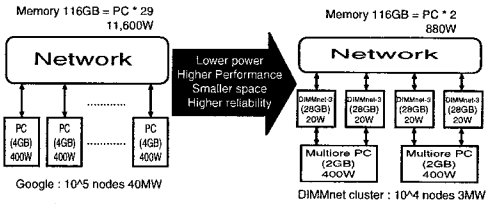

図 4 DIMMnet-3 による Google 型サーバーの電力削減

5.4 ビデオ配信サーバの高速化

ビデオデータを配信するサーバは，アクセスが集中する時間帯にはサーバのデータ送信能力をオーバーフローしてしまうケースが少なくない．配信要求の多いビデオの大半が半導体メモリ上にキャッシングされるならば，この限界を大幅に改善することが可能であると考えられる．DIMMnet-3 を用いることにより，既存のサーバを全面的にリプレースしなくてもアプリケーションから見て主記憶として見える半導体メモリの容量を大幅に向上させることができるため，低コストなビデオサーバの能力向上を行うことが可能となる．

5.5 糊しろ通信の高速化

QCD シミュレーションは 4 次元の配列データを扱い，隣接格子間でデータを交換して時間発展するため，複数の格子点を 1 台の PC 上に分担させる場合は糊しろ通信が発生してしまう．この場合，糊しろとなるデータはメモリ上では等間隔に配置されることになるため，従来はパッキング，アンパッキングをホスト上のソフトウェアで実現していた．これにより見かけ上の CPU 処理時間の割合が多くなっていた．しかし，**図 5** に示されるように MPI の派生データ型によってこの糊しろを定義して，その大半を DIMMnet-3 上のハードウェアで実行させるならば，これまで CPU 時間の多くを占めていたパッキング，アンパッキング処理にかかる時間を多数のメモリバンクを用いた高バンド幅な不連続アクセスによって大幅に削減することが可能と考えられる．

5.6 意思決定サポートシステムの高速化

データマイニングなどのデシジョンサポートシステムでは大量のデータを様々な角度から分析して，そこから新しい知見を発見し，経済的な価値に繋がるデシジョンを行うことをサポートする．この際，データの全て，または大半を DIMMnet-3 を装着した単体ノードまたはそのクラスタ上の搭載メモリに配置すれば，リレーショナルデータベースのフルスキャンが発生す

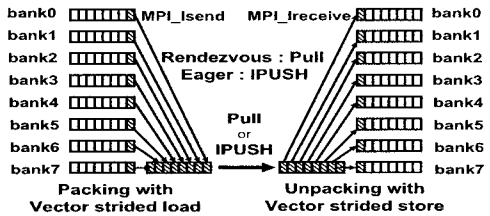

図 5 DIMMnet-3 による糊しろ通信の高速化

るようなクエリーが実行されても，等間隔アクセスに伴うキャッシュミスヒットや TLB ミスを抑制し，データを HDD に置くものや単純に主記憶上に置くものに比べて大幅に高速化 [17] することが可能となる．

5.7 OLTP システムの高速化

オンライントランザクションシステムでは，そのアプリケーションによってはデータベースのどこがアクセスされるか事前にはその傾向が把握できないため，RDMA 通信を行うことで高速化しようとしても，ピンダウンキャッシュ [6] がヒットしないなどの理由により，十分な加速が得られないものと考えられる．トランザクションのメッセージを受信するたびに OS に割り込みをかけて動的にピンダウンを行う Active Zero-copy 方式 [7][8] も提案されているが，割り込みはそれだけでもオーバーヘッドが大きく，効果は限定的になる．これに対して，**図 6** に示されるように DIMMnet-3 により I/O バスの外側に形成された大容量の半導体メモリに対して RDMA を行うように実装するならば，その領域はページングの対象にはなっていないためピンダウンのオーバーヘッドが不要であり，処理は極めて単純なものになり，低遅延でリモートノードからのデータ更新を行うことが可能になる．

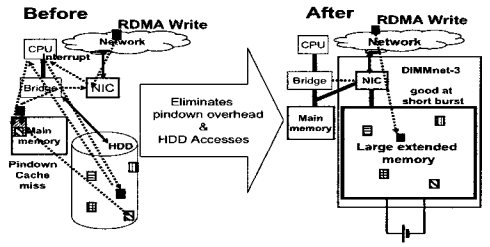

図 6 DIMMnet-3 による OLTP 処理の高速化

6. 評　　価

本報告では，主記憶以外の大容量メモリに応答として返すべきデータを保持し，ネットワーク側からのリクエストに応じて低遅延高バンド幅で応答を返すという処理をサポートする送信機構である LHC および受信機構である LHS を DIMMnet-2 プロトタイプ上に実装して，実機上で評価する．

6.1 測 定 方 法

本章の実験において用いた測定環境を表 4 に示す．本実験では以下に示す 4 種類の条件で通信遅延時間の測定を行った．送信するデータは主記憶だけでなく DIMMnet-2 上の SO-DIMM(外部メモリモジュール) に格納済みと仮定する．データサイズはエンベロープ (8Byte) を加えたサイズになっており，データサイズ=8Byte の場合は，エンベロープのみを送信している．

6.1.1 BOTF+IPUSH の場合

送信側 (1)
- Write Window に主記憶上のパケットヘッダ (16Byte), エンベロープ (8Byte), データ (0〜120Byte) を書き込む,
  - パケットヘッダは IPUSH のパケットになるように指定
- BOTF 要求を発行

受信側 (1)
- LLCM 内のパケット受信ステータスをポーリングし，パケットの受信を検知
- パケット受信ステータスを読み出し，受信データサイズ，受信先のアドレスを取得
- SO-DIMM からエンベロープ (8Byte) を読み出し，rank, context, tag の比較
- SO-DIMM からデータ部を読み出し，主記憶にコピー

受信側 (2)
- 送信側 (1) と全く同じ処理を行う

送信側 (2)
- 受信側 (1) と全く同じ処理を行う

表 4 評価環境

CPU	Pentium4 2.6GHz, L2=512KByte
Chipset	VIA VT8751A
Memory	PC-1600 DDR-SDRAM 512MBytes x1
DIMMnet-2	FPGA version (100MHz)
Switch	Voltaire ISR6000 (InfiniBand 4X) with SW-6IB4C (Switch module)
Cable	2m
OS	RedHat8.0 (Kernel 2.4.27)
Compiler	gcc 3.3.6 (compile option: -Wall)

6.1.2 BOTF+IPUSH+LHS の場合

送信側 (1)
- Write Window に主記憶上のパケットヘッダ (16Byte), エンベロープ (8Byte), データ (0〜120Byte) を書き込む,
  - パケットヘッダは LHS を使用する IPUSH のパケットになるように指定
- BOTF 要求を発行

受信側 (1)
- LH バッファをポーリングし，パケットの受信を検知
- LH バッファからエンベロープ部 (16Byte) を読み出し，rank, context, tag の比較
- LH バッファからデータを読み出し，主記憶にコピー
- 受信したデータが LH バッファの 1 エントリに収まらない場合 (データ部のサイズが 48Byte より大きい場合) は SO-DIMM からデータを読み出し，主記憶にコピー

受信側 (2)
- 送信側 (1) と全く同じ処理を行う

送信側 (2)
- 受信側 (1) と全く同じ処理を行う

6.1.3 VL+BOTF+IPUSH+LHS の場合

送信側 (1)
- ベクトルロード (VL) で SO-DIMM から送信するデータを読み出す
- 読み出している間に Write Window にパケットヘッダ (16Byte) を書き込む,
  - パケットヘッダは IPUSH のパケットになるように指定
- Prefetch Flag をポーリングし，VL の完了を検知
- 読み出したデータをエンベロープと一緒に Write Window に書き込む
- BOTF 要求を発行

受信側 (1)
- LH バッファをポーリングし，パケットの受信を検知
- LH バッファからエンベロープ部 (16Byte) を読み出し，rank, context, tag の比較
- LH バッファからデータを読み出し，主記憶にコピー
- 受信したデータが LH バッファの 1 エントリに収まらない場合 (データ部のサイズが 48Byte より大きい場合) は SO-DIMM からデータを読み出し，主記憶にコピー

受信側 (2)
- 送信側 (1) と全く同じ処理を行う

送信側 (2)
- 受信側 (1) と全く同じ処理を行う

6.1.4 LHC+IPUSH+LHS の場合

送信側 (1)
- Write Window にエンベロープ (8Byte) を書き込む
- LHC を使用する IPUSH 要求を発行

受信側 (1)
- LH バッファをポーリングし，パケットの受信を検知
- LH バッファからエンベロープ部 (16Byte) を読み出し,rank, context, tag の比較
- LH バッファからデータを読み出し，主記憶にコピー
- 受信したデータが LH バッファの 1 エントリに収まらない場合 (データ部のサイズが 48Byte より大きい場合) は SO-DIMM からデータを読み出し，主記憶にコピー

受信側 (2)
- 送信側 (1) と全く同じ処理を行う

送信側 (2)
- 受信側 (1) と全く同じ処理を行う

ただし，LHC+IPUSH+LHS の場合のみ，SO-DIMM から読み出すサイズを 0Byte にできない仕様にしてしまっているので，エンベロープ 8Byte+データ 8Byte の 16Byte から測定を

開始している．

6.2 測定結果と考察

LHC,LHS を用いた通信の遅延時間の測定結果を**図 7** に示す．通信遅延は送信側に BOTF，受信側に IPUSH と LHS を併用した場合が最も短い．つまり，単純な制御データのみをやり取りする場合は上記の方法を用いるのがベストである．受信側に LHS を用いないで送信側を BOTF で送った場合は受信データが全て外部メモリ (SO-DIMM) に書き出されてしまうので，ホストから MPI のエンベロープを読み出すにも外部メモリアクセスが発生し，メッセージ本体を読み出すにも再び外部メモリアクセスが発生してしまう．このため大幅な遅延上昇が発生してしまう．

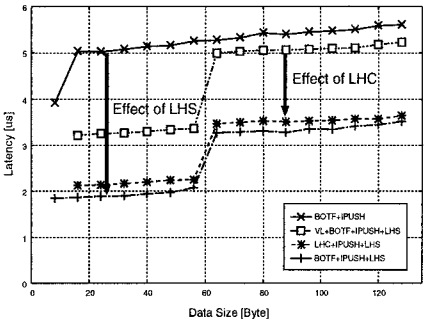

図 7 LHC,LHS の通信遅延に対する効果

これに対して送信側で LHC を用いた場合は，BOTF で送信するよりは若干遅延時間がかかるものの，受信側に LHS を用いない場合と比較して大幅に低遅延で SO-DIMM 上に存在するデータを送り出すことに成功している．BOTF をする際の Window メモリのサイズには 512 バイトという制限長があるが，LHC で送り出すデータは SO-DIMM 上に保持された大きなデータを送れるので，データが長かったとしても送信側ではホストによるソフトでコピーを行う必要もないため，LHC と LHS を用いれば SO-DIMM 上のデータを効率よく送信することができると考えられる．

一方，SO-DIMM 上にしかデータがなく，送信側で LHC を用いない場合は SO-DIMM 上のデータを一旦主記憶上にベクトルロードコマンド (VL) とホスト上のソフトによってコピーした後，それを改めてホスト上のソフトによって BOTF で送信しなおす必要がある．その場合，1 μ 秒以上の遅延上昇と CPU 負荷上昇が発生しており，LHC の効果が観測できた．

7. ま と め

本論文では，従来の主記憶における問題点を列挙し，その解決策を提案した．その具体例である DIMMnet-3 の概略仕様を示した．さらに，DIMMnet-3 が想定している利用形態の例について述べた．

以上により，通常の PC や Cell ベースのシステムの主記憶や I/O を DIMMnet-3 で強化することにより，様々なシチュエーションで高価なマスクを生成することなく，システムの高性能化や低電力化がはかれる可能性があることを示した．

さらに，大容量の主記憶以外のメモリに応答として返すべきデータを保持し，ネットワーク側からのリクエストに応じて低遅延高バンド幅で応答を返すという処理をサポートする送信機構である LHC および受信機構である LHS の評価を行ない，それらの効果を観測した．

今後は DIMMnet-3 のハードウェアやサポートソフトウェアの実装を進めるとともに，様々なアプリケーション上での有効性を実験確認する予定である．

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いる慶應義塾大学の西講師，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，表氏，森山氏，高柳氏，種田氏，藤岡氏，和歌山大学の齋藤講師，京都大学の上原助教授，日立 JTE 社の上嶋氏，今城氏，岩田氏，森山氏に感謝いたします．

参 考 文 献

1) L. A. Barroso, J. Dean, U. Holzle : "Web Search for a Planet : The Google Cluster Architecture", IEEE Micro, Vol.23, No.2, pp.22-28 (Mar. 2003)
2) 東芝セミコンダクター社 : "Cell Broadband Engine", http://www.semicon.toshiba.co.jp/product/micro/cell/index.html
3) Cell User's Group : "Cell 関連情報", https://www.cellusersgroup.com/modules/product/
4) M. Kistler, M. Perrone, F. Petrini : "Cell Multiprocessor Communication Network : Built for Speed", IEEE MICRO Vol.26, No.3, pp.10-23 (2006.5)
5) T. H. Dunigan Jr., J. S. Vetter, J. B. White III, P. H. Worley : "Performance Evaluation of the Cray X1 Distributed Shared-Memory Architecture", IEEE MICRO Vol.25, No.1, pp.30-40 (2005.1)
6) H. Tezuka, F. O'Carroll, A. Hori, Y. Ishikawa : "Pin-down Cache: A Virtual Memory Management Technique for Zero-copy Communication", 12th Int. Parallel Processing Symposium, pp.308-314 (Mar. 1998)
7) 青木, 山際, 和田, 小野 : "Maestro2 クラスタネットワーク向けメッセージパッシングライブラリの開発と評価", 電子情報通信学会論文誌 D Vol. J89-D No.5, pp.919-931 (2006.5)
8) S. Yamagiwa, K. Aoki and K. Wada : "Active Zero-copy:A performance study of non-deterministic messaging", Proceedings of the 5th International Symposium on Parallel and Distributed Computing (2005).
9) 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)
10) 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)
11) 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)
12) 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
13) 高橋, 朴, 佐藤 : Short Vector SIMD 命令を用いた並列 FFT の実現と評価, 情報処理学会論文誌 : コンピューティングシステム, Vol. 45, No. SIG 11(ACS 7), pp. 50-61, (Oct. 2004)
14) 田邊, 濱田, 山本, 今城, 中條, 工藤, 天野 : DIMM スロット搭載型ネットワークインタフェース DIMMnet-1 とその低遅延通信機構 AOTF, 情報処理学会論文誌ハイパフォーマンスコンピューティングシステム, Vol.43, No.SIG(HPS6), pp.10-23 (Jan. 2003)
15) 田邊, 山本, 濱田, 中條, 工藤, 天野 : "DIMM スロット搭載型ネットワークインタフェース DIMMnet-1 とその高バンド幅通信機構 BOTF", 情報処理学会論文誌, Vol.43, No.4, pp.866-878 (Apr. 2002)
16) 田邊, 安藤, 箱崎, 土肥, 中條, 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 PC 上での間接参照の高速化", 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.1-12 (Aug. 2005)
17) 田邊, 羅, 中條, 箱崎, 安藤, 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)
18) 宮代, 宮部, 北村, 田邊, 中條, 天野 : "DIMMnet2 を用いた間接メモリアクセスの高速化", 2006-ARC-170, pp.85-90 (Nov. 2006)
19) 宮部, 宮代, 北村, 田邊, 中條, 天野 : "ハードウエアによる MPI 派生データ型通信の支援", 2006-ARC-170, pp.91-96 (Nov. 2006)
20) 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)
21) 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)
22) 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)
23) 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)