Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

Gather 機能付き拡張メモリのアクセス性能の評価

田邊 昇[†]　Boonyasitpichai Nuttapon[††]　中條 拓伯[††]

各種プロセッサのメモリシステムにおける不連続アクセスにおける問題を解決するために筆者らは先行研究で後付けが可能な Scatter/Gather 機能を有する拡張メモリシステムを提案した。これまで Wisconsin ベンチマーク、疎行列ベクトル積などのアプリケーションで評価を行ない、有効性を示してきた。それらの評価研究では提案拡張メモリシステム側のスループットを十分にできるということが前提であった。本報告ではその前提が成り立つか否かについて、メモリシステム側のスループットの実現性を DRAMsim2 ベースのシミュレーションによって評価した。その結果、DDR3・8bit 幅・8 チャネル以上で先行研究の結果の正当性を支持する所望のスループットが得られることを確認した.

Performance Evaluation of an Extended Memory with Gather Functions

Noboru Tanabe[†]　Boonyasitpichai Nuttapon[††]
and　Hironori Nakajo[†]

In order to overcome the problems of discontinuous accessing in memory systems of various processors, we have proposed an extended memory system which has an additional function of scattering and gathering. So far, we have been evaluating our proposed system with Wisconsin benchmark suites and sparse matrix vector multiplications etc. In the evaluations, we assumed that throughput of our proposed memory system was sufficient. In this paper, we have confirmed the assumption on feasibility of throughput of the memory system is correct or not with a simulator based on DRAMsim2. As the result, we have confirmed desired throughput which can justify the proceeded results can be implemented with DDR3 of 8 bits width over 8 channels.

1. はじめに

キャッシュベース CPU のメモリシステムや、DMA ベースのメモリアクセスを行う Cell/B.E.系の CPU のメモリシステムや、広ビット幅メモリの GPU のメモリシステムなど、主なプロセッサのメモリシステムは連続アクセスを行うアプリケーションには効率的に稼動するように設計されている。一方、不連続アクセスを行うアプリケーションはメモリシステムに多くの投資をしているベクトル型スーパーコンピュータを除く上記のプロセッサでは大幅な性能低下が発生するという問題がある。

そのような問題を解決するために筆者らは先行研究[1]-[9]で後付けが可能な Scatter/Gather 機能を有する拡張メモリシステムを提案した。これまで NAS CG ベンチマーク、Wisconsin ベンチマーク、ボリュームレンダリング、疎行列ベクトル積などのアプリケーションで評価を行ない、有効性を示してきた。それらの評価研究では提案拡張メモリシステム側のスループットを十分にできるということが前提であった。本報告ではその前提が成り立つか否かについて、メモリシステム側のスループットの実現性をシミュレーション評価した。

以下、本報告では第 2 章でメモリシステムの課題を整理する。第 3 章ではその課題の解決に関連し、考慮しておきたいメモリシステムの最近の動向について論じる。第 4 章では Gather 機能付き拡張メモリの基本アーキテクチャとそれに付随する実現技術について述べる。第 5 章では想定されるアプリケーションを紹介し、第 6 章ではメモリシステム側のスループットの実現性に関するシミュレーション評価を示す。第 7 章でその考察を述べ、第 8 章で関連研究を紹介したのち、第 9 章でまとめる。

2. メモリシステムの課題

本章ではキャッシュベース CPU、Cell 系 CPU、GPU の各メモリシステムにおける不連続アクセスや容量不足に伴う課題を論じる。

2.1 キャッシュベース CPU のメモリシステム

2.1.1 ライン単位転送に伴うバンド幅浪費

キャッシュベースの CPU では主記憶へのアクセスは最も主記憶に近い階層のキャッシュのミスヒットに伴うリプレースメントで起きるため、最も主記憶に近い階層のキャッシュのラインサイズに固定される。通常、キャッシュラインサイズは 64 バイトか 128 バイトに設定されている．ところが，NAS CG ベンチマークや Wisconsin ベンチマークに代表されるいくつかの重要アプリケーションでは 8 バイトまたは 4 バ

[†]株式会社　東芝
Toshiba corporation
[††]東京農工大学
Tokyo University of Agriculture and Technology

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

イトの不連続アクセスが多発する．このため，このようなアプリケーションにおけるバスの実効バンド幅は 8/128 または 4/128 に低下してしまう．

2.1.2 ライン単位転送に伴うキャッシュ容量浪費

キャッシュベース CPU はキャッシュ上にライン単位で連続領域を記憶する。上記のようなアプリケーションでは、空間的局所性が乏しく、ライン上には 8/128 または 4/128 しか有効なデータが保持されていないことになるので、限られたキャッシュ容量を浪費することになる。

2.2 Cell 系 CPU のメモリシステム

2.2.1 DMA コマンド起動に伴うバンド幅浪費

DMA コマンドを発行するには少なからずソフトウェアオーバーヘッドが存在するので，細粒度の DMA 転送が頻繁に発生するアプリケーションの性能は制約される．この問題は Cell/B.E.にも実装されている DMA リスト[11]を用いることによりある程度軽減することが可能である．

2.2.2 内部バス調停に伴うバンド幅浪費

Cell/B.E.のように調停回路から内部バスのアクセス権利を取ってから DMA 転送を行なう種類の CPU では，少なからず調停オーバーヘッドが存在するので，細粒度の DMA 転送が頻繁に発生するアプリケーションの性能は制約される．この問題は Cell/B.E.にも実装されている DMA リスト[11]を用いても軽減することができない．

2.2.3 FLIT 単位転送に伴うバンド幅浪費

Cell/B.E.では前述の調停オーバーヘッドとの兼ね合いからも長めのバースト転送における内部バスの転送効率を向上させるために，内部バスの最小転送単位(FLIT)が 128 バイトに設定されている．よって、DMA リスト[11]を用いて DMA コマンドオーバーヘッドを軽減したとしても，キャッシュの場合と同様に，上記のようなアプリケーションにおけるバスの実効バンド幅は 8/128 または 4/128 に低下してしまう．

2.2.4 アラインメントに伴うバンド幅浪費

Cell/B.E.の DMA コントローラのように，DMA 転送を行う際のソースとデスティネーションの間でアラインメントがずれていると，直接 DMA でコピーできない実装がある．その場合，一旦バッファ領域に転送したいデータを含むブロックを DMA 転送し，その後にロードストア命令で所望の位置にソフト的にコピーしなおす必要がある．CTK(Cell Tool Kit)ではそのような操作をプログラマから隠蔽できるが、サイズが小さい場合は無視できないオーバーヘッドが存在する。[7][8] Cell/B.E.には上記のようなオーバーヘッドが存在するため，アプリケーションのアクセスパターンによっては性能低下の原因となる．

2.3 GPU のメモリシステム

2.3.1 統合アクセス成否による性能変動

GPU ではデバイスメモリへの複数のアクセス要求がまとめて実行される統合(Coalesced)アクセスが発生するようにプログラミングされないと、メモリバンド幅バウンドなアプリケーションの性能は大幅に低下する。グループ内アクセス間の順序、アラインメントのずれや、グループ内アクセス群が例えば「4 バイトアクセスの場合に 128 バイト以内」等の一定の領域内にないと Coalesced 転送が発生せず、実効バンド幅が大幅に低下する。GPU 側で世代が新しくなるごとに性能低下が発生する条件や性能低下率は改善されてきているが、現時点でも上記の範囲の条件は存在するので、デバイスメモリへの不連続アクセスとなる場合は大きな性能低下が起きる。

2.3.2 デバイスメモリ容量の少なさ

現時点ではキャッシュベース CPU を用いたシステムでは主記憶容量が 512GB 程度まで拡張できる道があるが、GPU のデバイスメモリは最新のものでも 6GB のものが出たばかりである。通信と演算をオーバーラップさせて十分に通信遅延を隠蔽できるアプリケーションの場合は GPU を並列で用いることでこの制約を回避できる。しかし、この程度のメモリ容量では並列 GPU にすると細粒度で不規則な通信が発生してしまう場合もあり、デバイスメモリ容量の限界を拡張可能にする方法が望まれる。

2.3.3 オンチップメモリ容量の少なさ

ISCA2010 で Intel(R)*が発表した研究[17]では、SpMV(疎行列ベクトル積)計算において GPU(Nvidia GTX280)は Intel(R) Core(TM) i7 と比較して 2.5 倍の演算あたりメモリバンド幅を消費した理由として、列インデックスが Intel(R)Core(TM) i7 は半分がキャッシュに載るのに対して、GPU ではオンチップメモリに十分に載り切らない点を指摘している。キャッシュや共有メモリなどの GPU 上のオンチップメモリに再利用性のあるデータを載せきれない場合はデバイスメモリアクセスが多くなるため性能低下の原因となる。

2.3.4 PCI express ボトルネック

GPU のデバイスメモリ容量では十分でないアプリケーションでは、ホスト上の主記憶とデバイスメモリ間の転送や、ホスト上の主記憶を経由したノード間通信が必要になる。この場合、PCI express のバンド幅またはそこに接続されたネットワークのバンド幅がボトルネックとなる。特にバースト長が小さなデータを PCI express やネットワークに通す場合の実効バンド幅は、バースト長が大きなデータよりも大幅に低下する。

3. メモリシステムの最近の動向

本章では前章で記載した不連続アクセスに伴う課題の解決に関連し、考慮しておきたいメモリシステムの最近の動向について、4 つの観点から論じる。

---

* Intel 、Intel Core は、米国およびその他の国における Intel Corporation の商標です。

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

3.1 オンボードバッファ

クラウドや CPU の仮想化の普及とサーバーの消費電力抑制の要求の高まりによって、サーバーPC へのメモリ容量への要求が年々高まってきている。その結果、Intel(R) の Scalable Memory Buffer(SMB)などのように CPU チップあたりのメモリ容量の拡張性を高めるオンボードバッファ製品が市販されるようになってきている。SMB の現行製品は 2 チャネルの DDR と 4 本の DIMM を制御できる。SMB とホストとは Scalable Memory Interface(SMI)で接続される。SMI はメモリ用のインタコネクトであるので、PCI express を代表とする汎用インタフェースよりも高いバンド幅を継続的に実現していくものと考えられる。

3.2 三次元実装

三次元実装は現在 2 つの方向からメモリシステムの変革に採用されようとしている。1 つは CPU チップの上にメモリチップを複数積み重ねていく方向での採用である。CPU の主記憶へのバンド幅をある程度改善できるが、三次元実装可能なメモリ容量には限度があり、メモリ容量への要求が高いサーバー用途や HPC 用途では CPU パッケージ外部に主記憶が当面残らざるを得ないと考えられる。つまり、CPU パッケージ外部に拡張メモリが入る余地は残されていると考えられる。

もう 1 つはメモリパッケージ内でメモリチップを複数積み重ねていく方向での採用である。DDR4 は 4Gbps 程度の高周波を実現するためコントローラとメモリパッケージ間を Point-to-point 接続する．同一パッケージ内のチップ間をシリコン貫通ビアで接続することで反射の影響を回避し、高周波への対応と容量の増加を両立する。この傾向はメモリパッケージ内部のバンク数を増加する方向に作用する。後述するようにこの傾向は不連続メモリアクセスにとっては朗報である。

3.3 GPU のキャッシュへの依存度増加

GPU の一般用途への応用である GPGPU の普及のトレンドにおいて、Nvidia 社の Fermi アーキテクチャのように GPU プログラミングへの負担軽減の観点からキャッシュへの依存度が高まってきている。特に不連続アクセスを主体とするアプリケーションでは、キャッシュのラインサイズでのアクセスはバンド幅やキャッシュ容量を浪費する。よって、キャッシュ容量が CPU より早く底をつく GPU においては、不連続アクセス対策は今後も重要性が増していくトレンドにあると考えられる。

3.4 ポスト DRAM 候補の立ち上がり

近年、MRAM、ReRAM などの次世代メモリの研究開発が進展している。これらはキャパシタに保持できる電荷の限界性により微細化の限界が近づいてきた DRAM より高集積化する可能性を持っている。これらは NAND 型 EEPROM の対抗馬の不揮発メモリとして位置づけられることも多いが、特に MRAM は書換え回数の制限が緩く、DRAM よりアクセス時間やサイクル時間が 1 桁程度高速かつ低消費電力であるため、HPC 用途でも主記憶の置き換えとして利用されていく可能性がある。不連続アクセス対策というコンテクストからはサイクルタイムの減少が MRAM の特徴として価値がある。

4. Gather 機能付き拡張メモリ

4.1 基本アーキテクチャ

前章での課題の解決策として，DIMMnet-2 と同様の連続化ハードウェア（分散/収集機構）を COTS プロセッサのコアから見て内部ネットワークよりメモリに近い場所に追加することを提案する．提案方式の基本コンセプトを図 1 に示す．

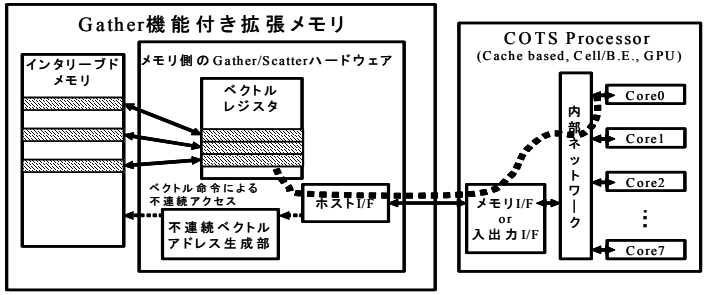


図 1　　提案する基本アーキテクチャ

表 1 に DIMMnet-2 の主なベクトル型アクセスコマンドを示す．このうち，等間隔ロード/ストア，リストロード/ストアのコマンドが不連続アクセスの連続化を実行するものである．ロード系が外部メモリから一種のベクトルレジスタである Prefetch Window への収集(Gather)処理を行い，ストア系が一種のベクトルレジスタである Write Window からの外部メモリへの分散(Scatter)処理を行なう．

表 1　　DIMMnet-2 の主なベクトル型アクセスコマンド

Load	Burst	VL
	Strided	VLS
	Indexed	VLI
Store	Burst	VS
	Strided	VSS
	Indexed	VSI

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

4.2 不連続アクセス向けメモリ構成

上記の基本アーキテクチャに基づく Scatter/Gather を効率的に行うには大容量メモリに対する不連続アクセスのスループットが高くなるようにする必要がある。本節では不連続アクセスのスループットを向上させる際の 4 つのポリシーについて述べる。

4.2.1 狭いビット幅のチャネル

DDR 系のメモリではプリフェッチのビット数だけのバースト長で常時アクセスすることで上記の差を吸収する。そのバースト長は DDR の場合 2、DDR2 の場合 4、DDR3 の場合 8 に設定されており、世代が進むごとにこれを長くすることで高い転送周波数に対応してきた。キャッシュのラインサイズ単位でのアクセスの場合、上記のバースト長は問題にならないが、不連続アクセスの場合には 4 サイクル分あるいは 8 サイクル分のデータのうち 1 サイクル分しか使わないという状況になりかねない。このため､不連続アクセスに強いメモリシステムは狭いビット幅のチャネル(例えば 8bit 幅）として、64bit 幅のメモリバスを採用している場合に捨てられてしまう転送サイクルを有効活用することが望ましい。

4.2.2 深いインタリーブ

大容量メモリチップのサイクルタイムと演算パイプラインのサイクルタイムの間には昔から 1 桁以上の差がある。このため、Cray-1 以来のベクトル型スーパーコンピュータの主記憶は、多数のメモリバンクを設けて、アドレスの下位部分をバンクを切り替えるためのビットにマッピングすることで並列動作させ、演算パイプラインへのデータ供給スループットを高めている。DDR2 あるいは DDR3 の DRAM にはメモリチップ内に 8 バンク存在し､同一チャネル上にそれらを 2 組動作させることができる。そのようなメモリチャネルを複数並列動作させることで現在は比較的少ないハードウェア量で深いインターリーブドメモリを構成できるようになってきている。特定のバンクへのアクセス集中（バンクコンフリクト）が回避できる場合、インターリーブドメモリは不連続アクセスに対しても高いスループットを実現できる。

4.2.3 Open ページポリシー

行アドレスを毎回のアクセスの際に DRAM に供給するのではなく、バンクを可能な限り Open としておくポリシーでの制御を行うことで、共通の行アドレスを持つアドレス群へのアクセスにおいては行アドレスの入力を省略できるとともに通常 2 サイクルおきに列アドレスのみ異なるデータをアクセスすることができる。

4.2.4 低サイクルタイム型メモリの活用

アプリケーションの性質上、バンクコンフリクトが回避できない場合には、上述のポリシーを全て導入したとしても大幅なスループット低下が発生する可能性がある。例えば 8 バイト単位でバンクが切り替わるマッピングがされたバンクが 64 本あるインタリーブドメモリにおいて、512×4K バイトおきに等間隔メモリアクセスをすると、常に 1 本のバンクにアクセスが集中しつつ、毎回行アドレスが変わることになる。その場合は上述の Open ポリシーも効かない。この場合はスループットが激減してしまう。しかし、MRAM 等の DRAM より一桁程度の短いサイクルタイムを有する大容量メモリを用いることで、この場合の性能低下もある程度抑制することができると考えられる。

4.3 ホストインタフェース

本節では提案拡張メモリのホストインタフェースの候補について述べる。

4.3.1 DIMM インタフェース

我々の研究グループが 2000 年以来世界に先駆けて用いてきた方式であり、DIMMnet-1[10], DIMMnet-2[1], DIMMnet-3[2]がこの方式を取る。DIMM インタフェースの Gather 機能付きメモリとして機能させたのは DIMMnet-2 が最初であり、DDR 型 DRAM ソケットに装着され、実際に動作させた。DIMMnet-3 では DDR2 ソケットに対応したとともに、ホストインタフェース部と機能メモリ部を別基板としているため、PCI express などの他のインタフェースにも対応しやすくなっている。

4.3.2 PCI express

PCI express は現在もっとも一般的に用いられている標準 I/O インタフェースである。レーン数は 16 のものが主に GPU 向けに、8 のものがネットワークインタフェースなどのその他のサーバーの I/O 製品向けに使われる。特に GPU と拡張メモリの接続を行う場合は PCI express を利用するのが適切と考えられる。ただし、現時点で得られる最高速のバンド幅が x16 Gen.2 で 8GB/s に留まる。よって、この 2 倍のバンド幅を実現するには Gen.3 の普及を待つ必要がある。

4.3.3 SMI

Intel(R)の Scalable Memory Buffer(SMB)[12]のインタフェースが SMI であり、サーバーのメモリと Intel(R)製 CPU 間のインタフェースとして有望である。バンド幅は 25GB/s であり PCI express x16 Gen.2 の 2 倍以上のバンド幅を有する。ただし、拡張メモリシステムのインタフェースを SMI にした場合は､マザーボードまたはメモリボードは専用のものを開発する必要がある。

4.3.4 Hypertransport

Hypertransport は AMD Opteron の CPU 間インタフェースである。拡張メモリのホストインタフェースを Hypertransport とした場合は、マザーボードは市販品で、CPU ソケットに拡張メモリを装着するという実装形態が可能になると考えられる。

4.3.5 QPI

QPI は Intel(R)版の Hypertransport と考えることができるので、拡張メモリのホストインタフェースを QPI とした場合は、マザーボードは市販品で、CPU ソケットに拡張メモリを装着するという実装形態が可能になると考えられる。

4.3.6 FlexIO

FlexIO は Cell/B.E.のフロントサイドバスとして用いられているインタフェースで

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

ある。バンド幅は最大 76.8GB/s と群を抜いて高い。このバンド幅は主記憶バンド幅より大きい。テレビなどの量産型民生機器への搭載を考慮すると、FlexIO で接続された専用コンパニオンチップに本拡張メモリの機能を入れるのが妥当と考えられる。

5. 想定されるアプリケーション

5.1 HPC 向けクラスタのメモリシステム強化

HPC 向け計算機においてはランダムに近いインデックスによる大規模な配列への間接参照を如何に高速にするかが重要である。多くのアプリケーションが最終的に連立一次方程式の反復解法による求解に帰着されるためである。この種のアクセスはキャッシュに載っている範囲の小さな配列の場合は COTS の CPU 利用でも問題にならないが、載り切らないほど巨大な問題において、実行時間の大半を占める可能性が高い。提案した拡張メモリは、CPU あるいは GPU からのこの種のメモリアクセスのバンド幅浪費や、ノード間の細粒度通信を抑制する付加装置として機能する。

5.2 ボリュームレンダリング

GPU のデバイスメモリに入りきらないほど大容量なデータの可視化においては、ボリュームレンダリングの際のメモリアクセスが上記と類似した性質を有することになる。MRI や CT などの高解像度な三次元医用データの可視化においてもボリュームレンダリングが行われる。現状の解像度や撮影範囲では GPU のデバイスメモリの容量で対応可能な範囲であると考えられるが、今後、解像度や撮影範囲が拡大してくると HPC 向けの可視化と同様になる。

5.3 顔画像認識

空港におけるテロリスト対策として、空港内の大量のカメラで撮影した動画から顔画像を認識し、ブラックリストの顔画像データベースからそれに近いものをリアルタイムに検索して提示するシステムの社会的ニーズが高まってきている。顔画像は多数の要素からなる特徴ベクトルで表現され、入力画像における特徴的な要素での絞込みを行って検索時間を短縮することが望ましい。その際、メモリ上のデータベースへの等間隔アクセスを主体にした大量の不連続アクセスが発生すると考えられる。1 枚の顔画像の特徴ベクトルは数 KB 程度の大きさであるため、アクセス間のストライドが大きく、要素のサイズは 4 バイトであるためキャッシュアーキテクチャでは効率的ではない。提案した Gather 機能付き拡張メモリは、CPU あるいは GPU からのこの種のメモリアクセスのバンド幅浪費を抑制できる。

5.4 動画検索

YouTube などの動画サイトでの動画の検索やレコメンドのニーズだけでなく、Cell REGZA のように複数の TV チューナから大容量のハードディスクに常時多チャンネル録画を行い、過去に録画した動画の中から、見たい動画を発掘するスタイルの動画サーバーが一般家庭向けに市販されている。ハードディスクのビット単価の低下の勢いは現在でも留まるところをしらず、動画を検索するニーズとその負荷が年々高まっていくと考えられる。動画は様々な観点から検索用の特徴ベクトルに圧縮変換されて保存されると考えられ、上記の顔画像認識と同様の処理の高速化が望まれる。

6. 性能評価

6.1 評価方法

6.1.1 シミュレータ

(1) ベースとして用いたシミュレータ

本研究の性能評価に際して、Maryland 大学の DRAMsim2[15]をシミュレータのベースとして用いた。DRAMsim2 はアドレストレースファイルを入力として動作する。旧バージョンの DRAMsim[13][14]はシミュレータ内で CPU とメモリシステムが一体になっている。評価対象は提案拡張メモリを装着する相手によってホスト CPU は異なる上、提案拡張メモリ内にあるベクトル型のアドレス生成部と DRAMsim で用意された CPU は、スループットが異なると思われる。さらに、メモリシステム構成や、メモリ種類の追加変更のしやすさも考慮して、本研究では DRAMsim2 を選択した。

(2) 改造内容

現在、DRAMsim2 は開発途上にあり、本研究においてはいくつかの不足分を独自に追加改造して用いた。その改造内容を以下に示す。

1) チャネル数を可変にした
2) アドレスマッピングをインタリーブに対応させた
3) トランザクション投入部多重度を可変にした

6.1.2 ワークロード

(1) 等間隔アクセス

Wisconsin ベンチマークのいくつかのクエリにおいて共通に現れる等間隔アクセスのアドレストレースを作成した。データサイズは 4 バイト、ストライドは 60 バイト、オフセットは 24 バイト、アクセス回数は 4096 回の条件でトレースを作成した。さらにバンクコンフリクトの影響を測定するため、ストライドを 64,128,256 に変化させた条件でもトレースを作成した。

(2) ランダムアクセス

ランダムアクセスのアドレストレースを作成した。データサイズは 4 バイトおよび 8 バイト、アドレスの範囲は 0～4G で 4096 個の乱数を生成し、トレースを作成した。

(3) 疎行列ベクトル積のベクトルへのアクセス

疎行列ベクトル積において提案拡張メモリにオフロードすることを想定し、その際のベクトルへの間接アクセスのトレースを University of Florida Sparse Matrix

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

Collection[16]から比較的小規模な疎行列によって作成した。使用した行列は Na5 である。Na5 の場合でも非零要素数が約 16 万あるので、等間隔アクセスやランダムアクセスの評価の 4096 回の場合より約 40 倍シミュレーション時間がかかる。

6.1.3 評価対象のメモリシステム

(1) DRAM チップ

評価に用いた DRAM チップのパラメータは DRAMsim2 に添付されている DDR2_micron_32M_8B_x4_sg25E.ini と DDR3_micron_64M_8B_x4_sg15.ini の二種類である。その主なパラメータ値を表 2 に示す。

表 2　評価に用いた DRAM チップの主なパラメータ

DRAM チップパラメータ	DDR2	DDR3
容量	2Gbit	2Gbit
バンク数	8	8
行数	16384	32768
列数	2048	2048
tCK(転送サイクルタイム)	2.5ns	1.5ns
CL(CAS レイテンシ)	5	10
BL(バースト長)	4	8
tRAS(RAS レイテンシ)	18	24
tRCD(RAS to CAS レイテンシ)	5	10
tCCD(CAS to CAS レイテンシ)	2	4

(2)システム構成

評価したメモリシステムのシステム構成パラメータを表 3 に示す。

表 3　評価したメモリシステムのシステム構成パラメータ

システム構成パラメータ	値
チャネル数	1,2,4,8,16
1 サイクルで発生する Transaction 数	1,2,4,8
チャネルあたりのビット幅	8,16,32,64,128
ランク数	2,4

チャネルあたりビット幅は 8,16,32,64 の場合を測定する。狭いほど 4 バイトや 8 バイトのアクセスの際の固定バーストに伴う無駄サイクルの影響が減少する。一方、チャネルあたりビット幅が 64 の場合は市販の DIMM を用いることができるようになる。それ以下の構成ではメモリチップを無駄なく使うには、ビット幅の狭い特殊仕様の DIMM を作るか、コントローラと同一基板上にメモリチップを実装する必要が生じる。

チャネル本数は 1～16 まで変化させた。データラインが 8bit 幅のチャネルを 16 本実装するコントローラは全体として 128bit 分のデータ用ピンを消費する。このピン数は DIMMnet-3 の半分、DIMMnet-2 や現在市販されている SMB の仕様と同等である。

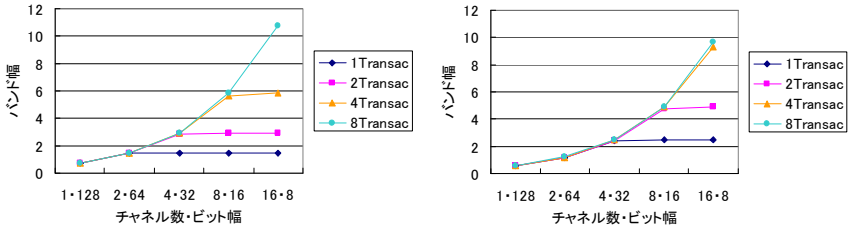
(3) アドレスビットマッピング

インタリーブ構成のアドレスマッピングはアドレスの下位から固定バースト分、チャネル、バンク、ランクの順に割り当てた。これにより、固定バースト長単位でチャネルが切り替わり複数チャネルの並列動作を促進した。ランクの切り替えには 1 サイクル待ちが入るので、一番上位に割り当て、ペナルティの発生頻度を抑制している。

6.2 結果

6.2.1 等間隔アクセスバンド幅

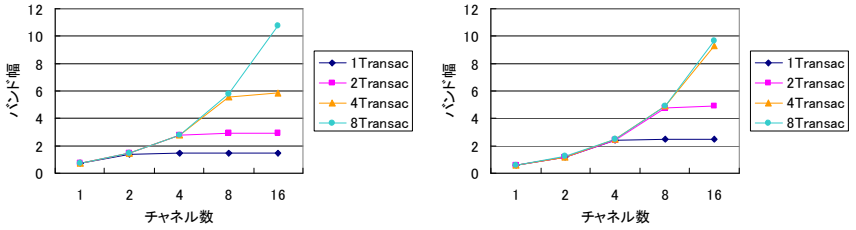
図 2 はデータに用いるビット幅の合計を 128 ビットに固定して、チャネルあたりのビット幅とチャネル数を変えて等間隔アクセスの実効バンド幅を示したものである。



(a) DDR2　(b) DDR3

図 2　データ線総数を 128 ビットに固定した時の等間隔アクセスの実効バンド幅

以上からビット幅が狭いチャネルの性能が良いので、以下の全ての実験は、チャネルあたりのビット幅は 8bit とする。図 3 はチャネル数を変えて等間隔アクセスに対する実効バンド幅を示したものである。

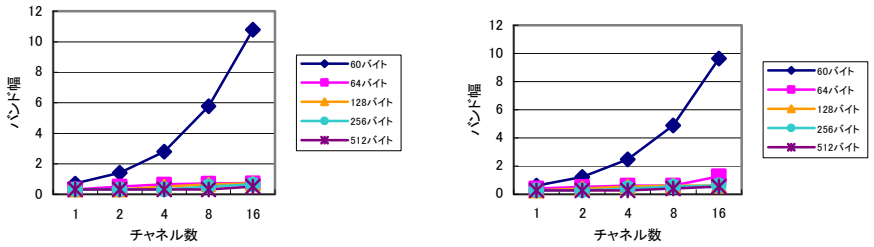


(a) DDR2　(b) DDR3

図 3　等間隔アクセスの実効バンド幅へのチャネル数の影響

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

図 4 はバンク数の倍数になるようストライドを変化させた場合の等間隔アクセスに対する実効バンド幅を示したものである。

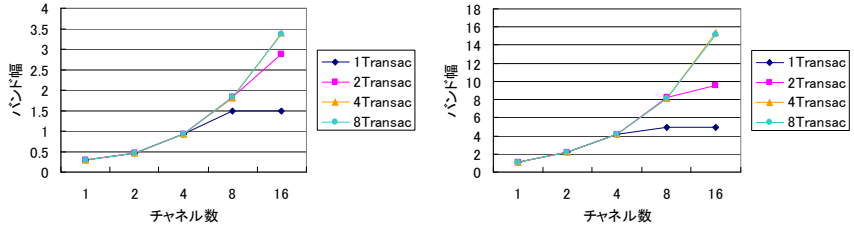


(a) DDR2　　(b) DDR3

図 4　　等間隔アクセスの実効バンド幅へのバンクコンフリクトの影響

6.2.2 ランダムアクセスバンド幅

図 5 は 8 バイトデータのランダムアクセスに対する実効バンド幅を示したものである。

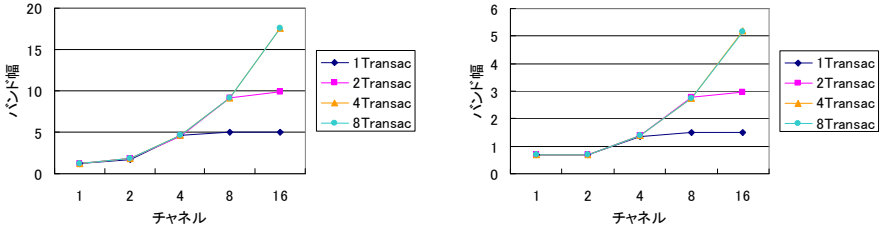


(a) DDR2　　(b) DDR3

図 5　8 バイトデータのランダムアクセスの実効バンド幅

6.2.3 疎行列ベクトル積アクセスバンド幅

図 6 は 8 バイトデータの疎行列ベクトル積実行時のベクトルアクセスに対する実効バンド幅を示したものである。使用した疎行列はフロリダ大学疎行列コレクションにある Na5 である。



(a) DDR2　　(b) DDR3

図 6　　疎行列ベクトル積実行時のベクトルアクセスの実効バンド幅

7. 考察

7.1 現時点で実現性のある実効バンド幅

本節では現時点で実現性のある実効バンド幅と想定しているアプリケーションとの関係について考察する。表 4 に、今回の測定範囲の各ワークロードとチャネル数=8,16 における DDR3 を用いた最大実効バンド幅を示す。

表 4　　DDR3 DRAM を用いた最大実効バンド幅

ワークロード	データサイズ	8bit 幅 8 チャネル	8bit 幅 16 チャネル
等間隔(60B 間隔)	4 バイト	5.76GB/s	10.78GB/s
ランダム	8 バイト	7.74GB/s	16.26GB/s
疎行列(Na5)	8 バイト	9.20GB/s	17.55GB/s

上記の中で最もハードウェア量が多い 8bit 幅 16 チャネルのメモリコントローラは、ピン数の観点からは一般的な 64bit 幅 2 チャンネルのメモリコントローラと比べアドレス信号と制御信号で 20 本×14＝280 本程度の増加で済むため、1000 ピン以上の FPGA が利用可能な現時点では十分に実装可能な範囲にある。よって表 4 に記載のバンド幅は全て現時点で実現性のある実効バンド幅と考えられる。

Cell/B.E.と組み合わせて Wisconsin ベンチマークを実行する場合においては、筆者らの先行研究において Cell/B.E.の 1SPE あたり 4 バイトの 60 バイト間隔のデータを 512 バイト分(128 回分)を 2 μ 秒以下で転送できれば性能低下がほとんどないことが実験確認されている。つまり 1SPE あたり 512/2000＝0.5GB/s であり、7SPE 全てを用いたとしても 3.5GB/s が得られれば良いことになる。今回のシミュレーション結果では 8bit 幅 8 チャネル以上で実効バンド幅 5GB/s 程度が実現できており、先行研究の結果どおりの性能を達成できることが明らかになった。

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

GPU と組み合わせて疎行列ベクトル積を実行する場合においては、筆者らの先行研究において PCI express x16 Gen.2 のバンド幅(ピーク 8GB/s、実効 6GB/s)がボトルネックであり、Gather 機能付きメモリ側はそれ以上の実効バンド幅を出していれば新たなボトルネックにはならないことが判っている。今回のランダムアクセスや疎行列ベクトル積のベクトルアクセスのシミュレーション結果では 8bit 幅 8 チャネル以上で実効バンド幅 8GB/s 以上が実現できており、先行研究の結果どおりの性能を達成できることが明らかになった。また 8bit 幅 16 チャネルの DDR3 の一部の構成において 15GB/s 以上出せているので、将来 PCI express x16 Gen.3 になった場合でも DDR3 ベースで対応できると考えられる。

なお、疎行列の性質を正確に反映するために Florida 大学の行列コレクションを用いたが、測定した範囲ではランダムな場合の性能より若干高めではあるが、概ねランダムでスループットを近似できることが確認できた。これは提案システムの性能が行列の非零要素の配置に鈍感であることを支持する一つの証拠といえる。このシミュレーションは 1 個の行列のメモリ側 1 構成分の実行時間が 1 日がかりであるため、あまり多くのサンプルを現段階では取れていない。よって、提案システムの性能が行列の非零要素の配置に鈍感であることを十分に確認できたとまでは言えず、その確認は今後の課題である。

7.2 チャネルあたりビット幅

本節ではチャネルあたりのビット幅が実効バンド幅に与える影響を考察する。図より、チャネルあたりのビット幅が狭いほど実効バンド幅が高くなっていることがわかる。4 バイトや 8 バイトのアクセスの際の固定バーストに伴う無駄サイクルの影響が減少する結果と考えられる。

Convey 社の HC-1 の SG-DIMM は通常の 64bit 幅 DIMM が装着されるスロットに装着されるが、「quad words (8bytes)で物理メモリをアクセスすることを許容する」と文献に記載されていることから、16bit(ECC を含めると 18bit)のワードを 4 サイクルで転送していると考えられる。このことは上記の測定結果から得られたチャネルあたりのビット幅が狭いほど良いという結果と合致する。HC-1 では 16bit 幅であるので 4 バイトのデータ転送においては無駄なサイクルが発生すると考えられる。これは ECC を導入しているため許容している無駄と考えられる。

一方、チャネルあたりのビット幅を 9bit と狭くしている XDR DRAM は既にこの方向性を取り入れており、本来は不連続アクセスに強いアーキテクチャである。XDR DRAM は Cell/B.E.の主記憶として使われているが、前述のとおり現状では上記の効果を消し去ってしまう様々な問題が Cell/B.E.側にある。それを回避する本提案と XDR DRAM を組合せれば、DDR3 より高い周波数で同様の効果が得られ、性能が向上すると考えられる。XDR DRAM を用いる場合の評価は今後の課題である。

7.3 インタリーブ多重度

本節ではインタリーブ多重度が実効バンド幅に与える影響を考察する。

入力トランザクションが十分に供給されている状態では、インタリーブの多重度をチャネルが増える方向に増やした場合は、バンド幅が多重度に応じて増加する。ところが、ランクが増える方向に増やした場合は、ほとんど横ばいか、若干性能低下する。本測定のようにチップ内のバンク数が 8 あるような状況では、インタリーブ多重度が増えることでバンクコンフリクトが減少した結果として得られるメリットはランク切換えオーバーヘッドが 1 サイクル増えるというペナルティとほぼ同程度と考えられる。それらが相殺しあった結果、ランク数の増加はあまり実効バンド幅に貢献しなかったと考えられる。

以上から、同じ個数のメモリチップを用いて実効バンド幅を向上させる場合は、ランクではなくチャネルを増やすほうが効果的であるといえる。ただし、ランクを増やす方向に比べ、チャネルを増やす方向はコントローラのピン数やロジック部を増やす方向に直接的に働くので、それらがコストや製造上の無視できない制約になる場合はランクを増やすことでメモリ容量は増やすことができる。

7.4 ページ制御ポリシー

本節では Open ポリシーと Close ポリシーの違いが実効バンド幅に与える影響を考察する。等間隔アクセスの場合 Open ポリシーの性能は Close ポリシーの性能より常に若干良かったので Open を用いる方が良い。ただし、その性能差は 3～16%に過ぎずあまり大きなものではなかった。ランダムアクセスになると、同じ行アドレスが続く確率が低くなるため、さらにその差は小さくなるものと考えられる。

7.5 トランザクション投入部多重度

本節ではトランザクション投入部多重度が実効バンド幅に与える影響を考察する。

トランザクション投入部多重度とは、1 サイクルで同時にメモリシステムに投入できるメモリアクセストランザクション要求の個数のことである。4 バイトあるいは 8 バイトのアクセスに分解された一つ一つのメモリアクセス要求は FIFO にキューイングされるが、それを 1 サイクルで読み出す多重度に相当する。

生産者にあたるトランザクション投入部多重度が小さいまま、その消費者に当たるインタリーブ多重度を上げても、トランザクションの生産不足によってスループットが上がらなくなる。特に DDR2 の場合は周波数が低くなるのでトランザクション投入部多重度不足の影響が DDR3 より強く出る傾向がある。

等間隔アクセス、ランダムアクセスの両方の場合で、DDR3 DRAM を 8 チャネル動作させるためには最低でも 2 並列、16 チャネル動作させるためには最低でも 4 並列のトランザクション投入部多重度を確保しないとトランザクションの生産不足によってスループットが上がらなくなる。

一方、トランザクション生産部の能力の観点からは、等間隔アクセスの場合は SIMD

Vol.2010-ARC-192 No.16
Vol.2010-HPC-128 No.16
2010/12/17

並列度 2 または 4 の整数加算ができれば上記を実現でき、比較的容易である。これに対してランダムや疎行列ベクトル積のベクトルアクセスのように間接アクセスの場合は、インデックス配列を 2 または 4 並列で読み出してくるバンド幅が消費される。これはインデックスが 32bit だとしても 8 バイトまたは 16 バイト幅での連続アクセスになるので、これをインタリーブドメモリ側から取り込む場合は相応のバンド幅低下を見込む必要がある。具体的には今回の評価の DDR3 は 1.5ns サイクルで、8 バイトを供給する場合は 5.3GB/s、16 バイトを供給する場合は 10.6GB/s が消費される。ホスト側からこれを供給する場合はホストインタフェースの書き込み側のバンド幅が溢れない限り、今回の評価でのバンド幅が得られることになる。つまり、8 チャネル動作では問題ないものの、16 チャネル動作では PCI express x16 Gen.2 のバンド幅限界を必要なバンド幅が超える。よって、16 チャネル動作の場合には、バンド幅の広いインタリーブドメモリ側からインデックスも供給するのが妥当であると考えられる。

7.6 バンクコンフリクト

本節ではバンクコンフリクトが実効バンド幅に与える影響を考察する。図 4 よりアクセスストライドを 60 バイトではなく、64 バイトにすると顕著にバンド幅が低下することがわかる。ただし、稼動チャネルの削減と同一行アドレスでのアクセス率向上という相反する 2 つの効果によって、バンクコンフリクトによる性能低下は同一行アドレスでのアクセスのバンド幅(チャネルのピークバンド幅の半分程度)を限度に留まると考えられる。

8. 関連研究

8.1 DIMMnet

キャッシュベースの CPU や Cell/B.E.や GPU と組み合わせた際の不連続アクセスの高速化に関する従来研究には筆者等が行なった DIMMnet-2 や DIMMnet-3 を用いた研究[1]-[9]がある．NAS CG やボリュームレンダリング上のリストアクセスの高速化や, Wisconsin ベンチマーク上の等間隔アクセスの高速化等が評価された. しかし, これらは全て DRAM のメモリシステム側のスループットが十分であるということが前提での評価か､十分に最適化されていない実機での限られたケースでの評価であり、本研究のような DRAM のメモリシステム側のスループットの多様な構成に対するシミュレーションに基づくものではない。本研究によって現在入手可能な DRAM ベースでも十分な不連続アクセススループットを確保でき、DIMMnet 関連の先行研究の正当性を担保する仮定が成り立つことが示された。

8.2 Intel(R) Larrabee

Intel(R) Larrabee[18]はプロセッサ側に Scatter/Gather を行うベクトル命令を持つメニーコアプロセッサである。本提案に似た構成にも見えるが、Scatter/Gather を行う部分がオンチップネットワークよりプロセッサ側である。このため、プロセッサ側に実装されているキャッシュ上に Scatter/Gather を行うアドレスのラインがあれば問題ないが、主記憶側にある場合はキャッシュライン単位での 4 バイトや 8 バイトのデータをアクセスしてしまうため､本提案で解決している問題が解決されていない。

8.3 Cray XMT

Cray XMT[19]は Cray XT 系マシンのマザーボードやシャーシを用い､CPU として AMD 社の Opteron の代わりに専用のマルチスレッド型 CPU を Opteron 用ソケットに装着した製品である。マルチスレッド型 CPU は近年の GPU のように 128 個のスレッドの高速切換えによって長いメモリアクセス遅延を隠蔽できる。よって多数のリモートメモリで構築される多バンクかつ大きな共有メモリに対して、専用プロセッサが命令で発生する細粒度でランダムなアクセスに従来システムより耐性があると考えられる。筆者が知る限りでは本研究のような Gather バンド幅の評価は明らかになっていない。しかし、このアプローチは本提案と異なり CPU が専用になってしまうため、高コストかつ演算性能が陳腐になりやすい。

8.4 Impulse

Impulse[20]はホストのメモリコントローラに Scatter/Gather 機能を入れる提案である。メモリコントローラを含めたシミュレーションを実施しているが、古い SDR DRAM をベースにした評価になっている。また、本提案と異なり、Impulse は既存システムに後付けできるようにはなっていない。

8.5 SDT

SDT[21]は主記憶データベースの加速を目的に FIFO をレジスタとして持つ特殊な CPU のメモリコントローラに等間隔の Gather 機能を入れる提案である。SDT は本提案と異なり、Impulse は既存システムに後付けできるようにはなっていない。SDT の研究ではメモリコントローラを含むシミュレーションを行っている。しかし、古い SDR DRAM をベースにした評価になっている。さらに、多バンクインターリーブ構成の高スループット指向のメモリシステムに関する検討はされていない。

8.6 Convey HC-1

Convey HC-1[22]は FPGA によってベクトルプロセッサやインタリーブドメモリを構築しており、ベクトル命令での Scatter/Gather も可能である。8 個の FPGA によるメモリコントローラを用いて全体として 64bit 幅の 16 本のメモリチャネルを実現している。筆者が知る限りでは本研究のような Gather バンド幅の評価は明らかになっていない。

通常の DIMM だけでなく、SG-DIMM という Scatter/Gather を加速できる DIMM インタフェースの専用メモリモジュールを装着できる。SG-DIMM は 64 バイト単位ではなく 8 バイト単位のアクセスを可能にすることで Scatter/Gather の際の実効バンド幅を向上させるが、Scatter/Gather 機能自体は提案方式のようなメモリ側ではなく

ベクトルプロセッサ側にあると考えられる。浮動小数演算自体も FPGA ベースのベクトルプロセッサで行わせるアーキテクチャであるため、日進月歩で進化する GPU と比較すると絶対性能や価格性能比で見劣りが避けられない。これに対して本提案は GPU 等と組み合わせて使うアーキテクチャである。

9. おわりに

筆者らが行った先行研究[1]-[9]では提案拡張メモリシステム側のスループットを十分にできることが前提の評価であった。本報告ではその前提が成り立つか否かについて、メモリシステム側のスループットの実現性をシミュレーション評価した。

Cell/B.E.と組み合わせて Wisconsin ベンチマークから抽出したクエリを処理する場合の先行研究では、7SPE 全てを用いたとしても 4 バイトデータの等間隔アクセスで 3.5GB/s が得られれば良いことがわかっていた。今回のシミュレーション結果では DDR3・8bit 幅・8 チャネル以上で実効バンド幅 5GB/s 程度が実現できており、先行研究の結果どおりの性能を達成できることが明らかになった。

GPU と組み合わせて疎行列ベクトル積を実行する場合の先行研究では、8 バイトデータの間接アクセスで 8GB/s が得られれば良いことがわかっていた。今回のシミュレーション結果で DDR3・8bit 幅・8 チャネル以上で実効バンド幅 8GB/s 以上が実現できており、先行研究の結果どおりの性能を達成できることが明らかになった。

今後の課題としては、より多くの疎行列での性能確認と、今回評価に用いた DDR2 や DDR3 より高性能が期待できる XDR-DRAM、DDR4 DRAM、MRAM 等のメモリを用いた場合の評価がある。

謝辞 本研究の一部(DIMMnet-3 の開発)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.

参考文献

1) N. Tanabe, M. Nakatake, H. Hakozaki, Y. Dohi, H. Nakajo, H. Amano : "A New Memory Module for COTS-Based Personal Supercomputing", 7th International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA2004), pp.40-48 Jan. 2004

2) N. Tanabe, H. Nakajo : " An Enhancer of Memory and Network for Cluster and Its Applications", IEEE PDCAT'08, pp.99-106, Dec. 2008.

3) N. Tanabe, H. Nakajo : " High Performance Computing and Database Processing with COTS and Extended Memory Modules", HPC Asia2009 (Best paper award), Mar. 2009.

4) N. Tanabe, M. Sasaki, H. Nakajo, M. Takata, K. Joe : "The Architecture of Visualization System using Memory with Memory-side Gathering and CPUs with DMA-type Memory Accessing", International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'09) , pp. 427-433, Jul. 2009.

5) N. Tanabe, H. Hakozaki, Y. Dohi, Z. Luo, H. Nakajo : " An enhancer of memory and network for applications with large-capacity data and non-continuous data accessing", The Journal of Supercomputing, Vol. 51, No. 3, pp. 279-309, Mar. 2010.

6) N. Tanabe, T. Tsukamoto, A. Ohta, H. Nakajo : " Efficiency Improvement for Discontinuous Accesses of Cell/B.E. Using Hardwired Scatter/Gather on Memory-side ", IEEE ICCEE'10, Nov. 2010

7) 塚本, 田邊, 大田, 中條 : “ベクトルアクセス機構を有するメモリモジュールによる不連続な DMA の効率化”, 情報処理学会 HPC 研究会, Mar. 2010.

8) N. Tanabe, Y. Ogawa, M. Takata, K. Joe : " Scaleable Sparse Matrix-Vector Multiplication with Functional Memory and GPUs", Euromicro PDP’2011 (Accepted) , Feb.2011

9) 小川, 田邊, 高田, 城 : “機能メモリと GPU の PCI express 接続によるヘテロ環境における超大規模疎行列ベクトル積の性能予測”, SWoPP’10 HPC, Aug. 2010.

10) N. Tanabe, J. Yamamoto, H. Nishi, T. Kudoh, Y. Hamada, H. Nakajo, H. Amano : "MEMOnet : Network interface plugged into a memory slot", IEEE International Conference on Cluster Computing (CLUSTER2000), pp.17-26 , Nov. 2000

11) M. Kistler, M. Perrone, F. Petrini : "Cell Multiprocessor Communication Network: Built for Speed", IEEE MICRO Vol.26, No.3, pp.10-23 (2006.5)

12) Intel(R) : "Intel(R) 7500 Scaleable Memory Buffer Datasheet", Mar. 2010. http://www.intel.com/Assets/PDF/datasheet/322824.pdf

13) D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, B. Jacob : "DRAMsim: a memory system simulator", SIGARCH Computer Architecture News Vol.33, No.4, pp.100-107, Sep.2005

14) B. Jacob : "DRAMsim: A Detailed Memory-System Simulation Framework", http://www.ece.umd.edu/dramsim/v1/

15) B. Jacob : "DRAMSim2", http://www.ece.umd.edu/dramsim/

16) Tim Davis : " The University of Florida Sparse Matrix Collection", http://www.cise.ufl.edu/research/sparse/matrices/

17) Victor W. Lee et al. : "Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU", ISCA 2010

18) L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerman, R. Cavin, R. Espasa, E. Grochowski, T. Juan, P. Hanrahan : "Larrabee: A Many.Core x86 Architecture for Visual Computing", ACM Trans. Graph. 27, 3, Article 18, Aug. 2008.

19) Cray Inc. : "Introducing the Cray XMT Supercomputer", http://www.cray.com/Assets/PDF/products/xmt/CrayXMTOverviewWhitepaper.pdf

20) Carter, Hsieh, Stoller, Swanson, Zhang, Brunvand, Davis, Kuo, Kuramkote, Parker, Schaelicke and Tateyama : "Impulse : Building a Smarter Memory Controller", International Symposium on High Performance Computer Architecture (HPCA-5), pp.70-79 (Jan. 1999)

21) K. Tanaka, T. Fukawa : "Highly Functional Memory Architecture for Large Scale Data Application", 7th International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA2004), pp.109-118 (Jan. 2004)

22) T.M.Brewer : "Instruction Set Innovations for the Convey HC-1 Computer", IEEE Micro, Vol.30 No.2 pp.70-79, Mar. 2010.