Vol.2011-ARC-197 No.14
Vol.2011-HPC-132 No.14
2011/11/28

不規則型応用を加速するメモリアクセラレータ ー Exa FLOPS マシンの文脈から

田邊 昇[†] Boonyasitpichai Nuttapon[††] 中條 拓伯[††]
小郷絢子[†††] 高田雅美[†††] 城 和貴[†††]

Byte/FLOP を現在より下げざるを得ない Exa FLOPS 級マシンにおいて，国内では階層キャッシュが重要視されている．一方，不規則型応用は必要な Byte/FLOP が大きく，かつ，文科省の重点指定アプリの中でも大きな比率をしめており，日本の Exa F LOPS 級マシンの文脈において対応が重要である．本報告では Gather 機能を有するメモリシステムについて上記の文脈上で考察する。その判断材料として現状の GPU のキャッシュヒット率と，キャッシュを前段に併用した Gather 機能を有するメモリシステムについて疎行列ベクトル積の性能評価を行なった．その結果，キャッシュによる性能向上は限定的で，その容量の 10 倍程度のベクトルサイズまでで頭打ちとなった．不規則型応用に対して再利用性に高速化原理をおくキャッシュに頼りきることは危険であり，Gather 機構が重要であることが確認された．

Memory Accelerator for Irregular Applications - From the Context of Exa FLOPS Machine

Noboru Tanabe[†] Boonyasitpichai Nuttapon[††]
Hironori Nakajo[††] Junko Kogou[†††]
Masami Takata[†††] and Kaz uki Joe[†††]

In Japan, hierarchical cache is currently given high priority for memory system of Exa FLOPS machine whose Byte/FLOP ratio must be smaller than current supercomputers. On the other hand, irregular applications require higher Byte/FLOP ratio. Since these occupy a large part of MEXT-selection of focused applications, supporting irregular applications is important in the context of Japanese Exa FLOPS machine. In this report, a memory system with gather functions is reconsidered in the context of Exa F LOPS machine. We explored hit ratio of a current GPU's cache and performance of memory system with gather function and its preliminary stage cache for sparse matrix vector multiplication. As a result, performance gain by the additional cache is limited and disappeared for vector with about 10times higher size than the cache capacity. It is confirmed that the importance of gather function for irregular applications is higher than that of cache whose acceleration principle is based on reusability.

1. はじめに

現在，日本では文部科学省と国内のスーパーコンピュータセンターが音頭を取り，多様なシステム系研究者を巻き込んだ形で，2018 年頃のスーパーコンピュータとして Exa FLOPS 級の性能を有するシステムの検討が盛んに行われている．米国においては 2008 年頃から検討されており，その流れを受けて日本でも独自の検討が行われるようになった．そこでの電力制限やデバイス技術トレンドの分析から，Exa FLOPS 級のターゲットマシンの Byte/FLOP 値を現状より 1/5 程度に低くしないと製造できない見込みが明らかになってきた．その対策として主に国内では階層キャッシュが最重要視されており，ソフトウェアスタックもそれを前提にした研究開発ロードマップ作りが行われている．

最近ではアプリケーション作業部会も発足し，一足先に動き出したシステム開発検討側に対し，使う側の視点からの要望が出されるようになってきた．文部科学省の重点指定になっている 40 弱のアプリケーションのうち 4 割程度は Byte/FLOP 値が実効性能に直結するため，高い Byte/FLOP 値を要求していることも明らかになってきた．概ねそれらは巨大な疎行列係数の連立一次方程式を繰り返し解くことに帰着されているアプリケーションであると考えられる．ところが利用者側での試算では，メモリアクセスが不連続(間接参照)か否かを区別せず，同じ時間がかかるメモリアクセスとしてカウントしたものがほとんどであるとされている．実際には間接参照はキャッシュベースのシステムでは 10 倍のオーダーのバンド幅浪費が発生するため，上記は低めな Byte/FLOP 値要求になっていると考えられる．よって，そのバンド幅浪費効果も考慮した真の要求にキャッシュで応えきれるのか否かを評価することが急務である．

連立一次方程式求解や固有値計算の中で実行時間の大半を占めるのは疎行列ベクトル積であり，高い Byte/FLOP 値を要求する典型例である．近年では GPU の高バンド幅を用いたその高速化の研究も盛んに行われている．広いビット幅構成にした GDDR 型 DRAM による GPU のメモリシステムは，同タイミングで動作するスレッド群が発生するアクセス群のアドレスが連続になるアプリケーション（例えば構造格子系）には効率的に動作する。一方、不連続アクセスになるアプリケーション（例えば非構造格子系）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いて，GPU を含む様々なプロセッサで大幅な性能低下が発生するという問題がある。特に対象問題が大きくなり列ベクトルが GPU のキャッシュには載りきら

[†]株式会社 東芝
Toshiba corporation
[††]東京農工大学
Tokyo University of Agriculture and Technology
[†††]奈良女子大学
Nara women's university

Vol.2011-ARC-197 No.14
Vol.2011-HPC-132 No.14
2011/11/28

なくなると激しい性能低下が発生するようになると考えられる。

そのような問題を解決するために筆者らは先行研究[1]-[12]で Scatter/Gather 機能を有する拡張メモリシステムを提案した。文献[8]-[12]では疎行列ベクトル積においても評価を行ない、有効性を示してきた。

本研究では，何も足さない現状の GPU や，提案済みの Gather 機能付きメモリシステムの前段にキャッシュを併用したメモリアクセラレータにおいて，疎行列ベクトル積におけるキャッシュの効果を，より多くの疎行列に対して定量的に評価を行う．これにより Exa F LOPS マシンが対象にすると考えられる大規模な行列におけるキャッシュと Gather の効果の優劣を確認する．

以下、本報告では第 2 章で Exa FLOPS 級マシンのメモリシステムの課題について述べる．第 3 章では日米の Exa FLOPS 級マシンのメモリシステムの動向について述べる．第 4 章では疎行列ベクトル積の Byte/FLOPS について述べる．第 5 章では Gather 機能付き拡張メモリの基本アーキテクチャとキャッシュの併用に関する提案について述べる．第 6 章では疎行列ベクトル積に対するメモリシステム側のキャッシュヒット率や効果の評価を示す。第 7 章で関連研究を紹介したのち、第 8 章でまとめる。

2. Exa FLOPS 級マシンのメモリシステムの課題

本章では電力制約(20MW 程度)を制約条件として検討されている Exa FLOPS 級マシンのメモリシステムにおいて浮上してきた課題について，3 つの観点から論じる．

2.1 メモリバンド幅(Byte/FLOP)の低下

最近 Micron 社からプロトタイプが公開された Hybrid Memory Cube(HMC)[17]-[19]が，主に電力制約の観点から，バンド幅と容量を両立する必要がある多くのアプリケーション向けの主記憶として有望視されている。HMC の採用により電力効率で約 10 倍，バンド幅で約 20 倍の向上が見込める．HMC の採用を仮定したとしても電力制約およびピン数制約からソケットあたりのメモリポート数は少数に絞らざるを得ない．その結果，ある程度の汎用性を有する Exa FLOPS 級マシンにおいて，演算能力とメモリバンド幅のバランス指標である Byte/FLOP 値は 0.1 程度になると予想されている．これは現状(京コンピュータ)の 1/5 に過ぎない．

一方，使う側の視点からは文部科学省による重点指定になっている 40 弱のアプリケーションのうち 4 割程度(最大のグループ)は Byte/FLOP 値が実効性能に直結するため，高い Byte/FLOPS 値を要求しているものであることも明らかになってきた．それらは Top500 の評価指標である Linpack(密行列系)が速くなったとしても意味が無いということから，主に疎行列系の連立一次方程式に帰着されているものが多いと考えられる．メモリバンド幅を絞ってチップ内に演算器を詰め込んでも，電力効率が悪化するだけで上記アプリの性能向上には全く繋がらない．

以上の状況を合わせると，現時点で電力制約に由来する性能問題の最重要課題は，短くて少ない配線しか使えない状態での実効的な Byte/FLOPS 値の向上にあると言っても過言ではない状況である．サイエンスドリブンな観点からすれば，疎行列処理での実効バンド幅向上が極めて重要と考えられる．

2.2 メモリ容量(Byte/FLOPS)の低下

電力やバンド幅の観点からは HMC が極めて有望であるが, DRAM の積層枚数の限界性から HMC のパッケージ内に収容できるメモリ容量は，従来の DIMM ベースのメモリ実装方式が電力制約を無視して実現できる容量に比べて低くならざるを得ない。

一方，使う側の視点からは，1 割程度の重点アプリでは演算器を詰め込んだ上で SoC(System on Chip)のオンチップメモリで対応が可能な程度のメモリ容量しか必要ない．しかし，汎用型のマシンには 1TFLOPS あたり 100GB のメモリ容量が必要とされている．これは少なすぎて困るという声が上がっている現状(京コンピュータ)と同等で，これ以下にしたら使えないという我慢の限界ラインと考えられる．

この実現には DRAM のようにリーク電力が大きいメモリではなく，大容量性と高速性と低電力性を兼ね備えたタイプの新型メモリ(MRAM 等)の可能性に期待される。

2.3 アクセスアドレスの局所性の低下

電力制約から FLOPS 値確保のためにマルチコアやメニーコアの採用が必然的になる。電力制約およびピン数制約からコアあたりのメモリポート数は極めて低い値となる．結果として，1 本のメモリポートに多数のコアからのメモリアクセス要求が同時に発生する．これによりキャッシュや DRAM がバンド幅向上のために利用しているメモリアクセスの際のバースト長が十分に確保できなくなる．つまり，メモリシステムにはランダムアクセスに対する耐性がこれまで以上に求められるようになる．

この実現には，短いサイクルタイムを有する大容量メモリの開発が重要と考えられる．Micron 社の HMC のように DRAM ベースでも細いメモリチャネルを多数インタリーブ構成とすることで，ある程度は対応できると考えられる．しかし，MRAM のように DRAM より短いサイクルタイムを有するタイプのメモリの方が有利であり，その重要性が高まってきている．

3. 日米の Exa FLOPS 級マシンのメモリシステムの動向

本章では Exa FLOPS 級マシンのメモリシステムに関して日米においてどのような視点に重点を置いて検討されているかについて論じる。

3.1 米国の動向

米国での Exa FLOPS 級マシンの開発に関する検討は DARPA 予算で 2007 年頃に始まっており，日本より 3〜4 年先行している．必ずしも米国の開発体制は 1 枚岩ではなく，NVIDIA 社が中心になった Echelon プロジェクト[13]のようにシステムベン

Vol.2011-ARC-197 No.14
Vol.2011-HPC-132 No.14
2011/11/28

ダー主体の検討もあるし，ワークショップで大勢が議論して方向性を決めているものもある．サイエンスドリブンの視点から注目すべきは，アプリユーザを多く抱える国立研究所(OakRidge 国立研究所および Sandia 国立研究所)が主体になって組織されている Exa FLOPS マシン開発組織である IAA の動向[14]-[16]である．その Web サイト上で，メモリシステムの Focus area[15]として”advanced data movement functionality into the memory subsystem hardware that support advanced atomic memory operations and distributed memory operations such as gather/scatter capabilities”を掲げている．さらに”The project will also explore accelerating sparse memory accesses by exposing the underlying data structure to the CPU more intelligently. Longer term efforts based on 3D stacking will be focused on significantly improving bandwidth and lowering power.”とあるように，これまでプロセッサ側で行われてきた gather/scatter 機能をメモリシステム側で行うことにより，疎行列処理における実効バンド幅を向上させることに注力して設計している．これは前章のサイエンスドリブンな立場から自然な方向に進んでいると言える．さらに長期的にはその仕組みを 3D 積層技術によってバンド幅と電力を改善しようとしている．2008 年に発行された別の資料[14]ではその開発元は Micron 社となっており，その第一弾の中間成果が HMC[17]-[19]となって 3 年後である本年になって表に出てきたと考えられる．以上の動向を総合すると，gather/scatter 機能を有する HPC 向けの HMC が，やがて Micron 社から出てくるのが自然な流れと考えられる．

3.2 日本の動向

日本では米国に先立ち 2003 年頃から DIMMnet プロジェクトにより IAA と同様の方向性を持った検討が行われてきた．筆者らは日米の Future generation の Innovative architechture を議論するワークショップ(IWIA)において 2004 年[1]から毎年，現在の米国の Exa ロードマップ[20]作成の中心的役割をになう P. M. Kogge らとこの技術に関して議論を重ねてきた歴史がある．Kogge らは元々PIM(Processor in Memory)の研究を DIMMnet よりだいぶ前の 1990 年代から進めてきたが，どちらかというとオンチップメモリ内での演算に重点が置かれていた．しかし，メモリ容量と演算能力のバランス維持と，半導体プロセス使い分けの両面から，DIMMnet のようにメモリシステム内での高機能なデータ移動にフォーカスした研究の価値が徐々に認められてきた結果として，Kodge の弟子が率いる BoB プロジェクトを経由して，現在の IAA を中心とした流れが形成されたと考えられる．しかし，その歴史や，サイエンスドリブンな視点からの重要性に気づいている日本人は少数と考えられる．

現在，日本におけるメモリシステムの Focus area は深い階層メモリシステムである．つまり，深いメモリ階層を設定し，階層キャッシュの仕組みを適切に運用することでメモリバンド幅(Byte/FLOP)の低下という課題を解決することに重点が置かれており，ソフトウェアスタックもそれに対応することがロードマップとして掲げられる方向に進んでいる．これは， IAA ではなく，主に Echelon[13]の方向性と類似しており，その源は Stanford 大の教授で NVIDIA 社の W. J. Dally の設計思想にある．本報告は,そのような流れに対し,国内で重視が決まったサイエンスドリブンな視点から，疎行列処理への適合性という評価軸で，定量的手法により警鐘を鳴らすものである．

4. 疎行列ベクトル積の Byte/FLOP

疎行列ベクトル積は連立一次方程式求解や固有値計算の中で用いられ，実行時間の大半を占めるため，最も重要な HPC 計算カーネルの一つである．反復解法である CG 法などを包含するクリロフ部分空間法の中で大半の計算時間を消費する。疎行列はメモリ容量と計算量の節約のため、通常、非零要素のみを値とインデックスを格納する 1 次元配列の組として格納する。疎行列ベクトル積の処理の特徴は 1 次元インデックス配列を添字とした 1 次元配列の間接参照にあり、その結果として様々なプロセッサ上でメモリバンド幅律速に陥る。

疎行列ベクトル積の Byte/FLOP 値は以下の三種類の配列アクセスによって読み出した値を用いて浮動小数積和演算(2FLOP)を行う比率になる．

(a)再利用性の無い浮動小数行列データ 1 次元配列の連続読み出し
(b)再利用性の無い整数インデックス 1 次元配列の連続読み出し
(c)(b)を用いて多少の再利用性がある浮動小数ベクトル 1 次元配列間接読み出し

(a)(b)(c)を連続アクセスと間接アクセスの効率の違いを区別せずに算出する Byte/FLOP は，以下の式(1)で表現できる．

倍精度の場合：(8B＋4B＋8B)/2FLOP＝10 **(1)**
単精度の場合：(4B＋4B＋4B)/2FLOP＝6 **(2)**

しかし，実際には京や Nvidia 社の GPU などと同様の 128 バイトのキャッシュラインを有するプロセッサを用いて，上記(c)のアクセスがランダムに近いインデックスでキャッシュサイズより十分に大きなベクトルを間接参照した場合は，128 バイトのライン中に 8 バイトまたは 4 バイトしか有効なデータが存在しない．以上の転送効率を反映した実状に近い Byte/FLOP は，以下の式で表現できる．

倍精度の場合：(8B＋4B＋128B)/2FLOP＝70 **(3)**
単精度の場合：(4B＋4B＋128B)/2FLOP＝68 **(4)**

例えば京で上記の動作モードで動く場合はソケットあたりの主記憶バンド幅が 80GB/s なので，倍精度の場合，ソケットあたり 80GB/s÷70B/FLOP＝1.14GFLOPS しか期待できない．この値はプロセッサのピーク性能 128GFLOPS の 100 分の 1 以下である．この利用形態ではいくら演算能力のみを追加しても全く無意味である．強化すべきことはメモリバンド幅であり，式(3)(4)ではなく式(1)(2)で動くようにバンド幅を有効活用する対策である．

実際の疎行列では完全なランダムアクセスではなく，かつ，キャッシュ容量分だけ

Vol.2011-ARC-197 No.14
Vol.2011-HPC-132 No.14
2011/11/28

ベクトルへの主記憶アクセスが減少するため，程度はキャッシュ容量と行列依存であるが上記より高い FLOPS 値が観測されるはずである．しかし，この利用状況で，京よりも大きな問題の実行が求められる Exa FLOPS 級マシンにおいて，キャッシュにどの程度期待できるのかという点は検証が必要であり、本報告では実験により確認する．

5. Gather 機能付きメモリとキャッシュの併用

5.1 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した．提案方式の基本コンセプトを図 1 に示す．

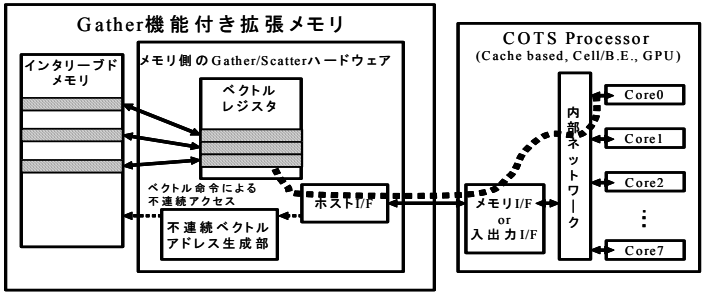

図 1　　提案する基本アーキテクチャ

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	

5.2 キャッシュの併用

疎行列の対角付近には非零要素が比較的集まっているケースが多い．そのような位置の処理では疎行列ベクトル積のベクトル部のアクセスにおいてデータの再利用性が期待できる．よって，基本構成の補助機構として，外部メモリアクセスの前段においてキャッシュを導入し，キャッシュにヒットした場合は外部メモリへのアクセスを抑制することで，外部メモリのバンド幅を有効活用することを提案する．この方式についてどの程度のサイズの行列で，どの程度の効果が得られるのかについて，定量的に評価することが必要である．

5.3 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 つのポリシーを列挙する。詳細は文献[10]で述べているので参照されたい。なお、後述する性能評価では DDR3 型の現時点で最も一般的に使用されている DRAM を用いており，以下の最後の項目は反映していない。新型メモリの効果の評価は今後の課題である．

- 狭いビット幅のチャネル
- 深いインタリーブ
- Open ページポリシー
- 低サイクルタイム型メモリの活用

5.4 ホストインタフェース

Exa FLOPS 級マシンのメモリシステムにおける提案メモリシステムは，DRAM または MRAM を 3D 積層実装した HMC として実装されることが有望である．つまり，ホストインタフェースは HMC のチャネルインタフェースに準拠することになる．これはこれまで筆者らが提案してきた DIMMnet-2 ベースのメモリシステムを最新の 3D 積層技術で実装しなおしたものと言い換えることができる．まだ，現時点では実機が出現していないが，前述の通り IAA における開発方針と Micron 社の開発動向をあわせれば，この提案システムと類似した実機(Gather 機能つき HMC)が出現する可能性は高いと考えられる．

6. 性能評価

6.1 評価方法

6.1.1 GPU における実機性能評価

階層キャッシュの疎行列処理における性能評価として，汎用キャッシュ(L1：64KB×14，L2：768KB，合計 1.6MB)を備えた GPU である NVIDIA 社の C2050 上でプロファイラを用いて，単精度の疎行列ベクトル積実行時の L1 キャッシュおよび L2 キャッ

Vol.2011-ARC-197 No.14
Vol.2011-HPC-132 No.14
2011/11/28

シュのヒット率を測定する．計測に用いたカウンタ値は l1_global_load_hit，l1_global_load_miss ， l2_subp0_read_sector_misses ， l2_subp1_read_sector_misses ，l2_subp0_read_sector_queries，l2_subp1_read_sector_queries の 6 つである．

6.1.2 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[23]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[21][22]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
4) 前段キャッシュのある構成に対応させた

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 DDR3_micron_64M_8B_x4_sg15.ini 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 種類	DDR3
容量	2Gbit
バンク数	8
行数	32768
列数	2048
tCK(転送サイクルタイム)	1.5ns
CL(CAS レイテンシ)	10
BL(バースト長)	8
tRAS(RAS レイテンシ)	24
tRCD(RAS to CAS レイテンシ)	10
tCCD(CAS to CAS レイテンシ)	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。

チャネル本数は 16 に固定した。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。

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	16
1 サイクルで発生する Transaction 数	8
チャネルあたりのビット幅	8
ランク数	4

(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。

(4) 前段キャッシュの構成

不規則アクセスへの耐性とメモリコントローラ ASIC への実装を考慮し，ラインサイズ 8 バイト，1K 語(容量 8KB)および 10K 語(容量 80KB)の前段キャッシュを挿入した．キャッシュに当たった場合は DRAM へのアクセスはせず，ミスした場合は FIFO 方式で選択した古い 1 ラインを上書きする．

6.1.4 ワークロード

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 Collection[22]から比較的小規模な疎行列によって作成した。上記コレクションの拡張子 mtx のファイルの index 部分を index としてデータサイズ 8 バイトとして 0 番地から配置される配列をアクセスする際のアドレストレースを DRAMsim が受け付ける形式で生成した。本評価に使用した行列を表に示す。今回の実験はシミュレータの実行時間の制約から行列のサイズはあまり大きくせず、文献[8][9]でも取り上げた行列と、文献[28]で GPU 上での疎行列ベクトル積の FLOPS 値が評価されている行列の中から小さい順に 10 個を選択し、評価を行った。表 4 に評価に用いた疎行列の特徴を示す。

また、今回の実験では GPU 向けの評価として、文献[8][9]にて提案した提案拡張メモリと GPU を組み合わせたシステム向けのアルゴリズムの前処理部分を適用したアドレストレースを用いた場合のバンド幅も測定した。なお、今回用いた前処理では折

Vol.2011-ARC-197 No.14
Vol.2011-HPC-132 No.14
2011/11/28

り畳み幅の個別調整は行っていないものを用いた。また、0 パディングが index ファイルには入っているが、値は 0 に固定されるためメモリアクセスを行わなくてもコントローラ内部のレジスタまたはキャッシュなどで代用できるため、0 パディングに対応するアクセスはトレースファイルから省略されている。

表 4　　評価に用いた疎行列の特徴

行列名	行数	非零要素数			
		合計	行平均	行最大	標準偏差
msc01440	1440	23855	16	40	12.3
bcsstk13	2003	42943	21	84	14.68
fv1	9604	47434	4	5	2.92
nasa4704	4704	54730	7	20	4.28
bcsstk15	3948	60882	15	39	6.83
aft01	8205	66886	8	11	2.56
Dubcova1	16129	134569	8	12	3.57
s2rmq4m1	5489	143300	26	30	5.04
bcsstk16	4884	147631	30	42	9.66
Na5	5832	155731	26	185	35.71
msc10848	10848	620313	57	300	49.4
exdata_1	6001	1137751	189	1501	390.27
thermal	147,900	3,489,300	23	27	6.86
hood	220,542	5,494,489	24	51	13.31
F1	343,791	13,590,452	39	306	19.97
G3_circuit	1,585,478	4,623,152	2	4	2.18

6.2 結果

GPU(C2050)上での単精度疎行列ベクトル積実行時の L1 キャッシュおよび L2 キャッシュのいずれかでヒットしたアクセスの比率を図 6 に示す．ここでのメモリアクセスはベクトルへのアクセスだけでなく，行列の非零要素の値やインデックスへの連続アクセスも含んだ全てのアクセスにおけるヒット率である．回数で 2/3 を占める連続アクセスの部分はキャッシュラインの先頭部でのアクセスはミスヒットとなるが，後続要素へのアクセス 31 回はヒットとなる. 再利用性はないので上記連続アクセス分の外部メモリアクセスは回避できない．この程度の行列サイズでは合計 1.6MB のキャッシュ上にベクトルの大部分が載り, ヒット率は 50～70%の間と比較的高い状況にある．しかし，言い換えると 30～50%は 4 章の(4)式のモードで外部メモリアクセスをしており，メモリバンド幅律速で，ピーク演算速度には遠く及ばない．

評価に用いた各行列でのヒット率の分布を線形近似した場合，行数に対してマイナスの傾きが観測されており，行数が大きくなるにつれて概ねヒット率が減っていく傾向が現れている．近似式は y =-4E-06x+63.537 であった．これから予測されるキャッシュのヒット率が 0 に近づく行数は 1500 万行近辺である．ただし，キャッシュのサイズ 1.6MB は 400K 行のベクトルと等価であり，さらに大きな行列で評価すれば，1500 万行より小さくても激しい性能低下が起きる可能性が高いと考えられる．

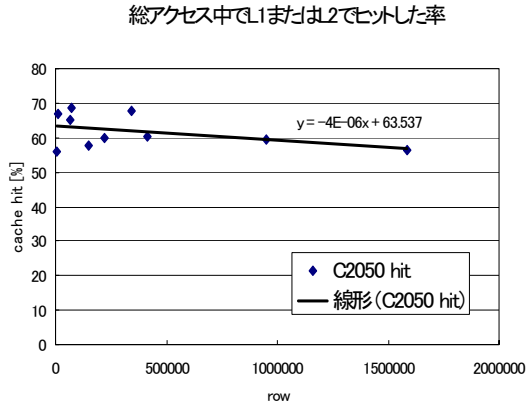

図 6　GPU(C2050)上での疎行列ベクトル積のヒット率

次に提案メモリシステムにおけるシミュレータにより計測した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を表 5 に示した．加速率が高いものほど暖色系に塗った．概ね，行数が多くなるとキャッシュによる加速率が鈍る傾向がわかる．

図 7 は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に対する行数(横軸)とキャッシュ追加による加速率(縦軸)の関係を図示したものである。

追加したキャッシュの語数が 1K 語(8KB)の場合，その語数と行数が近い **msc01440 や bcsstk13 は大幅に加速した**. 追加したキャッシュの語数が 10 倍程度の行数を持つ行列は 0～30%程度の加速が観測でき，非零要素の配置次第ではある程度の効果があると言える． しかし，追加したキャッシュの語数の 100 倍以上の行数を持つ行列(thermal, hood, F1, G3_ciircuit) では-0.2～1.3%で効果はほとんど観測できなかった.

Vol.2011-ARC-197 No.14
Vol.2011-HPC-132 No.14
2011/11/28

表 5　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

		Cache なし	1K 語 Cache 追加		10K 語 Cache 追加	
行列名	行数	バンド幅	バンド幅	加速率	バンド幅	加速率
msc01440	1,440	17.749	38.913	119.2%	39.352	121.7%
bcsstk13	2,003	18.044	30.076	66.7%	39.736	120.2%
bcsstk15	3,948	18.901	23.813	26.0%	39.733	110.2%
nasa4704	4,704	19.073	22.123	16.0%	39.732	108.3%
bcsstk16	4,884	19.361	24.124	24.6%	39.736	105.2%
s2rmq4m1	5,489	19.097	22.348	17.0%	39.735	108.1%
Na5	5,832	16.137	20.107	24.6%	39.735	146.2%
exdata_1	6,001	17.367	17.718	2.0%	39.736	128.8%
aft01	8,205	16.729	17.586	5.1%	32.22	92.6%
fv1	9,604	19.464	20.108	3.3%	39.731	104.1%
msc10848	10,848	18.449	20.897	13.3%	39.708	115.2%
Dubcova	16,129	19.309	23.882	23.7%	39.734	105.8%
thermal	147,900	19.771	19.835	0.3%	21.143	6.9%
hood	220,542	18.458	18.696	1.3%	20.353	10.3%
F1	343,791	19.202	19.367	0.9%		
G3_circuit	1,585,478	19.642	19.609	−0.2%	19.721	0.4%

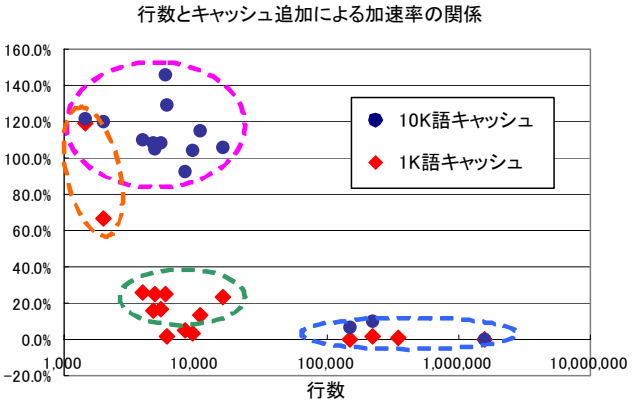

図 7　疎行列ベクトル積実行時のベクトルアクセスの実効バンド幅に対する行数とキャッシュ追加による加速率の関係

追加したキャッシュの語数が 10K 語(80 KB)の場合，その語数と行数が近い図 7 の左上の点線で囲った多くの行列がキャッシュに全て入るようになったため，Gather の 2 倍強の性能が出ている．しかし，キャッシュ容量が足りなくなると急速にキャッシュによる加速が無くなり，10 倍程度の語数でもほとんど加速が無くなってしまった．

つまり，疎行列ベクトル積の場合，キャッシュに列ベクトルの大半が入る場合はキャッシュの効果は絶大だが，溢れたら効果はほとんど無くなる．それに対して Gather は行列のサイズにほとんど依存せず，安定した加速が得られる．

7. 関連研究

不連続アクセスに伴う実効バンド幅の低下問題を解決するために筆者らは先行研究[1]-[12]で Scatter/Gather 機能を有する拡張メモリシステムを提案した。文献[8]-[12]では疎行列ベクトル積においても評価を行ない、有効性を示してきた。ただし，これらはキャッシュの効果を全く利用しない場合の評価であった． 本報告では Gather 機能とキャッシュを併用し，それぞれの効果と行列サイズの関係を明らかにしている．

従来行われている CPU や GPU 上での疎行列ベクトル積高速化の研究[24]-[27]は、ほとんどが列ベクトルはキャッシュ上に載ることが多い状態での性能評価となっており、今後計算対象が大きくなってきた場合の性能低下の問題解決にはほとんど取り組んでいない。文献[8]-[12]および本報告のターゲットとするアプリケーションの設定は Exa FLOPS 級マシンが取り扱うことが予想される「列ベクトルがキャッシュには到底入らない領域」を対象としている．よって，行列サイズや並列ノード数のスケーラビリティ（細粒度ランダム通信の排除）と、GPU のキャッシュメモリ容量の限界性を重視してキャッシュには基本的には頼らない設計になっている。

8. おわりに

本報告では Gather 機能を有するメモリシステムについてエクサ FLOPS 級マシンの設計という文脈上で考察を行った。その判断材料として現状の GPU と，キャッシュを前段に併用した Gather 機能を有するメモリシステムについて疎行列ベクトル積に対する性能評価を行なった．その結果，階層キャッシュを備えた GPU のキャッシュヒット率は行列行数に概ね比例して低下することが観測された．キャッシュを前段に併用した Gather 機能を有するメモリシステムにおけるキャッシュによる性能向上は限定的で，その容量の 10 倍程度のベクトルサイズまでで頭打ちとなった．不規則型応用に対して再利用性に高速化原理をおくキャッシュに頼ることは危険であり，Gather 機構が重要であることが改めて確認された．

検討が 3 年先行している米国の Exa FLOPS マシン開発機関のメモリシステムに関する注力技術は Gather 機能を有するメモリシステムである．既に HMC のプロトタイ

プが Micron 社から発表されており，Gather 機能がそこに入る可能性が高まっている．一方, 国内における現在のメモリシステムに関する注力技術は階層キャッシュである．そのような作る側視点からのアプローチより，サイエンスドリブンな視点に移行し，Gather 機能を有するメモリシステムによる疎行列処理における実効的 Byte/FLOP の向上に軸足を移行すべきことを提言したい．

今後の課題としては、より大きな行列群でのスケーラビリティの評価や、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。処理性能だけでなく電力性能の評価も重要である．これらの検討は米国の Micron 社任せではなく国内の得意分野を生かした独自開発を進めるべきか否かを判断する材料として重要性が高まってきている．さらに Gather 機能を有するメモリシステムの利用を促進するベクトル化コンパイラや行列演算ライブラリなどの基本ソフトウェアの整備も今後の課題である。

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, 太田, 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, 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”, 情報処理学会 HPC 研究会 Vol.2010-HPC-126 No.20, Aug. 2010.
10) 田邊, Nuttapon, 中條 : “Gather 機能付き拡張メモリのアクセス性能の評価”, 情報処理学会 HPC 研究会, Vol.2010-HPC-128, Dec. 2010.
11) 田邊, Nuttapon, 中條, 小川, 高田, 城 : “GPU と拡張メモリによる疎行列ベクトル積性能の行列形状依存性軽減”, 情報処理学会 HPC 研究会, Vol.2010-HPC-129, Mar. 2011.
12) 小郷, 田邊, 高田, 城: “メモリアクセラレータで強化した GPU の CG 法による評価”, 情報処理学会 HPC 研究会, Vol.2010-HPC-130, Aug. 2011.
13) W. J. Dally : “GPU Computing to Exa scale and Beyond”, slide at SC10 http://www.nvidia.com/content/PDF/sc_2010/theater/Dally_SC10.pdf
14) IAA : “Exascale Computing and The Institute for Advanced Architectures and Algorithms (IAA) ”, http://www.hpcuserforum.com/presentations/Norfolk/Sandia%20IAA.hpcuser.ppt
15) IAA : “Focus area”, http://iaa.sandia.gov/focus-areas/index.html
16) R. C. Murphy, A. F. Rodrigues, J. A. Ang : “Memory Opportunities for High Performance Computing (MOHPC) Final Report”, SANDIA REPORT, SAND2009-7291 Feb. 2009. http://www.cs.sandia.gov/CSRI/Workshops/2008/MOHPC/MOHPC-1.pdf
17) Micron Technology, Inc. : "ハイブリッドメモリキューブ", http://jp.micron.com/innovations/hmc.html
18) Micron Technology, Inc. : " Hybrid Memory Cube : Breakthrough DRAM Performance with a Fundamentally Re-Architected DRAM Subsystem”, Hotchips 23, Aug. 2011.
19) Hisa Ando :” Hotchips23 – メモリバンド幅を画期的に高める Hybrid Memory Cube”, Sep. 2011. http://journal.mycom.co.jp/articles/2011/09/13/hot_chips23_micron/index.html
20) P. M. Kogge et.al. : “ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems," Univ. of Notre Dame, CSE Dept. Tech. Report TR-2008-13, Sep. 2008.
21) 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
22) B. Jacob : "DRAMsim: A Detailed Memory-System Simulation Framework", http://www.ece.umd.edu/dramsim/v1/
23) B. Jacob : "DRAMSim2", http://www.ece.umd.edu/dramsim/
24) Tim Davis : " The University of Florida Sparse Matrix Collection", http://www.cise.ufl.edu/research/sparse/matrices/
25) N. Bell, M. Garland : "Eficient Sparse Matrix-Vector Multiplication on CUDA", NVIDIA Technical Report NVR-2008-004, Dec. 2008
26) M. M. Baskaran, R. Bordawekar : "Optimizing Sparse Matrix-Vector Multiplication on GPUs", IBM Research Report, RC24704, Apr. 2009
27) A. Cevahir, A. Nukada, S. Matsuoka : "CG on GPU-enhanced Clusters", 情報処理学会 HPC 研究会 Vol.2009-HPC-123 No.15 Nov. 2009.
28) A. Cevahir, A. Nukada, S. Matsuoka : " An Efficient Conjugate Gradient Solver on Double Precision Multi-GPU Systems", 先進的計算基板システムシンポジウム SACSIS2009, pp.353-360, May 2009.