Vol.2012-HPC-135 No.17
2012/8/2

疎行列のキャッシュへの適合性分類に関する予備評価

冨森苑子[†1]　田邊 昇[†2]　高田雅美[†1]　城 和貴[†1]

エクサスケールマシンは複雑なメモリシステムとなることが予想されている．同マシンへの適用を視野に入れた疎行列ライブラリの実現に向け，本報告では疎行列のキャッシュへの適合性分類に資する疎行列の特性に関する新しい指標として「列インデックス列の空間的局所性」を提案する．さらに，入力疎行列および Fold 法前処理後の提案指標の値をフロリダ大学の疎行列コレクションを用いて評価した．その結果，疎行列ベクトル積処理性能と L1 キャッシュヒット率と新指標の間には有意な相関関係があることが確認できた．よって，従来から指摘していた行列サイズと併せ，本指標をアプリ固有の最適化を避けたメモリアクセス機構や前処理アルゴリズム自動選択の指標の一つとする．

Preliminary Evaluation for Classifying Suitability for Cache Memory of Sparse Matrices

SONOKO TOMIMORI[†1]　NOBORU TANABE[†2]
MASAMI TAKATA[†1]　KAZUKI JOE[†1]

In Japan, memory system of ExaFLOPS machines is expected more complex. In this paper, we propose a new characteristic of sparse matrices about spatial locality of row-index sequences in order to classify suitability for cache memory systems. Moreover, we evaluate proposal characteristic of input matrices and pre-processes (folding). Test matrices are chosen from University of Florida Sparse Matrix Collection. As a result, it is confirmed that there are significant correlations between performance of Sparse Matrix-Vector Product(SpMV) and the general purpose cache (L1) hit rate. Therefore, our characteristic is suitable for auto-tuning pre-processes and memory access mechanisms to avoid application specific optimization in conjunction with matrix size (the number of rows).

1. はじめに

近年，2018 年頃のエクサスケールマシンの実現に向けた検討[1]が日本でも行われるようになった．エクサスケールマシンではメモリバンド幅を十分に確保できなくなり，それをカバーするための複雑なメモリシステムの採用が予想されている．大規模疎行列を係数とする連立一次方程式(疎行列ベクトル積)に帰着される応用ではメモリバンド幅への要求が高い．国内の重点アプリケーションの多くがこのクラスに属するため，高速化へのニーズが高い．

また，近年ではグラフ処理に代表されるビッグデータ処理が注目を集めている．これらにおいても，巨大で複雑な非零要素配置を有する疎行列で表現される処理が必要になる．Web サイトの重要性を与える PageRank などの大規模非定型情報の検索[2]や嗜好分析・リコメンーションにおいて，疎行列処理の大規模化と高速化が求められている．

現時点での世界第二位のスーパーコンピュータである京は 2 階層のキャッシュをベースにした比較的シンプルなメモリシステムを有する．しかし，その上での最適化でさえ容易ではない．ごく少数の選抜アプリケーションのみに最適化のプロが配置され，そのアプリケーション固有の性質を利用した極限に迫る最適化が行われている．一方，選抜されなかったアプリケーションや，寿命が短いアプリケーションでは，ユーザーの科学者たちは性能チューニングに時間と人手を投資できない．疎行列処理に帰着される応用だけを取ってみても，フロリダ大学の疎行列コレクション[3]をみれば明らかなように，その非零要素配置は多種多様である．そこで重要になるのが最適化済みのライブラリや自動最適化コンパイラである．複雑なメモリシステムを有するエクサスケールマシンでは，自動化された最適化機構の重要性が一段と高まる．

以上のような認識から筆者らは，アプリケーション固有の最適化を避けた二種類の統一的調整機能や，アクセス機構選択による疎行列向けの新しい自動チューニング手法を提案し，それらに基づく大規模疎行列やメモリシステムの特性を考慮した高速な汎用疎行列ライブラリの構築に着手した．[4]

疎行列処理を主なターゲットとした計算時間の観点からの自動最適化機構を実現する端緒として，筆者らはやや複雑なメモリシステムを有する GPU(NVIDIA C2050)と各種疎行列の間での，疎行列ベクトル積性能を変動させる要因の明確化を試みることとした．

本研究で取り上げる GPU は L1/L2 キャッシュを有し，コアレスドアクセス条件を満たさない場合は実効メモリバンド幅が大幅に低下するメモリシステムを有する．そのようなキャッシュベースのメモリシステムとの組合せにおいて，

†1 奈良女子大学
Nara Women's University
†2 (株)東芝
Toshiba corporation

a) Intel，Xeon は，米国およびその他の国における Intel Corporation の商標です．

処理性能を決める要因としてキャッシュサイズと行列の行数の関係が重要であることを筆者らは明らかにした．[5]

ところが，行数や非零要素数が大幅に少ないにもかかわらず性能やヒット率に低いような逆転現象があるなど，行数以外にも性能やヒット率に大きな影響を与える因子が存在することを発見した．本研究はそのようなキャッシュへの適合・不適合を左右する重要因子として，疎行列の特性の新指標(列インデックス列の空間的局所性)を提案し，その有効性を評価する．

2. GPU のメモリシステムの特性

疎行列処理を主なターゲットとした計算時間の観点からの自動最適化機構を実現する端緒として，筆者らはやや複雑なメモリシステムを有する Fermi 世代の GPU(NVIDIA C2050)[6][7]をとりあげる．

2.1 Fermi 世代 GPU のアーキテクチャ

Fermi 世代 GPU のアーキテクチャの特徴は以下のとおりである．

- GPU あたり 16 個までの SM
- SM あたり 1～48Warp（Warp あたり 32 スレッド）
- SM ごとに 32 個の CUDA コアと 64KB の構成選択可能な L1 キャッシュ兼共有メモリ
- メモリアクセスは Warp(32 スレッド) 単位で実行
- 全てのグローバルメモリアクセスは L2 キャッシュ(容量 768KB)を経由
- キャッシュラインサイズ 128 バイト
- 外部メモリは GDDR5 型 DRAM，64bit 幅 6 ポート，最小アクセス粒度(Non-caching 設定時)32 バイト，ピークバンド幅 177GB/s
- ECC 機能のサポート
- CUDA コアあたり 32 個の単精度浮動小数 FMA(Fused Multiply Add)ユニット，16 個の倍精度浮動小数 FMA ユニット

2.2 疎行列ベクトル積における挙動

Fermi 世代 GPU のメモリシステムの疎行列ベクトル積における挙動は以下のとおりである．アクセスの効率化のために多くの場合，文献[7][8]に示されるような 0 パディング(ミスアラインによる効率半減防止)を行なうことが有効である．キャッシュのヒット率は以下の二種類のメモリアクセスの効果が合成されたものとなる．

- 不規則な非零要素配置に伴う列ベクトルアクセスが間接参照となり，メモリバンド幅がボトルネックとなる．その効率は index 配列の内容によって変化する．図 1 に示されるように Warp 内の 32 スレッドのメモリアクセスが N キャッシュラインに散らばる場合は，128 バイト単位のメモリアクセスを N 回繰り返して 32 個のデータをロードする．その結果，実効メモリバンド幅が単精度浮動小数の場合 N/32 に低下する
- index 配列と 1 次元化した疎行列配列については，ライン内の後続データは有効活用されるのでデバイスメモリへのアクセス頻度を抑制できる．ミスヒットレイテンシを隠せるだけのスレッドを稼動させられる場合，各スレッドのアクセスが単精度浮動小数の場合 32 回のうち 31 回が直前のミスヒット時に取り込まれたキャッシュラインにヒットするため，これらのアクセスにおけるキャッシュラインサイズの問題は無い．１回の疎行列ベクトル積処理の中で値の再利用性(時間的局所性)は無いため，キャッシュに入れると上記の列ベクトルを追い出してしまう可能性がある．京のセクタキャッシュのような制御が有効だが，Fermi 世代の GPU にその機構は無いため，キャッシュ容量が不足気味の状況ではこの問題が顕在化する恐れがある．

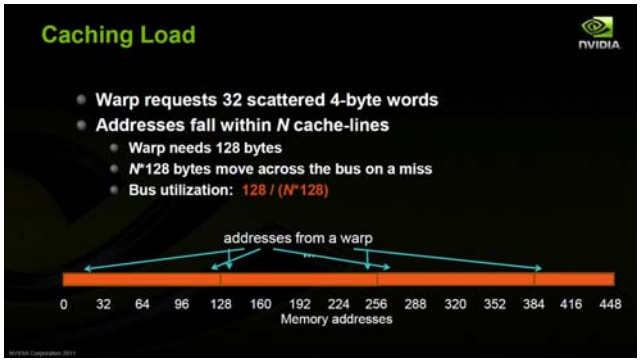


図 1　疎行列ベクトル積の列ベクトルへの間接参照における GPU のバンド幅低下問題 (文献[6]より引用)

Figure 1　The bandwidth degradation on indirect accesses of row vector for SpMV (Refered from [6]).

3. 疎行列の特性

3.1 疎行列コレクションにおける行列特性情報

フロリダ大学疎行列コレクションには多種多様な疎行列が行列の特性指標とともに公開されている．そのうちキャッシュヒット率に関連を持つと考えられる指標は以下の通りである．

- 行数(number of rows) ：これは疎行列ベクトル積においてメモリバンド幅ボトルネックの原因となる列ベクトルのサイズを与える．これにデータ型のサイズを乗じたものと L1 および L2 キャッシュとの大小関係が急激な性能変動をもたらす[5]ため極めて重要な指標
- 列数(number of columns)：行数と同じ（正方行列）であることが多いが正方行列ではないこともある
- 全非零要素数(nonzeros)：index 配列と 1 次元化した疎行列配列の要素数を与える．前処理アルゴリズムや格納法によってはキャッシュのヒット率に影響を与える

- データ型(type : real, complex, integer, or binary) : データサイズが大きい型ほど同じ個数のデータでもライン内の有効データの率が上がるためミス時の実効バンド幅が向上する
- 数値の対称性(numeric value symmetry) : 対称(値が 1)の場合，配列サイズを半減できるため，キャッシュやデバイスメモリへの負担が減る
- アプリ種別(kind) : 同じアプリ種別に由来するものは同じような特性を持つことが多い
- 行列形状図(Matrix pictures : CSparse パッケージの Matlab 用関数 cspy による出力．ただし色は要素の絶対値に対応するためキャッシュヒット率とは無関係)

3.2 行列サイズと疎行列ベクトル積性能の関係

筆者らの先行研究では行列サイズとキャッシュ容量の関係が疎行列ベクトル積性能に重要な影響があることが示されている[5]．キャッシュ容量が足りている状況では行列サイズ（行数）の増加とともに緩やかにヒット率が減少する傾向を示すが，足りなくなると急激にヒット率が低下し，急激な性能低下を引き起こし，列ベクトルサイズがキャッシュ容量の 10 倍以上になってくるとキャッシュの効果は殆ど見えなくなることが報告されている．その評価で用いたキャッシュ容量は Fermi 世代の GPU のものより小さく設定して完全に溢れる状況を観測しているが，フロリダ大学の疎行列コレクションより大きな行列を扱う場合には GPU においても同様な激しい性能低下が起きる状況になるものと考えられる．

GPU(C2050)の実機上でキャッシュのヒット率を観測した場合，行数と L1 ヒット率の関係は図 2 のようになる．図 2 から判るように全体としては両者には負の相関(行数が大きくなると L1 ヒット率が減少する傾向)がある．ただし，表 1 のように行数が大きくても行数が小さい行列の L1 ヒット率より高い逆転現象も観測されており，L1 ヒット率は行数だけで決まっていないこともわかる．よって，行数以外の L1 ヒット率への有意な相関を有する疎行列の特性に関する指標の探求が望まれる．

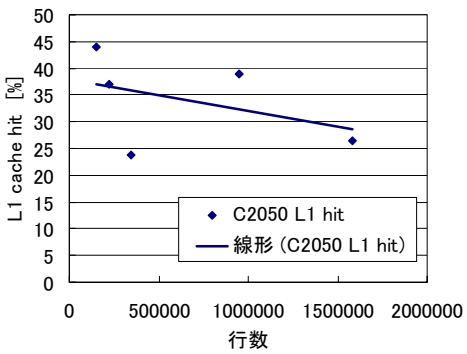
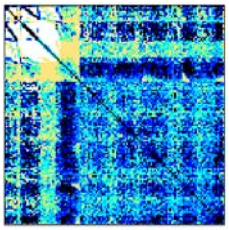
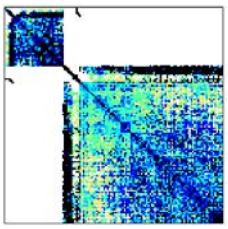


図 2　行数と L1 キャッシュヒット率の関係
Figure 2　The relation between the number of row and L1 cache hit rate .

表 1　逆転現象を起こしている行列の例
Table 1　Example of matrices with reverse phenomenon.

	F1	ldoor
行数	343,791	952,203
非零要素数	13,590,452	23,737,339
L1 ヒット率	24%	39%
GFLOPS	9.41 GFLOPS	13.27 GFLOPS
アプリ種別	構造解析	構造解析
行列形状図		

3.3 提案する行列特性指標

筆者らは疎行列のキャッシュへの適合性分類に資する疎行列の特性に関する新しい指標として「列インデックス列の空間的局所性」を提案する．図 3 にその概念図を示す．行列の特性値を表す場合の定義を以下に示す.「非零要素のみを CRS 形式で格納し，列ベクトル読み出しに用いるインデックス配列を先頭から順に読み出した際に，下位 5bit 以外が継続して一致している回数をカウントする．不一致が生じた時のカウント値を出力し，1 からカウントしなおす．その出力されたカウント値の数列の平均を取ったものを，列インデックス列の空間的局所性と定義する.」

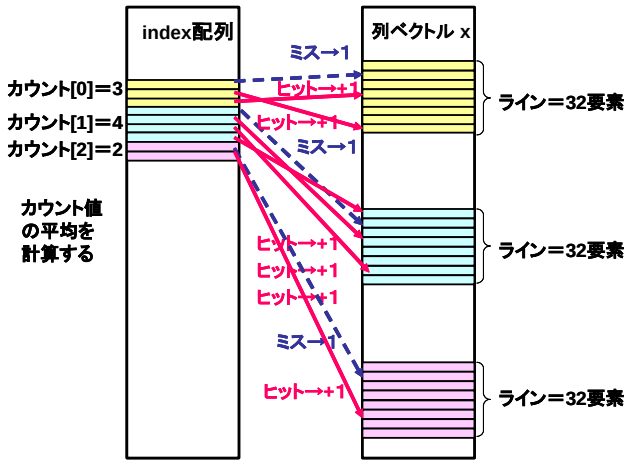


図 3　提案指標(列インデックス列の空間的局所性)
Figure 3　The proposed property (spatial locality of row-index sequences)

上記の値は，1 本のラインしかないキャッシュに対して疎行列ベクトル積を CRS 形式の疎行列に対して行なう際の列ベクトルをアクセスする際のキャッシュラインあたりにいくつの有効データが存在するのかという平均値を与え

る.その逆数は上記のアクセスに対するメモリバンド幅が，どれ位薄まってしまうのかを意味する．ただし，実際のキャッシュは多数のラインを用意することで時間的局所性を引き出すので，疎行列とキャッシュの相互作用の一面しか見ていないことから，上記の平均値は実機特性を厳密に表すものではなく，近似値である．

なお、上記の定義で下位 5bit としているのは 128 バイト(GPU 等の一般的なキャッシュラインサイズ)のラインの中に存在する 32(2^5)個の 4 バイトデータのいずれかへのアクセスは，キャッシュがヒットすることに対応している．

並べ替えを伴う前処理に対する特性値を表す場合の広義の定義を以下に示す.「列ベクトル読み出しに用いるインデックス配列を先頭から順に読み出した際に，下位 5bit 以外が一致している回数をカウントする．不一致が生じた時のカウント値を出力し，1 からカウントしなおす．その出力されたカウント値の数列の平均を取ったものを，列インデックス列の空間的局所性とする.」

4. 評価

4.1 実験環境と評価行列

今回の実験に用いた計算機環境aを　表 2 に示す．また，評価に用いた行列を表 3 に示す. 行列は University of Florida Sparse Matrix Collection から抜粋したものである. University of Florida Sparse Matrix Collection とは，実際のアプリケーションでよく生じる疎行列を集めたものである．これらの疎行列は，疎行列アルゴリズムの開発と性能評価のための数値線形代数の研究者に多く使用されている. 本研究では，先行研究[8][9]で使用した疎行列に新たに追加を行った. 追加された行列は，行列形状図上で非零要素が不規則に散らばっているように見える(キャッシュ向けの最適化が効きにくいと考えられる)疎行列を選んだ．それらは構造解析，電子回路解析，web 解析，道路網解析のアプリケーションに由来する疎行列である．nd24k については作者不明(アプリ種:2D/3D 問題)な行列で，行列形状図上では他の行列との質的な差が判らないものである．

表 2 測定環境

Table 2　Experimental environment.

CPU	Intel®Xeon®CPU　X5670 @ 2.93GHz
GPU	Nvidia Tesla C2050 (コア数 448)
デバイスメモリ	メモリバンド幅　144GB/s,3GB
ホスト I/F	PCI express x16 Gen.2 (最大バンド幅 8GB/s)
OS	RedHat Enterprise Linux Client release5.5
CUDA	Cuda3.2

表 3　評価に用いた行列

Table 3　Experimented matrices.

行列名	非零要素数	行数
crankseg_2	7,106,348	63,838
nd24k	14,393,817	72,000
thermal2	3,489,300	147,900
hood	5,494,489	220,542
F1	13,590,452	343,791
msdoor	10,328,399	415,863
rajat29	4,866,270	643,994
ASIC_680ks	12,329,176	682,712
apache2	2,766,523	715,176
ldoor	23,737,339	952,203
webbase-1M	3,105,536	1,000,005
delaunay_n20	2,097,124	1,048,576
roadNET-TX	1,281,106	1,393,383
Hamrle3	5,514,242	1,447,360
G3_circuit	4,623,152	1,585,478
roadNET-CA	1,844,404	1,971,281

4.2 評価実験

3.3 節で述べた疎行列の特性指標の有効性を確認するため，CRS 形式と前処理後における提案指標と，前処理後の L1 ヒット率を測定した．本研究における前処理は，Fold 法[8][9]を用いた．この Fold 法という前処理では，CRS 形式からインデックス配列内部のアクセス順序を GPU 向けに転置を用いて変更している．そのため，キャッシュのヒット率はその影響を受ける. つまり CRS 形式のアクセス順序とはキャッシュへの相性が大幅に異なるため，前処理の影響が顕著に観測できるものと期待できる．

CRS 形式で格納されたインデックス配列を先頭から読み込むと，非零要素が行内では昇降順となったインデックス値列が読み込まれる. 提案指標値を計測するには，下位 5bit 以外が一致している回数をカウントする．不一致が生じるとその時のカウント値を出力し，1 からカウントしなおす．その出力されたカウント値の数列の平均を取る．

図 4 は各評価行列に対して，前処理前（CRS 形式）と Fold 法による前処理後のキャッシュライン内部の有効データ率(赤と青の棒)の変化を示したものである．青い折れ線として前処理後の L1 キャッシュのヒット率も示している．行列の並び順は左から右に行数が大きくなる順に示している. 青い棒（有効データ数/ライン）と青い折れ線(L1 キャッシュのヒット率)の形状の類似性に不一致点が感じられる．

0 パディングの影響でヒット率が水増しされている現象とその度合いを示すために 0 パディングの index を削除した index 列に対する同指標の測定値(緑の棒)も併記した．0 パディング削除すると大きく有効データ率が減少する(水増しされていた)行列もあるが，変化が少ない行列もある．

表 1 に示した逆転現象がある二つの行列(F1，ldoor)の間には有効データ率に大きな差が認められ，行数の観点からL1 キャッシュヒット率が想定的に小さくなると考えられていた ldoor は，高い空間的局所性によって F1 より高い L1 ヒット率を示したものと考えられる．

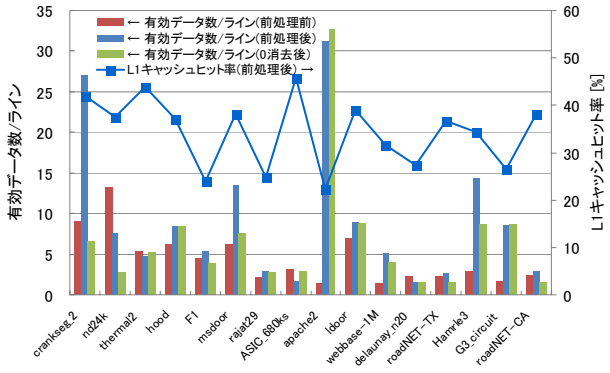


図 4 各種行列における L1 ヒット率と有効データ数/ライン（前処理前=CRS 形式，前処理=Fold 法）

Figure 4 The L1 cache hit rate and the number of valid data/line for various matrices. (Before pre-process = CRS, pre-process = Fold method)

次に，図 5 は Fold 法前処理後の L1 キャッシュのヒット率とライン内部の有効データ数の関係を示したものである．図 4 の青線と青棒の関係がはっきりしなかったように，図 5 は一見すると全体的にばらけているように見える．しかし，点線で囲ったように 3 つの集団に分類できると考えた．

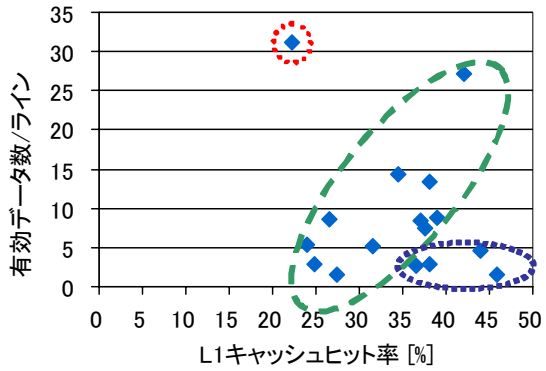


図 5 L1 ヒット率と有効データ数/ラインの関係(Fold 法前処理後)

Figure 5 The relation between L1 cache hit rate and the number of valid data/line (Pre-processed by Fold-method)

まず赤い点線で囲ったサンプル点(apache2)は図 4 に示されるように CRS 形式では空間的局所性が非常に少なく，Fold 法前処理後は空間的局所性が劇的に増加している．通常の CPU で実行する場合は連続アクセスに匹敵するほどの空間的局所性が高い状況で，L1 ミス 1 回に対して 31 回のヒットが続く状況に近い．プリフェッチも有効に効く可能性が高いと考えられる．ところが，GPU の場合は Warp 単位で多数のスレッド（Fermi 世代では 32 スレッド）が同時にアクセスすることになり，同じラインに 32 個のスレッドがアクセスしたとしても，そのラインが L1 になければ全部がミスとなる．次にスケジュールされたスレッドでも同じことが続けば L1 ミスが継続してしまう．apache2 の場合はこのような特殊な状況にあると考えられる．

次に青い点線で囲った 4 つのサンプル（thermal2，ASIC_680ks，roadNET-TX，roadNET-CA）では空間的局所性は低くなっている． L1 ヒット率と有効データ数/ラインの関係をこれらの 4 点について抜き出し，線形近似直線を引いたものを図 6 に示す．図 6 では，相関がほぼ 0 であることがわかる．よって，空間的局所性とは無関係な別の要因で L1 ヒット率が変動していることがわかる．具体的にはキャッシュヒット率を大きく左右するであろうもう一つの要因である時間的局所性の差によって L1 ヒット率が変動していると考えられるが，その指標化と分析は今後の課題とする．

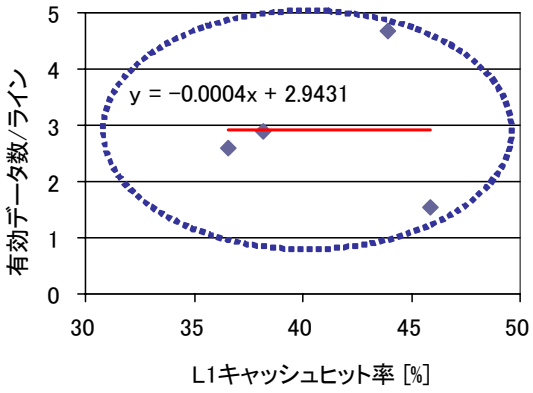


図 6 有効データ数/ラインと関係が薄いサンプル(Fold 法前処理後)

Figure 6 The relation between L1 cache hit rate and the number of valid data/line (Pre-processed by Fold-method)

上記の 2 グループを除いた緑の点線で囲まれたサンプルのみ抜き出して作成した L1 ヒット率と有効データ数/ラインの関係を図 7 に示す．正の相関(キャッシュライン内部の有効データ数が上昇するにつれて L1 キャッシュのヒット率も上がる) 傾向になっている．なお，有効データ数/ラインが 0 になったとしても L1 キャッシュヒット率は 0 には落ちず，20%強の値を示しそうな傾向が読み取れる．これは，列ベクトル以外にもインデックスや行列値を格納する配列に対するバーストアクセスに対する空間的局所性に起因するヒットがあるためであり，異常な結果ではないと考えられる．

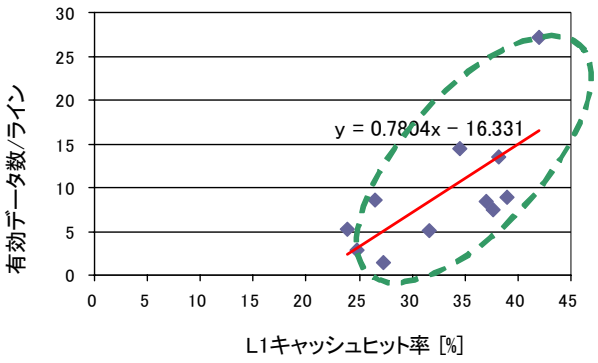


図 7　相関があるサンプルに対する L1 ヒット率と有効データ数/ラインの関係(Fold 法前処理後)

Figure 7　The relation between L1 cache hit rate and the number of valid data/line (Pre-processed by Fold-method)

図 4 を以上のように 3 つのグループに分けて示したのが図 8 である．図 7 に示したサンプルに対応する図 8 の左側のグループでは青い棒（有効データ数/ライン）と青い折れ線の形状には類似点が多いことがわかるようになった．青い折れ線の形状は，文献[5]が指摘している行数に伴うヒット率低下傾向に加え，各種行列の空間的局所性の差に伴う上下変動が合成されたものと考えられる．

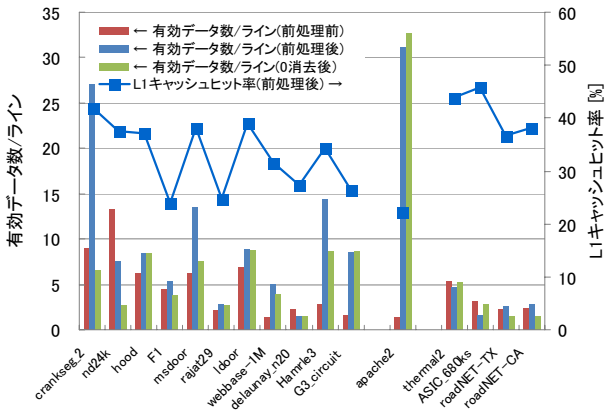


図 8　各種行列における L1 ヒット率と有効データ数/ライン（前処理前=CRS 形式，前処理=Fold 法）

Figure 8　The L1 cache hit rate and the number of valid data/line for various matrices. (Before pre-process = CRS, pre-process = Fold method)

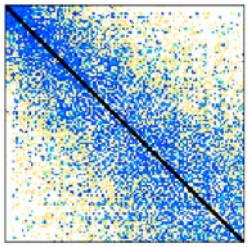


図 9　nd24k の行列形状図

Figure 9　The map of non-zero elements for nd24k

唯一作者不明(アプリ種:2D/3D 問題)な nd24k については，図 9 のように行列形状図上で非零要素が不規則に散らばっているように見えるにもかかわらず，今回測定に用いた行列の中では最もキャッシュライン内部の有効データ数が大きく，CRS 形式と良い相性をもつアクセスパターンであることが判る．つまり，行列形状図の見た目では全く判定できなかった重要な行列の特性を，提案指標によって明快に顕在化できたことがわかる．

GPU 向けの Fold 法は転置を用いているため CRS 形式とはキャッシュに対する相性が正反対の関係にある．その傾向は nd24k において顕著に出ており，元から CRS と良い相性のアクセス順であった nd24k が前処理によって大幅に空間的局所性が崩れている．0 パディングを削除するとそのダメージが顕著であることがわかる．これは例えば Fold 法の効果の評判の良さを聞きつけて，もし nd24k に適用してしまうと逆効果になってしまうが，提案指標を事前に測定しておけば元の CRS 形式を用いる方が高速であろうことは容易に判別できると考えられる．

逆に赤よりも青の棒が伸びている webbase-1M や Hamrle-3 をはじめとする多くの行列で Fold 法前処理が空間的局所性由来のキャッシュヒット率改善に寄与し，結果として高速化が得られる可能性を裏付ける一つの指標であると考えられる．

一方，図 6 で表されているような空間的局所性の低さを補えるほど時間的局所性が高いものでない限り，F1 のように空間的局所性で性能が左右され，CRS でも Fold 法でもどちらも変わらず低い（ 1〜5 程度の）有効データ数に留まるものは，ソフトだけによる高速化には限界がある．そのようなものは，Gather 機構[5][8][9][10]のようなハードウェアによる支援が無いとメモリバンド幅の有効活用が困難であると予想される．

以上のように，提案指標は疎行列が与えられた時に，前処理の選択や，アクセスハードウェアの自動選択に際して，有効な方向性を与える指標であると考えられる．

5. 関連研究

京の上では，疎行列計算を主体にした 2 つのアプリケーション上で，アプリケーション固有の性質を利用した最適化を人手で行っていることが報告されている．例えば 1 行の最大非零要素数を 27 に固定する制約を課した最適化などが行われている[11]．汎用志向のアプローチでは複数のソフト実装の中から実行時に自動で試して選択する自動チューニングの研究も行われている．例えば CPU 上での処理アルゴリズムの選択[12]，GPU 上での行列格納方式の選択[13]基づく研究が報告されている．筆者らの先行研究では行列サイズとキャッシュ容量の関係が疎行列ベクトル積性能に重要な影響があることが示されている[5]．

ただし，筆者らの知る限りでは「列インデックス列の空

間的局所性」を疎行列や疎行列向け前処理アルゴリズムの特性として事前に測定し，疎行列処理の自動チューニングに用いている前例は無い．

6. おわりに

エクサスケールマシンは複雑なメモリシステムとなることが予想されている．同マシンへの適用を視野に入れた疎行列ライブラリの実現に向け，本報告では疎行列のキャッシュへの適合性分類に資する疎行列の特性に関する新しい指標として「列インデックス列の空間的局所性」を提案する．さらに，入力疎行列および Fold 法前処理後の提案指標の値をフロリダ大学の疎行列コレクションを用いて評価した．その結果，疎行列ベクトル積処理性能と L1 キャッシュヒット率と新指標の間には有意な相関関係があることが確認できた．行列形状図では判別できなかった規則性やアルゴリズムとキャッシュの相性が明快に判定できる場合があることがわかった．よって，従来から指摘していた行列サイズと併せ，本指標をアプリ固有の最適化を避けたメモリアクセス機構や前処理アルゴリズム自動選択の指標の一つとする．

ただし，今回の予備実験によれば，行列サイズと列インデックス列の空間的局所性以外にも疎行列ベクトル積性能を左右する別の要因が存在していることも明らかになってきた．今後の課題は，時間的局所性などのその他の疎行列ベクトル積性能決定要因の指標化と分析などである．

謝辞 本研究の一部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) 平木 : "[招待講演]将来の HPC アーキテクチャ", 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 2012 (HPCS'12), pp.163-167, Jan.2012.
2) X. Yang, S. Parthasarathy, P. Sadayappan : "Fast sparse matrix-vector multiplication on GPUs: implications for graph mining", Proc. VLDB Endowment, Vol.4, No.4, pp.231-242, Jan. 2011.
3) Tim Davis : " The University of Florida Sparse Matrix Collection", http://www.cise.ufl.edu/research/sparse/matrices/.
4) 冨森，田邊，小郷，高田，城 : “大規模疎行列やメモリシステムの特性を考慮した高速な汎用疎行列ライブラリ実現に向けて”, 先進的計算基盤システムシンポジウム(SACSIS’12)ポスター, pp.65-66, May 2012
5) 田邊，Nuttapon，中條，小郷，高田，城 : “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”, 情報処理学会研究報告 2011-HPC-132，Nov. 2011.
6) NVIIDIA Corp. : “Whitepaper NVIDIA’ s Next Generation CUDA Compute Architecture Fermi” , http://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf
7) Timo Stich : “Fermi Hardware and Performance Tips”, http://theinf2.informatik.uni-jena.de/theinf2_multimedia/Website_downloads/NVIDIA_Fermi_Perf_Jena_2011.pdf
8) N. Tanabe， Y. Ogawa， M. Takata， K. Joe : " Scaleable Sparse Matrix-Vector Multiplication with Functional Memory and GPUs", Euromicro PDP2011, Feb.2011
9) 田邊，小郷，小川，高田，城 : "Gather 機能を有するメモリアクセラレータの疎行列計算への応用", 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 2012 (HPCS'12), pp.32-41, Jan.2012.
10) 田邊，堀，Nuttapon，中條 : "Gather 機能を有する Hybrid Memory Cube の FPGA を用いた予備評価", 情報処理学会 HPC 研究会, Vol.2010-HPC-133, Mar. 2012.
11) 南，井上，堤，前田，長谷川，黒田，寺井，横川 : "「京」コンピュータにおける疎行列とベクトル積の性能チューニングと性能評価", 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 2012 (HPCS'12), pp.32-41, Jan.2012.
12) 櫻井，直野，片桐，中島，黒田 : "OpenATLib : 数値計算ライブラリ向け自動チューニングインターフェース", 情報処理学会論文誌コンピューティングシステム, Vol.3, No.2, pp.39-47，2010.
13) 久保田，高橋 : "GPU における格納形式自動選択による疎行列ベクトル積の高速化", 情報処理学会 HPC 研究会, Vol.2010-HPC-128, Dec. 2010.