

Xeon® Phi™ の Gather 命令と HMC 内の Gather の差異

富森苑子[†] 田邊昇[‡] 高田雅美[†] 城和貴[†]

1. はじめに

科学技術計算の多くが大規模な疎行列ベクトル積に帰着される。この種の応用ではメモリバンド幅への要求が高い。エクサスケールマシンにおいてはそれにも増して電力制約が厳しい。省電力の観点から、演算よりもデータ移動の電力の大きさが問題とされている[1]。つまり、無駄なデータを長距離移動しないことが望まれる。一方、グラフ処理に代表されるビッグデータ処理が近年注目を集めている。これらも、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り、疎行列処理の大規模化と低電力化と高速化が求められている。Graph500 [2]はビッグデータ処理および疎行列処理のベンチマークであり、エクサスケールマシンとクラウドの両者にとって重要な意味を持つ。筆者らの研究[3]によれば、Graph500 ベンチマークを代表とするグラフ解析処理では、キャッシュライン内に平均1個程度しか有効データがない。このようなアプリケーション実行時に、キャッシュベースのプロセッサがキャッシュから溢れる配列に対して Scatter/Gather を行うと、消費電力と処理性能の両面で深刻な問題が発生する。

本ポスターでは、この問題の解決策として Intel® Xeon® Phi™ [4] のように Scatter/Gather 機能をプロセッサ側に実装するのではなく、Hybrid Memory Cube (HMC) [5]内に実装することの優位性を示す。提案方式の電力や処理性能への効果のモデルを構築しつつ、Graph500 ベンチマーク課題行列に対する疎行列ベクトル積を実例に優位性を考察する。

2. 従来の Scatter/Gather の問題点

Larrabee[6], Xeon® Phi™[4]といった Intel®製 メニーコアプロセッサには、キャッシュとの組合せを想定した Scatter/Gather を行なう命令が搭載されている。図1はその動作を示す。

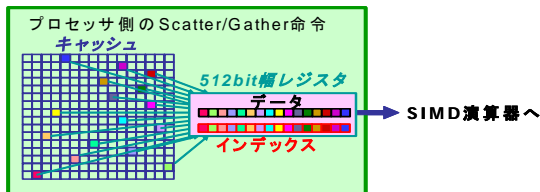


図1 プロセッサ側の Scatter/Gather 命令の動作

しかし、プロセッサ側の Scatter/Gather 命令には図2に示す問題点がある。キャッシュのヒット率が低い状況では、Scatter/Gather 命令の有無ではなく、これらの問題の影響が支配的になる。

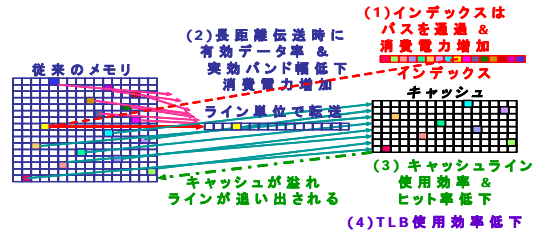


図2 プロセッサ側の Scatter/Gather の問題点

3. 提案メモリスシステム

ランダム性が高い大きなデータの際は、タイリングなどのキャッシュの挙動を十分に意識したソフトウェア的最適化が困難である。特に、ライン内の有効データ数が少ない場合の電力の浪費とヒット率の低下は回避できない。ここで、図3に示すように Gather 機能付 HMC[7]のようなメモリ側の Scatter/Gather 機能により、問題解決を目指す。図1と図3では、Scatter/Gather 機能が実装される場所と、HMC[5]という三次元実装技術の活用有無が構成上異なっている。

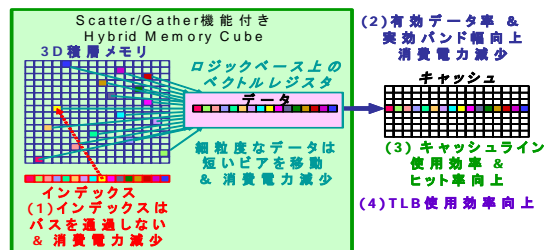


図3 提案メモリスシステムの概要

4. 提案方式の電力や処理性能への効果のモデル

キャッシュライン単位で外部メモリをアクセスする場合と、提案メモリスシステムを用いる場合について、疎行列ベクトル積における処理速度とメモリアクセスに関する消費電力のモデル[8]を配列ごとに Byte/FLOP, ヒット率, ライン内有効データ率や距離に応じた電力を用いて構築し、それらの比を取ることによって効果のモデルが得られる。文献[8]の処理速度モデルおよびメモリアクセスに関する消費

[†] 奈良女子大学
[‡] 株式会社 東芝

電力のモデルから、以下の効果モデルが得られる.

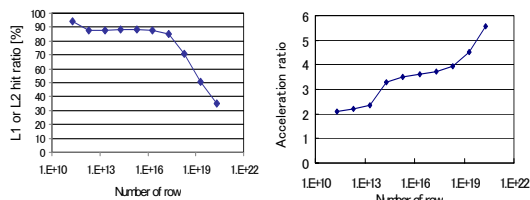
$$\text{加速率} = W_{gather} (2+I/2+(1-hit_x)*128/S)/4W_{cache}$$

$$\text{メモリアクセス節電率} \approx 35(1-hit_{all})$$

ここで W_{gather} は Gather のスループット, W_{cache} は主記憶(GPU の場合はデバイスメモリ)のバンド幅, I はインデックス 1 個のバイト数, hit_x は列ベクトル x へのアクセスのヒット率, S はライン内 32 個中の平均有効データ数, hit_{all} は全アクセスのヒット率である. 節電率においては長距離転送電力 \gg 短距離転送電力および $S \approx 1$ という近似が入っている.

5. 評価

Graph500 用疎行列の SCALE を 11 から 20 ままで変化させ、疎行列ベクトル積を 100 回実行した際の実行時間と L1・L2 キャッシュヒット率をプロファイラにより測定した. インデックスおよび配列データ型は 4 バイトとした. 現段階では Xeon® Phi™ が手元にないため、キャッシュベースのメニーコアという共通の性質を有する GPU Tesla C2050(L1 キャッシュ: 16KB, L2 キャッシュ: 768KB) 上で予備評価を行った. デバイスメモリ上にベクトルの Gather が完了している状態に初期化し、連続アクセスにより配列値とベクトル値を取得して計算を行うプログラムで提案方式の実行時間を模擬した. 図 4 に GPU 向けに前処理された疎行列ベクトル積実行時の L1 または L2 へのヒット率と、提案方式による加速率を示す.



(a)L1 or L2 ヒット率 (b)提案方式による加速率

図 4 Graph500 用疎行列の疎行列ベクトル積特性

加速率のモデル式に SCALE=20 におけるヒット率 35.1%, $I=4B$, $W_{cache}=144GB/s$, $W_{gather}=W_{cache}/2$, 文献[3]で報告されているライン内の平均有効データ数 $S=1.2$ を代入すると、加速率は 11.8 倍となる. これは、図 4(b)の実測値 5.76 倍の約 2 倍である. このずれは、提案 HMC 用の計測プログラムが GPU 用で IF 文排除のための 0 パディングによって本来の 2 倍程度にメモリアクセス回数が増えていることによる. キャッシュ方式では 0 パディングはキャッシュヒットとなり外部メモリアクセスは増えない. これを考慮すれば妥当な結果が得られていると考えられる. Xeon® Phi™ と提案方式を組み合わせる場合は、IF 文のダメージが GPU ほど顕著でないため、0 パディングおよびそのダメージは回避可能である.

一方、節電率の式に SCALE=20 の L1 または L2 キャッシュにヒットする率 35.1% を代入すると、20.7 倍の差が出ると予測される. 図 4(a) が示す通り、右端の SCALE=20 では L1・L2 キャッシュヒット率の減少は鈍っておらず、行列サイズが大きくなるとさらに加速率や節電率が上がる余地が大いにあると考えられる.

6. まとめ

Scatter/Gather を Xeon® Phi™ のようにプロセッサ側で行なうことに伴う問題点とその解決策を示した. Scatter/Gather を提案 HMC 上で動作させることによる疎行列ベクトル積の処理速度と消費電力の効果モデルを整理した. GPU 上での実験の結果、SCALE=20 の Graph500 用課題行列の加速率は Xeon® Phi™ のように IF 文実行に大きなダメージが無いプロセッサと組合わせた場合、十倍を超える大きなものになると予測された. 節電率は 20.7 倍の差が出ると予測された. より大きな行列ではさらに差が開くと予測される傾向が観測された. 以上から、Scatter/Gather 機能は HMC 側に設けるべきであると考えられる. 今後の課題としては、Xeon® Phi™ の実機を用いた評価や、より大きな行列に対する評価などがある.

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] W. J. Dally : "Power, Programmability, and Granularity: The Challenges of ExaScale Computing", IPDPS2011 keynote (2011).
- [2] Graph500 : <http://www.graph500.org/>.
- [3] 田邊, 富森, 高田, 城 : "疎行列のキャッシュ適合性に基づく Graph500 ベンチマークの特性解析", 情報処理学会研究報告 2012-HPC-138, Feb. 2013.
- [4] George Chrysos : "Knights Corner, Intel's first Many Integrated Core (MIC) Architecture Product ", Hotchips 24, Aug. 2012.
- [5] Micron Technology, Inc. : "Hybrid Memory Cube : Breakthrough DRAM Performance with a Fundamentally Re-Architected DRAM Subsystem", Hotchips 23, Aug. 2011.
- [6] L. Seiler et al. : "Larrabee: A Many.Core x86 Architecture for Visual Computing", ACM Trans. Graph. 27, 3, Article 18, Aug. 2008.
- [7] 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", 情報処理学会研究報告 2012-HPC-133, Mar. 2012.
- [8] 田邊, 富森, 高田, 城 : "Scatter/Gather をどこで行なうべきか? - 低電力で高スループットなランダムアクセスのために -", 情報処理学会研究報告 2013-ARC-204, Mar. 2013.

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