

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

富森苑子[†] 田邊 昇[‡] 小郷絢子[†] 高田雅美[†] 城和貴[†]

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。

現時点での世界最高速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、選拔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。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。

本ポスターでは、エクサスケールマシンを視野に入れたライブラリの高度化に関する一研究アプローチを紹介する。具体的には、アプリケーション固有の最適化を避けた、統一的調整機能やアクセス機構選択により、行列種類やサイズへの優れた適応性を有する疎行列ライブラリを実現するための予備検討について述べる。

2. 課題

疎行列ライブラリでは演算の最適化よりもメモリアクセスの最適化の重要性が高い。なぜなら、演算性能に対して実効メモリバンド幅が十分に確保

できず、ここがボトルネックとなって疎行列計算の性能が決まるためである。キャッシュベースの CPU や GPU 上では疎行列計算で典型的な間接参照のアクセスパターンから実効バンド幅が大幅に低下してしまう。エクサスケールマシンにおいてはバンド幅不足の傾向が一層高まる上に、メモリシステムの複雑さの増大に伴い、最適化の困難度も一層高まる。扱うべき行列サイズも増加するため、キャッシュ容量の限界性も性能低下方向に作用する。

3. 従来手法

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[2]。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[3]や、GPU 上での行列格納方式の選択[4]に基づく研究が報告されている。

4. 提案手法

4.1 統一的調整機能

統一的調整機能とは、行列に応じ個別対応や、複数の実装の中から適した実装を選ぶという従来の手法ではなく、1 個の方法によって全ての行列に対して最適化がなされるタイプの自動調整機能である。提案する疎行列ライブラリは以下の 2 つの統一的調整機能を用いることを想定する。

● 折り目の最適化による負荷分散調整

行列を行方向に圧縮後、一行あたりの非零要素数の平均と分散の関数で与えられる閾値を基に、長い行を折り畳むことで疎行列計算の GPU などの並列計算時の負荷分散を調整する方法[5]。折り目を決定するパラメータをライブラリが前処

[†] 奈良女子大学

[‡] 株式会社 東芝

理実行時に自動調整する。

● 間接参照のハード(Gather)による連続化

間接参照におけるメモリアクセス列におけるアドレスの不連続性を, **Gather** 機能を有するメモリシステムがバッファ上に連続化することで, 非零要素配置の多様性の影響を排除する方法[6]. ソフトだけでは十分に解消できないメモリバンド幅の浪費を防止できる. 疎行列ベクトル積においてこの機構をユーザから隠蔽した状態で用いるための関数をライブラリが提供する.

4.2 アクセス機構選択

従来の疎行列処理の自動チューニングがソフト実装を選択するのにに対し, アクセス機構選択は複数のハードの中から実行時に選択する方法の一種である. 疎行列処理はメモリアクセスがボトルネックになりがちであるため適切なメモリアクセス機構(例えば **Texture** キャッシュ/L1&L2 キャッシュ/Gather 機構)を選択することが重要になる. ベクトルサイズとキャッシュのサイズの大小関係によっては前処理時間も含めた全体の処理時間は単純にキャッシュを用いた方が **Gather** 機構を用いるより高速な場合もありうる. **Gather** 機構は事前の準備が必要だが, ベクトルサイズが大きいほどその影響が消えて行くため大きい問題に適する. 逆に, キャッシュは単純にアクセスするだけなので事前の準備は不要だが, 限られたキャッシュ容量をベクトルサイズが超えると性能劣化が激しくなるので小さな問題に適する.

5. 予備評価

単精度の疎行列ベクトル積における **L1** キャッシュのヒット率と, 処理性能の行列サイズとの関係について評価した. 後者は **L1/L2** キャッシュおよび **Gather** 機構の 2 つのメモリアクセス機構について評価した. 測定環境は GPU として **Nvidia Tesla C2050** (CUDA コア数 448, メモリバンド幅 144GB/s)と **CUDA Version 3.2** を用いた. 評価用疎行列はフロリダコレクション[7]から抜粋した.

その結果, **L1** ヒット率(図 1)は行列サイズが大きくなるにつれて低下する傾向が確認された. **L1/L2** キャッシュおよび **Gather** 機構の 2 つのメモリアクセス機構によるカーネル部の処理性能(図 2)については, 前者の性能(青点)は **L1** ヒット率(図 1)と強い相関があり, 行列サイズが大きくなるにつれて低下していく. これに対し, 後者(赤点)はサイズによる変動幅が少なく, ベクトルアーキテクチャの一般的傾向と同様に増加する傾向が確認された.

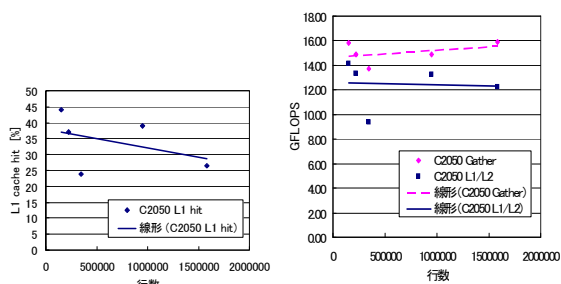


図 1 L1 ヒット率と行数

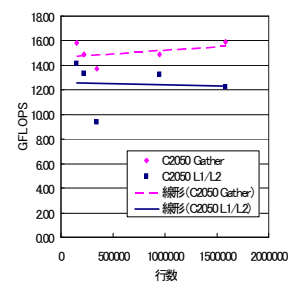


図 2 処理性能と行数

6. おわりに

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た. 予備評価の結果, キャッシュは行列が大きくなると性能劣化し, **Gather** 機構は安定している. 自動チューニングにおいて適切な選択を行う上での重要な指針を得た. 今後はエクサスケールマシンを視野に入れた, 行列種類やサイズへの優れた適応性を有する疎行列ライブラリの開発を行う予定である.

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] 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.
- [3] 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.
- [4] 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.
- [5] N. Tanabe, Y. Ogawa, M. Takata, K. Joe : "Scaleable Sparse Matrix-Vector Multiplication with Functional Memory and GPUs", Euromicro PDP2011, Feb.2011
- [6] 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.
- [7] Tim Davis : "The University of Florida Sparse Matrix Collection", <http://www.cise.ufl.edu/research/sparse/matrices/>.