

MIC アーキテクチャ上での Hierarchical Diagonal Blocking を利用した疎行列ベクトル積の性能評価

山口翔[†] 藤井昭宏[†] 田中輝雄[†]
工学院大学[†]

1. はじめに

大規模数値計算には疎行列ベクトル積 (SpMV) が使用される. SpMV の計算には一般的に CRS (Compressed Row Storage) 形式が使用されている. CRS で SpMV を計算する際, ベクトルへの非連続アクセスが原因でキャッシュヒットミスが起こるとい問題がある.

行列を分割, リオーダーリング, 階層的にブロック化することでこの問題を改善できる Hierarchical Diagonal Blocking (HDB) [1]が提案されている. HDB では分割の数と階層の数を指定する必要がある. 分割数を幅, 階層の数を深さという. 行列ごとに最適な幅と深さは異なり, 最適な値で計算を行う必要がある. 幅についての研究は文献[3]で行われている. 深さについての研究は深さ2まで文献[4]で行なわれている.

一方, 60 コア 240 スレッドを持つ Many Integrated Core Architecture (MIC) がある [2]. MIC のような高並列環境では, 240 スレッドを最適な形で並列化することによってパイプラインを有効利用する必要がある.

本研究では, HDB を利用した SpMV の効果について, MIC 上で格納形式が性能に与える影響を検証する.

2. CRS を用いた SpMV

CRS は疎行列を格納する形式のひとつである. 各行における非零要素の値が入る開始位置を示す index, 非零要素の列番号を表す col, 非零要素の値を格納する val の3つの値で表す. CRS では, ベクトルxの参照に非零要素の列番号を使用するため, ランダムアクセスとなり, キャッシュヒット率が悪くなる問題点がある. 図1に CRS の構造と CRS を用いた SpMV のイメージ図を示す.

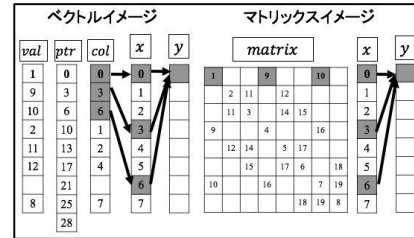


図1 CRS を用いた SpMV

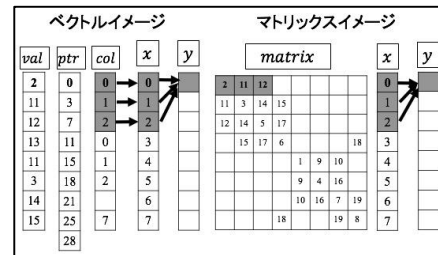


図2 HDB を行った場合の SpMV

3. HDB を用いた SpMV

HDB は, 疎行列の非零要素が対角に集まるように分割, リオーダーリング, 階層的にブロック化することで, CRS の問題点であるベクトルxの非連続アクセスを改善する手法である (図2).

4. MIC 上での適応化

MIC のような 60 コア 240 スレッドの高並列環境においては, HDB によって分割された各階層のブロックを処理する方法が重要である. 各階層ブロックの処理では SpMV を行うスレッドのスレッド ID を指定することによって複数の小行列を並列計算する手法であるタスク分割がある.

本研究では各ブロックを HDB でそれぞれ 240 スレッドで行った場合 (HDB240) とタスク分割を行った場合 (HDB2-2-4) について比較を行った.

表1 環境変数

KMP_PLACE_THREADS	60C,4T
KMP_AFFINITY	compact, fine
OMP_SCHEDULE	static
KMP_BLOCKTIME	infinite

Performance Analysis of Sparse Matrix-Vector Multiplication using Hierarchical Diagonal Blocking on MIC Architecture
Sho Yamaguchi[†], Akihiro Fujii[†] and Teruo Tanaka[†]
[†]Kogakuin University

表 2 疎行列一覧

疎行列名	nnz	size N	nnz / row
Dubcova2	1030225	65025	15.8
audikw_1	77651847	943695	82.3
inline_1	36816342	503712	73.1

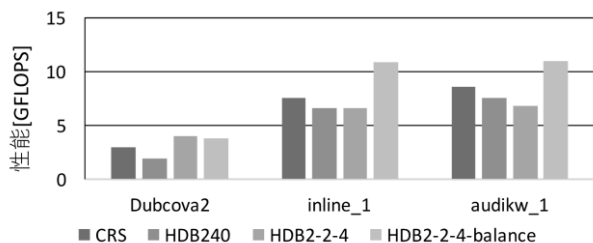


図 3 疎行列の性能比較

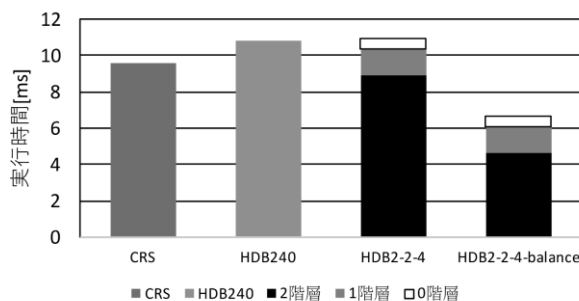


図 4 inline_1 についての各階層の実行時間

5. 負荷分散による影響

HDB のブロック分割を行い、各スレッドに均等に疎行列の行数を割り当てると負荷分散が均一にならない。そのため、非ゼロ要素数の均等分割による負荷分散を HDB2-2-4 について行った場合 (HDB2-2-4-balance) も比較した。

6. 数値実験

6.1 実験環境

実行環境は Intel Xeon Phi 5110P, 1.053GHz, 60 コア, キャッシュは 30MB (60core×512kB), コンパイラは Intel ICC 15.0.0 を使い, コンパイルオプションは“-mmic -O3 -openmp”を用いた。

本研究はノードでの性能評価のため MIC 上で直接プログラム実行する native モデルを使用した。

環境変数は, The Univ. Florida Sparse Matrix Collection (フロリダコレクション) [5]の疎行列で CRS による SpMV の結果から平均的に高い性能が得られた変数を利用した (表 1)。

実験に使用する疎行列は, フロリダコレクションの対称行列問題から選択した (表 2)。

6.2 格納形式の違いによる SpMV の性能

HDB のパラメータは, 深さ 2, 幅は 1 階層が幅 2, 2 階層が幅 4 の場合に関して計測した。各疎行列で CRS, HDB240, HDB2-2-4, HDB2-2-4-balance を行った SpMV の性能比較を図 3 に示す。また, SpMV の inline_1 についての各階層の実行時間を図 4 に示す。

図 3 から, すべての疎行列で HDB2-2-4-balance を行った場合の SpMV が最も良い性能を得られた。inline_1 は CRS による SpMV と比べて 1.4 倍の性能向上が見られた。負荷分散を行うことによってすべての疎行列で, HDB による分割の性能向上が見られた。

図 4 から, 負荷分散により, inline_1 の 2 階層部分では 49% の実行時間の削減ができた。HDB 全体についての実行時間は 44% 削減したことを確認した。

以上の結果より, HDB を行うことによるキャッシュヒット率の性能向上を得ることができた。HDB を行い, 負荷分散を行うことで, 性能向上が見られた。

7. おわりに

HDB について, CRS と比較することによって以下のような結果を得ることができた。

HDB を利用し, 負荷分散を行うことにより, CRS での SpMV と比較して最大 1.4 倍の性能向上が見られた。性能向上の理由としては, 対角ブロックに要素が集まったことにより, キャッシュヒット率が向上したためと考えられる。

今後の課題として, HDB の最適なパラメータの検証や負荷分散のさらなる検証がある。

参考文献

- [1] Guy E. Blelloch, et al., Hierarchical Diagonal Blocking and Precision Applied to Combinatorial Multigrid, Super Computing 2010 (2010).
- [2] Intel® Xeon Phi™ Coprocessor | Intel® Developer Zone, <http://software.intel.com/en-us/mic-developer>, 2017.1.6 閲覧。
- [3] 花上直樹, 佐々木信一, 菱沼利彰, 藤井昭宏, 田中輝雄, Hierarchical Diagonal Blocking を用いた疎行列ベクトル積の特性評価, 第 77 回情報処理学会全国大会 (2015).
- [4] 泉花穂, Hierarchical Diagonal Blocking を用いた疎行列ベクトル積の深さに注目した特性評価, 工学院大学 2015 年度卒業論文
- [5] The University of Florida Sparse Matrix Collection, <http://www.cise.ufl.edu/research/sparse/matrices/>, 2017.1.6 閲覧。