

Hierarchical Diagonal Blocking の階層構造に着目した 疎行列ベクトル積の性能評価

室療[†] 藤井昭宏[†] 田中輝雄[†]
工学院大学[†]

1. はじめに

疎行列ベクトル積(SpMV)は大規模数値シミュレーションにおける反復解法に用いられる。

疎行列の格納には CRS(Compressed Row Storage)形式が広く用いられている。CRS 形式を用いた SpMV の問題点は、ベクトル部への参照が非連続であるため、CPU キャッシュミスが発生し、性能が低下することである。

そこで我々は HDB(Hierarchical Diagonal Blocking)形式[1]に着目した。

HDB 形式は疎行列に 1.パーティショニング 2.リオーダーリング 3.ブロッキングを再帰的に繰り返し、階層的にブロック化する。HDB 形式を用いた SpMV の性能を決めるパラメータには階層の深さ d , 1 階層の幅 w (1 階層が持つブロック数)がある。

HDB 形式の SpMV は問題によって効果があるが[2], 浅い階層で並列度が下がり性能が律速される問題があった。そこで我々は浅い階層にワークベクトルを持たせ並列度を保つ手法を提案する。本研究では深さ, 幅パラメータとワークベクトルが SpMV の性能にどのように影響するか調査する。

2. CRS 形式

疎行列は成分のほとんどが零で構成される。CRS 形式は非零要素の値を格納する val , 列番号を記録する $index$, $index$ の行が変わる位置を記録する row_ptr の 3 つの配列で疎行列を格納する。

CRS 形式の SpMV は, $y = Ax$ のベクトル x への参照が非連続になる(図 1)。そのため, x のアクセスが CPU キャッシュの領域外であるときキャッシュミスが発生し性能が低下する。

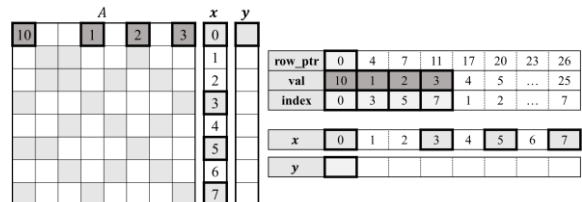


図 1 CRS 形式を用いた SpMV の例

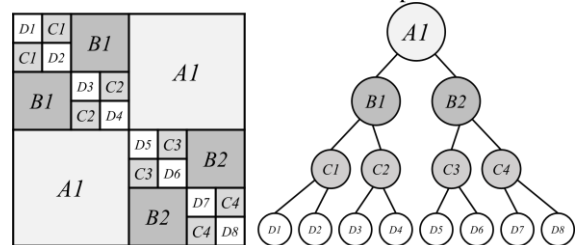


図 2 HDB 形式の木構造

3. HDB 形式

HDB 形式は疎行列の非零要素をリオーダーリングで対角上に集めてベクトル x のアクセス連続性を高め, さらにブロック化によるキャッシュブロッキングの効果で x のキャッシュ再利用性を高め性能を向上させる格納形式である。

疎行列のリオーダーリングには Metis ライブラリ[3]を使用した。

HDB 形式を深さ 3, 各階層の幅を 2 とした時には図 2 のような木構造になる。

4. 対称疎行列向け SpMV の並列化

対称性を利用し対角要素と上三角成分のみ保持した CRS 形式の SpMV は, 行分割で並列化するとベクトル y に書き込み競合が起こる。これを解決するためにはスレッドごとにワークベクトルを保持しなければならない。

HDB 形式を用いた SpMV[2]は, 木構造の各ノードのブロックを 1 つのタスクとして扱うことで並列化する。子ノードの計算が完了してから自らのノードの計算を行うことで, y への書き込み競合を排除できる。ただし, この手法では浅い階層で並列度が下がり性能が律速される問題があった。

Performance Analysis of Sparse Matrix-Vector Product focusing on Hierarchical Structure of Hierarchical Diagonal Blocking
Ryo Muro[†], Akihiro Fujii[†] and Teruo Tanaka[†]

本研究では浅い階層に対して階層ごとにワークベクトルを持たせ、子の計算を待たない独立したタスクとすることで並列度の低下を防ぐ手法を提案する。

5. 数値実験

5.1 実験環境・条件

対角を含む上三角成分のみ保持した対称疎行列の SpMV の性能について CRS 形式と HDB 形式で比較を行う。

CRS 形式は行単位で並列化し、書き込み競合の解決のためにコア数分のワークベクトルを用意している。

HDB 形式は前述の方法で並列化する。従来手法であるワークベクトルを持たない HDB 手法 1, 根にワークベクトルを持つ HDB 手法 2, 根と深さ 1 のノードにワークベクトルを持つ HDB 手法 3 の 3 パターンの計測を行う。

HDB のパラメータは深さを 1~4, 幅は末端の階層で 2,4,8,16,20,32 でそれ以外では 2,4,8 にそれぞれ変化させ計測を行う。

計測環境は表 1 に、使用する疎行列の名称, 非零要素数(NNZ), 行数(ROW)などの一覧を表 2 に示す。疎行列は The University of Florida Sparse Matrix Collection[4]の疎行列を使用した。

表 1 実験環境

CPU	Xeon E5-2650v3	L2 Cache	256KB
コア数	10 *2(CPU)	L3 Cache	25MB
周波数	2.3GHz	メモリ容量	64GB
L1 Data Cache	32KB	コンパイラ	icc
コンパイルオプション		-O3 -qopenmp	

表 2 実験に使用した疎行

名称	ROW	NNZ	NNZ/ROW
audikw_1	943695	39297771	41.6
Dubcova2	65025	547625	8.4
ecology2	999999	2997995	3.0
G3_circuit	1585478	4623152	2.9
nd24k	72000	14393817	199.9
pwtk	217918	5926171	27.2

表 3 HDB 最高性能時パラメータ

疎行列名	手法	深さ	幅 1	幅 2	幅 3	幅 4
audikw_1	3	4	4	8	2	2
Dubcova2	3	3	4	8	4	
ecology2	2	1	8			
G3_circuit	2	3	2	2	20	
nd24k	3	4	2	2	4	2
pwtk	3	4	4	8	4	2

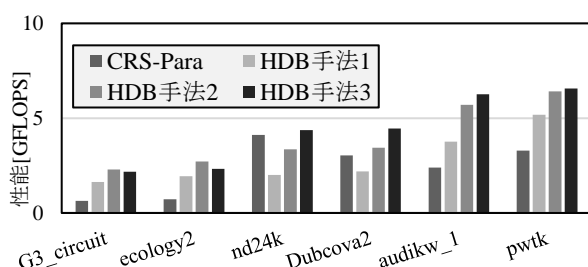


図3 SpMVの性能

5.2 実験結果・考察

実験で得られた性能を図 3, HDB 形式を用いた際の最高性能時パラメータを表 3 に示す。

HDB 形式は手法 2 で 2 種類, 手法 3 で 4 種類の疎行列が最高性能となった。従来手法である手法 1 より総じて性能が向上した。

最高性能時パラメータは疎行列により異なった。これは疎行列の非零要素の分布, マシン性能, ワークベクトルの本数によって最適な深さ, 幅が大きく変化するためだと考えられる。

Dubcova2,nd24k では, HDB 形式を用いた優位性が低かった。CRS 形式の SpMV ではコア数分のワークベクトルを保持しているが, 数万自由度の疎行列はマシンの持つ 20 コア分のワークベクトルが L3 キャッシュに収まりきるため, CRS 形式の欠点であるキャッシュ効率の低下が起こりにくかったのだと思われる。

6. まとめ

本研究では HDB 形式は行数が大きい疎行列に特に有効であった。これは疎行列をブロック化することによるキャッシュブロッキングの効果と, CRS 形式よりも多くのワークベクトルを保持しない利点によるものと思われる。

本研究では, HDB 形式の問題点であった浅い階層の並列度の低下に対し, 浅い階層にワークベクトルを持つことで並列度を保つ手法の効果認められた。HDB 形式のワークベクトルを持たない従来手法では CRS 形式と比較しおおよそ 0.5~2.7 倍の性能なのに対し, ワークベクトルを用いることでおおよそ 1.1~4.2 倍の性能になった。

今後の課題としては数千万自由度を超える大規模な疎行列への HDB 形式の適応が考えられる。

参考文献

- [1]. Guy E. Blelloch, et al., Hierarchical Diagonal Blocking and Precision Applied to Combinatorial Multigrid, Super Computing 2010 (2010).
- [2]. 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性能評価, 2015 年度工学院大学情報学部卒業論文(2016.3).
- [3]. METIS, <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [4]. The Univ. of Florida Sparse Matrix Collection, <http://www.cise.ufl.edu/research/sparse/matrices/>.