分散メモリ型並列計算機 AP1000 上への疎行列用 BLAS-3 の実装

3Q−2

内野 聡, 萩原 純一, 安江 俊明, 山名 早人, 村岡 洋一
早稲田大学

1 はじめに

疎行列を対象とした BLAS-3(Basic Linear Algebra Subroutine Level 3) を並列化し、富士通の分散メモリ型並列計算機 AP1000 上に実装した。BLAS-3 は、次のサブルーチンで構成されている。

- 行列の積 (GEMM, SYMM)
- 対称行列に対する階数 k と $2k$ の更新 (SYRK, SYR2K)
- 三角行列との積 (TRMM)
- 右辺が複数列で、三角行列を係数に持つ連立一次方程式 (TRSM)

このうち TRSM 以外の 5 つは、行列同士の積が主演算である。そのため、実装にあたって疎行列の積の並列化方法が重要になる。

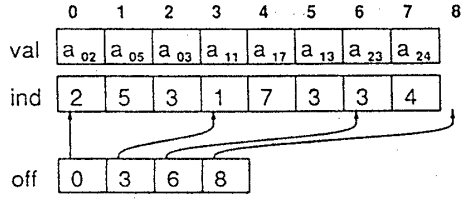
密行列の積の並列化と比較して、疎行列の積の並列化においては、次の点が問題になる。

1. 疎行列の積 $C \leftarrow C + AB$ の計算においては、疎行列が圧縮されて格納されているため C の書き換えに時間がかかる。
2. 後述する方法で行列をセルへ分割した時、各セル (PE) の持つ部分行列の大きさに偏りがあるために、その格納に必要なメモリと通信量に偏りが生じる。

そこで、本稿では 1 の点を解決するための計算の実行順序について提案し、その計算順序で実行した場合における 2 の点を解決するための通信方法について提案する。さらに、提案した方法に基づいて実装した GEMM(非対称行列同士の積) ルーチンを用いて評価する。

なお、疎行列は以下の条件で格納されているものとする。汎用的なプログラムとするために、一般的に使われている方法を選択した。

1. x- 方向のセル台数 N_x と、y- 方向のセル台数 N_y が等しい。
2. 行列のセルへの分割は、ブロック分割や巡回的分割を一般化した、ブロックを巡回的に配置する方法 ($\bar{m} \times \bar{n}$ "blocked panel-wrapped"strategy [3]) を用いる。具体的には、要素 a_{ij} をセル $((j/\bar{n}) mod N_x, (i/\bar{m}) mod N_y)$ に割り当てる。
3. 各セルごとに割り当てられた行列は、図 1 のように、1 行単位で 1 次元配列に格納する [2]。



val: 要素の値, ind: 要素の列番号, off: 各行の先頭位置

図 1: 疎行列の格納法

2 計算の実行順序

疎行列の積 $C \leftarrow C + AB$ を実行するアルゴリズムを図 2 に示す。これは、SPARSKIT[2] の BLASSM を元にした。

C の非ゼロ要素の増加数が予測できないため、古い C から新しい C へコピーしながら 1 行単位で C を書き換える。そのため、C の書き換えに時間がかかる。

図 2 に示した計算を並列実行する方法としてブロック単位の参照法 (図 3) が考えられる。図 3のマス目が 1 つのセルに対応し、C の網掛け部分のセルが A,B の網掛け部分のセルに割り当てられた 1 から 3 までのブロックを 1 組ずつ参照する。

```
C を CC にコピーする;
C をクリアする;
for (i = 0; i < C の行数; i++) {
    CC の i 行を C の i 行へコピーする;                /* (a) */
    for (ll = 0; ll < A の i 行目の要素数; ll++) {
        ail = A の i 行目の ll 番目の要素;
        l   = A の i 行目の ll 番目の列番号;
        for (jj = 0; jj < B の j 行目の要素数; jj++) {
            blj = B の l 行目の jj 番目の要素;
            j   = B の l 行目の jj 番目の列番号;
            C の要素 ij に ail * blj を加算する;   /* (b) */
        }
    }
}
```

図 2: 疎行列の積 $C \leftarrow C + AB$ の計算

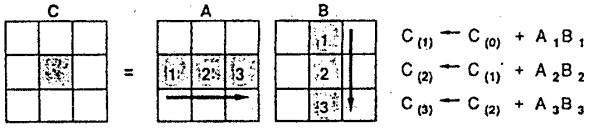


図 3: ブロック単位の参照法 (3 × 3 セルの場合)

ブロック単位の参照法では、1 度に A,B の部分行列のコピーを 1 組持っていれば十分である。また、A と C への参照が連続的であるが、C を N_x 回書き換える。

しかし図 2 に示した計算においては、古い C から新しい C へ 1 行ずつコピーする処理 (図 2 の (a)) に、密度 1% の時で処理の約 10〜20% の時間 (実測値) がかかる。C は次第に非ゼロ要素が増加するため、この処理時間は書き換えを繰り返していくうちに増大する。

そこで、C の書き換え操作を 1 回で済ませるために 1 行単位の参照法を提案する (図 4)。この方法では、C と A を 1,2,...,m の順に 1 行ずつ参照しながら、1 回で C を書き換える。

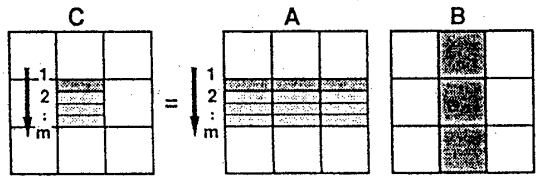


図 4: 1 行単位の参照法 (3 × 3 セルの場合)

1 行単位の参照法では、C の濃い網掛け部分の要素を 1 行書き換えるために、A と B の濃い網掛け部分の要素を参照する。そのためブロック単位の参照法に比較してメモリを大量に必要とするが、C の書き換え操作が 1 回で済むため高速に実行できる。

3 部分行列の共有方法

前節で述べた 1 行単位の参照法において、A を 1 行ずつ通信すると、通信の粒度が小さくなるためオーバーヘッドが増加する。そこで本実装では、参照する部分行列は通信により最初から共有する。ここで、共有とはセル同士で互いにコピーを持ち合うこととする。

本節では、参照する部分行列を格納するためのメモリ確保の方法と部分行列の通信方法について述べる。

3.1 メモリ確保と通信順序

部分行列のコピーを持つ時、非ゼロ要素数分だけのメモリを確保すれば十分である。非ゼロ要素数は図 1 における offを参照することにより得ることができる。

よって、受信側は offを受信後に ind と val のためのメモリを確保し、続いて ind と val を受信することで、非ゼロ要素数分だけのメモリを確保することができる。

An Implementation of Sparse BLAS-3 on a Distributed Memory Parallel Machine AP1000.
UTINO Satosi, HAGIWARA Junichi, YASUE Toshiaki,
YAMANA Hayato, MURAOKA Yoichi
Waseda University

3.2 通信方法

行(または列)方向のセル同士で A(または B)を共有するための通信方法として、次の2つが考えられる。

1. 1対1通信を用いたバケツリレー方式(図5)
2. 行(または列)方向への放送機構を用いた逐次放送方式(図6)

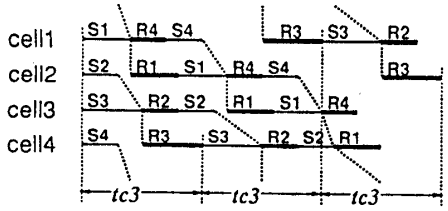


図5: バケツリレー方式の通信パターン例

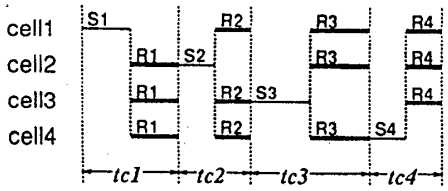


図6: 逐次放送方式の通信パターン例

セル4台の場合の例を図5,6に示す。ここで、各セルの持つ部分行列の送信から受信までにかかる時間を $tc1$,$tc2$,$tc3$,$tc4$ とした。また、Sn,Rn はセル n の持つ部分行列の送信と受信の処理を表す。

本実装では以下の理由より、逐次放送方式を採用する。

1. 図5,6より明らかな通り、通信の回数はバケツリレー方式の方が1回分少ない。しかし、各セルの持つ部分行列の大きさの偏りが大きい場合、バケツリレー方式では最も大きい部分行列の通信に実行が律速されるため、逐次放送方式の方が速くなる。
2. 3.1節で述べたとおり、必要十分なメモリを確保するためには、offを受信後にindとvalのためのメモリを確保する。その場合、逐次放送方式では、送信とメモリ確保を並行して実行できる。また、部分行列を3度に分けて通信する結果、送信と受信をパイプライン的に並行して実行できる。一方、バケツリレー方式では、送信の回数が全体として多くなるために、これらを並行して実行可能な部分が少ない。
3. 部分行列が大きく、各セルの持つ部分行列の大きさに偏りがあると、バケツリレー方式では、同期を取らなければ部分行列を受信し終わらないうちに次の部分行列が到着してしまい、受信バッファが溢れる可能性がある。

4 評価

以上述べた方法を、GEMM(非対称行列同士の積)の行列の転置を含まないもの($C \leftarrow \alpha AB + \beta C$)を用いて評価した。使用したAP1000は8×8の64台構成である。

並列化に伴うオーバーヘッドを調べるために、台数の変化に対してプロセッサ1台当たりの行列計算の処理量が一定となるように次の条件でデータを取った。なお、行列 A,B,C の大きさをそれぞれ $M \times K$,$K \times N$,$M \times N$ とする。

1. $M/N_x = N/N_x = K =$ 一定
2. A,B,C は一様乱数を用いて生成した密度1%の疎行列。

評価結果を図7,8に示す。比較のために、ブロック単位の参照法による結果を破線で示す。

図8の実線において、2台以上の時と1台の時の差は、部分行列の共有のための通信時間によるものである。台数が増えても行列に対する処理の時間を一定に保つことが出来ているため、台数分だけの速度向上が得られる。一方、ブロック単位の参照では、台数が増えるに従って行列に対する処理時間が長くなるため、台数分だけの速度向上が得られない。

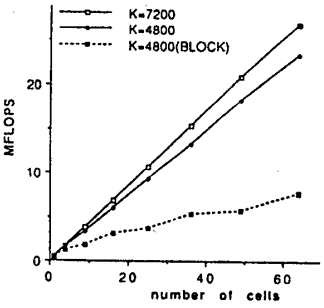


図7: 台数に対する性能

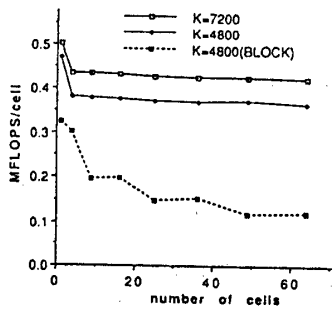


図8: 台数対1台当りの性能

次に、正規乱数を用いて、対角成分の近くに非ゼロ要素が多く分布するように生成した疎行列についての性能を図9に示す。

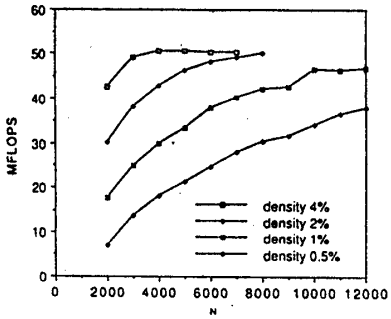


図9: 帯状の疎行列に対する性能(8×8セル)

1行当りの要素数(N×密度)が小さいと、A,B を参照するループの回数が少なくなるため遅くなる。1行当りの要素数が十分大きいとき、64台で最大50MFLOPSを超える性能を得ることができた。これは、AP1000のFPUのピーク性能5.56MFLOPS/cellに対して約15%の性能に当る。行列 C の書き換え操作のオーバーヘッドが大きいため、このような性能に留まっている。今後、このオーバーヘッドの削減を目指したい。

5 おわりに

本稿では、疎行列の積の並列化の方法について述べた。この手法を基本にして、現在BLAS-3のサブルーチンがすべてAP1000に実装され、表1に示す性能を得ることができている。

表1: Sparse BLAS-3の性能(8×8セル, 密度1%)

演算	実行時間[s]	MFLOPS
GEMM($C \leftarrow \alpha AB + \beta C$)	4.29	46.6
GEMM($C \leftarrow \alpha AB^T + \beta C$)	4.54	44.1
GEMM($C \leftarrow \alpha A^T B^T + \beta C$)	4.85	41.2
SYMM($C \leftarrow \alpha AB + \beta C$)	4.51	44.4
SYRK($C \leftarrow \alpha AA^T + \beta C$)	4.23	23.7
SYR2K($C \leftarrow \alpha AB^T + \alpha BA^T + \beta C$).	8.33	24.0
TRMM($B \leftarrow \alpha AB$)	4.22	47.4
TRSM($B \leftarrow \alpha A^{-1} B$)	12.4	79.5

SYMMのA、SYRK,SYR2KのCは下半分のみ格納した対称行列
TRMM,TRSMのAは下三角行列
TRSMの右辺は密行列で10000×500、それ以外は10000×10000

謝辞

本研究に当り、AP1000を使わせて頂いたことを(株)富士通および並列処理センターに深く感謝致します。

参考文献

[1] 小国力 編著: "行列計算ソフトウェア", 丸善 (1991)

[2] Y.Saad: "SPARSKIT: a basic tool kit for sparse matrix computations," ftped from icarus.riacs.edu(1990)

[3] R.P.Brent: "The Implementation of BLAS level 3 on the AP 1000: Preliminary Report," Proc. CAP Workshop 91, ANU (1991)