応用数理工学

5月18日

1　連立1次方程式の解法（密行列の場合）

1.1　線形計算における高性能化の基本的な考え方

共有メモリ型並列計算機における演算の高速化には重要な点が3つある。

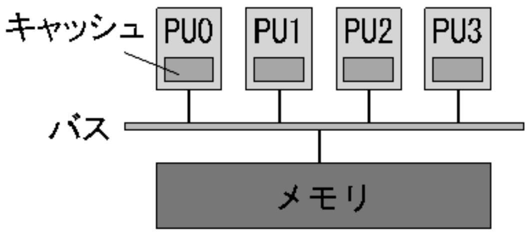


図 1: 共有メモリ型並列計算機

1.1.1　各プロセッサの処理量を均等化

各プロセッサの処理量を均等にすることで，演算が早く終えたプロセッサの待ち時間が減り高速になる（図2）。

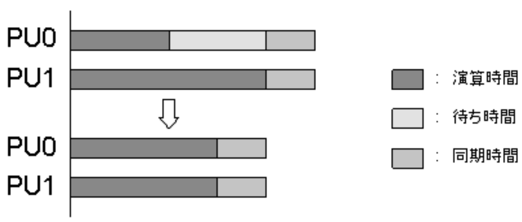


図 2: 処理量均等化

1.1.2　プロセッサ間の同期の削減

図3の例で、PU0で計算した部分和とPU1で計算した部分和を足し合わせるといった場合、別のプロセッサの結果を参照しなければならない。この様な作業を同期と呼ぶ。この同期をとるときはお互いの演算は止まっている。この同期を減らし、各PUが独立に処理できるを大きくすれば、演算を高速にすることができる。

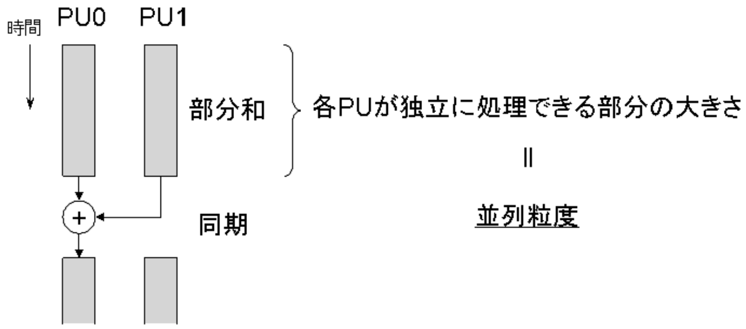


図 3: 並列粒度

1.1.3　キャッシュ・メモリの有効利用

いったんデータをキャッシュに持ってきたら、そのデータに対して必要な計算を集中して行うよう、計算の順序を変更し、データ参照の局所性を高める、また，メモリにあるデータを参照する際に、他のプロセッサとのアクセス競合が起きないようにする。

1.2　BLASの利用による高性能化

1.2.1　BLASの種類

BLAS1: ベクトルとベクトルの演算

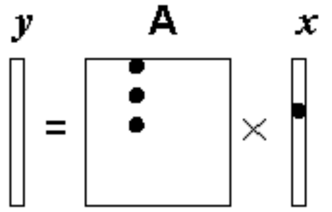
BLAS2: 行列とベクトルの演算

BLAS3: 行列と行列の演算

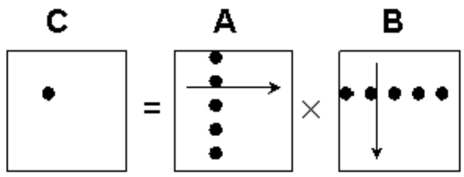
1.2.2　プロセッサ計算機

BLAS1では、内積やAXPYなど演算量が$O(N)$となる演算を行う．BLAS1ではx_iやy_iが1度しか使われないためデータの再利用性がなく、演算量の削減はしにくい。

BLAS2では、ベクトルデータのみに再利用性があり、BLAS1よりは高速である。しかし、1度しか使われない行列Aがあるので、キャッシュでの局所性が活かされず高速化しにくい。



BLAS3では$O(N)$のデータ再利用性が可能である。

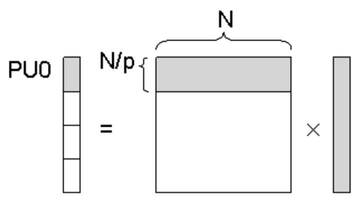


よって、1プロセッサの場合、BLAS3をうまく使えば演算を高速にすることができる。

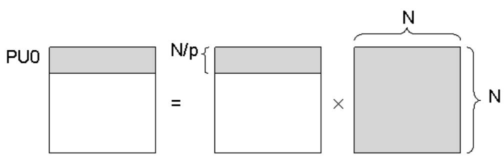
1.2.3 並列計算機

BLAS1の演算量は1プロセッサの演算量と比べ、プロセッサ数pの分だけ分割されるので$O(N/p)$となる。しかし、ベクトルの要素が1度しか使われないのは変わらないので、高速化しにくい。

BLAS2の演算量もBLAS1と同様に、演算量がp分割される。しかし、これも行列の要素が1度しか使われないので高速化しにくい。



BLAS3では、行列の要素が再利用するので局所性が高めることができ、演算を高速にすることができる。



よって、並列計算機でも、BLAS3が高速化に有効である。プログラムではBLAS3が使用できるように改良するべきである。

2　ガウスの消去法とLU分解

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

$$A\vec{x} = \vec{b}$$

$A = LU$ と分解
L:対角要素が 1 の下三角行列
U:上三角行列

$$A\vec{x} = \vec{b} \Longleftrightarrow LU\vec{x} = \vec{b} \Longleftrightarrow \begin{cases} L\vec{y} = \vec{b} \\ U\vec{x} = \vec{y} \end{cases}$$

$$L : \begin{vmatrix} 1 & 0 & \cdots & 0 \\ * & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & 1 \end{vmatrix}$$

$$U : \begin{vmatrix} * & * & \cdots & * \\ 0 & * & * & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * \end{vmatrix}$$

$$L\vec{y} = \vec{b}$$

$$\begin{cases} l_{11}y_1 = b_1 \\ l_{21}y_1 + l_{22}y_2 = b_2 \\ \vdots \\ l_{n1}y_1 + l_{n2}y_2 + \cdots + l_{nn}y_n = b_n \end{cases}$$

これは代入だけで解ける (上から)

$$U\vec{x} = \vec{y}$$

これも代入だけで解ける (下から)
問題は LU に分解すること

ガウスの消去法による LU 分解の計算

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} & & & k & & & \end{array} \\
k & \left|\begin{array}{ccccccc}
* & * & \cdots & \cdots & \cdots & \cdots & * \\
0 & * & * & \cdots & \cdots & \cdots & * \\
0 & 0 & * & * & \cdots & \cdots & * \\
 & \ddots & 0 & a_{k,k} & \cdots & \cdots & \\
\vdots & & 0 & a_{k+1,k} & \cdots & \cdots & \\
 & & \vdots & \vdots & \vdots & \vdots & \\
0 & \cdots & 0 & a_{n,k} & \cdots & \cdots &
\end{array}\right|
\end{array}
\Longrightarrow
\begin{array}{cc}
 & \begin{array}{ccccccc} & & & k & & & \end{array} \\
k & \left|\begin{array}{ccccccc}
* & * & \cdots & \cdots & \cdots & \cdots & * \\
0 & * & * & \cdots & \cdots & \cdots & * \\
0 & 0 & * & * & \cdots & \cdots & * \\
 & \ddots & 0 & a_{k,k} & \cdots & \cdots & \\
\vdots & & 0 & 0 & a_{k+1,k+1} & \cdots & \\
 & & \vdots & \vdots & \vdots & \vdots & \\
0 & \cdots & 0 & 0 & a_{n,k+1} & \cdots &
\end{array}\right|
\end{array}
$$

第 k 回目の消去前の行列　$A^{(k-1)} \Longrightarrow$ 第 k 回目の消去後の行列　$A^{(k)}$

$$
L^{(k)}
$$

$$
\boxed{\begin{array}{cccccccc}
1 & & & & & & & \\
 & 1 & & & & & & \\
 & & 1 & & & & & \\
 & & & 1 & & & & \\
 & & & l_{k+1,k} & \ddots & & & \\
 & & & l_{k+2,k} & & \ddots & & \\
 & & & \vdots & & & \ddots & \\
 & & & l_{n,k} & & & & 1
\end{array}}
$$

$$
l_{i,k} = -\frac{a_{ik}}{a_{kk}}
$$

$$
L^{(k)}A^{(k+1)} = A^{(k)}
$$

$$
\underbrace{L^{(n-1)} \cdots L^{(2)} L^{(1)}}_{L'} A^{(0)} = A^{(n)} = U(\text{上三角})
$$

補題

1. 下三角行列どうしの積は下三角行列
2. 下三角行列の逆行列は下三角行列

$$
\begin{aligned}
L'A &= U \\
A &= L'^{-1}U \quad :LU\text{ 分解}
\end{aligned}
$$

$$L'^{-1} \equiv L(\text{下三角})$$

　LU 分解の一意性
$A = LU = \bar{L}\bar{U}$ とすると

$$\bar{L}^{-1}L = \bar{U}U^{-1}$$

補題の拡張版

1. 対角要素が 1 の下三角行列どうしの積は対角要素が 1 の下三角行列

2. 対角要素が 1 の下三角行列の逆行列は対角要素が 1 の下三角行列

これより $\bar{L}^{-1}L$ は下三角、$\bar{U}U^{-1}$ は上三角
これを満たすものは対角行列かつ対角要素が 1
つまり単位行列
よって $\bar{L}^{-1}L = I$

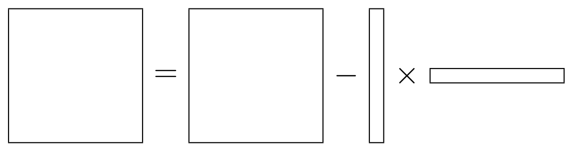
$$L = \bar{L}$$

同様に $U = \bar{U}$
　LU 分解に使われる BLAS 第 k 行の $\frac{a_{ik}}{a_{kk}}$ 倍を第 i 行から差し引く

$$\vec{a}_i^t = \vec{a}_i^t - \left(\frac{a_{ik}}{a_{kk}}\right)\vec{a}_k^t$$

これは AXPY(BLAS1) であり、あまり性能がでない。
しかし、まとめて書くと

$$\begin{pmatrix} \vec{a}_{k+1}^t \\ \vec{a}_{k+2}^t \\ \vdots \\ \vec{a}_n^t \end{pmatrix} = \begin{pmatrix} \vec{a}_{k+1}^t \\ \vec{a}_{k+2}^t \\ \vdots \\ \vec{a}_n^t \end{pmatrix} - \begin{pmatrix} a_{k+1k}/a_{kk} \\ a_{k+2k}/a_{kk} \\ \vdots \\ a_{nk}/a_{kk} \end{pmatrix} \vec{a}_k^t$$



行列の rank1 更新 (BLAS2)
並列粒度の点から言えばいいが、キャッシュの効率はよくない。
そのためあまり性能が出ない。