

# チューニング技法入門(2012年6月1日版) 正誤表

2012年6月1日

「チューニング技法入門」(2012年6月1日版)に掲載されておらず、今後の改訂版に掲載する予定の項目を本資料に示します。

P 18 下記の例で、MODULE COM内で宣言されている変数A,B,Cのうち、サブルーチンSUBではAとBのみを使用できるようにしたい場合、下記の下線部を指定します。

MODULE COM REAL A,B,C END	SUBROUTINE SUB USE COM, <u>ONLY:A,B</u> : END
---------------------------------	--

P 26 で説明した prof や gprof では、サブルーチン(または関数)単位でしかホットスポットが分かりません。ホットスポットのサブルーチンに DO ループが複数含まれている場合、各 DO ループの前後にタイマールーチンを入れて CPU 時間を測定し、どの DO ループがホットスポットなのかを調べて下さい。

P 40 の図 4-4-1 (1) では、1つのループ内で使用する配列の数が多いため、キャッシュミスが発生しています。1つのループ内の配列の数を減らすために、図 4-4-2 (1) ではループを2つに分割しましたが、ループを分割せずに、下記の下線部のようにして減らす方法もあります。

```

:
DIMENSION ABC(3,4),D(4),E(4)
DIMENSION W(4),X(4),Y(4),Z(4)
DO I=1,4
  E(I) = ABC(1,I)+ABC(2,I)+ABC(3,I)+D(I)+E(I)
  Z(I) = W(I)+X(I)+Y(I)+Z(I)
ENDDO
:

```

P 46 の図 4-5-3 (1) と類似の例のチューニング方法を示します。以下の図 1 は、配列 A,B,C が間接アドレス指定のため、図 4-5-3 (1) と同様にストライドが大きくなり、キャッシュミスが多発します。この場合、図 2 に示すように、配列 A,B,C を一つの2次元配列 ABC にまとめると、例えば I=1 のとき、ABC(1,IND(1))、ABC(2,IND(1))、ABC(3,IND(1)) はメモリー上で連続するため、図 1 よりキャッシュミスが減少します。

```

DIMENSION A(N),B(N),C(N),IND(1000)
:
DO I=1,1000
  A(IND(I)) = B(IND(I)) + C(IND(I))
ENDDO
:

```

図 1

```

DIMENSION ABC(3,N),IND(1000)
:
DO I=1,1000
  ABC(1,IND(I)) = ABC(2,IND(I)) + ABC(3,IND(I))
ENDDO
:

```

図 2

P 61 の図 5-2-14 (1) (2) に、DO ループ内の IF 文を除去する例を示しましたが、より一般的な例で説明します。以下の図 3 で、配列 IFLAG の各要素に「k」か「m」(いずれも整数値)のいずれかの値が設定されている場合、図 4 のようにすれば、図 3 の IF 文を除去することができます(図 4 で IFLAG(I) に「k」を代入して整理すると図 3 の①に、IFLAG(I) に「m」を代入して式を整理すると図 3 の②になります)。

テキストの図 5-2-14 (1) の場合は、IFLAG(I) の値は「1」か「0」のいずれかなので、図 5 を図 6 のように変形し、図 6 に対して図 3 → 図 4 と同じ変換を行って式を整理すると、図 7 (テキストの図 5-2-14 (2)) になります。

この変換によって必ず速くなる訳ではなく、却って遅くなる可能性もあります。なお、図 4 で配列 A と処理 1, 2 で使用される変数や配列が例えば実数の場合、k, m, IFLAG も整数でなく実数(同じデータ型)にした方が、型変換が不要になるため若干速くなる可能性があります。

1 2 3 4 5 6 7 8 9 10  
IFLAG 

k	m	m	m	k	m	k	m	m	k
---	---	---	---	---	---	---	---	---	---

 (kとmは整数値)

```

:
DO I=1,10
  IF (IFLAG(I)==k) THEN
    A(I) = 処理1           ①
  ELSEIF (IFLAG(I)==m) THEN
    A(I) = 処理2           ②
  ENDIF
ENDDO
:
    
```

図 3

```

:
DO I=1,10
  A(I) = ( (m-IFLAG(I))*処理1 + (IFLAG(I)-k)*処理2 ) / (m-k)
ENDDO
:
    
```

図 4

```

:
DO I=1,N
  IF (IFLAG(I)==1) A(I) = A(I) + B(I)
ENDDO
:
    
```

図 5 (テキストの図 5-2-14 (1) と同じ)

```

:
DO I=1,N
  IF (IFLAG(I)==1) THEN
    A(I) = A(I) + B(I)
  ELSEIF (IFLAG(I)==0) THEN
    A(I) = A(I)
  ENDIF
ENDDO
:
    
```

図 6

```

:
DO I=1,N
  A(I) = A(I) + IFLAG(I)*B(I)
ENDDO
:
    
```

図 7 (テキストの図 5-2-14 (2) と同じ)

P 106 の 13 行目 チューニング用にジョブの計算時間を短くする方法として、チューニング個所の前後に CPU 時間を測定するタイマールーチンを入れ、チューニング個所を通過したら、CPU 時間と(チューニング部分に関する)計算結果を書き出して、すぐにジョブをストップするという方法もあります。