

「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。

■P2-14 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のみを用で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, ONLY: A, B
:
END
    
```

■P3-6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ムルーチンを入れて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。

■P4-10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しまいます。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
:
    
```

■P4-14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1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

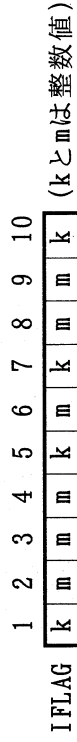
```

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

■P5-13の図5-2-14(1)(2)に、D0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1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2のようにすれば、図1のIF文を除去することができます(図2でIFLAG(I)に「k」を代入して整理すると図1の①に、IFLAG(I)に「m」を代入して式を整理すると図1の②になります)。

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



```

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

```

図1

```

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

```

図2

```

:
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4

```

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

```

図3 (テキストの図5-2-14(1)と同じ)

```

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

```

図5 (テキストの図5-2-14(2)と同じ)

■P8-2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ムルーチンを入れ、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關係する)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。