

5E-4

FORTRAN最適化の強化

—ユーザ定義配列によるベクトル作業領域削減方式—

斎藤裕峰 川口荘太郎¹⁾ (日立東北ソフトウェア)

田中義一 (日立製作所中央研究所)

青山明夫 (日立製作所ソフトウェア工場)

1. はじめに

ベクトル演算の中間結果を主記憶に置く方式のベクトルプロセッサ(例えばHITAC M280H, M680HIAP)においては、ソースプログラムが必要としていた主記憶以上に、大量の中間データが必要となる問題が発生する。このような問題を解決する方法として、ユーザがDOループ内で定義している配列を中間的に破壊してもよいことを、配列データフロー情報を利用して判定することにより、ベクトル作業領域を削減することができる。本論文ではその適用条件と処理方式について述べる。

2. 適用条件

図1に、あるFORTRANプログラムのベクトル化された表現を示す。図において、 $\#V(I)=\dots$ は作業ベクトル配列であり、 $\#V(I)$ 及びユーザ定義配列 $A(I)$ は、ともに添字1-Nまで参照されている。

```

#V(I)=...           ;I=1,...,N ①
...=#V(I)...       ;I=1,...,N
A(I)=...           ;I=1,...,N ②
    
```

図1. ベクトル化後のプログラムの例

作業ベクトル配列 $\#V(I)$ に対して、ユーザ定義配列 $A(I)$ を用いることができるかどうかの適用条件を次に示す。

ここで作業ベクトル配列は、中間ベクトル演算結果に割当てられるので、その定義配列は1つであり、使用箇所も1つであり、かつ隣接文に存在することを前提としている。

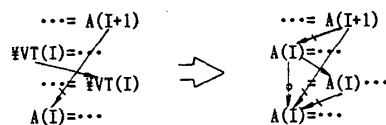
適用条件. ユーザ定義配列 $A(I)$ を終点とする、

- 逆依存*1の始点の配列参照の文が①
- と②(②の右辺は含む)の間に存在しないこと。

この条件が満足されればユーザ定義配列の結果の同一性が保証される。

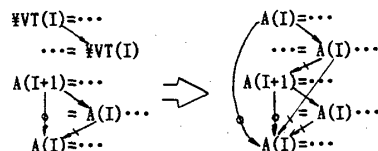
条件1について以下例を挙げる。

図2はユーザ定義配列が適用できる例であり、図3は、 $A(I)$ が使用される前に値が壊されてしまうため、ユーザ定義配列が適用できない例である。



逆依存 / 出力依存 / フロー依存

図2. ユーザ定義配列を適用できる例



逆依存 / 出力依存 / フロー依存

図3. ユーザ定義配列を適用できない例

複数の使用可能なユーザ定義配列がある場合には、次の条件によりベクトル性能上最適なものを選ぶ。

選択条件. 複数のユーザ定義配列のうち、配列参照の挙動を調べ、そのDOループで連続してアクセスされている方を選ぶ。

この条件はメモリアクセスに関してベクトル機能上最適なものを選ぶための条件である。

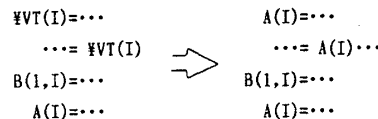


図4. ユーザ定義配列が適用できるもののうち、連続であるA(I)の方を使用した例。

*1 : 使用している配列が、定義配列で置き換えられることを示すもので配列の依存として他にフロー依存、出力依存がある。

Size reduction of vector work area by using user declared array's area.

- 1) Hiroyuki Saitou 1) Soutarou Kawaguchi
- 2) Yoshikazu Tanaka 3) Akio Aoyama

- 1) Hitachi Tohoku Software Co., Ltd.
- 2) Central Research Laboratory, Hitachi, Ltd.
- 3) Software Works, Hitachi, Ltd.

3. 処理方式

(1) 作業配列候補テーブルの作成

図5に示したようなベクトル化されたFORTRANプログラムに対して、図6に示すような作業候補テーブルを作成する。このとき、ユーザ定義配列が逆依存を持っている場合は、文番号及び定義配列の他に、依存の始点となる配列を抑止配列参照チェーン上に登録しておく。

```

#VT1(I)=B(I)+C(I) ;I=1,...,N 51
A(I)=#VT1(I)+D(I) ;I=1,...,N 52
#VT2(I)=C(I+1)+A(I) ;I=1,...,N 53
C(I-1)=#VT2(I)+B(I) ;I=1,...,N 54

```

図5. ベクトル化後のプログラム

文番号	ユーザ定義配列	抑止配列参照チェーン
51	NIL	NIL
52	A(I)	NIL
53	NIL	NIL
54	C(I-1)	

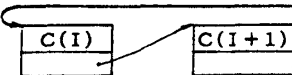


図6. ベクトル作業配列候補テーブル

(2) 置換すべきユーザ定義配列の選択

代入文の代入先がベクトル作業配列のときには作業配列候補テーブル中から、当該文番号以上のエントリのユーザ定義配列で、かつ抑止配列参照チェーンがNILのものを見つけ、一つでも存在すれば、その中から配列参照アドレスが連続に近いユーザ定義配列をとってきて置き換える。

(3) 候補テーブルの更新

当該文で使用しているユーザ配列の参照終了に伴って、抑止配列参照チェーン上の当該エントリを削除する。

4. 効果

(1) ユーザ定義配列が使用できることによりベクトル作業領域が削減できる。

(2) (1)の一般的効果の他に、ユーザ定義配列を用いることにより、IAPで用意されている積和命令の適用範囲が広がった。すなわち、HITAC M280H, M680H IAPの積和命令では、ハードウェアの制約上、第1, 2オペランドが同一でなければならず、中間結果を作業ベクトル配列に代入すると、積和命令が使えない。図7において左辺のユーザ定義配列が使用できると、積和命令が適用できる。実際によると、本論文の方式を適用しない場合、47MFLOPSであるのに対して、ユーザ定義配列を利用した場合には84MFLOPSに高速化された。

```

DO 10 I=1,256
A(I,1)=S1*A(I,2)+S2*A(I,3)
+S3*A(I,4)+S4*A(I,5)
+S5*A(I,6)

```

10 CONTINUE

図7. 積和命令を含むFORTRANプログラム

5. おわりに

中間ベクトル演算結果を主記憶に置く方式のベクトルプロセッサにおいて、必要なベクトル作業領域を削減する方式として、ベクトル作業配列の代わりに、ユーザ定義配列を利用する方式を開発した。本方式の副次的効果として、積和命令の逐次演算においてオブジェクトプログラムの実行性能が改良されることがわかった。

参考文献

- [1] KUCK 他: DEPENDENCE GRAPHS AND COMPILER OPTIMIZATION
- [2] 金田 他: FORTRAN最適化の強化—大域配列データフロー解析法
情報処理学会第22会全国大会 4F-3