

ベクトルプロセッサPUのFortran コンパイラの開発

5N-1

井上 淳 真鍋 俊彦 前田 明
(株) 東芝 総合研究所

1. はじめに

画像処理を始め、様々な分野で高速処理に対する要求が高まっている。

我々は大プロ成果の高度化研究としてより広い応用に適用するため、通産省工業技術院の大型プロジェクト「科学技術用高速計算システムの研究開発」[1][2][3][4]の一環として開発したベクトルプロセッサPU (Processor Unit) をターゲットマシンとした自動ベクトル化、最適化機能を持つPU-Fortranコンパイラ的设计、開発を進めている。PUはメモリーメモリー演算を主体とするベクトル計算機で、ベクトル添字値の集合「インデクスセット」を独立に処理する機能を有しており、この機能を用いることによりベクトルの間接参照や同一ベクトル内での条件処理など柔軟なベクトル処理を実行することができる。

本稿では、このPU-Fortranコンパイラの構成と、インデクスセット機構を有効活用するためのベクトル化手法について説明する。

2. PU-Fortranコンパイラの構成

PU-fortranコンパイラはホスト計算機AS-3160 上で動作するクロスコンパイラで、

- (1) 構文解析、意味解析部
- (2) ベクトル化部
- (3) パイプライン最適化部
- (4) コード生成部

の4フェーズから構成される。

ベクトル化は二段階に分けて行う。ベクトル化部ではプログラム内のDOループを対象としたベクトル化処理を行い、マシンに依存しないベクトル化中間形式への変換を行う。次のパイプライン最適化部では、その中間形式からインデクスセットを用いたベクトル演算を生成する。また同時にベクトル化形式を演算単位に分割し、各演算でのデータ参照関係を調べた上で、演算の並べ替えを行う。これにより演算パイプラインの稼働率向上を狙っている。

3. PU-Fortranコンパイラのベクトル化方式

本コンパイラで実現した特徴的なベクトル化方式について説明していく。

(1) ベクトル分類処理

PUではベクトルインデクスの分類操作により、DOル

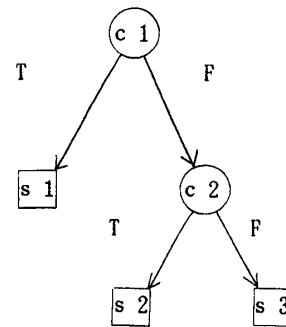
プ内の条件文を効率よく処理することができる。この機能を活用するため、本コンパイラではベクトル化のための前処理としてフロー解析を行っている。例えば図1(a)に示すソースプログラムに対して、処理対象となるDOループ内を基本ブロック単位に分割し、これらを結合した制御フローグラフを生成する(図1(b))。このグラフ上を辿り、条件分岐に到達した時点でインデクス分類操作を実行する。分類結果のインデクスセットは分岐後の基本ブロックの実行条件に対応しているので、このインデクスセットを使って各文をベクトル化すればよい(図1(c))。

```

DO 10 I=1,N
c 1 : IF(A(I).GT.10.0)THEN
s1: B(I)-C(I)+D(I)
c 2 : ELSE IF(A(I).GT.5.0)THEN
s2: B(I)-C(I)-D(I)
ELSE
s3: B(I)-C(I)*D(I)
END IF
10 CONTINUE

```

(a) ソースプログラム



(b) フローグラフ

```

IX=(1:N)
IX1,IX2=A(IX).GT.10.0
B(IX1)-C(IX1)+D(IX1)
IX3,IX4=A(IX2).GT.5.0
B(IX2)-C(IX2)-D(IX2)
B(IX3)-C(IX3)*D(IX3)

```

(c) ベクトル化結果

図1 ベクトル分類処理例

(2) 間接参照のベクトル化

図2(a)のようなベクトル間接参照を含むDOループの場合、PUではポインタ配列値をとるインデックスを新たに定義することでベクトル処理が可能である。

この場合、ベクトル化部は間接参照部分を値が未知の添字としてベクトル化判定を行う。図2ではデータ依存がないのでベクトル化可能と判定され、図2(b)のようなベクトル化中間形式に変換される。この形式はベクトル式 $ITBL(IX)+1$ を一つのインデックスセットとして扱うことを示しており、パイプライン最適化部で図2(c)のように必要なインデックス(IXTMP)生成演算とベクトル演算とに分割して処理が行われる。

(3) ループ内スカラー変数処理

本コンパイラではインデックスセット機構を有効に活用するため、DOループ内に出現するスカラー変数に対する解析

```

DO 10 I=1,N
  JP(1)=IP(ITBL(1)+1)
10 CONTINUE      (a) ソースプログラム
↓
IX=(1:N)
JP(IX)=IP(ITBL(IX)+1) (b) ベクトル化形式
↓
IX=(1:N)
IXTMP=ITBL(IX)+1
JP(IX)=ITBL(IXTMP)  (c) 演算形式

```

図2 間接参照のベクトル化例

機構を強化した。

具体的には、DOループ内に出現するスカラー変数の出現位置、定義形式を元に次の4クラスに分類する。

(i) ループ内不変(loop invariant)

ループ内で値を代入されない変数。

(ii) ループ帰納変数(induction variable)

ループの進行と共に等差数列的に値を変える変数。配列添字に現れた場合は対応するインデックスセットに変換される。式の右辺に現れた場合は等差数列生成関数に変換される。

(iii) テンポラル格納(temporal storage)

ベクトル式を代入される変数。スカラー展開[5]によりベクトルテンポラルとして処理する。

(iv) 再帰変数(reduction variable)

ベクトルの要素方向アクセス結果を代入する変数。ベクトル総和、内積などがこれに相当する。適当なベクトル関

数コールに変換される。

これらのスカラー解析結果を元にベクトル化を行った例を図3に示す。このループは、帰納変数(J,K)、テンポラル格納(S,T)を含んでいるが、各々適当なインデックスセット、等差数列関数(VSEQ)、ベクトルテンポラルに変換され図3(b)のようにベクトル化される。

```

K=N
DO 10 I=1,N
  S=A(1)+B(K)
  T=S*C(1)
  J=2*I
  D(J)=T+J
  K=K-1
10 CONTINUE      (a) ソースプログラム
↓
IX=(1:N)
IXK=(N:1;-1) /* INDEX FOR K*/
IXJ=(2:N*2) /* INDEX FOR J*/
STMP(IX)=A(IX)+B(IX)
TTMP(IX)=STMP(IX)*C(IX)
D(IXJ)=TTMP(IX)+VSEQ(2:2*N;1)

```

(b) ベクトル化結果

図3 スカラー処理によるベクトル化例

4. おわりに

ベクトル計算機PUの自動最適化Fortranコンパイラの構成、及び特徴的なベクトル化手法について説明した。現在、上記仕様を満たすコンパイラを開発している。

参考文献

- [1] Inoue A. and Maeda A.: "The Architecture of a Multi-vector Processor System, VPP", proc. of VAPP3, Parallel Computing, Vol.8, No.1-3, 1988
- [2] 真鍋他: 並列処理システムVPPのアーキテクチャ、情処学会研究報告88-ARC-71、1988
- [3] 関戸他: 並列処理記述言語の概要、第30回情処全4B-1、1985
- [4] 橋本他: VPPシステムによる画像処理、第34回情処全6Q-2、1987
- [5] Kuck D.J. et al.: "Dependence Graphs and Compiler Optimization", Proc. of ACM 8th annual Symp. on Principles of Prog. lang. pp.207-218, 1981