条件分岐向けソフトウェアパイプラインスケジューラの実装

5L－2

糸賀 裕弥 (筑波大学) †　山下 義行 (筑波大学) ††　中田 育男 (図書館情報大学) †††

1. はじめに

条件分岐を含むループのソフトウェアパイプライン法として Enhanced Modulo Scheduling (EMS) がある[1]. EMS では，条件分岐の結果に応じて異なる複数のカーネルをあらかじめ用意しておき，繰返しごとに適切なカーネルが選択され実行される。

著者らは EMS よりさらに高い実行効率を得るために，条件分岐の結果に応じて実行されうる，最も演算量の少ないコードを基準とした命令スケジュール法を提案した．これを改良 EMS と呼ぶ[2].

EMS および改良 EMS では，述語付き命令コードをモジュロスケジュールし，逆 IF 変換[3] によって複数のカーネルとして展開する．逆 IF 変換には，パイプラインステージ数に応じて，実行コードの大きさが指数関数的に大きくなるという性質がある．このため，高い実行効率を保持したまま，ステージ数をできる限り小さくするスケジュール法の改良が必要である．

本稿では，改良 EMS にもとづくスケジュール法について述べ，実行効率の保持とコードサイズの縮小を目的とした改良法を提案する．

2. 改良 EMS のスケジュール

スケジュールの流れはおよそ次のようになる．

(1) IF 変換により述語付き命令を作る．
(2) データおよび制御の依存グラフを作る．
(3) Initiation Interval(II) を算出する．
(4) モジュロスケジューリングによりリソース予約テーブルに命令を配置する．
(5) 逆 IF 変換によりコードを生成する．

図 1 の例題プログラムに対して，ノードを命令語として依存グラフを描くと図 2 のようになる．枝に付随した数字は命令レイテンシをあらわす．

EMS では条件分岐の結果実行されうるコードのうち，性能を決定する II が最も大きい場合を基準としてスケジュールを行なっていた．しかし改良 EMS では，実行効率を向上させるために，II の最も小さい場合を基準とする．例題の場合，改良 EMS では図 2 の右側，すなわち条件分岐の結果が偽の場合である．

改良 EMS 用のリソース予約テーブルには，同じ実行タイミングに条件分岐の結果の真偽によって異なる複数の命令を格納する．また，必要に応じてテーブルを拡張し II の大きい側のコードを格納する．拡張した部分が実行される場合にのみ II が伸長されるため，最適な実行が可能である．

このコードを通常のアーキテクチャで実行するために逆 IF 変換を施し，ステージごとの条件分岐の履歴から異なるカーネルに展開する．真偽はステージごとに独立であるから，ステージ数の 2 のべき乗種類のカーネルとなる．

```
DO I=1, 1000
  IF ( A(I) > 1.0 ) THEN
    C(I)=((A(I)+1)*2+3)*4
  ELSE
    C(I)=(A(I)+1)**2*5
  END IF
END DO
```

図 1　例題プログラム
Fig. 1　A sample program.

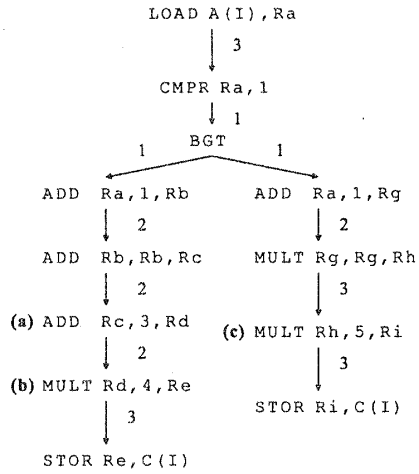

図 2　依存グラフ
Fig. 2　A simple dependency graph.

Implementation of Improved Enhanced Modulo Scheduler for Loops with Conditional Branches

† Hiroya ITOGA, Doctoral Program in Engineering, Univ. of Tsukuba, itoga@lang.esys.tsukuba.ac.jp
†† Yoshiyuki YAMASHITA, Inst. of Eng. Mech. and Sys., Univ. of Tsukuba
††† Ikuo NAKATA, Univ. of Library and Info. Sci.

3. 基本的なスケジュール法

図2のプログラムをスケジュールする場合，一般的には上方から優先してスケジュールする．

条件分岐命令より下方の命令は，左側あるいは右側のいずれかのみが実行される．改良EMSでは，条件分岐の真偽で異なる命令を命令ペアとして，同じ実行タイミングにスケジュールリングする．改良EMSを実現する条件[2]から，演算量の少ない側の命令は必ず命令ペアとなる必要があるが，演算量の多い側の命令は必ずしもペアとなっていなくとも良い．

そこで，命令のレイテンシを考慮しつつ，依存グラフの上方から順に命令ペアを選択し予約テーブルに格納する．命令の依存関係を満たすように留意することで，必ずスケジュールが可能である．

4. スケジュール法の改良

基本的なスケジュール法により改良EMSが実現される．しかし，逆IF変換によって展開されたコードは大きなサイズになっている場合がある．

逆IF変換によって展開されるコードは，条件分岐命令よりも下方の部分であり，この部分の占めるステージ数によって，コードサイズが大きく変化する．スケジュール法の改良によりステージ数を減らすことで，命令キャッシュなどを最大限利用することができる．

4.1 命令ペアの選定方法の改良

図2に対して基本的なスケジュール法を用いると，上方より命令ペアをつくるため，(a)の加算命令と(c)の乗算命令が命令ペアとなる．この場合には条件分岐以降の占めるステージ数が3となり，逆IF変換によって生成されたカーネルは図3左のように8種類となる．

しかしここで，(b)の乗算命令を(c)の乗算命令と命令ペアとすると，条件分岐以降の占めるステージ数は2となる．カーネルは半分の4種類に展開されるのみである．展開されたコードは図3右のように小さくなっている．

(c)の乗算命令は命令レイテンシよりも長く実行を待つことになるが，ソフトウェアパイプラインの性能はIIによって決定されるので，パイプライン処理による実行性能に影響は生じない．(a)の加算命令は，適切な位置のリソース予約テーブルを拡張し格納する．これにより実行効率を保持したまま，コードサイズを小さくすることが可能である．

命令ペアとなる組合わせを求める方法はいくつかあり，全ての組合せでスケジュールを試す方法，命令の実行タイミングを予測しておく方法などがある．

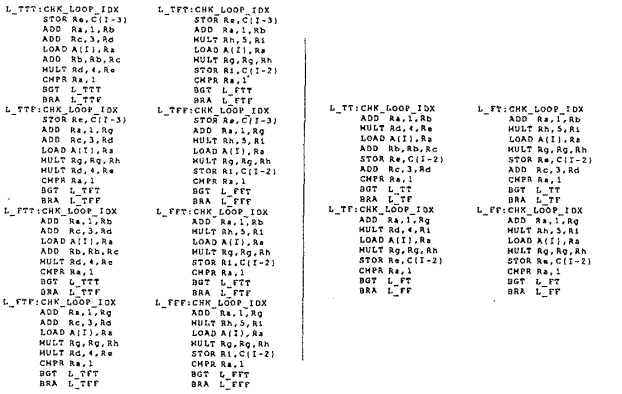

図3 スケジュール結果
Fig. 3 Result of scheduling.

4.2 条件分岐命令を基準とするスケジュール

図2において条件分岐命令より上方の命令は，逆IF変換によるコードの増大に影響を与えない．

スケジュールは依存グラフの上方を優先するのが一般的であるが，改良EMSでは，条件分岐命令より下方を優先し，その後上方にスケジュールしていくことで，コードサイズをより小さくすることが可能である．

このときには実行パスを下方からスケジュールするため，必要レジスタ数が増えるなどの問題が起こる可能性がある．

5. まとめと今後の課題

改良EMSで逆IF変換によってカーネルを複数に展開する．そこで，スケジュール時の命令ペアの選択を適切に行なうこと，および条件分岐命令以後を優先してスケジュールすることで，改良EMSの利点である高い実行効率を保持したまま逆IF変換による展開数を減らす方法を提案した．

今後はこの提案をもとに，改良されたスケジューラを実装し，改良EMS自体の性能の評価を行ない，実際のコンパイラへの組み込みを目指す．

参考文献

1) Warter, N.J., Haab, G.E. and Bockhaus, J.W.: Enhanced Modulo Scheduling for Loops with Conditional Branches, *Proc. of the MICRO-25*, pp. 170–192 (1992).
2) 山下義行, 中田育男: ループ中に条件分岐を含む場合の最適なソフトウェア・パイプライニング, 並列処理シンポジウム JSPP'94, pp. 17–24 (1994).
3) Warter, N. J., Mahlke, S. A., mei W. Hwu, W. and Rau, B. R.: Reverse If-Conversion, *Proc. of the ACM SIGPLAN 1993 Conf. on Prog. Lang. Des. and Impl. (PLDI)*, pp. 290–299 (1993).