

2D-8 先行評価を用いたFORTRANプログラムの並列化と実行方式

山名早人, 安江俊明, 村岡洋一

早稲田大学理工学部

1. はじめに

本報告では, FORTRANプログラムをマルチプロセッサ上で高速に実行するための方式として, 先行評価を用いたプログラムの並列化手法と実行方式を提案する. 従来, 条件分岐を含むプログラムを並列化する手法として, タスクの最速実行条件を求める手法[1][2]や制御依存を越えた実行方式[3][4][5]が提案されている. しかし, (1)最速実行条件を求めるだけでは十分な並列性が得られない, (2)対象プログラムが限定され, かつ, 実行方式の提案がないといった問題を持つ. これらの問題に対して我々は, フローグラフ展開を用いた仮実行方式[6], データ駆動を用いた条件分岐のn段先行評価制御方式[7]を提案している. 本稿では, これらの手法を一般化すると共に, 理論的な速度向上について論じる.

2. 対象マルチプロセッサシステム

各要素プロセッサ(PE)内において(1)細粒度の並列処理を行うことができ, (2)PE内にデータスプール機構あるいはローカルメモリを持つとする. また, 全体を制御する機構として, (3)放送機構を備える, (4)タスクの割当を動的に行えることとする. (1)の例としては, PE内部に複数の演算処理装置を持つ並列処理システム—晴—[8]や, スーパスカラプロセッサをPEに持つマルチプロセッサシステムがある. PE内部で細粒度の並列処理を行うことによって, 以下で述べるマクロタスク内の並列処理を行う. (2)のローカルメモリは, 多くのマルチプロセッサが持つ機構であり, 先行評価時に, 共有メモリに対する副作用を避けるためのバッファとして用いる. (3)の放送機構は, 制御情報の伝達に用いる. (4)は, 多段仮評価時に, 新たなタスクをプロセッサへ割り当てるために用いる.

1. マクロタスクの定義

先行評価を行う際のタスク単位をマクロタスク(以下MT)と呼ぶ. また, 1MTは1PEに割り当てられ, PE内部で並列処理されるものとする. MTの最小単位としては, 基本ブロックが考えられるが, 基本ブロックをMTとした場合, (1)MT数が膨大になり, タスク制御オーバーヘッドが大きい, また, (2)多段の先行評価時に必要なプロセッサ数も増加する. これらの問題を解決するため, 基本ブロック間の制御依存とデータ依存に基づいて融合し, MTを構成する.

手順(1) 基本ブロックの融合 先行評価に伴う副作用を回避する為, 条件分岐によって異なったデータが流れる(参照される)基本ブロックを複製する. すなわち, 基本ブロック間の広域コード移動の規則R1[9]を用いて, 条件分岐を含む基本ブロックを複製する(図1(b)). 次に, 2つの基本ブロックBa, Bbが同一の制御依存を持ち, かつ, Ba→Bbのデータ依存を持つ場合, これらを融合する(図1(c)). これを基本スレッドと呼ぶ. ここで, 融合される基本ブロック間には, データ依存があるため, 融合しても並列度は増加しない.

手順(2) 基本スレッドの融合 実行に際して必要となるプロセッサ数を削減するために基本スレッドの融合を行う. 融合は, 基本スレッド間のデータ依存に着目し, ある基本スレッドが制

御依存上の直後の基本スレッドに対してデータ依存を持ち, かつ, 同一基本スレッド内の後続の基本ブロックが他の基本スレッドからデータ依存を受けない場合, これらを融合する. 図1(c)の基本スレッドの例では, 手順(2)によって7つの基本スレッドが4つに融合される. これをマクロタスク(MT)と呼ぶ.

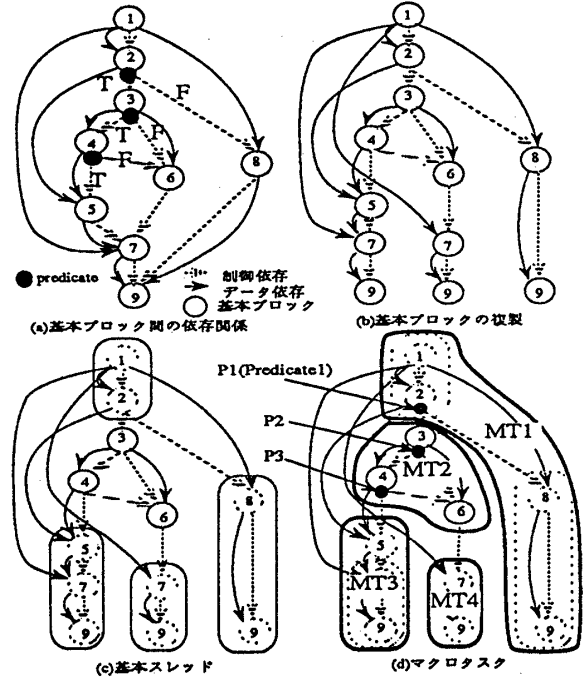


図1 基本ブロック・基本スレッド・マクロタスク

4. 非ループ部分におけるマクロタスク間先行評価

非ループ部分では, MTの起動条件を各々のMTが判断し, 実行を開始する. 先行評価を行う為, 実行開始条件は, データ依存の保証のみである. また, 先行評価開始後のMTの選択・非選択を判断するために, 各MTは, 自MTの制御が確定した時点で, 制御の選択情報を他のMTに対して放送し, その制御の選択情報に基づいて, 自分のMTの実行を続行するか停止するかを決定する. ここで, 先行評価開始時の状態を仮実行, 制御が確定した状態での実行を本実行と呼ぶ. 図1(d)に示すMT1...MT4の例における, 実行開始条件, 仮実行から本実行への移行及び本実行停止条件を表1に示す.

表1: 図1(d)のマクロタスク(MT)の制御条件

MT	実行開始条件	本実行移行条件	実行停止条件
1	T	-	-
2	T	P1(T)	P1(F)
3	TE(1-2)	P3(T)	P2(F) or P3(F)
4	TE(1-2)	P2(F) or P3(F)	P3(T)

T:true, TE(n-m):End of BasicBlock m in MTn
Pn(T/F):predicate n is True/False

5. ループ部分におけるマクロタスク間先行評価

まず, MT生成について述べる. 最初に, 3節の手順(1)によって, 1イテレーション内のコードについて基本スレッドを生成する(図2(b)). この際, イテレーションにまたがる基本ブロックの融合は行わない. 次に, 手順(2)により, 基本スレッドを融合する. 本例では, 1つのMTとなる. 最後に, 合成された1つのMT内で閉じて

いない制御の入口(図2(b)のA, B, C)を先頭とするようにMTを再構成する。(図2(c))

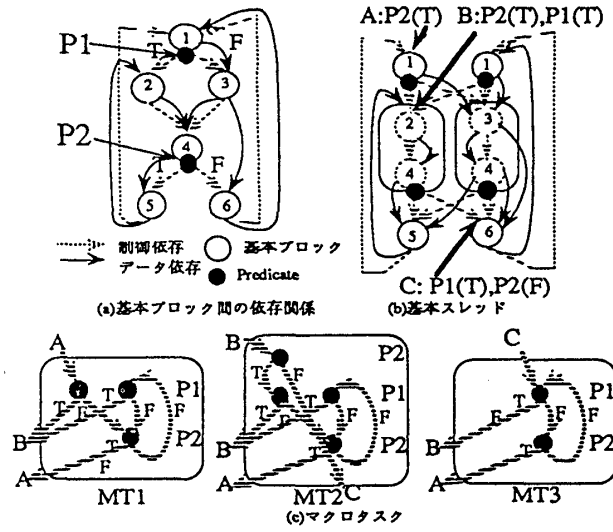


図2 ループ部分におけるマクロタスク構成

図3にループ部分におけるMTの先行評価例を示す。図2(c)に示すように、イテレーション間で依存をもたないパスを選択する場合には、新たなMTの生成を行う。制御の確定したMTをルートマクロタスクと呼ぶことにすると各MTは、次に示す処理を行う。

(a)ルートMTから放送される条件分岐の結果(経路選択情報,0/1で示す)と各MTが内部に持つ経路情報を比較する。各MTが持つ経路情報にはルートMTから自MTまでの経路が記述(True側を0, False側を1とし、1の後にその経路を並べたもの。例えばT/F/Tであれば1010)されており、経路選択情報と経路情報の左から2bit目が一致した場合には、持っていた経路情報を更新し、一致しない場合にはMTの実行を停止する。このようにして、経路情報が1になったMTが新たなルートMTとなり、経路選択情報を放送する。

(b)各マクロタスクは、自分の後続MTの経路情報を計算し、後続が自MTでない場合には、新たなMTを生成する。

以上の2つの処理を各MTが実行することによって、多段先行評価を行うことができる。先行評価の段数は、各MT持つ経路情報のビット数を用いることによって制御可能である。また、ルートMT以外の実行中MTは、仮実行というモードで実行することによって、共有メモリへのライトデータを自動的にスプールさせ、先行評価による副作用を防ぐ。

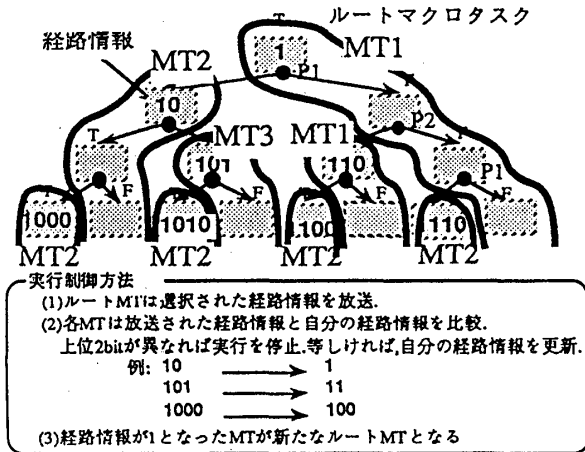


図3 ループ部分の先行評価の実行例

6. 多重ループの先行評価

多重ループになっている場合、図4に示すように内側のループを1つのマクロタスクと考えることによって、多重ループに渡る先行評価を行うことができる。

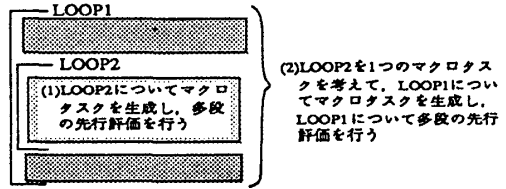


図4 多重ループの先行評価

7. 先行評価による処理速度向上

先行評価による速度向上率 R_{up} は、マクロタスクを MT_0, \dots, MT_{n-1} , 先行評価するMTの段数が n , すなわち、ある MT_i の後 MT_j , MT_k のどちらか一方が選択されるような場合先行評価を行うとすると $n=2$ となる。さらに、各同一段のMTの最大実行時間を T_0, \dots, T_{n-1} , 第 i 段の実行開始時刻から第 $i+1$ 段が実行開始できるまでのディレイ時間を D_i で表すとすると、

$$R_{up} = \frac{\sum_{i=0}^{n-1} T_i}{\text{Max}_{j=0..n-1} (\sum_{i=0}^{j-1} D_i + T_j)}$$

となる。 R_{up} は、 $D_i(i=0..n-1)=0, T_1=T_2=\dots=T_{n-1}$ の時最大となり $R_{up} = n$ となる。

8. むすび

本報告では、先行評価を用いたプログラムの並列化手法と実行方式を提案した。今後は、プログラムをパラメータ化し、本方式の効果を予測する手法の検討を行う。

謝辞 熱心に御討論いただいた本研究室、石崎一明氏、石井吉彦氏に感謝いたします。

参考文献

- [1] C.D.Polychronopoloulos: "α-Coral: A Control/Data Flow Multiprocessor and its Compiler", Proc. of JSSP'91, pp.1-12, 1991
- [2] 本多他: "OSCAR上でのFortran並列処理系のインプリメントと性能評価", 信学技報, CPSY-89-57, pp.75-80, 1989
- [3] E.Riseman et.al: "The Inhibition of Potential Parallelism by Conditional Jumps", IEEE Trans. on Comp., pp.1405-1411, 1972
- [4] U.Banerjee et.al: "Fast Execution of Loop with IF statements", IEEE Trans. on Comp., pp.1030-1033, 1984
- [5] A.K.Uht: "Requirements for Optimal execution of Loops with Tests", Proc. of ICS'88, pp.230-237, 1988
- [6] 山名他: "並列処理システム一瞥一におけるフローグラフ展開を用いた条件分岐の並列実行", 早大情科センター紀要, Vol.12, pp.8-18, 1991
- [7] 山名他: "並列処理システム一瞥一における条件分岐の先行評価制御方式", 情処研報, 計算機アーキテクチャ, Vol.89-19, pp.135-142, 1991
- [8] H.Yamana et.al: "System Architecture of the Parallel Processing System -Harray-", Proc. of ICS'88, pp.76-89, 1988
- [9] J.A.Fisher: "Trace Scheduling: A Technique for Global Microcode Compaction", IEEE Trans. on Comp., pp.478-490, 1981