

階層型マクロデータフロー処理における 2B-4 サブルーチン並列処理手法

宮沢稔† 岡本雅巳† 笠原博徳†

†早稲田大学理工学部

1 はじめに

マルチプロセッサシステムにおける従来の Fortran 自動並列化コンパイラでは Do-all や Do-across などのループ並列化 [1] のみが用いられていた。この場合、ループ以外の部分の並列性、たとえば基本ブロック内部の並列性や、基本ブロック、ループ、およびサブルーチン間の粗粒度並列性を利用することはできなかった。筆者らは以上のような問題を解決するため、従来よりマルチグレイン並列処理手法 [2] を提案してきた。これは、基本ブロック、ループ、サブルーチンより定義される粗粒度タスク（マクロタスク）の並列処理（マクロデータフロー処理）[2, 3, 4]、中粒度並列処理（ループ並列化）、細粒度並列処理を階層的に適用した並列処理手法である。

またマクロデータフロー処理において、ループ内あるいはサブルーチン内に存在する基本ブロック、ループ、およびサブルーチン間の粗粒度並列性を階層的に利用する階層型マクロデータフロー処理も現在インプリメント中である。この階層型マクロデータフロー処理では、サブルーチン並列処理手法としてインライン展開を用いる方法とサブルーチンをマクロタスクとして定義する方法の両方が用いられている。後者の方法においてサブルーチンと他のブロック間の並列性やサブルーチンコールを含む DO ループのイタレーション間の並列性を利用するにはインタープロシージャ解析が必須となる。本稿では、階層型マクロデータフロー処理におけるインライン展開及びインタープロシージャ解析を用いたサブルーチン並列処理手法について述べる。

2 階層型マクロデータフロー処理手法

本節では、階層型マクロデータフロー処理手法 [5] について述べる。

OSCAR マルチグレインコンパイラ [2, 4] ではプログラムを以下に示す三種類のマクロタスク (MT)[2] に階層的に分割する (図 1)。

- BPA(Block of Pseudo Assignments)
基本ブロックおよび複数の小基本ブロックを融合したブロック
- RB(Repetition Block)
最外側ナチュラルループ
- SB(Subroutine Block)
インライン展開が有効に適用できないサブルーチン

各階層の MT は図 2 に示すように階層的に定義されたプロセッサクラスタ (PC) 間で並列処理される。また、各階層で PC に割り当てられた MT は、さらに PC 内のプロセッサエレメント (PE) により階層的に並列処理される。例えば MT が BPA であるならば、BPA 内部では細粒度並列処理を適用する。RB ならば Do-all や Do-across などの中粒度並列処理、あるいはループボディの細粒度並列処理、また RB が大規模

であり内部でサブ MT が階層的に定義できる場合には階層的にマクロデータフロー処理を適用する。この場合、階層型マクロデータフロー処理ではサブ MT は PC 内で定義されるサブ PC に割り当てられる。

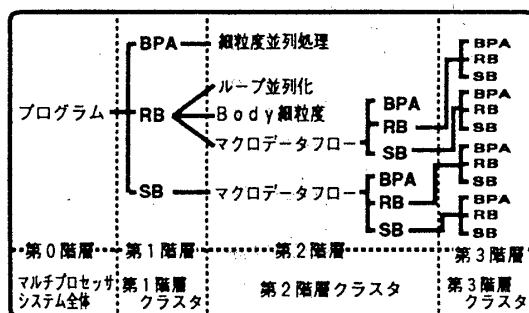


図 1: マクロタスクの階層的な定義

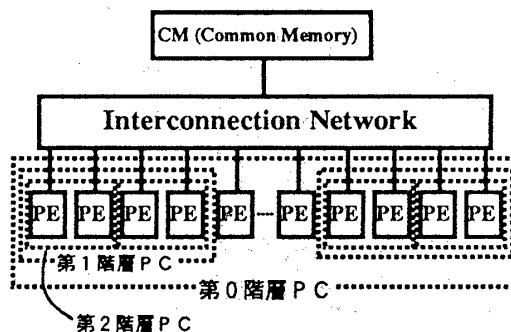


図 2: プロセッサクラスタの階層的な定義

サブルーチンに対しては主にインライン展開を用いるが、コード長を考慮してインライン展開が有効でない場合には SB として定義する。このような SB に対しては内部を階層的にサブ MT に分割し、階層型マクロデータフロー処理を適用する。階層的に MT を生成したら、各階層で MT 間のコントロールフロー解析、およびデータ依存解析を行う。解析したコントロールフローとデータ依存は各階層毎に生成するマクロフローグラフで表現する。次に各階層の (サブ) マクロフローグラフに対して最早実行可能条件解析 [3] を行い、MT 間の並列性を抽出する。この結果得られた各階層における各 MT の最早実行可能条件をグラフで表現したマクロタスクグラフを各階層毎に生成する。

各階層の MT はスタティックスケジューリング [1]、またはダイナミックスケジューリング [4] によって、PC に割り当てられる。このとき、各階層のマクロタスクグラフに実行時不確定性 (条件分岐等) が存在する場合はダイナミックスケジューリングを用い、それ以外の場合には実行時オーバーヘッドの最小化の可能なスタティックスケジューリングを用いる。本手法ではダイナミックスケジューリングにおいても OS コール等

A Subroutine Parallel Processing Scheme for Hierarchical Macro-dataflow Computation

Minoru MIYAZAWA†, Masami OKAMOTO†, Hironori KASAHARA†

†School of Science and Engineering, Waseda Univ.

を用いず、コンパイラが各階層のダイナミックスケジューリングルーチンを生成するので、オーバーヘッドを平均 50 クロック程度と極めて低く抑えられる。

3 階層型マクロデータフロー処理におけるサブルーチン並列処理手法

3.1 インライン展開を用いたサブルーチン並列処理手法

インライン展開を用いる場合には、サブルーチン内部のコードは呼び出し箇所を展開される。この方法では、呼び出しが存在する階層に階層型マクロデータフロー処理を適用することによってサブルーチン内の各最外側 MT は呼び出し側階層の他の MT と同一レベルで粗粒度並列処理が可能になる。また、インライン展開された各ブロックの内部に階層型マクロデータフロー処理を適用することも可能であり、これによりサブルーチン内のどの階層においても自在に粗粒度並列処理を適用することが可能になる。

本来サブルーチン内ローカル変数にはサブルーチン外部とのデータ依存は存在しないが、インライン展開を用いて同一のサブルーチンを複数回展開した場合、それらの間でローカル変数によってデータ依存が生じ、粗粒度並列性が失われる可能性がある。この場合には、展開される度にローカル変数をリネームし、同一サブルーチン間の粗粒度並列性が損なわれないようにする。同様に、ループ内部にインライン展開を適用する場合には、ローカル変数によるイタレーション間データ依存を除去するためにスカラー・アレイエクステンション、あるいはアレイプライベートイタレーションなどの手法を用いる。

3.2 インタープロシージャ解析によるサブルーチン並列処理

マクロデータフロー処理により生成される SB は、サブルーチンコール部分とサブルーチン本体実行部分の 2 つからなる。

SB に関するデータ依存解析を行なうためには、SB における各変数の入出力を正確に把握する必要がある。SB の入力とは、ある MT において定義されるよりも前に参照される変数のことを指す。つまり先行 MT で定義された変数を使用する場合である。変数の出力とは、ある MT 内での同一変数に対する定義の内、最後に行なわれるものを指す。

サブルーチンに対する入出力変数は、引数または COMMON 変数である。これらの変数の入出力の解析を行なうのがインタープロシージャ解析 [6, 7] である。この手法では、サブルーチンコールグラフの最も深いところから順に、各サブルーチンの入出力変数の解析を行ない、それらの情報を呼び出し側 SB に伝播する。この時、各変数を呼び出し側の対応する変数にリネームすることで呼び出し側 SB における変数の入出力が得られる。これを繰り返すことで、全ての SB における入出力が得られる。入出力変数が配列変数の場合には、その入出力の添字範囲を解析することが重要である。そのためにはその配列変数の各次元の添字について、それらのサブルーチン内で取り得る値の範囲を調べることが必要である。その範囲の上限・下限が変数で表される場合には、それらの変数についてさらに解析していくことで、入出力範囲が正確に得られる。これらの入出力範囲を連立線形不等式にみたて、その解の有無によってデータ依存判定を行なう Global Region 手法 [6] が提案されている。ただし、この手法は範囲情報に冗長さが含まれる点とデータ依存判定に時間がかかってしまう点に問題がある。これに対して、配列の添字に用いられるループの制御変数とそのループの上限・下限値によって入出力範囲を表す Atom Image 手法 [7] が提案されており、これは Benerjee の exact 解析を用いたイタレーション間データ依存解析に有効である。さらにループの制御変数以外の変数で添字ループの初期値・終値に用いられる変数に対する解析を行なうことにより、より正確なデータ依存解析が可能である。

階層型マクロデータフロー処理において、SB には以下に示すような各手法が適用される。

- 階層型マクロデータフロー処理を適用して SB 内部の各階層で粗粒度並列処理を行う。
- 内部に SB を含む DO ループに、インタープロシージャ解析及びイタレーション間データ依存解析を行なった結果、
 - Do-all, Do-across 可能な場合、その DO ループ内部に中粒度並列処理を適用する。その際そのループ内から呼ばれるサブルーチンは内部での並列性を利用しないので 1 プロセッサ用のコードを生成する。
 - Do-all, Do-across 不可能な場合、そのループボディ部に階層型マクロデータフロー処理を適用してループ内の各 MT 間で粗粒度並列処理を行う。

以上のように SB には各階層において各種並列処理が適用されるが、同一のサブルーチンであっても呼び出される階層が異なると階層内の PC 数や PC 内の PE 数が異なるために生成するコードが異なる。また Do-all, Do-across ループから呼び出されるサブルーチンも他の場合と生成するコードが異なる。異なるコードを生成する場合は、サブルーチンをコピーする。また、それらのコピーされたサブルーチンを呼び出すそれぞれの SB 間にインタープロシージャ解析を適用した結果並列性が検出され、同時に実行される場合には、そのサブルーチン内の仮引数やローカル変数へのアクセスにメモリコンフリクトが生じる可能性があるため、これらの変数については PC 毎にメモリを確保し、メモリコンフリクトを回避する。

4 おわりに

本稿では OSCAR マルチグレインコンパイラ上での階層型マクロデータフロー処理におけるインタープロシージャ解析を用いたサブルーチン並列処理手法について述べた。提案手法では、インライン展開によるサブルーチン並列処理やインタープロシージャ解析を用いたサブルーチン並列処理に加えて階層型マクロデータフロー処理を適用することにより、自由度の高いサブルーチン並列処理を行なうことが可能になる。

今後、サブルーチンまたはサブルーチンを呼び出している階層に対して、インライン展開、DO ループ並列化、階層型マクロデータフロー処理などをどのサブルーチンにどの階層で適用するかをコンパイラが自動的に決定する手法を開発する予定である。

謝辞

本研究の一部は、文部省科学研究費 (一般研究 (b)05452354, 一般研究 (c)05680284) により行なわれた。

参考文献

- [1] 笠原, 並列処理技術, コロナ社, 1991-06
- [2] H.Kasahara, H.Honda, S.Narita, "A Multi- Grain Compilation Scheme for OSCAR," Proc. 4th Workshop on Languages and Compilers for Parallel Computing, Aug. 1991.
- [3] 本多, 岩田, 笠原, Fortran プログラム粗粒度タスク間の並列性検出手法, 信学論, J73-D-I(12), 12 1990
- [4] H.Kasahara, H.Honda, M.Iwata, M.Hirota, "A Macrodataflow Compilation Scheme for Hierarchical Multiprocessor System," Proc.Int'l Conf. on Parallel Processing, Aug. 1990.
- [5] 岡本, 合田, 宮沢, 本多, 笠原, OSCAR マルチグレインコンパイラにおける階層型マクロデータフロー処理手法, JSPP'93
- [6] R.Triolet, "Interprocedural Analysis for Program Restructuring with Paraphrase," CSR D Rpt. No. 538, University of Illinois at Urbana-Champaign, Dec. 1985.
- [7] Z.Li and P.C.Yew, "Interprocedural Analysis for Parallel Computing," CSR D, University of Illinois at Urbana-Champaign.