1L－1

階層型マクロデータフロー処理のための マクロタスクスケジューリング手法*

赤鹿 秀樹†, 岡本 雅巳†, 宮沢 稔‡, 安田 泰勲†, 笠原 博徳†

早稲田大学理工学部電気工学科†, 三菱電機‡

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ループ並列化 [1] が主に用いられていたが，プロセッサ台数の増加と共に，ループ並列化だけではスケーラブルな処理速度が望めなくなっている．そこで，筆者等は従来自動並列化が不可能であったループ以外の並列性，例えば，基本ブロック，ループ，サブルーチン間の並列性を利用した粗粒度タスクの並列処理 (マクロデータフロー処理) 手法 [2, 3, 4]，および基本ブロック内部の近細粒度並列処理手法を提案している．また，筆者等は粗粒度・中粒度 (ループ並列化)・近細粒度並列処理を階層的に適用する並列処理理手法であるマルチグレイン並列処理手法 [2]，さらにループ内あるいはサブルーチン内の粗粒度並列性を階層的に利用してクラスタ内部で階層的にマクロデータフロー処理を行なう階層型マクロデータフロー処理も提案している．この階層型マクロデータフロー処理では，粗粒度タスク間のスケジューリング方法として，ダイナミックスケジューリング，スタティックスケジューリングを使い分けて行なうことにより，スケジューリングの際に生じるオーバーヘッドを抑えるようにしている．本稿では，階層型マクロデータフロー処理におけるマクロタスクのスケジューリング手法について提案する．

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, 7] 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 内部では近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近細粒度並列処理，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，サブ MT は PC 内で定義されるサブ PC に割り当てられる．階層的に MT を生成後，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ごとに生成する．

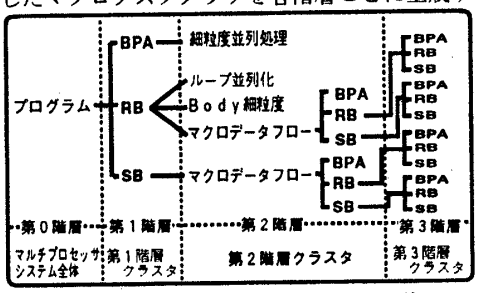

図 1: マクロタスクの階層的な定義

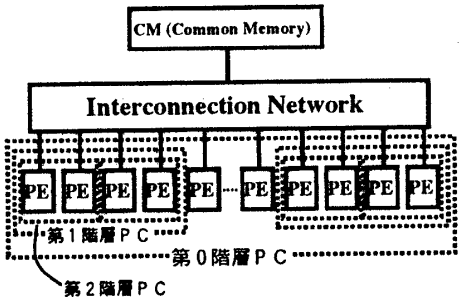

図 2: プロセッサクラスタの階層的な定義

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, 6] 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，MT1 つの割り当てにかかる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スケジューリング処理手法

3.1 ダイナミックスケジューリング処理手法

階層型マクロデータフロー処理におけるダイナミックスケジューリングでは，実行時に各階層の PC に各階層の MT を割り当てる．その際，集中型スケジューラ方式 [6] と分散型スケジューラ方式の 2 種類の方式をとることができる．集中型スケジューラ方式では，単一の PE をスケジューラとして使用し，分散型スケジューラ方式では，スケジューリングルーチンを各 PE に割り当てる．分散スケジューラ方式は共有メモリへの排他アクセスが低オーバーヘッドで実現でき，プロセッサ

*A Scheduling Scheme of Macro-tasks for Hierarchical Macro-dataflow Computation

†Hideki AKASHIKA, Masami OKAMOTO, Yasunori YASUDA, Hironori KASAHARA
†Department of Electrical Engineering, Waseda University
‡Minoru MIYAZAWA
‡MITSUBISHI Electric Corporation

数が少ない場合に有利である．一方，集中型スケジューラ方式は，PE 数が多く多数の階層を定義する場合に有利である．

本手法ではダイナミックスケジューリングアルゴリズムに，Dynamic-CP 法 [6] を採用している．これは，スタティックスケジューリングアルゴリズムの CP 法をダイナミックスケジューリング用に拡張したものである．Dynamic-CP 法ではコンパイル時に，各 MT の推定処理時間と，条件分岐を含む場合は分岐確率により，各 MT からマクロタスクグラフの出口ノードまでの最長パス長 (CP 長) を計算することにより得られる優先度に基づいて，実行時にスケジューリングルーチンが，実行可能な MT の PC への割当てを行なう．つまりスケジューラは，実行開始条件が満たされ，実行可能になった MT のうち優先度の高い MT から PC へ割り当てる．

3.1.1 コード生成及び実行手順

集中型スケジューラ方式 [6] では，第 k 階層プログラムのダイナミックスケジューリングルーチンを単一の PE に割り当て，第 k 階層 PC のコントロールプロセッサ (CP) として使用する．第 k 階層 PC の他の PE には第 k 階層プログラム内の全 MT の並列化オブジェクトコードを割り当てる．分散型スケジューラ方式では，スケジューリングルーチンが各マクロタスクの前後に挿入され，共有メモリ上のスケジューリング用管理テーブルに排他アクセスをしながら，スケジューリングを行なう．第 k 階層の並列実行管理コードは以下の機能からなる．

- ダイナミックスケジューリングコード
  1. 第 k 階層 PC 内各 PE への第 k 階層プログラムの MT の割り当て．
  2. MT 実行中の PE から通達される MT 実行情報 (分岐・終了) の監視．
  3. 実行開始条件の検査とレディMT キューへの MT の投入．
  4. レディMT キューの管理 (優先度順にソート)．
  5. 第 k 階層プログラム終了を親階層のスケジューラに通達．
- MT コード
  1. MT の割当を待つ．
  2. MT の実行と，実行情報のスケジューラへの通達．

各コードはコンパイル時にソースプログラム専用のものが，並列化オブジェクトコードと共に生成される．

3.2 スタティックスケジューリング処理手法

条件分岐を含まないプログラム，マクロタスク，あるいはマクロタスクグラフの一部にはスタティックスケジューリングを適用する．スタティックスケジューリングでは，コンパイル時に MT を各 PC に割り当て，MT 間に同期コードを挿入することにより各 PC 間で同期をとりながら実行を行なう．本手法で用いる同期のオーバーヘッドはダイナミックスケジューリングオーバーヘッドに比べてかなり小さく抑えられるため，より高速な実行が可能である．しかし，スタティックスケジューリングは MT 中に条件分岐が存在する場合，制御依存を受ける MT はコンパイル時に効果的なスケジューリングができない．その場合，以下の手法により条件分岐を含まない部分でグループ化 (以下このグループを SS-MTG(Statically Scheduled-Macro-Task Group) とする) を行なう．

1. マクロタスクタスクグラフ中から，MT が実行されることが最も速く確定する条件である実行確定条件 [3] が等しい，つまり制御依存が等しい MT 集合を抽出する．
2. 抽出された MT 集合中の MT の先行タスク，後続タスクに MT 集合中に属さない MT が存在する場合，その MT を MT 集合から分割して新しく MT 集合を生成する．
3. これらの分割により分かれた各 MT 集合を SS-MTG とする．

そして，SS-MTG 内の MT に対してはスタティックスケジューリングを適用し，SS-MTG 自身は SS-MTG 実行のために挿入したダイナミックスケジューリングコードにより，動的に制御するという手法をとる．

3.2.1 コード生成及び実行手順

第 k 階層内プログラムコードは以下のように生成される．

1. プログラム (第 k 階層マクロタスクタスクグラフ) を SS-MTG に分割し，MT 間に存在する冗長なデータ依存を取り除く．
2. 分割された SS-MTG 内の MT を，データ転送を考慮したスタティックスケジューリング手法である DT/CP/MISF 法 [1] を用いて PC へ割り当てる．
3. 各 PE に MT 内の並列化オブジェクトコードと MT 間同期コードの生成を行なう．
4. SS-MTG をダイナミックスケジューリングするためのスケジューリングコードを生成する．

SS-MTG は 3.1節で述べたダイナミックスケジューリング手法により以下のように実行される．

1. SS-MTG の分岐方向は単一 PE で評価され，SS-MTG の分岐方向が決定した時点でその PE が次にどの SS-MTG を実行するかを同一階層内の全 PE に通達する．
2. 各 PC に割り当てられた SS-MTG 内の全 MT の実行が終了後，同一階層の全 PC 間でバリア同期を行なう．
3. バリア同期後，各 PE は 1 により通達された SS-MTG の実行を行なう．
4. 1〜3 を終了まで繰り返す．

4 性能評価

本節では，対称帯状マトリクス係数行列を持つ連立方程式の解法である CG 法 (Conjugate Gradient Method) のプログラムを用いたスケジューリング手法を OSCAR[7] 上で行なった時の性能評価について述べる．PE が 6 台の場合，ダイナミックスケジューリング方式ではシーケンシャル実行時間の 3.11 倍，スタティックスケジューリング方式ではシーケンシャル実行時間の 4.10 倍という結果が得られた．

5 むすび

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スケジューリング手法について述べた．提案手法では，ダイナミックスケジューリングとスタティックスケジューリングを併用した階層型マクロデータフロー処理を適用することにより，低オーバーヘッドの実行ができる．今後最適なスケジューリング手法をどの階層で適用するかをコンパイラが自動判定する手法を開発する予定である．

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) (1990-12).

[4] H.Kasahara, H.Honda, M.Iwata, M.Hirota, "A Compilation Scheme for Macro-dataflow Computation on Hierarchical Multiprocessor Systems," Inter. 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, 情処論, Vol.35(4) (1994-4).

[6] 本多, 合田, 岡本, 笠原, Fortranプログラム粗粒度タスクの OSCAR における並列実行方式, 信学論, J75-D-I(8), (1992-8).

[7] 笠原, 本多, 橋本, OSCAR のアーキテクチャ, 信学論 D, Vol.38,No1, (1989-1).