C-014

階層統合型粗粒度タスク並列処理におけるマクロタスク生成手法

Macrotask Generation Scheme for Layer-unified Coarse Grain Task Parallel Processing

吉田 明正†
Akimasa Yoshida

1 はじめに

マルチプロセッサシステム上での並列化コンパイラを用いた並列処理では，従来よりループ並列化技術 [1] が広く用いられてきたが，近年，高い実効性能を達成するために粗粒度タスクレベルの並列性 [2, 3] を利用する粗粒度タスク並列処理が有効と考えられている．このような粗粒度タスク並列処理 [2] では，対象プログラムをループやサブルーチン等の粗粒度タスクに階層的に分割し，それらを階層的にグループ化したプロセッサグループ上で並列実行していた．しかしながら，プロセッサ数に制限がある場合には，全階層の粗粒度タスク間並列性を利用できない可能性があり，異なる階層間にまたがった粗粒度タスク並列性を最大限に利用する階層統合型粗粒度タスク並列処理 [4] が提案されている．

本稿では，この階層統合型粗粒度タスク並列処理において，ダイナミックスケジューリングオーバヘッドと並列性を考慮し，実行時間を最小化するマクロタスク生成手法を提案する．また，本稿ではダイナミックスケジューリングオーバヘッドを考慮したシミュレーションにより，その性能を評価する．

2 階層統合型粗粒度タスク並列処理

階層統合型粗粒度タスク並列処理では，粗粒度タスク並列処理手法 [2] で用いられている並列性抽出技術を用いて，階層型マクロタスクグラフ（MTG）を生成し，その階層型 MTG に対して階層開始マクロタスク [4] を導入する．その後，全階層のマクロタスクを統一的に取り扱い，最早実行可能条件を満たした粗粒度タスク（マクロタスク）から順に，プロセッサに割り当てるダイナミックスケジューリングルーチンを生成する [2, 4].

ここで，従来の粗粒度タスク並列処理 [2]（階層型実行制御方式と呼ぶ）では，各階層のマクロタスク（MT）は，プロセッサグループ（PG）に階層的に割り当てられて実行される．例えば，第1階層が MT1～MT8，MT5 内部の第2階層が MT51～MT53，MT51 内部の第3階層が MT511～MT512（ループでそれぞれ2回実行される）で構成されているプログラムを，2PG*2PE（各 PG は 2PE 構成で計 4PE）上で実行したイメージは，図1(a) のようになる．それに対して階層統合型粗粒度タスク並列処理 [4] では，図1(b) に示すように，全階層のマクロタスクが統一的に取り扱われ，異なる階層のマクロタスク間並列性が最大限に利用される．

3 MTG 並列度を考慮したマクロタスク生成手法

階層統合型粗粒度タスク並列処理では，通常，全ての階層のマクロタスクをスケジューリングの対象とすることにより，プログラム中の並列性を最大限に利用することが可能となる．しかしながら，対象プログラムの上位階層のマクロタスク間並列性が，全 PE を有効利用するのに十分な場合，下位階層のマクロタスク間並列性を利用する必要はない．そこで，本手法では，上位階層の

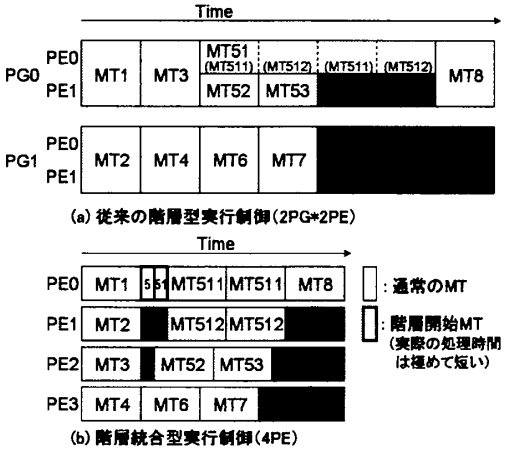

図1 粗粒度タスク並列処理の実行イメージ.

MTG から順に利用できる並列性を解析し，十分な並列性を確保できた段階で，その階層のマクロタスクを最下位階層マクロタスク（その内部のマクロタスクに対してはダイナミックスケジューリングを適用しない）と定義し，そのマクロタスクを1回のダイナミックスケジューリングで PE に割り当てる．この結果，並列性を最大限に利用した上で，ダイナミックスケジューリングオーバヘッドを軽減することが可能になる．

3.1 MTG 並列度による最下位階層 MTG の検出

本手法では，各階層の MTG_i に対して，MTG_i のクリティカルパス長（入口ノードから出口ノードまでの最長パス長）CP_{MTG_i} と，MTG_i の逐次処理コスト Seq_{MTG_i} を求める．なお，CP_{MTG_i} は PE 数が無制限の場合の MTG の最小処理時間であり，Seq_{MTG_i} は 1PE で実行した処理時間となる．このコスト算出の際，各マクロタスクの処理時間としては，逐次処理コストを用いる．ここで，MTG_i の並列度（MTG_i の平均的な粗粒度タスク間並列性）$Para_{MTG_i}$ は，Seq_{MTG_i}/CP_{MTG_i} として求められる．

次に，上位階層の MTG_i から順に，その並列度 $Para_{MTG_i}$ を，MTG_i の処理に必要な PE 数として確保する．下位階層の MTG_j を並列実行するときには，その MTG の上位階層 MT の PE も使用することができるため，MTG_j のために別途必要な PE 数は，$Para_{MTG_j}-1$ となる．なお，PE に余りがなくなった場合は，その MTG_j は最下位階層 MTG として定義され，その MTG 内部の MT は最下位階層 MT となる．

3.2 スケジューリング時間を考慮した並列処理時間による最下位階層 MTG の決定

前節で求めた最下位階層 MTG は，全 PE を利用するのに十分な並列度を得るために必要となる最下位階層の MTG のことである．但し，この最下位階層 MTG_i は，上位階層 MTG の並列度によっては，必ずしも $Para_{MTG_i}$ の PE 数を確保できる訳ではない．ここでは，確保でき

†東邦大学理学部情報科学科
Department of Information Science, Toho University

る PE 数（上位階層の 1PE を含む）を $GivenPE_{MTG_i}$ と表す．

本手法では，最下位階層 MTG_i に対して，1PE 上での逐次処理時間と $GivenPE_{MTG_i}$ 台の PE 上での並列処理時間を求める．この際，1 つの MT を割り当てるのに要するダイナミックスケジューリング時間を $ScheCost$ とする．スケジューリングオーバヘッドを考慮した逐次処理時間は，$Seq_{MTG_i}+ScheCost$ として求められ，スケジューリングオーバヘッドを考慮した並列処理時間は，$MAX(CP_{MTG_i}, Seq_{MTG_i}/GivenPE_{MTG_i})+ScheCost*MTnum_{MTG_i}/GivenPE_{MTG_i}$ として求められる．$MTnum_{MTG_i}$ は MTG_i を構成する MT 数とする．

ここで，MTG_i のスケジューリングオーバヘッドを考慮した逐次処理時間が，スケジューリングオーバヘッドを考慮した並列処理時間より短い場合には，MTG_i を 1PE で実行した方が処理時間が短縮可能であるため，MTG_i の並列性は利用されることなく，MTG_i の上位 MT が最下位階層 MT として定義され，この MTG_i は 1PE で実行されることになる．

4 性能評価

本性能評価では，図 2 に示す 8 階層からなるマクロタスクグラフ（MTG）を用いて，階層統合型粗粒度タスク並列処理のスケジューリングオーバヘッドを考慮したシュミレーションにより実行時間を求め，提案するマクロタスク生成手法の評価を行う．

評価に用いる図 2 の 8 階層 MTG は，第 1 階層（上位階層）が 9 個のマクロタスク（MT）により構成されており，MT1～MT6 及び MT9（図中で背景白）は内部 MTG を持たない処理時間 100[u.t.] の MT として定義する．MT7～MT8（図中で背景グレー）はそれぞれ 2 回転の繰返し文で，内部（ボディ部）に第 2 階層 MTG を持っており，MT7 の内部が MT21～MT29 からなる第 2 階層 MTG であることを表している．また，MT8 の場合には図示されていないが，MT7 と同様に内部に第 2 階層 MTG を持っているものとする．第 2 階層 MTG においても，MT21～MT26 及び MT29 は，処理時間 100[u.t.] の MT であり，MT27～MT28 は内部に第 3 階層 MTG を持つ MT とする．第 3 階層 MTG の構成も第 2 階層 MTG と同等とする．なお，第 8 階層 MTG の MT81～MT89 は，処理時間 100[u.t.] の MT とする．

本評価では，1 つの MT をプロセッサに割り当てるダイナミックスケジューリング時間を，MT（内部に MTG を持たないもの）の平均処理時間の 20%（20[u.t.]）とし，また，スケジューリング処理は各 PE で分散して行う分散スケジューリング方式（全 PE 共通のレディ MT キューのアクセスは排他制御）を対象としている．並列実行の結果は図 3 に示す通りであり，8PE 実行の場合，従来の MT 生成手法では 4.72 倍の速度向上（1PE の逐次処理時間比）しか得られなかった．それに対して，提案する MT 生成手法では，不必要なダイナミックスケジューリングオーバヘッドを軽減することが可能であり，7.41 倍の速度向上が得られている．次に，MT の粒度がダイナミックスケジューリングコストに比べて相対的に小さい場合を想定し，ダイナミックスケジューリング時間を平均 MT 処理時間の 30%（30[u.t.]）とすると，従来の MT 生成手法では 8PE で 3.14 倍の速度向上しか得られなかったが，提案する MT 生成手法では 6.17 倍の速度向上が得られている．

以上の結果から，提案手法は，PE 数が増加した場合やスケジューリング時間が増加した場合にも，並列性を有効利用しつつダイナミックスケジューリングオーバヘッドを軽減することが可能であり，顕著に処理時間が短縮されることが確かめられた．

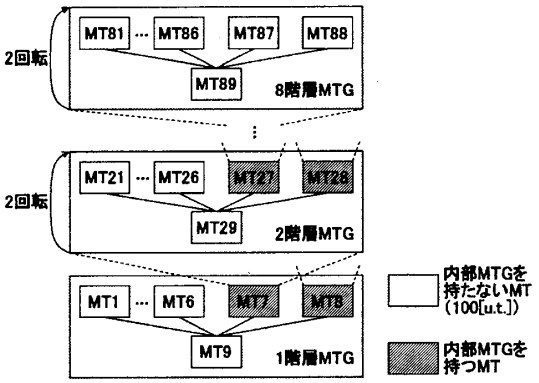

図 2 評価用 8 階層プログラムの MTG.

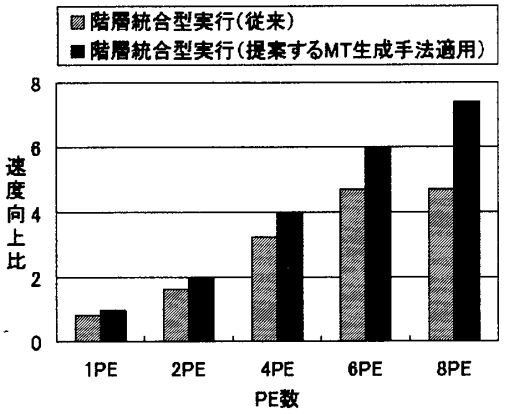

図 3 8 階層 MTG における階層統合型実行の結果.

5 おわりに

本稿では，階層統合型粗粒度タスク並列処理のための並列度を考慮したマクロタスク生成手法を提案した．本手法では，上位階層で十分な並列性が得られた場合に，下位階層の MT 集合は 1 回のダイナミックスケジューリングで PE に割り当てることにより，ダイナミックスケジューリングオーバヘッドを軽減することが可能である．

シミュレーションによる性能評価の結果，提案する MT 生成手法の適用により，不必要なダイナミックスケジューリングオーバヘッドを軽減することが可能であり，階層統合型粗粒度タスク並列処理の有効性が確かめられた．

今後の課題としては，ベンチマークプログラム等で有効性を評価することがあげられる．

参考文献

[1] R. Eigenmann, J. Hoeflinger, and D. Padua. On the automatic parallelization of the Perfect benchmarks. *IEEE Trans. on Parallel and Distributed System*, Vol. 9, No. 1, Jan. 1998.

[2] 笠原博徳, 小幡元樹, 石坂一久. 共有メモリマルチプロセッサシステム上での粗粒度タスク並列処理. 情報処理学会論文誌, Vol. 42, No. 4, 2001.

[3] X. Martorell, E. Ayguade, N. Navarro, et al. Thread fork/join techniques for multi-level parallelism exploitation in NUMA multi-processors. *Proc. of ICS*, 1999.

[4] 吉田明正. 粗粒度タスク並列処理のための階層統合型実行制御手法. 情報処理学会論文誌, Vol. 45, No. 12, 2004.