

# SMP 上での階層型粗粒度並列処理のための 多重割当てを伴う分散型ダイナミックスケジューリング

矢作 浩 板橋 瑞穂 吉田 明正

東邦大学 理学部 情報科学科

## 1 はじめに

自動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手法 [1] が用いられているが、単一ループのイタレーション間の並列性しか利用することができないという問題点があった。この問題点を解決するために、ループ等の粗粒度タスクレベルで並列処理を行う階層型マクロデータフロー処理 [2] が提案されている。階層型マクロデータフロー処理ではダイナミックスケジューリングを用いることによって粗粒度タスクをプロセッサクラスタ(PC)に実行時に割り当て並列処理を行う。一般的には1PCに1つの粗粒度タスクが割り当てられるが、粗粒度タスク内の並列性によってはPC利用率が低下してしまうことがある。そこで本稿では、複数の粗粒度タスクを同一PCに多重に割り当てることにより、PC利用率を向上させる多重割当てを伴う分散型ダイナミックスケジューリング手法を提案する。また、本稿では、SMP(Sun Enterprise3000)上で行った性能評価についても述べる。

## 2 階層型マクロデータフロー処理

階層型マクロデータフロー処理 [2] では、コンパイラがプログラムを擬似代入文ブロック (BPA)、繰り返しブロック (RB)、サブルーチンブロック (SB) の3種類のマクロタスク (MT) と呼ばれる粗粒度タスクに分割する。この時、MT内にサブMTを含む場合には、それらを第2階層MTとして定義する。

次に、各MT間のコントロールフロー、データフローを解析し各MTの最早実行可能条件を求め、マクロタスクグラフ (MTG) [2] を生成する。

実行時には、各MTはコンパイラによって生成されたダイナミックスケジューリングルーチンによりプロセッサクラスタ (PC:グループ化されたプロセッサ集合) に割り当てられて並列処理される。ダイナミックスケジューリングの基本的なアルゴリズムとしては Dynamic-CP 法 [3] を用いるが、提案手法では第4章で述べるアルゴリズムを採用する。

Distributed Dynamic Scheduling with Overlapping Assignment for Hierarchical Coarse Grain Parallel Processing on SMP

Hiroshi Yahagi, Mizuho Itahashi, Akimasa Yoshida  
Department of Information Science, Toho University

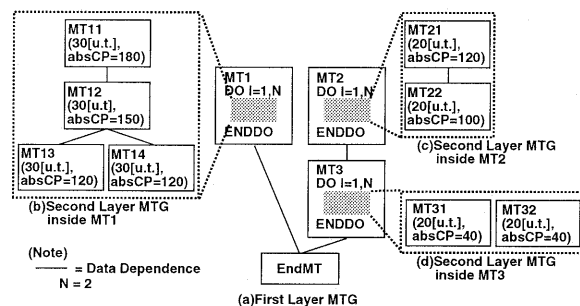


図 1: プログラム 1 の MTG.

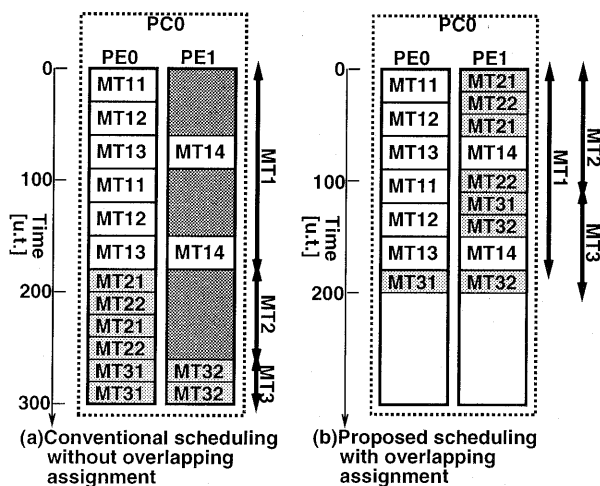


図 2: プログラム 1 の実行イメージ.

## 3 多重割当てを伴うダイナミックスケジューリングの概念

通常の階層型マクロデータフロー処理では、1つのPCには1つのMTが割り当てられる。しかし、対象プログラムのMT内の並列性によってはPCの利用率が低下してしまうことがある。

例えば、図1のプログラムを1PC(内部は2PE構成)で実行した場合、図2(a)のような実行イメージになり、PC0内のプロセッサPE1にはアイドル状態が生じる。それに対して提案手法では、図2(b)のように、PC0にMT1, MT2, MT3を多重に割り当てることにより、PCの利用率が向上している。

このように、従来のダイナミックスケジューリング手法では、MT内のサブMT間の並列性が低い場合に、PCの利用率が低下してしまうが、本手法

ではMTをPCに多重に割り当てることによってPCの利用率を100%に近づけることが可能になる。

## 4 多重割当てを伴う分散型ダイナミックスケジューリング

前述の多重割当てを伴うダイナミックスケジューリングを実現する場合、集中スケジューラ方式と分散スケジューラ方式が考えられる[4][5]。本稿では分散スケジューラ方式の実現方法について述べる。

分散型ダイナミックスケジューラの動作は大きく分けて、上位階層分散スケジューラと下位階層分散スケジューラに分けることができる。

### 4.1 上位階層分散スケジューラ

上位階層分散スケジューラは、プログラム開始時及び下位階層分散スケジューラから呼び出された時に、上位階層MTのスケジューリングを行う。本スケジューラでは、上位階層MTの最早実行可能条件をチェックし、実行可能MTをレディキューに投入し、CP(Critical Path)長の大きい順にソートする[3]。

次に、レディキューの先頭のMTをPC利用率が最小のPC(利用率は100%未満)に割り当てる。

### 4.2 下位階層分散スケジューラ

下位階層分散スケジューラでは各PCに割り当てられた上位階層MT内のサブMTのスケジューリングを行う。各PCはそれぞれレディキューを持っており、分散スケジューラ方式では自PEに対して割当てを行う。自PEを含むPCに割り当てられている上位階層MTのサブMTに対して最早実行可能条件をチェックし、実行可能サブMTを自PCのレディキューに登録し、絶対CP長(図1においてabsCPと表記)[5]の大きい順にソートする。

次に、自PCのレディキューの先頭のサブMTを自PEに割り当て、そのサブMTを実行する。ここで、下位階層分散スケジューラが実行されているということは、そのPEに対してはサブMTが割り当てられていないことを意味する。

サブMTの実行が終了したら、各サブMTのステータス情報[4]に登録する。この時、上位階層のステータス情報に変更があった場合には、上位階層分散スケジューラを呼び出し、上位階層MTのスケジューリングを行う。

## 5 SMP上での性能評価

本章では多重割当てを伴う分散型ダイナミックスケジューリングを用いた階層型マクロデータ処理を、Enterprise3000(UltraSPARC250MHz, メモリ256MB)の4PE上でSolarisスレッドを用いて実装し、性能評価した結果について述べる。

性能評価に用いた図3のプログラムは第1階層が5つのMTで構成されており、各MTに対してサブMTが存在する。

このプログラムの多重割当てなしの階層型マクロデータフロー処理による実行結果は、1PC\*2PE構

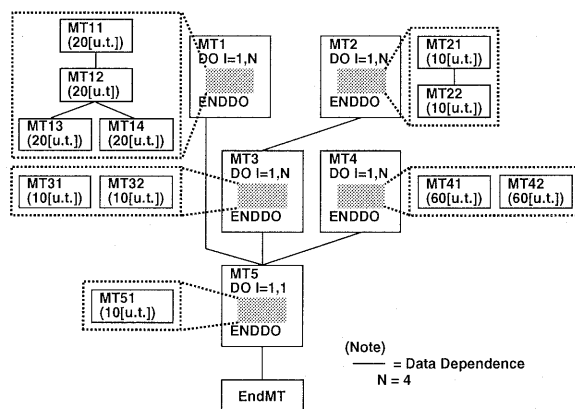


図3: プログラム2のMTG.

成(第1階層を1PCとし、PC内の第2階層を2PE構成とする)で1.57倍、1PC\*4PE構成で1.57倍、2PC\*2PE構成で2.58倍の速度向上となった。ここで、1PC\*4PE構成の場合に性能が出ていないのは、MT内のサブMT間の並列性が少ないためである。

次に、提案する多重割当てを伴うダイナミックスケジューリングを用いた階層型マクロデータフロー処理による実行結果は、1PC\*2PE構成で1.96倍、1PC\*4PE構成で3.58倍、2PC\*2PE構成で3.59倍の速度向上となった。これより、多重割当てを伴うダイナミックスケジューリングによる速度向上が確認できた。

## 6 おわりに

本稿では、階層型マクロデータフロー処理における、粗粒度タスクの多重割当てを伴う分散型ダイナミックスケジューリング手法を提案した。本手法では、複数の粗粒度タスクを同一PCに多重に割り当てることにより、PC利用率を向上させることが可能である。また、提案手法をSMP上でSolarisスレッドを用いて実装し評価したところ、実行時間が20%~29%短縮され、提案手法の有効性が確認された。

本研究の一部は科学研究費補助金(奨励研究(A)12780243)により行われた。

## 参考文献

- [1] M. Wolfe.: "High performance compilers for parallel computing." Addison-Wesley Publishing Company(1996).
- [2] 岡本雅巳, 合田憲人, 宮沢稔, 本多弘樹, 笠原博徳: "OS-CAR マルチグレインコンパイラによる階層型マクロデータフロー処理手法", 情報処理学会論文誌, Vol.35, No.4, pp513-521 (1994).
- [3] 笠原博徳: "並列処理技術", コロナ社 (1991).
- [4] 合田憲人, 岩崎清, 岡本雅巳, 笠原博徳, 成田誠之助: "共有メモリ型マルチプロセッサシステム上でのFortran粗粒度タスク並列処理の性能評価", 情報処理学会論文誌, Vol.37, No.3, pp418-429 (1996).
- [5] 吉田明正: "階層型粗粒度タスク並列処理のための多重割当てを伴うダイナミックスケジューリング", 情報処理学会計算機アーキテクチャ研究会, 2000-ARC-139-31(2000).