

2D-4

# OSCAR マルチグレイン並列化コンパイラにおける階層的並列処理手法

山本 晃正<sup>†</sup>, 稲石 大祐<sup>†</sup>, 宇治川 泰史<sup>†</sup>, 小幡 元樹<sup>†</sup>, 岡本 雅巳<sup>†</sup>, 笠原 博徳<sup>‡</sup>

<sup>†</sup>早稲田大学理工学部電子電気情報工学科, <sup>‡</sup>(株) 東芝

## 1 はじめに

マルチプロセッサシステムを並列処理の専門家だけでなく誰もが高実行効率で使用できるようにする1手段として自動並列化コンパイラの開発が進められている。しかし、プロセッサ数の増加に対して従来のループ並列性 [5, 6] のみを抽出するコンパイラ技術では、従来の中粒度並列処理手法 (ループ並列化) に加えて、粗粒度並列性を用いた並列処理手法 (マクロデータフロー処理手法) [1, 2, 3] や基本ブロック内部の近細粒度並列処理手法 [4] を階層的に用いるマルチグレイン並列処理手法 [1, 4] を提案している。このマルチグレイン並列処理において常に最小の処理時間を得るためには粗粒度・中粒度・近細粒度並列処理の各処理方式をプログラム中のどの部分階層に適用するかが重要なファクタとなる。本稿では処理しようとするプログラムの並列性を考慮した各プログラム階層に対する適切な処理手法とプロセッサクラスタリングを与える手法を提案する。

## 2 並列処理の粒度と階層型マクロデータフロー処理

OSCAR Multi-Grain FORTRAN Compiler におけるマルチグレイン並列処理では、近細粒度・中粒度・粗粒度の3種類の並列処理手法を用いる。近細粒度並列処理とは、基本ブロック内のステートメントをスタティックスケジューリングを用いてプロセッサ (例えばシングルチップマルチプロセッサ内の各プロセッサ) に割り当てて並列処理を行う方式である。また中粒度並列処理はループのイタレーション間の並列性を用いたループ並列処理である。粗粒度並列処理はループ・サブルーチン・基本ブロック等の間の粗粒度並列性を用いた並列処理手法である。

粗粒度並列処理のための階層型マクロデータフロー処理では、プログラムを次に示す3種類のマクロタスク (MT) で表し、階層的に分割する。

- BPA (Block of Pseudo Assignment statements)  
基本ブロック, 及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でないサブルーチン

このように分割された MT は各階層で MT 間のコントロールフロー解析やデータフロー解析を行うことにより階層毎のマクロフローグラフ (MFG) が生成される。さらに MT 間の

並列性を抽出するために各階層で最早実行可能条件解析が行われ、解析によって得られた結果は図3のように各階層毎のマクロタスクグラフ (MTG) で表現される。各階層の MT はスタティックスケジューリングまたはダイナミックスケジューリングによって、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不確定性 (条件分岐等) が存在する場合に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、処理とデータ転送のオーバーラッピング等も含めて、データ転送・同期・スケジューリングオーバーヘッドの最小化可能なスタティックスケジューリングを用いる。

図1に示すように MT が BPA ならば BPA 内部のステートメントレベルの近細粒度並列処理が、RB ならばループ並列処理やシーケンシャルループに対してボディ部の近細粒度並列処理あるいはマクロデータフロー処理、SB もマクロデータフロー処理が適用される。階層的マクロデータフロー処理は図1に示すようにプログラムのネストレベルにより複数階層で定義でき、どの階層で並列処理を行うかは実行性能に大きな影響を与える。

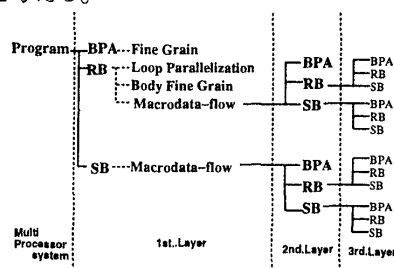


図1: マクロタスクの階層的定義

また MT 分割に対応してプロセッサクラスタ (以下 PC) も図2のように階層的に定義される。第0階層をマルチプロセッサシステム全体と定義し、第1階層 PC を、第0階層 PC 内のプロセッサエレメント (以下 PE) をグループ化して定義される PC とする。そして、各階層の PC はマクロタスクグラフの各階層の並列性に合わせて定義され、その階層の MTG は PC 間で並列処理される。また各階層の PC に割り当てられた MT は、PC 内の PE あるいは内部で定義される下位階層 PC により並列処理される。

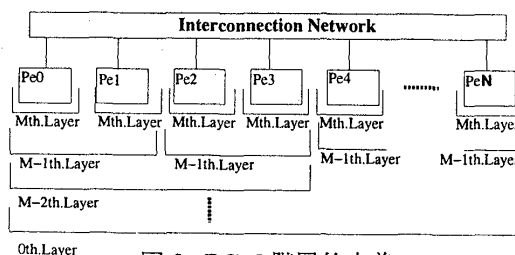


図2: PC の階層的定義

上述のような最も効果的な並列処理が行えるプログラム (MTG) の階層及びそれに合った PC の階層を選択するマルチグレイン並列処理は、従来の中粒度並列性のみを用いた並列処理と違い粗粒度・近細粒度という従来用いていなかった

\* An Hierarchical Parallel Processing Scheme in OSCAR Multi-Grain Parallelizing Fortran Compiler  
Terumasa YAMAMOTO<sup>†</sup>, Daisuke INAISHI<sup>†</sup>, Yasushi UJIGAWA<sup>†</sup>, Motoki OBATA<sup>†</sup>, Masami OKAMOTO<sup>†</sup>, Hironori KASAHARA<sup>†</sup>

<sup>†</sup> Department of Electrical, Electronics and Computer Engineering, Waseda University

<sup>‡</sup> TOSHIBA Corporation

並列性を使用することができるため、プロセッサの利用効率の向上が期待できる。

### 3 階層の自動決定手法

マルチグレイン並列処理を効果的に実現するためには前述のようにどの階層の MTG をどのグレインで並列処理するかという処理階層の決定方法と、各階層で何台のプロセッサあるいはプロセッサクラスタを割り当てるかを決定する方法が重要となる。ここでは各階層における処理方式及びプロセッサクラスタ数の決定方法を提案する。

#### 3.1 各階層の並列性

各階層の MTG に対して最高の処理効率を与える処理方式及びそれに合った PC 数・PE 数を決定していくためには、各階層中の並列性 (以下 Para) を考慮する必要がある。並列性の高い階層にプロセッサを多く割り当てることにより、プロセッサの利用効率を上げることが可能となる。

ここである MTG<sub>i</sub> の並列性 Para<sub>i</sub> を以下のように定める。

$$\text{Para}_i = \frac{1\text{PC での逐次処理時間}}{\text{MTG のクリティカルパス長}}$$

この式は直観的には、MTG<sub>i</sub> が単一の DOALL 可能な RB であればその回転数を、複数の MT からなりマクロデータフロー処理が適用される場合には [Para<sub>i</sub>] がクリティカルパス長で並列処理を行うためのプロセッサ数の下限を表しているということが分かる。すなわち各階層にはこれ以下の PC 数では最小処理時間が得られないことが分かる。

#### 3.2 各階層における処理方式決定手法

マルチグレイン並列処理における各階層の並列処理ではなるべく粒度の粗い外側階層をなるべく多くの PC あるいは PE で並列処理することがプロセッサ利用率の面から好ましい。すなわち、一般的に外側階層の MT の処理時間は大きい相対的にスケジューリングオーバーヘッドを小さくすることが出来るとともに、外側の方が内側に比べ同期の繰り返し回数が少ないためにトータルとして同期オーバーヘッドを小さくできる。しかし外側階層で並列処理を行う場合には、個々の MT の処理時間が大きいために負荷のアンバランスによるプロセッサ利用率の低下が生じる可能性があることを注意しなければならない。

以上の点を考慮しつつ 3.1 で計算した並列度 Para を用いて、次の手順で各階層における処理方式及び PC 数の決定を行う。ここでマクロタスクグラフ MTG<sub>i</sub> の並列度を Para<sub>i</sub>、その MTG<sub>i</sub> を実行する際に使用できる PC あるいは PE 数を PEnum<sub>i</sub> とする。

1. MTG<sub>i</sub> を実行するために生成すべき PC 数を  $\text{PCnum}_i = \lceil \text{Para}_i \rceil$  と決め MTG<sub>i</sub> 中の各 MT を実行する PE 数 (PC 中の PE 数) をこれより  $\text{PEnum}_{i+1} = \text{PEnum}_i / \text{PCnum}_i$  とする。ただし  $\lceil \text{Para}_i \rceil > \text{PEnum}_i$  の時は  $\text{PCnum}_i = \text{PEnum}_i$  とする。
2. 1. の手順を MTG<sub>i</sub> の各 MT 内部の MTG に階層的に適用する。

### 4 OSCAR 上での性能評価

本章では、提案手法を CG 法プログラムに適用し OSCAR シミュレータ上で評価した結果について述べる。図 3 に CG 法プログラムの階層型マクロタスクグラフを示す。

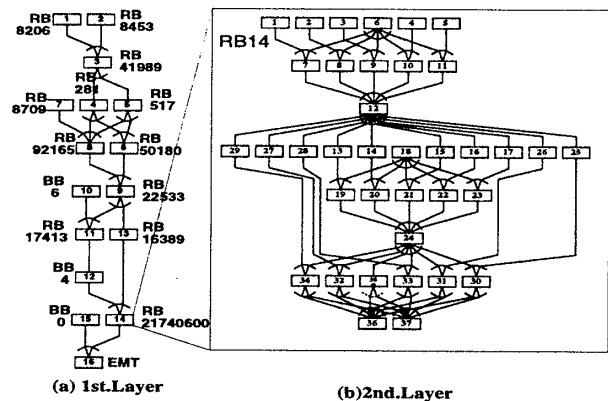


図 3: CG 法の階層型マクロタスクグラフ

図 3(a) は CG 法の第 1 階層の MTG を表しており、各 MT の隣の数字は処理時間を表している。第 1 階層の並列度は  $\text{Para}_1 = \frac{22007445}{21923674} = 1.00$  であり、並列性がないことが分かる。そこで第 1 階層では単一の PC を生成することにする。次に第 1 階層の処理時間のほとんどを占める RB14 に着目する。RB14 内部の MTG を図 3(b) に示す。この MTG の並列性は  $\text{Para}_{1,14} = \frac{217406}{38654} = 5.62$  となるので PC 数を 5 とする。この第 1 階層を PC=1、第 2 階層を PC=5 とするのが適切かを評価するために、第 1 階層を 1PC、RB14 内部 MTG を 3 から 8 までの PC 数で実行した時の処理時間を表 1 に示す。この表から PC 数が 3 から 5 までは処理時間が効果的に短縮されるが 5 以上では顕著な速度向上が得られず提案した自動階層決定手法が有効であることが分かる。

表 1: CG 法での PC 割当時の性能評価

PC 数	3	4	5	6	7	8
処理時間 [sec]	0.98	0.93	0.62	0.61	0.61	0.58
スピードアップ率	2.74	2.88	4.37	4.40	4.38	4.66

### 5 まとめ

本稿ではマルチグレインコンパイラにおける階層の自動決定手法について述べた。

今後の課題としては、本手法の SPEC、Perfect Club 等を用いた性能評価が挙げられる。

本研究の一部は通産相次世代情報処理基礎技術開発事業マルチプロセッサコンピューティング領域研究により行われた。

### 参考文献

- [1] 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.
- [2] 本多, 岩田, 笠原, "Fortran プログラム粗粒度タスク間の並列性の検出手法", 信学論, J73-D-I(12), Dec. 1991.
- [3] 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.
- [4] 笠原 博徳, "並列処理技術", コロナ社, Jun. 1991.
- [5] M.W.Hall, J.M.Anderson, S.P.Amarasinghe, B.R.Murphy, S.-W.Liao, E.Bugnion, M.S.Lam, "Maximizing Multiprocessor Performance with the SUIF Compiler", IEEE Computer, Dec. 1996.
- [6] R.Eigenmann, J.Hoefflinger, D.padua, "On the Automatic Parallelization of the Perfect Benchmarks", IEEE Transaction on parallel and distributed systems, Vol.9, Jan. 1998.