

石坂一久†, 小幡元樹†, 瀧康太郎†, 笠原博徳†
早稲田大学理工学部電気電子情報工学科

1 はじめに

主記憶共有型マルチプロセッサ (SMP) アーキテクチャはパソコンレベルサーバーからスーパーコンピュータまで広く使用されており, 将来的にはゲーム用チップやホームサーバー等として用いられると予想される。

このようなマルチプロセッサシステムは並列処理に関する専門技術のないユーザーでも簡単に使用できなければならない。このためには逐次プログラムを自動的に並列化するコンパイラ [1, 2, 3] の開発が重要である。従来の SMP 用自動並列化コンパイラはループ並列化のみを行っていたが, ループ並列化技術では今後大幅に並列処理性能を伸ばすのは困難である。この問題を解決するためには, さまざまな粒度の並列性を階層的に用いるマルチグレイン並列処理の実現が重要である [1]。

本稿ではマルチグレイン並列処理を各種 SMP マシンで実現するために, SMP 用標準 API である OpenMP を用いてマルチグレイン並列処理プログラムを自動生成する方法を提案する。本手法ではコンパイラが, 逐次 FORTRAN プログラムから OpenMP ディレクティブを加えた並列プログラムを自動生成することで, プログラムの変更をすることなく, 多くの SMP マシン上でマルチグレイン並列処理を行うことが可能となる。

2 マルチグレイン並列処理

SMP 上での OpenMP を用いたマルチグレイン並列処理では, 処理のオーバーヘッドを考慮して粗粒度・中粒度の 2 種類の並列性を用いる。粗粒度並列処理とはループ・サブルーチン・基本ブロック等の間の並列性を用いた並列処理手法で, 中粒度並列処理と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。

粗粒度並列処理のための階層型マクロデータフロー処理 [6] では, プログラムを次に示す 3 種類のマクロタスク (MT) [7] 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

さらに実行時間の大きいループ並列処理困難な RB あるいは SB に関しては, その内部に対しても同様なマクロタスク分割を行い, 図 1 における RB2 のようにサブ MTG を生成することによって, 階層的に粗粒度タスク並列処理を適用する。

このように分割された MT に対して, 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い, 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 [1] を行う。解析によって得られた結果は各階層毎に図 1 のようなマクロタスクグラフ (MTG) で表現される。図 1 ではプログラム全体を第一階層とし, 200 回転の DOALL1 とループ並列化の困難な回転数不定の収束ループ RB2 と BB3 に分割する。さらに, シーケンシャルループ RB2 の内部を第二階層として, BB2.1, LOOP2.2, LOOP2.3, BB2.4 の 4 つの MT に分割し階層的に粗粒度タスク並列処理を行う。

Multi-Grain Parallelization using OpenMP
† Kazuhisa ISHIZAKA, † Motoki OBATA
† Koutarou TAKI, † Hironori KASAHARA
Dept. of Electrical, Electronics and Computer Engineering, School of Science and Engineering, Waseda Univ.

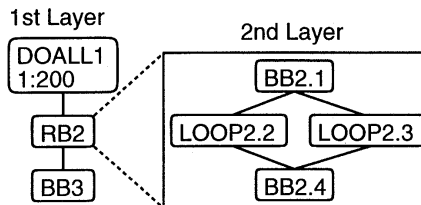


図 1: 階層型 MTG の例

粗粒度タスク並列処理では, MT はプロセッサエレメント (PE) をグループ化したプロセッサクラスタ (PC) に割り当てられる。PC 内では割り当てられた MT に対しては, 内部 PE でループ並列処理や階層的なマルチグレイン並列処理が行なわれる。

3 OpenMP を用いたマルチグレイン並列処理の実現

本 OpenMP を用いたマルチグレイン並列処理の実現では図 2 に示すように, OSCAR マルチグレインコンパイラにおけるバックエンド部に OpenMP を生成するルーチンを新たに生成した。

OSCAR コンパイラにより生成された OpenMP コードでは, まずプログラムの開始時に一度だけ PARALLEL SECTIONS ディレクティブを用いて, プロセッサ台数分のスレッドを生成する。このスレッドを PE として考えて, それをグループ化して PC を構成して階層型粗粒度タスク並列処理を行う。例えば, 6 スレッドを生成して 2PC で並列処理を行う場合, スレッド 0, 1, 2 を PC0 の PE0.0, 0.1, 0.2 と考え, スレッド 3, 4, 5 を PC1 の PE1.0, 1.1, 1.2 とする。

3.1 粗粒度タスク並列処理

粗粒度タスク並列処理時のマクロタスクの PC への割り当て方法には実行時に PC に MT を割り当てるダイナミックスケジューリング, コンパイル時に実行 PC を決めるスタティックスケジューリングがある。

ダイナミックスケジューリングの場合は, 各スレッド用のセクションには実行する可能性がある MT のコードをすべて記述しておき, それを実行時にスケジューラーに従って選択実行することによって, MT の並列実行を実現する。すなわち, 各スレッドの動作は, 基本的に図 3 に示すようになる。MT 割り当て待ちビジーウエイトループを実行し, スケジューラーによ

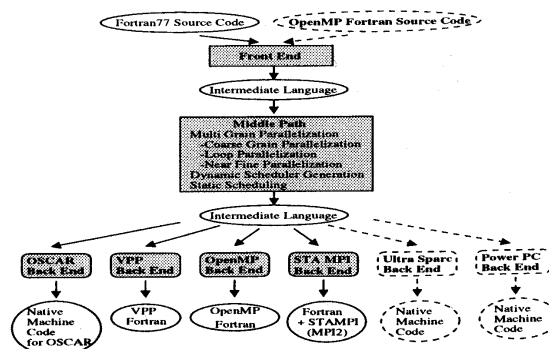
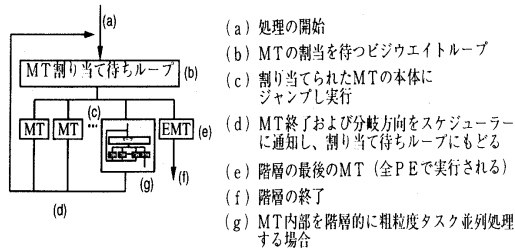


図 2: OSCAR マルチグレインコンパイラの構成



- (a) 処理の開始
- (b) MTの割当て待ちループ
- (c) 割り当てられたMTの本体にジャンプし実行
- (d) MT終了および分岐方向をスケジューラに通知し、割り当て待ちループにもどる
- (e) 階層の最後のMT (全PEで実行される)
- (f) 階層の終了
- (g) MT内部を階層的に粗粒度タスク並列処理する場合

図 3: ダイナミックスケジューリングのスレッドの動作

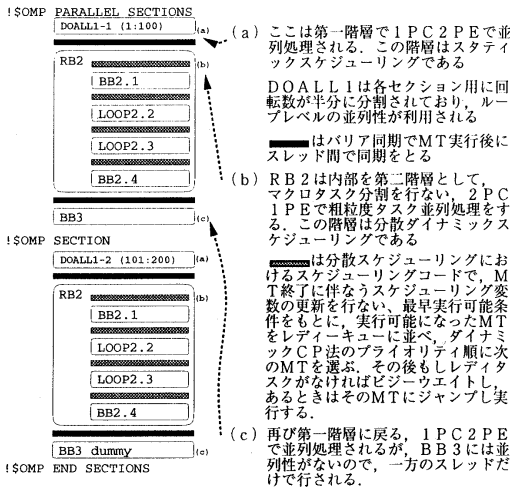


図 4: マルチグレイン並列処理

りMTが割り当てられるまで待つ。MTが割り当てられたらそのMTの実行コードにジャンプしMTを実行する。タスクの実行が終了したら、その終了あるいは分岐方向をスケジューラに通知し、再びMT割り当て待ちループに戻る。実行したMTがEMT (End MT) の場合、その階層の実行を終了する。

MTの内部でさらに階層的に下階層の粗粒度タスク並列処理を行うときは、図3の(g)の様、MTの内部で前述のMT割り当て待ちループとMT実行の処理を繰り返し、新階層でのEMTを実行したら、旧階層の処理にもどる。

図4に図1のMTGの場合のOpenMPを用いたマルチグレイン並列処理のイメージを示す。この場合第一階層は並列性がないので1PC2PEでスタティックスケジューリングを行うものとする。図4左図の一つ目のセクションのコードを実行するスレッド0をPC0のPE0.0とし、二つ目のセクションを実行するスレッド1をPC0のPE0.1とする。スタティックスケジューリングでは、ダイナミックスケジューリングとは異なり、コンパイラで各スレッドのセクションに記述したMTを順番に実行していく。第一階層のDOALL1は並列処理可能な200回転のループなので、100回転のループDOALL1-1とDOALL1-2に分割して、それぞれのセクションに記述して、PE間でループ並列性を利用した並列処理を行う(4の(a))。

第一階層でRB2の実行が始まるとRB2を第二階層として、この階層ではスレッド0をPC0のPE0.0、スレッド1をPC1のPE1.0として、2PC1PEで分散ダイナミックスケジューリングを行う。分散ダイナミックスケジューリングでは、各スレッドがMT実行終了後に、スケジューリング処理を行い、最

表 1: SMP 上での Benchmark による評価結果

Program	Speedup(4pe)		Speedup(8pe)	
	OSCAR	XL	OSCAR	XL
CG	4.16	3.94	8.36	6.36
tomcatv	2.95	1.35	4.33	1.43
swim	4.29	3.93	8.95	4.90
mgrid	3.70	3.14	6.76	4.20
hydro2d	4.41	3.49	8.10	4.69
ARC2D	2.23	1.28	3.32	1.29
平均	3.62	2.69	6.64	3.81

OSCAR: OSCAR FORTRAN Compiler
IBM XL: XL FORTRAN Compiler

早実行可能条件解析をもとにMTをPCに割り当て、次に実行すべきMTにジャンプする(4の(b)).

4 SR4300 上での評価

自動並列化コンパイラ OSCAR FORTRAN コンパイラに OpenMP FORTRAN を出力する OpenMP BackEnd を実装し、逐次型 FORTRAN から OpenMP FORTRAN を出力する並列化プリプロセッサとして用いて評価を行った。

評価プログラムはCG法による行列求解プログラムと Spec95 Benchmark の tomcatv, swim, mgrid, hydro2d, Perfect Benchmarks の ARC2D を用いた。OSCAR コンパイラから出力された OpenMP FORTRAN は IBM XL FORTRAN コンパイラ Ver.5.5.1 を用いて 8 プロセッサ SMP マシン IBM RS/6000 SP 604e High Node 上で実行した。OSCAR FORTRAN Compiler で並列化した場合と、XL FORTRAN コンパイラで自動並列化した場合の評価結果を表1に示す。表中の値は逐次実行時間に対する速度向上率である。

IBM XL FORTRAN コンパイラでは 4PE で 1PE に比べ平均 2.69 倍の速度向上、8PE で平均 3.81 倍の速度向上であったのに対し、OSCAR コンパイラの場合は 4PE で平均 3.62 倍、8PE で 6.64 倍の速度向上と、平均 1.87 倍性能を向上させ、tomcatv で約 3 倍の速度向上率を示す等、IMP SMP マシン上では OSCAR コンパイラの方が XL FORTRAN コンパイラよりも優れた速度向上を示すことが確認された。

5 まとめ

本稿では OpenMP を用いたマルチグレイン並列処理の実現方法を提案し、IBM RS/6000 SP 604e High Node 上で Benchmark プログラムを用いて評価を行った結果について述べた。評価の結果 OSCAR コンパイラは、プロセッサ数の増加に対してスケラブルな速度向上率の伸びを示し、IBM XL FORTRAN Compiler の自動並列化に対して最大で約 3 倍の速度向上率という結果を得た。

本研究の一部は、通産省次世代情報処理基盤技術開発環境事業並列分散分野マルチプロセッサコンピューティング領域研究の一環として行われた。

参考文献

- [1] 笠原: “並列処理技術”, コロナ社 (June 1991).
- [2] Michel Wolfe: “HIGH PERFORMANCE COMPIERS FOR PRALLEL COMPUTING”, The Addison-Wesley Publishing Company
- [3] Utpal Banerjee: “Loop Transformations for Restructuring Compilers The Foundations”, Kluwer Academic Publishers
- [4] <http://www.openmp.org/>
- [5] Lenald Dagum, Ramesh Menon: “OpenMP: An Industry-Standard API for Shared Memory Programming”, IEEE Computational Science & Engineering (1998)
- [6] 岡本, 合田, 宮沢, 本多, 笠原: OSCAR マルチグレインコンパイラに置け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 (1995)
- [7] 笠原, 合田, 吉田, 岡本, 本多: Fortran マクロデータフロー処理のマクロタスク生成, 信学論 (1992)
- [8] 吉田明正, 越塚健一, 岡本雅巳, 笠原博徳: 階層型粗粒度並列処理における同一階層内ループ間データローカライゼーション手法, 情報処理学会論文誌 (1999)