

## 並列化コンパイラにおける組合せ並列化技術

和田 清美<sup>\*1\*</sup> 佐藤 真琴<sup>\*1\*</sup> 飯塚 孝好<sup>\*1</sup>

<sup>\*1</sup> (株) 日立製作所システム開発研究所 <sup>\*2</sup> アドバンスト並列化コンパイラ研究体

SMP 上でプログラムを高速実行するために、コンパイラはループに対して様々なループ変換を適用するが、静的なコンパイラ解析情報だけでは、最適なループ変換やループ展開数、ループタイルサイズ等の決定は困難である。このため、実行時情報に基づいて最適な指示文を決定するオプションチューニングツールの開発を行った。本ツールの特徴技術は、(1) 1回の試行では、多重ループ毎に同じ指示文の組を適用し、多重ループ毎に実行時間を計測し、全試行後、多重ループ毎に最高速となる指示文の組を適用すること、及び(2) 一部実施要因計画法を用いて、多重ループの指示文の組合せを指定することである。これより、試行回数を少なくして、多重ループ全体で最適な指示文の組合せを決定できた。

### Feedback-directed Selection Technique of Compiler Directives for Parallelized Compiler

Kiyomi Wada<sup>\*1\*</sup>, Makoto Sato<sup>\*1\*</sup>, Takayoshi Iitsuka<sup>\*1</sup>

<sup>\*1</sup> Systems Development Laboratory, Hitachi, Ltd.

<sup>\*2</sup> Advanced Parallelizing Compiler Project

Although a compiler applies various loop conversions to a loop in order to perform the program on SMP at high-speed, the determination of the optimal loop conversion, the number of loop unroll, and the size of loop tiling is difficult only by static compiler analysis. For this reason, the option tuning tool which determines the compiler directive based on information of execution time was developed. This tool has two features. One is applying the combination of the same compiler directives for nested loops and measuring execution time of each nested loop by one trial. And after all trials, the combination of compiler directives which perform at the maximum speed for each nested loop was applied. The other is specifying the combination of the compiler directives for nested loop using the Fractional Factorial Designs. Consequently we could reduce the number of trial and determine the optimal combination for nested loop.

#### 1. はじめに

コンパイラは、ループに対して様々なループ変換を適用する。しかし、静的な情報だけでは、最適なループ変換の選択やループ展開数やループタイルサイズの決定等を行なうことは困難である。このため、実行時情報に基づいて、最適なコンパイルオプションや指示文を決定するオプションチューニングが重要である。従来の、ユーザプログラム向けオプションチューニングツールは、プログラム全体に指定するコンパイルオプションを最適化するものや、ループ毎に指定する指示文を最適化するものがあつた。しかし、前者はループ毎のきめ細かな最適化が指定できない、後者は多重ループ全体にとって最適な指示文の組合せを考慮していないという問題点があつた。そこで、本稿では、多重ループ

に対しても最適な並列化・最適化指示文の組合せを短時間で検出可能なオプションチューニングツールの実現を目的とする、組合せ並列化技術について述べる。これまで、従来技術の課題を抽出し、これらの課題に基づいて開発機能の検討を行い、ツールの設計・開発を進めてきた。本ツールは、次の2つの特徴をもつ。1つめは、1回の試行では、複数の多重ループに対して、同じ並列化・最適化指示文の組を適用し、多重ループ毎に実行時間を計測する。そして、全試行終了後、多重ループ毎に、最高速となる指示文の組を適用することである。2つめは、多重ループにどのような指示文の組合せを指定するのかを、一部実施要因計画法を用いて決定することである。以上により、試行回数を削減して、多重ループ毎に、ほぼ最適な指示文の組合せを決

定できるようになった。

以下の構成は次のようになる。2章では機能概要、3章では処理概要、4章では評価結果について述べ、5章で全体のまとめを述べる。

尚、本研究は、NEDOの国家プロジェクト「アドバンスド並列化コンパイラ(Advanced Parallelizing Compiler Project:APC)の中で実施した。

## 2. 機能概要

従来技術から課題を見つけ、この課題に基づいて、開発機能の検討を行った。

表1は、従来技術とAPCで開発中のチューニングツールに対する比較表である。表1において、ATLAS、ILIB、TEA Expertは数値計算ライブラリ向けのチューニングツールである。APCでは、一般プログラム向けにチューニングすることを目的にしているため、直接比較対象となるのはADAPTとIntelである。このうち、ADAPTは線型探索で、Intelは一部実施要因計画を用いる。線型探索とは、最適コンパイルオプションの組合せを、最初にオプションを全てOFFとし、1つずつオプションをONにして実行したとき、前回の性能を上回ればONを採用し、そうでなければOFFを採用する探索方法である。一部実施要因計画とは、多数の因子が同時に作用することによる効果は小さいという考えに基づき、全数探索に比べて実験回数を大幅に減らす方法で、実際問題では非常に有用な計画として広く用いられている。試行するコンパイルオプションの数を $n$ とすると、試行回数は、ADAPTでは $O(n)$ となり、Intelでは $O(2^{n/2})$ や $O(2^{n/4})$ となる。ADAPTは試行回数は少ないが、コンパイルオプション同士の相乗効果を見逃しているため、精度上問題がある。一方、Intelは

相乗効果も考慮可能で、試行時間と精度を調整可能であるが、プログラム全体の最適化のみで、ループ毎のきめ細かな最適化ができない。

また、ADAPTは個別ループを対象にチューニングするが、各ループに対し、線形探索で変換種別を決定しているため、多重ループ全体では最適な変換の組合せがみつけれない。

そこで、APCでは、試行回数と精度の両面からみて、ADAPTとIntelに対して優れた技術を開発することを目標とし、Intelが採用した一部実施要因計画を、多重ループ全体の最適化の組合せに適用した。

以下に試行回数を削減して、精度をあげるための方法を示す。

### (a) 対象ループ削減方法

プロファイル情報を利用して、チューニング対象ループを絞る。

### (b) ループ組合せ削減方法

各試行では、全てのチューニング対象の多重ループに対して、同じ最適化指示文の組合せを適用し、多重ループ毎に実行時間を計測する。全試行終了後、多重ループ毎に最高速となる最適化指示文の組合せを適用する。

### (c) 最適化方法削減方法

最も効果が期待でき、実績がある以下の4最適化方法に限定する。

- ・ループ交換順序
- ・ループ並列化の適否
- ・ループタイルサイズ
- ・ループ展開数

### (d) 最適化組合せ削減方法

一部実施要因計画法を適用する。

表1. 比較表

開発元 (ツール名)	最適化対象 (最適化指示方法)	実行方法	課題	参考文献
テネシー大 (ATLAS)	多重ループ全体 (指示文)	全探索	なし	[1]
東大 (ILIB)			チューニングコードの コーディング	[2]
RWCP (TEA Expert)			指示文と計測区間の挿入 パラメータ空間の設定	[3]
Purdue大 (ADAPT)	プログラム全体 (コンパイルオプション)	線型探索	オプション間の相乗効果を 十分考慮せず	[4]
	個別ループ (指示文)	各ループに対し、 線形探索で、 変換種別を決定	多重ループ全体で 最適な変換の組合せを 見つけられず	
Intel	プログラム全体 (コンパイルオプション)	一部実施要因計画	ループ毎に最適化できず	[5]
APC	多重ループ全体 (指示文)	一部実施要因計画		

### 3. 処理概要

図1に組合せ並列化システムの全体構成を示す。図1において、プログラムチューニングを2段階に分けて行うため、組合せ並列化システムは、ループ並列化・構造変換決定部とループ展開数決定部から構成される。これは、多重ループネスト全体に対する最適な組合せを見つけるために、ループ並列化、ループ交換、ブロック化、ループ展開の組合せを同時に適用すると、ブロック化がループの構成を変えてしまうため、展開対象となるループが変わる可能性があるからである。例えば、2重ループの各ループをブロック化すると、4重ループになる。この4重ループの新たに

生成された内側ループをループ展開したい場合、元のループに現れないために、チューニング対象ループの指定ができない。

図2は、ループ並列化・構造変換決定部の構成である。ループ展開数決定部の構成もほぼ同様であり、入力がループ並列化・構造変換決定処理で得られた OpenMP プログラムである点が異なる。

図2で用いるプログラムは、次の4つ、即ち、手続き間自動並列化コンパイラ (WPP:Whole Program Parallelizer)、組合せ並列化支援ツール、実験計画作成ツール (JMP<sup>1</sup>)、入出力用ユーザインタフェース (I/F) である。

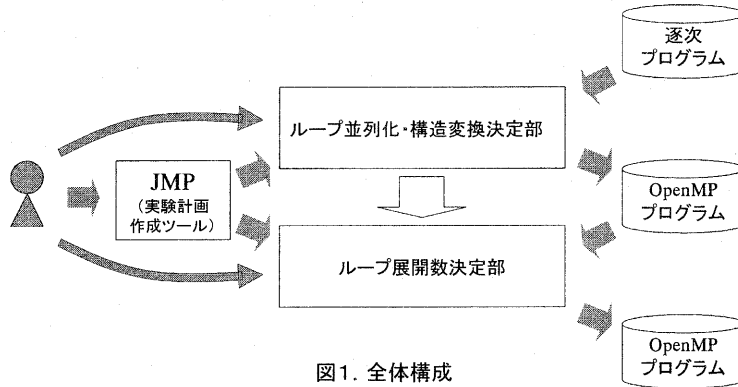


図1. 全体構成

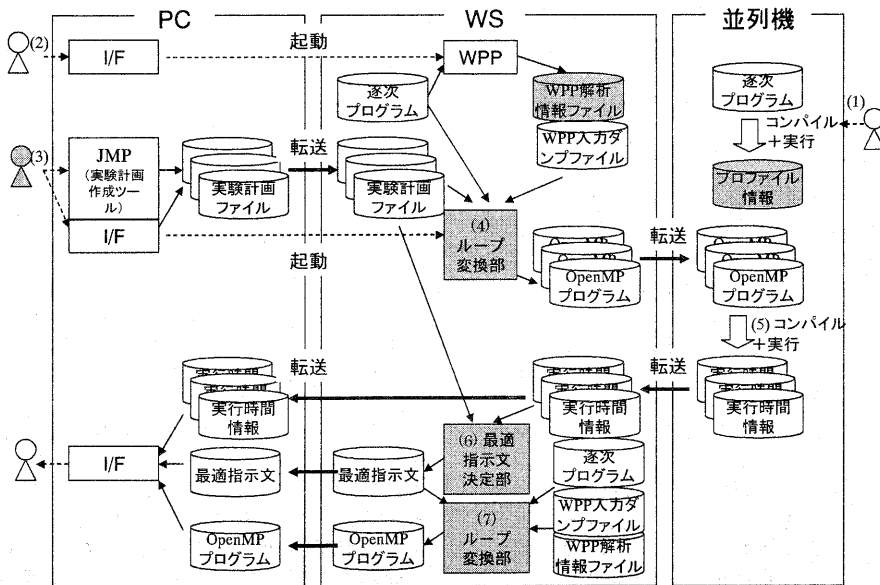


図2. ループ並列化・構造変換決定部

<sup>1</sup> JMP は米国 SAS Institute Inc. の商標です。

次に、それぞれの基本的な役割について説明する。

(a) WPP

逐次プログラムまたは指示文付きプログラムを入力し、ループ並列化・最適化を行い、解析情報ファイルと入力ダンプファイル（コンパイラ解析情報のループ番号と、ループ開始・終了行番号とを対応づける）及び OpenMP プログラムを出力する。

(b) 組合せ並列化支援ツール

本ツールは、ループ変換部と最適指示文決定部からなる。ループ変換部は、チューニング対象ループとループ並列化・最適化指示文が格納された実験計画ファイルを入力し、プログラム中に実行時間取得コードと指示文を挿入し、WPP を起動して、OpenMP プログラムを出力する。最適化指示文決定部は、指示文を変化させたときのループ毎の実行時間が格納された実行時間情報ファイルを入力して、ループ毎の最高速の指示文をみつけ、最適なループと指示文の組合せを出力する。

(c) JMP

試行すべき指示文を因子に、変化させるパラメータを各水準として、プログラムに適用したい指示文とパラメータの範囲を指定すると、実験計画法の1つであるカスタム計画に従って、指示文とパラメータの組合せを生成する。

(d) ユーザインターフェース

ユーザが、WPP を起動するためと、チューニング対象ループと指示文およびパラメータの範囲の指定を入力するため、及び最適なループと指示文の組合せを出力するために用いる。

次にループ並列化・変換決定部（図2）の処理の流れを説明する。

- (1) ユーザは、予め評価対象の共有メモリ型並列計算機（SMP）上でプログラムを実行してプロファイル情報を取得する。
- (2) ユーザは、WPP を起動して、解析情報ファイルと入力ダンプファイルを取得する。
- (3) ユーザは、プロファイル情報の最外側ループ毎の実行時間からホットスポットループを見つかる。ここで見つけたホットスポットループネストに対して、コンパイラ解析情報（適用した最適化方法、即ち並列化の適否、ループ変換方法）と、プロファイル情報（ループ長、呼び出し回数）から最適化指示文の組合せや、パラメータの設定範囲について検討する。この結果から、ユーザは JMP の実験計画作成手順に従って、多重ループに適用し

たい指示文と、その指示文に対するパラメータの範囲を入力し、JMP を実行する。JMP は、ループ毎に試行すべき指示文とそのパラメータの組合せからなる実験計画ファイルを出力する。

- (4) ループ変換部は、実験計画ファイルと、WPP 入力ダンプファイルとコンパイラ解析情報から、逐次プログラムに指示文と計測コードを挿入したプログラムを生成し、本プログラムを WPP でコンパイルして、指示文に従ってループ変換と OpenMP コード生成を行い、OpenMP プログラムを出力する。
- (5) SMP は、OpenMP プログラムを入力し、コンパイル及び実行し、ループ毎の実行時間を格納して実行時間情報ファイルを出力する。

以上の(4)、(5)の処理は、組合せ数の回数だけ繰り返す。すべての実行が終了したら(6)へ移る。

- (6) 最適化指示文決定部は、ループ毎の実行時間を入力して、ループ毎に最高速の指示文の組合せを求め、その結果を最適指示文ファイルに出力し、ユーザインタフェースを介して、各ループの最適な指示文を表示する。
- (7) ループ変換部は、WPP 入力ダンプファイルとコンパイラ解析情報から、逐次プログラムに最適な指示文の組合せを挿入したプログラムを生成し、本プログラムを WPP でコンパイルして、指示文に従ってループ変換と OpenMP コード生成を行い、OpenMP プログラムを出力する。

#### 4. 評価結果

SPECfp95 の mgrid に対して、日立 SMP マシン SR8000 (1 ノード、8PE) を用いて評価した。

(1) ホットスポットの検出

サブルーチン RESID の D0 600 が実行時間の 40.1%、サブルーチン PSINV の D0 600 が実行時間の 21.4% を占めていた。そこで、この2つのループをチューニング対象とした。

(2) チューニング方法

上記2つのループは、参照する配列名が異なるが、同じプログラムパターンであるため、同じ最適化方法を適用した。これらのループネストは密多重の3重ループで、最外側ループは自動並列化されている。各ループはループ展開されていないため、ループ展開を試みた。

(3) 実験計画法の適用

ここでは、3重ループの各ループに対して、

ループ展開数を1、2、3と変化させる場合に対するカスタム計画の適用を説明する。各ループの展開指示文を因子とし、展開数を水準と考える。カスタム計画では、何因子間の交互作用を考慮するかをユーザが選択できる。表2-1は、1因子のみの効果をみるための実験表である。表2-2は2因子間の交互作用の効果もみるための実験表である。各因子水準が均等に出現するようにして一定数の実験組数だけ作成している。3因子間の交互作用の効果をみる実験数は、3因子全水準の全組合せ数の27組となる。

(4) 試行結果

表2-1より、1因子のみの効果をみる実験では、実験#4が最高速となる。表2-2より、2因子間の効果をみる実験では、実験#12が最高速となる。3因子間の効果をみる実験では、1因子のみの実験#4と同じ展開数の組合せが最高速となった。また、チューニング前のループ展開なし(コンパイラによる自動最適化)の場合は17.7秒であったので、チューニングしたことにより、実行時間が64%まで短縮できた。

5. おわりに

オプションチューニングに関する従来技術の課題の抽出を行い、組合せ並列化ツールの開発機能の検討、ツールの設計と開発を進めた。本ツールは、次の2つの特徴をもつ。

- ・1回の試行で、多重ループ毎に同じ並列化・最

適化指示文の組を適用し、多重ループ毎の実行時間を計測し、全試行後、多重ループ毎に最高速となる指示文の組合せを適用する。

- ・多重ループにどのような指示文の組合せを指定するのかを、一部実施要因計画法を用いて決定する。

以上により、試行回数を削減して、多重ループ毎に、最適な指示文の組合せを決定できるようになった。

参考文献

[1] R.C.Whaley, A.Petite and J.J.Dongara, "Automated Empirical Optimization of Software and the ATLAS Project" Sep.2000, <http://math-atlas.sourceforge.net>  
 [2] 片岡, 黒田, 工藤, 金田(東大), "PC クラスタにおける並列数値計算ライブラリILIBの性能評価", HPC研究会 87-4, 2001.7月  
 [3] 佐藤, 建部, 関口, 朴(RWC), "自動適応並列プログラム性能最適化ツール TEA Expert", HPC研究会 72-3, 1998.8月  
 [4] Michael J.Voss and Rudolf Eigenmann (Purdue 大),"High-Level Adaptive Optimization with ADAPT", PpoPP'01  
 [5] Kingsum Chow and Youfeng Wu(Intel), "Feedback-Directed Selection and Characterization of Compiler Optimization", Second Workshop on Feedback-Directed Optimization'99

表2-1. 実験表(1因子)

実験#	ループ展開数			時間 [秒]
	外側	中側	内側	
1	3	3	3	18.1
2	2	3	1	12.8
3	3	2	2	15.3
4	2	1	3	11.3
5	3	1	1	13.5
6	1	1	2	13.9
7	1	2	3	12.3

表2-2. 実験表(2因子)

実験#	ループ展開数			時間 [秒]
	外側	中側	内側	
1	1	1	3	12.3
2	3	3	3	18.1
3	3	2	2	15.3
4	3	2	1	12.8
5	1	2	2	13
6	3	3	2	15.2
7	1	3	3	16.9
8	1	3	1	13.3
9	1	1	2	13.9
10	2	2	3	16
11	3	1	1	13.5
12	2	1	2	11.9
13	1	2	3	12.3
14	2	2	1	13.6
15	2	3	2	15.4
16	2	3	3	16.8
17	3	3	1	12.9
18	2	1	1	12.7
19	3	1	3	14.2

**付録**

組合せ並列化表示・入力ツールのインターフェイスを示す。

図3は、全体画面である。本画面から、構造変換／ループ展開の切り替え、WPP 実行、ループ変換処理、結果表示を行う。

図4は、計画ファイル指定画面である。ループ変換を実行するとき、本画面から計画ファイルを指定する。

図5は、レイヤ画面といて、計測点（計測区間開始点）を指定するための画面である。

図6は、実行時間表示画面である。画面の上半分が実行時間を表示しており、行方向が測定点、列方向が試行番号である。下半分は測定点に対するソースプログラムである。

図7は、最適指示文表示画面である。

手続き名、ループ開始行番号、ループネスト数に対する深さごとに実行時間と、最適化指示文が表示される。

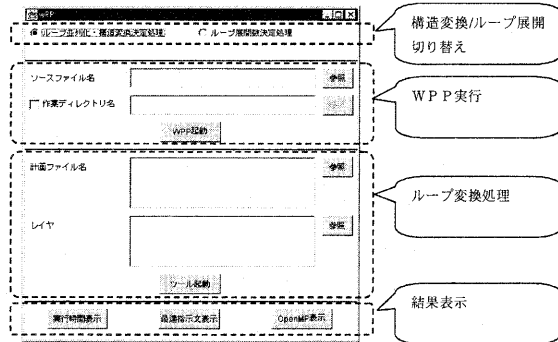


図3. 全体画面

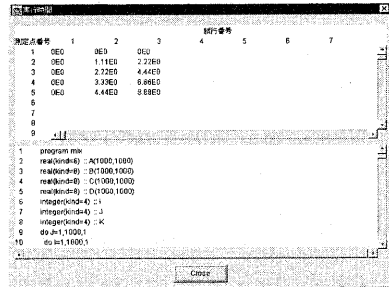


図6. 実行時間表示画面

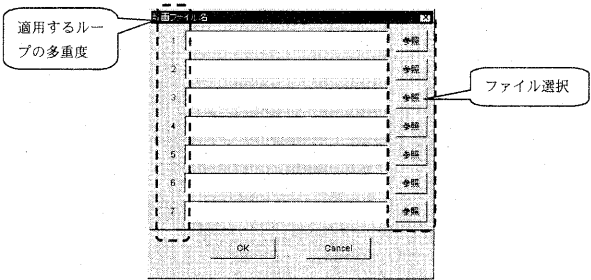


図4. 計画ファイル画面

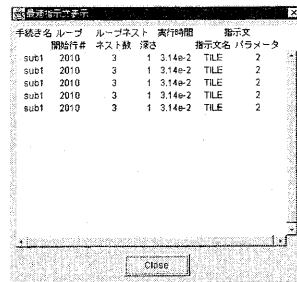


図7. 最適指示文表示画面

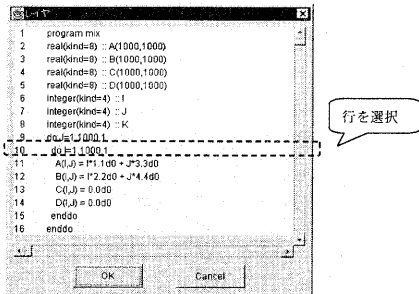


図5. レイヤ画面