

並列化支援システム「Parassist」の試作  
 — 並列性検査方法 —

5 D - 9

橋本 博幸\* 菊池 純男\* 佐藤真琴\* 佐々木祥吾\*\*

(株)日立製作所中央研究所\* 日立東北ソフトウェア(株)\*\*

1. はじめに

既存のプログラムの並列化を支援するシステムParassist[1](Parallelization Assist System)における並列性検査方法について述べる。並列化のためのチューニング作業を行う上で、どの変数・配列のメモリアクセスが並列性を阻害しているかを探索するのは困難が予想される。そこで、並列性検査ツールはプログラム実行時に得られたメモリ参照情報をもとに、データ参照順に着目した並列性検査と検査過程を参照状況の可視化[2][3]アニメーションに反映することを行い、ユーザが並列化を図っていく上でのひとつの有用な手段を提供する。本発表では、並列性検査ツールの機能と構成の概要について報告する。

2. システム構成と機能概要

図1に並列性検査ツールのシステム構成を示す。ユーザインタフェースはOSF/Motifを採用した。並列性検査ツールは、並列性検査範囲指定部、対象変数指定部、トレース収集制御部、並列性検査・アクセス状況表示部の4つで構成される。

並列性検査ツール利用時の処理手順を以下に示す。

(1) 並列性検査範囲、対象変数指定処理

並列プログラム簡略表示[4]ウィンドウ上で、検査範囲とするDOループを指定する。その後、検査候補変数名表示[4]ウィンドウ上で、検査の対象となる変数・配列名を指定する。

(2) トレース収集制御処理

トレース収集制御部は、検査範囲、対象変数・配列名の格納されたファイルとソース解析情報ファイルとを入力し、ソースプログラム中に、メモリアクセス情報をトレースデータとして収集するためのディレクティブ文(コンパイラへの

指示文)を挿入する。

(3) コンパイル処理

FORTRANコンパイラは、ソースプログラムを入力し、トレースデータ出力用のコードを挿入した逐次オブジェクトを生成する。

(4) リンク処理

リンカは、逐次オブジェクトとトレースデータ出力ライブラリとのリンクを行い、ロードモジュールを生成する。

(5) 実行処理

ロードモジュールを実行し、メモリ参照情報をトレースデータとして収集する。

(6) アクセス状況表示処理

(並列性検査、参照状況の可視化アニメーション)

検査範囲、対象変数・配列名の格納されたファイルとソースプログラムとメモリ参照情報とを入力し、アクセス状況と並列性検査結果をウィンドウ上に表示する。

3. 並列性検査方法

並列性検査では、同一の変数・配列要素に対し、並列実行される可能性のある複数のプロセスから定義・引用が発生した場合、DOループに対する対象変数・配列の並列実行は不可であると判定する。この判定を行うために、プロセスグループ番号とプロセス番号という概念を設ける。

プロセスグループとは、あるプロセスからforkされた子プロセスの集合とする。プロセスグループ番号とは、このプロセスグループをプログラム全体にわたって区別するためにつける識別番号である。プログラムを実行する際に最初に唯一存在しているプロセスは、プロセスグループ番号0番のプロセスグループに属していることにする。

プロセス番号とは、プログラム全体でプロセスの区別ができるようにするために各プロセスグループ内で各プロセスにつける識別番号である。

検査には実行時に収集したメモリ参照情報を用いる。メモリ参照情報は、以下の情報を含む。

- ・ fork時のプロセス情報
- ・ forkを行ったプロセスのプロセスグループ番号とプロセス番号、forkを行ったときに生成したプロセスグループ番号
- ・ 対象変数・配列アクセス時のアクセス情報
- ・ グループ番号、プロセス番号、定義/引用の区別

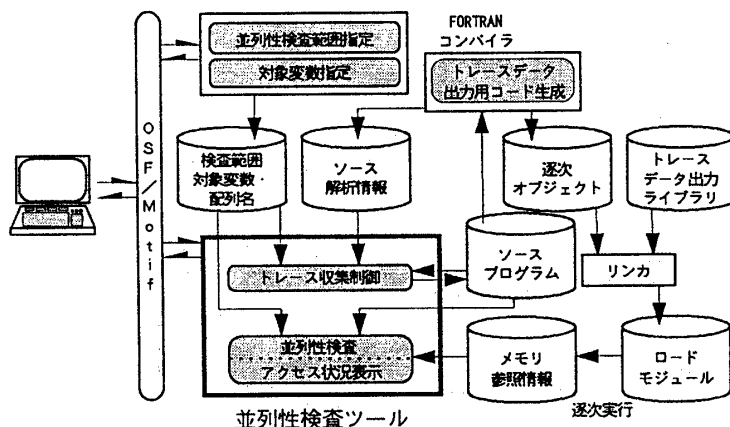


図1 並列性検査ツールのシステム構成

Prototyping of Parallelization Assist System "Parassist"

-Checking Method of Parallelization-

Hirouyuki HASHIMOTO, Sumio KIKUCHI, Makoto SATO, Shogo SASAKI

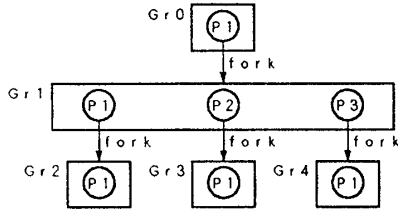
\*HITACHI, Ltd. ,

\*\*HITACHI TOHOKU SOFTWARE, Ltd.

次に、並列性検査の処理概要を述べる。

(1) プロセスグループグラフ作成

fork時のプロセス情報から下図に示すプロセスグループグラフを作る。



(2) プロセス集合の作成

プロセスグループグラフを利用し、並列実行される可能性のないプロセスの集合を作成する。1つの集合には、次の条件を満たすプロセスを要素として登録する。

[条件] : プロセスグループグラフ上の末端のプロセスに対し、Gr0(P1)のプロセスからグラフ上をたどれるパス上のすべてのプロセス。

(3) 並列性検査

トレースデータのメモリアクセス情報を読み込み、対応する変数・配列要素のメモリアクセス履歴情報に追加登録する。追加登録の際、既に登録してある履歴情報との間で並列性検査を行う。検査は、プロセス集合を利用し、新しく読み込んだアクセス情報と既に登録してあるアクセス情報とが同じプロセス集合内に存在するかを調べる。存在する場合は、この2つのプロセスが並列実行可能であると判定する。この操作をトレースデータがなくなるまで行う。

4. アクセス状況表示方法

対象変数・配列要素に対するアクセス状況は、次の5つが考えられる。

- (a) アクセスが一度も行われていない。
- (b) 引用のみが行われている。
- (c) 並列実行される可能性のない複数プロセスによる定義のみが行われている。
- (d) 並列実行される可能性のない複数プロセスによる定義と引用が行われている。
- (e) 並列実行される可能性のある複数プロセスによる定義と引用、定義と定義が行われている。

このメモリアクセス状況の決定は、新たなトレースデータを読み込む直前のメモリアクセス状況と、読み込んだトレースデ

ータのアクセス情報と、その要素に対する並列性検査結果とで行う。メモリアクセス状況の決定方法を表1に示す。

表1 メモリアクセス状況の決定方法

メモリアクセス状況	アクセス情報	定義	引用
(a) アクセスなし	(c)	(b)	
(b) 引用のみ	(d)	(b)	
(c) 並列実行される可能性のないプロセスによる定義のみ	(a)	(d)	
(d) 並列実行される可能性のないプロセスによる定義と引用	(d)	(d)	
(e) 並列実行される可能性のあるプロセスによる定義と引用、定義と定義	(e)	(e)	

表中の記号は、前記の (a) ~ (e) に対応する。ここで、新アクセス状況に [ ] が掛けられている場合は、新たなトレースデータと変数・配列要素のすべてのメモリアクセス履歴情報との間で並列性検査を行い、並列実行不可と判定された場合にはアクセス状況を (e) にする。

5. 検査結果出力例

図2に並列性検査ツールで行ったアクセス状況、並列性検査結果の出力例を示す。

6. おわりに

FORTRANプログラムのDOループで指定された変数・配列に関するメモリアクセス状況を視覚化すると共に、並列実行可能か否かを判定する並列性検査ツールを試作した。実行時に得られたメモリ参照情報が、並列化のためのチューニング作業などにも役立つことが分かった。

参考文献

- [1] 菊池ほか: "並列化支援システム「Parassist」の試作-機能と構成-", 情報処理学会第44回全国大会(1992)
- [2] O. Brewer, et al.: Tools to aid in the analysis of memory access patterns for FORTRAN programs: Parallel Computing 9(1988/89)25-35
- [3] M. H. Brown: Exploring Algorithms Using Blasa-II, IEEE Computer, Vol. 21, No. 5, pp. 14-36, May 1988
- [4] 佐藤ほか: "並列化支援システム「Parassist」の試作-不正並列化検出方法-", 情報処理学会第44回全国大会(1992)

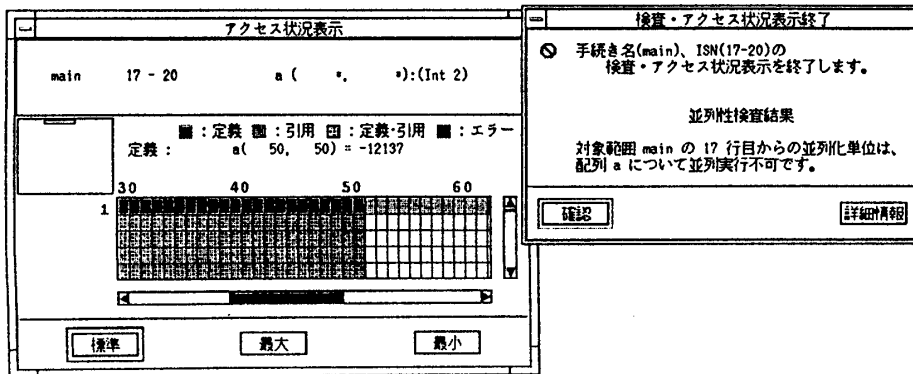


図2 アクセス状況表示と検査結果