

並列化支援システム「Parassist」の試作

5D-4

— 機能と構成 —

菊池純男 飯塚孝好 岩澤京子 佐藤真琴

(株)日立製作所 中央研究所

1. はじめに

共有メモリ型並列計算機に向けた対話型並列化プログラミング支援システムParassist(Parallelization Assist System)について述べる。本システムはUNIXワークステーション環境で稼働する統合型支援システムを指向したものであり、基本的には逐次FORTRANプログラムの並列化作業の支援を目的とした実験評価用システムである。

並列処理によって逐次処理を遥に凌ぐ高速性能を実現しようとする試みは古くからあるが、近年やっと実用化の兆しが見えてきたところである。並列処理が本格的に普及するためには、並列記述言語の標準化とともに、FORTRANのように過去のソフトウェア資産が多い場合には従来プログラムを自動並列化するコンパイラおよび並列化作業を支援するツール[1][2]が重要な鍵を握る。

本発表では、並列化を支援するParassistのコンセプトとこの上に実現されている幾つかのツール群の機能と構成について述べる。

2. システムコンセプト

過去の膨大なソフトウェア資産の有効利用を図り、かつユーザの並列化作業の負担を極力軽減するには、これまでの自動ベクトル化技術同様、並列処理においても自動並列化は極めて重要な技術である。

一方で、並列処理は並列制御のオーバーヘッドが大きいため粒度(並列処理単位の大きさ)を大きくしないとプロセッサ台数に見合った性能が得られない。粒度を大きくするためには多重ループ、さらに手続きにまたがった並列化を行うことが必要となる。このためには手続き間にまたがった高度なデータフロー解析技術[3]が必須である。しかるに、プログラムワイドにデータ依存を完全に解析することは技術的に極めて困難である。

さらに、このような技術的な問題以前に致命的な点はソースプログラムという静的な情報からだけでは確定しないデータ依存関係、即ち実行時に初めて確定するようなものが存在することである。この問題はベクトル化についても言えることであるが、プログラムワイドなデータ依存情報を必要とする並列化においては、静的解析技術に基づく自動化技術とともにこれらの限界への対処手段を持つことが極めて重要である。しかも、これらが有

効な手段となるためには、コンパイラとの密な連携が取れていなければならない。

また、プログラムのデバッグでは如何に早く不良箇所を絞り込むかが重要である。とくに並列プログラムにおいては、従来型のシンボリックデバッガを用いて複数プロセスを逐一追跡する手段だけではプログラムの挙動を調べ、不良箇所を絞り込むには大変な労力が必要である。

そこで、Parassistでは以下の点に主眼をおいている。

- 大きな性能向上を得るために
 - 手続き間並列化技術
- 静的解析の限界に対処するために
 - ユーザとの対話環境
 - 実行プロファイル情報の利用
- 効率的なデバッグのために
 - ユーザ指定の並列化バグ箇所の早期検出
- プログラムの挙動理解を支援するために
 - プログラム挙動のマクロ表示

3. システム構成

Fig 1にParassistの構成を示す。各ツールの機能については次章で述べる。

本システムは以下の特徴を持つ。

- UNIXワークステーション環境で稼働
- 対話型の並列化支援環境の提供
- コンパイラ機能と支援機能との密な連携
- OSF/Motif標準ユーザインタフェースの採用

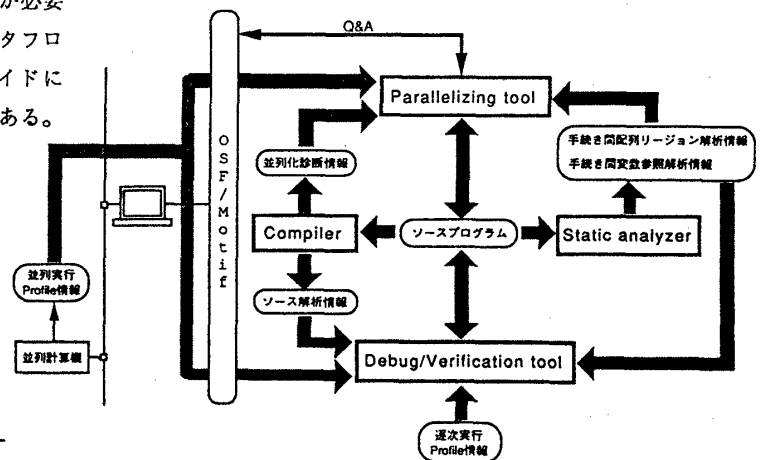


Fig 1 Parassistの構成

Prototyping of Parallelization Assist System "Parassist"

- System Overview -

KIKUCHI Sumio IITSUKA Takayoshi IWASAWA Kyooko SATOH Makoto
Central Research Lab., HITACHI Ltd.

4. 機能

本章ではParassistを構成する4つの主要なコンポーネントの機能概要を述べる。

4.1 コンパイラ

- (1) 多重DOループ自動並列化
- (2) ユーザ指示文・並列記述文に従った並列化
 - ・並列ループ文
 - ・Parallel Section文
 - ・同期排他制御文
- (3) 並列プロセス制御の自動化
 - ・プロセスの生成、消滅
 - ・プロセス起動、待ち
- (4) 並列化診断情報出力
 - ・並列化ループに関する情報
 - ・並列化不能要因
 などを出力する。

4.2 Static analyzer

制御フローを考慮した手続き間にまたがった変数、配列のデータフロー解析を行う。配列については参照範囲(リージョン)を解析する。解析手順としては、まず変数について手続き内、手続き間解析を行い[4]、ついでこの結果を利用して配列の手続き内、さらに手続き間解析を行う。

(1) 手続き内解析

手続き内の全変数・全配列が解析対象である。各手続き内の手続き呼び出し点間の制御フローを要約したサマリグラフを作成し、各ノード間のパス上での各種データフロー情報を解析する。これをすべての手続きに対して行う。

(2) 手続き間解析

コモン、引数を対象に、手続き内解析結果を用いて手続き間にまたがるデータフローを解析する。

4.3 Parallelizing tool

ユーザが選択したDOループの並列性を解析し、並列化判定を行う。並列化可能ならば、コンパイラが並列化するのに必要な情報をコンパイラへの指示としてソースプログラムに挿入する。

(1) 並列実行負荷表示

各手続き、各ループの占める実行時間を表示する。

(2) 並列性解析

ユーザが選択したDOループについて並列性を解析する。ループ内に出現する手続きにまたがるデータ(コモン、引数)についてはStatic analyzerの解析結果を取り込み、並列性を解析する。

(3) 並列化判定

並列性解析の結果から並列化できるか否か、さらに粒度を考慮し並列化すべきか否かを判定する。粒度不足と判断できる場合には、並列化可能でも並列化しないと判定する。

静的な解析結果だけでは並列化判定できない場合には、並列化条件を作成し、ユーザとの対話処理にて解消を図る。

(4) ユーザとの対話

並列化判定する上で必要な情報(データ依存、ループ長など)のうち、不明なものについてユーザとのQ&Aを行い、これを解消する。その結果、並列化条件を満たせば並列化可能と判定する。

(5) ソースプログラム変換

並列化可能と判定された場合、コンパイラの並列化診断情報に照らして、コンパイラが並列化できるための情報をソースプログラムに挿入する。

4.4 Debug/Verification tool

(1) 並列構造表示

ソースプログラムから並列指示文、並列記述文およびDOループ文を抽出し表示する。この表示は(2)以降の機能を利用する際の共通インタフェースである。

(2) 不正並列化指示文の検出

逐次実行結果との値比較による不正なユーザ指示並列化文を一括検出する。

(3) 並列性検査

逐次実行時のメモリ参照情報を利用した並列化阻害要因の検出とメモリアクセス状況をアニメーション表示する。複数プロセスの同一メモリ領域への参照履歴情報を用いて並列性を検査する。

(4) 並列実行プロセス表示

並列実行時のプロセスに関するProfile情報を解析し、プロセスの実行状況を表示する。

- ・プロセスの実行、待ち状態表示
- ・POST、WAITの対応表示
- ・並列実行がネストした時、プロセスをグルーピングしマクロ表示

5. おわりに

本発表では、共有並列計算機を対象とした並列化プログラミング支援システムParassistの概要について述べた。今後、実験評価を重ね、機能の充実を図る予定である。

参考文献

- [1]Dr. John M. Levesque. An Interactive Tool for Vectorization and Parallelization of Fortran Programs. Pacific Sierra Research Corp.
- [2]Ken Kennedy, K. S. McKinley, and Chau-Wen Tseng. Analysis and transformation in the ParaScope Editor. In Proc. ACM International Conf. on Supercomputing, Cologne, Germany, June, 1991.
- [3]M. Burke and R. Cytron. Interprocedural dependence analysis and parallelization. In Proc. SIGPLAN'86 Symposium on Compiler Construction, Palo Alto, CA, July 1986.
- [4]木村ほか: 手続き間解析機能の検討, 情報処理学会第41回全国大会(1991)