

## 並列化支援環境 *PCASE*

†渡辺 幸光、†白戸 幸正、‡草野 和寛、‡蒲池 恒彦、‡妹尾 義樹

† NEC 情報システムズ (株)  
〒 213 川崎市高津区坂戸 3-2-1

‡ NEC C&C 研究所  
〒 216 川崎市宮前区宮崎四丁目 1-1

あらまし

Fortran プログラムの並列化作業を支援するツールである、並列化支援環境 *PCASE* (Parallelizing CASE) を開発した。*PCASE* は 逐次 Fortran プログラムを並列 Fortran プログラムに変換する過程において、グラフィカルユーザインタフェースを用いた対話処理をしターゲットマシンに最適なチューニングを行うツールである。様々なプログラム情報表示機能を持ち、利用者は簡単な指示を行う事でプログラムの並列化が可能である。また、プログラム実行時間を予測する機能を持ち、プログラムを実行する事なく最適なチューニング作業が行える。本論文では *PCASE* の概要を述べ、*PCASE* の並列化支援手法について説明する。

和文キーワード 並列化, 性能予測, 手続き間解析, Fortran

## Parallel programming environment *PCASE*

†Yukimitsu Watanabe,†Yukimasa Shiroto  
‡Kazuhiro Kusano,‡Tunehiko Kamachi,‡Yoshiki Seo

†NEC Informatec Systems, Ltd.  
〒 213 3-2-1 Sakado, Takatsu, Kawasaki

‡C&C Research Laboratories, NEC Corporation  
〒 216 4-1-1 Miyazaki, Miyamae, Kawasaki

Abstract

We have been developing an interactive parallel programming environment *PCASE* which supports programmers to translate sequential Fortran programs into efficient parallel programs without pains. *PCASE* provides a graphical user interface(GUI) for an interactive parallelization. Users can obtain the various kinds of information about the target program and specify instructions to control parallelizaion through the GUI. In addition, *PCASE* has the capability of predicting performance without running on the target machines.

This paper describes a variety of powerful analysis capabilities and techniques in *PCASE*.

英文 key words parallelization, Performance prediction, interprocedural analysis, Fortran

## 1 はじめに

近年、並列計算機の研究、開発が盛んに行われており、商用並列計算機も開発されるようになってきた。しかし、これら並列計算機のハイパフォーマンスを最大限に引き出すアプリケーションを開発するには、並列アルゴリズムに関する知識はもちろんの事、対象マシン固有のアーキテクチャ特性についての十分な知識が必要となる。

また、利用者が既存アプリケーションの並列計算機への移植を行う場合、上記の知識をふまえたプログラムの解析/書き換えが必要となる。しかし莫大な時間を掛け、書き換えを行ったアプリケーションも、場合によっては予想より大幅に遅い動作をする事がある。こうした場合、十分な実行速度が得られるまでプログラムの書き換え作業を繰り返す事になる。これらが、並列計算機の普及を阻害する要因である。

我々は、このような背景から Fortran プログラムの並列化作業を支援するツールとして、並列化支援環境 PCASE (Parallelizing CASE) を開発した。近年、国内外において並列化解析/支援ツールが研究開発されているが、PCASE は Fortran で記述された逐次実行型のプログラムを並列実行型プログラムに自動変換するとともに、その性能を予測し、対話的にプログラムの高速化を可能にする特徴をもつ。

本稿では、2章で PCASE の概要を述べた後、3章で並列化支援手法を、4章で今後の予定について述べ、5章でまとめを述べる。

## 2 PCASE の概要

PCASE は利用者が記述した逐次プログラムを並列実行プログラムに変換するが、その過程において、利用者は GUI を介して対話形式に指示を与え並列化作業を行う。PCASE は利用者が直感的に情報を理解出来、容易に並列化作業が行える強力な並列化作業環境を提供する (図 1参照)。

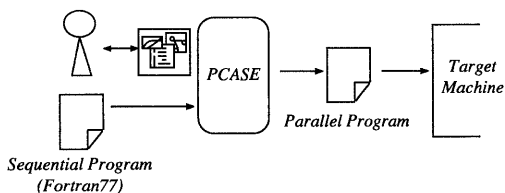


図 1: PCASE 利用イメージ

また、PCASE はターゲットマシンのアーキテクチャを変更する事が出来、ターゲットマシンに最適なデータ分散配置、データ転送コードの挿入を自動的に行う機能を有する。

PCASE は Xpallas と DCM(Data distribution & Communication Manager) の 2 つのツールで構成される (図 2参照)。

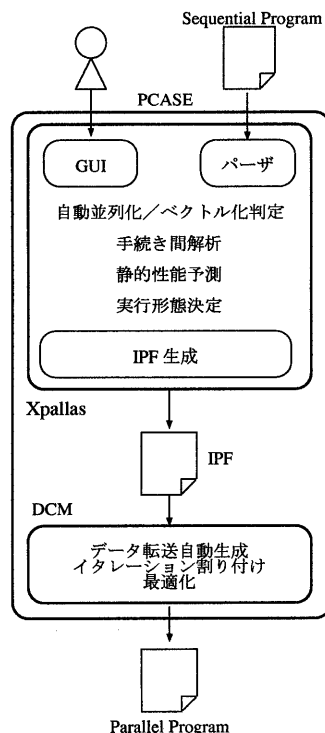


図 2: PCASE の System 構成

Xpallas は入力されたアプリケーションプログラムの解析をし、利用者との対話を行い、実行情報やデータ分散情報を記述した中間言語で記述されたファイル (以下 IPF<sup>1</sup>) を生成する。

DCM は IPF を入力し、ターゲットマシンに最適なデータ転送、ループイタレーションのプロセッサへの割り付け、同期最適化処理をおこなひ、並列プログラムに変換するトランスレータである。

中間言語 IPF を導入する事により、ターゲットマシンに依存しない解析、する解析を分離する事が出来、PCASE 環境をより多くのマシンに対応させる事が可能な構成となっている。

<sup>1</sup>Intermediate Parallel Form

## 2.1 PCASE の特徴

並列計算機を必要とする利用者は多種多様である。また、ターゲットマシンの利用が困難で頻繁に実行評価／チューニングを繰り返さない場合も考えられる。

PCASE は様々な利用目的や作業環境に柔軟に対応し、快適な並列化作業が行えるよう、以下の特徴をもつ。

- Fortran77 ソースコードを入力とし、マシン依存の並列化 Fortran コードを出力する
- X Window System を利用した、利用者インタフェースを備え、ビジュアルな情報表示、利用者からの指示が行え、高速なアプリケーションの開発が可能
- WS 上で性能予測が可能であり、予測性能を評価しながらチューニングが行え、並列化作業時に並列マシンを必要としない
- アーキテクチャが複数用意されており、各種パラメータを変更する事で柔軟なターゲットマシンの選択／変更が可能である
- 中間言語 IPF を導入しており、ターゲットマシンに依存する処理は DCM が行い、マシンに合わせた最適なチューニングが可能である
- ベクトル PE を有するターゲットマシンの場合、並列化と合わせてベクトル化も考慮される

## 2.2 サポートアーキテクチャ

PCASE ではターゲットマシンのアーキテクチャは以下のパラメータを設定する事により変更できる。

- PE 台数
- ベクトルプロセッサ有無
- メモリアーキテクチャ

メモリアーキテクチャとして、下記の2種類のメモリ空間を提供している。利用者はターゲットマシンにあわせてメモリ空間を組合せ選択できる。この選択によりデータ転送コードの挿入を最適に行える。

- Local Area  
各プロセッサに固有のメモリ空間であり、高速アクセスが行える
- Shared Area  
各プロセッサからアクセス可能な空間

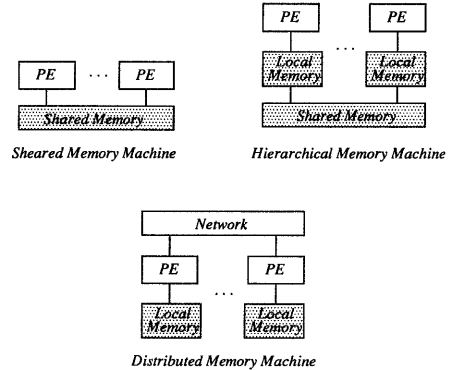


図 3: PCASE サポートメモリアーキテクチャ

以上のパラメータを組み合わせる事で、ターゲットマシンのアーキテクチャを表現する。

## 2.3 Xpallas の機能

Xpallas は、アプリケーションを解析し、様々な情報を利用者に提供し、対話処理を経て IPF を出力するが、その機能を以下に示す。

- Do ループの並列化／ベクトル化／ループ再構成の可能性判定
- 手続き間に渡る、定数伝播、配列アクセス範囲解析、並列化可能性判定
- 静的性能予測を用いた、自動実行形態<sup>2</sup>決定、及び自動最適化
- 利用者指示による並列化／ベクトル化
- 利用者指示による実行形態の決定
- 静的性能予測
- 高速実行トレース採取
- プログラム情報 (Call Graph, Control Flow, Data 依存関係, 予測性能, etc..) 表示

アプリケーション情報、ユーザ指示情報、実行形態情報をデータベース化し、Xpallas の各モジュールはデータベースを参照、更新する事でアプリケーションの並列化／ベクトル化を進める。

データベース化する事により、手続き間解析時には必要な情報のみ参照でき、巨大アプリケーションの手続き間並列化解析が可能となる。また、各モジュールは独立に動作し作業時に起動するため、Xpallas 実行時に必要なメモリ空間を節約している。

<sup>2</sup>多重ループ内の、個々のループ実行形式の組合せ

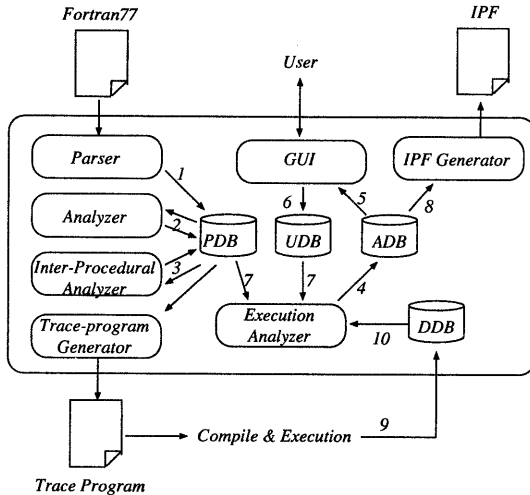


図 4: Xpallas System 構成

IPF 生成までの流れを、図 4 を用いて説明する。番号は図 4 中の矢印番号に対応する。

1. 利用者に指定された Fortran77 記述の逐次プログラムは *Parser* によってプログラム情報データベース (PDB) となる
2. 手続き内基本解析 (データ依存解析、データアクセス解析、フロー解析、等) を *Analyzer* が PDB より行い、結果を PDB に付加する
3. 全ての手続きの基本解析が終了した場合、*Inter-Procedural Analyzer* によって、手続き間データ依存解析、手続き間データアクセス解析、手続き間フロー解析が行われ、結果は PDB に付加される
4. 以上の解析情報を元に *Execution Analyzer* は実行形態を決定し、解析結果データベース (ADB) を生成する。ADB は PDB の内容に実行形態情報が付加されたものである
5. GUI 部はこの ADB を参照し、利用者にプログラム情報や、解析結果情報を伝える
6. GUI を通して指示された利用者指示情報は利用者指示データベース (UDB) として保存される
7. UDB が存在する場合、*Execution Analyzer* は UDB と PDB を参照し、ADB を生成する
8. 矢印 4-5-6-7 の繰り返しにより ADB を最適な情報をもつデータベースにし、最終的に

IPF Generator が ADB から IPF を生成する

9. *Trace-program Generator* が PDB より生成した *Trace Program* を実行する事でプログラム実行トレース情報 (DDB) が生成される。
10. DDB は、*Execution Analyzer* が静的性能予測に利用する

## 2.4 IPF

IPF は *Xpallas* によるチューニングを行った結果であり、データ分散情報、実行形態情報、データアクセス情報を指示行形式で記述した Fortran コードである。

各指示行の種類と意味を表 1 に示す。

なお、現在 *Xpallas* はデータ分割の解析を行っておらず、*DISTRIBUTE* 指示行は利用者が記述しなければならないが、*DISTRIBUTE* 指示行を挿入したプログラムを *Xpallas* の入力とした場合は、データの分割配置を考慮した並列化解析を行う。

表 1: IPF 指示行

指示行	意味
PARDO[FOR=n]	直後の DO ループをスライス n で並列化する
BY=n	並列ループを繰り返し数 n でスライスする
VECTOR	直後の DO ループをベクトル化する
SCALAR	直後の DO ループをシリアル実行する
LOCAL	ループ内の変数のアクセス範囲/属性の指示
RESERVE[=n]	n の並列タスク数で実行する
BARRIER	全タスクで同期をとる
MACRO	直後のループはマクロ <sup>3</sup> 演算である。総和、累積、漸化式、最大、最小をサポートする。
DISTRIBUTE	並列の分割配置の指定である。BLOCK 分割、CYCLIC 分割をサポートする

例 1 は、配列 A,B,C は 1 次元目で BLOCK 分散配置し、16 並列実行を要求している。DO 10 ループは並列実行させる事を指示し、配列 A は

<sup>3</sup>データ依存関係に不適合があるが、並列化/ベクトル化を行う演算パターン

DO 10 ループ内において2次元目が1-31の間に更新アクセスが、配列Bは2次元目が2-32間に参照アクセスが、配列Cは2次元目に1-31間の参照アクセスがそれぞれ発生する事を示している。

また、ループカウンタの式が記述されている次元(例1ではA,B,Cの1次元目)は分散配置が行われている次元を示し、式はループ繰り返しにおけるアクセスの変化を示している。

DCM はこれらの情報からデータ転送コードを生成する。

```
*PDIR DISTRIBUTE A(BLOCK,*)
*PDIR DISTRIBUTE B(BLOCK,*)
*PDIR DISTRIBUTE C(BLOCK,*)
*PDIR RESERVE=16

.P
*PDIR PARDO
*PDIR LOCAL A(i,1:31),MOD
*PDIR LOCAL B(i+1,2:32),REF
*PDIR LOCAL C(i,1:31),REF
do 10 i=1,n-1
  do 20 j=1,n-1
    a(i,j)=b(i+1,j+1)*c(i,j)
  20 continue
10 continue
```

例 1: IPF の例

## 2.5 DCM の機能

DCM は *Xpallas* が出力した中間並列化形式プログラム IPF からターゲットマシン用の並列プログラムを生成するトランスレータであり、以下の機能を有する。

- ランタイムメモリ管理解析
- イタレーションのプロセッサへの割り付け
- データ転送コード/同期の生成/挿入
- マクロ演算の並列化
- 最適化

今回分散共有メモリ並列マシン Cenju2用のDCMを作成した。Cenju2は、

- 他のPEのローカルメモリへはシステムでユニークなグローバルアドレスによってアクセスすることができる。

- 各PEは送受信の2つのDMAC(Direct Memory Access Controller)を備え、ローカルメモリ間でデータコピーを高速に行う事が出来る。
- 各PE間はマルチキャスト機能を組み込んだ多段接続網で接続される。

といった特徴をもつ。これらの特徴を考慮し、Cenju2の性能を最大限に引き出せるようにCenju2-DCMが実装されている。

## 3 PCASE による並列化支援手法

利用者が並列アプリケーションを作成する場合、一般的に図5の方法をとると考えられるが、十分な結果が得られるまでには、何回もの性能評価とプログラムの書き換えが発生する。又、それぞれの作業では様々な知識を必要とする。

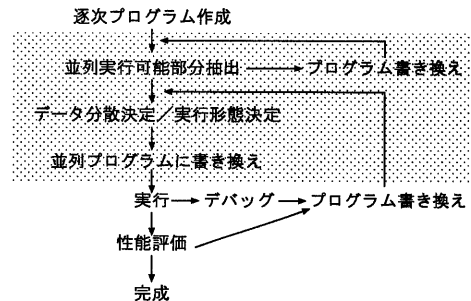


図 5: PCASE を利用しない場合の並列化手順

PCASE は図5の網かけ部分の作業を支援する事により、並列化作業の負荷を軽減させる。

PCASE を利用した場合、プログラム内の並列実行可能なDOループを抽出し、静的性能予測を用いて自動的に実行形態を決定し利用者に提示する。性能予測情報によって、十分な結果が得られるならば並列ソースプログラムを生成し実マシンでの実行性能評価をする。しかし、十分な予測結果が得られない場合、実行形態を選択的に変更する、並列不可原因を取り除く等を対話的に行う。このようにPCASEは、並列実行可能部分の抽出、並列不可原因の除去、実行形態決定、静的性能予測をする事により利用者の並列化作業の負担を大幅に軽減させる。

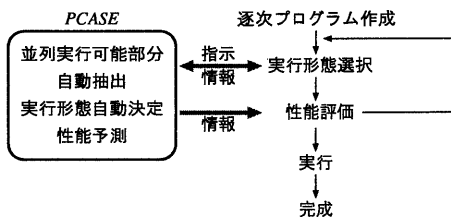


図 6: PCASE を利用した場合の並列化手順

利用者に対する情報提供、対話処理は全て *Xpallas* が行う。以降本稿では、*Xpallas* の並列化支援手法について述べる。

### 3.1 *Xpallas* による自動並列化 / ベクトル化

*Xpallas* は始めにアプリケーションに対して以下の解析を行い、解析結果を完全自動並列化結果として利用者に示す。

#### 手続き内 / 間解析

ループ内データ依存解析、アクセス範囲解析、定数伝播解析、マクロ演算解析を手続き内 / 間に渡り解析する。

#### 並列可能性判定

Do Loop に対する並列化 / ベクトル化 / ループ再構成の可能性を判定する。判定には手続き間 / 手続き内定数伝播情報、手続き間 / 手続き内データ依存情報、マクロ演算認識情報、ループ再構成 (分割, 入れ替え) 情報を用いており、高度な並列実行の可能性を判定している。

#### 自動実行形態決定

並列実行の可能性のあるループの中から、実際に並列実行を行い効率が上がるものを選択する。現在の DCM の機能制限として Loop の多重並列実行は行えない為、多重の並列ループ内から効率の上がるループを選択する。選択には、静的性能予測情報を利用し最適な実行形態を生成する。

#### 静的性能予測

データ分散配置情報、実行形態を元に各種のオーバーヘッドを考慮し、実行時間、台数効果、性能向上率を生成する。

利用者は静的性能予測が提供するターゲットマシンでの予測実行時間、並列台数効果、性能向上率情報より十分な性能が得られるものかを判断する。しかし、何らかの原因によりアプリケーションの最大性能を引き出す事が出来ない場合があ

る。その場合、利用者は *PCASE* に対して少しの指示を与える事によりアプリケーションをその最大性能に近づける事ができる。

### 3.2 ボトルネックの発見

完全自動並列化では期待した性能向上が得られないと判断した場合、利用者は初めにチューニング対象のアプリケーションのどの部分を重点的にチューニングするのが効果的かを判断する。*PCASE* では、CallGraph を表示するとともに、それぞれの手続きの予測実行速度を表示する。利用者はこの情報を元に台数効果の得られていない、並列化がほとんど行われていない手続きを選択する。

また、手続き選択後は手続き内のどの Loop が並列化不可能であるか、どの Loop がボトルネックになっているかを参照できる為、重点的にチューニングすべき、Loop を容易に検出できる。

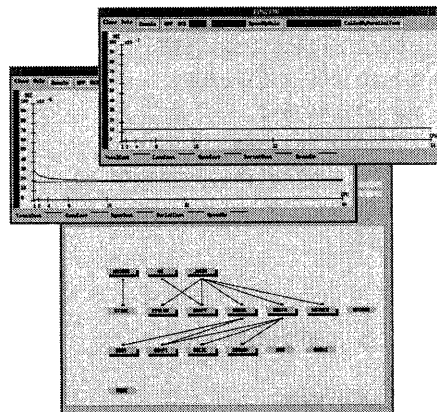


図 7: ボトルネック発見

### 3.3 利用者指示による並列化 / ベクトル化

並列可能性判定時に並列実行不可能と判断されたものは、その原因が利用者に提示される。利用者が、その情報を元に依存の切断指示やソースの書き換えを行う事により、一層の Loop の並列化が可能である。

提示される原因は

- 並列化 / ベクトル化不可能なデータ依存関係がある
- I/O 文がある (並列実行時に I/O をサポートしていない場合)

● etc...

でありそれぞれの原因を無効にする指示が行える。これは、動的要因により並列化/ベクトル化を判断できない場合に *PCASE* では並列化/ベクトル化阻害原因として扱う為、その原因を解消する為のものである。

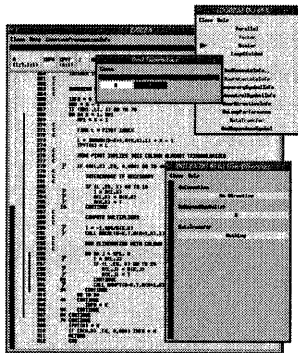


図 8: 並列化/ベクトル化阻害原因表示

### 3.4 実行形態の選択

ループ並列化作業によって、いくつもの Loop が並列実行可能となった場合、その最適な組合せを選択する。*PCASE* によって最適な実行形態が自動的に示されるが、利用者が意図する形態にした場合は各 Loop に対して実行形式を指示する事が可能である。但し、指示に矛盾がある場合、例えばベクトルループの内側に並列ループが指定されているような場合には外側ループに指示されたものを有効とする。静的性能予測情報を参照する事で、利用者は指示の効果を確認出来、ほぼ、実時間の最速になる組合せを指示できる。

### 3.5 性能評価

通常、チューニングしたアプリケーションの速度評価は実際にターゲットマシンで動作させて評価するが、*PCASE* はその性能を予測する機能をもつ。この機能を利用する事により、ターゲットマシンを利用する事なく性能評価/チューニングの繰り返しを行える。また、性能予測結果はユーザの指示が即座に反映されるため、的確な並列化指示が行える。

予測結果は静的性能予測情報として自動実行形態決定にも利用されている。

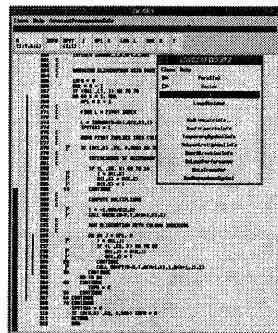


図 9: 実行形態指定

### 3.6 高速実行トレース採取

前述の性能予測は行実行回数をプログラム内情報や定数伝播等の静的パラメータより予測するが、動的パラメータがアプリケーションの動作に著しく依存する場合等は、予測結果が実際の結果と大きく異なる事がある。このような場合、*PCASE* はアプリケーションを仮実行し、そのトレース情報を利用し性能予測精度を向上させる機能をもつ。

トレース情報をとるプログラムは *PCASE* によって生成される。利用者が、このプログラム(トレースプログラム)をコンパイル&実行する事でトレース情報を生成する。

また、トレース情報を高速に採取する為、基本 Block の実行回数に影響が無い文、不必要となった Symbol の宣言が削除される。この機能によりトレース情報を CPU パワーの無い WS 等でも採取する事が可能となる。

表 2 は高速実行トレース採取を行った場合と通常のトレース採取を行った場合の比較である。

表 2: トレース採取結果

Application Name	Target program+ $\alpha$	Trace program
SPIN	5.1(sec)	0.24(sec)
FFT	1.64(sec)	0.06(sec)
LU	301.86(sec)	32.81(sec)

SPIN: 相移解析

FFT: 高速フーリエ変換

LU: LU 分解

$\alpha$  は実行トレース採取のオーバーヘッド

アプリケーションによっては約 20 倍の速度向上が得られる。データ採取には NEC EWS/230<sup>7</sup>を用いた。

### 3.7 プログラム情報表示

利用者がアプリケーションのチューニングを行う場合の支援として、以下の情報を表示する。

- 手続き CallGraph
- 手続き / Loop 性能予測情報
- Control Flow Graph
- 基本 Block 情報
- ソースプログラム
- データ転送情報
- Symbol 定義情報
- Common Block 情報
- 実行形態情報
- 並列 / ベクトル実行不可原因情報
- データアクセス情報
- データフロー解析情報

利用者は情報を参照しながら各種の指示を PCASE に対して与える。

## 4 今後の予定

並列マシンが商用化され、今後並列マシン利用者が増える事を考慮すると、様々な作業の支援が必要となる。

今後は支援項目を拡大し、並列プログラム作成、コンパイル、実行、デバッグ等の並列マシンを利用する全ての作業をサポートする予定である。

PCASE は現在、中間言語に IPF, Cenju2-DCM では出力言語に Fortran77+Cenju Prallel Library を採用しているが、将来的には IPF は HPF(High Performance Fortran) に、出力言語には Fortran90+MPI(Message Passign Interface) Library とし、インタフェースの標準化を行う。また、現在はデータ分散の指示を利用者が行っているが、PCASE が自動で行うようにするとともに、各種のコンサルティング機能や、強力なソース変形機能等も採り入れて行く予定である。

## 5 まとめ

並列化支援環境 PCASE の概要を述べ、その並列化支援手法を PCASE の構成ツールであ

<sup>7</sup>CPU: VR3600A 33MHz

る Xpallas の機能を中心に述べた。また、今後の PCASE の予定機能を述べた。

今後は PCASE の評価を行い、並列化支援システムに求められる機能、性能等を検討し、その結果を元に PCASE の開発を進めて行く予定である。

### 謝辞

本研究を進めるにあたり、貴重な提案・助言を頂きました基本ソフトウェア事業部 第一言語開発部 左近 彰一主任ならびに、評価を行って頂いた関係者の方々に感謝します。

### 参考文献

- [1] 草野 他: 手続き間定数伝播処理の検討, 情処 46 回全大, 5-67, 1993.
- [2] 渡辺 他: プログラム実行トレースの高速採取法とその評価, 情処 45 回全大, 5-247, 1992.
- [3] 蒲池 他: 並列化支援環境 PCASE における分散メモリ対応機能, jspp93, 31-38, 1993.
- [4] High Performance Fortran Forum: High Performance Fortran Language Specification, Version 1.0, 1994.
- [5] Message Passing Interface Forum: A Message Passing Interface Standerd, Version 1.0, 1994.