



4. HPF トランスレータ “Parallel FORTRAN” の開発と評価

Development and Evaluations of HPF Translator “Parallel FORTRAN” by Makoto SATO, Hiroshi OHTA (Systems Development Laboratory, Hitachi Ltd.) and Eiji NUNOHIRO (Software Development Center, Hitachi Ltd.).

佐藤 真琴¹ 太田 寛¹ 布 広 永 示²

¹ (株)日立製作所システム開発研究所

² (株)日立製作所ソフトウェア開発本部

1. はじめに

HPF 1.0¹⁾の全仕様と Fortran 90 をサポートするトランスレータ Parallel FORTRAN を製品化した。HPF 指示文と独自拡張仕様を用いて記述した NPB 1.0 (NAS Parallel Benchmarks)²⁾ をトランスレートして、当社超並列機 SR 2201 で実測した結果、8 本中 7 本について、良好な台数効果を得た。

2. Parallel FORTRAN の概要

本章では、Parallel FORTRAN の仕様と最適化項目の概要を述べる。

2.1 入力言語仕様

(1) HPF 1.0 全仕様のサポート

HPF 1.0 全仕様である HPF 指示文、Forall 構文などの新しい文、HPF ライブラリをサポートしている。

(2) Fortran 90 のサポート

配列構文やモジュールなど Fortran 90 全仕様をサポートする。

(3) 独自拡張仕様のサポート

並列化の促進や不要コードの生成抑止のため、3 つの拡張指示文を設けた。2.4 節で詳述する。

2.2 出力言語仕様

PARALLELWARE (Express)^{*}または当社独自の高速リモート DMA (Direct Memory Access) 通信を PE (Processor Element) 間通信関数として含む Fortran 90 プログラムを出力する。

2.3 最適化項目

Parallel FORTRAN は以下の通信最適化と計算分割最適化機能を備えている。この内、(5)、(6)、(7)は 3 章で詳述する。その他は、文献 4) 参照。

- (1) 通信ベクトル化
- (2) 通信融合
- (3) 冗長通信削除
- (4) 集合通信・リダクション変換
- (5) ループ繰返し範囲分割
- (6) IF 文のループ外移動
- (7) タイリング

2.4 独自拡張仕様

Parallel FORTRAN は、HPF 1.0 以外に 3 つの独自拡張仕様をサポートしている。これらは「!PFD\$」で始まる行に記述する。

(1) OVERLAP 指示文

各 PE が所有する配列の宣言範囲を拡張して、通信データの受信領域として利用する部分を、オーバーラップ領域と呼ぶ。本指示文は、配列のオーバーラップ領域の大きさを各次元の上下限值に対する拡張幅として指定する。

!PFD\$ OVERLAP 配列名 (下拡張幅：
上拡張幅,...)

本指示文は、手続き引数に適用した時、効果がある。本指示文がないと、トランスレータは実引数と仮引数に対して異なる宣言のオーバーラップ領域を生成する可能性があるため、実引数ではアドレスが連続する 2 つの配列要素が、仮引数では非連続になるなどの現象が発生し、手続きの入口と出口でデータのコピーが必要となる。本指示文は両者のオーバーラップ領域の宣言を等しくすること

^{*} PARALLELWARE は新日本製鉄(株)の登録商標です。PARALLELWARE の米国での製品名称は Express です。Express は米国での Parasoftware Corporation の商標です。

で、コピーを不要にする。同様な仕様は HPF 2 で提案されている²⁾。

(2) REDUCTION オプション

INDEPENDENT 指示文⁴⁾のオプションとして、リダクション⁴⁾の種類と変数名を指定する。

```
!PFD$ INDEPENDENT, REDUCTION:種類
      (変数名)
```

種類は総和 (SUM) や最大値 (MAX) などを、変数名はこれらの結果を格納する変数名を指定する。HPF 2 ではリダクションを表す式を単純なパターンに限っている⁴⁾が、本オプションではリダクションの種類と変数名を指定すればよいので、パターンに制限はない。

(3) LOCAL PURE 指示文

PURE 手続き⁴⁾が、(a)他 PE の分割データを参照せず、(b)同手続きで値が変更される可能性のある分割データは、実引数ならびにある配列要素に限る、ことを指示する。

```
!PFD$ LOCAL_PURE
```

手続き呼び出しを含むループを分散するには、同手続き内の通信文と実行文の実行 PE を知る必要がある。前者は、条件(a)より通信が発生しないので考慮は不要であり、後者は、条件(b)と owner computes rule⁴⁾より実引数の実行 PE に等しい。

したがって、上記実引数の実行 PE とループ内のほかの部分の実行 PE より、ループの分散が可能となる。

3. 最適化技術

本章では、2.3 節であげた最適化機能のうち、特徴的なものと、最適化を促進する解析技術としてデータ再分割解析を説明する。

3.1 データ再分割解析

データ再分割指示文をトランスレート時に解析し、プログラム中の任意位置でのデータのマッピングを決定する。これが一意に決定できれば、たとえば、ブロック分割であることを利用した通信のベクトル化などの最適化が可能となる。

本解析は、各データのマッピング情報をデータ再分割指示文ごとに更新しながらプログラムの流

れに沿って伝播させることで、任意位置のデータマッピング情報を求める⁶⁾。

なお、データマッピングがトランスレート時に一意に決定できないときは、マッピング情報を格納する構造体を用いて、プログラム実行時に決定する。

3.2 計算分割最適化

(1) ループ繰返し範囲分割⁶⁾

ループ内の複数の代入文を 1 プロセッサが実行するときの、各々のループ繰返し範囲が異なる場合を考える (図-1)。PE 番号を 0 から 9 までとすると、PE 1 が所有する配列要素は、A (11:20)、B (11:20) となり、owner computes rule⁴⁾より、同 PE が各文を実行するループ繰返し範囲は、1 行目は J=11~20、2 行目は J=10~19 となる (図-2)。

このとき、単純な並列化では、並列化後のループ範囲は各文の繰返し範囲の和集合 J=10~20 となり、各文には繰返し範囲を判別して文の実行を制御する IF 文が付加される (図-3(a))。このため、余分なループ繰返しや余分な IF 文の実行が発生する。

本最適化は、並列化後のループ範囲を、各 IF

```
REAL A(100), B(100)
!HPF$ PROCESSORS MP(10)
!HPF$ DISTRIBUTE (BLOCK) ONTO MP :: A, B
DO J = 1, 99
  A(J) = ...
  B(J+1) = ...
```

図-1 繰返し範囲が異なる複数の代入文

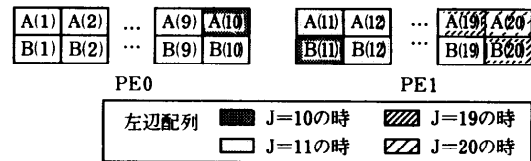


図-2 左辺配列要素とループ繰返しの対応

```
DO J = 10, 20
  IF(10<J)
    A(J) = ...
  IF(J<20)
    B(J+1) = ...

DO J = 10, 10
  B(J+1) = ...
DO J = 11, 19
  A(J) = ...
  B(J+1) = ...
DO J = 20, 20
  A(J) = ...
```

(a)適用前 (b)適用後

図-3 ループ繰返し範囲分割前後のプログラム

```

REAL A(100)
!HPF$ PROCESSORS MP(10)
!HPF$ DISTRIBUTE (*, BLOCK) ONTO MP :: A
DO J = 1, N
  A(J, 1) = ...      ! PE0のみが実行
    
```

図-4 特定 PE による代入文の実行

```

DO J = 1, N
  IF(P=0)
    DO J = 1, N
      A(J, 1) = ...
    
```

(a) 移動前

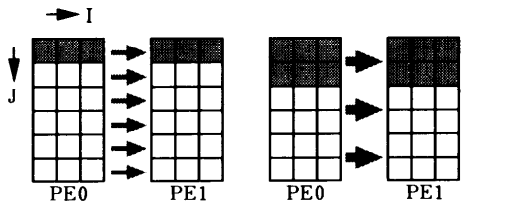
(b) 移動後

図-5 IF 文のループ外移動前後のプログラム

```

!HPF$ PROCESSORS MP(10)
!HPF$ DISTRIBUTE (BLOCK,*) ONTO MP :: A
DO J = 1, N
  DO I = 1, N
    A(I, J) = A(I-1, J) + ...
  
```

図-6 DOACROSS ループの例



(a) 適用前

(b) 適用後

→ 通信

図-7 タイリング前後の通信

文の条件式のループ範囲を用いて細分する。この結果、すべての IF 文は削除される (図-3(b))。

なお、PE によって、ループ数が異なることがあるので、コード出力に工夫が必要である⁶⁾。

(2) IF 文のループ外移動

特定 PE のみがループ内の文を実行する場合 (図-4)、本最適化を適用しないとループ繰返しごとに IF 文を実行する (図-5(a))。ループ全体を特定 PE 番号で制御すると、IF 文の実行回数は 1 回になる (図-5(b))。

3.3 通信最適化

(1) タイリング

ループ繰返し間に実行順序関係のある DO-ACROSS ループ (図-6)⁹⁾ を単純に並列化すると、J 繰返しごとに 1 回、通信が発生する (図-7(a))。複数の J 繰返しをまとめて計算してもプログラムの意味が不変な場合、複数繰返しに 1 回の割合での通信生成が可能となり、通信回数

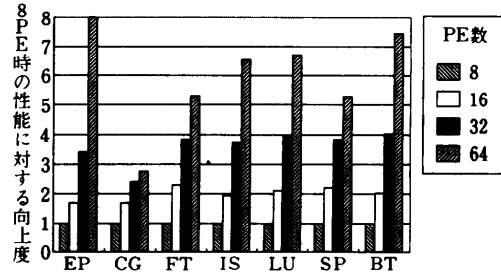


図-8 Parallel FORTRAN による NPB 1.0 実測性能

表-1 NPB 1.0 に対する指示文・最適化の寄与

	EP	CG	FT	IS	LU	SP	BT
OVERLAP					○		
REDUCTION	○				○	○	○
LOCAL_PURE	○		○				
通信ベクトル化					○		
タイリング					○		

が削減できる (図-7(b))。過度な適用は実行の逐次化を招き、遅くなるので、適切な回数を決定している⁵⁾。

4. 評価

並列計算機の代表的ベンチマーク NPB 1.0³⁾ を HPF 指示文と独自拡張指示文を用いて記述し⁷⁾、これを本トランスレータで並列化して、SR 2201 で評価した。問題規模は Class A である。NPB 全 8 本のプログラムの並列化が可能となり、内 7 本は良好な台数効果を得た (図-8)。表-1 は、同結果に対する独自拡張指示文と最適化技術の寄与を表す。

5. おわりに

HPF トランスレータ Parallel FORTRAN は、HPF 1.0 の全仕様と Fortran 90 をサポートする。独自拡張仕様と最適化技術により、NPB 1.0 全 8 本の並列化が可能となった。

参考文献

- 1) High Performance Fortran Forum: High Performance Fortran Language Specification Ver. 1 (May 1993).
- 2) High Performance Fortran Forum: High Performance Fortran Language Specification Ver. 2.0. α . 2 (May 1996).
- 3) Bailey, D. et al.: The NAS Parallel Benchmarks, RNR Technical Report, RNR-94-007,

- NASA Ames Research Center (Mar. 1994).
- 4) 妹尾義樹: HPF 言語の現状と将来, 情報処理, Vol. 38, No. 2, pp. 90-99 (Feb. 1997).
 - 5) Ohta, H., Saito, Y., Kainaga, M. and Ono, H.: Optimal Tile Size Adjustment in Compiling General DO-ACROSS Loop Nests, ICS '95, pp. 270-279 (1995).
 - 6) Sato, M., Hirooka, T., Wada, K. and Yamamoto, F.: Program Partitioning Optimizations in an HPF Prototype Compiler, Compsac '96, pp. 124-131 (1996).
 - 7) 太田, 西谷, 小林, 布広: HPF 処理系による NAS Parallel Benchmarks の並列化, 情報処理学会研究報告, 96-HPC-62, pp. 57-62 (1996).
 - 8) 佐藤, 根岸, 小林: HPF 処理系における再分散解析機能の開発, 情報処理学会研究報告, 96-HPC-62, pp. 63-68 (1996).
 - 9) Zima, H. and Chapman, B.: Supercompilers for Parallel and Vector Computers, ACM Press (1991).

(平成8年9月25日受付)



佐藤 真琴 (正会員)

1959年生. 1987年神戸大学大学院理学研究科数学専攻修了. 同年, (株)日立製作所入社. 中央研究所を経て, 現在, システム開発研究所勤務. 入社以来, 最適化コンパイラ, デバッガの研究に従事. 並列処理および最適化に興味をもつ.
e-mail: msatoh@sdl.hitachi.co.jp



太田 寛 (正会員)

1962年生. 1987年東京大学大学院理学系研究科地球物理学専攻修了. 同年, (株)日立製作所入社. システム開発研究所勤務. 入社以来, 論理型言語処理系, 並列化コンパイラの研究に従事. 並列処理ソフトウェア全般, 並列アーキテクチャに興味をもつ. 電子情報通信学会, ACM, IEEE 各会員.
e-mail: ohta@sdl.hitachi.co.jp



布広 永示 (正会員)

1957年生. 1985年日本大学大学院生産工学研究科数理工学専攻博士課程中退. 工学博士. 現在, (株)日立製作所ソフトウェア開発本部勤務. 数値解析の分野においてとくに丸め誤差解析, 並列処理向き言語, 開発支援システムなどに興味をもつ. 日本応用数学会会員. e-mail: nunohiro@soft.hitachi.co.jp