

分散メモリ型並列計算機を対象とする 並列化トランスレータ

忠 政 慎 也*・小 畑 正 貴**

*岡山理科大学大学院工学研究科博士課程システム科学専攻

**岡山理科大学工学部情報工学科

(1997年10月6日 受理)

1. はじめに

流体計算などの大規模数値計算には、並列処理による高速化が期待されている。また、ワークステーション・クラスタや分散メモリ型並列計算機により、並列処理が身近になってきている。

しかし並列処理を行なうために記述するプログラム（並列プログラム）は、並列化可能部分の認識や通信関数の導入などを必要とし、一般的なC言語やFORTRAN言語で記述されたプログラム（逐次プログラム）に比べて、プログラミングがかなり困難である。HPF (High Performance Fortran) などの並列言語を用いることにより、多少簡単に並列プログラムを記述できるようになってきているが、プロセッサへどのようにデータを割り当てれば並列効果が得られるかなどの知識は必要とされる。そのため、並列処理によって高速化可能な場合でも、並列プログラミングの複雑さに、並列処理を扱えないユーザが多いのが現状となっている。ユーザにとって、並列処理を意識せずに実装できるということはとても魅力的である。

そこで本研究では、一般的なFORTRAN言語でプログラムを記述できる程度の知識で並列処理を実装するための支援として、並列化トランスレータを開発している¹⁾²⁾。並列処理を実装する計算機には、Intel社の分散メモリ型並列計算機Paragon³⁾を対象としている。

開発中の並列化トランスレータは、FORTRANプログラムとプロセッサ数を入力することにより、自動的にFORTRAN言語にNX通信関数が拡張されたSPMDコードを生成する。生成されたコードはParagon用のコンパイラif77によりコンパイルされ実行形式となる。こうすることにより、Paragonに適した最適化をコンパイラに任せることができ、また、通信関数の変更により他の分散メモリ型並列計算機に対応させやすくなると考えられる。

本稿では、まず2章で分散メモリ型並列計算機を対象とする並列プログラミングについて述べ、3章では並列化トランスレータの構成、4章では、データ分割配置を決定する際の前処理となるメインブロックの決定方法について、5章では簡単な数値計算プログラムの並列化に対する有効性について述べ、6章でまとめる。

2. 分散メモリ型並列計算機を対象とする並列プログラミング

Paragonのような分散メモリ型並列計算機は、各プロセッサにローカルメモリを持っているため、共有メモリ型並列計算機と比べて、メモリへのアクセスが集中しにくくなり、並列処理効果を期待できる。しかし、他のプロセッサのデータを参照するには通信の記述が必要となり、プログラミングはかなり困難となる。

また、分散メモリ型並列計算機上で実装するプログラムの記述の特徴として、プロセッサへのデータ割り当てを明示しなければならない。例えば、プログラム中に存在する配列データAの要素の各々をどのプロセッサに割り当てるかということである。他のプロセッサが受け持つ要素に関しては、通信にてデータを参照しなければならないため、プロセッサへのデータ割り当て方により通信量が異なり、並列処理効果の良否が決定されてしまう。分散メモリ型並列計算機では、並列処理効果を上げるために、プロセッサへのデータ割り当ても重要となる。

しかし一方、必ずしも並列処理効果が良くなるプロセッサへのデータ割り当てが望ましいとは限らない。これは、データをできるだけ均等に分割しプロセッサへ割り当てた方が、より大規模な問題を扱うことができるからである。分散メモリ型並列計算機上に実装するプログラムを考える際、より並列処理効果を上げるか、より大きい問題規模を扱えるようにするかの選択が問われる。

3. 並列化トランスレータ

開発中の並列化トランスレータは、図1に示すような構成からなる。以下に、各処理について述べる。特徴として、入力プログラムをブロックで区切り、そのブロックから最も計算がかかると予測されるメインブロックを決定し、プロセッサへのデータの割り当て方を決定している。ブロックの決定およびメインブロックの決定の方法は3章で説明する。

(1) ユーザによる入力

FORTRAN 言語で記述されたプログラムと最終的に生成される並列プログラムを実装するプロセッサ数を並列化トランスレータに与える。

(2) 構文解析部

入力された FORTRAN プログラムを、字句解析、構文解析により、並列化トランスレータの内部表現に変換する。内部表現により FORTRAN プログラムの構成を単純化し、後々の解析を行ないやすくする。

(3) データ依存解析部

内部表現より、データ間の依存関係を解析し、依存解析情報を生成する。

(4) ブロック設定部

内部表現をブロックに区切り、最も計算時間がかかると推測されるブロック（メインブロック）を設定する。

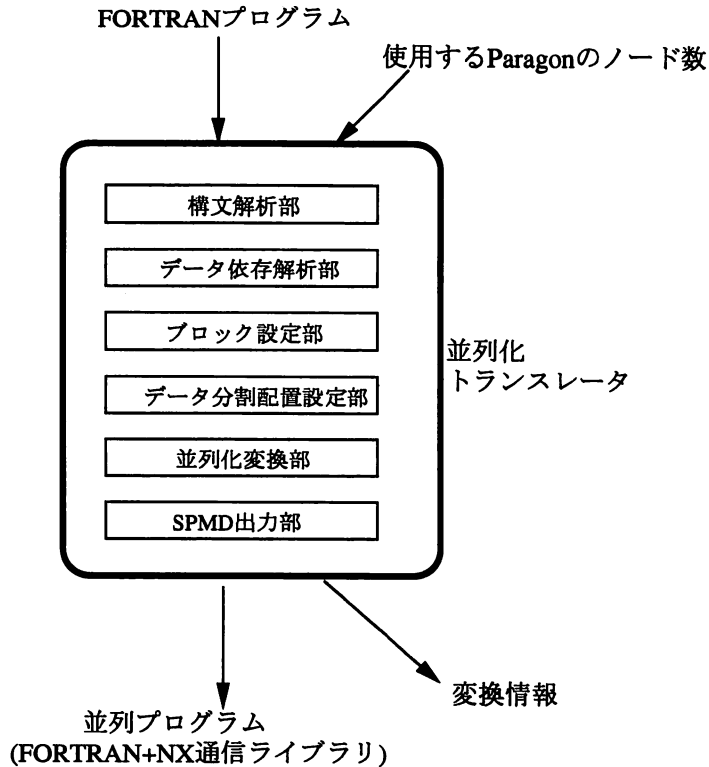


図1 並列化トランスレータの構成

(5) データ分割配置設定部

内部表現，プロセッサ数および依存解析情報より，各配列データの分割およびプロセッサへの割り当てを設定し，データ分割配置情報を生成する。設定には，メインブロックの並列処理効果が最適になるような各配列データの分割およびプロセッサへの割り当てを行なう。また，各配列の分割には，第1次元のブロック分割を用いる。

(6) 並列化処理部

データ依存解析情報，データ分割配置情報を用いて，中間表現を並列化変換する。ここで，並列化処理を行なう際，以下の主旨に従う。

- ループ (DO 文) は，DO 文のパラメータよりブロックに分割する。
- 出力 (WRITE 文) は，出力されるデータが割り当てられているプロセッサが実行する。
- 並列化トランスレータで並列化変換できないアルゴリズムについては変換を破棄する。

(7) SPMD コード出力部

並列化変換された内部表現を NX 通信関数を含む FORTRAN 言語で記述された SPMD コードに変換し，出力する。

(8) if77によるコンパイル

得られた並列化コードを Paragon 用のコンパイラ if77によりコンパイルし、実行形式を得る。

4. メインブロック

データ分割配置の設定を行なう要因として、メインブロックを決定する。メインブロックとは、プログラム中の最も計算時間がかかると推測するプログラムの一部分を示す。メインブロックの決定には先ず、並列化トランスレータの内部表現をネストループを含んだ1ループ、あるいは、ループの外に現れる演算式の集まりごとに区切っていく。この区切られた領域をブロックと呼ぶ。ブロックの例を図2に示す。

次に、各ブロックについて演算量を求める。この演算量は、 $+$ 、 $-$ 、 $*$ 、 $/$ の演算子および、ループのイタレーションの数から計算する。例えば、図2のブロックAならば、(イタレーションの数 $10 \times$ 演算子の数 2)より、演算数 20 が求まる。同様にブロックBならば演算数 4 、ブロックCならば演算数 18 が求まる。

最後に、求めた各ブロックの演算数を比較して、最も大きい値をメインブロックと設定する。図2では、ブロックAがメインブロックとなる。

5. 評価

並列化トランスレータの評価方法および評価結果について述べる。

5.1 評価方法

行列積を求めるプログラム（プログラムの一部を図3に示す）を実際に並列化トランスレータに入力し、変換結果を得る。1) 得られた変換結果から、配列の宣言によるメモリ

```

-----
                DO 100 I = 1, 10          ブロック A
                  A(I)=I*2
100    CONTINUE
-----

                B=C*3                    ブロック B
                D=C+2
-----

                DO 200 I = 2, 20         ブロック C
                  E(I)=10
200    CONTINUE
-----

```

図2 ブロック設定例

の使用について評価を行なう。また、2)変換結果を Paragon 上に実装し、速度向上率を求め、並列効果の評価を行なう。

5.2 評価結果

図3のプログラムとプロセッサ数4を並列化トランスレータに入力した結果、図4（出力結果の一部）を得た。また、行列の大きさが 256×256 および 512×512 のプログラムをそれぞれ、プロセッサ数2, 4, 8, 16, 32で入力し、出力結果および図3のプログラム（プロセッサ数1）を Paragon に実装し実行時間を計測した。その計測結果から図5のような速度向上率を得た。これらの結果から以下のことについて評価する。

(1) メモリ使用の評価

図3と図4を比較すると、配列Aの第1次元、配列Cの第1次元が変換された。これらの配列の大きさは、元の大きさをプロセッサ数で割った値となっており、配列A, Cに関してはプロセッサの増加とともに小さくなっていくことが確認できた。また、配列Bは分割配置されず、全てのプロセッサに全ての要素が割り当てられた。この配列Bの変換結果の評価として、手作業により配列Bを配列A, Cと同様に第1次元で分割したプログラムを作成し、速度向上率の評価と同様の評価を行なった結果、データの参照のため通信量が膨大になりあきらかに実行時間が増加した。このことから、配列Bの宣言は適切であったといえる。

(2) 速度向上率による評価

図5のような理想値に近い速度向上率を得た。出力結果（図4）より、データ通信が必要なく最外ループも均等に分割できるので理論上速度向上率は理想値となると

```

...
INTEGER A(16,16),B(16,16)
INTEGER C(16,16),S
...
DO 100 I=1,16
    DO 200 J=1,16
        S=0
        DO 300 K=1,16
            S=S+(A(I,K)*B(K,J))
300     CONTINUE
        C(I,J)=S
200     CONTINUE
100    CONTINUE

```

...
図3 行列の積を求めるプログラム

予測できるが、多少プロセッサの増加とともに速度向上率が落ち込んだ。これは、実行時間に対して時間計測の際に行なうバリア同期にかかる時間の影響が現れたと考えられる。

6. ま と め

本研究では、FORTRAN プログラムからユーザの指示なく並列プログラムを得る並列化トランスレータの開発を行なっている。この並列化トランスレータは、メインブロック

```

...
INTEGER A(4,16),B(16,16)
INTEGER C(4,16),S
...
DO 100 I=1,4
  DO 200 J=1,16
    S=0
    DO 300 K=1,16
      S=S+(A(I,K)*B(K,J))
300    CONTINUE
      C(I,J)=S
200    CONTINUE
100  CONTINUE
...

```

図4 出力結果

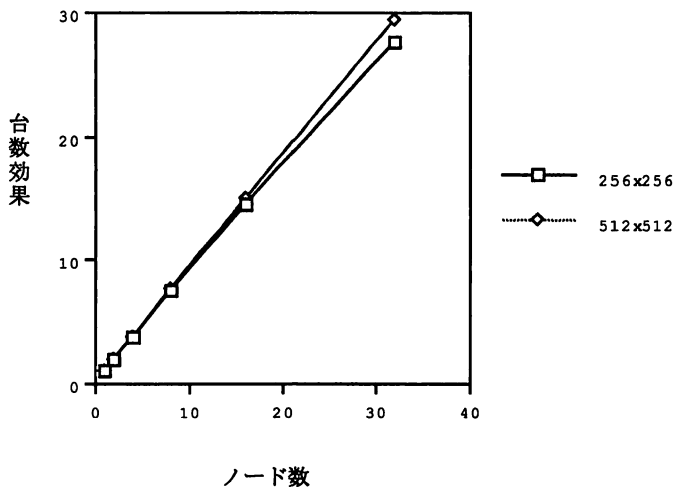


図5 速度向上率

によりプロセッサのデータ割り当てを行なっている。

本稿では、この並列化トランスレータの構成、メインブロックの決定方法、および、行列の積を求めるプログラムによる評価について述べた。評価結果より、行列の積のプログラムにおける並列化トランスレータでの有効性を示した。

問題点としては、並列化可能なアルゴリズムが行列積のような単純なアルゴリズムに限定されていること、Paragonに限定しているため、汎用性が少ないことである。今後の課題として、さまざまな数値計算における評価から並列化トランスレータの並列化処理の拡張、およびメインブロックの評価、他の通信関数への対応の検討を考えている。

参考文献

- 1) 忠政慎也：“自動並列化に関する研究”，岡山理科大学情報処理センター研究員報告，17号，pp.87-92 (1996)。
- 2) 忠政慎也，小畑正貴：“分散メモリ型並列計算機を対象とする並列化トランスレータ”，情報処理学会研究報告，96-PRO-10，Vol. 96，No. 107，pp. 1-6 (1996)。
- 3) インテルジャパン(株)：“パラゴンプログラミングガイド”，(1997)。

A Parallelizing Translator for Distributed Memory Machines

Shinya TADAMASA* · Masaki KOHATA**

**Graduate School of Engineering,*

***Department of Information & Computer Engineering*

Faculty of Engineering,

Okayama University of Science

Ridai-cho 1-1, Okayama 700-0005, Japan

(Received October 6, 1997)

In large scale numerical analysis, high speed computation by parallel processing is expected and various subjects have been studied. There are many difficult things in description of parallel programs, such as understanding parallelizable part and insertion of communication function.

We are developing a parallelizing translator which translates sequential programs to parallel programs for distributed memory machine. This paper presents a scheme of translation and its implementation and its evaluation.