

分散メモリ型並列計算機用自動並列化コンパイラ

5G-1

V-Pascal/DM

梅田 憲

上原 哲太郎

津田 孝夫

京都大学工学部情報工学教室

1 はじめに

分散メモリ型並列計算機用のプログラム開発には、アルゴリズムの並列化とともに同期・通信文を用いた並列プログラムの記述が必要になる。本稿では、データパラレルアプローチに基づいた分散メモリ型並列計算機用自動並列化コンパイラ V-Pascal/DM の開発について述べる。本コンパイラは当研究室で開発中の自動ベクトル化コンパイラ V-Pascal[1] を拡張することで実現される。V-Pascal/DM は自動ベクトル化コンパイラの技法を継承し、高度な自動並列化処理を実現する。プログラム中の大規模配列は、コメントとして記述された指示に従って分割される。データの分割に従って、コンパイラがループの並列化、同期・通信文の挿入を行なう。

2 V-Pascal/DM

V-Pascal/DM は現在、富士通 AP1000 を対象計算機として開発されている。AP1000 はユーザに各計算機間の同期・通信を実現するライブラリを提供している。V-Pascal/DM もこのライブラリを用いることで AP1000 の同期・通信機構を利用する。

V-Pascal/DM の処理は 3 フェーズからなる。フェーズ 1 ではデータ分割等のユーザ指示文を含む Pascal プログラムから、ホスト計算機（以降 host）用中間プログラム、セル計算機（以降 cell）用中間プログラムを生成する。フェーズ 2 では host 中間プログラムに対してスカラ最適化処理を、cell 中間プログラムに対して依存解析処理と並列化処理を行う。フェーズ 3 は AP1000 の通信ライブラリコールを含む C プログラムを生成する。生成された C プログラムを AP1000 の C コンパイラで処理することで実行プログラムが得られる。

生成された host プログラム、cell プログラムは次の手順で実行される。(1)host 上で host プログラムを起動。(2)host プログラムはファイル/端末からデータを入力。(3)host プログラムが、cell プログラムを起動し、データを送信。(5)cell 上で逐次・並列演算を実行。(6)cell プログラムから host プログラムにデータを送信。(7)host プログラムはファイル/端末にデータを出力。

V-Pascal/DM, an Automatic Parallelizing Compiler for Distributed Memory Machines
Ken UMEDA, Tetsutaro UEHARA, Takao TSUDA
Kyoto University

3 データの分割・分散実行ブロックの指定

大規模配列の分割はプログラム中にコメントとして記述する。プログラム中の array 型変数について静的な分割を指定することができる。array 型変数を分割するためには、変数の各次元について、block, cycle(scatter), 非分割の指定を行なう。分割は 1 変数につき最大 2 次元まで指定できる。各次元の分割数は、cell の実台数を意味する定数 ncel, ncelx, ncely を含む定数式によって指定する。図 1 に例を示す。分割した配列は 1cell に 1 分割ずつ割り付けられる。

```
var a,b,c : array[1..3000,1..3000] of real
          (* distribute[block(ncelx),block(ncely)] *) ;
```

図 1: 分割指定の例

ソースプログラム中で cell によって実行される部分を分散実行ブロックと呼び、プログラム中にコメントにより指定する。分散実行ブロック内では I/O 処理は禁止される。分散実行ブロック内から呼び出される手続き・関数もまた cell によって実行される。分割変数を実引数として呼び出す場合、対応する仮引数は実引数と同様の分割を指定されたものとして扱う。呼び出しが複数存在し、異なる分割変数が実引数として記述されているとき、手続き・関数を複製することで、仮引数の分割を静的に決定する。

4 host プログラム

host プログラムは、ソースプログラムにスカラ最適化処理を施して生成される逐次プログラムである。ただしソースプログラム中の分散実行ブロックに相当する部分は、cell プログラム起動ライブラリを呼び出すように変更される。cell プログラム起動ライブラリは、まず分割変数の分割数から必要な cell 構成を決定し、その構成に基づいて cell プログラムを起動する。次に cell プログラム中で引用される非分割変数、分割変数の値を送信する。その後 cell プログラム中の演算の終了を待ち、cell プログラムから変数の値を受信する。

5 cell プログラム

5.1 分散メモリ上での逐次実行

各々の cell は、cell プログラムのコピーを実行する。フェーズ 2 モジュールは、cell プログラムが逐次実行される時全ての cell 上で同時に演算が実行されるよう

にプログラムを変換する。全ての cell は、非分割変数のコピーをローカルメモリに保持しているの、cell プログラム中の非分割変数の引用はローカルメモリ中の値を引用するよう変換される。cell プログラム中の分割変数の要素の引用は、その要素を保持している cell がその値を放送し、他の cell が受信するよう変換される。従って逐次実行部分では、全ての cell が同時に同じ演算を行ない、同じ結果を得る。演算により得た値を非分割変数に定義するときは全ての cell がストアを行ない、分割変数に定義するときはその要素を保持する cell のみがストアを行なう。cell プログラムの逐次実行部分では、全ての cell が同じ演算結果に基づいて分岐命令を実行するので、等しい実行パスをとる。

5.2 並列化処理

フェーズ 2 の並列化処理では、cell プログラム中の次の条件を満たすループを対象とする。(a) 繰り返し実行間に依存がない。(b) ループ中に関数の呼出がない。(c) ループ中の分割変数の参照は全て、分割されている次元の添字式がループ制御変数の線形式になっている。

代入文の演算は、左辺項を保持する cell が実行する。代入文を含むループは次のような並列プログラムに変換される。(1) 各 cell はループの繰り返し空間から、添字式と分割指定、ループ上下限をもとに、保持する右辺項を引用する繰り返し空間のみを抽出。(2) 抽出した繰り返し空間について、右辺項の値を送出先毎に分けて packet に詰める。(3) packet を送出。(4) ループの繰り返し空間から、保持する左辺項を定義する繰り返し空間のみを抽出。(5) packet を受信。(6) 抽出した繰り返し空間について、演算と結果のストアを行なう。

フェーズ 2 モジュールは上に述べた並列化処理を最内側ループから施していく。送出される項がより外側のループについて不変ならば、外側のループについても並列化することができる。最内側二重ループが並列化される例を図 2 に示す。

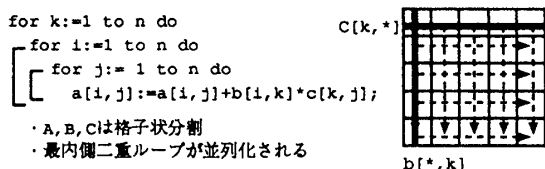


図 2: 外積法による行列積プログラムの並列化

5.3 マクロ演算

各繰り返し実行間に依存があるループは並列化することができない。ただしそのような場合でも、ループ中の演算が最大値検出、総和演算、一次回帰演算等の演算を意味しているときは、並列アルゴリズムによる演算に置き換え、並列化を実現する。これらの演算の検出は V-Pascal において開発・実装されたイディオム認識処理 [2] により行なう。

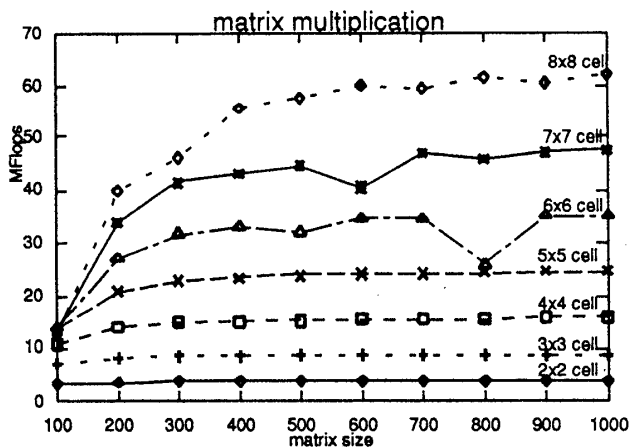


図 3: 行列積演算における MFLOPS 値

6 評価

現在実装が行なわれているフェーズ 3、C コード生成モジュールを用いて性能評価を行なった。

6.1 行列積演算

列方向、行方向について block 分割した正方行列 A, B, C について、 $A = B \times C$ の行列積を素朴な外積法アルゴリズムを用いて求める。行列のサイズと cell 台数を変化させたときの MFLOPS 値を図 3 に示す。

6.2 LINPACK ベンチマーク

LINPACK ベンチマークの結果を表 1 に示す。行列は行方向に cycle 分割した。表 1 中の「時間 2」「比較」の欄は、AP1000 用に最適化を施した BLAS3 を用いた LINPACK の実行時間 (文献 [3])、及び比較である。

表 1: Linpack ベンチマーク n=1000 (倍精度)

台数	時間 (s)	MFlops	台数効果	時間 2(s)	比較
1	472.21	1.417	1	160	33.9%
2	318.29	2.102	1.48	81.4	25.57%
4	165.36	4.047	2.86	41.3	24.98%
8	88.84	7.53	5.32	22.6	25.43%
16	50.49	13.25	9.35	11.5	22.77%
32	30.90	21.65	15.28	6.71	21.72%
64	20.49	32.67	23.05	3.51	17.13%

7 おわりに

現在、V-Pascal/DM は自動ベクトル化コンパイラ V-Pascal の multi processor version とともに開発・実装中である。

本測定では富士通研究所 (株) の AP1000 を利用させて頂きました。

参考文献

[1] Takao Tsuda and Yoshitoshi Kunieda. V-pascal: an automatic vectorizing compiler for pascal with no language extensions. *The Journal of SUPERCOMPUTING* (Kluwer Academic Publisher), Vol. 4, pp. 251-275, 1990.

[2] 中村素典, 津田孝夫. 自動ベクトル化コンパイラにおけるイディオム認識法. 情報処理学会論文誌, Vol. 32, No. 4, pp. 491-503, April 1991.

[3] Richard P. Brent and Peter E. Strazdins. Implementation of the BLAS Level 3 and LINPACK Benchmark on the AP1000. *FUJITSU SCIENTIFIC & TECHNICAL JOURNAL*, Vol. 29, No. 1, pp. 61-70, 1993.