

バス結合マルチプロセッサ型ベクトル計算機における
線形計算アルゴリズムの評価

4 L-2

岡部 寿男 王 蘇煙 津田 孝夫
京都大学 工学部

1 はじめに

共有メモリマルチプロセッサ型計算機において、演算性能に比べ主記憶からプロセッサへのデータ供給能力が不十分な場合、並列化による性能向上を得るためにはアルゴリズム上の工夫が必要である。本発表では、行列積のいくつかのアルゴリズムについてデータ転送量の比較を行ない、行ブロック *jki* 型のアルゴリズムをアンローリングしたものが最適であることを示す。さらに、4CPU のバス結合マルチプロセッサ型ベクトル計算機 TITAN 3000 上での性能実測によりこれを確認する。

2 TITAN のメモリ・アーキテクチャ

TITAN 3000 の最大構成は、4枚の CPU ボードと2枚のメモリボードが2本のバスにより結合されたアーキテクチャを持つ[1]。各 CPU ボードはインテジャーユニット (32MIPS) とベクトルユニット (32MFLOPS) からなり、ベクトルユニットはキャッシュを介さずに主記憶と直接データをやりとりする。

バスは R-BUS (リード専用) と S-BUS (リード/ライト) の2本であり、それぞれ 128MB/sec のバンド幅をもつ。インテジャーユニットは S-BUS を介して読み書きするのに対し、ベクトルユニットは R-BUS を読み出し専用、S-BUS を書き込み専用として用いる。ベクトル演算性能 (128MFLOPS) に比べ、R-BUS (32Mword/sec) の能力が不十分なので、データ供給能力の不足により演算器の性能が十分に発揮されないことが簡単に生ずる。

3 行列積アルゴリズムの比較

行列積を計算する3重ループ (Livermore FORTRAN Kernel 21)

```
REAL*8 PX(MDIM,*), VY(MDIM,*), CX(MDIM,*)
DO 21 I=1, N
DO 21 K=1, N
DO 21 J=1, N
  PX(I,J)=PX(I,J)+VY(I,K)*CX(K,J)
21 CONTINUE
```

の添字 I, K, J を入れ換えた6つのプログラムを考える。最内側の添字が I, J のループについては単純にベクトル化可能、K のループについては内積演算を使ったベクトル化が可能である。並列化は、添字が I, J のループについて可能である。TITAN 3000 の自動ベクトル化/並列化 FORTRAN 77 コンパイラ (Arden Fortran ver. 18.7) は、ベクトル化に最適なループを示すコンパイラ指示行 (C\$DOIT VBEST) を適宜挿入することにより、良く知られた以下の8つの型のアルゴリズムに相当する目的コードを生成可能である。

内積法 *ijk* 型, *jik* 型

最内側添字 K について内積演算の形でベクトル化される。また、最外側ループで並列化が可能である。

$$ijk \text{ 型 } \begin{bmatrix} i & \times & \times & \cdots & \times \end{bmatrix} \leftarrow \begin{bmatrix} \longrightarrow \end{bmatrix} \begin{bmatrix} \updownarrow & \updownarrow & \cdots & \updownarrow \end{bmatrix}$$

このアルゴリズムのロードストアのコストは、 $2N^3$ vector load、 N^2 scalar store である (それぞれ最高次の項以外は無視した)。最内側ループでのメモリ参照は行方向と列方向が1回づつであり、また、並列化した際の並列性粒度 (並列処理単位中の演算数) は $2N^2$ である。

中間積法 *ikj* 型, *jki* 型

最内側添字 K についてベクトル化され、最外側ループで並列化が可能である。

$$jki \text{ 型 } \begin{bmatrix} j & \updownarrow \end{bmatrix} \leftarrow \begin{bmatrix} \updownarrow & \cdots & \updownarrow \end{bmatrix} \begin{bmatrix} j & \times & \times & \cdots & \times \end{bmatrix}$$

ロードストアのコストは $2N^3$ vector load、 N^3 vector store、 N^2 scalar load であり、並列性粒度は $2N^2$ である。

外積法 *kij* 型, *kji* 型

ロードストアのコストは、中間積法と同じである。並列性粒度は $2N$ である。

行ブロック *jki* 型

jki 型の I のループをベクトル長で区切って二重化 (iv と I) し、iv と K を交換した形である。ロード/ストアのコストは N^3 vector load、 N^2 vector store、 $N^3/32$ scalar load と小さい。

```
DO PARALLEL J=1, N
DO iv=1, N, 32
  rv = MIN(N, 31 + iv)
  DO K=1, N
    DO VECTOR I=iv, rv
      PX(I, J) = PX(I, J) + VY(I, K) * CX(K, J)
    END DO
  END DO
END DO
```

$$\begin{bmatrix} j & \updownarrow \\ i & \end{bmatrix} \leftarrow \begin{bmatrix} \updownarrow & \cdots & \updownarrow \end{bmatrix} \begin{bmatrix} j & \times & \times & \cdots & \times \end{bmatrix}$$

行ブロック kji 型

kji 型の I のループをベクトル長で区切って二重化し、 iv と J を交換した形である。ロードのコストは行ブロック jki 型と同じであるが、ストアのコストは N^3 vector store と大きい。また並列性粒度は 2×32 である。

$$\begin{bmatrix} \uparrow \cdots \uparrow \\ \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \uparrow \\ \vdots \end{bmatrix} \times \cdots \times \begin{bmatrix} \uparrow \\ \vdots \end{bmatrix} \quad k$$

性能比較

それぞれのプログラムを実際に実行させ性能を測定した。 $N = 500$ とし、バンクコンフリクトの影響を調べるため $NDIM = 1001$ と $NDIM = 1024$ についてベクトル実行と並列ベクトル実行の速度を測定した。測定結果を表 1 に示す。

2つの $N \times N$ 行列の積の総演算量は $2N^3$ であるから、TITAN 3000 の場合、ベクトルロードのコストが例えば $2N^3 \text{ words}$ の場合、R-BUS のデータ供給能力の性能により、

$$\frac{2N^3}{\frac{8 \times 2N^3}{128 \times 10^3}} = 16 \text{ MFLOPS}$$

という理論性能限界が導かれる。測定結果は、特に並列ベクトル実行時に R-BUS のデータ供給能力がボトルネックになって性能が抑えられていることを示している。TITAN 3000 上での行列積アルゴリズムとしては、ベクトルロードのコストの小さい行ブロック jki 型が最も優れているといえる。

4 ループアンローリングによる高速化

行ブロック jki 型のアルゴリズムは、ループアンローリングを適用することにより、ロードの回数を減らすことが可能である。例えば以下のように 4 重にアンローリングすると、ロードのコストは $1/4$ になる。

```

DO 21 J=1,M-(4-1),4
DO 22 I=1,M
DO 23 K=1,M
  PX(I,J)=PX(I,J)+VY(I,K)*CX(K,J)
  PX(I,J+1)=PX(I,J+1)+VY(I,K)*CX(K,J+1)
  PX(I,J+2)=PX(I,J+2)+VY(I,K)*CX(K,J+2)
  PX(I,J+3)=PX(I,J+3)+VY(I,K)*CX(K,J+3)
23 CONTINUE
22 CONTINUE
21 CONTINUE

```

TITAN 3000 でアンローリングを適用したプログラムを性能測定した結果を表 2 に示す。アンローリングのファクタを増やすにしたがってベクトルロードのコストが減少し、最大性能 (128 MFLOPS) に近い性能が得られることがわかる。

表 2: アンローリングの効果

unrolling	ベクトル実行	並列ベクトル実行
1	18.55	31.95
2	28.03	62.94
3	28.56	87.99
4	28.67	108.46
5	26.51	104.37
6	26.91	112.54
7	24.66	91.67
8	24.83	92.18

(単位は MFLOPS)

さらに、この行ブロック jki 型行列積サブルーチンを LAPACK に組み込むと、 2000×2000 の密行列の LU 分解が、4 重アンローリング、ブロック幅 32 の場合 68.373 MFLOPS の性能で行なえる。

5 おわりに

バス結合マルチプロセッサ型計算機においては、多重ループにわたるループ交換とループアンローリングが有効であることが分かった。現在我々の研究室で開発中の自動ベクトル化/並列化コンパイラにこれらの機能を組み込むべく研究中である。なお、本研究は一部文部省科学研究費補助金による。

参考文献

- [1] D. P. Siewiorek and P. J. Koopman, Jr.: "The Architecture of of Supercomputers — TITAN, a case study —", Academic Press (1991).

表 1: 行列積アルゴリズムの比較

algorithm	vector load	vector store	scalar load/store	ベクトル実行 (MFLOPS)		並列性粒度	並列ベクトル実行 (MFLOPS)	
				NDIM=1001	NDIM=1024		NDIM=1001	NDIM=1024
ijk	N^3 (行方向) N^3 (列方向)	0	N^2	12.64	4.66	$2N^2$	16.26	14.17
jik	N^3 (行方向) N^3 (列方向)	0	N^2	12.86	4.70	$2N^2$	16.33	13.47
ikj	$2N^3$ (行方向)	N^3 (行方向)	N^2	14.39	2.76	$2N^2$	16.04	4.74
kji	$2N^3$ (列方向)	N^3 (列方向)	N^2	9.56	9.55	$2N^2$	16.06	16.13
kij	$2N^3$ (行方向)	N^3 (行方向)	N^2	13.77	2.74	$2N$	16.30	9.19
kji	$2N^3$ (列方向)	N^3 (列方向)	N^2	9.36	9.13	$2N$	16.14	16.24
行ブロック jki	N^3 (列方向)	N^2 (列方向)	$N^3/32$	18.55	18.96	$2N^2$	31.91	31.96
行ブロック kji	N^3 (列方向)	N^3 (列方向)	$N^3/32$	15.99	16.00	2×32	13.17	13.18