

中村絹代* 安藤憲行** 西 直樹**

Parallelization Techniques for
Highly Parallel Vector Computer

by

Kinuyo NAKAMURA

National Aerospace Laboratory

Noriyuki ANDO and Naoki NISHI

NEC Corporation

ABSTRACT

We concern with the assumed highly parallel vector computer with hierarchical memory that consists of many vector computers with local memory and one common memory. To speedup parallel processing of CFD (Computational Fluid Dynamics) programs on that parallel computer, it needs parallelization techniques such as enlarging parallel portion, equal work load, enlarging task granularity and minimizing volume of data access between local memory and common memory. This paper describes about high efficiency of parallel processing of that parallel computer obtained by applying parallelization techniques to a CFD program.

1 はじめに

科学技術用高速計算機として発展してきたベクトル計算機はより一層の高速処理の実現のため、それを要素計算機として並列化する方式のものに移行している。

本報告では高並列ベクトル計算機としてローカルメモリ(LM)及び共有メモリ(CM)をもつ階層メモリ型並列計算機(図1)を想定して、CFDプログラムの並列処理による処理時間を予測し、有効性の評価を行った。

評価の対象としたCFDプログラムは3次元薄層近似ナビエ・ストークス方程式をIAF解法で解く

ものである。このプログラムではプログラム記述において逐次処理しなければならない軸方向が計算空間の3方向に交代するため、並列処理において計算空間の分割方向の変更が必要になり、多量のデータ転送が発生する。これを回避するため、並列化プログラム技術を駆使してプログラムを再構成し、並列処理時間による評価を行った。

2 並列計算機モデルとCFDプログラム

2.1 並列計算機モデル

想定した高並列ベクトル計算機の要素計算機(PE)はNEC SX-2相当とし、その台数を最大128台とした。PEとCM間のスループットはPE1台当たり100MB/秒とし、PE台数が128台では12.8GB/秒と仮定した。

* 航空宇宙技術研究所

** 日本電気㈱

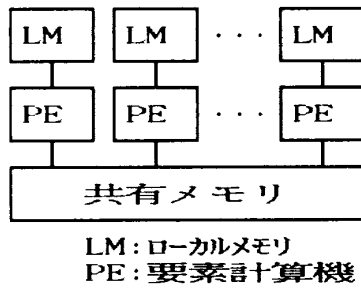


図1 階層メモリ型並列計算機

2.2 並列計算機のプログラム高速化技術

階層メモリ型並列計算機におけるプログラムの高速化のためにはベクトル化技術の他に、

- ① 並列化率の拡大
 - ② PEのワークロードの均等化
 - ③ 並列化オーバーヘッドの低減化
 - ④ CMとLM間のデータ転送の頻度と量の低減化
- 等の並列化技術が必要である。

①はプログラムの処理において並列処理可能な部分の割合を大きくすることである。並列化率が100%に近ければ、PEの台数倍に近い高速化倍率を得ることができる。

②は並列処理を行う際のPEに与える仕事量を均等にすることである。各PEの仕事量が不均等であれば最も処理に時間のかかるPEの終了時間迄、他のPEは待たされるため処理時間の短縮が望めない。

③は並列化に伴って生ずるオペレーティングシステムのオーバーヘッド等の時間を少なくすることである。これを回避するためには並列処理を行う際の処理単位をできるだけ大きくする必要がある。

④はデータ転送時間を削減することである。プログラム処理において用いられる変数はローカル変数と共有変数に分けられる。ローカル変数はLMに配置され、そのLMをもつPEでのみ定義・参照・更新され、他のPEからはこれらの処理はできない。また、共有変数はCMに配置され、どのPEからも参照・更新が可能である。PEはLM上のデータを用いて演算を行うので、CM上のデータはLMにデータ転送をしなければ演算処理できないため、CMとLMの間でデータ転送が発生する。

2.3 階層メモリ型並列計算機における

プログラムの処理

CFDプログラムの階層メモリ型並列計算機における処理の理解を容易にするため、ここで階層メモリ型並列計算機におけるプログラムの処理について説明する。なお、コンパイラはDOループにまたがる最適化を行わないものとし、DOループ処理終了後、同期をとるものとする。以下はプログラム例である。

```

DO 1 K=1,128
DO 1 J=1,128
DO 1 I=1,1024
C(I,J)=A(I,J,K)+B(I,J,K)
:
1 E(I,J,K)=C(I,J)**2+F(I,J,K)
DO 2 K=1,128
DO 2 J=1,128
DO 2 I=1,1024
2 G(I,J,K)=E(I,J,K+1)-2*E(I,J,K)
*      +E(I,J,K-1)
DO 3 K=1,128
DO 3 J=1,128
DO 3 I=1,1024
3 G(I,J,K)=0.5*G(I,J,K-1)+0.5*F(I,J,K)

```

これらのDOループを並列処理するには、各PEはDOループの並列処理の前に参照する必要のある共有変数をCMより転送し、LM内にそのコピーを作る。DOループの並列処理が終了すると更新された共有変数のみをCMに書き戻す。

DO1ループではDO変数*i, j, k*の内いずれを用いても並列化可能であるが、配列データの連続性を考慮して、*i*でベクトル化し、*k*で並列化する。各PEは*k*で分割されたある空間の配列を対象に演算処理することになる。分担する*k*の値が1つであれば、1つの(*i, j*)平面に対してPEは処理を行うことになる。

配列A, B, E, F及びGは共有変数としてCMに置かれる。また、配列CはPE内でのみ用いられる配列であり、ローカル変数となる。便宜的に128台のPEにこれらの計算を均等に割り当てるとすると、DO1ループ処理前に共有変数がCMからPEのLMへ次

のようにデータ転送される。

PE 1 ← A(I, J, 1), B(I, J, 1), F(I, J, 1)

:

PE128 ← A(I, J, 128), B(I, J, 128), F(I, J, 128)

DO 1 ループ処理後に更新の必要のある配列 E のみ LM から CM へデータ転送される。

E(I, J, 1) → CM

:

E(I, J, 128) → CM

DO 1 ループ処理終了後、同期がとられる。

DO 2 ループでは共有変数の配列 E が k+1, k-1 及び k の値を用いるので k で並列化するとそれぞれの PE において配列 E の k+1, k-1, k の 3 つの (i, j) 平面のデータ参照が必要となり転送量が増大する。並列化を j で行うことにより並列台数を 128 台とすれば PE 当たり 1 つの (i, k) 平面の転送となる。

DO 3 ループは DO 変数 k にデータ依存関係があるため、j で並列化、i でベクトル化すると良い。

2. 4 評価対象 CFD プログラム^{1) 2)}

3次元薄層近似ナビエ・ストークス方程式の IAF 解法は次の形式の差分方程式、

$A_{\xi} A_{\eta} A_{\zeta} \Delta Q^{n+1} = (L_{\xi} + L_{\eta} + L_{\zeta}) Q^n$ を解き、 ΔQ^{n+1} を求める操作を収束するまで行う。計算手順は以下の①～⑦の繰り返しである。

- ① $R^n = L_{\xi} Q^n$
- ② $R^n = R^n + L_{\eta} Q^n$
- ③ $R^n = R^n + L_{\zeta} Q^n$
- ④ $A_{\xi} x = R^n$
- ⑤ $A_{\eta} y = x$
- ⑥ $A_{\zeta} \Delta Q^{n+1} = y$
- ⑦ $Q^{n+1} = Q^n + \Delta Q^{n+1}$

3 CFD プログラム高度並列化のためのプログラム再構成

3.1 オリジナルプログラム構造及び処理方法

(1) オリジナルプログラム構造

オリジナルプログラムの構造と 2.4 節①～⑦の処理との関係を図 2 に示す。

各サブルーチンは全て 1 個の 3 重 DO ループからできている。DO ループ構成を表 1 に示す。これら

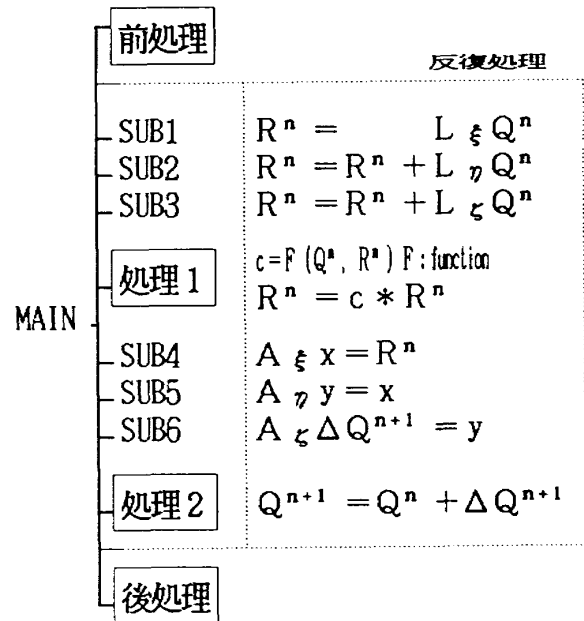


図 2 プログラム構造

表 1 サブルーチンの 3 重 DO ループ構成

サブルーチン	SUB1	SUB2	SUB3	SUB4	SUB5	SUB6
第 3 重 (最外)	K	K	J	K	K	J
第 2 重 (中)	I	J	K	I	J	K
第 1 重 (最内)	J	I	I	J	I	I

全ての DO ループにおいては、最内及び最外 DO ループの DO 変数に関して並列化可能である。このプログラムでは第 2 重目の DO ループの DO 変数に関して逐次処理となっており、第 2 重目の DO ループの DO 変数では並列化できない。

図 2 の処理 1 は配列 Q, R などから求められるスカラー係数 c を配列 Rⁿ に乗ずるものである。

処理 2 は主に配列 Q の更新を行うものである。Q の更新は内点の更新及び境界点の更新の 2 種類に分けられる。

表 2 にサブルーチンに対する配列の入出力量を示す。各サブルーチンにおいて入力とあるのは参照配列名及びデータ量であり、出力は更新配列名及びデータ量である。N (=I₀ × J₀ × K₀) は 1 つの配列変数の大きさを示す。I₀、J₀、K₀ はそれぞれ格子点数を表す。

(2) オリジナルプログラムの処理方法

オリジナルプログラムを階層メモリ型並列計算

表2 サブルーチンに対する配列の入出力量

サブルーチン	入力		出力	
	配列名	量	配列名	量
SUB1	Q (I ₀ , J ₀ , K ₀ , 5) RM (I ₀ , J ₀ , K ₀ , 3, 3)	5N 9N	R (I ₀ , J ₀ , K ₀ , 5) E (I ₀)	5N 1 ₀
SUB2 SUB3	Q (I ₀ , J ₀ , K ₀ , 5) RM (I ₀ , J ₀ , K ₀ , 3, 3) R (I ₀ , J ₀ , K ₀ , 5) E (I ₀)	5N 9N 5N 1 ₀	R (I ₀ , J ₀ , K ₀ , 5) E (I ₀)	5N 1 ₀
処理1	Q (I ₀ , J ₀ , K ₀ , 1) R (I ₀ , J ₀ , K ₀ , 5) E (I ₀)	N 5N 1 ₀	R (I ₀ , J ₀ , K ₀ , 5)	5N
SUB4 SUB5 SUB6	Q (I ₀ , J ₀ , K ₀ , 5) RM (I ₀ , J ₀ , K ₀ , 3, 3) R (I ₀ , J ₀ , K ₀ , 5)	5N 9N 5N	R (I ₀ , J ₀ , K ₀ , 5)	5N
処理2	Q (I ₀ , J ₀ , K ₀ , 5) R (I ₀ , J ₀ , K ₀ , 5) RJ (I ₀ , J ₀ , K ₀)	5N 5N N	Q (I ₀ , J ₀ , K ₀ , 5) RESMX	5N 1

機で処理する場合、その構造を考慮して3重D0ループの最内D0ループのD0変数によりベクトル化し、最外D0ループのD0変数により計算空間を分割し、PEに割り当てる。これにより2.3節で示した方針に基づいてプログラム処理が行われる。この時、PE台数をnとするとSUB1, SUB2, SUB4, SUB5においては各PEはK₀/n個の(i, j)平面を担当し並列処理を行うことになる。また、SUB3及びSUB6では各PEはJ₀/n個の(i, k)平面を担当することになる。説明の便宜上、I₀、J₀、K₀はnで割り切れるものとする。

表2に示す配列は共有変数となる。処理1及び処理2は1台のPEに共有変数Q, Rなどの全データを集めて処理を行う。

3.2 再構成プログラム

プログラムの並列計算機における高速化を図るためプログラム構造を以下の通り変更する。

① LMに転送される共有変数の効率的利用を図るため、SUB3の多重D0ループをD0変数kで並列化する。これにより共有変数のLMとCM間のデータ転送の量と頻度を削減できる。その変更内容は以下の通りである。

SUB3はプログラム構造上kで逐次処理となっているがプログラムを書き直すことによりkで並列処理可能である。そのためには、担当計算空間の外にkに関してその前後2つの空間、即ち、4平面分の配列データを保持する必要がある。従って1PE当たり配列のK₀/n+4個の(i, j)平面の共有変

表3 オリジナルプログラムと再構成プログラムの構造の比較

	オリジナルプログラム	再構成プログラム
転送 SUB 1 転送	(K ₀ /n)Nij 並列処理 (K ₀ /n)Nij	(K ₀ /n+4)Nij 並列処理 —
転送 SUB 2 転送	(K ₀ /n)Nij 並列処理 (K ₀ /n)Nij	— 並列処理 —
転送 SUB 3 転送	(J ₀ /n)Nik 並列処理 (J ₀ /n)Nik	— 並列処理 —
転送 処理 1 転送	6N 逐次処理 5N	n 逐次処理 1
転送 SUB 4 転送	(K ₀ /n)Nij 並列処理 (K ₀ /n)Nij	— 並列処理 —
転送 SUB 5 転送	(K ₀ /n)Nij 並列処理 (K ₀ /n)Nij	— 並列処理 (K ₀ /n)Nij
転送 SUB 6 転送	(J ₀ /n)Nik 並列処理 (J ₀ /n)Nik	(J ₀ /n)Nik 並列処理 —
転送 処理 2 転送	11N 逐次処理 5N	— 並列処理 (J ₀ /n)Nik

数のデータをLMに転送する必要が生じる。これによりSUB1~SUB5までは全てD0変数kで並列処理可能となり、各PEの担当計算空間を決めてしまえばSUB1~SUB5までは並列処理するD0変数kを変えることなく並列処理が可能となり、サブルーチンの前後に行われるデータ転送の量と頻度を削減することができる。但し、係数cの計算及び各PEへの係数cの通知のため、処理1の前後で同期がとられる。

② MAINプログラム内の小規模なD0ループを並列処理可能となるようにプログラムを書き直す。これにより共有変数のLMとCM間のデータ転送量の削減及び並列処理による処理時間の短縮を図ることができる。その変更内容は以下の通りである。

処理1においてはスカラー係数cを求めるための中間結果を各PE毎にそのPEのもっている共有変数QやRのデータから求めさせ、全ての中間結果を1台のPEに集め、最終的な係数cを求める。求められたスカラー係数cを各PEに通知し、PE毎に配列Rを更新するようにプログラムを書き換える。

また、処理2においては配列Qの更新をSUB6においてΔQが求められた段階で各PEが行うように変更する。

表3にオリジナルプログラムと再構成プログラムのPEの処理方法とデータ転送の比較を示す。処

表4 必要なローカルメモリ容量 (単位 MB)

	PE 16台			PE 32台			PE 64台			PE 128台		
	C	L	合計	C	L	合計	C	L	合計	C	L	合計
オリジナルプログラム SUB1,2,4,5	1216	100	1308	608	100	708	304	100	404	152	100	252
オリジナルプログラム SUB3,6	1216	50	1266	608	50	658	304	50	354	152	50	202
再構成プログラム SUB1,2,3,4,5	1520	100	1620	912	100	1012	608	100	708	456	100	556
再構成プログラム SUB6	1216	50	1266	608	50	658	304	50	354	152	50	202

C:共有変数のための容量, L:ローカル変数のための容量

理方法は並列処理/逐次処理の別を示す。データ転送における $N_{i,j}$ 、 $N_{i,k}$ はそれぞれ (i,j) 平面、 (i,k) 平面の大きさであり、1つの配列の転送平面の大きさを示す。

4 階層メモリ型並列計算機の

CFDプログラムの処理性能予測

4.1 評価法

3章で述べたオリジナルプログラムと再構成プログラムの階層メモリ型並列計算機における処理時間により、CFDプログラムの高速化効果を考察する。格子点数は $I_o \times J_o \times K_o = 1024 \times 512 \times 256$ とした。

PEの処理時間は適当な規模の配列寸法をもつ問題をSX-2により処理し、その処理時間を基に算出した。CM及びLMは無制限に使用できるものとし、CMとLM間とのデータ転送時間はシミュレーションにより予測した。PEの並列台数は16台、32台、64台、128台とした。

4.2 ローカルメモリ容量の評価

本CFDプログラムの並列計算機による処理において、表2に示す配列は共有変数となる。従って、共有変数のためのCM容量は20GBになる。

LMには共有変数のための領域とローカル変数のための領域との2種類の領域が必要となる。表4にPEの並列台数毎のオリジナルプログラムと再構成プログラムのLMの必要量を示す。

再構成プログラムの共有変数のためのLM容量はオリジナルプログラムのそれよりどの並列台数に

表5 1要素計算機当たりのデータ転送量の比較

単位: GB

並列台数	オリジナルプログラム	再構成プログラム
16台	10.38	2.97
32台	5.19	1.59
64台	2.59	0.91
128台	1.30	0.56

おいても大きい。また、並列台数を2倍に増大しても容量は半減しない。その理由はプログラム再構成により生じた共有変数の参照データの増大による。

4.3 ローカルメモリ/共有メモリデータ転送量

表5はオリジナルプログラムと再構成プログラムのサブルーチンの前後で発生する共有変数のCMとLM間のデータ転送量を比較したものである。並列処理の1PE当たりのデータ転送量を示す。表5において処理1及び処理2のデータ量は除いて計算してある。

オリジナルプログラムにおいては並列台数を2倍にするとデータ転送量は半減する。再構成プログラムでは共有変数の転送に4平面余分な転送を必要とするので並列台数を2倍にしてもデータ転送量は半分にはならない。PE1台あたりの計算空間が小さくなればなる程4平面の負担は大きくなる。

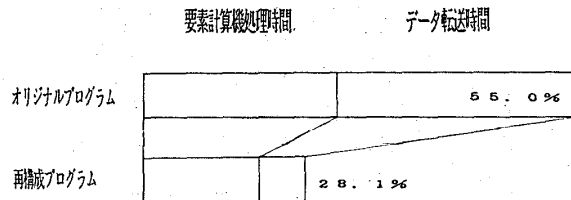
オリジナルプログラムに対して再構成プログラ

表6 要素計算機台数に対する高速化倍率

並列台数	16台	32台	64台	128台
オリジナルプログラム	11.6	18.0	24.4	30.5
再構成プログラム	14.8	27.9	49.7	81.3

表7 データ転送部比率 単位 %

並列台数	16台	32台	64台	128台
オリジナルプログラム	19.3	31.4	44.9	55.0
再構成プログラム	3.5	8.3	15.9	28.1

図3 要素計算機128台時の
処理時間とデータ転送時間

ムのデータ転送量は約3割～4割強に減少している。

4.4 性能予測値の考察

表6はオリジナルプログラムのPE1台による処理時間を1とした場合のオリジナルプログラム及び再構成プログラムのPE台数毎の高速化倍率を示す。オリジナルプログラムの処理性能はPE台数を増大すると30倍にまで高速化されるが、その向上の程度は飽和傾向にある。再構成プログラムの処理性能はPE台数を増大すると81倍にまで高速化されるが、その傾向からPE台数の増大により更なる高速化を図ることができると予測できる。

表7はオリジナルプログラム及び再構成プログラムの処理時間に対するCMとLM間のデータ転送時間の比率をPE台数毎に示したものである。オリジナルプログラムの場合には128台では処理時間の

半分以上を占めている。再構成プログラムでは30%以内に納まっている。いずれのPE台数においても再構成プログラムのデータ転送量削減の効果が表れている。

図3はPE台数が128台時の2つのプログラムの処理時間とデータ転送時間の比較である。共有変数転送時間の減少と並列化部分の拡大の様子がわかる。

5 おわりに

階層メモリ型の高並列ベクトル計算機を計算機モデルとし、並列化による高速化を図るためCFDプログラムを再構成し直し、プログラムの書き換えを行った。PE台数が128台の場合、オリジナルプログラムではPE1台の処理時間の30倍の高速化を得たのに対し、再構成プログラムでは81倍まで高速化された。これにより、プログラムの書き換え効果の大きさを示すことができた。

このプログラム修正のために約1ヶ月を要したが、それにより得られるプログラム高速化による利益はその労力を補って余りあるものと考ええる。

参考文献

- 1) Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow ; Thomas H.Pulliam, Joseph L.Steger , AIAA JOURNAL VOL.18 NO.2
- 2) Implicit Total Variation Diminishing(TVD) Schemes for Steady-State Calculations ; H.C.Yee,R.F.Warming,A.Harten, NASA Technical Memorandum 84342
- 3) 安藤、西 ; 2階層メモリを持つ密結合型スーパーコンピュータにおける並列化技法、第8回航空機計算空気力学シンポジウム論文集、SP-13