NAL TR-1430

ISSN 0389-4010
UDC 681.31

航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-1430

SX－3R システムにおける宇宙推進系プログラムの自動並列化による並列処理

中 村 絹 代 ・ 高 橋 政 浩

2001年8月

独立行政法人 航 空 宇 宙 技 術 研 究 所
NATIONAL AEROSPACE LABORATORY OF JAPAN

目　次

SX－3R システムにおける宇宙推進系プログラムの自動並列化による並列処理*

中　村　絹　代*1　高　橋　政　浩*2

Parallel Processing of Space Propulsion System Programs by Automatic Parallelization of SX-3R System *

Kinuyo NAKAMURA *1　Masahiro TAKAHASHI *2

ABSTRACT

An automatic parallelization function is effective for the reduction of parallel program coding problems. In a shared-memory vector-parallel computer, Fortran program DO loops are automatically parallelized, generally, but this does not occur in a distributed-memory vector-parallel computer.

This report describes the comparison and the investigation of the parallel processing time of two programs on space propulsion system by an automatic parallelization function of the SX-3R system. As a result, the following is quantitatively obtained.

- The performance efficiency of a program parallelized relatively highly and efficiently is large.
- The memory conflict time is the longest in the difference between an estimated time and a measured time. And a ratio of the memory conflict time per the difference time of a highly parallelized program is higher than for a low one.

Keyword: parallel vector computer, Fortran program, automatic parallelization

概　　要

並列ベクトル計算機の自動並列化機能はユーザーのプログラムコーディングの負担を軽減するのに有効である。共有メモリ方式の並列ベクトル計算機では主にDOループを対象に自動並列化が行われている。しかし、分散メモリ方式の並列ベクトル計算機では実現されていない。

本報告は、その自動並列化機能を用いて2本の宇宙推進系プログラムの並列処理を行い、それらの実行結果の比較・検討を行ったものである。結果として、以下のことが定量的に得られた。

- 相対的に並列化率が大きく、並列効率のよいプログラムは、性能向上率が大きい。
- 使用したPE台数に対する予測値と実測値との差に最も影響を与えたものはメモリ競合時間であった。また、メモリ競合時間の割合は並列化率の高いプログラムの方が低いプログラムより高い。

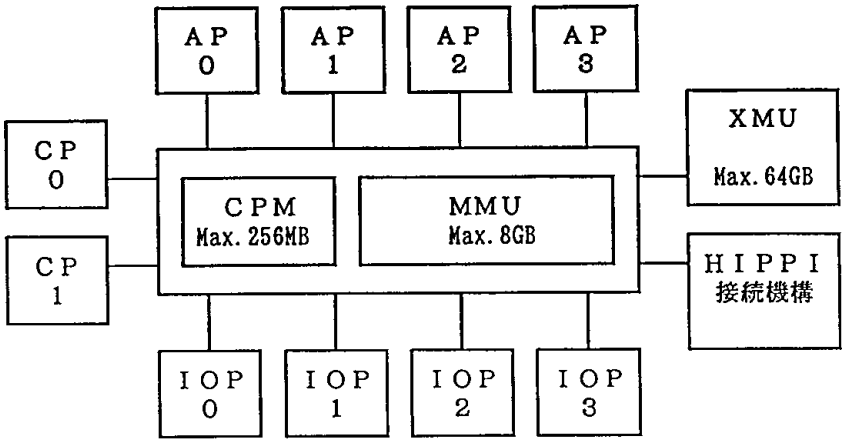
第１章　はじめに

ベクトル計算機が開発されて25年、そして並列ベクトル計算機が稼働開始して約20年が経つ。これらの計算機の有効性については科学技術分野における計算機の発展がそれを物語っている。

当初は数台のベクトル計算機を並列にしたものにすぎなかった並列ベクトル計算機も、現在では航空宇宙技術研究所で要素計算機数166台、ピーク性能280GFLOPSの数値風洞が稼働し数百GFLOPSの同様の並列ベクトル計算機が国内外で稼動している。1 TFLOPS 級の並列ベクトル計算機が稼働開始となるのも時間の問題と言えよう。

＊　平成13年4月5日受付（received 5 April 2001）

＊1　CFD技術開発センター（CFD Technology Center）
＊2　角田宇宙推進技術研究所（Kakuda Space Propulsion Laboratory）



AP …… 演算プロセッサ
MMU …… 主記憶装置
CP …… 制御プロセッサ
CPM …… 制御プロセッサメモリ
XMU …… 拡張記憶装置
IOP …… 入出力処理装置

図2.1 SX－3Rシステム構成図

また、科学技術計算においてこれら計算機で使用されるプログラム言語はFORTRAN言語である。ベクトル計算機においてはFORTRANコンパイラは自動ベクトル化機能を備えており、この機能は現在かなり成熟した段階にあると言ってよい。

並列ベクトル計算機においては命令及びデータの分割／要素計算機への割り付けが重要であり、これらは自動並列化機能と密接に関係する。現在、自動並列化は共有メモリ方式の並列ベクトル計算機においてDOループを対象に行われているのが一般的である。サブルーチン等を対象とした比較的タスク粒度（複数プロセッサの並列処理における仕事の大きさ）の大きいものを対象とした並列処理に関しては自動並列化は殆ど行われていない。

本報告は、並列ベクトル計算機SX-3Rシステムの自動並列化機能を使用して2本の宇宙推進系プログラムの並列処理を行い、プロセッサ数1台、2台及び4台での実行結果の比較・検討を行ったものである。

以下、2章ではSX-3Rシステムの概要及びSX-3Rシステムにおける並列処理の概要を述べ、3章では2本の宇宙推進系プログラムの概要について述べる。また、4章では宇宙推進系プログラムを並列処理し、シングルプロセッサ及びマルチプロセッサにおける処理結果の検討を行う。最後に、5章では本報告で得られたこと及び今後の課題について述べる。

第2章 SX－3Rシステム及び並列処理の概要

2.1 SX－3Rシステム

(1) 概 要

SX-3Rシステムは、日本電気株式会社製の主記憶共有型マルチプロセッサ構成のベクトルスーパーコンピュータシステムである。ハードウェアの全体構成を図2.1に示す。最大4台の演算プロセッサを備え、最大ベクトル演算性能は25.6GFLOPSである。

演算プロセッサ（AP）は、科学技術計算を高速に処理するベクトル演算パイプライン機構、スカラ演算パイプライン機構、キャッシュ機構などから構成されている。制御プロセッサ（CP）はディスクなどの周辺装置との入出力処理を制御する。

主記憶装置（MMU）は最大1024ウェイのインターレース機能のサポートにより高速なデータ供給能力を実現しており、最大51.2GB/秒のデータ供給能力をもつ。また、最大8GBの容量をもち、大規模データを使用するプログラムを効率よく実行させることが出来る。拡張記憶装置(XMU)は、最大64GBの容量を持ち、最大3.2GB/秒の転送速度を実現した高速大容量の半導体記憶装置である。

(2) 演算プロセッサ（AP）

演算プロセッサの構成を図2.2に示す。演算プロセッサはベクトルユニットとスカラユニットから成り、2.5ナノ秒のマシンサイクルで動作する。

アーキテクチャは前機種SX-2のRISCアーキテクチャ（SXアーキテクチャ）を継続し、これを強化している。範囲の広い数値を扱えるように、浮動小数点データ形式として、SX-2の標準形式（IBM形式）に加えて拡張形式（CRAY形式）の採用、固定小数点データに於ける64ビット整数の採用、並列処理を高速にするためのコミュニ

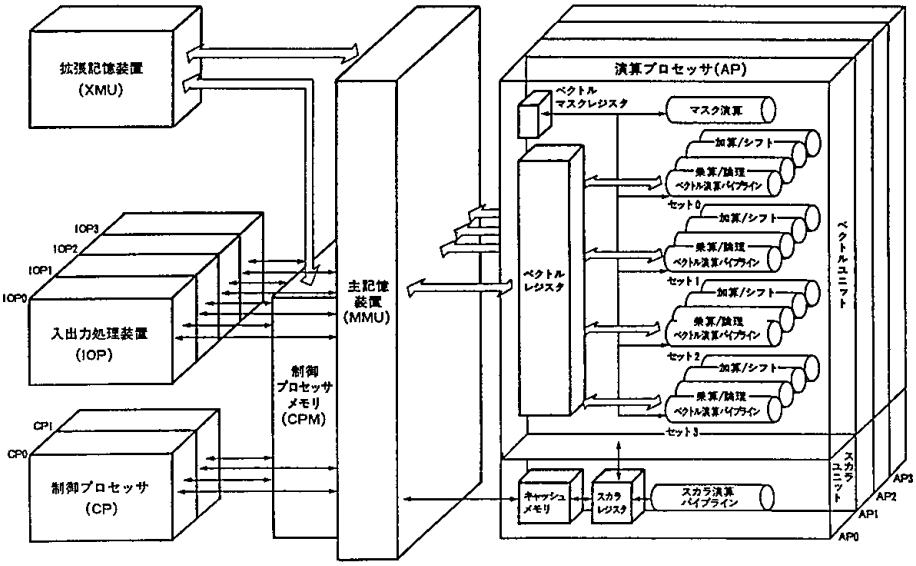


図2.2　SX-3Rシリーズの演算プロセッサ構成

ケーションレジスタ(CR)の採用などが主なものである。

①ベクトルユニット

ベクトル演算パイプラインは、加減算／シフト演算パイプライン及び乗算／論理演算パイプラインをそれぞれ2本、合計4本を基本構成としている。この4本のベクトル演算パイプラインにより同時に4つの浮動小数点演算が実行できる。この構成を1セットとし、演算プロセッサ当たり最大4セットをもつベクトルユニットでは、合計16本のベクトル演算パイプラインを同時に動作させることが可能である。144KBの大容量ベクトルレジスタの活用とともに単一プロセッサで6.4GFLOPSのベクトル性能を実現している。

②スカラユニット

命令の単純化によって基本性能を向上させるRISCアーキテクチャを採用し、これを効果的に動作させるために128個のスカラレジスタを用意している。また、容量64KBのキャッシュ機構を備え、さらにハードウェアによるリオーダリング機能（命令の発行順に関わらず、データの揃った命令から実行すること）により、スカラ用パイプライン演算器の有効活用を図っている。

(3) ソフトウェア

オペレーティングシステムはUNIXベースのSUPER-UXである。SUPER-UXは、UNIX System Vに準拠し、4.3BSD（Berkley Softwear Distribution）機能を取り込んで、オープンシステム環境を提供している。また、高速入出力機能などを追加して高速化を図り、システム運用管理機能、自動運転機能、RAS（Reliability Abailability Servicebility）機能などを強化したスーパーコンピューターセンター向けのオペレーティングシステムである。

さらに、高速化を図る最適化プリプロセッサ、自動ベクトル化／自動並列化機能を持つFORTRAN 77/SXコンパイラ、自動ベクトル化C/SXコンパイラを初めとして、C++プリコンパイラ、その他各種の高速化支援ツール、デバッグツール、科学技術計算ライブラリASL/SXを備えた、使い易いシステムとなっている。

また、並列処理機能としては、単一プロセッサでの多重並列パイプラインによる処理の並列化に加え、マルチプロセッサではマルチタスク方式による並列処理をサポートしている。これは、マルチプロセッサ環境において複数のプログラムを各演算プロセッサで処理してシステムのスループットを向上させるだけでなく、一つのプログラムを複数の演算プロセッサに割り付け、同時並列処理するものである。これによりプログラムのターンアラウンドタイムの短縮を図っている。

(4) 評価に使用したシステム

SX-3Rシリーズには、表2.1(a)のようなモデルがある。

本報告においてプログラム処理に使用したモデルは、プロセッサ当たり6.4GFLOPS、4プロセッサ構成で総合25.6GFLOPSのモデル44R、主記憶4GB、拡張記憶6GBの構成のSX-3Rシステムである。これを表2.1（b）に示す。

2.2　並列処理の概要

1本のプログラムは同時実行可能な部分と可能でない部分に分けることができる。同時実行可能な部分をいくつかの部分に分割し、複数のプロセッサに割り振り、そ

表2.1(a) SX－3Rシリーズのシステム諸元

モデル		モデル44R	モデル42R	モデル41R	モデル34R	モデル24R	モデル22R	モデル21R	モデル14R	モデル12R	モデル11R	モデル1LR
最大ベクトル演算性能		25.6 GFLOPS	12.8 GFLOPS	6.4 GFLOPS	19.2 GFLOPS	12.8 GFLOPS	6.4 GFLOPS	3.2 GFLOPS	6.4 GFLOPS	3.2 GFLOPS	1.6 GFLOPS	0.8 GFLOPS
演算プロセッサ	台数	4台			3台	2台			1台			
	レジスタ ベクトルレジスタ	144KB ×4	72KB ×4	36KB ×4	144KB ×3	144KB ×2	72KB ×2	36KB ×2	144KB ×1	72KB ×1	36KB ×1	
	レジスタ ベクトルマスクレジスタ	256ビット ×8個 ×4	128ビット ×8個 ×4	64ビット ×8個 ×4	256ビット ×8個 ×3	256ビット ×8個 ×2	128ビット ×8個 ×2	64ビット ×8個 ×2	256ビット ×8個 ×1	128ビット ×8個 ×1	64ビット ×8個 ×1	
	レジスタ スカラレジスタ	64ビット ×128 個×4			64ビット ×128 個×3	64ビット ×128 個×2			64ビット ×128 個×1			
	データ形式 固定小数点	32／64ビット										
	データ形式 浮動小数点	32／64／128* ビット										
	データ形式 論理	64ビット										
	ベクトル演算パイプライン	16本 ×4	8本 ×4	4本 ×4	16本 ×3	16本 ×2	8本 ×2	4本 ×2	16本 ×1	8本 ×1	4本 ×1	2本 ×1
	スカラ 演算パイプライン	4セット			3セット	2セット			1セット			
	マスク 付ベクトル処理	可										
	リストベクトル 処理	可										
	キャッシュメモリ	64KB×4			64KB ×3	64KB×2			64KB×1			
主記憶装置	容量	1GB～ 8GB	512MB ～8GB		1GB ～ 8GB	512MB ～8GB	512MB ～4GB	256MB ～4GB		128MB ～4GB	128MB ～2GB	
	インターレース	1024 WAY	512/1024WAY		1024 WAY	512/1024WAY		256/512/1024 WAY		128/256/512/1024WAY		
入出力処理装置	台数	1～4台										
	最大総合データ転送能力	1GB／秒										
	最大入出力チャネル本数	256本										
超高速チャネル	最大チャネル速度	100MB／秒										
	最大チャネル本数	40本										
拡張記憶装置	容量	1GB～64GB										1GB ～ 32GB
	最大転送速度	3．2GB／秒										
制御プロセッサ	台数	1～2台										
	制御プロセッサメモリ	64MB～256MB										

＊スカラ命令のみ

れらを並列に処理することにより、プログラムの実行時間（経過時間）を減少させることができる。同時実行可能であるとは2つの処理において、片方の処理結果をもう一方の処理において必要としないことである。プログラムを並列処理することができるような形にすることを並列化という。

また、プログラムの全処理時間に対する並列処理可能な部分の処理時間の割合を並列化率といい、百分率で表す。

並列処理可能である部分も可能でない部分も、順番に実行することを逐次処理という。

図2.3の(a)は1本のプログラムの逐次処理を示す。(b)は（a）のプログラムの並列処理可能部分を3台のプロセッサで処理することを示す。図中、sn, pnのs, pはそれぞれ逐次処理、並列処理を示し、nはプログラムの分割部分を示す。また、tnはプログラム分割部分nの実行時間を示す。

（a）の場合のプログラム実行時間（経過時間）Taは、

$$Ta = t1 + t2 + t3 + t4 + t5$$

である。（b）の場合のプログラム実行時間Tbは、

$$Tb = t1 + \max(t2, t3, t4) + t5 + \alpha$$
$$= t1 + t4 + t5 + \alpha$$

である。αは並列化に伴う種々のオーバーヘッドである。ここで、t4＞t2＞t3であることから、分割部分2及び3を実行するプロセッサは分割部分4を実行するプロセッ

表2.1(b)　評価に使用したシステム

モデル			モデル４４Ｒ
最大演算性能			２５．６GFLOPS
演算プロセッサ AP	台数		４台
	レジスタ	ﾍﾞｸﾄﾙﾚｼﾞｽﾀ	１４４KB×４
		ﾍﾞｸﾄﾙﾏｽｸﾚｼﾞｽﾀ	２５６ﾋﾞｯﾄ ×８個×４
		ｽｶﾗﾚｼﾞｽﾀ	６４ﾋﾞｯﾄ ×１２８個×４
	データ形式	固定小数点	３２／６４ﾋﾞｯﾄ
		浮動小数点	３２／６４／１２８* ﾋﾞｯﾄ
		論理	６４ﾋﾞｯﾄ
	ﾍﾞｸﾄﾙ演算ﾊﾟｲﾌﾟﾗｲﾝ		１６本×４
	ｽｶﾗ 演算ﾊﾟｲﾌﾟﾗｲﾝ		４ｾｯﾄ
	ﾏｽｸ 付ﾍﾞｸﾄﾙ処理		可
	ﾘｽﾄﾍﾞｸﾄﾙ 処理		可
	ｷｬｯｼｭﾒﾓﾘ		６４KB×４
主記憶装置 MMU	容量		４GB
	ｲﾝﾀﾘｰｽ		１０２４WAY
拡張記憶装置 XMU	容量		６ GB
	最大転送速度		３．２ GB ／秒

＊スカラ命令のみ

サが実行を終了するまで待つ。t2＝t3＝t4の場合にはプロセッサに均等に仕事が割り当てられており、ワークロードバランスが良いという。

これらから分かるように並列化率が100%、αが0、ワークロードバランスが均等であれば、使用したプロセッサ数倍の高速化倍率を得ることができる。しかし、実際にはこういった状況にはならない。

従って、並列化率をできるだけ大きく、種々のオーバーヘッドをできるだけ少なく、ワークロードバランスをできるだけ均等にするように努力することが並列化による高速化倍率の向上につながる。

以下、本節では、SX-FORTRANによるプログラムの並列処理の概要を示す。

2.2.1　SX－FORTRANによる並列処理機能

マルチスレッド型の並列処理機能をSX-FORTRANではマルチタスク機能[1]と呼んでいる。本機能にはマクロタスク機能とマイクロタスク機能の二つがある。前者はサブルーチンやサブルーチンの集合を並列処理の単位とし、後者はDOループ（外側ループ、内側ループ）、文の集合、文を並列処理の単位として並列化を行う。本節では、マクロタスク機能及びマイクロタスク機能の概略及び利用例について説明する。

(1) マクロタスク機能

マクロタスク機能は並列実行可能なサブルーチンやサブルーチン群で構成される複数のマクロタスクに対して、各々別のプロセッサにタスクを割当、並列実行を行う。プログラムに対するこれらの指示はライブラリの呼び出しにより行われる。

マクロタスク処理を行うために、タスク制御機能、排他制御機能、イベント制御機能及びバリア制御機能の4つの制御機能が用意されている。タスク制御とはプログ

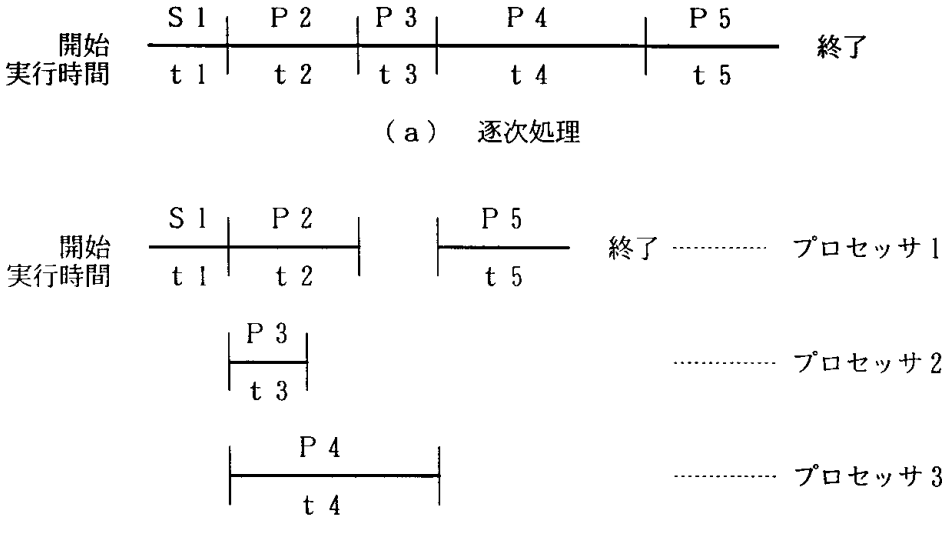


図2.3　プログラムの処理

表2.2 マクロタスク制御ルーチン一覧表

	ルーチン名・関数名	機能
タスク制御ルーチン	PTFORK	マクロタスクの生成
	PTJOIN	子タスクの終了待ち合わせ
	PTPARAM	タスクパラメータの取り出し
	IPTSTAT	子タスクの状態調査
	IPTID	タスク識別番号の返却
	IPTNAP	演算プロセッサ台数の返却
	PTLIST	タスクの状態を示すリストの出力
	PSTUNE	タスクスケジューラのパラメータの変更
	IPSMAX	最大物理タスク数の返却
排他制御ルーチン	PLASGN	ロックの割当
	PLLOCK	ロックを掛ける
	PLUNLOCK	ロックを外す
	PLFREE	ロックの解放
	IPLSTAT	ロックの状態の調査
	IPLSTATL	ロック状態を調査しロックが掛かっていなければロックを掛ける
イベント同期制御ルーチン	PEASGN	イベントの割当
	PEWAIT	イベントの通知待ち
	PEPOST	イベントの通知
	PECLEAR	イベントのクリア
	PEFREE	イベントの解放
	IPESTAT	イベントの状態の調査
バリア同期制御ルーチン	PBASGN	バリアの割当
	PBSYNC	タスクの待ち合わせタスクの再開
	PBFREE	バリアの解放
	IPBSTAT	バリアの状態の調査

ラムの一部を複数のタスクに分割したときに、それらのタスクや生成の起動を指示したり、現在のタスクの状態を調べるなど、タスクを管理するための機能である。タスク制御により並列実行されるタスク間では、データ共有したり、サブルーチンの実行結果を他のサブルーチンで利用するなど、サブルーチンの実行順序が結果に影響を与える場合がある。このような場合、排他制御機能、イベント制御機能、バリア制御機能を用いて矛盾なくプログラムの実行を行うことができる。排他制御機能は、複数タスクの排他的な実行を指示する。イベント制御機能は、複数タスクを同期をとって実行することを指示する。バリア制御機能は、イベント制御機能と同様に同期をとることを指示し、同時に実行を開始したいタスクの数を指定することにより、同時実行開始する。表2.2にマクロタスク制御ルーチンの一覧表を示す。

理解を容易にするために、表2.2の制御ルーチンを用いたプログラムの並列処理を図2.4にマクロタスク使用例として示す。図2.4 (a) はオリジナルプログラムである。(b) は組み込みルーチンを用いた並列処理プログラムである。アンダーラインのある文はタスク制御ルーチンである。(c) は並列動作を示したものである。

(2) マイクロタスク機能

マイクロタスク機能は、DOループ及び文の集合を対象に並列化指示行を挿入することにより並列処理を行う機能である。

並列化指示行は、並列化を行う場所の直前に挿入し並列処理の指示を行う。並列化指示行は、

```
      EXTERNAL SUB
      COMMON /COM/A(1000)
      WA=0
      DO 10 I=N1,N2
        WA=WA+A(I)
   10 CONTINUE
      RETURN
      END
```

（ａ）

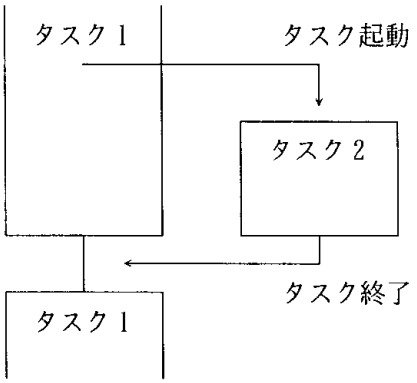
メインタスク（TASK1）

```
      EXTERNAL SUB
      COMMON /COM/A(1000)
      CALL PTFORK(TID,TPARAM,SUB,S,1,500)
      CALL SUB(T,501,1000)
      CALL PTJOIN(TID)
      W=S+T
      RETURN
      END
```

サブタスク（TASK2）

```
      SUBROUTINE SUB(WA,N1,N2)
      COMMON /COM/A(1000)
      WA=0
      DO 10 I=N1,N2
        WA=WA+A(I)
   10 CONTINUE
      RETURN
      END
```

（ｂ）



（ｃ）

図2.4　タスク制御ルーチンを用いたマクロタスク使用例

＊PDIR　並列化制御オプション

の形式をとる。並列化制御オプションを表2.3に示す。これらの並列化制御オプションは次節の「SX-FORTRANによる自動並列化機能」と関係するので、本節での使用例の説明は冗長となるので省き、次節において説明する。

2.2.2　SX－FORTRANによる自動並列化機能

SX-FORTRANの自動並列化機能[1]は、FORTRANコンパイラの最適化プリプロセッサ（fopp;fortran optimized pre-processor）を起動することにより実現される。

foppはループの繰り返し間のデータ依存解析等を解析して、オリジナルプログラム中に並列化指示行を自動挿入する。また、必要に応じてオリジナルプログラムを局所的に変形する。

foppにより生成されたプログラムにさらにオプションや並列化指示行などにより並列化の指定を行い手を加えることが可能である。

foppにより自動的に生成される指示行を表2.3に示す。foppの自動並列化のためのオプションを指定して実行することにより、表に示す指示行が自動生成される。

図2.5にPARDOを挿入する場合の例を示す。(a)はオリジナルプログラムであり、(b)は自動変換後のプログラムである。

図2.6にオリジナルプログラムの並列実行可能な部分にPARCASE群を自動挿入した例を示す。(a)はオリジナルプログラムであり、(b)は自動変換後のプログラムである。

図2.7に条件並列化による自動並列化の例を示す。(a)はオリジナルプログラムであり、(b)は自動変換後のプログラムである。多重DOループの繰り返し数M×Nがある値（この例では2000）を越えれば並列処理を行い、越えなければオリジナルプログラムの並列化されない多重

表2.3 マクロタスク並列化制御オプション及び自動並列化制御オプション一覧表

並列化制御オプション	機能	自動並列化制御オプション
RESERVE＝n	最大n個のマイクロタスクからなるマイクロタスクグループを生成することを指示する。	○
RELEASE	マイクロタスクグループの解放を指示する。	○
PARDO BY＝n PARDO FOR＝n	DOループの繰り返しを並列に実行することを指示する。BY及びFORはDOループの繰り返しの量を指定するのに用いる。	○
PARCASE（配列名） CASE END CASE	これらの制御オプションで囲まれた部分を並列に実行することを指示する。	○
SERIAL END SERIAL	この2つの制御オプションに囲まれた部分をマスタタスクで実行することを指示する。	○
CRITICAL＝n END CRITICAL＝n	この2つの制御オプションで囲まれた部分をマイクロタスク間で排他的に処理することを指示する。	○
POST（＝n） WAIT（＝n）	DOループ内又はPARCASEブロックに属する文の間で実行順序に依存関係がある場合に、これらの制御オプションを用いて実行順序を保証することを指示する。	——
EXIT	PARDO及びPARCASEにおいてEXITの位置に到達した時点以降、全てのタスクでまだ処理されていない実行の中止を指示する。	——
PRIVATE（変数名）	指定した変数を、あるPARDOループまたはPARCASEグループの中でだけローカル領域に割り当てられた作業用変数で置き換えることを指示する。	——

```
      COMMON /X/ A(1000,1000),B(2000,1000)
           ..........
      DO 300 J=1,1000
      DO 300 I=1,1000
      K=I+L
      T=SQRT(A(I,J))
  300 B(K,J)=T+1.0/T
```

（a）

```
      CALL LOOP300$1 (A,B,L)
           ..........
      SUBROUTINE LOOP300$1 (A,B,L)
      REAL A(1000,1000),B(2000,1000),T
      INTEGER J,I,L
*PDIR PARDO FOR=4
      DO 300 J=1,1000
*VDIR    NODEP
         DO I=1,1000
            T=SQRT(A(I,J))
            B(I+L,J)=T+1.0/T
         ENDDO
  300 CONTINUE
      RETURN
      END
```

（b）

図2.5 PARDOを使用した自動並列化

```
      DO 202 J=1,M
        DO 201 I=1,N
          A(I,J)=(A(I-1,J)+A(I,J-1))*0.5
  201   CONTINUE
  202 CONTINUE
      NK=N-K
      ML=M+L
      DO 302 J=1,ML
        DO 301 I=1,NK
          C(I,J)=(C(I-1,J)+C(I,J-1))*0.5
  301   CONTINUE
  302 CONTINUE
```

（a）

```
*PDIR PARCASE
      DO 202 J=1,M
        DO 201 I=1,N
          A(I,J)=(A(I-1,J)+A(I,J-1))*0.5
  201   CONTINUE
  202 CONTINUE
*PDIR CASE
      NK=N-K
      ML=M+L
      DO 302 J=1,ML
        DO 301 I=1,NK
          C(I,J)=(C(I-1,J)+C(I,J-1))*0.5
  301   CONTINUE
  302 CONTINUE
*PDIR ENDCASE
```

（b）

図2.6 PARCASE群を使用した自動並列化

```
      DO 50 J=1,M
        DO 40 I=1,N
          A(I,J)=0.0
   40   CONTINUE
   50 CONTINUE
```

（a）

```
      IF(M*N.GT.2000) THEN
         CALL T$1(M,N,A)
      ELSE
         DO J=1,M
*VDIR      NODEP
           DO I=1,N
             A(I,J)=0.0
           ENDDO
         ENDDO
      ENDIF
        ..........
      SUBROUTINE T$1(M,N,A)
      INTEGER M,N,I,J
      REAL A(999,999)
*PDIR PARDO
      DO 50 J=1,M
*VDIR   NODEP
        DO 40 I=1,N
          A(I,J)=0.0
   40   CONTINUE
   50 CONTINUE
      RETURN
      END
```

（b）

図2.7　作業量を基準にした条件並列化による自動並列化

DOループを実行する。図中、

＊VDIR NODEP

の指示行は、後に続くDOループをベクトル化することを示す。

この他、総和、累積などのマクロ演算に対する自動並列化、最内側DOループの自動並列化等を行うための変換などもfoppで行うことができる。

第3章　推進系プログラムの概要

3.1　プログラムIの概要

プログラムIは、スクラムジェットエンジンの2次元燃焼器流れを解析するプログラムである。

一般座標系ナビエ・ストークス方程式

$$\partial Q/\partial t+\partial F/\partial \xi+\partial G/\partial \eta=\partial F_V/\partial \xi+\partial G_V/\partial \eta$$

Q, F, G, F_V, G_V：4次元ベクトル

にKRCスキーム[2]を適用して差分方程式を求めIAF解法を用いて解いている。この方程式は次のように書くことができる。

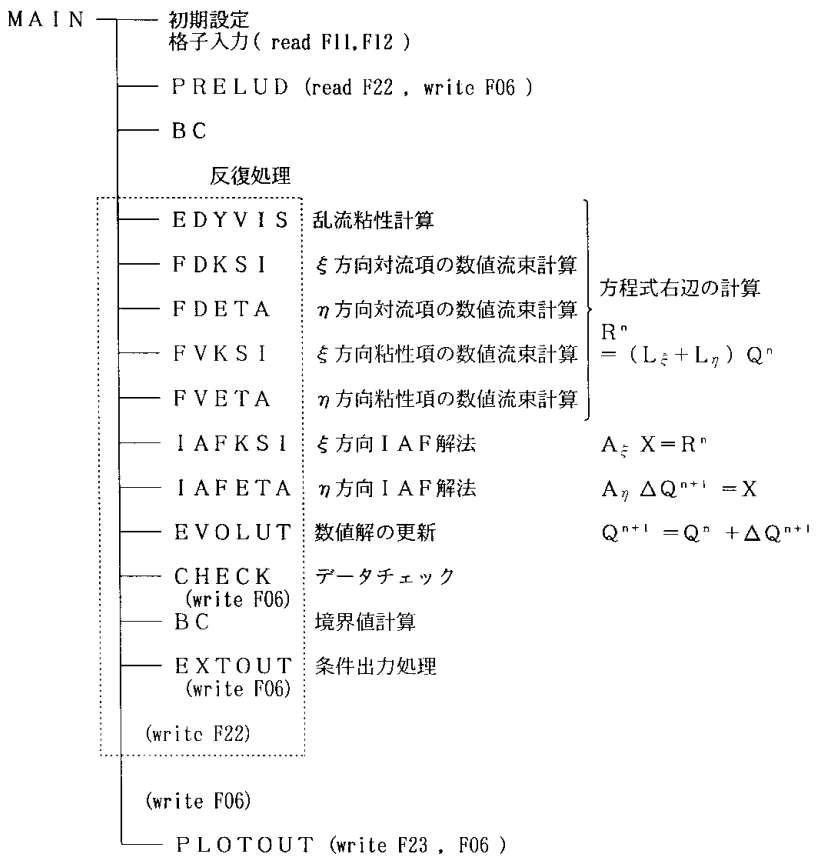


図3.1　プログラムIのプログラム構造

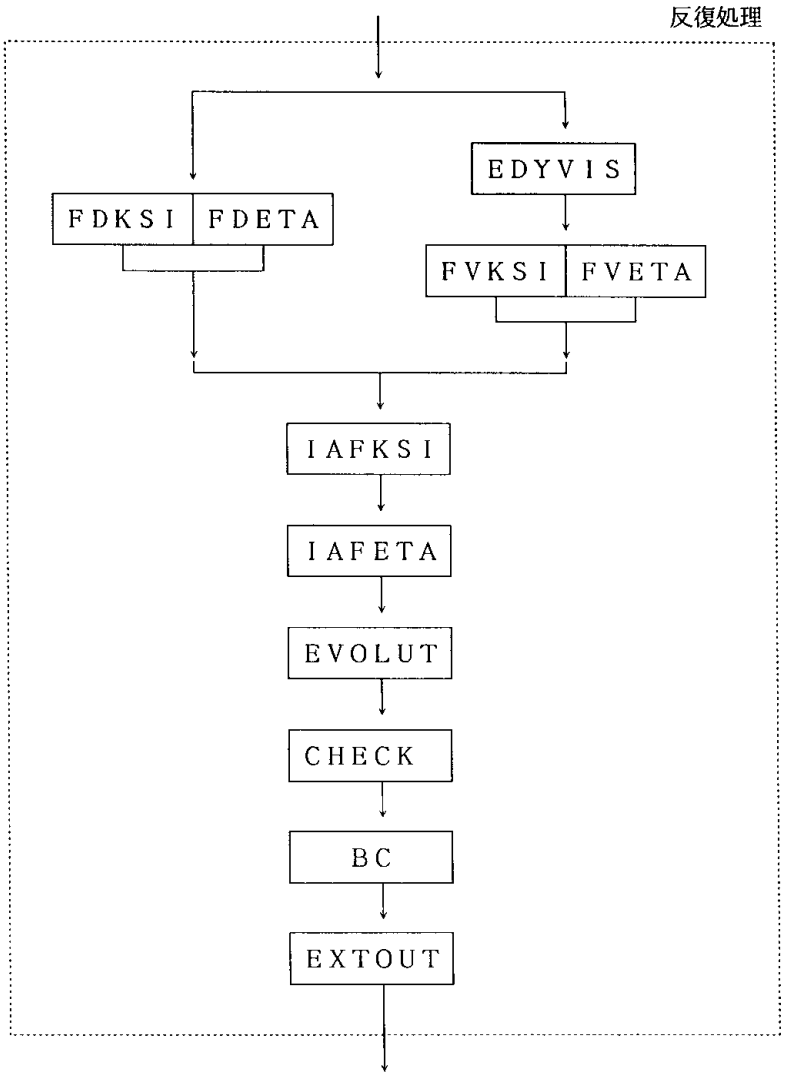


図3.2 サブルーチン単位の並列処理構造

$$A_\xi A_\eta \Delta Q^{n+1} = (L_\xi + L_\eta) Q^n$$

L_ξ, L_η, A_ξ, A_η：オペレータ

図3.1にプログラム構造及びサブルーチンの処理内容を示す。

本プログラムでは、方程式の右辺を計算し、方程式を解き、ΔQ^{n+1}を求める操作を指定回数、繰り返す。プログラム構造図に示す機能をもつサブルーチンのうち、FDKSI, FDETA, EDYVIS, FVKSI, FVETAは図3.2に示す形式でサブルーチン単位での並列処理が可能である。

本プログラムで用いられる配列は1次元配列、2次元配列及び4次元配列であり、その殆どが2次元配列である。特に、複数のサブルーチン間で用いられるグローバルな配列は全て2次元配列であり、約50個ある。そのほか、サブルーチン内でのみ使用されるローカルな配列は約170個ある。1データ当たり倍精度（8バイト）で計算すると約100MBのデータ量となる。

また、各サブルーチンのDOループは1重DOループと2重DOループから構成されている。2重DOループの最内DOループの個数は1つまたは2つであり、その殆どは1つである。それらの数と並列性の有無を表3.1に示す。本表で並列処理可能とは、DOループの処理の内容から基本的に並列処理できるものをいう。従って、並列ベクトル計算機のコンパイラでは、ベクトル対象とならないread文／write文などを含むDOループも並列処理可能となっている。表に示すように殆どのDOループは並列処理可能であるが、IAFKSI及びIAFETAサブルーチンの2重DOループの内、半分のDOループについては解法上、1方向は逐次処理となっている。

本プログラム実行条件は格子点数601×101、反復回数2000回である。

3.2 プログラムIIの概要

プログラムIIは、高温衝撃風洞の軸対称2次元化学反

表3.1　プログラムIのDOループの並列性

サブルーチン名	DOループ数		DOループの並列性	
	2重	1重	2重DOループ	1重DOループ
MAIN*	3	0	両方向並列処理可能	―――
PRELUD*	9	0	両方向並列処理可能	―――
EDYVIS	4	6	両方向並列処理可能	並列処理可能
FDKSI	20	3	両方向並列処理可能	並列処理可能
FDETA	20	3	両方向並列処理可能	並列処理可能
FVKSI	2	0	両方向並列処理可能	―――
FVETA	2	0	両方向並列処理可能	―――
IAFKSI	4	1	2：1方向並列処理可能 他方向逐次処理 2：両方向並列処理可能	並列処理可能
IAFETA	4	1	2：1方向並列処理可能 他方向逐次処理 2：両方向並列処理可能	並列処理可能
EVOLUT	1	0	両方向並列処理可能	―――
CHECK	1	0	両方向並列処理可能	―――
BC	1	8	両方向並列処理可能	並列処理可能
EXTOUT	1	4	両方向並列処理可能	並列処理可能
PLOTOUT*	2	0	両方向並列処理可能	―――

＊印は反復処理外サブルーチンである。

応流・非定常流れを解析するプログラムである。基礎方程式は以下の一般座標系ナビエ・ストークス方程式である。

$$\partial U/\partial t+\partial F/\partial \xi+\partial G/\partial \eta=\partial F_V/\partial \xi+\partial G_V/\partial \eta+S_r+S_s$$

$U,\ F,\ G,\ F_v,\ G_v,\ S_r,\ S_s$
：11次元ベクトル

これを、Strung's Time Splittingにより、対流項＋粘性項（＋軸対称項）と化学反応生成項の式に時間分割し、対流項＋粘性項（＋軸対称項）に対しては陽的2次Runge-Kutta法を、そして化学反応生成項には陰的Crank-Nicolson法を適用し、時間2次のオペレータにより計算する。これにより、Uを求める。

L_f：　$\partial U/\partial t+\partial F/\partial \xi+\partial G/\partial \eta=\partial F_v/\partial \xi+\partial G_v/\partial \eta+S_r$

L_s：　$\partial U/\partial t=S_s$

$U^{n+1}=L_f(\Delta t/2)L_s(\Delta t)L_f(\Delta t/2)U^n$

L_f：陽的2次Runge-Kutta法による時間積分オペレータ

L_s：陰的Crank-Nicolson法

なお、空間差分については、対流項には2次精度KRCスキーム[2)]、粘性項には中心差分法を用いている。

図3.3にプログラム構造及びサブルーチンの処理内容を示す。図において、MAINプログラムの前処理は初期設定及び格子生成のサブルーチン群である。後処理は本プログラムで求められた数値解や基本的な配列の磁気ディスクへの出力処理である。煩雑のため、プログラム構造図への記載は省略する。

本プログラムはプログラム構造図に示すような一連のサブルーチン群の連続処理となっている。これらのサブルーチンの内、FDKSI2及びFDETA2はサブルーチン単

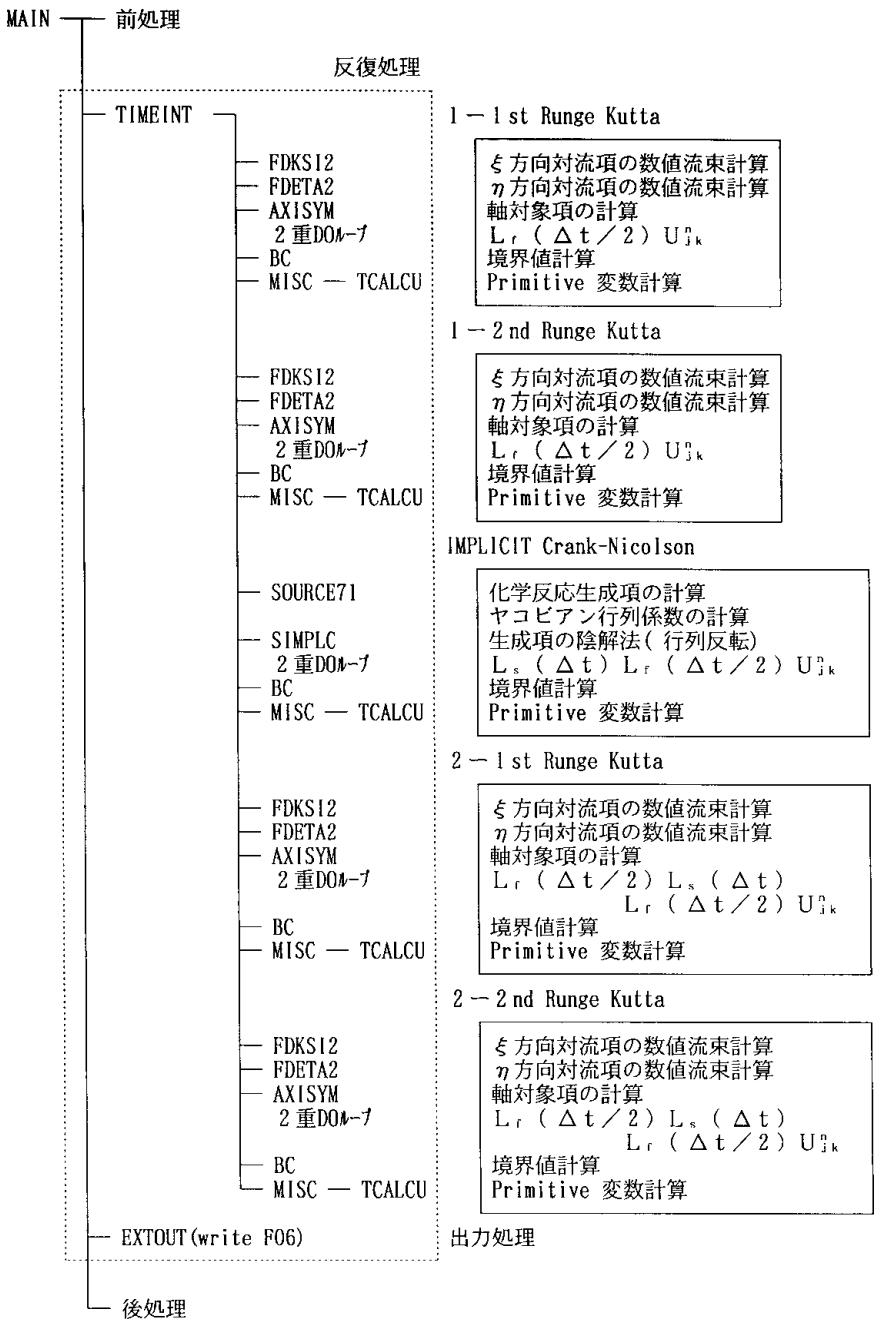


図3.3 プログラムIIのプログラム構造

位で並列処理することが可能である。

本プログラムに用いられる配列は1次元配列、2次元配列及び3次元配列である。これらの配列の内、複数のサブルーチンで使用されるグローバルな配列は約250個、サブルーチン内でのみ使用されるローカルな配列も約250個ある。1データ当たり倍精度（8バイト）で計算すると約130MBのデータ量となる。

また、サブルーチンのDOループは1重DOループと2重DOループから構成されており、2重DOループの最内DOループの個数は1～11個と巾広いが、その殆どは1個である。最内DOループの個数が多いサブルーチンはSIMPLC，FDKSI2，FDETA2およびTCALCUである。

DOループの種類及びその並列性の有無を表3.2に示す。本表において並列処理可能であるとはDOループの処理内容から基本的に並列処理できるものをいう。表に示すようにDOループは並列処理可能であるが、SIMPLCサブルーチンの2重DOループ3つの内、2つの2重DOループについては解法上、1方向は逐次処理となっている。このように、本プログラムは行列反転を除くDOループについては全て並列処理可能である。

本プログラムの実行条件は格子点数801×51、反復回数1000回である。

表3.2　プログラムIIのDOループの並列性

（反復処理内サブルーチン）

サブルーチン名	DOループ数		DOループの並列性	
	2重	1重	2重DOループ	1重DOループ
TIMEINT	6	0	両方向並列処理可能	———
FDKSI2	1	0	両方向並列処理可能	———
FDETA2	1	0	両方向並列処理可能	———
AXISYM	1	0	両方向並列処理可能	———
BC	0	4	———	並列処理可能
MISC	5	0	両方向並列処理可能	———
TCALCU	2	0	両方向並列処理可能	———
SOURCE71	0	5	———	並列処理可能
SIMPLC	3*	1	2：1方向並列処理可能 他方向逐次処理 1：両方向並列処理可能	並列処理可能
EXTOUT	1	0	両方向並列処理可能	———

（反復処理外サブルーチン）

サブルーチン名	DOループ数		DOループの並列性	
	2重	1重	2重DOループ	1重DOループ
MAIN	3	0	両方向並列処理可能	———
PRELUD	1	0	両方向並列処理可能	———
EQCOMP7	0	4	———	並列処理可能
QUDRATC	0	1	———	逐次処理
QUARTIC	0	1	———	逐次処理
LOWTEMP	0	1	———	逐次処理
CHARAC	0	6	———	並列処理可能
GRIDRD	7	2	両方向並列処理可能	並列処理可能
SETVISC	1	1	両方向並列処理可能	並列処理可能

＊：DOループはその一部に3重DOループとなる部分を含む。

第4章　推進系プログラムの並列処理性能の分析

4.1　測定方法

(1) 測定環境

・使用ハードウェア：SX-3/44R
　　主記憶4GB

・使用ソフトウェア：
　オペレーティングシステム：SUPER-UX (R4.2)
　コンパイラ：FORTRAN77/SX R082
　最適化プリプロセッサ：FOPP4.07J3
　解析ツール：ANALYZAR-P/SX R154

(2) 測定条件

測定条件は以下のとおりである。

・拡張記憶装置（XMU）は使用しない。これは、プログラム上、拡張記憶を使用する指示が入らないことを示す。

・コンパイルオプションはサブルーチン単位で指定を

行った。

・オリジナルプログラムを修正せずに評価対象プログラムとした。ただし、プログラムＩに含まれるサブルーチンCHECK内のDOループ中のWRITE 文はベクトル化を極端に阻害しており、機能的にも中間結果の出力のみを行っており、最終結果に影響を与えないので省き、そのDO ループをベクトル化されたDO ループとして処理を行った。

・基本的に自動ベクトル／自動並列化機能を用いてチューンアップを行った。

・並列化に関してはDO ループの並列化、即ち、マイクロタスク処理のみ適用した。

(3) 測定手順

プログラムのチューニングにおいては、基本的には最適化プリプロセッサとコンパイラによる、ソース・コードとオブジェクト･コードの最適化が主たる作業であり、以下の手順を繰り返す。

① 最適化プリプロセッサはベクトル化や並列化の対象となる一連のDO ループを最適化の対象にし、オリジナルプログラムをベクトル化や並列化に最適なプログラムへと変形する。

最適化プリプロセッサの並列化に関するソースコードの修正とは、多くの場合並列化の対象となるループをサブルーチン化してそれを呼び出す形に変形する事である。例えば以下のようになる。

```
オリジナルプログラム
        ・・・・・
      DO 10 I=1, N
        DO 10 J=1, M
          ・・・
       10 CONTINUE
```

ループ削除
ルーチン新規作成

```
変形プログラム
        ・・・・・
      CALL SUB1
        ・・・・・

      SUBROUTINE SUB1
      DO 10 I=1, N
        DO 10 J=1, M
          ・・・
       10 CONTINUE
```

② 次に、コンパイラ FORTRAN77/SX により、並列化及びベクトル化の最適オブジェクトコードを生成する。

③ 測定結果により、ベクトル化状況、並列化状況、CPU コスト分布、キャッシュ・ミスヒット時間、バンク競合時間等のバランスを鑑みて、ソースコードに最適化プリプロセッサやコンパイラに対する指示行を適宜挿入して、性能を向上させる。

4.2 プログラムＩの並列処理

(1) シングルプロセッサ処理

プログラムＩのサブルーチン単位のCPU 負荷分布を図4.1 に示す。

上位6ルーチンで全経過時間の94%の経過時間を占め、特にIAFKSI，FDETA，FDKSI，IAFETA の4 ルーチンはほぼ均等に分布している。メインルーチンには最初と最後で生じるファイル入出力時間、約 18 秒が含まれてい

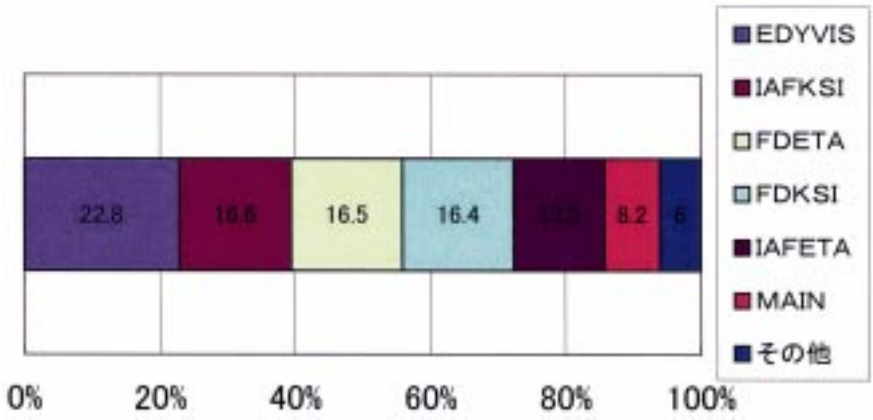


図4.1 プログラムＩの CPU 負荷分布

る。

プログラムＩの平均ベクトル化率は99.5%であるが、最も負荷の大きいEDYVISルーチンのベクトル化率は95.3%でありやや低い。これはEDYVISルーチンのCPU負荷の大きい、又はCPU時間を多く消費する（以下、高コストという）ループに4つの条件文とGOTO文があるためにベクトル化されない部分があることによる。

EDYVISルーチンに続くFDETA, FDKSI, IAFETA, IAFKSIルーチンについては各ルーチンとも表3.1に示すように、ループは沢山あるがすべて最深ループがベクトル処理されている。

プログラムＩの平均ループ長は170.2と長い方であり、良い性能が得られている。

(2) マルチプロセッサ処理

プログラムＩのマルチプロセッサ処理においては、全てのサブルーチン内のDOループにマイクロタスク処理を適用した。以下に示すルーチンを並列化した。

- EDYVIS
- FDETA
- FDKSI
- IAFETA
- IAFKSI
- MAIN
- BC
- FVETA
- FVKSI
- EVOLUT （並列効果は小）
- CHECK （並列効果は小）
- PRELUD （並列効果は小）
- PLOTOUT （並列効果は小）

並列化は基本的に自動並列化を行っている。並列処理可能であるが、最適化プリプロセッサによる並列処理指示行の自動挿入がなされないDOループに対しては、並列化指示行を挿入した。本プログラムにおいては、表4.1に示すDOループがその対象となった。

以上のように、本プログラムに並列化を適用した結果を表4.2に示す。高コストルーチンを代表として、そのルーチンの中の高コストループのループ長及びベクトル／並列演算状況を示す。

最高コストルーチンEDYVISの中で最も時間のかかるDOループ内において、2次元配列の第2添字についてのデータの引用があり、同時に最小値計算のための条件／GOTO文がある。こうした場合、ベクトル化が困難であるためスカラー演算で計算を行うこと、及びキャッシュミス（キャッシュメモリにアクセスデータが存在しない状況）が起こり易いために最もCPUコストのかかるルーチンとなっている。

IAFETA及びIAFKSIルーチンにおいてはルーチン内の最高コストループは2重ループであり、内側のループはベクトル化されているが、外側のループは漸化式計算となるため、ベクトル化も並列化も不可能である。したがって、並列処理による性能向上がそれほど期待できない。

FDETA及びFDKSIルーチンはすべてのループに関して、内側ループがベクトル化、外側ループが並列化されている。

3章の表3.1と表4.1を比較すると高コストループにおいて、並列処理可能な部分は全て並列処理又はベクトル処理されることがわかる。

(3) シングルプロセッサ及びマルチプロセッサ処理結果の検討

以下に示す測定結果の時間は、いずれもプログラムの経過時間を示す。

並列処理の経過時間の分析を行うためには、オリジナルプログラムをいくつかのバージョンに変形させたプログラムの目的別オブジェクトプログラムを作成し、必要なプロセッサ数を使用して実行する必要がある。それらをケースとしてまとめ表4.3に示す。

表中、ケース名として示すOPV11はシングルプロセッサ対応のロードモジュールを１プロセッサで実行する場合を示す。PPV11は最適化プリプロセッサがオリジナルプログラムを並列処理用プログラムに書き換えたものを、

表4.1 プログラムＩのチューニング

サブルーチン名	修正内容	修正箇所
ＥＤＹＶＩＳ	並列化指示行によるＤＯループの並列化（＊ＰＤＩＲ　ＰＡＲＤＯ　ＦＯＲ＝２）	ＤＯ　３０　ループ直前に挿入

表4.2 プログラムIの並列処理

サブルーチン	高コストループ	ループ長	ベクトル／並列	備考
EDYVIS	DO 30 DO 31 DO 32	601 99 101	P V V	最小値計算部分で、部分ベクトル化である。
FDETA	DO 10 DO 10	101 599	P V	本サブルーチン内の他の2重DOループもDO 10 ループと同処理される。
FDKSI	DO 10 DO 10	99 601	P V	本サブルーチン内の他の2重DOループもDO 10 ループと同処理される。
IAFETA	DO 100 DO 30	98 599	V	本2重DOループの他の最内DOループも同様にベクトル化。外側DOループは添え字の繰上のため並列化出来ない。
IAFKSI	DO 100 DO 30	598 99	V	本2重DOループの他の最内DOループも同様にベクトル化。外側DOループは添え字の繰上のため並列化出来ない。

注） P：並列化可能　　V：ベクトル化可能

表4.3 各種ケース一覧表

ケース名	ソースプログラム	オブジェクトプログラム	実行時使用プロセッサ数
OPV11	オリジナルプログラム	シングルプロセッサ用 ベクトル化のみ	1台
PPV11	並列処理用プログラム	シングルプロセッサ用 ベクトル化のみ	1台
PPVP21	並列処理用プログラム	2プロセッサ用 ベクトル化＋並列化	1台
PPVP22	並列処理用プログラム	2プロセッサ用 ベクトル化＋並列化	2台
PPVP44	並列処理用プログラム	4プロセッサ用 ベクトル化＋並列化	4台

1プロセッサのベクトル演算のみで実行するようにコンパイルしたロードモジュールを1プロセッサで実行する場合を示す。PPVP21は2プロセッサ用の並列処理プログラムを1プロセッサで実行する場合を示す。PPVP22及びPPVP44はそれぞれ2プロセッサ及び4プロセッサ用の並列処理プログラムのロードモジュールを2台または4台

表4.4　プログラムIの測定値分析結果（1）経過時間一覧表（単位：秒）

ケース名	測定値			理論値	
	経過時間	倍率	ｷｬｯｼｭ・ﾐｽﾋｯﾄ時間	経過時間	倍率
OPV11	232	1.00	5	—	—
PPV11	237	—	5	237	1.00
PPVP21	244	—	6	—	—
PPVP22	169	1.37	9	151	1.57
PPVP44	152	1.53	16	108	2.19

のプロセッサで実行する場合を示す。

ここで、PPV11は並列化率から理論性能倍率を計算する目的で、ソースプログラム種別を統一するためのものである。また、PPVP21の処理により並列処理を行ったことによるオーバーヘッド時間を経過時間積算として計測する。

本プログラムの並列化率は並列処理プログラムのFORTRANアナライザの実行結果より72.5%であった。本プログラムの1プロセッサでの実行時間のうち、並列実行可能部分の実行時間が72.5%あることを示す。

並列化率から並列化による実行時間の理想的な理論性能倍率を求める。プロセッサ数nで並列処理する場合の理論性能倍率r_nを次式により求める。

$$r_n = 1/\{p/n + (1-p)\}$$

n：並列台数

p：並列化率の実数表示

これにより理論性能倍率は、2プロセッサの場合1.57倍、4プロセッサの場合2.19倍である。

表4.4に5つのケースの実行結果及び理論性能を示す。理論性能倍率から算出した理論予測値は2プロセッサでは151秒であり、4プロセッサでは108秒である。

表4.4より実測による並列化効果は、2プロセッサの理論性能倍率1.57に対し実測性能倍率1.37であり、4プロセッサの理論性能倍率2.19に対し1.53である。ここで理論と実測では、倍率のベースとなるロードモジュールが異なるので、厳密には比較できないものであるが、実際には大差ないので比較しても良いであろう。

図4.2に各ケースの経過時間及び理論性能倍率による経過時間を示す。PPV11とOPV11の経過時間の差をt_1、PPVP21とPPV11の経過時間の差をt_2、PPVP22及びPPVP44とそれぞれの理論性能倍率による経過時間の差をt_3、t_4で表す。それぞれ、$t_1=5$秒、$t_2=7$秒、$t_3=18$秒、$t_4=44$秒である。

t_1は最適化プリプロセッサにより作成された並列処理用ソースプログラムの書き換えられることにより増大した部分の経過時間を表す。

t_2は各プロセッサとの並列処理命令やデータをやり取りする際の前処理と後処理時間である並列処理に関するオーバーヘッド時間積算分及びキャッシュ・ミスヒット時間である。

t_3及びt_4は

(a) ソースプログラムの書換えによる増加時間

(b) 並列処理に関するオーバーヘッド時間

(c) 負荷分散（並列に実行するプロセスの実行時間のばらつき）による増加時間

(d) キャッシュ・ミスヒット時間

(e) メモリ競合時間（異なるプロセスが共有メモリを同時にアクセスしたときの待ち合わせ時間）

であると考えられる。

分析結果の詳細を表4.5に示す。ここで並列処理オーバーヘッド及び負荷分散の情報はアナライザー等の情報からの推定値である。表より2プロセッサにおいても4プロセッサにおいても理論予測値と実測値との差に最も大きく影響を与えるのがメモリ競合時間である。これはSX-3Rのメモリスループット性能が、2プロセッサの場合でも4プロセッサの場合でも同じであることがその原因で

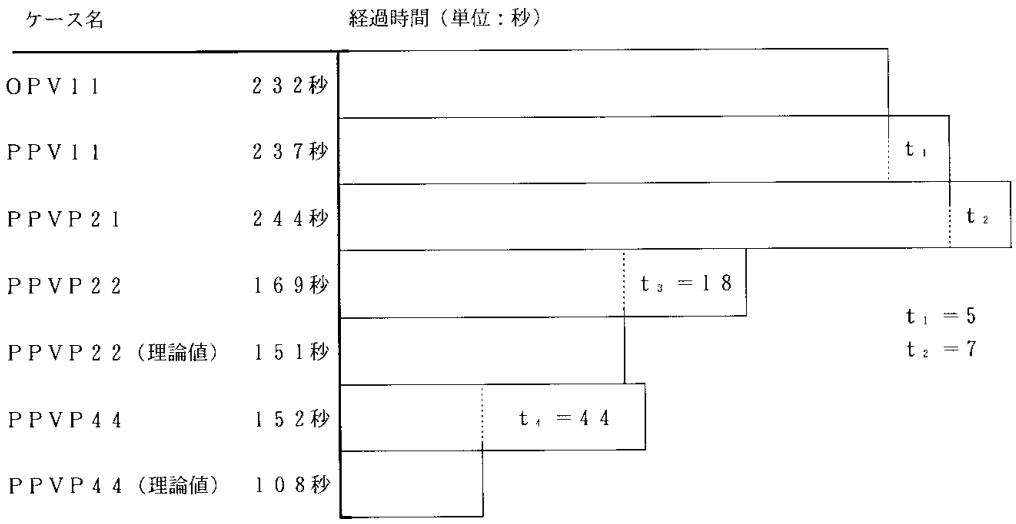


図4.2 プログラムIのケース毎の経過時間

表4.5 プログラムIの測定値分析結果（2）経過時間詳細分析（単位：秒）

ケース名	経過時間の差	経過時間の差の内訳				
		ソース修正	並列処理オーバーヘッド	負荷分散	キャッシュ・ミスヒット 時間	メモリ競合
OPV11	0 *1	—	—	—	—	—
PPV11	5 *1	5	—	—	—	—
PPVP21	12 *1	5	3×2	—	1	—
PPVP22	18 *2	5	3	1	4／2	7
PPVP44	44 *2	5	3	1	11/4	32

＊1：OPV11の経過時間との差である。
＊2：各々の理論値の経過時間との差である。
注） 並列処理オーバーヘッド及び負荷分散の情報はアナライザ等の情報からの推定値である。

ある。本プログラムの並列処理において2プロセッサの場合、プログラム処理の経過時間のメモリ競合時間の占める割合は全体の4%程度であるが、4プロセッサになると全体の21%にものぼり、理論値と実測値の差の時間の70%以上を占めている。SX-3Rの4プロセッサ時のメモリスループット性能が2プロセッサ時のそれの2倍のメモリスループットを持てば、メモリ競合による実行時間の低下は数%程度と予想されるので、4プロセッサ時の性能倍率は1.53から1.8程度に上がることが推測される。

4.3 プログラムIIの並列処理

(1) シングルプロセッサ処理

プログラムIIのサブルーチン単位のCPU負荷分布を図4.3に示す。

高負荷ループのループ長は平均200程度（50～800程度）と大きく、効率的なベクトル演算を行っている。ベクトル演算率も99.5%と十分ベクトル化されている。ただし、最もコストの高いFDETA2ルーチンの最内側DOループのループ長が平均50程度であることが、唯一ベクトル性能の効率を下げる要因となっている。

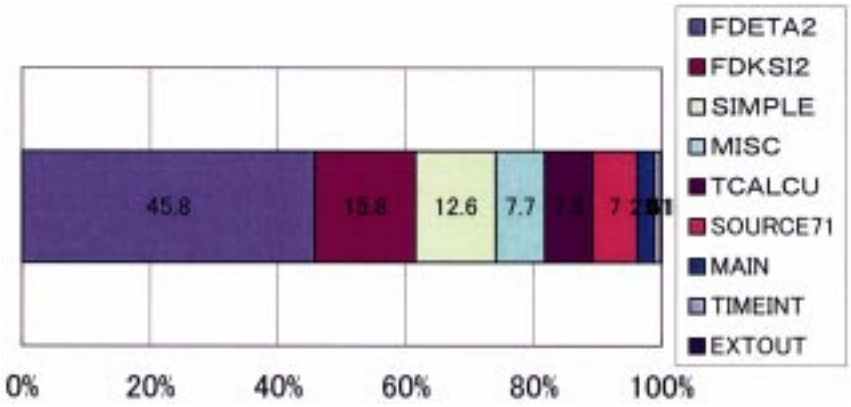


図4.3 プログラムⅡのCPU負荷分布

(2) マルチプロセッサ処理

本プログラムのマルチプロセッサ処理においては、プログラムⅡの全てのサブルーチン内のDOループにマイクロタスクによる並列処理を適用した。

以下に示すCPUコストの高い上位8ルーチンを並列化した。

- FDETA2
- FDKSI2
- SIMPLC
- MISC
- TCALCU
- SOURCE71
- MAIN
- TIMEINT

自動並列化機能を適用したが、表4.6に示すFDETA2, FDKSI2ルーチンのそれぞれのDOループにおいては表に示す並列化指示行の挿入を行った。

高コストルーチンの中の高コストループについて、各ループ長とベクトル／並列演算状況を表4.7に示す。FDETA2ルーチンでは、並列化対象となる外側DOループのループ長は799と大きく、並列処理における粒度の大小はこの大きさで決まる。また、内側の10個のDOループが一括して並列演算できるので、一層粒度が大きく、処理効率の良いものとなっている。

表4.7のその他のルーチンにおいても、ループ長が長く、ベクトル化又は並列化にとって効果的である。また、表3.2に示すようにSIMPLCルーチンのDOループにおいて一方向しか並列化可能でないDOループがあるが、並列化可能な最深DOループのループ長が約4万と非常に大きいループ長となっているので、並列ベクトル化が可能となり、並列化を促進している。

(3) シングルプロセッサ及びマルチプロセッサ処理結果の検討

以下に示す測定結果の時間はいずれも経過時間を示す。また、ケース名については表4.3にて説明したものと同様である。

本プログラムの並列化率は、PPVP21の並列処理プログラムのFORTRANアナライザの実行結果より95.1%であった。これより並列化による実行時間の理想的な理論性能倍率を求めると、2プロセッサの場合1.91倍、4プロセッサの場合3.49倍である。

表4.8に5つのケースの実行結果と理論性能の経過時間及び倍率を示す。理論性能倍率より算出した理論予測値

表4.6 プログラムⅡのチューニング

サブルーチン名	修正内容	修正箇所
ＦＤＥＴＡ２	並列化指示行による外側ループでの並列化（＊ＰＤＩＲ　ＰＡＲＤＯ　ＦＯＲ＝２）	ＤＯ　１　ループ直前に挿入
ＦＤＫＳＩ２	並列化指示行による外側ループでの並列化（＊ＰＤＩＲ　ＰＡＲＤＯ　ＦＯＲ＝２）	ＤＯ　１　ループ直前に挿入

表4.7 プログラムIIの並列処理

サブルーチン	高コストループ	ループ長	ベクトル／並列	備考
FDETA2	DO 1 DO 10 ・・・・・ DO 300	799 51 49	P V V	サブルーチン全体をループ1で並列化した。ループ1内に10個のベクトル化ループがある。
FDKSI2	DO 1 DO 10 ・・・・・ DO 300	49 801 799	P V V	サブルーチン全体をループ1で並列化した。ループ1内に10個のベクトル化ループがある。
SIMPLC	DO 1 DO 2 DO 3 DO 3 DO 4 DO 20 DO 21 DO 30 DO 30 DO 40	7 39247 4 39247 39247 6 39247 4 39247 39247	 PV PV PV PV P V PV	本サブルーチン内でループ長の明白な定義がある。そのループ長が40000程度で十分大きいので、「PV」で処理される。
TCALCU	DO 100 DO	49 801	P V	DO 100 内の5個の内側ループを自動融合し、ひとつの内側DOループとした。
MISC	DO 1 DO 1 DO 11 DO 11	51 801 51 801	P V P V	DO 1と DO 11 はループ融合され1つのDOループとなっている。
SOURCE 71	DO 1	39247	PV	サブルーチン内の他の5個のDOループも同様に同時並列・ベクトル化する。

注） P：並列化　　V：ベクトル化　　PV：並列ベクトル化

は、2プロセッサでは256秒であり、4プロセッサでは140秒である。実測による並列化効果は、2プロセッサの理論性能倍率1.91に対し、実測性能倍率1.75、4プロセッサの理論性能倍率3.49に対し、2.36である。

図4.4に各ケースの経過時間及び理論性能倍率による経過時間を示す。OPV11とPPV11の経過時間に差がない事から、並列処理のためのプログラム修正におけるオーバーヘッドはないことがわかる。したがって、理論値と実測値の差の原因は次の通りであることがわかる。

t_1は11秒であり、これは並列処理に関するオーバーヘッド時間積算値及びキャッシュ・ミスヒット時間である。

T_2及びT_3はそれぞれ24秒、67秒であり、これは（a）並列処理に関するオーバーヘッド時間、（b）負荷分散、（c）キャッシュ・ミスヒット時間、（d）メモリ競合時間である。

分析結果の詳細を表4.9に示す。表において、並列処理オーバーヘッド及び負荷分散の情報はアナライザー等の情報からの推定値である。

表4.8　プログラムIIの測定値分析結果（1）経過時間一覧表（単位：秒）

ケース名	測定値			理論値	
	経過時間	倍率	ｷｬｯｼｭ・ﾐｽﾋｯﾄ時間	経過時間	倍率
OPV11	489	1.00	24	—	—
PPV11	489	—	24	489	1.00
PPVP21	500	—	27	—	—
PPVP22	280	1.75	32	256	1.91
PPVP44	207	2.36	44	140	3.49

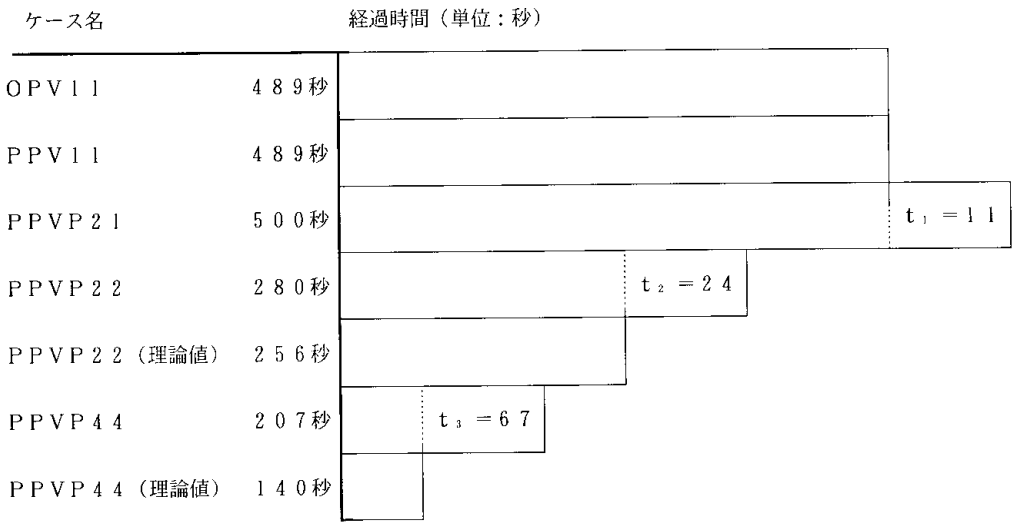


図4.4　プログラムIIのケース毎の経過時間

表からはプログラムIと同様に、メモリ競合時間に最も負荷のかかっていることがわかる。2プロセッサの場合、プログラム全経過時間の5%、4プロセッサでは全体の27%もあり、理論値と実測値の差の時間の80%以上を占めている。原因はプログラムIで述べたことと同様にメモリスループット性能によるものである。4プロセッサの場合のメモリスループットが2プロセッサの2倍であれば、メモリ競合による実行時間の低下は数%程度と予想されるので、4プロセッサ時の性能倍率は2.36から3.0に上ることが推測される。

4.4　まとめ

表4.10は、プログラムI及びプログラムIIのベクトル化率及び並列化率をまとめたものである。図4.5は両プログラムのプロセッサ数毎の測定値及び理論値を図示したものである。また、図4.6は両プログラムの速度向上率を示したものである。これらの図から、並列化率の大きいプログラム程並列化率から求めた理論倍率及び実測倍率がいずれのプロセッサ数の場合も大きいことがわかる。また、並列化率には関係なく、いずれのプロセッサ数の場合も実測倍率は理論倍率より小さい。

図4.7は両プログラムの測定値と理論値との経過時間の差の内訳を2プロセッサ及び4プロセッサについて百分率表示したものである。理論値と測定値との差は、並列化によるオリジナルプログラム修正によるもの、並列処理オーバーヘッド時間、負荷分散、キャッシュ・ミス

表4.9 プログラムIIの測定値分析結果（2）経過時間詳細分析（単位：秒）

ケース名	経過時間の差	経過時間の差の内訳				
		ソース修正	並列処理オーバーヘッド	負荷分散	キャッシュ・ミスヒット 時間	メモリ競合
OPV11	0 *1	—	—	—	—	—
PPV11	0 *1	0	—	—	—	—
PPVP21	11 *1	0	4×2	—	3	—
PPVP22	24 *2	0	4	3	8／2	13
PPVP44	67 *2	0	4	3	20/4	55

＊1：OPV11の経過時間との差である。

＊2：各々の理論値の経過時間との差である。

注） 並列処理オーバーヘッド及び負荷分散の情報はアナライザ等の情報からの推定値である。

表4.10 ベクトル化率及び並列化率

プログラム	プログラムI	プログラムII
ベクトル化率	99．5%	99．5%
並列化率	72．5%	95．1%

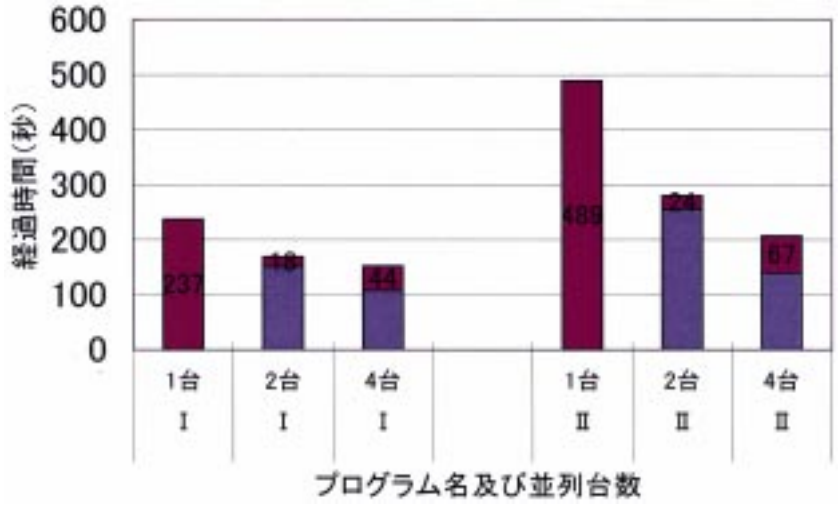


図4.5 プログラムI及びプログラムIIの並列台数毎の経過時間（秒）

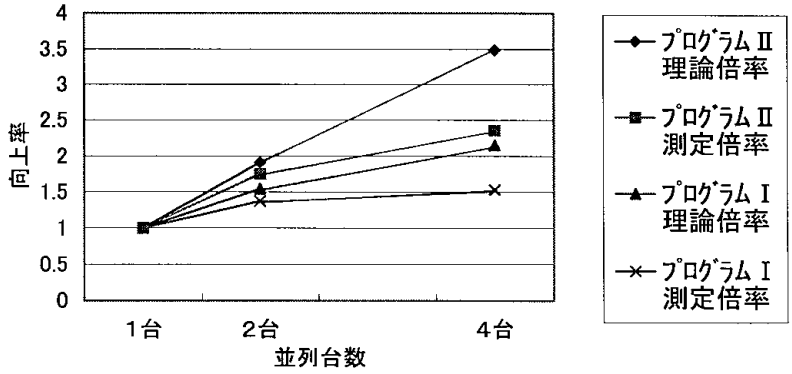


図4.6　プログラムⅠ及びプログラムⅡの速度向上率

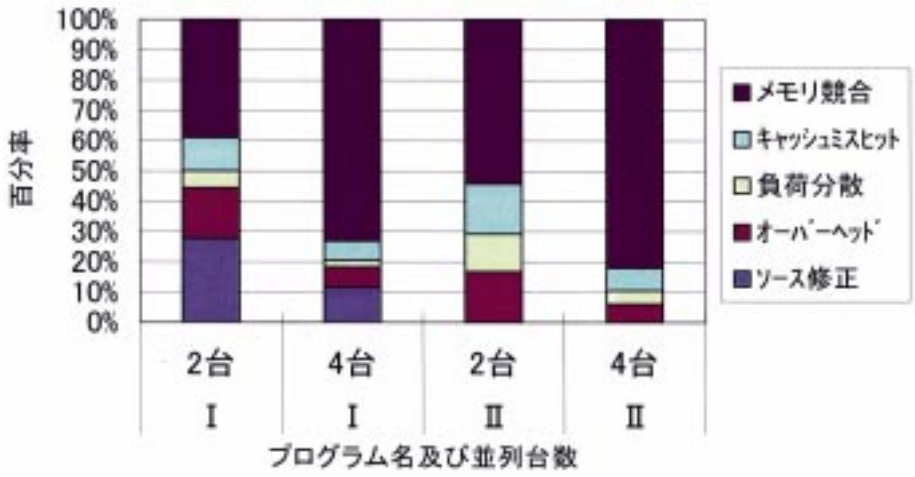


図4.7　プログラムⅠ及びⅡの経過時間詳細分析

ヒット時間、メモリ競合時間等により説明される。

表4.10及び図4.7からは、並列化率やプロセッサ数に関係なく経過時間の差の内訳はメモリ競合時間の割合が最も大きいことがわかる。また、メモリ競合時間はプロセッサ数が増大すると大きくなる。また、並列化率の大きいプログラムの方がプロセッサ数を増大させるほどメモリ競合による負荷が増大する。

プログラムⅠ及びプログラムⅡの並列化率はそれぞれ72.5%、95.1%であり、プログラムⅠの方がプログラムⅡより並列化率が小さい。この理由として以下のことが言える。

プログラムⅠはプログラムⅡに比較して、以下の特徴がある。

① 漸化式計算がある。
② 実行頻度の多いDOループの中に条件文判定によるGOTO文を含むものがあり、完全にはベクトル化されない。
③ 入出力命令の負担が大きい。
④ 全体的に粒度の小さいDOループが多い。

①の添え字繰り上げの漸化式計算を行うDOループ処理において、2重ループの場合、SX-3Rシステムでは通常外側ループを並列化するが、外側ループが添え字の繰り上げなどを行っている場合は並列化ができない。この場合、2つのループの入れ替えによって並列化はできるが、入れ替えの結果ベクトル化が阻害される場合は入れ替えをせず、ベクトル化のみで並列化しない方式が選ばれる。すなわち、ベクトル化しないで並列化するか、ベクトル化はするが並列化しないかのどちらかの選択になる。

この問題は2次元の数値シミュレーションを対象とする場合、避けて通れない。しかし、2次元問題であってもベクトル化可能なループ長が十分長ければ、内側ループをベクトル化と同時に並列化することで、性能向上できる場合がある。

また、3次元問題を対象とする数値シミュレーションであれば3重DOループとなり、3つのループの内2つのループがベクトル化、または、並列化可能になり、効率の良い処理ができる。

②の条件文判定によるGOTO文を含むループの回避に

ついては、以下のことが言える。

サブルーチンEDYVISにおいては、高コストループが条件文判定によるGOTO文を含んだループになっており、ほとんどベクトル化されていない。こうしたスカラ演算を行うループでは、配列参照の際のアドレスの飛びが大きいとキャッシュミスが起こり易いことから、なるべく連続にメモリを参照するようなコーディングが望ましい。

また、一般にGOTO文を含んだDOループはベクトル化は可能であるが、ネストが深いとベクトル化が困難になるので、ネストの深さは2か3程度が望ましい。(この場合、もし最終結果に影響がなければ判断文を外してベクトル化ができれば、余分な演算を行ったとしても速度向上が望める場合がある)

③の入出力命令の実行に関しては、次のことが言える。入出力文は並列化対象外であるのでこれの使用は並列化率を引き下げる。これを使用する場合には必要最小限に、そして、書式つき入出力文に比較して実行速度の速い書式なし入出力文を用いて頻度少なく行われることが望ましい。

④の粒度の大小に関しては以下のことが言える。SX-3Rシステムの最適化プリプロセッサは可能な限りループの融合を行う。これはベクトル化や並列化の対象となるループ内の演算量の増大により、粒度を高め、効率よくベクトル／並列演算を行うためである。オリジナルプログラムがある程度の粒度を持っている方が効率よく最適化が出来ることから、独立かつ一括計算できる部分を大きくするようなプログラム設計が望ましい。これはベクトル化と並列化の両方に効果的である。

以上述べてきたように、プログラムⅠはプログラムⅡに比較して並列処理を阻害する要因や並列効率を引き下げる特徴をもっている。裏返せば、プログラムⅡはプログラムⅠに比較して並列化の単位を示すDOループ内の演算数が多く、またループ長が長く、粒度が大きく、並列計算向きのプログラムであるといえる。

以上のように、プログラムはそれぞれ特徴があり、それらは並列処理や計算機アーキテクチャと共に処理性能を左右する。従って、プログラムの特徴について解析する必要性が生じる。

第5章　おわりに

本報告では、並列ベクトル計算機SX-3Rシステムの自動並列化機能を使用して、宇宙推進系プログラム2本の並列処理プログラムを作成し、プロセッサ数1台、2台及び4台での実行結果の比較・検討を行った。並列ベクトル計算機における自動並列化機能は、ユーザーのプログラムコーディングの負担が少なく、共有メモリ方式の並列ベクトル計算機においては、DOループを対象に実現されているが、分散メモリ方式のそれにおいては、未だ実現されていない。その自動並列化機能を用いて、以下の結果を得た。

2本のプログラムの内、並列化率が95.1%と高いプログラムⅡにおいては、4台のプロセッサ使用の場合、2.4倍の性能向上を、また、プログラムⅠにおいては、1.5倍の性能向上を得た。理論予測値と実測値との差に最も影響を与えたものはメモリ競合時間であった。

また、プログラムの並列処理における性能向上は計算機アーキテクチャとともに、プログラムの性質及び実行条件に左右される。今後、これらプログラムの特徴と計算機の性能との関係を分析する上でも、プログラム解析を行い、プログラムの特徴を把握する必要が生じよう。

最後に、本研究は、航空宇宙技術研究所角田宇宙推進技術研究センターと日本電気株式会社との間で、平成5年7月から平成6年12月までに行われた共同研究「宇宙推進系の研究開発に関する並列処理技術の研究」の成果の一部であることを補足する。また、本共同研究及び報告書執筆にあたり、日本電気株式会社、松本寛氏並びにNECソフトウェア、村瀬匡氏のご協力に感謝の意を表する。

参考文献

(1) SXシステムソフトウェア：SUPER-UX FORTRAN/SX並列処理機能利用の手引き

(2) 伊藤勝宏、高橋政浩、平岩徹夫；Pointwise Non-oscillatorスキームの開発と非定常・定常超音速流問題への応用、第6回数値流体力学シンポジウム、1992

航空宇宙技術研究所報告1430号

平成13年8月発行

発行所　独立行政法人　航空宇宙技術研究所
東京都調布市深大寺東町7-44-1
電話（0422）40-3935　　〒182-8522
印刷所　株式会社　実業公報社
東京都千代田区九段北1-7-8

Printed in Japan