

FORTRANベクトルコンパイラとチューニングソフトウェア

FORTRAN Vectorizing Compilers and Tuning Software Tools

スーパーコンピュータの需要の拡大に伴い、高速化、使いやすさへの要求はますます強くなっている。このようなニーズにこたえるには、高速のハードウェアとともに、コンパイラやチューニング支援ツールなどのソフトウェアの充実が必要不可欠である。

FORTRANベクトルコンパイラは、HITAC S-810用として開発し、高度な自動ベクトル化、効率の良いベクトル命令の生成、高速入出力などの高い機能を備えている。今回、最新鋭のHITAC S-820の性能を最大限に引き出すことを目的として、これらの機能の強化、拡張を図った。この結果、HITAC S-820では、従来のHITAC S-810に比べて大幅な実行性能の向上を実現することができた。

また、スーパーコンピュータの能力をよりきめ細かく引き出すためには、ユーザーによるプログラムのチューニングが必要となる。このような作業の効率化を促進するために、対話形のベクトルチューニング支援ツールを新たに開発し、提供している。

長堀 正* *Tadashi Nagahori*
田中義一** *Yoshikazu Tanaka*
積谷利夫* *Toshio Sekiya*

1 緒 言

大規模科学技術計算の高速化への要求に伴い、スーパーコンピュータシステムに対する性能向上、使いやすさへの要望はますます増大しつつある。このようなニーズにこたえるためには、より速いハードウェアの開発とともに、自動ベクトル化^{※1)}機能を持つ高性能コンパイラや、チューニング支援ツールなどのソフトウェアの強化、拡充が必ず(須)となっている。FORTRANベクトルコンパイラHAP FORTRAN 77(以下、FORT77/HAPと呼ぶ。)では、最新鋭のスーパーコンピュータHITAC S-820(以下、S-820と略す。)の超高速演算能力を最大限に引き出すことを目的として、従来のHITAC S-810⁵⁾(以下、S-810と略す。)で開発した自動ベクトル化⁴⁾、効率的なベクトル命令オブジェクトの生成、高速入出力などの機能を強化、拡張した。

一般に、自動ベクトルコンパイラでは、実行性能を向上させるために次の2点が重要課題となる。

- (1) プログラムのベクトル化率^{※2)}を上げる。
- (2) ベクトル化部分の加速率(スカラー命令で実行した場合に対してベクトル命令で実行した場合の速度比)を上げる。

FORT77/HAPでは、ソースプログラムの解析能力を大幅に強化するとともに、新しいベクトル化技術を取り入れ自動ベクトル化の範囲を拡大した。この結果、多重DOループや複雑な制御構造を持つDOループなどもベクトル化の対象となり、性能の向上に大きく貢献している。

また、S-820のハードウェアアーキテクチャに合った最適なオブジェクトを生成する機能を新たに追加した。これにより、ベクトル命令の効率化、ベクトルとスカラーの並列動作の強化などの効果から、ベクトル化部分の加速率はS-810のときに比べて格段に向上している。

更に、スーパーコンピュータの性能を極限まで発揮させるためには、ユーザーによるプログラムのチューニング^{※3)}が必

※1) 自動ベクトル化：コンパイラがソースプログラムを解析して、繰返し演算の中で並列に実行可能な部分を検出し、自動的にベクトル命令に変換すること。
※2) ベクトル化率：プログラム全体をはん(汎)用のスカラー計算機で実行したときの実行時間のうち、ベクトル化によりベクトル命令で実行可能な時間の比率として定義される。
※3) チューニング：ソースプログラムを書き替えることによって、コンパイラがベクトル化できない部分をベクトル化可能とし

たり、より効率的なベクトル命令へ変換できるように操作したりすること。FORT77/HAP用のチューニング方法としては、次の二通りがある。

- (1) ソースプログラムを直接修正する。
- (2) *VOPTION文(ベクトル化注釈行)でコンパイラへ必要な情報を通知する。

要となる場合が多い。一方、チューニング作業には専門的な知識とノウハウが必要であり、どこをどう書き替えればよいかは、一般ユーザーにはなかなか分からない。対話形ベクトルチューニング支援FORT/VF(FORTRAN Vectorizing Facility)は、このようなユーザーの負担を軽減するために開発したチューニング支援ツールであり、効率的で使い勝手のよい作業環境を提供する。

本稿では、FORT77/HAPの自動ベクトル化のうち、機能強化部分に焦点を当てて説明する。これらの機能は、S-810、S-820両機種に適用される。更に、S-820アーキテクチャに適合した高速化機能について述べる。また、FORT/VFをはじめとする支援ツールについて、その機能を紹介する。

2 FORT77/HAPの自動ベクトル化機能

2.1 自動ベクトル化

S-810及びS-820は、ベクトル命令をパイプライン方式で高速実行することによって、はん(汎)用のスカラー計算機に対して大幅な実行時間の短縮を実現するベクトル計算機¹⁾である。これらの計算機では、プログラム全体のうち、どれだけの部分をベクトル命令で実行できるか、すなわちベクトル化率の大小が実行速度を決定的に左右する。ベクトル化率を α 、ベクトル化部分についての、スカラー命令で実行した場合に対するベクトル命令で実行した場合の速度比を k とすると、ベクトル計算機での実行速度向上比 p は、

$$p = 1 / ((1 - \alpha) + \alpha / k)$$

という式で表される。

ベクトル化率が50%では、 k がどれほど大きくても、速度向上比 p はたかだか2倍程度である。一方、10倍以上の高速化を達成するためには、 $k=30$ であればベクトル化率は約93%以上が必要になる。このように、ベクトル計算機の高速処理能力を十分に引き出すには、高いベクトル化率が必要不可欠であり、コンパイラの自動ベクトル化技術は極めて重要である。

高い自動ベクトル化を実現するためには、

- (1) プログラム中のデータの依存関係(定義、参照の関係)を解析して情報を収集する。
- (2) 依存関係の情報をもとに、ベクトル化可能部分を検出し、ベクトル命令を生成する。

の二つを実現する強力な機能が必要である。FORT77/HAPでは、データの依存関係の解析能力を強化し、多重DOループ間のデータ関係まで十分に把握できるようにした。また、IF文などによって制御構造が複雑な場合にも、データの流れを解析できるようにした。これらの解析情報を使用して、特に多重ループのベクトル化機能を強化し、従来と比較して大幅なベクトル化率の向上を実現している。

表1にFORT77/HAPの主要な自動ベクトル化機能項目を示す。

2.2 単一DOループのベクトル化機能

多重ループの最内側や単独のループなどでの、そのループ内だけを対象とする基本的ベクトル化機能としては、IF文を含むベクトル化、リストベクトルによるベクトル化、ベクトルマクロ命令(積和、総和、一次巡回演算、最大値、最小値、

表1 自動ベクトル化の機能項目 FORT77/HAPでは、単一DOループの多様なベクトル化機能に加え、多重ループのベクトル化を大幅に向上している。

分類	ベクトル化機能
単一DOループのベクトル化機能	IF文を含むDOループのベクトル化
	組込み関数のベクトル化
	リストベクトルを添字に持つ配列のベクトル化
	ループ外飛び出しを含むDOループのベクトル化
	ループ外のデータの流れを利用したベクトル化
多重DOループのベクトル化機能	DOループ内の演算の並べ替えによる非ベクトル化部分の局所化
	ループ分割
	ループ交換
	ループ一重化
	ループ展開

探索、圧縮、伸長操作の命令)による複合演算のベクトル化などがある。また、データ解析能力の強化によって、DOループ外のデータの流れを利用してベクトル化の判定をする機能なども備えている。

このうち、ベクトルマクロ命令の適用は、行列計算や差分法などの数値解析アルゴリズムに対して効果が大きいことが知られている^{1), 2)}。今回の機能強化では、命令の適用範囲を従来よりも広げ、ベクトル化率の向上を図っている。図1に、一次巡回演算命令VITR(Vector Iteration)の適用拡大によりベクトル化が可能となった例を示す。従来VITRは一文内でしか適用されなかったが、複数文にまで対象範囲を広げたため、SOR法(Successive Overrelaxation Iteration Method: 逐次過大緩和法)など一次巡回形の計算となるアルゴリズムのベクトル化がより容易となった。

2.3 多重DOループのベクトル化機能

多重ループのベクトル化機能としては、ループ交換、ループ一重化、ループ分割及びループ展開(アンローリング)がある。これらの機能は、ループ構造の変換を伴うものであり、前述の単一ループのベクトル化機能と組み合わせて適用することにより、高いベクトル化率と実行性能の向上を得ることができる。

ループ交換は、最内側と外側のループを交換することによってベクトル化を促進する方法である。最内側に関してはベクトル化不能であるが、外側ループについてはベクトル化可能のときループを交換する。また、内側ループに対して外側ループのループ長が大きい場合にも、コンパイラが判断してループを交換し、実行速度の向上を図ることがある。

図2に、ループ交換によるベクトル化の例を示す。

多重ループを一重ループに変換してループ長を大きくすることは、ベクトル命令の加速率を向上するために重要である。S-820ではS-810に比べて、ベクトル命令の立上り速度が大幅に改善され、小さいループ長でも高い加速率が得られるようになった。しかし、ループ長が10前後までのごく小さい範囲

では、S-820でも本来の能力を十分引き出すことができない。このため、コンパイラでは、可能な限り一重化を行ってループ長の増大を図っている。

FORT77/HAPでは、通常の一重化変換のほかにリストベクトルを用いた一重化も実施している。これにより、図3のような、行列計算によく出現する内側ループの範囲が外側ループの制御変数に依存するようなケースも一重化が可能となっている。

```

DO 10 J=2, 6
DO 10 I=2, KN
QA=ZA (I, J+1) * ZR (I, J)
+ZA (I, J-1) * ZB (I, J)
+ZA (I+1, J) * ZU (I, J)
+ZA (I-1, J) * ZV (I, J)
+ZZ (I, J)
ZA (I, J) =ZA (I, J)
+. 175 * (QA-ZA (I, J))
10 CONTINUE

```

↓

```

DO 10 J=2, 6
DO 10 I=2, KN
VITR命令を適用
ZA (I, J) = . 175 *
(ZA (I-1, J) * ZV (I, J) ) +ZA (I, J)
+. 175 *
(ZA (I, J+1) * ZR (I, J)
+ZA (I, J-1) * ZB (I, J)
+ZA (I+1, J) * ZU (I, J)
+ZZ (I, J)
-ZA (I, J) )
10 CONTINUE

```

図1 一次巡回演算命令VITRによるベクトル化の例 VITR命令は、従来一文内に出現するパターンにしか適用されなかったが、複数文にまで範囲を拡大した。本例では、QAを展開し、演算を並べ替えて適用パターンを検出している。

```

DO 10 J=1, N
DO 20 I=1, N
A (I, J) =A (I-1, J) +B (I, J)
20 CONTINUE
10 CONTINUE

```

↓

```

DO 20 I=1, N
DO 10 J=1, N
A (I, J) =A (I-1, J) +B (I, J)
10 CONTINUE
20 CONTINUE

```

図2 ループ交換によりベクトル化が可能となる例 Iのループではデータの依存関係によりベクトル化不能であるが、Jについてはベクトル化可能であるため、ループ交換する。

DOループ中にベクトル化可能な部分と不能な部分が混在しているときには、ループ分割の手法によりベクトル化率を向上することができる。FORT77/HAPでは、ループ間のデータ解析能力の強化によって、多重ループの場合でもきめ細かくループ分割を行うことが可能である。図4に、連立一次方程式をガウスの消去法で計算するプログラム³⁾をループ分割によりベクトル化している例を示す。

ループ展開は、DOループ中の演算量が少ないときに式を複

```

DO 10 J=1, 3
DO 20 I=1, J
A (I, J) =I
20 CONTINUE
10 CONTINUE

```

↓

```

INTEGER LI (6) /1,1,2,1,2,3/
INTEGER LJ (6) /1,2,2,3,3,3/
DO 30 K=1, 6
A (LI (K), LJ (K)) =LI (K)
30 CONTINUE

```

図3 リストベクトルを使用したループ一重化 一重化すると配列の参照が非連続になり、通常は一重化ができない場合でも、コンパイラがリストベクトルを生成することによって一重化が可能となる。

```

DO 100 J=K+1, N
IF (IPK. NE. K) THEN
W=A (IPK, J)
A (IPK, J) =A (K, J)
A (K, J) =W
END IF
DO 110 I=K+1, N
A (I, J) =A (I, J) +A (I, K) *A (K, J)
110 CONTINUE
100 CONTINUE

```

↓

```

DO 101 J=K+1, N
IF (IPK. NE. K) THEN
W=A (IPK, J)
A (IPK, J) =A (K, J)
A (K, J) =W
END IF
101 CONTINUE

```

ループ分割

```

DO 102 J=K+1, N, 2
DO 110 I=K+1, N
A (I, J) =A (I, J) +A (I, K) *A (K, J)
A (I, J+1) =A (I, J+1) +A (I, K) *A (K, J+1)
110 CONTINUE
102 CONTINUE

```

ベクトル化

図4 ループ分割によるベクトル化の例 配列Aについてループ間の依存関係を解析できるようになったため、ループ分割によるベクトル化が可能になった。

数個に展開した形に変換することによって、ループによる分岐オーバーヘッドを削減し、同時にベクトル演算パイプラインの並列動作による高速化を図るものである。この手法は、特にS-810のように多数の演算器を持ち、オブジェクトプログラムから並列動作を制御できる場合には効果が大きい。S-810/20では、6本のロード、2本のロード/ストア、4本の加算、及び4本の乗算のパイプラインがある。これらを有効に活用するために、コンパイラではDOループ内の演算量とループ長を見て、最大8倍までDOループを展開している。

一方、S-820では、要素並列パイプライン方式を採用しているため、オブジェクトプログラムから制御できるパイプラインはロード一組、ロード/ストア一組、加算、乗算各一組であり、展開数を多くしても性能上の効果が薄い。このため、展開数は4倍までに制限して不要にオブジェクトが増大するのを防いでいる。

ループ展開は、通常は多重ループの外側に関して実施し、

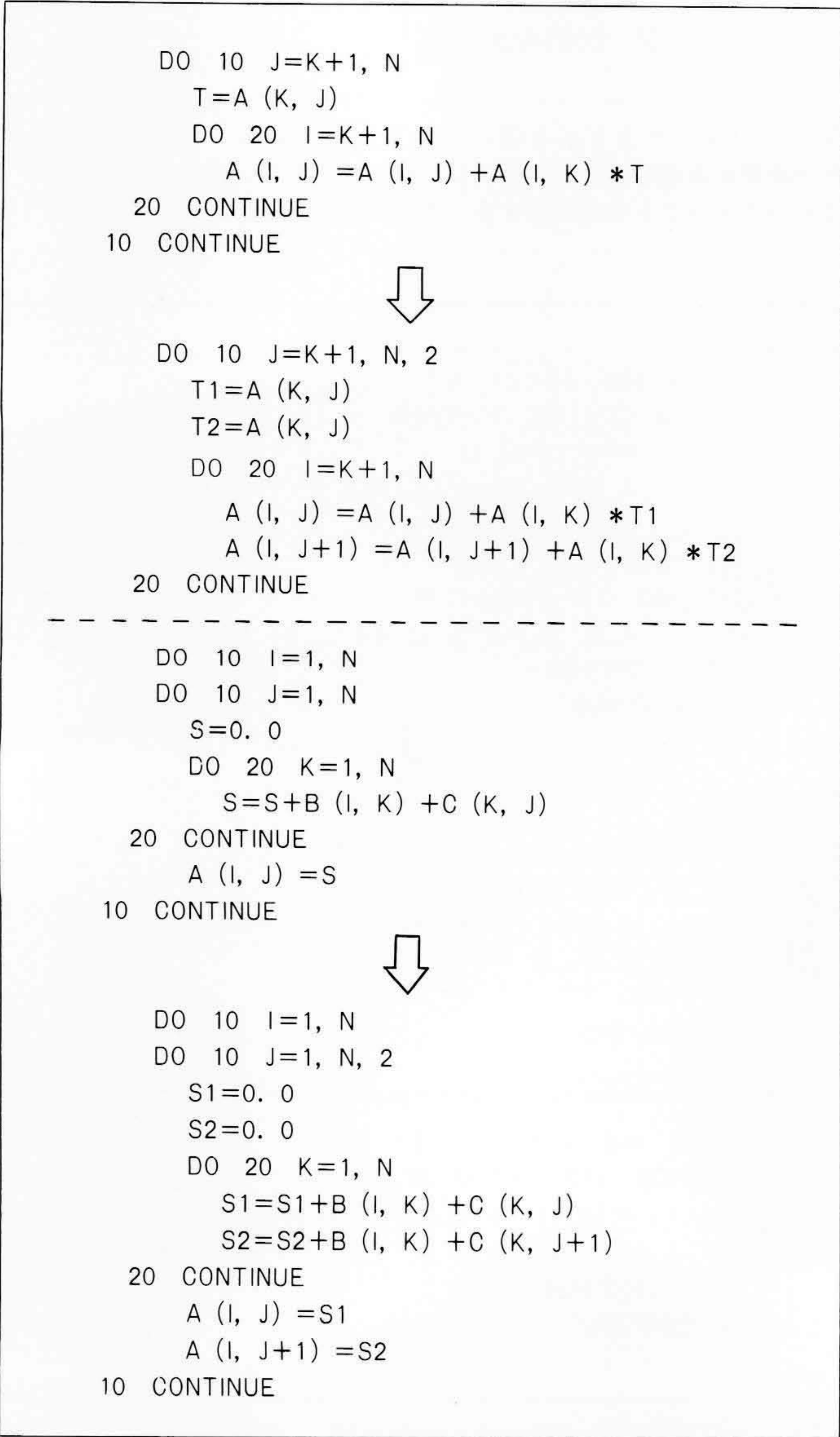


図5 ループ展開の適用例 密多重ループでない場合も、ループ展開するように機能を拡張した。

表2 リバモア14カーネルループによる性能比較 ベクトル化の効果を明らかにするため、新旧コンパイラを同じハードウェア(S-810/20)で測定した。ベクトル化の強化により、1.24倍実行性能が向上している。

ハードウェア機種	S-810/20	S-810/20
コンパイラのバージョン	FORT77/HAP 02-02 (旧バージョン)	FORT77/HAP 20-00/A (ベクトル化強化版)
カーネル 1	252.6	304.8
カーネル 2	272.8	276.7
カーネル 3	217.7	341.6
カーネル 4	73.1	66.8
カーネル 5	4.9	23.6
カーネル 6	4.4	26.2
カーネル 7	283.5	293.5
カーネル 8	8.8	116.8
カーネル 9	255.8	271.1
カーネル10	66.0	68.0
カーネル11	9.9	16.8
カーネル12	100.4	113.2
カーネル13	4.7	6.1
カーネル14	8.5	12.5
平均	111.6	137.9

単位:MFLOPS (Million Floating Operations Per Second)

ループ長を減少させずにパイプライン並列動作と分岐オーバーヘッド削減を達成する。この場合、従来は密多重ループ、すなわち外側ループと内側ループの間に実行文が存在しないような多重ループしか展開を適用していなかった。したがって、ベクトル化されない部分が外側ループに残っていると、前述の多重ループの分割機能を使っても密多重ループが形成されず、ループ展開が適用されないケースがあった。今回はこの部分についても改良している。図5にこの例を示す。

このように、多重ループを含めたベクトル化の強化によってユーザープログラムのベクトル化率は従来よりも更に向上し、実行時間の短縮を実現している。これらの性能向上の効果を、ベンチマークジョブとして著名なリバモア14カーネルループの測定結果(表2)によって示す。

3 S-820用の高速化機能

FORT77/HAPでは、S-820のハードウェア特性を十分に考慮することによって、オブジェクトプログラムの実行速度を向上させる機能を新たに追加している。追加した主な機能は以下のとおりである。

- (1) ベクトル命令の分割起動による並列動作の強化
- (2) 新設命令の使用による実行速度の改善
- (3) ベクトル組込み関数の高速化
- (4) ベクトルレジスタ割当ての最適化
- (5) 拡張記憶入出力機能の強化

なお、従来のS-810用に作成されたオブジェクトプログラムでも、S-820では互換性があるため動作は可能である。しかし、S-820の持つ高速演算能力を十分に発揮するためには、再コンパイルにより上記機能を使用してS-820用の効率的なオブジェクトを生成する必要がある。

3.1 ベクトル命令の分割起動

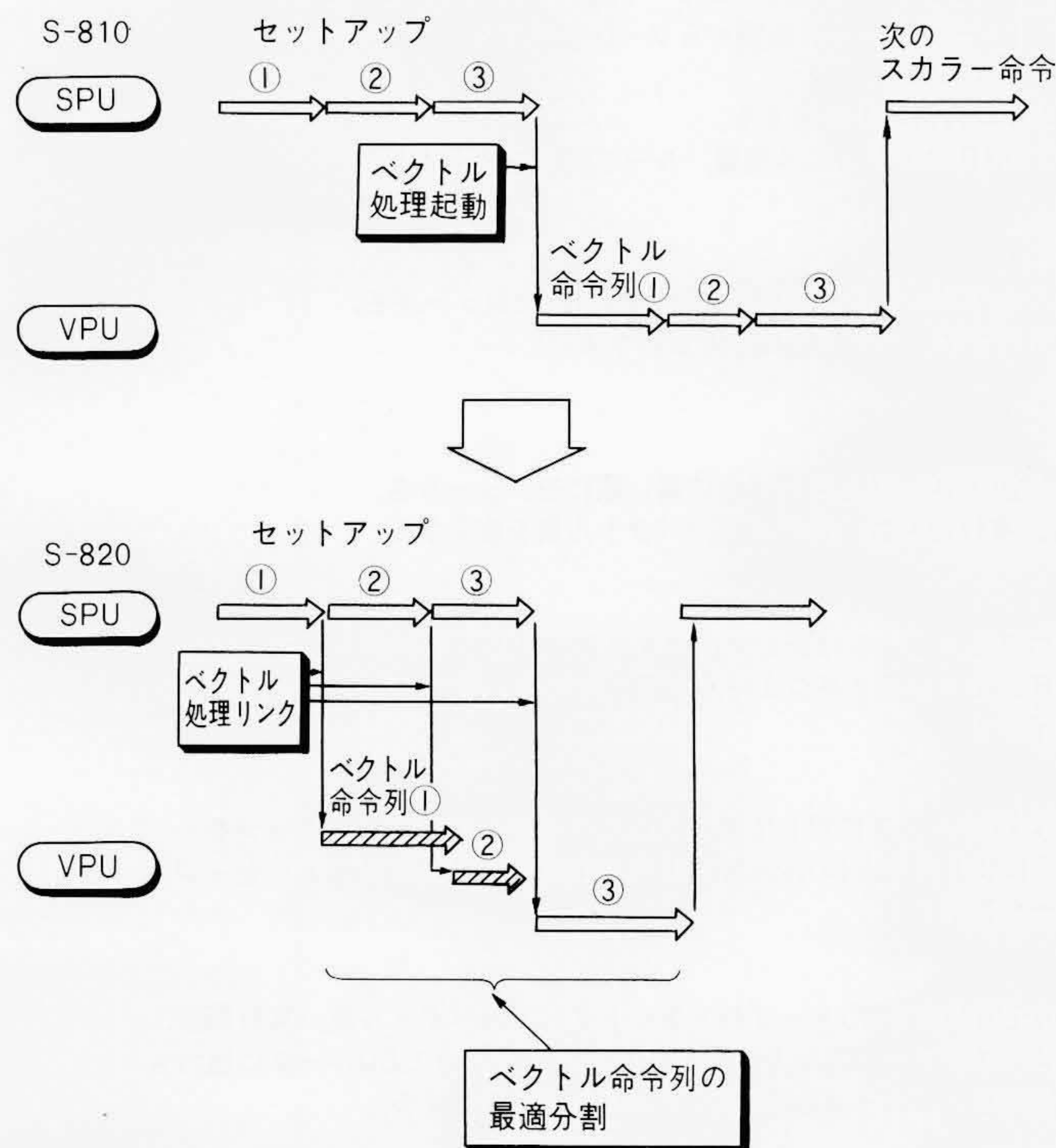
S-820では、スカラープロセッサからベクトルプロセッサを起動する方法として、従来のベクトル処理起動のほかにベクトル処理リンク方式が追加された。この方式では、以前のベクトル命令列の実行の完了を待たずに次のベクトル命令列を開始できるため、ベクトル命令の開始時のオーバーヘッドを削減できる。

コンパイラは、この起動方式を利用してスカラー処理側でのベクトル処理準備(セットアップ)とベクトル命令の実行の並列化を促進する。すなわち、ベクトル命令列を最適に分割し、必要なセットアップが終わりしだい、次々にリンク方式により起動するようなオブジェクトプログラムを生成する。この並列化により、実行時間の大幅な短縮が実現される。分割起動方式の概念を図6に示す。

3.2 拡張記憶入出力機能の強化

大規模な数値計算では、主記憶装置に一度に格納できないような大量のデータを扱う場合が多い。このような場合には、補助記憶装置との入出力が発生するために実行経過時間が大きく低下し、スーパーコンピュータを使うことによる実用上の効果が半減してしまう。S-810及びS-820では、拡張記憶装置を付加することにより入出力を高速化し、これに対処している。S-820では、S-810での使用実績による効果が大きいことから、拡張記憶装置を更に強化し、大容量、高速化を図った。

拡張記憶装置は、FORTRANの標準の入出力文により使用



注：略語説明 SPU (Scalar Processing Unit)
VPU (Vector Processing Unit)

図6 ベクトル命令の分割起動による並列動作の向上 S-820では、ベクトル命令を分割起動することによって、スカラー側でのセットアップとベクトル命令の実行を並列処理する。

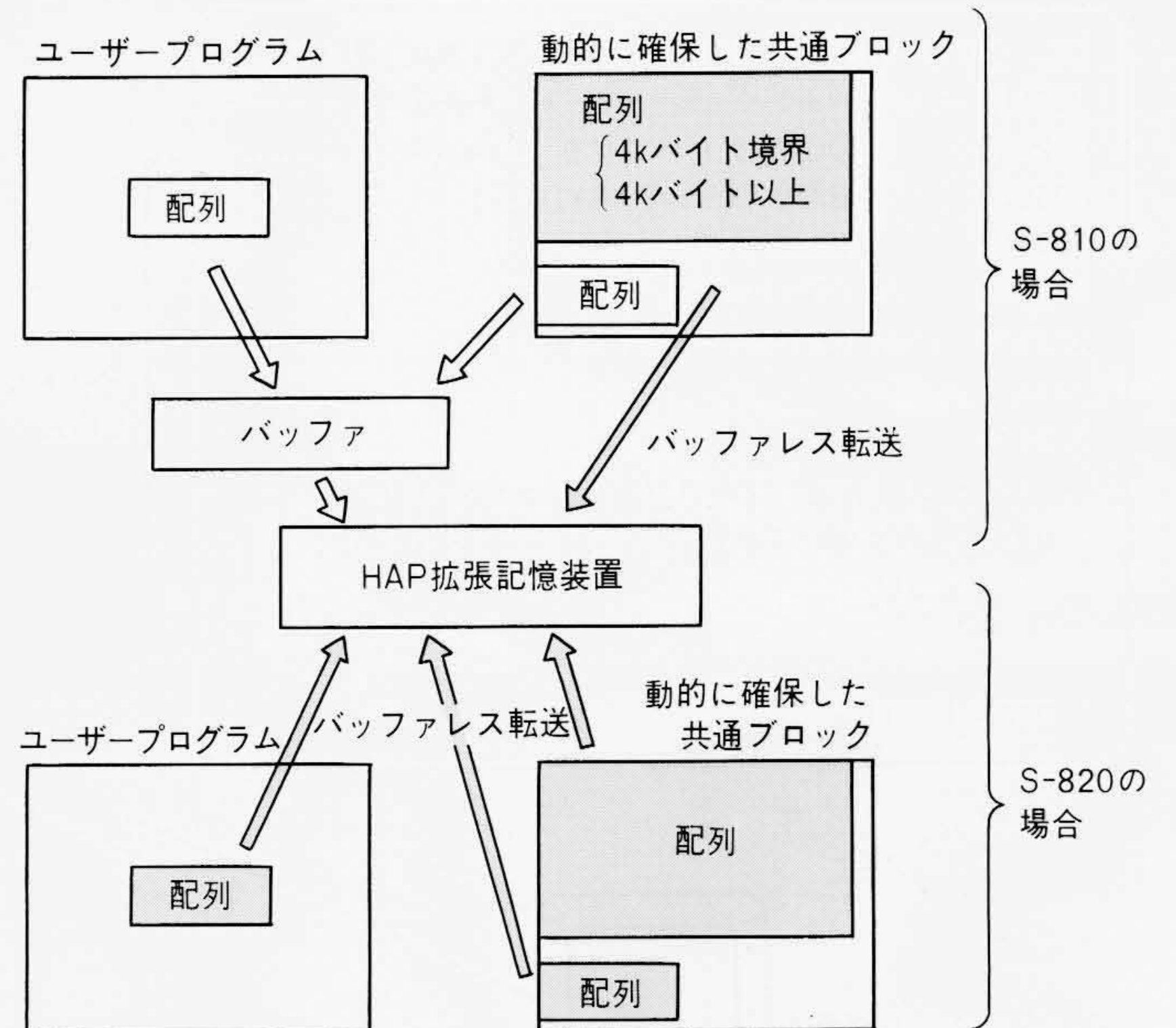


図7 バッファレス転送の適用 S-820では、拡張記憶装置に対してバッファレス転送できる範囲を拡大し、高速化を図っている。

することができる。拡張記憶装置を有効に使うためには主記憶装置とのデータ転送をバッファを経由しない(バッファレス転送)ようにする必要がある。

従来は、主記憶上の領域が4kバイト境界にないとバッファレス転送ができなかったが、S-820ではこれを緩和し、8バイト境界にあればよいように機能を拡張して高速化を実現している。

4 チューニングソフトウェア

スーパーコンピュータやFORTRANベクトルコンパイラの性能が向上しても、コンパイル時に決まらない要因があるなどの理由によりベクトル化できないことがある。また、ベクトル化に適したコーディングに不慣れな利用者も多い。そこで、人手によるチューニング作業が必要になる。

チューニングソフトウェアは、この人手の作業を軽減させるための支援システムであり、今回、FORTRAN開発支援システムFORT/ASSIST(FORTRAN Assist)と、対話形ベクトルチューニング支援FORT/VFを開発した。ここでは、チューニング作業について詳細を述べる。

4.1 チューニングソフトウェアの構成

チューニングソフトウェアはVOS3(Virtual-storage Operating System 3)の対話環境で稼動し、FORTRANプログラムの開発支援システムとしてASPEN E2, FORT77/TD, FORT/ASSIST, 及びFORT/VF(図8内略語説明参照)から構成されている。これはFORTRANプログラムのコンパイル、デバッグ及びチューニングができる総合システムであり、各プログラムの組合せにより、チューニング機能を実現する。図8に、チューニングソフトウェアの構成を示す。

4.2 機能概要

チューニングソフトウェアがサポートする機能の概要と機能実現に当たり必要となるプログラムについて次に述べる。

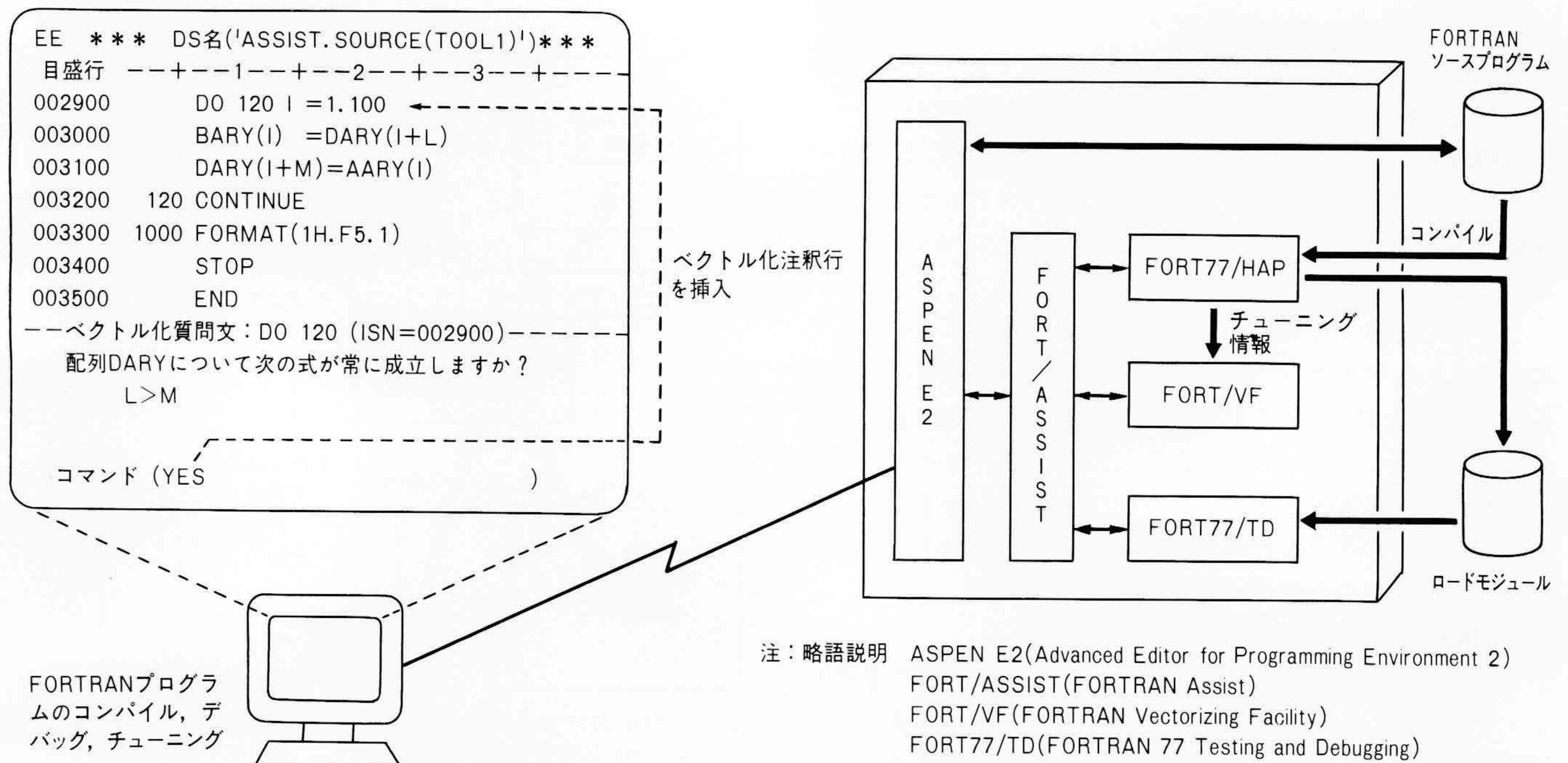


図8 チューニングソフトウェアの構成 チューニングソフトウェアは、ASPEN E2の下でFORT/ASSIST, FORT/VK, FORT77/TDなどから構成される。

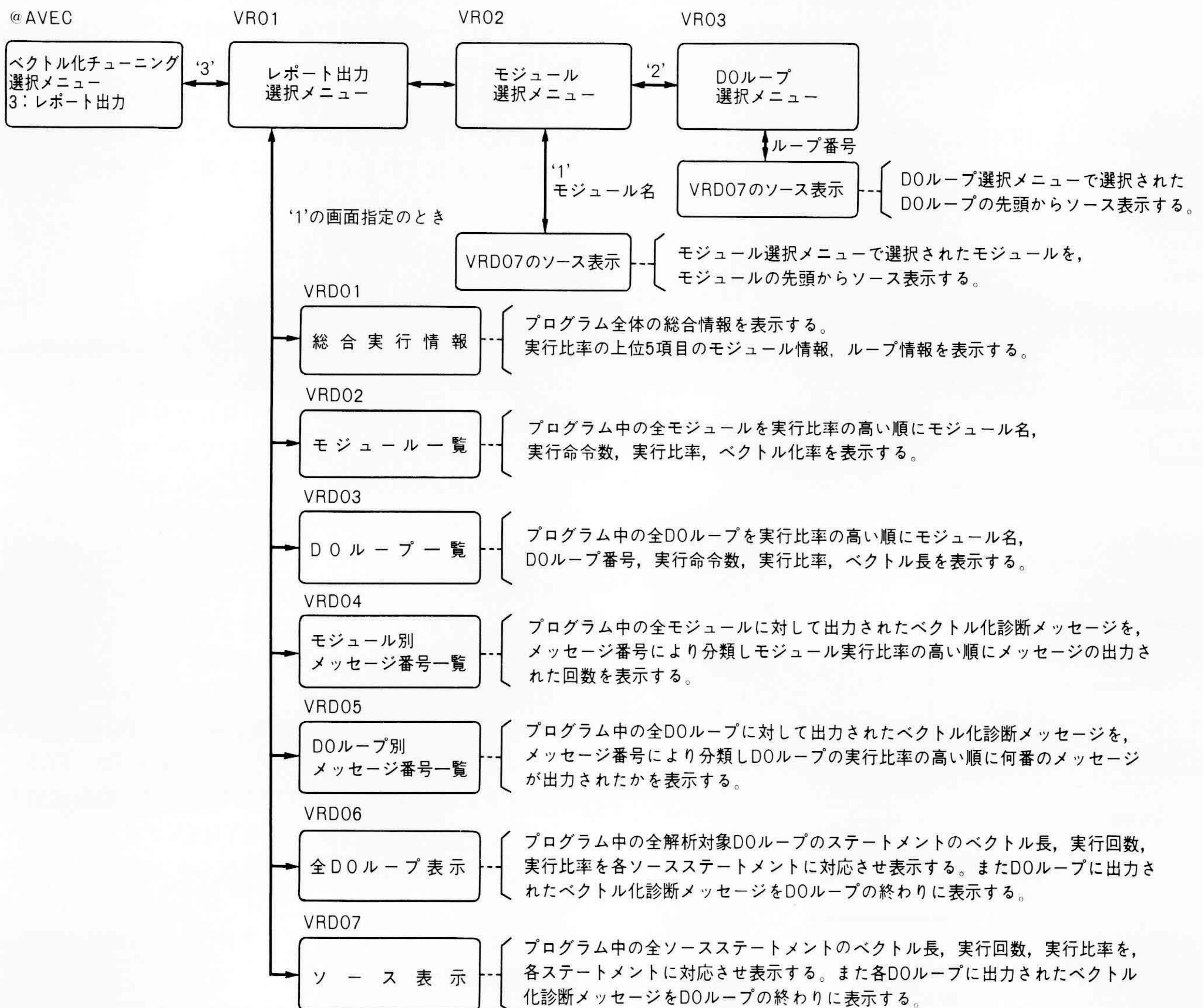


図9 レポート出力時の画面遷移 FORT/VFのレポート出力機能について、TSS画面上の遷移を示す。

(1) コンパイル・解析実行機能

FORT/ASSISTとFORT77/HAPの機能により実現する。画面から指定されたデータセット名やオプションに従って、FORTRANプログラムのコンパイルやジョブ制御文の作成ができる。FORTRAN専用のコンパイル画面のため、コンパイラ用オプションなどの指定も可能である。

(2) テスト・デバッグ機能

FORT77/TDの機能により実現する。画面上にソースプログラムを表示し、実行中のステートメントを高輝度で表示する。利用者は、プログラムの動きを目で追いながら、必要な時点で停止させて種々のデバッグ情報を得ることができる。

(3) 対話形ベクトルチューニング機能

```

EE ***** D S 名('GR11656.PSE.TP(TFLPS2)')*****
目盛行 -----1-----2-----3-----4-----5-----6-----7--
001700      DO 100 I = 1,10
001800      DARY(I) = K * L
001900      WRITE(*,1000) DARY(I)
002000      100 CONTINUE
002100      1000 FORMAT(1H ,F5.1)
002200      STOP
002300      END
***** ** データ行の末尾 **
-----ベクトル化チューニング参考例：DO 100 (ISN=001700)-----
ベクトル化対象外の文をループ外へ移す。
      ( 変更前 )      |      ( 変更後 )
      DO 10 I=1,100    |      READ(5,*)((B(I),C(I)),I=1,100)
      READ(5,*) B(I),C(I) |      DO 10 I=1,100
      10 A(I)=B(I)+C(I)  |      10 A(I)=B(I)+C(I)
                        |
VL : チューニンググループ選択 ; VMSG : 診断メッセージ表示
(VUP/VD) : 参考例の上下スクロール
コマンド ( )

```

図10 ベクトル化診断メッセージの例 ベクトル化診断メッセージを見ながら、ソースプログラムの修正が可能である。

```

EE ***** D S 名('GR11656.PSE.TP(TFLPS2)')*****
目盛行 -----1-----2-----3-----4-----5-----6-----7--
001700      DO 100 I = 1,10
001800      DARY(I) = K * L
001900      WRITE(*,1000) DARY(I)
002000      100 CONTINUE
002100      1000 FORMAT(1H ,F5.1)
002200      STOP
002300      END
***** ** データ行の末尾 **
-----ベクトル化チューニング参考例：DO 100 (ISN=001700)-----
ベクトル化対象外の文をループ外へ移す。
      ( 変更前 )      |      ( 変更後 )
      DO 10 I=1,100    |      READ(5,*)((B(I),C(I)),I=1,100)
      READ(5,*) B(I),C(I) |      DO 10 I=1,100
      10 A(I)=B(I)+C(I)  |      10 A(I)=B(I)+C(I)
                        |
VL : チューニンググループ選択 ; VMSG : 診断メッセージ表示
(VUP/VD) : 参考例の上下スクロール
コマンド ( )

```

図11 ベクトル化チューニング参考例の出力例 ベクトル化できない要因に対する適切なチューニング例を表示する。

FORT/VFの機能により実現する。コンパイルや解析実行で収集した情報により、ベクトル化を妨げる要因や、ベクトル化率を向上させるための情報を表示する。利用者は、これらの情報をもとにベクトルチューニングをすることができる。なお、チューニング中に表示するソース情報は、画面上で直接修正ができる。

4.3 対話形ベクトルチューニング機能の詳細

対話形ベクトルチューニング機能には、コンパイルや解析実行で収集した情報を画面やリストに出力するレポート機能と、ベクトル化率を向上させるための情報を表示し、ソース修正ができるチューニング機能がある。

(1) レポート機能

レポート機能は、コンパイルや解析実行で収集した7種類のレポートを、端末の画面あるいはリストに出力する。レポート出力時の画面遷移、及び出力するレポートの内容を図9に示す。

(2) チューニング機能

利用者が効率よくベクトルチューニングを実施できるように、以下の機能を用意している。

(a) ベクトル化診断メッセージの表示

実行比率の高い順に表示されたモジュールとDOループに対し、チューニング対象を選択すると、そのDOループのソース情報とベクトル化診断メッセージを同一の画面に表示する(図10)。

(b) ベクトルチューニング参考例の表示

DOループのソース情報とそのDOループのベクトル化診断メッセージに対応する修正例を、チューニング参考例として画面下部に表示する(図11)。利用者は参考例を見ながら、画面上で直接ソース修正ができる。

(c) ベクトルチューニング質問文の表示

DOループのベクトル化できない要因に対して質問文を表示し、利用者に応答を依頼する。利用者の応答によって、対象DOループの直前にベクトル化注釈行(*VOPTION文)を自動的に挿入する(図12)。これにより、コンパイラでのベクトル化が可能となる。

5 結 言

最近のスーパーコンピュータの普及は目覚ましく、研究所などの専門機関だけでなく、産業分野での利用も急速に進んでいる。FORT77/HAPは、このような背景の下で開発された最新鋭の超高速機S-820のためのFORTRANコンパイラとして、高い自動ベクトル化能力と効率的なオブジェクト生成機能を持っている。また、同時に、ユーザーの作業負担を軽減するため、FORT/VFをはじめとする支援ツールも開発した。

スーパーコンピュータの価値は、なんといってもその高い演算能力にある。これを極限まで追求するには、ハードウェアの性能を十分に引き出せるソフトウェアが必要不可欠である。FORT77/HAPでは、自動ベクトル化や並列化の技術を更に強化してユーザーの高性能化への要求に対応してゆく考えである。また、現在の急速な普及の中で、多くの一般ユーザーのはん用システムからの移行性や、使い勝手を考慮したソ

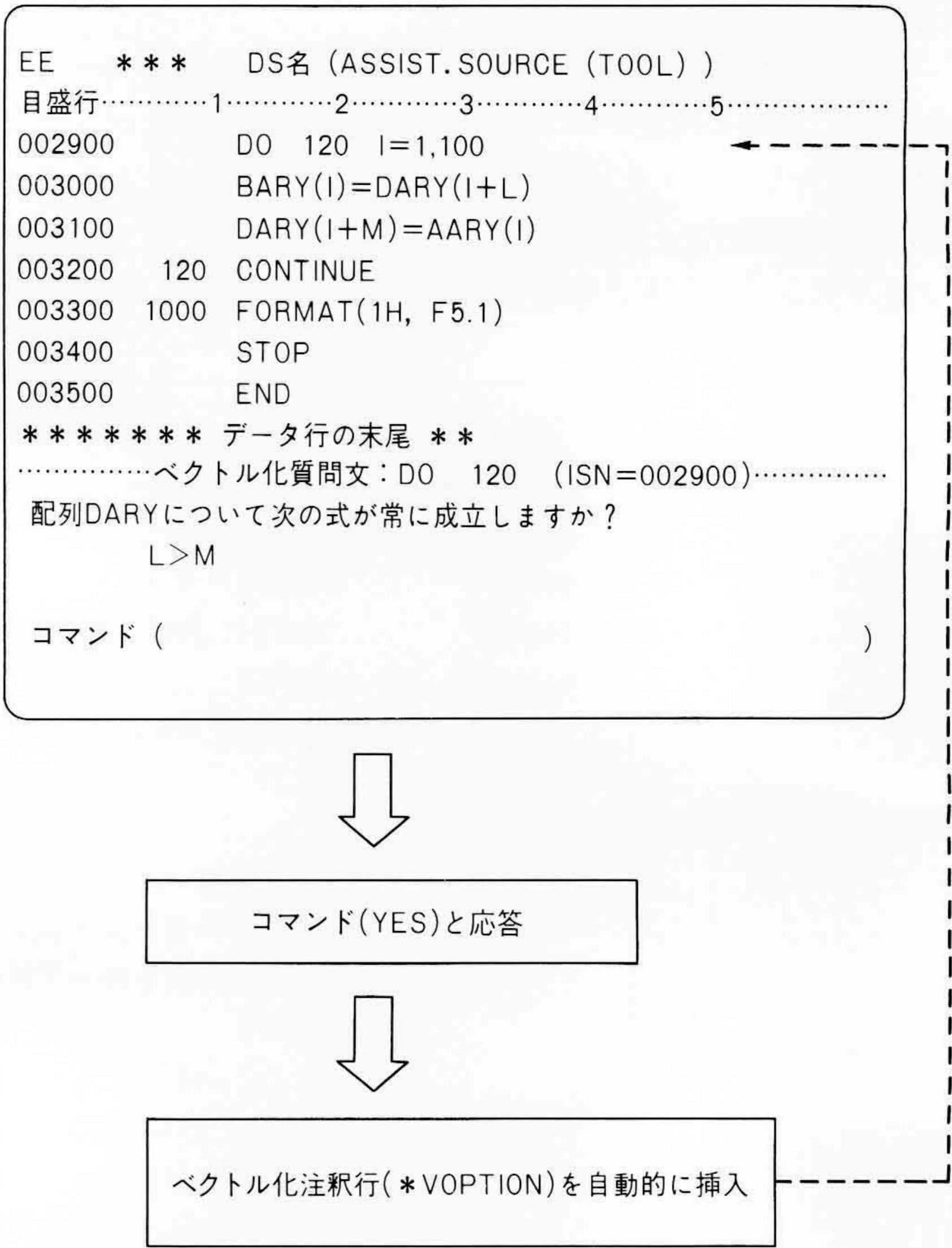


図12 ベクトル化質問文による自動チューニングの例 ベクトル化質問文に対してユーザーが‘YES’と応答すると、FORT/VFが*VOPTION文を自動的にソースプログラムに挿入し、ベクトル化の指示をする。

フトウェアシステムの形成も急務である。このような方向に対しては、今回開発したFORT/VFを中心とする支援ツールをいっそう強化、拡充してゆく考えである。

参考文献

- 1) 日本物理学会編：スーパーコンピュータ，培風館(1985)
- 2) 村田，外：スーパー・コンピュータ科学技術計算への適用，丸善(1985)
- 3) 小国：Fortran 77，丸善(1985)
- 4) 梅谷，外：技術計算プログラムの自動ベクトル化技術，情報処理，Vol.23, No.1, pp.29~40(Jan.1982)
- 5) 小高，外：最大性能が630MFLOPSで1Gバイトの半導体拡張記憶が付くスーパーコンピュータHITAC S-810，日経エレクトロニクス，314, 159~184(昭58-4)