JAERI-M
82-200

ベクトル計算機を含む複合計算機システムの構成法

１９８２年１２月

浅井　清

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-M レポートは，日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319—11 茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319—11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Section, Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　日立高速印刷株式会社

JAERI－M 82－200

ベクトル計算機を含む複合計算機システムの構成法

日本原子力研究所東海研究所計算センター
浅井　清

（１９８２年１２月９日受理）

過去６年間原研計算センタは所内研究者，技術者，計算機メーカと協力しながら，ベクトル計算機，あるいはいわゆるスーパーコンピュータによるベクトル計算処理の原子力コードへの適応性について調査，研究をおこなってきた。この間対象とした原子力コードは２，３の重複を含め，４０本にのぼる。この調査研究の結果，対象コードの約７割，全原研計算時間の７割がベクトル計算処理に適合することがわかった。これらのデータをもとに本報告では(1)当面のベクトル化可能計算量，(2)ベクトル計算機必要台数，(3)原子力コードのベクトル化に必要なマンパワー，(4)ベクトル計算機の持つべき演算性能，メモリ量，並行入出力パス数，入出力バッファ装置のメモリ容量と速さ，(5)ベクトル計算機運用に必要なソフトウェアと政策，を明らかにし，最後に(6)原研の複合計算機システムの構成を提案する。

JAERI-M 82-200

A Design of a Computer Complex Including Vector Processors

Kiyoshi ASAI

Computing Center, Tokai Research Establishment, JAERI

(Received December 9, 1982)

We, members of the Computing Center, Japan Atomic Energy Research Institute have been engaged for these six years in the research of adaptability of vector processing to large-scale nuclear codes. The research has been done in collaboration with researchers and engineers of JAERI and a computer manufacturer.

In this research, forty large-scale nuclear codes were investigated from the viewpoint of vectorization. Among them, twenty-six codes were actually vectorized and executed.

As the results of the investigation, it is now estimated that about seventy percents of nuclear codes and seventy percents of our total amount of CPU time of JAERI are highly vectorizable. Based on the data obtained by the investigation, (1)currently vectorizable CPU time, (2)necessary number of vector processors, (3)necessary manpower for vectorization of nuclear codes, (4)computing speed, memory size, number of parallel I/O paths, size and speed of I/O buffer of vector processor suitable for our applications, (5)necessary software and operational policy for use of vector processors are discussed, and finally (6)a computer complex including vector processors is presneted in this report.

Keywords: Computer, Supercomputer, Vector Processor, Nuclear Code, Vector Processing, CRAY-1, FACOM230-75 APU, Computer Complex, Computer Configuration, Simulation, Queueing Theory, Multiprogramming

目　　次

Contents

はじめに

1.1 ベクトル計算機の代表例

時間的に同時に，空間的に並行して生起する自然現象を模擬しようとする科学技術計算の分野においては計算機の演算速度がどのように速くても速すぎるということはない。既存の回路素子を使って演算速度を向上させる方法は大別して2つあり，そのひとつはプロセッサを数多く並べて並行動作させるパラレル（あるいはマルチ）プロセッサ方式であり，他のひとつは多数のデータについて同一演算を繰返すパイプライン・ベクトル方式である。

パラレル・プロセッサ方式ではFig. 1.1に見られるように，複数の，通常はメモリまで備えたプロセッサを並行動作させ，適当な時点で各プロセッサの計算結果の同期を取りながら計算を進めてゆく。既存の原子力（計算）コードをこの方式に合うように書き換えることはなかなか難しい。従来の計算コードがこの方式に合うように作られていないし，どのように計算コードを分割，計算すれば各プロセッサを万遍なく動作させられるかの判断が人間にまかせられているからである。比較的簡単な使用法は，対象とする物理モデルを空間的に分割し，その1区画に1台のプロセッサを割り当てるものである。原子力分野においてはマイクロコンピュータを多数結合して使用したＰＡＣＳ(1)，その他(2) の例がある。

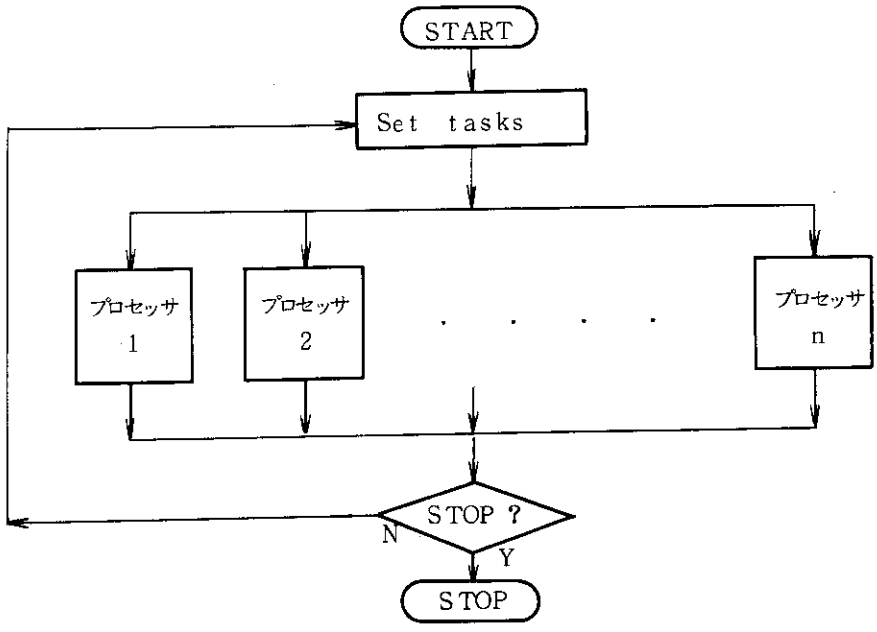


Fig. 1.1 A model of multiprocessor operation.

パイプライン・ベクトル方式はパイプライン方式を拡張したもので，その概念はFig. 1.2のように説明できる。この図で「階段」は浮動小数点加算命令の部分動作に相当するとしよう。浮動小数点の加算命令は，指数合せ，加算，正規化，丸め等，通常は8段階程度の基本動作

（第２章Fig．2.3参照）に分割される。各基本動作の回路が独立して動くように作られていると，各基本動作に異なったデータを対応させることができる。それはFig．1.2でいえば，階段の各段に人が居るのと同じである。１段上がるのに１秒かかるとすれば，階段の上に居る観察者から見れば１秒に１人の割で連続的に人が上ってくるように見えるであろう。この多数のデータについて連続的に加算をおこなえる階段を加算パイプラインと呼ぶ。

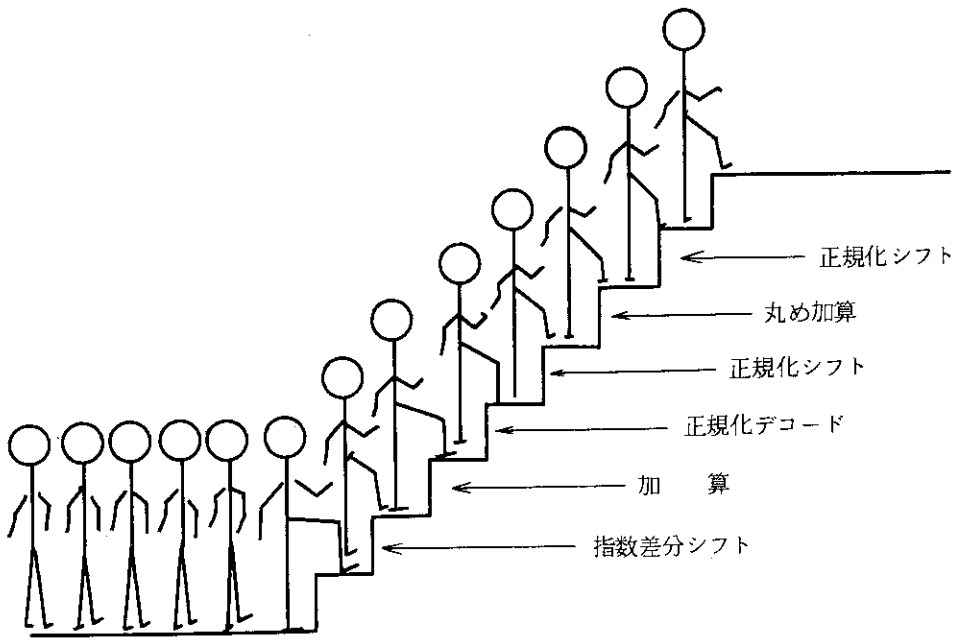


Fig. 1.2 A concept of pipeline operation for floating addition.

パイプラインの考え方はスカラ計算機でも採用されているが，それがもっとも有効に働くのは$a_i = b_i + c_i$，$i = 1 \sim N$ などのベクトル，あるいは配列（アレイ）についての演算の場合である。スカラ演算の場合は１個のデータについて階段の下の踊り場から上の踊り場に到達するまでの時間が専有されることが多いが，上の例だとNが充分大きければ，個々のデータの加算に要した時間はN÷８に近くなる。このためには加算の命令が実行される直前に下の踊り場へ素早く多数の人を集めておく必要があり，また上下の踊り場（レジスタ群）も用意しておかなければならない。ベクトル計算機とは算術，論理演算についてこのような機能を持つものをいう。並列処理の計算機としてはILLIAC－Ⅳ（３，４）が有名であるが，原子力分野で使用されたパイプライン・ベクトル計算機としてはSTAR－１００（１９７３年，ローレンス・リバモア研が最初（３，４）），CRAY－１（１９７６年，ロス・アラモス研が最初[5]）がよく知られている。

パイプライン方式の計算機に非常な高性能を期待することはできない。上述の例からもわかるように，ひとつの算術演算をあまり多くの基本部分に分けることはできないからである。これを補う方法としてはパイプラインの数を複数にするとか，踊り場（レジスタ群）を広くするなどが考えられる。ベクトル計算機のシリーズで上位と下位の機種に性能の違いがあるのは，このような理由による。この例からわかるように，配列要素a_iはパイプラインを連続的に動作させるために必要ではあるが，必らずしも物理的な意味でのベクトル要素である必要はない。

ILLIAC－Ⅳ，STAR－100は1960年代後半に開発が始まり1970年代前半に実用に供されたが，ベクトル演算処理の有効性を一般に認めさせるようには至らなかった。

ILLIAC－Ⅳについていえばプログラミングの難しさ，STAR－100についていえば，その第1の理由は演算を始めるまえに「踊り場」に数百以上の人（データ）を集めておかなければ性能が出ないということであろう。その上スカラ演算も速くなかった。これに対し1970年代前半に開発が始まり，1976年に1号機が出荷されたCRAY－1では，このような弱点は克服されていて，スカラ演算は速く，データを集めておくのに便利な踊り場（ベクトル・レジスタ群）が用意され，データ数（ベクトル長，あるいは配列要素数ともいう）は20～30でスカラ演算の何倍もの性能が出る。この結果既存計算コードのベクトル化のための書き換え作業は大いに軽減され，ベクトル計算機利用は促進されている。この故にCRAY－1の出現をもって世界のベクトル演算処理元年といってよい。

日本ではCRAY－1とほぼ同じ頃1977年（昭和52年）にベクトル計算機FACOM 230－75APU（アレイ・プロセッサ）が実用に供された(6)。このF75APUは2台作られ，うち1台は昭和52年8月から昭和57年1月まで商用稼動し，他の1台は研究開発用として，富士通川崎工場で使用されていた。この工場機は昭和55年10月から昭和57年12月まで日本原子力研究所計算センタに設置され，原子力コードのベクトル演算処理適応性研究に利用された。その経緯については次の1.2節で述べる。このF75APUはスカラ演算がF75CPUの2分の1と遅く，また入出力命令を持たず，そのために入出力はタスク・スイッチでCPUに渡していた。この弱点がF75APUの汎用機的使用を妨げ，したがってベクトル演算処理の展開に遅れをもたらしたことは否めない。ベクトル演算の立ち上り時間はSTAR－100に比べるとはるかに短く，CRAY－1に比べると少し長い。ベクトル演算処理ができるように修正された計算コードについて対CPU性能比で7～8倍の能力を持ち，これは現在原研計算センタで使用しているM－200計算機の2.5倍の性能である。ソフトウェアとしてはAP－FORTRAN，これで書かれたプログラムを標準FORTRANプログラムに逆変換するAPTRAN，また少し後にはFORTRANプログラムを擬似的に実行してベクトル処理可能なステートメントを診断摘出するベクトライザなどを備えていた。ハードウェアとこれらソフトウェアを概説した文献(6)は，先の文献（3，4）などと共にベクトル計算機への良い入門書である。

また同じ頃にM－180計算機用内蔵型アレイ・プロセッサIAPが出荷されている。これ自身は計算機ではなく，いくつかのベクトル演算命令を実行する回路で，M－180計算機にオプションとして付けられる。このIAPはM－180の上位機種M－200Hにも付けることができる(7)。IAPの価格はCPUの約10分の1と安く，そのために価格性能比はよいが，CRAY－1，F75APUなど本格的なベクトル計算機とは異なり，対CPU性能比は落ちる。

1.2 原研におけるベクトル計算処理の試み

これ以後計算コードのような大きな単位をベクトル計算機で処理することをベクトル計算処理，ひとつのステートメント単位のベクトル演算，あるいはベクトル演算回路による処理はベクトル演算処理と呼ぶことにする。

原研計算センタが具体的に，すなわち導入の対象としてベクトル計算機を考えるようになったのは昭和５２年である。この当時原研計算センタにはＦACOM２３０－７５計算機２システム（４CPU）が設置されていたが，計算需要の伸びが著しいために将来に備えて何らかの対策を立てておく必要があった。その対策のひとつとしてＦ７５APUによる大型長時間原子力コードの処理が考えられたわけである。原子力コードのベクトル化適応性検討の第１歩として昭和５２年５月にＦ７５APUのソフトウェア設計者を招いてＦ７５APUの説明会を持った。このとき出席者は利用者１０名，計算センタ１５名，説明者３名の２８名程度であった。以後現在まで機会あるごとに原子力コードのベクトル化作業をおこなってきた。それを年毎に記すと次のようになる。

(1) 昭和５２年５月～７月

長時間原子力コードTWOTRAN，RELAP 3B，EDDYTORUSのベクトル化とＦ７５ＡＰＵによる実行（この結果は3.1節Table 3.1 参照）

(2) 昭和５２年８月～１０月

昭和５１年７月～５２年６月末のジョブ統計から長時間ジョブ７本を選んでベクトル化可能性の調査（Table 3.1 参照）

(3) 昭和５３年５月

富士通によるＦ７５APUベクトライザの機能説明会。以後開発されたベクトライザを計算センタにて随時使用。

(4) 昭和５４年１月～５６年

核融合研究部理論解析研究室と計算センタでＥRATO4コードのベクトル化に着手，以後AEOLUS－R１，APOLLOコードのベクトル化を行なう（Table 3.1 ）。

(5) 昭和５４年８月～１１月

物理部固体物理第３研究室と計算センタでMONTE２３コードのベクトル化を行なう（Table 3.1）。

(6) 昭和５５年４月～５７年１２月

ＦACOM２３０－７５計算機システムからＦACOM M－２００計算機システムへの計算機交換に伴ない，日本原子力研究所と富士通株式会社は「原子力コードの並列計算処理手法に関する研究」につき共同研究契約（昭和５５年１０月～５７年３月）を結び，ベクトル計算処理について研究をおこなった。５７年４月から１２月までさらに事例研究のための共同研究をおこなった。このときの研究実施分担はTable 1.1の通りである。なおTable 1.1備考欄の数字は本報告巻末の文献番号で，この共同研究の成果である。

なお，上記(1)～(5)の作業は富士通株式会社の好意により富士通川崎工場Ｆ７５APUを使用しておこなわれた。

Table 1.1 Items and accomplishments of collaboration in research of vector processing of JAERI and FUJITSU, Ltd.

項　　目	分担		成果（発表論文，報告）
	原研	富士通	
・原子力コード・プログラム構造の分類と検討	○	○	18, 20, 29
・原子力コード・プログラム構造の再構成手法	○	○	8, 11, 12, 13, 15, 16, 21, 22, 23, 24, 25, 26, 27, 28
・並列計算処理のための標準指針の作成	○	○	14, 17
・事例研究用プログラム・パッケージの作成	○	○	9, 10, 19
・計算機構成法	○		本報告

注：成果番号は本報告巻末の引用文献番号に同じ。

1.3 共同研究成果と計算機構成法

共同研究期間中のものを含め４０本の原子力コードのベクトル計算処理適応性を調査，検討してきた。このなかには１部同じコードを繰返し，違った技術者，異ったデータあるいは異った使用法について調査したものもある。これらのコードを選んだ理由は使用頻度が高く，かつ長時間の計算時間を要していることによる。ベクトル化が容易であるからという理由で選択したものはない。これらの結果として判明したことを項目別に挙げると

(1) 当面のベクトル化可能計算量（ＣＰＵ量），
(2) ベクトル計算機必要台数，
(3) 原子力コードのベクトル化に必要な工数（マンパワー），
(4) ベクトル計算機の持つべき演算性能，メモリ量，並行動作すべき入出力パス数，入出力バッファ装置のメモリ容量と速さ，
(5) ベクトル計算機を運用する際に必要なソフトウェアと計算機運用政策

である。さらにこれらのデータをもとに

(6) ベクトル計算機を含む当面の複合計算機システムの構成

を本報告の末尾で提案する。

上記(1)から(6)に至るまでの説明はデータを多用し，わかりにくいので，その論理の道筋をFig. 1.3 に示した。

なお，本報告ではベクトル計算機はスカラ計算機と同一データセット構造であると仮定している。そうでない場合に考慮すべき情報を持っていなかったせいである。

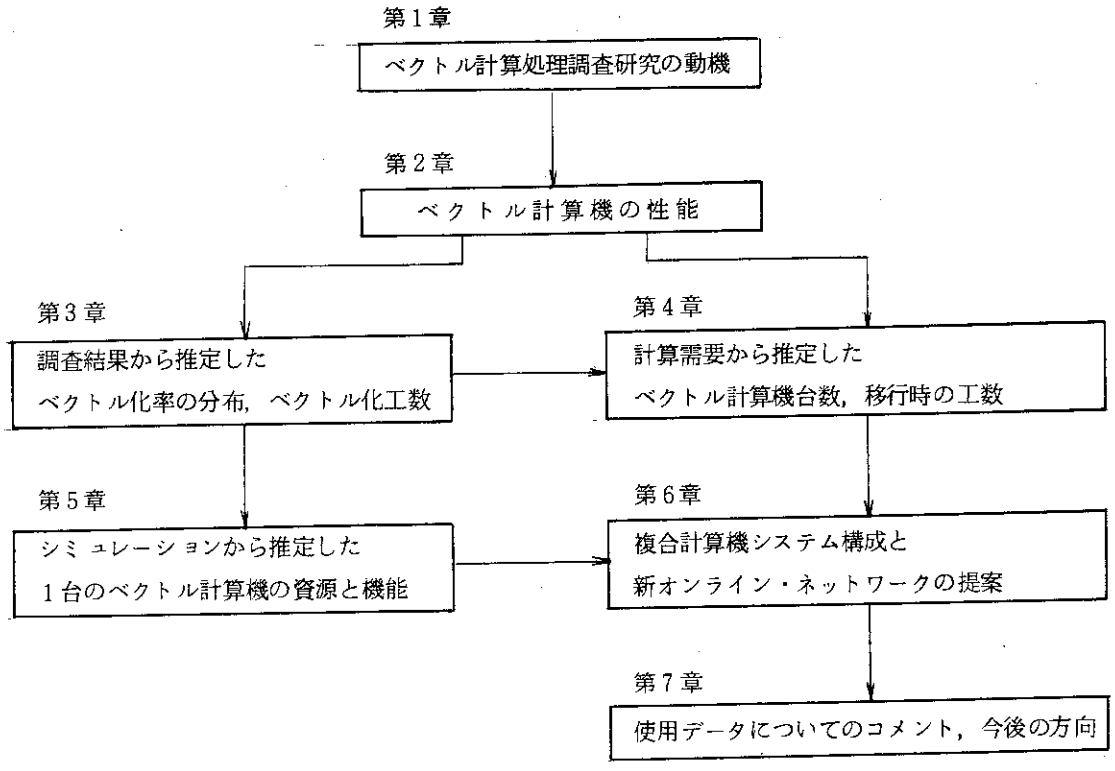


Fig. 1.3 Relational flow of subjects in this report.

2　ベクトル計算機の性能

2.1　ベクトル計算機による性能向上の方法

Fig. 2.1 は代表的なベクトル計算機CRAY－1 のレジスタ構成である（30）。

FORTRAN 文

```
DO  ××  I＝1, N
××  C(I)=C(I)－E＊A(I)－E＊B(I)
```

を実行するときの配列と計算の中間結果B(I), C(I), C(I)＊…… I＝1, ……, の取扱いをCRAY－1で見ると次のFig. 2.2 のようになる。

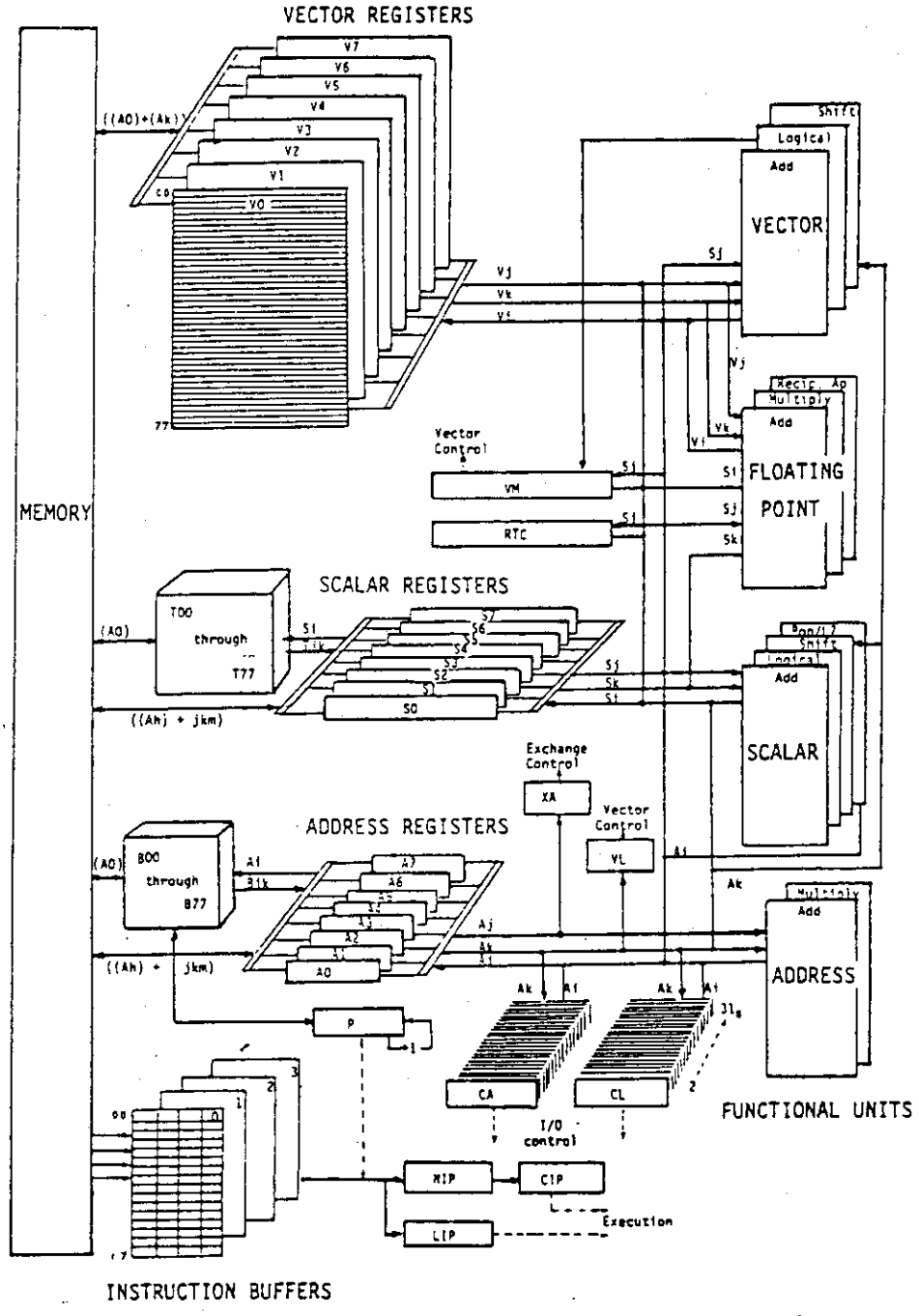


Fig. 2.1　Diagram of registers of CRAY-1 computer(30).

Action	Instruction	Comment
	VL A3	(A3) to VL, i.e., load vector length to VL.
	.	
	.	
Load A_i	V1 ,A0,1	Read (VL) words to V1 from (A0) incremented by 1.
ES*A_i	V7 S6*RV1	Rounded floating products of (S6) and (V1) to V7.
	.	
	.	
Load B_i	V0 ,A0,1	Read (VL) words to V0 from (A0) incremented by 1.
FS*B_i	V5 S7*RV0	Rounded floating products of (S7) and (V0) to V5.
	.	
	.	
Load C_i	V6 ,A0,1	Read (VL) words.
C_i-ES*A_i	V4 V6-FV7	Floating differences of (V6) and (V7) to V4.
V4-FS*B_i	V3 V4-FV5	Floating differences of (V4) and (V5) to V3.
Store C_i	,A0,1 V3	Store (VL) words from V3 to (A0) incremented by 1.
	.	
	.	

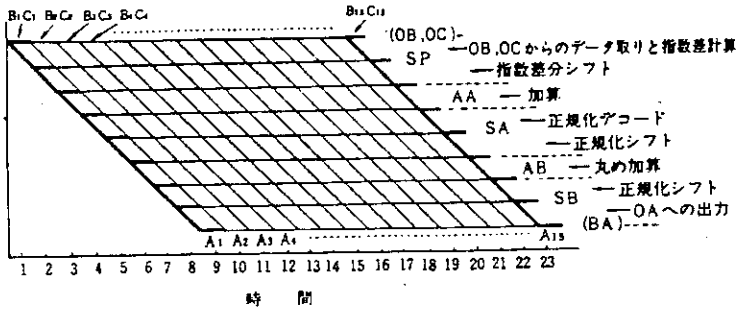
Fig. 2.2 Operational procedure of CRAY-1 registers for expression C(I) = C(I) - E*A(I) - E*B(I).

このFig. 2.2からわかるとおり，一度メモリからベクトル・レジスタにロードされた配列要素B(I), C(I), … , 計算の中間結果E*A(I) 等はできるだけベクトル・レジスタから離れてメモリへ移ることがないよう翻訳実行される。

相続いて演算の対象となる配列（ベクトル）要素を，ベクトル・レジスタに格納しておくことによって速いアクセス時間を実現できる。

FACOM230−75APU においてはベクトル・レジスタVRにある2語が乗算回路へ達するまでの時間は90ナノ(n) 秒，メモリMEMにあるデータ2語が乗算パイプラインに到着するまでの時間は90ナノ 秒である(6)。

つぎにパイプライン処理によって演算がどのように高速化されるかをFACOM230−75 APU（F75APU）を例として見てみよう。Fig. 2.3はF75APUの倍精度浮動小数点ベクトル加算命令VADのタイム・チャートである。これでみるとVAD命令は8個のフェーズから成っている。

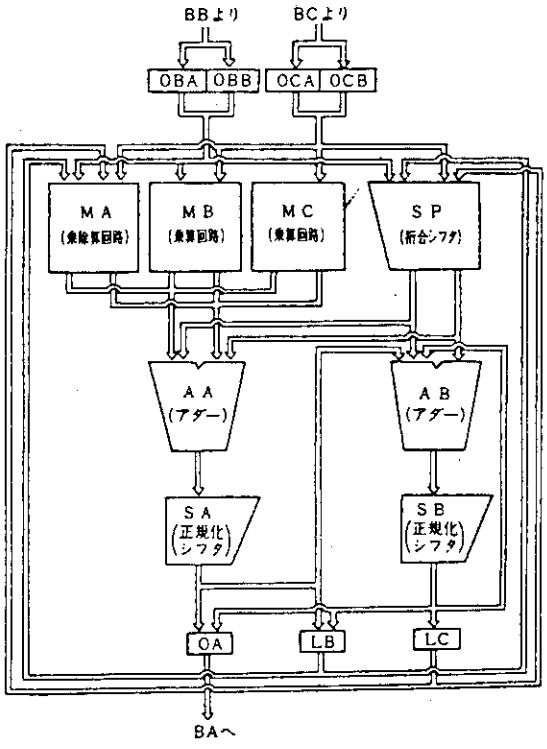


VAD (倍精度) のタイムチャート

Time chart of VAD (floating double data).

Fig. 2.3 Pipeline operation of FACOM230-75 APU computer for double-precision addition (6).

これらのフェーズに対応する算術演算のブロック・ダイアグラムをFig. 2.4に示す。



算術演算パイプラインのブロックダイアグラム

Block diagram of arithmetic operation pipe line.

Fig. 2.4 Block diagram of pipelining for arithmetic operation of FACOM230-75 APU(6).

各フェーズが常に動作しているようにオペランドを連続的に供給できれば1クロック（90ナノ秒）に1回の加算がおこなわれる。また単精度加算の場合にはパイプラインを2系列に分割して演算するので1クロックに2回の加算がおこなわれる。論理演算のパイプライン処理も同様に行なわれる。これら演算のパイプライン処理，命令の先行制御などを合せて計算の高速化が達成される。現代の（スカラ）計算機でもこのような処理はおこなわれているが，演算のパイプラインを常時連続的に動作させる機会は少ない。これに対し，利用者がベクトル計算機を意識して計算機プログラム（計算コード）で長に要素数（ベクトル長ともいう）の配列を多用すれば，演算パイプラインを連続的に動作させることができる。F75APUの単精度浮動小数点加算では1クロック（90ナノ秒）に2個の計算結果を出すから，性能としては1秒÷（45ナノ秒）＝22MFLOPS（Million floating point operations per second）となる。ただ原研の計算コードでは22百万語も連続的に数秒間加算をおこなう必要があるものは現在のところはない。したがって通常の計算コードにおいてはベクトル計算機のもつ最高性能を生かせる部分とその処理時間はそれほど長くはない。最高性能が「最大瞬間風速」と呼ばれる由縁である。一方，最高性能を出すことにこだわらなけれが，計算コードを修正して，配列要素数をある程度長くし，それによってベクトル計算機の性能を出すことができる。それについては第3章で述べる。

2.2 ベクトル・スタート・アップ時間

ベクトル命令を実行するには，ベクトル・レジスタにデータを詰めるなどの準備作業が必要で，ベクトル計算機がこの作業に消費する時間をベクトル立上り（ベクトル・スタートアップ）時間と呼ぶ。この時間が長ければベクトル命令の効率は落ちてくる。古いタイプのベクトル計算機，例えばCDC STAR－100 ではこの時間は長い。次のFORTRAN 文ではこの時間は長い。

```
      DO  10  I=1,  length
10    A(I)=B(I)+C(I)
```

においてSTAR－100 ベクトル演算の立上り時間を配列要素数との関係で示したものがFig. 2.5である。これで見るとSTAR－100 は配列要素数が５００を超えるあたりから性能が出るように設計されていることがわかる（31）。これに対し，CRAY－1では同じ操作について配列要素数が8を超えたあたりからベクトル命令の性能が生きてくる（Fig. 2.6）。この図でM200H－IAPのデータは文献（32)による。これはCRAY－1計算機の大きな特長であり，この特長のためにベクトル計算機の原子力コードへの適用範囲が大きく広がったといっても過言ではない。我々の調査の範囲では，既存の原子力コードでは要素数が10～60の範囲に入る配列が多い。計算時間を短くするためということもあるが，それよりも従来は計算機のメモリに磁気コア装置が使われ，その価格が高価であったために配列要素数を極端に切りつめて計算コードを作らなければならなかったせいである。

2.3 ベクトル化率と計算コードの速度向上

前節のFig. 2.5，2.6からわかるように配列の要素数が大きいときのベクトル計算機のベクトル演算性能は，同じ計算機のスカラ演算性能に較べて飛躍的に増大する。いま，ある計算コードの実行に要する時間がスカラ演算でT時間，そのうちFORTRAN 文

```
      DO  10  I=1, N
10    A(I)= B(I)+ C(I)
```

の実行に要するスカラ演算時間が全体のV×１００パーセントを占めるとしよう。

この計算コードをF75APUで実行するとN≒100 のときベクトル演算では7倍以上の性能向上を期待できる（Fig. 2.6）。したがって計算コード自体の性能向上の倍率P（V, α）は，ベクトル計算機のベクトル演算能力をスカラのそれのα倍と想定し，α＝7とすれば

$$P(V,\ 7) = \frac{T}{\frac{V\times T}{7} + (1-V)\times T} = \frac{1}{\frac{V}{7} + (1-V)} \qquad (1)$$

となる。一般にPは

$$P(V,\ \alpha) = \frac{1}{\frac{V}{\alpha} + (1-V)} \qquad (2)$$

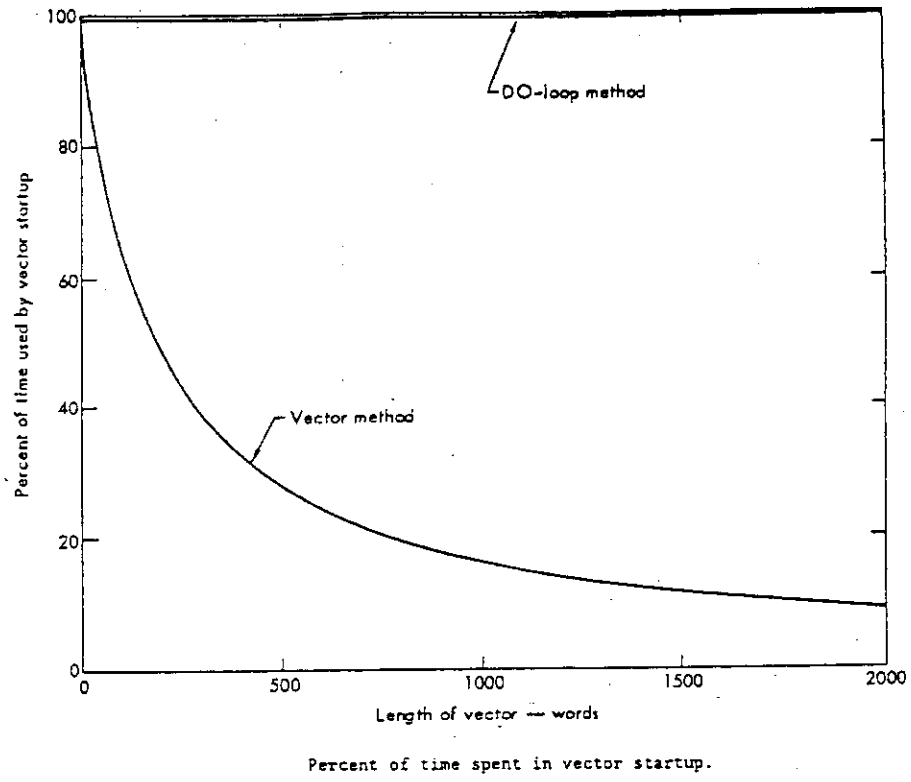


Fig. 2.5 Percentage of time spent in vector startup of STAR-100 computer(31).

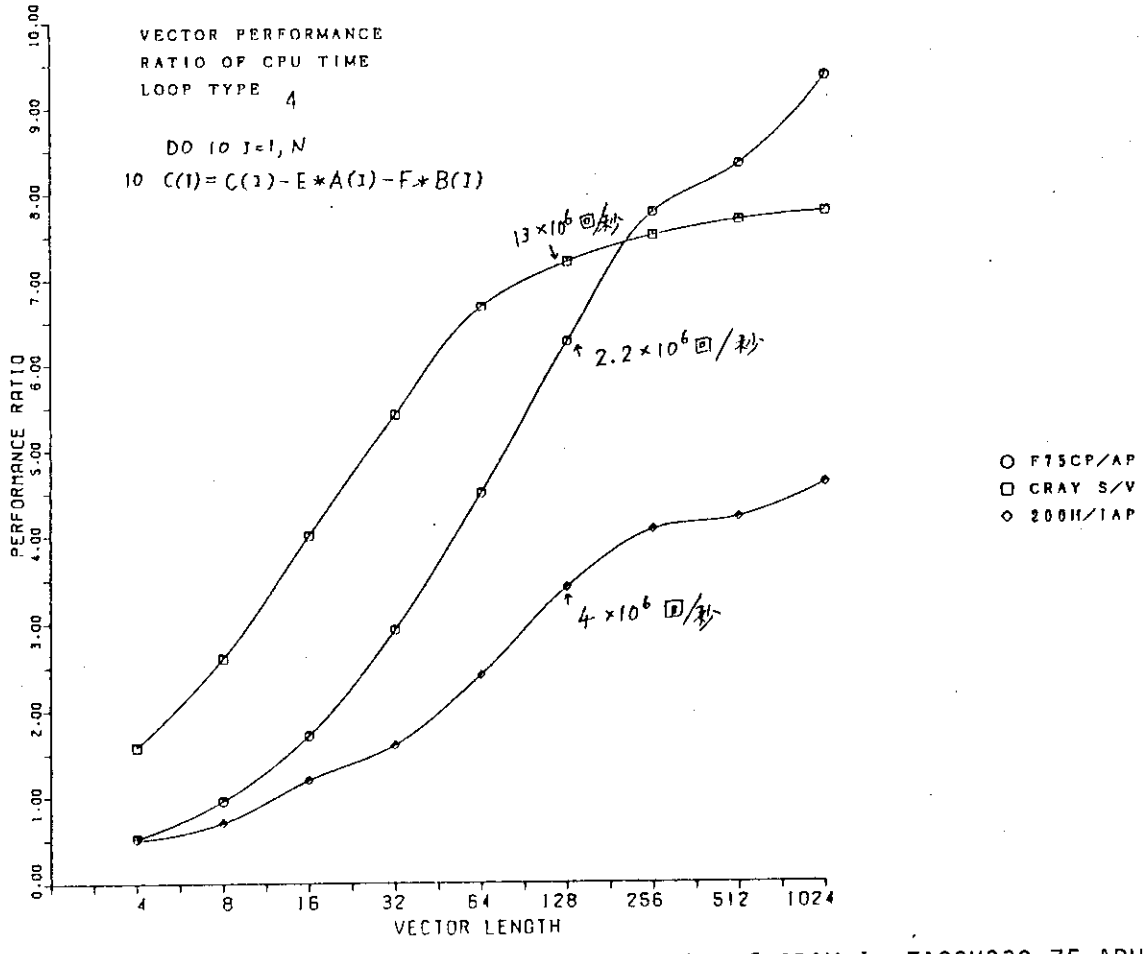


Fig. 2.6 Vector vs. scalar performance ratio of CRAY-1, FACOM230-75 APU, and HITAC M200H IAP.

とかける。ここでαはFig. 2.5, 2.6からわかるとおり，使用されるベクトル演算の種類，配列の要素数によって異なるから厳密にいえば，各FORTRAN文毎にT_i，α_iを対応させ

$$P=\frac{T}{\sum_i \frac{T_i}{\alpha_i}+(T-\sum_i T_i)}\quad,\quad T=\sum_i T_i$$

と表現すべきものである。しかし，簡単な目安を得るために通常は(2)式を使用する。このP，V，αの関係を図示したものがFig. 2.7である（18）。αが一定のときは計算コードの性能向上の倍率Pは計算コードのベクトル化率Vに依存する。ベクトル化率Vは計算コードのスカラ演算時間の何割をベクトル演算で置き換えることができるかを示す値であって，ベクトル計算処理の可能性を示す重要な指標である。このVは計算コードに依存し，個々のベクトル計算機からは比較的独立している。これに対しαの値は計算コード（主として演算，種類，演算実行頻度，配列要素数）とベクトル計算機の性能に依存する（Fig. 2.6, 2.7）。計算コードをベクトル計算機で速く処理するためには，ベクトル化率Vの値を大きくすることが望ましい。しかし，無理にベクトル化しても無駄な演算が多くなれば結果として倍率Pの値が1よりも小さくなることがあるので注意を要する。

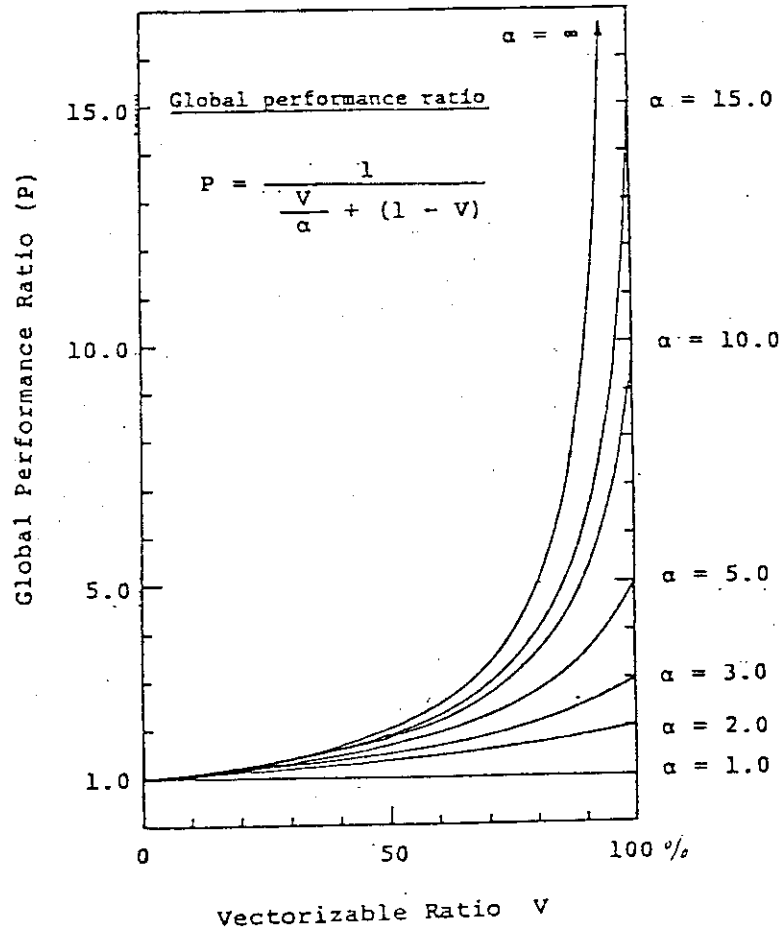


Fig. 2.7 Relation of vectorizable ratio (V, α) and performance P.

3. 原子力コードのベクトル化率

3.1 原研におけるベクトル処理適用性調査の時期とレベル

原子力コードへのベクトル計算機の適用性を見るために，原研では昭和５２年から５７年11月までに５回の調査をおこなっている。それらの調査は次のレベルに分類される。

調査したベクトル化技術のレベル

① コンパイラによる自動ベクトル化ならどれだけ速くなるか。

② プログラマによるソース・プログラムの簡単な手直しならどれだけ速くなるか

③ 数値計算法の変更ならどれだけ速くなるか

④ 物理モデルの手直し（計算式，境界条件の変更，計算順序の変更，使用実験式の変更）ならどれだけ速くなるか。

上記の調査レベルのうち①は５２年当時のＦ７５APU FORTRAN ベクトライザ（33）相当のベクトル化ツールを想定している。最近の自動ベクトル化ツールはもっと進歩している筈である。

５回の調査を実施した時期とそのレベルは次のとおりである。

第１回 長時間の計算時間を要する数本の原子力コードのベクトル計算適応性（昭和５２年７月），③－レベル

第２回 一年間の大型ジョブ処理の統計から大型ジョブを数本選択し調査（昭和５２月１０月），②－レベル

第３回 富士通との共同研究の一環として１０本の大型原子力コードのベクトル計算適応性（昭和５６年１０月），②－レベル

第４回 核融合分野 数本のコードのベクトル演算化（昭和５５～５６年１２月），③－レベル

第５回 富士通との共同研究の一環として炉物理，環境安全性，核融合分野の１０本の大型原子力コードのベクトル計算適応性（昭和５７年４月～１１月），③－レベル

これら第１～５回の調査結果の一覧表をTable 3.1に示す。

Table 3.1の見方について説明しておこう。

Table 3.1のデータは次のような手順で求めている。

(1) 対象となる計算コードをよく使用されているデータでランし，その実行過程をソフトウェア・ツール FORTUNE でトレースする。FORTUNEはソース・プログラムの各ステートメント毎にそのステートメントの実行に要した時間の積算を出力する。したがって計算時間の積算量が大となるステートメントがいくつかあれば，その部分のみに注目すればよい。その部分のみ，あるいはそれを含むサブルーチンを簡単に書き換えてベクトル計算処理が可能ならばその変換レベルは②レベルである。書き換えが簡単ではなく，数値計算法を変更すればベクトル計算処理が可能になる場合は変換レベルは③レベルである。逆

Table 3.1 Vectorizable ratio, performance, sizes of 40 investigated nuclear codes.

番号	コード名	分野（基礎方程式）	解法	ソース枚数	ベクトル化率	速度向上	ソース修正率	変換のレベル	メモリ増加率	ファイル数×容量	I／O回数	計算機	年	人月
1	TWOTRAN	2次元多群中性子輸送計算（中性子輸送方程式）	有限差分法ガウス反復法，x－y形状	7,000枚	83%	4.5倍	4%	3				F75APU	52	4
2	RELAP-3B	軽水炉一次系の熱流体挙動予測計算	差分法による時間依存方程式	22,000	28	0.93	4	2				F75APU	52	6
3	EDDYTORUS	JT-60真空容器及び架台表面の渦電流計算	ガウス・ルジャンドル4重積分	4,000	87	5.8	10	3				F75APU	52	2
4	RELAP4／MOD5	軽水炉冷却材喪失事故解析（熱水力学的保存式，熱伝導方程式）	有限差分法ガウス消去法	50,000	4		3	2				F75APU	53	2
5	FEM3D	JT-60 3次元渦電流分布計算（マックスウェル方程式）	有限要素法ガウス消去法	1,900	64		13	2					53	1
6	MCDRAN	反応度事故時の燃料過渡挙動解析（熱伝導方程式）	有限差分法ガウス消去法	12,000	15		11	2					53	1
7	LHCONE	トカマクLower hybrid波の共鳴加熱による伝搬計算（移流方程式）	高速フーリェ変換	2,300	80		30	2				F75APU	53	2
8	DVTT	ダイバータ・トカマク荷電粒子追跡。2次元PIC法（ポアソン方程式）	高速フーリェ変換	4,000	60		12	2					53	1
9	MORSE-CG	3次元中性子輸送計算（中性子輸送方程式）	モンテカルロ法	8,000	1		9	2					53	1
10	SAFE-3D	任意形状非均質3次元弾性反応力解析（平衡方程式）	有限差分過緩和法	5,700	82		27	2				F75APU	53	2
11	MONTE23	ハイゼンベルグ・スピン系中性子散乱実験データの解析（ボルツマン分布則）	モンテカルロ法	2,000	75	2.7	10	3				F75APU	54	3
12	ERATO4	線型MHD安定性解析のための固有値解法ルーチン（線型MHD方程式）	有限要素法，コレスキー分解＋逆ベキ乗法	1,400	97	6.2	93	3	1.81	24×5MB（作業域）	9230（ワーク用）	F75APU	55	9.2

Table 3.1 (Cont'd)

番号	コード名	分野（基礎方程式）	解法	ソース枚数	ベクトル化率	速度向上	ソース修正率	変換のレベル	メモリ増加率	ファイル数×容量	I／O回数	計算機	年	人月
13	AEOLUS-R1	非線型抵抗性MHD内部不安定性解析（非線型MHD方程式）	有限差分法コンボルージョン＋混合差分法	2,400枚	99%	6.6倍	3%	3	1.14	0	0	F75APU	55	4.5
14	APOLLO	トカマク・プラズマの1 1/2輸送計算（プラズマ輸送方程式）	有限差分法 FACR／DCR	4,500	59	1.5	48	3	3.29	0	0	F75APU	55	4
15	ALPHMN	α－Mn 結晶のd電子解析（シュレディンガー方程式）	Implicit QL法	1,400	98	7.9	37	3	1.02	20MB（ワーク用）	4500	F75APU	56	3.5
16	NICER	量子化学固有値計算（密行列標準，一般固有値問題）	Jennings法 ハウスホルダ＋QR逆反復法	840	99	9.3	19	3	1.04	0	0	F75APU	56	0.7
17	ANISN	1次元多群中性子輸送計算（中性子輸送方程式）	有限差分法	3,000	21	1.2	23	2				α=5	56	0.2
18	CITATION	3次元多群中性子拡散計算（中性子輸送方程式）	有限差分	26,000	90	3.2	4	2				F75APU	57 56	4
19	KENO4	多群臨界安全性計算	モンテカルロ法	7,600	12	1.1	10	2				α=5	56	0.2
20	DWBA	原子分子衝突過程の断面積計算（波動関数のコンボルージョン）	2重数値積分	4,000	79	2.7	13	2				F75APU	57 56	2
21	PALLAS-2DCY	2次元中性子輸送方程式に基づく遮蔽計算（中性子輸送方程式）	数値積分	3,800	90	4.0	8	2				F75APU	57 56	1
22	RELAP5／MOD1	軽水炉安全解析（質量，運動量保存，熱伝導方程式）	差分法による時間依存微分方程式	43,000	15	1.1	50	2				α=5	56	0.3
23	SAP5	構造物の線型構造解析（連立一次方程式あるいは固有値問題）	有限要素法とあるいは行列式探索法	16,000	72	2.0	2	2				F75APU	57 56	3

Table 3.1 (Cont'd)

番号	コード名	分野（基礎方程式）	解法	ソース枚数	ベクトル化率	速度向上	ソース修正率	変換のレベル	メモリ増加率	ファイル数×容量	I／O回数	計算機	年	人月
24	THYDE－P	加圧水型軽水炉安全性解析（質量，運動量保存方程式，熱伝達関数）	差分法による時間依存微分方程式	26,000枚	39%	1.4倍	2%	2				$\alpha=5$	56	1
25	VENTURE	3次元多群中性子拡散計算（中性子拡散方程式）	有限差分法	36,000	63	2.1	1	2				CRAY－1 F75APU	56 56	3
26	CASKMARL	放射線照射固体材質中のはじき出しカスケード・シミュレーション	カスケード法（大半はテーブル探索）	10,000	39	1.4	10	2	1.03	342KB	196	$\alpha=3.4$	57	4
27	GMSCOPE	電子顕微鏡像シミュレーション	コンボリューション，フーリェ変換	240	99	7.5	22	3	1.25	190KB	200	F75APU	57	2
28	FEM－BABEL	3次元中性子拡散計算（中性子拡散方程式）	有限要素法	6,600	89	4.3	11	3	1.0	80KB（作業域）	9350	F75APU	57	1
29	TWOTRAN（臨界）	2次元中性子輸送計算（中性子輸送方程式）	有限差分 x－y 形状	7,000	75	2.0	5	3				F75APU	57	0.5
30	TWOTRAN（遮蔽）	2次元中性子輸送計算	有限差分 r－z 形状	7,000	20	1.1	5	3				F75APU	57	0.5
31	DOT 3.5（遮蔽）	2次元中性子輸送計算（中性子輸送方程式）	有限差分 x－y 形状	6,500	78	2.0	8	3		200KB	5670	$\alpha=3$	57	0.25
32	DOT 3.5（遮蔽）	2次元中性子輸送計算	有限差分 r－z，r－θ 形状	6,500	20	1.1	8	3		200KB	5670		57	0.25
33	DOT 3.5（臨界）	2次元中性子輸送計算	有限差分 x－y 形状	6,500	78	2.0	8	3		200KB	5670		57	0.25
34	DOT 3.5（臨界）	2次元中性子輸送計算	有限差分 r－z，r－θ 形状	6,500	20	1.1	8	3		200KB	5670		57	0.25
35	BERMUDA－2DN	2次元中性子輸送計算	P∞直接積分法，r－z 形状	5,000	70	2.5	1	2					57	0.5

Table 3.1 (Cont'd)

番号	コード名	分野（基礎方程式）	解法	ソース枚数	ベクトル化率	速度向上	ソース修正率	変換のレベル	メモリ増加率	ファイル数×容量	I／O回数	計算機	年	人月
36	MATHEW	3次元風速場計算（ポアソン方程式，流体連続方程式）	有限差分 SOR法	3,000枚	98%	7.4倍	5%	3	1.7	4MB	121	F75APU	57	2
37	ADPIC	汚染物質移流拡散計算（拡散方程式）	セル内粒子法	3,000	51	5	15	3	1.1	1.2MB	94	α=3 (M200)	57	2
38	GAMPLU	外部被曝線量計算（線量評価積分式）	ガウス積分	2,000	90	0	20	3	2.7	0	0	α=6 (M200)	57	2
39	JT60POV2	中性子輸送計算（1次元粒子輸送方程式）	径路数値積分	23,000	34		9	2					57	3
40	IMPURV3	不純物効果計算（1次元粒子輸送方程式）	径路数値積分	17,500	49		4	2					57	3

に目立って時間を消費するステートメントやサブルーチンが存在しない場合あるいは時間消費量が大であってもその部分の計算法がベクトル計算処理に向かない場合に，物理的条件，数値計算法を変更すれば，特定のサブルーチンに時間を集中できることがある。この場合の変換レベルは③あるいは④レベルである。

(2) 計算機がＦ７５APU，ＣＲＡＹ－１と記してあるのは実測値である。$\alpha = 3.4$ と具体的に数値があるのは図2.6その他から計算式に応じて性能倍率を求め，その平均としてαを求めたものである。（M－２００）とあるのは直ちにベクトル計算機にランできるまでにM－２００計算機の上でベクトル演算に再構成されていることを示す。

(3) ベクトル化によるメモリ増加率，入出力回数，ファイル容量等データを収集し始めたのは最近で，それ以前は集めていない。

(4) この表のベクトル化率は最小値を示していると見るのが妥当である。努力すればもっと上がる。

3.2 調査対象コードと調査方法

各回の対象コードと調査の方法は以下のとおりであった。

第1回の調査（昭和52年7月）－数値計算法レベル

当時原研で使用されて大型原子力コード3本を選んでベクトル化をはかり，FACOM230－75APU（アレイ・プロセッサ）でテストした。選んだコードはTWOTRAN（原子炉内中性子束計算），RELAP3B（軽水炉一次系の熱流体挙動の時間変化計算），EDDYTORUS（JT－60真空容器および架台の渦電流計算）である。

これらのコードを選んだ理由は

1) これらが当時よく利用されていた，
2) 書き換え作業をおこなう担当者（計算センタ外来研究員）がこれらコードの数値解法について知っている，
3) これらコードはFACOM230－75 のCPU1時間以上を使う，
4) これらコードの原研センタのCPU使用時間に占める割合が大きい，

などである。

実測テストの結果をTable 3.1番号1～3に示す。これから2～3ケ月程度の人手をかければベクトル計算機では2～6倍程度の性能向上を期待できる場合が多いことがわかる。TWOTRANとEDDYTORUS ではコード全体のCPU時間の大半を消費するサブルーチンがそれぞれ1～2個あり，それらをベクトル化することで性能向上をはかることができた。RELAP3Bにはそのようなサブルーチンはなかった。TWOTRANとEDDYTORUS の書き換えのレベルは，数値計算法の変更に当る。RELAP3B のそれはプログラマによる手直しに相当する。

第2回の調査（昭和52年10月）－簡単な手作業レベル

昭和52年10月に計算センタの1年間（昭和51年7月～52年6月）のジョブのなかからCPU時間が大きく，かつ使用頻度の多い巨大ジョブ7本を選び，ベクトル計算機への適応性を調査した。このとき最初はF75APUで直接テストすることはせず，測定ツールFORTUNE を使って調べた。FORTUNEはF75 で該当コードを実行し，各サブルーチンが必要とする計算時間を測定する。

この調査で巨大ジョブの全ジョブに対する比率はFig. 3.1のとおり。7本のコードの概要と調査結果をTable 3.1 ，番号4～9に示す。この当時はベクトル処理の適否の閾値を75%とおいて判断していた。75%としたのはスカラ演算のスピードがCPUの1/2，タスク・スイッチに時間をとられる，などのF75－APUの性能上の問題点を考慮したもので，CRAY－1，或いはこれと類似のベクトル計算機などではコードのベクトル化率が50%以上であれば，ベクトル処理によって性能向上を期待できる（Fig. 2.7）。したがって現在ではTable 3.1，番号4～9の7本のコードのうち4本がベクトル処理に適している。

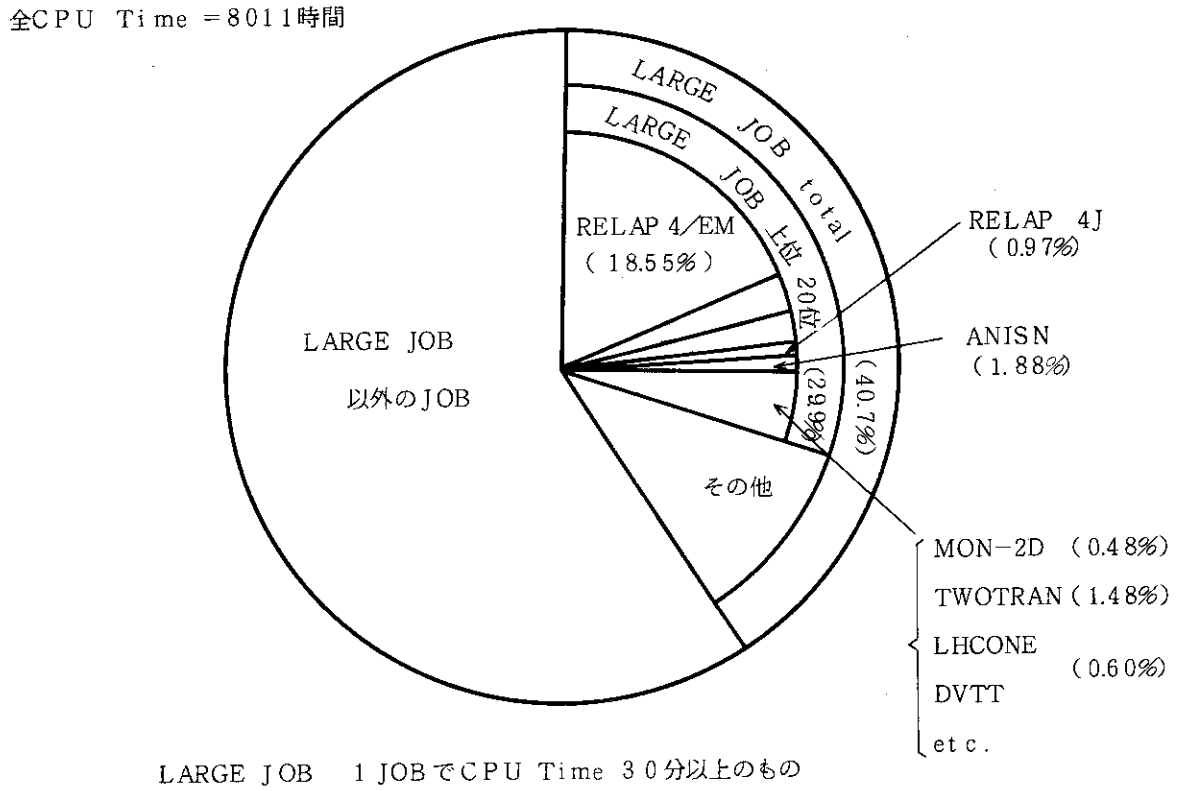


Fig. 3.1 Percents and names of large scale codes in July 1976 - June 1977 of JAERI.

これら7本のうち番号7，10の2本の計算コードをF75-APU でテストした。この調査の後でさらに1本のテストを試みた。Table 3.1，番号11のコードがそれである。

第3回の調査（昭和56年10月）－簡単な手作業レベル

昭和54年FACOM230-75からM-200計算機への交換の一環として原子力コードへの並列処理適用の共同研究が富士通(株)から提案され，昭和55年10月原研と富士通とで「原子力コードの並列計算処理手法に関する研究」について共同研究契約を締結した。期限は57年12月末である。この契約の結果共同研究の期間中は，富士通からFACOM230-75APU（アレイ・プロセッサ）が無償提供され原研に設置されていた。この共同研究のひとつの作業として計算センタでは昭和56年2～3月中に使用頻度の高いジョブ10本を選び，測定ツー

ルFORTUNEを使用し，これら１０本の原子力コードへのベクトル計算処理の適応性を調査した。その結果各コードのベクトル化率の推定値はTable 3.1，番号１７～２５，および２９のとおりである。この調査は第２のレベル，すなわち簡単な手作業でソース・プログラムを変更することで性能向上をはかれるかどうかを主眼におこなった。このうち中性子拡散コードVENTUREはCRAY－1とＦ７５－APUで実測し，ほぼ同様の結果を得ている。

第４回の調査（昭和５５～５６年１２月）－数値計算法レベル

これは正確には調査ではなく作業の結果である。ERATO 4，AEOLUS－R 1，APOLLOはベクトル計算処理を意識して改良，または新規作成された核融合研究部の原子力コード，ALPHMN，NICERは並列処理に関する共同研究の中でベンチマーク・テスト用に提供されたコードである。これらのコードはベクトル計算処理を意識して数値計算法を選択しているので，いずれもベクトル化率が高い。倍率はF７５－APUによる実測値を示す。これらの結果から，数値計算法のレベルまで堀り下げれば，ベクトル計算処理によって相当の性能向上が期待できることがわかる。これらのコードはTable 3.1の番号１２～１６である。

第５回の調査（昭和５７年４月～５７年１１月）数値計算法，物理モデル・レベル

炉物理，環境安全性，核融合分野の原子力コード約１０本を選び，数値計算法，物理モデルのレベルでベクトル計算処理適用可能性について調べている。各原子力コードにつき，その問題を熟知している研究者が協力しているので，従来の第１～第３回の調査よりは広い適応性，高いベクトル化率を達成できたものが多い。これらコードはTable 3.1の番号２８，３０～４０である。

3.3 調査結果

調査の結果明らかになったことを定性的，定量的に述べる。本報告の主題である計算機構成法ではとりあえず定量的側面のみが関係する。

3.3.1 定性的所見

調査対象としたコード数の約６５％は前述①②レベルの範囲の手直しで1.5～２倍以上の性能向上が期待できる。

調査対象としたコードのうち上述③レベルの数値計算手法がベクトル処理に適合したコードでは４～８倍の性能向上を達成している。

また当初からベクトル計算処理を意識して開発したコードも同程度の性能向上を得ている。

調査対象としたコードのうち上述①，②，③の手直しで大巾な性能向上できなかったコードの代表例にLOCA 解析コードRELAP4がある。①，②，③の手直しはFig．3.2に見られるように，特に計算時間を必要とするサブルーチンの特定部分のベクトル演算化をはかるものが多い。RELAP４にはこの種の「突出した」部分がない。ベクトル化をおこなうには根本的な書直しが必要である。しかし書直して性能向上できるかどうかはわかっていない。この種の

コードの開発には多くの人手と年月を費しているので根本的な書直しは大変である。また次々と修正版が出るので追随することは難しい（付録２参照）。

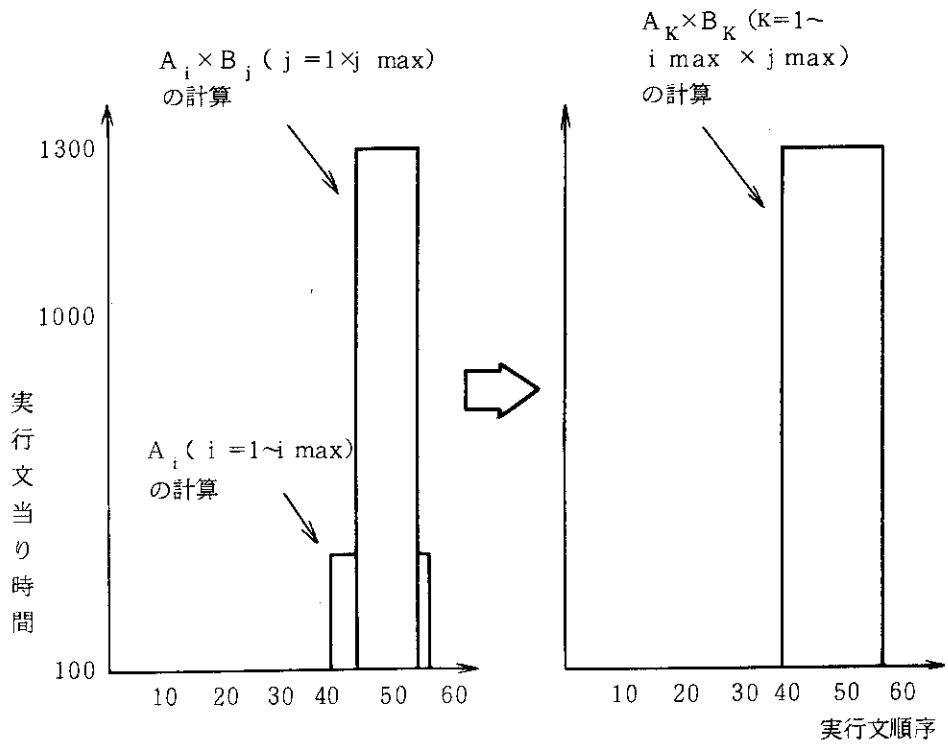


Fig. 3.2 Execution cost per statement of KCALC subroutine of EDDYTORUS.

他の代表例にKENO４（番号１９），ADPIC（番号３７）などがある。

モンテカルロ法コードKENO４，セル内粒子法コードADPICは共に計算時間の突出したルーチンを持つが，前者はｎ番目の世代の計算に（ｎ－1）番目の計算結果を使用し，またIF文による分岐が多い。後者は広い空間内の一部の領域に移流拡散する粒子群を追跡する。空間メッシュ，あるいは粒子群についてベクトル処理が可能であり，粒子の移流拡散が大きくない初期的状態においても５０～６０％程度のベクトル化ができる。しかしこのとき粒子の動きに関係のない空間座標についての余分な添字計算を含むことになり，実質的な計算時間はさほど短くならない。Table 3.1 の結果から，複雑な幾何形状の領域で発生する事象を部分毎に追うセル内粒子法，モンテカルロ法等の計算手法を採用しているコードはベクトル処理が難しいことがわかる。

いずれにしても上述の①レベルで調査する限りは結果はいつも同じで，調査対象となった原子力コードの５０％程度は性能向上を期待できず，残り５０％が1.5～２倍の性能向上を見込めるに過ぎない。

一方ベクトル計算処理を意識して開発したコード，あるいは変換したコードでは２～８倍の性能向上を得ている。このことから開発当初からベクトル計算処理を意識したコードの数をできるだけ増加させ標準コード・システムとし，それらコードの使用頻度をできるだけ高めれば，かなりの計算需要はベクトル計算機で吸収できると予想される。今後はこの方向の努力が必要

である。

これまでの原研での経験では，ベクトル計算で大巾な性能向上を期待できる場合はTable 3.1からも想像できるように次の条件a）～c） の論理積である。

a) 研究者が参加し最初からベクトル処理を意識してコードが作られているとき，あるいは既存コードであっても物理モデルを熟知する研究者の協力が得られる。

b) ベクトル化しやすい数値計算法が使われている。

c) コードの計算時間を消費する部分がFORTRAN で5,０００枚以内の２～３のサブルーチンに集中している。

上記の条件 b) ，c)は研究開発機関としての原研固有の理由によるもので，b) についていえば，原子力コードの並列化という問題に専門分野の研究者を多く投入することはできない。c) についていえば，１～２名のプログラマ，あるいは研究者が６ケ月程度の期間でベクトル計算向きにコードを書換えられなければ意味ないということである。特に条件 c）は重要で，計算センタとしてはこれから新規に作成されるコードができるだけこの条件ｃ）に合うよう

(i) 計算コードのモンジュラ化をはかり，ベクトル計算処理に適した部分をモジュール化して共用をはかる，

(ii) 計算時間のかかる部分はできるだけ短いサブルーチンに集中するようコードを作成する，

などの何らかの手を打つ必要がある。

3.3.2 定量的所見

原研の大型計算機システムは現在FACOM M−２００ 4CPUを持つ疎結合３システムである。このうちの２システム（３ＣＰＵ）は半年後にM−３８０（3CPU）２システムに置き換えられることになっている。この半年後のシステムを前提にベクトル計算機の利用を考える際にもっとも問題となるのは，現在処理されている原子力コードの何割が，どの程度の性能向上を期待してベクトル処理ができるかということである。これがわからなければ計算機の構成を定量的に考えることはできない。この件に関し発表されている資料は皆無に近い。わずかに米国アルゴンヌ国立研究所（ANL）において，１０本の原子力コードのＣＲＡＹ−１によるベンチマーク・テストが（3.4），また西独マックス・プランク研究所（ガルヒン）でいくつかの核融合関係の計算コードの同じくＣＲＡＹ−１によるベンチマーク・テストの結果が散見されるだけである（35）。結果はいずれも肯定的なものであるから，性能の出るコードを選んであるということであろう。このようなデータは原研の計算機構成を検討する上ではあまり役に立たない。

我々の所有するデータはTable 3.1 である。これは６年の歳月にわたって主として計算機技術者の約８３人月のマンパワーとかなりの計算機時間を投じて得られたものである。統計的推定をおこなうには通常１００のオーダーのデータ数が必要とされているが（36, 37, 38），その数のデータを集めるにはさらに時間が必要となるだろう。現在原研で開発使用されている計算コードはTable 3.1に現われているコードを変形したものが多いことがわかっているのでここではTable 3.1 を統計的推定の出発点とする。

3.3.2.1　ベクトル化率と2項分布

ベクトル化率は計算コードのベクトル処理可能性を見る上での重要な指標である。Table 3.1のデータからベクトル化率を縦軸，ソース・ステートメント数を横軸としてグラフを作るとFig．3.3のようになる。これによって調査のレベル，計算コードの大きさ，数値計算法などとベクトル化率の関係を一目で見ることができる。JT−60コード・ライブラリ系，RELAP系，ANISN，KENO系以外のほとんどの計算コードは60％以上のベクトル化率を持っている。

Table 3.1のデータからベクトル化率を横，縦軸とする度数分布図を作るとFig．3.4のようになる。

このFig．3.4の度数分布でベクトル化率が50％より下の部分に該当する計算コードRELAP，ANISN，KENO，JT−60コード・ライブラリを除外して新しく度数分布を作ってみよう。除外された計算コード群は原研では主として安全解析部，安全工学部と大型トカマク開発部の特定の研究室で使用されており，その年間計算量をほぼ正確に推定することができる。したがってこれらRELAP，ANISN，KENO，JT−60コード・ライブラリ系列の計算コード群は前もって原研全体の計算需要から取除き，スカラ計算機で処理するものとしよう。その計算量は昭和56年度においてはM−200CPU時間で約5631時間であった。これは全計算時間の約31パーセントに当る。その内訳はTable 3.2のとおりである。Fig．3.4からこのパーセントの計算量に当るコード群を取除いた残りのコードでベクトル化率を階級とする度数分布を作るとFig．3.5のようになる。

このFig．3.5の度数分布を2項分布に当てはめてみたのがTable 3.3である。この表の数値を求める手順は次のとおり。

Fig．3.5のベクトル化率50％以上の26本の計算コードを2項分布で近似すると

$$\text{級平均} = \frac{\sum_{0}^{2} X_i \cdot F_i}{\sum_{0}^{2} F_i} \quad , \quad F_i : \text{度数}$$

$$= \frac{29}{26} = 1.12$$

2項分布なら平均＝NP＝2・Pであるから

$$P = \frac{1.12}{2} = 0.56$$

N＝2，P＝0.56　のとき

X_i	P_i	nP_i
0	0.194	5.04
1	0.493	12.8
2	0.314	8.16

となり，χ^2 を

$$\chi^2 = \sum_{0}^{2} \frac{(F_i - nP_i)^2}{nP_i}$$ から計算すると

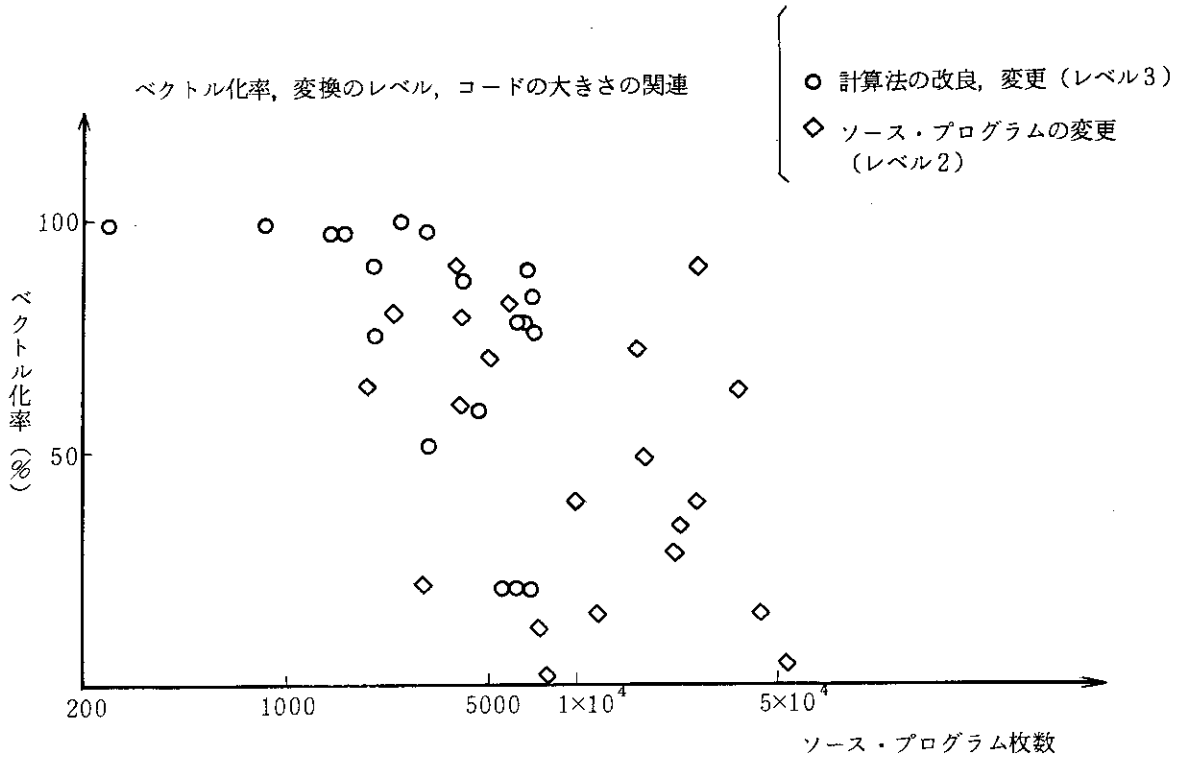


Fig. 3.3 Vectorized ratio, number of source statements, and level of investigation or modification on codes.

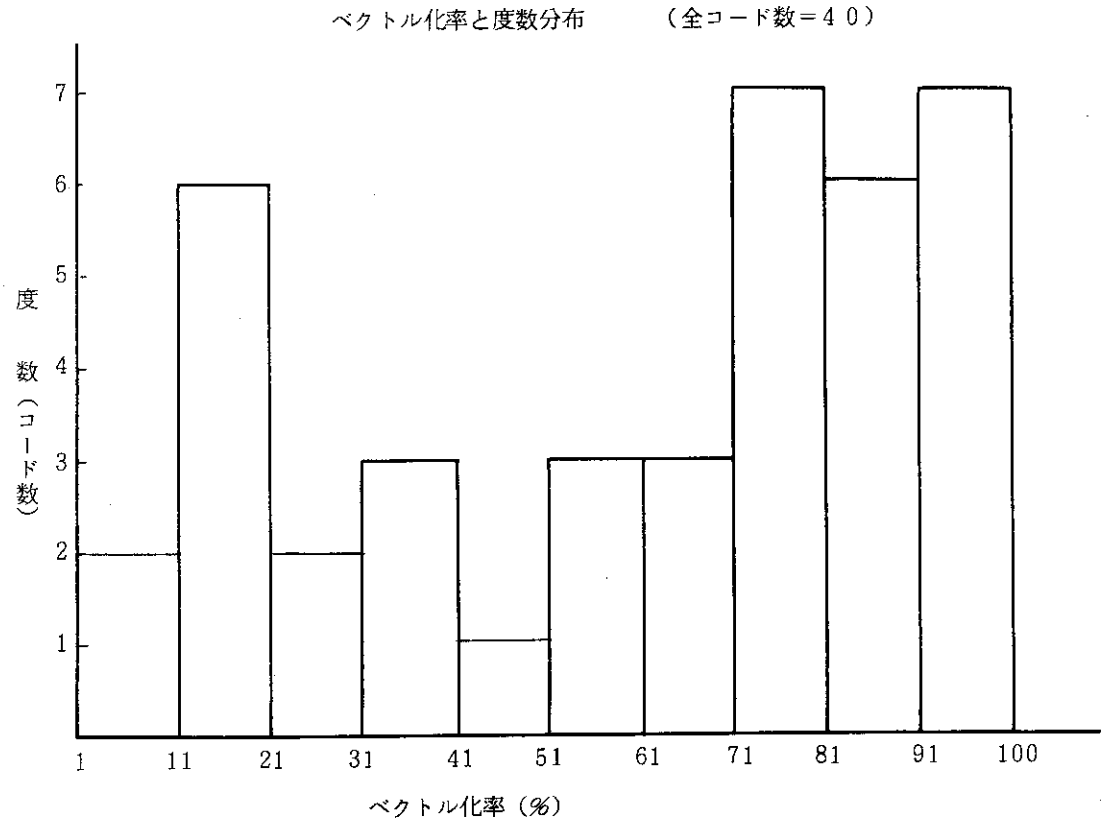


Fig. 3.4 Vectorized ratios and their frequencies of investigated 40 codes.

Table 3.2 Total CPU time used at JAERI in 1981 fiscal year and percent of nonvectorizable CPU time.

昭和５６年度使用ＣＰＵ量

M-200		13,421
F75APU		1,950×2.5
		計18,296時間
（ベクトル計算不適合）	大型トカマク開発部	2,015÷2
	安全工学部	2,937
	安全解析部	1,746
		計 5,691 (31%)

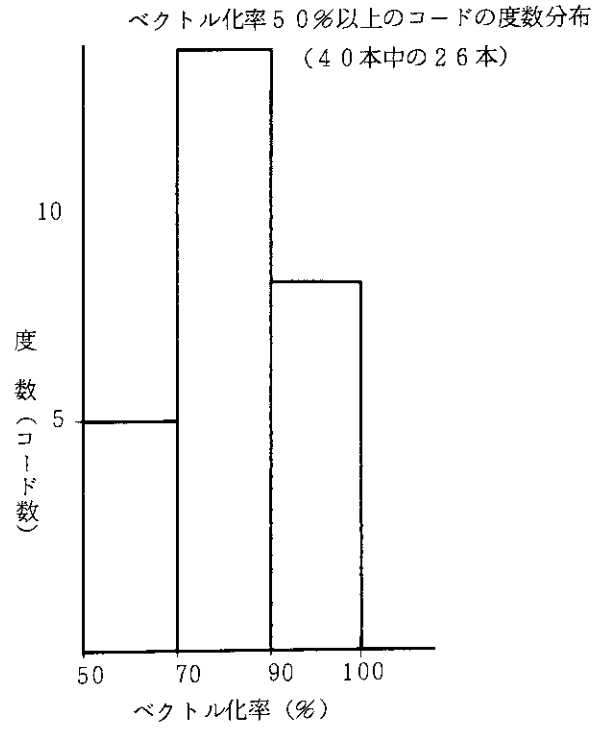


Fig. 3.5 Vectorized ratios and frequencies of highly vectorizable 26 codes.

Table 3.3 Frequencies and binomial distribution of 26 highly vectorizable nuclear codes.

階級	級中数	X_i	度数	B(N, X, P)	理論度数
50~69	60%	0	5	0.194	5.04
70~89	80	1	13	0.493	12.8
90~100	95	2	8	0.314	8.16
合計			26	1.001	26

$$x^2 = \frac{(5-5.04)^2}{5.04} + \frac{(13-12.8)^2}{12.8} + \frac{(8-8.16)^2}{8.16}$$

$$= \frac{(0.04)^2}{5.04} + \frac{(0.2)^2}{12.8} + \frac{(0.16)^2}{8.16} = 0.006$$

他方自由度2のx^2分布Pについて

$$P(x^2 \gg 0.01) = 0.995, \quad P(x^2 \gg 0.02) = 0.990$$

であるから，2項分布の仮定が正しくないとはいえない。もっとも統計的議論には2ケタ以上の標本数が必要とされているのでn＝26はあまりにも少ない。しかし原研で使用されている原子力コードは基本的なものの変形が多いので標本数を多くしても，それらコードのベクトル化率の分布はここで述べたものとあまり変らないであろう。

上述したように標本数が26程度で分布を論じるのは問題ではあるが，その数を増やすには相当の手間がかかるので，本報告では原研の年間全計算量の70パーセントに当るジョブのベクトル化率が上記の2項分布に従うものと仮定する（70パーセントは潜在的な可能性であって，これを達成するには，ベクトル計算機導入当初は2～3分のジョブもベクトル化をするなど相当の努力が必要である）。

このようにジョブのベクトル化率の分布が定められると，任意の仮想的ジョブ例$\{J_i\}$について，その個々のジョブJ_iが持つベクトル化率をランダムに与えることができ，ベクトル計算機の性能評価に大変便利である。この仮定を利用し，我々は第5章においてベクトル計算機の持つべきハードウェア的条件と運用方法を推定する。

3.3.2.2　ベクトル化に要する工数（マンパワー）

原研で利用されている計算コードの大部分，すなわち全CPU時間の7割程度を占めるコードは50%以上のベクトル化率を持っていると推定できるが，それらのコードをベクトル化するに要する手間はどの程度であろうか。この作業量が非常に大きなものであれば現実にはベクトル化は難しいことになろう。その量は少なくて済むことを以下に示そう。まず，或るコードをベクトル化して効果があるかどうかを判断するに要する手間を第3章，Table 3.1 のデータから求めるとTable 3.4 のようになる。これで見ると，ひとつのコードについてベクトル化効果を判断するために，そのコードの全ステートメント数の25%程度を調査していることになる。100%を25%にまで絞り込む手間は，既に述べたようにFORTUNEと呼ぶソフトウェア・ツールによって機械的に回避できる。

また同じ表からひとりのプログラマ（この場合は経験年数5年程度）は月7,600枚程度の処理能力を持ち，7600枚のうち1900枚程度を重点的に調査していることになる。これは通常のプログラミング作業の処理能率と比較してみるとわかりやすい。Fig. 3.6 は原研計算センタに外来研究員として計算機メーカから派遣されているプログラマのプログラミングに関する作業能率をまとめたものである（39）。調査期間は昭和52年度と53年度の2年間，調査対象プログラマは10名，調査対象コード数は58本であった。調査対象プログラマの能力と経験年数は，第3章，Table 3.1 のベクトル化作業をおこなったプログラマのそれとほ

Table 3.4 Manpower necessary for investigation or for modification on codes for vector processing.

ベクトル化調査に必要な工数

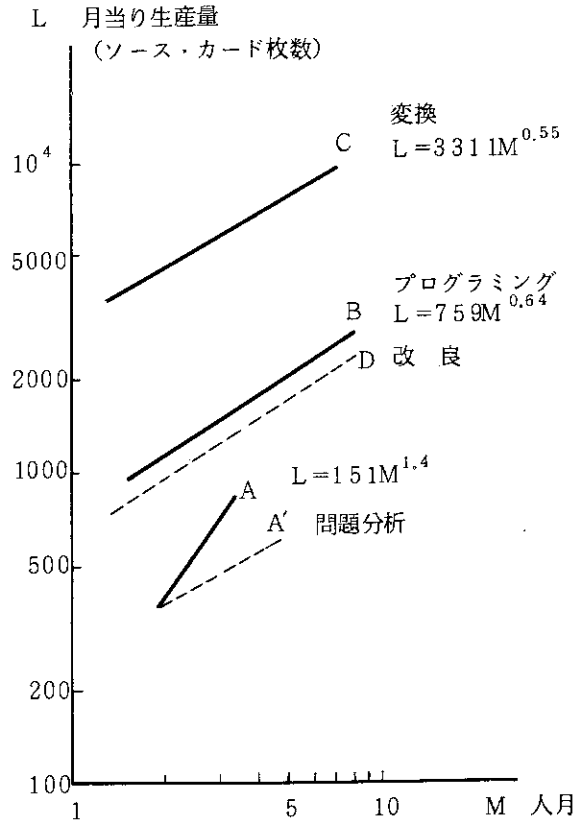
対象コード番号	A	B	所要人月	B／A	A／所要人月	B／所要人月
5, 6, 8, 9, 17, 19, 22, 24, 26, 31, 32, 33, 34, 35, 39, 40	128,500枚	32,137枚	17人月	25%	7,559枚／月	1,890枚／月

注　A：ステートメント数合計，B：調査対象ステートメント数合計，
対象コード番号は Table 3.1 のコード番号。

ベクトル化に必要な工数

対象コード番号	A	B	所要人月	B／A	A／所要人月	B／所要人月
1~4, 7, 10~16, 18, 20, 21, 23, 25, 27~30, 36, 37, 38	218,180枚	14,724枚	66人月	6.7%	3,306枚／月	223枚／月

注　A：ステートメント数合計，B：修正ステートメント数合計，
対象コード番号は Table 3.1 のコード番号。



注(1) 直線Aは後の調査で点線A′となった。

Fig. 3.6 Software productivity in scientific programming.

ぼ同等と考えられる。実際両者の作業をおこなったプログラマのうちの数名は同一人物である。５８本のコードは1,2の例外を除きFORTRANで書かれている。「変換」はCDC版コードをFACOM版コードに書換えるものがほとんどである。「定型的プログラミング」は依頼者の提供する仕様書通りに計算コードを作成すれば正しい結果が出るか，あるいは結果についてのチェックは依頼者の責任においておこなうものである。「問題分析」は，データと数値計算法との適合性のチェック，データに合った数値計算法の発見などをもプログラマに委ねられている作業をいう。Fig. 3.6の説明が少し長くなったが，本題に返えって，このFig. 3.6と前掲のTable 3.4を比較してみると次のようにいえる。

（ベクトル化が有効かどうかの調査に必要な工数）

(1) 計算コードをベクトル化して効果があるかどうかを調査する処理の能率は一般の計算コードを「変換」するそれの２倍である。

(2) ベクトル化の効果をみるには対象コードのFORTRAN文の２５％をしらべなければならないが，その部分を処理する能率は「定型的プログラミング」のそれの２倍である。あるいは「変換」の２分の１倍である。

（実際にベクトル化するに必要な工数）

ベクトル化の効果があるかどうかを判断するだけでなく，実際にベクトル化してスカラ演算の場合と同等の結果を出すのは少し手間がかかる。その手間はTable 3.4とFig. 3.6から以下のように推定することができる。

(1) 計算コードをベクトル化するための作業の能率は，一般の計算コードを「変換」するときのそれと同じである。

(2) ベクトル化するためには計算コードFORTRAN文の７％程度を変更，修正しなければならないが，その部分の作業能率は「問題分析」のそれに等しい。

以上から既存コードのベクトル化に必要な工数は，平均的にいえばCDC計算機で書かれたコードをＩＢＭ型計算機のそれに変換するのと同程度の手間であるといえる。

4. 計算需要からみた計算機構成

この章では原研計算センタの大型機で処理されている計算量（ＣＰＵ時間）のうち，ベクトル化可能な割合，そのとき期待できるベクトル化性能（第２章Fig. 2.7 のP（V，α），当面必要となるベクトル計算機の台数，およびベクトル計算機へ容易に移行するためのプログラミングの工数について推定する。

4.1 ベクトル計算処理可能なＣＰＵ時間の割合

この節では，原研計算センタの計算機システムで使用されている全CPU 時間のうちベクトル化可能な割合を推定する。前節で述べたベクトル化不適のコード，RELAP，KENO，ＪＴ－６０POV２などの利用は２，３の研究室に集中しており，その室名，使用ＣＰＵ 量をTable 4.1のように特定することができる。(*)

Table 4.1 Names of laboratories and divisions which comsume nonvectorizable CPU time.

部名 研究室名	使用ＣＰＵ時間		
	昭和５７年７月	昭和５７年８月	昭和５７年９月
大型トカマク開発部			
計画室	１４１×0.5	１７１×0.5	８５×0.5
安全工学部			
安全工学第１研究室	１４６	１１１	２１９
安全工学第２研究室	２２	１５	１７
安全解析部			
原子炉データ解析室	１５４	１７１	１８２
安全性コード開発室	５４	１７	２５
ＴＳＳ	６６	５８	５７
ＣＰＵ時間計	５１３	４５８	５４３

上記の表でＴＳＳとあるのはタイムシェアリング処理に使用されたＣＰＵ量である。昭和５７年７，８，９月に限らず年間を通してこの傾向は変らない。

計画室の使用量が0.5倍されているのは，ＣＰＵ時間のほぼ半分の量がＪＴ－６０POV２ とＬＩＢＪＴ－６０ コードを利用した計算に使用されたという意味である。同じ３カ月間の全計算

(*) 最近計算センタと計画室でおこなったＪＴ-60POV２ 最適化作業で，このコードの計算時間はスカラ計算で１０分の１に短縮された（２８）。

量はTable 4.2のようになる。この期間中のFACOM F230−75APUのAPU時間が2.5倍されているのは，このAPUを利用している核融合研究部理論解析研究室，物理部固体物理第1研究室のコードがよくベクトル化されていて対F230−75CPU比で7倍以上，M−200CPU換算で2.5倍の性能を出しているためである。

Table 4.2 Total CPU time and nonvectorizable CPU ratio of JAERI at July, Aug. and Sept. 1982.

CPU時間種別	使用CPU時間		
	昭和57年7月	昭和57年8月	昭和57年9月
M−200CPU 時間	1358	1374	1260
F75APU時間 (理論解析研究室 固体物理第 研究室 共 同 研 究)	161×2.5	137×2.5	299×2.5
全CPU時間	1760	1716	2008
ベクトル化不適比率	29%	27%	27%

4.2 当面必要となるベクトル計算機台数

原研計算センタにベクトル計算機を導入した場合の計算量の負荷配分については報告(18)で考察されている。そこでは昭和56年10月を現在とし，「現在」から3年後の計算需要の延びを予想し，ベクトル計算機を導入すべきであるとしている。このときベクトル計算機で処理できる計算量は，中・大型ジョブ（M−200計算機CPU時間で20分以内のジョブ）の1／3，巨大ジョブ（同2時間以内）の1／2，新規ジョブの3／5としていた。また，こうするために今後大型原子力コードのベクトル化技術の確立，およびコード変換・整備のための支援体制の確立など，相当の努力が必要であると結論づけている。

上記報告書の結論は大筋において正しいが，本報告第3.3.2節などのデータからわかるとおり，原子力コードのベクトル計算機への適応性はもっと広く，肯定的に考えてよい。また現在のところ不適応コードとその計算量，使用研究室を特定することができることは前節で述べた。したがって残る問題は，予算を別にすれば，導入された台数のベクトル計算機の性能に見合うだけの計算需要が移行期に存在するかどうかということ，およびスカラ計算機からベクトル計算機へ計算コードを円滑に移行させてゆくためのプログラミング工数はどの程度かということ，この2つである。以下この2つの問いへの定量的回答を試みよう。

4.2.1 移行期の計算需要

過去の経験をもとに移行期の計算需要について考えてみる。

原研計算センタにおける過去5～6年間の計算需要の増加は年率40パーセント程度である（Fig. 4.1 報告（40）から引用）。

原研計算センタの運用では夜間の長時間ジョブも2時間までのCPU時間という制限をつけられることが普通であり，そのために1ケース数10時間のジョブを1～2時間以内のジョブに分割してランさせている利用者もある。そのような利用者にとっては計算需要が年率40%で徐々に増加しているという考えに不審を感じられるであろう。実はこの疑問はあたっている。それを以下に説明する。Fig. 4.2 は報告（41）から引用したもので，原研計算センタの計算機システムの性能とCPU時間を示したものである。Fig. 4.2 において1964年から1969年までCPU時間が一定なのは1シフトで計算機を運転していたためである。また1969年から1971年の間にCPU時間が増加しているのは，開放利用制度を設け，研究者に夜間の計算機運転を認めたためである。1972年から1982年までの間の計算機運用体制は次のTable 4.3のとおりであった。

この表からFACOM230-60計算機時代の計算需要の伸びはシフト（直勤務）数の増加で吸収できたが，FACOM230-75以降は計算機性能評価作業とそれに続くシステムの改良か，あるいはより高性能な計算機へのリプレイスによるかしか方法がないことがわかる。現在の運用体制のもとでは年間の総CPU時間の上限は18,000時間程度である。F230-60 時代の総CPU時間の上限は15,000時間程度であった。F230-60，F230-75計算機でユーザ側が得るCPU時間の割合は年間通して65～70%程度，M-200では80～90%である。Fig. 4.1の昭和49年，52年のM-200換算CPU時間518, 3017 時間はFig. 4.2の計算機性能値からわかるとおり，当時においてはそれぞれ

518×6倍（F75分）×3倍（M-200分）≒9,300 ，

3,017×3倍≒9,000

時間である。これらのCPU時間はいずれもユーザ側の時間であるから，計算機の運転時間はそれぞれ9,300÷0.65≒14,300，　9,000÷0.65≒13,800　時間といっていずれも上限に近くなっている。

昭和50年はF230-60，F230-76計算機が併置運用された端境期であり，F230-75計算機システムが本格稼動を開始したのは昭和51年からである。この場合計算機のCPU性能は6倍向上したにもかかわらず，2年で処理の上限に近づいている。その後のF75CPU時間の伸びは主として計算センタと計算機メーカによる計算機性能評価作業とそれに続くシスムの改良に負うところが大きい。同じことがF230-75からM-200へ移行したときにもいえる。Fig. 4.1で54年第4四半期から55年上期までは端境期である。この図は55年度までであるが，56年度のユーザ側CPU時間は13,421時間であった。このときのM-200運転時間の総和は13,421÷0.8≒16,800　時間となり，上限に近づいている。F75 からM-200へのCPU性能の向上は3倍であった。

以上の議論から，過去においては，CPUの性能向上が6倍であれ，3倍であれ2年以内に処理の上限に達していることがわかる。それ以後のCPU時間の伸びは鈍化する。1982年の現在においてもかなりの潜在需要を抱えていることがわかっているので，CPU性能が6倍程度の計算機であっても2年後には処理の上限に達するといってよいであろう。長々と議論してきたが，ここではっきりいえることは，「原研は常にぼう大な計算需要をかかえており，6倍程度のCPU性能向上には2年程度で追いつく」ということである。

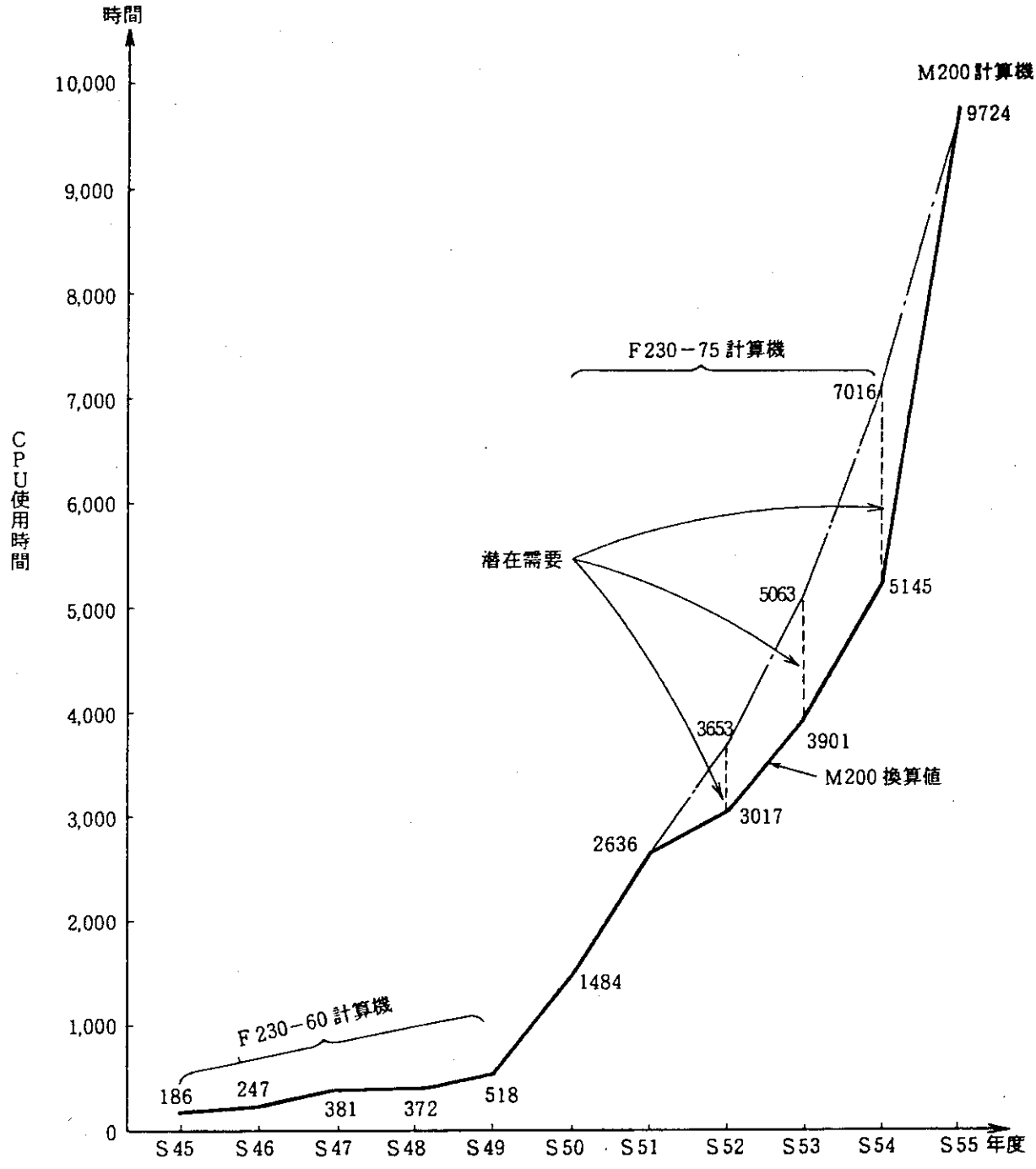


Fig. 4.1 Trend of computing demand of JAERI.

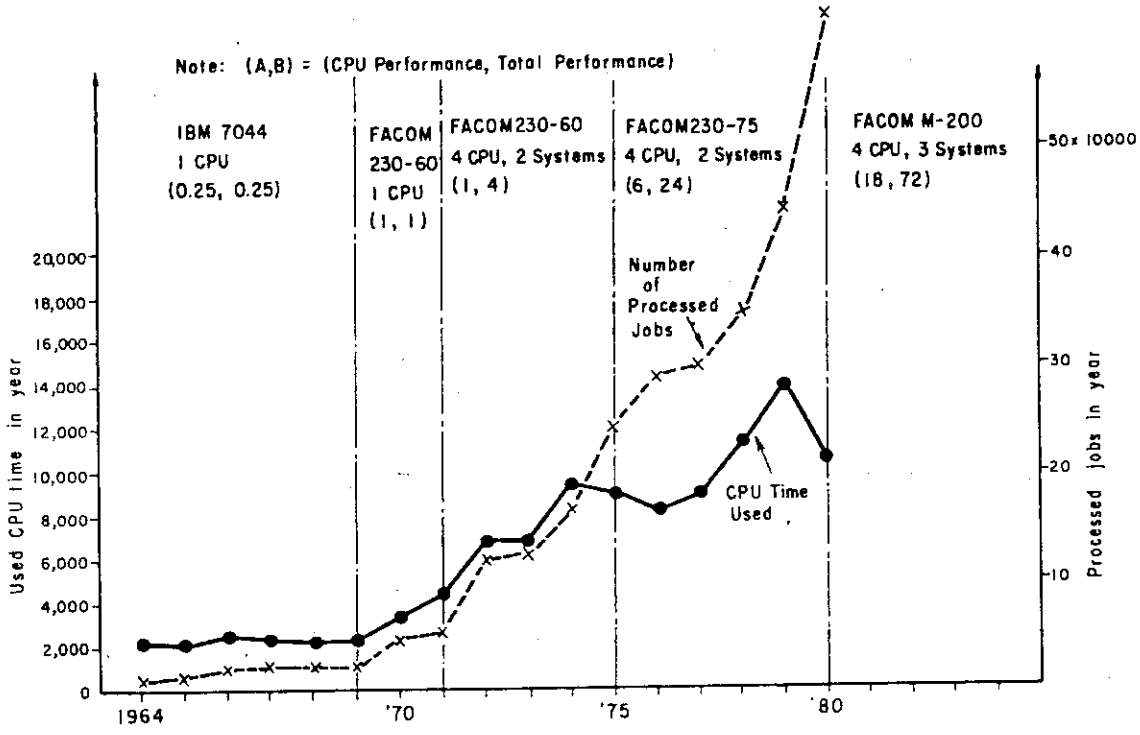


Fig. 4.2 Transition of computers and computing demand of JAERI.

Table 4.3 Transition of operational shifts at JAERI computing center.

年度		計算機運転体制
西暦	昭和	
1972	47	FACOM230−60#1 システム1シフト
1973	48	#2 システム3シフト
1974	49	FACOM230−60#1 システム2シフト
1975	50	#2 システム3シフト
1976	51	FACOM230−75#1, 2システム3シフト
〃	〃	以後すべて3シフト，ただし日曜，祭日は運転休止。

F230−75 はユーザ側のＣＰＵ使用率が低くてシステムの改良によって性能向上の余地を残していた。M−２００の場合は最初からユーザ側のＣＰＵ時間が多く，改善による性能向上はあまり期待できないので，現在の運転体制を継続する場合にはCPU 時間の伸びは急速に鈍化する。

4.2.2　計算量のベクトル計算適応性とコードの性能向上倍率

大量の計算を比較的安価に処理する方法のひとつにベクトル計算機の採用がある。ベクトル計算機を導入するための前提条件としてベクトル計算処理に適合する原子力コードの数が多くなければならない。Fig．3.5の２６本のコードが同じ計算時間の重みを持つとすれば，性能 α（αについては第2.3節）のベクトル計算機では１本当りのコードの性能向上倍率P（V，α）はTable 3.3のベクトル化率と性能向上倍率式（第2.3節）を使い，

α＝５のとき　（５本×1.6倍＋１３×2.8＋８×4.2）÷２６＝３倍／本

α＝１０のとき　（5×2.2＋１３×3.6＋８×9.5）÷２６＝5.1 倍／本

となる。

4.2.3　計算需要からみたベクトル計算機台数

本節では当面必要となるベクトル計算機の台数と性能を，第3.3.2節のデータ及び第3.2節Table　3.1の調査結果一覧表を使って推定してみる。

昭和５７年１１月現在から昭和５８年７月への原研計算センタの計算能力は，M−２００計算機の１台のCPU 性能を１としたとき，Fig．4.3のようになる。F75APU に0.6の係数がかかっているのは，この計算機が３シフトではなく，２シフトで使用されているためである。この計算機は昭和５７年１２月の「並列処理計算手法」に関する原研・富士通(株)の共同研究終了時に撤去された。M−２００計算機１ＣＰＵ，１システムは1.5シフト間はタイムシェアリング処理，そして３シフト通して疎結合３システムの入出力プロセッサとして動いている。したがって厳密にいえば，このシステムの計算能力も0.5倍すべきであるが，議論を簡単にするため1.0としておこう。

ベクトル計算機を導入できるのは早くても昭和５９年度以降になるであろうから，構成変更の出発点を昭和５８年７月時点のシステムに置くことにする。同程度の借料増で次のFig．4.4（第１案）とFig．4.5（第２案）の２つの案が考えられる。いずれの案でもグローバル・プロセッサの役割を果しているM−２００はM−３８０に交換することが望ましい。現在のM−２００が過負荷の傾向にあるので，疎結合方式を前提とすればM−３８０にすべきである。

第１案，第２案の特長を挙げる。

第１案の特長

(1) 第３章Table　3.1のデータから推定できるように，ここ数年は配列要素数（ベクトル長）が大きくない原子力コードが多い。したがってα値＝１０のベクトル計算機１台よりはα値＝５のベクトル計算機２台のほうがコードの特性に合う（第２章Fig．2.6 からわかるようにα値はベクトル長にかなり依存する）。

(2) ベクトル化されたジョブを多数処理でき，個々のジョブの回転時間が短縮されるため，

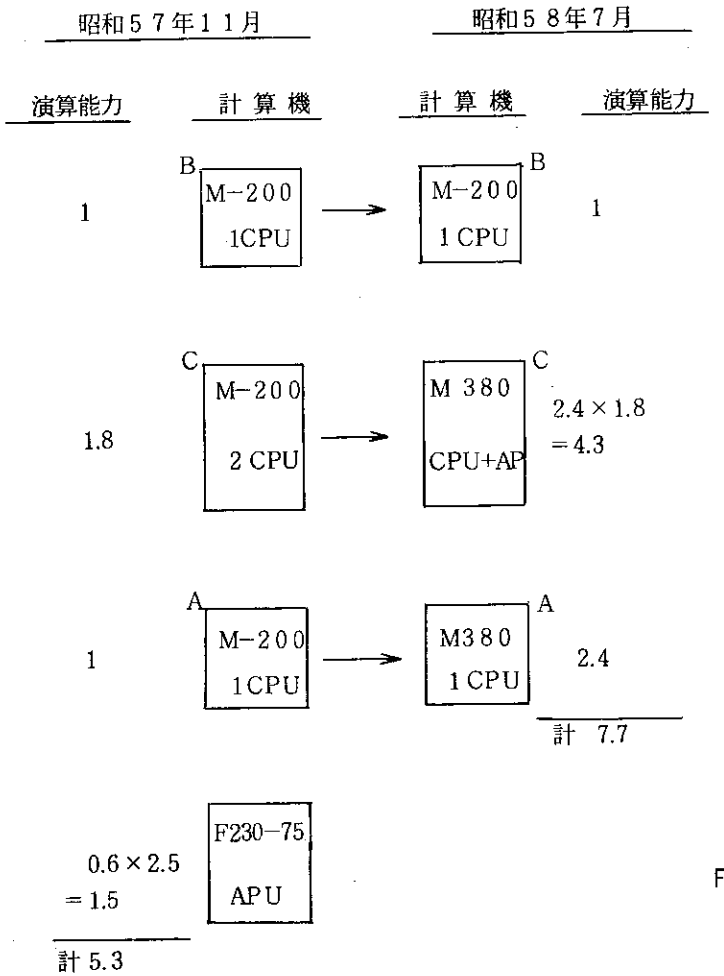


Fig. 4.3 Current computer systems(Nov. 1982) and systems at June 1983 of JAERI.

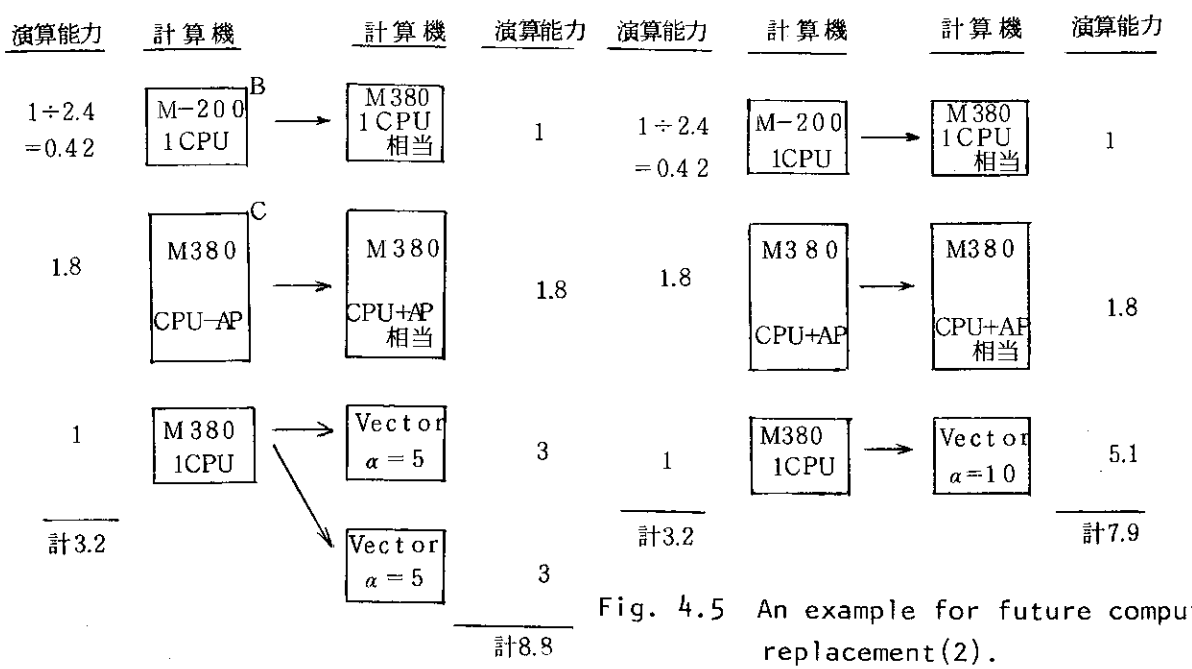


Fig. 4.5 An example for future computer replacement(2).

Fig. 4.4 An example for future computer replacement(1).

ユーザの計算をベクトル計算処理へ誘導することが容易となる。

第2案の特長

(1) 高いベクトル化率，長い配列要素数を持つコードでは高性能を発揮する。したがって特定のコード群についての専用機的使用法では高性能を期待できる。

第1案，第2案ともにスカラ計算量は36パーセント程度である。第2案の場合は$\alpha=10$の性能をフルに発揮できる多数のコードの存在が前提となっているが，現在まで得られたデータでは$\alpha=5$に合うコードが多い。したがって当面は第1案を採用し，30パーセント残っているスカラの計算量をベクトル化できる見通しがついた時点でM−380AP（付加プロセッサAP付きM−380）をより高性能のベクトル計算機に交換する案をまず検討すべきである。

4.2.4 ベクトル計算機への移行に必要なプログラミング工数

ベクトル計算機はスカラ計算機とは異なり，導入したその日から直ちに性能を発揮できるわけではない。性能をフルに引き出すためには計算コードのかなりの本数を事前にベクトル計算機向きに変更しておく必要がある。

それではどの程度の工数が必要となるであろうか。Table 4.4 は昭和57年7，8，9月のM−200計算機3システムのバッチ・ジョブ処理の統計である。これからわかるようにCPU時間が5分以上，件数にして約7パーセントのバッチ・ジョブが全バッチCPU時間の76パーセントを占めている（57年7月の例）。同じ昭和57年7，8，9月の各月毎に，平均5分以上のジョブを実行した職員数は52名，外注その他では17名であった。52名の職員に限ってその使用コード名，使用人数，コードの規模等について調査した結果をTable 4.5に示す。

このTable 4.5の外国コードのほとんどはTable 3.1 で調査済で，現在のところベクトル化が難しいものが多い。原研開発コードのうちTHYDE−P（Table 3.1 の24番），LIBJT−60（IMPURV3はLIBJT−60の一部），JT60POV2（Table 3.1の39，40番）も同じく一度調査をしており，現在のところベクトル化については悲観的である。自己開発コードは研究者本人が外注，あるいは自己作成したもので，その使用が本人自身にとどまっているものをさす。Table 4.5 から，研究者個人で使用しているコード数は50÷21＝2.4本，1本当りの大きさは118200÷50＝2364枚となる。また原研開発共用コードの1本当りの大きさは206000÷12＝17200枚，利用者1人当りの大きさは206000÷19＝10800枚となる。ベクトル化が難しいとわかっているコードを除いた残りのコードについて工数を計算するとTable 4.6のようになる。

ベクトル化までおこなうときに必要となる70人月には調査段階の30人月も含まれている。ベクトル化に要する工数はこれでおわりではない。コードは生きもので，間断なく改良・修正がおこなわれているから継続してベクトル化の工数が必要になる。なおTable 4.5をまとめる対象となった利用者のリストを付録1に挙げておく。

以上のデータから60〜70%程度のCPU使用量を占めるベクトル化可能なコードの変換に要する工数は約70人月である。このうち研究者の作業は比較的少なく，これまでの経験からいえば1〜2割程度で，残りはプログラマの作業量である。逆にいえば，70人月のプログラ

マの作業量があればベクトル計算機への移行は容易になるといえる。研究者が自分の研究に使用するコードの寿命は，これまでの経験からいえば，２～３年である。したがってベクトル計算手法が広く研究者の常識となるまでは毎年７０人月程度の作業量が必要である。

他方３０％のＣＰＵ使用量を占める計算コード群をベクトル計算処理に適した形に変換することは容易ではない。その代表例であるＲＥＬＡＰ系コード群の開発と変換に要した工数を付録２に掲げる。これらのコードのどれも単純な変換だけでは５０％以上のベクトル化率は得られないことがわかっている。

以上の議論をまとめるとTable　4.7のようになる。

Table 4.4 CPU time, I/O accesses consumed in July, Aug. and Sept. 1982.

1982.7

UNIT:
CPU = HOUR
MEMORY = X KB
I/O = X K
GET/PUT(TSS) = X K

*** JOB CLASS***

	JOBS	CPU TIME	I/O	AVERAGE MEMORY
S	26676	59.46	23613.79	607.14
M	10378	343.65	26563.35	1000.94
L	1650	138.47	5458.51	1426.38
G	176	54.99	4425.04	1399.16
NGT	899	262.51	6168.71	1243.21
ETC.	1365	434.13	12307.78	935.32
TOTAL	41144	1293.21	78537.12	767.50
TSS	37323	62.82	3067.67	284.47

*** T-VALUE ***

	JOBS	CPU TIME	I/O	AVERAGE MEMORY
LE 1S	10342	1.00	1390.05	360.86
LE 10S	12669	14.04	13870.91	711.89
LE 1M	9169	64.64	22080.59	930.96
LE 5M	6018	230.69	25350.55	1076.80
LE 10M	1737	201.20	6228.95	1185.27
LE 30M	714	170.96	5051.00	1414.95
LE 1H	212	158.57	2521.77	1606.55
LE 2H	243	372.57	1669.42	1389.15
UP	40	80.00	389.12	1590.80

Table 4.4 (Cont'd)

1982.8

UNIT:
CPU = HOUR
MEMORY = X KB
I/O = X K
GET/PUT(TSS) = X K

*** JOB CLASS***

	JOBS	CPU TIME	I/O	AVERAGE MEMORY
S	23224	49.82	20647.20	598.00
M	11106	305.55	24959.60	1043.93
L	1784	160.51	5600.37	1457.15
G	180	49.00	3902.51	1576.47
NGT	1076	373.07	5251.37	1174.98
ETC.	1276	377.09	10693.93	1021.75
TOTAL	38646	1315.04	71054.94	800.43
TSS	32095	58.24	3498.82	291.93

*** T-VALUE ***

	JOBS	CPU TIME	I/O	AVERAGE MEMORY
LE 1S	9778	0.95	1260.00	368.04
LE 10S	11469	12.95	14063.62	771.35
LE 1M	8548	63.13	19846.48	933.13
LE 5M	5784	207.75	21947.46	1118.30
LE 10M	1617	186.17	5833.25	1323.28
LE 30M	818	191.54	4387.98	1295.02
LE 1H	394	266.16	2431.53	1252.11
LE 2H	211	332.81	1052.95	1372.55
UP	27	54.00	246.38	2272.59

Table 4.4 (Cont'd)

1982:9

UNIT:
CPU = HOUR
MEMORY = X KB
I/O = X K
GET/PUT(TSS) = X K

*** JOB CLASS***

	JOBS	CPU TIME	I/O	AVERAGE MEMORY
S	23469	51.08	22767.18	628.78
M	9451	278.50	28179.66	1044.78
L	1581	132.43	4500.82	1377.21
G	269	112.64	4186.50	1079.93
NGT	964	246.15	5409.35	1337.85
ETC.	1298	382.09	8908.45	891.71
TOTAL	37032	1202.90	73951.94	797.85
TSS	32644	55.56	3270.42	275.73

*** T-VALUE ***

	JOBS	CPU TIME	I/O	AVERAGE MEMORY
LE 1S	9224	0.89	1069.53	387.12
LE 10S	11531	13.16	14738.25	749.59
LE 1M	8420	61.07	20136.98	952.23
LE 5M	5290	184.53	26124.77	1136.66
LE 10M	1405	170.62	4527.65	1160.81
LE 30M	672	179.16	3715.94	1359.14
LE 1H	243	184.02	1941.59	1586.11
LE 2H	228	371.66	1493.22	1483.42
UP	19	38.00	217.01	1735.16

Table 4.5 Number of users of large-scale codes, code names and total source statements used in July, Aug. and Sept. 1982.

コード名（ソース枚数）	利用者数	ソース枚数	コード数
外国コード KENO 4, RELAP 4, RELAP 5, TWOTRAN, CITATION, ORIGEN 2, DOT 3.5, ANISN, RETRAN	12名		9
原研開発コード THYDE-B1 (15000), THYDE -P(26000), EQUCIR(10000), RADHEAT-V3(20000), LIBJT 60(30000), MCINP(18000), JT60POV2(23000), SLOWALFA (3000), SLOWIN9(4000), SRAC(50000), FEM(2000), FEDM(5000)	19	206000	12
個人開発コード	21	118200	50

Table 4.6 Necessary manpower for vectorization of current JAERI's big, vectorizable codes.

作業内容	必要工数
ベルトル化調査のみ	(112000+118200) ÷7600=30人月
ベクトル化まで	(112000+118200) ÷3300=70人月

Table 4.7 Estimated manpower per year necessary for vectorization of 70 percent of CPU time at JAERI.

CPU量	ベクトル化可能性	必要工数
70%のジョブ群	可　能	70人月／年
30%のジョブ群	?計算手法の研究開発必要	?250人月

5. ベクトル計算機の持つべき資源と機能

第3章で原研の計算需要とそれのベクトル計算機への適応性，第4章で当面必要となるベクトル計算機の台数を論じた。この章では1台のベクトル計算機に着目し，その計算機の持つべき主記憶容量，入出力バッファ容量等について推定する。

5.1 計算機必要資源推定の方法

推定は次のようにしておこなう。

(1) ジョブ・パターンとベクトル化率

昭和57年1月21日の原研計算機システムは大変混雑していたので，その様子を見るために会計情報ファイルからジョブ情報を抜き出して整理してある。この情報を加工してベクトル計算機用のジョブ系列を作り出すことにする。原研計算センタのA，BおよびCの3つの大型計算機システムのうち，AシステムとCシステムに1月21日9時から1月22日7時までに投入されたジョブ列から，CPU時間がそれぞれ120秒(A)，300秒(C)以上の63，75件を抜き出し，それらジョブにランダムにベクトル化率を与える。その与え方は第3章Table 3.3の2項分布に従うようにする。

(2) ベクトル化によるCPU時間の短縮

ジョブ系列中の各ジョブのCPU時間C_iを与えられたベクトル化率V_iによって短縮する。短縮後のジョブのCPU時間は$C_i \div P_i$とする。ここでP_iは2.3節の(2)式で次のように与えられる。

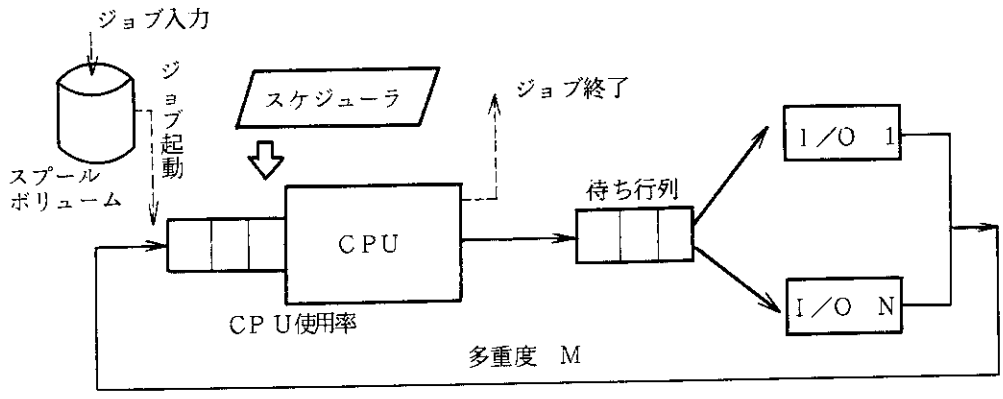
$$P_i = \frac{1}{\frac{V_i}{\alpha} + (1 - V_i)}$$

ここでαの値として$\alpha = 5$，または10を採用する。

(3) 待ち行列モデルによる最適多重度の決定

ベクトル計算機と2次記憶（磁気ディスク）装置から成る系（Fig. 5.1 の計算機システム）を想定する。このシステムで上記(2)のジョブ系列が与えられたとき，ユーザ側のCPU使用率を最大にする多重度を求める。多重度をどんどん上げてゆけばユーザ側のCPU使用率は上限の一定値に近づく。

その後は多重度をいくら上げても計算機のメモリ，チャネル，オペレイティング・システム（OS）のオーバヘッドを余計に消費するだけでユーザ側のCPU量は増加しない。



注 ジョブ多重度Mで動いているシステムで，ジョブ終了の度にスケジューラは
スプール・ボリュームからひとつのジョブの起動を試みる。

Fig. 5.1 Simulation model based on queueing theory.

(4) メモリ（主記憶）容量の決定

（4a） ユーザ・プログラム用実メモリ量

ユーザ側のＣＰＵ使用率が最大の値になるように多重度を変更してゆくにつれ，多重度の枠内で起動されているジョブのメモリの総和は変化する。この変化するメモリ総和の最大値は，差し当り必要なユーザ・プログラム用実メモリ量を表わしている。ただしベクトル計算機が必要とするメモリはすべて仮想記憶ではなく実記憶方式で与えられると仮定する。

（4b） ワーク・ファイル代替実メモリ量

計算の途中結果を２次記憶装置に収納することは，どの計算コードにおいても一般的におこなわれている。ベクトル計算機では，この作業用（ワーク）ファイルの容量を実メモリに持つことが望ましい。ＩＢＭ互換のＭシリーズ計算機では，これが仮想入出力（Virtual I/O，略してVIO）機能として実現されている。

仮想入出力の機能を使えば，ディスク・ファイルの入出力回数を減らすことができる。これはFig. 5.2 のようにメモリ上にバッファを置き，ファイルへの入出力をこのバッファへの入出力とする。こうすることでディスオ装置への入出力を減少させることができる。現在はワーク・ファイルならこの機能を利用することができる。このためには対象ファイルをメモリ上に置かなければならず実メモリが必要となる。

上記(3)の方法で求めた多重度，および別の資料（後述 5.3 節）から求めた１ジョブ当りのワーク・ファイル容量をかけ合せてワーク・ファイル代替実メモリ量を推定する。この実メモリ量は，第３章Table 3.1 の調査時に一緒にデータを収集すべきであったが，当時はメモリが安価でないことからおこなっていない。

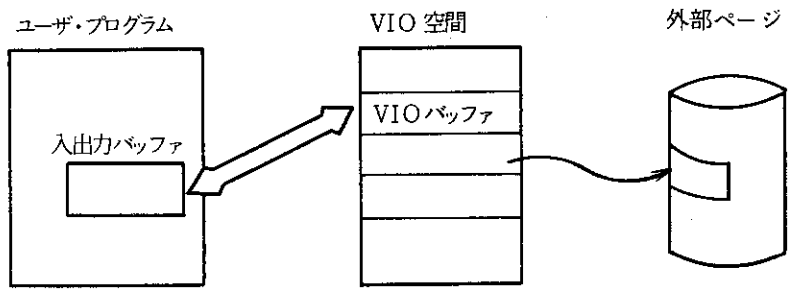


Fig. 5.2 Scheme of virtual I/O.

(5) 必要な並行入出力パス（径路）数の推定

計算機システムが入出力バッファ装置を持たないとき，入出力待ちを生じないだけの並行動作する入出力パスの数を推定する。

(6) ユーザ・ファイル用入出力バッファ効果の推定

ユーザ・ファイルに高速で読み書きできれば，それだけベクトル計算機の性能が生きてくる場合もあろう。ユーザ・ファイル用入出力バッファ装置の存在を仮定し，その性能でジョブ処理性能がどれだけ向上するかを推定する。

5.2 必要資源種類と量の推定計算

この節ではベクトル計算機を有効利用するに必要な資源の種類と量を5.1節で述べた順序に従って計算で推定する。

5.2.1 最適ジョブ多重度の計算

ここで使用する待ち行列理論に基づいた計算方法は，報告(42)と論文(43)にその原型がある。しかしこの原型版は近似精度があまり良くない。この節で使用する拡張版ではそれがある程度改善されている。その理論と計算式は付録3に掲げる。ここでは計算に使用したデータと計算結果について説明する。

(1) 計算の方法

Fig. 5.1 のバッチ処理用計算機システムを考える。ＣＰＵの入口にスケジューラがある。ジョブがスプール・ボリュームに入力された時刻 t_i か，あるいはジョブの処理が終了した時刻 t_i に，スケジューラはスプール・ボリュームを調べ，ジョブがあればその起動を試みる。起動したと仮定して推定したＣＰＵ使用率 $\rho_i^{(2)}$ を，起動しなかったときの使用率 $\rho_i^{(1)}$ と比較し，

$\rho_i^{(2)} > \rho_i^{(1)}$ ならそのジョブを起動する，

$\rho_i^{(2)} < \rho_i^{(1)}$ ならそのジョブを起動しない。

この操作を $t_1, t_2, \cdots\cdots, t_n$ について繰返す。この過程で多重度Ｍは変動する。これを可変多重度（Variable degree of multiprogramming)と呼ぶ。現在の計算機で可変多重度を採用することは技術的に可能であるが，おこなわれていない。これに対し，固定多重度（Fixed degree of multiprogramming）は現代のすべての計算機で採用されている。

Fig. 5.1のシステムでジョブの起動は先着順でおこなわれ，実メモリの制限はないとする。

(2) 多重度推定の方法

我々の計算モデルではスプール・ボリュームから J_i を起動し，次に J_{i+1} を起動しようとして $\rho_i^{(2)} < \rho_i^{(1)}$ となったとき J_{i+1} は起動されない。このとき J_{i+2} を起動するか，あるいは J_{i+1}，J_{i+2} を同時に起動すれば $\rho_i^{(2)} > \rho_i^{(1)}$ となることもあるが，我々の計算モデルではこのような場合を考慮していない。したがって我々の計算モデルによるＣＰＵ使用率 ρ は最良の ρ の第1近似値を与えることになる。

ρ の第1近似値が求まると，これの時間平均値 $\bar{\rho}$ を計算し，次に $\bar{\rho}$ に近いいくつかの固定多重度でジョブ処理をおこなう。この結果ジョブ処理経過時間の一番短い固定多重度を求める多重度とする。本報告では ρ は第1近似値で十分なので第1近似値のみを使用する。

(3) ジョブ・パターン

このモデルで処理するジョブ・パターンは計算センタにおける実際のジョブ処理から選んだ。具体的には昭和５７年１月２１日にＡシステム，あるいはＣシステムに入力されたＣＰＵ時間がそれぞれ１２０秒（Ａシステム），３００秒（Ｃシステム）以上のジョブを対象としてシミ

ュレーションをおこなう。

スプールへの入出力，その他でユーザ側が負うべきシステムの入出力回数を1ジョブにつきAシステムでは1037，Cシステムでは1786 回加算してある。その数値はTable 5.1のデータから求めた。

Table 5.1 System I/O accesses which should be added to users'.

ボリューム	データセット名	入出力回数
user 22	proclibr.	34757
user 08	check point	16306
user 09	DSCF	67915
PAGE03	MANX, MANY	46751
SYSPL2		17114
RACF01		16189
RACF02		18198
SYSPL1		21210
		238440

82. 1. 21 #A
14:01~17:00
ジョブ・ステップ602
ジョブ230
1037回／ジョブ

ボリューム	データセット名	入出力回数
user 20	link libr.	89109
user 21	link libr.	51409
user 08	check point	18275
user 09	DSCF	74030
SYSPL2		25953
RACF01		22441
RACF02		22248
SYSOSC		18265
PAGE02	MANX, MANY	84776
SYSPL1		32852
		439348

82. 1. 21 #C
14:01~17:00
ジョブ・ステップ710
ジョブ246
1786回／ジョブ

Aシステムでスカラ演算性能＝1.0倍，ベクトル演算性能＝1.0倍のときのジョブ列をTable 5.2に，それらジョブのCPU時間をFig. 5.3 に示す。ベクトル演算性能＝5.0，10.0 の場合をそれぞれTable 5.3, Fig. 5.4, Table 5.4, Fig. 5.5 に示す。なお，Table 5.2以下のUCPU, VCPU の単位は10ミリ秒である。

同じくCシステムについてはTable 5.5, 5.6, 5.7, Fig. 5.6, 5.7, 5.8に示す。

Table 5.2 Job pattern for simulation of A computer system(scalar=1.0, vector=1.0).

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
164	F3380707	00000001	21122130	21123434	0	20851	1600	1192	0	177	2	0.95	20851
4	F2937874	00000002	21091013	21093338	0	57807	1600	1420	0	3238	2	0.95	57807
337	F8224117	00000003	22065456	22075522	0	88758	2112	2056	0	754	2	0.95	88758
514	C1482165	00000004	21184050	21194341	0	120006	832	724	0	16442	2	0.95	120006
201	F2937881	00000005	21123932	21131251	0	57899	1600	1404	0	3235	2	0.95	57899
22	F3701452	00000006	21101059	21102603	0	27482	3136	2844	0	6063	2	0.95	27482
596	F2036692	00000007	22012544	22020504	0	63177	1088	1036	0	324	2	0.95	63177
256	C3022925	00000008	21135224	21143338	0	93987	832	832	0	670	2	0.95	93987
120	C2335593	00000009	21105729	21111229	0	18506	1600	1536	0	8778	2	0.95	18506
466	F9261810	00000010	21221536	21225531	0	60008	2112	1700	0	2889	2	0.95	60008
675	C8234011	00000011	22011000	22014059	0	21329	1088	592	0	24410	2	0.95	21329
629	F3089037	00000012	22071754	22074039	0	13296	1600	660	0	31834	2	0.95	13296
660	F9238155	00000013	22062909	22064343	0	19997	2112	832	0	2332	2	0.95	19997
522	F9805182	00000014	21184312	21184840	0	22699	2112	1612	0	115	2	0.95	22699
673	C8233025	00000015	21221940	22010034	0	358360	1600	1448	0	9019	2	0.95	358360
276	09265976	00000016	21144750	21151526	0	49614	2112	1900	0	957	2	0.95	49614
527	F2347060	00000017	22024901	22055021	0	271029	4160	3556	0	31157	2	0.95	271029
226	F3044903	00000018	21131254	21132740	0	24422	1600	1536	0	5054	2	0.95	24422
674	C3022941	00000019	22002422	22064634	0	719581	832	832	0	743	2	0.95	719581
421	C9265978	00000020	21221814	22002102	0	273127	2112	1936	0	2372	1	0.80	273127
7	F2937877	00000021	21092952	21095644	0	58196	1600	1420	0	3439	1	0.80	58196
150	F3023714	00000022	21115516	21121914	0	45367	1600	1016	0	12438	1	0.80	45367
220	F1185740	00000023	22010053	22012542	0	14071	1088	832	0	29398	1	0.80	14071
214	09265972	00000024	21130322	21132229	0	50712	2112	1900	0	954	1	0.80	50712
56	F9667008	00000025	21105406	21111415	0	12163	1600	1548	0	4206	1	0.80	12163
193	N8181214	00000026	22055023	22062906	0	66318	1600	1308	0	1313	1	0.80	66318
242	F2937882	00000027	21132234	21135155	0	57863	1600	1404	0	3230	1	0.80	57863
5	F3701451	00000028	21091436	21092948	0	26966	3136	2844	0	6539	1	0.80	26966
23	F1560969	00000029	21094518	21101043	0	41874	2112	1904	0	10262	1	0.80	41874
156	F3023716	00000030	21194343	21200547	0	43115	1600	1016	0	11841	1	0.80	43115
151	N8192611	00000031	21143734	21151212	0	60007	1600	1492	0	10607	1	0.80	60007
127	C3022919	00000032	21111232	21113953	0	93554	832	832	0	665	1	0.80	93554
558	N8192616	00000033	22051737	22073240	0	359240	1600	1492	0	13622	1	0.80	359240
370	F2937885	00000034	21151217	21154140	0	57894	1600	1396	0	3235	1	0.80	57894
401	C3022929	00000035	21155618	21162643	0	95584	832	832	0	667	1	0.80	95584
518	C3022932	00000036	21174213	21180835	0	96742	832	832	0	668	1	0.80	96742
425	C9506417	00000037	21162645	21164956	0	71557	832	832	0	479	1	0.80	71557
426	C9506417	00000038	21165431	21172106	0	82185	832	832	0	479	1	0.80	82185
443	F2937886	00000039	21172307	21174211	0	57854	1600	1404	0	3236	1	0.80	57854
227	F3044904	00000040	21132847	21135216	0	31049	1600	1536	0	5045	1	0.80	31049
626	F9082718	00000041	22073338	22074157	0	17512	1600	1560	0	38	1	0.80	17512
676	C9092577	00000042	22083601	22092132	0	25920	1600	1536	0	5577	1	0.80	25920
568	F2937887	00000043	21184845	21191643	0	57825	1600	1420	0	3198	1	0.80	57825
520	C3022933	00000044	21181909	21184309	0	76394	832	832	0	662	1	0.80	76394
595	F2036690	00000045	22005013	22031902	0	264214	1088	1036	0	326	1	0.80	264214
3	F3701450	00000046	21090331	21091009	0	13030	3136	2844	0	3414	1	0.80	13030
173	F9805177	00000047	21121921	21123928	0	41790	2112	1604	0	95	1	0.80	41790
541	F3349632	00000048	21181207	21182923	0	12369	1600	1196	0	3005	1	0.80	12369
251	F9280955	00000049	21140321	21143411	0	60005	1600	1376	0	1661	1	0.80	60005
644	F3363378	00000050	22020507	22022009	0	24548	2624	2580	0	67	1	0.80	24548
284	09265977	00000051	21154149	21155558	0	49578	2112	1900	0	957	1	0.80	49578
602	F2031126	00000052	22033127	22051143	0	176600	4160	3660	0	19243	1	0.80	176600
525	F9805183	00000053	21191648	21192633	0	21404	2112	1612	0	116	1	0.80	21404
19	F3380701	00000054	21093341	21094515	0	22140	1600	1192	0	177	0	0.60	22140
219	F1185739	00000055	22002517	22004821	0	14050	1088	832	0	29397	0	0.60	14050
176	F3071527	00000056	21124327	21130458	0	20936	832	768	0	9987	0	0.60	20936
240	F1560973	00000057	22022011	22024856	0	46507	3136	2692	0	2855	0	0.60	46507
142	F2937880	00000058	21112235	21114903	0	57902	1600	1420	0	3238	0	0.60	57902
154	F3023715	00000059	21122920	21130319	0	44922	1600	1016	0	12254	0	0.60	44922
254	F3701454	00000060	21195717	21201112	0	32811	3136	2844	0	7237	0	0.60	32811

JAERI-M 82-200

Table 5.2 (Cont'd)

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
584	F8221056	00000061	22032045	22033123	0	18146	2112	2052	0	111	0	0.60	18146
625	F1035311	00000062	21192844	21195716	0	60011	1600	1360	0	5104	0	0.60	60011
475	F9220005	00000063	21180838	21181904	0	30928	2624	2552	0	641	0	0.60	30928
TOTAL						49909(SEC)				382246(TIMES)			49909(SEC)

```
LMAX   = 2
P      = 0.194 0.493 0.314
V      = 0.600 0.800 0.950
SCALAR= 1.0
VECTOR= 1.0
IOTIME= 0.03
CPULT =  120.(SEC)
MIN JOB NO. =     3
MAX JOB NO. =   676
TOTAL USER   CPU      49909(SEC)
TOTAL VECTOR CPU      49909(SEC)
TOTAL I/O  TIME       11467(SEC)
TOTAL  JOBS              63
```

L	FREQ (THEORY)	PROB	FREQ (ACTUAL)
0	12.2	0.194	10
1	31.1	0.493	34
2	19.8	0.314	19

KHI-SQUARE= 0.7134

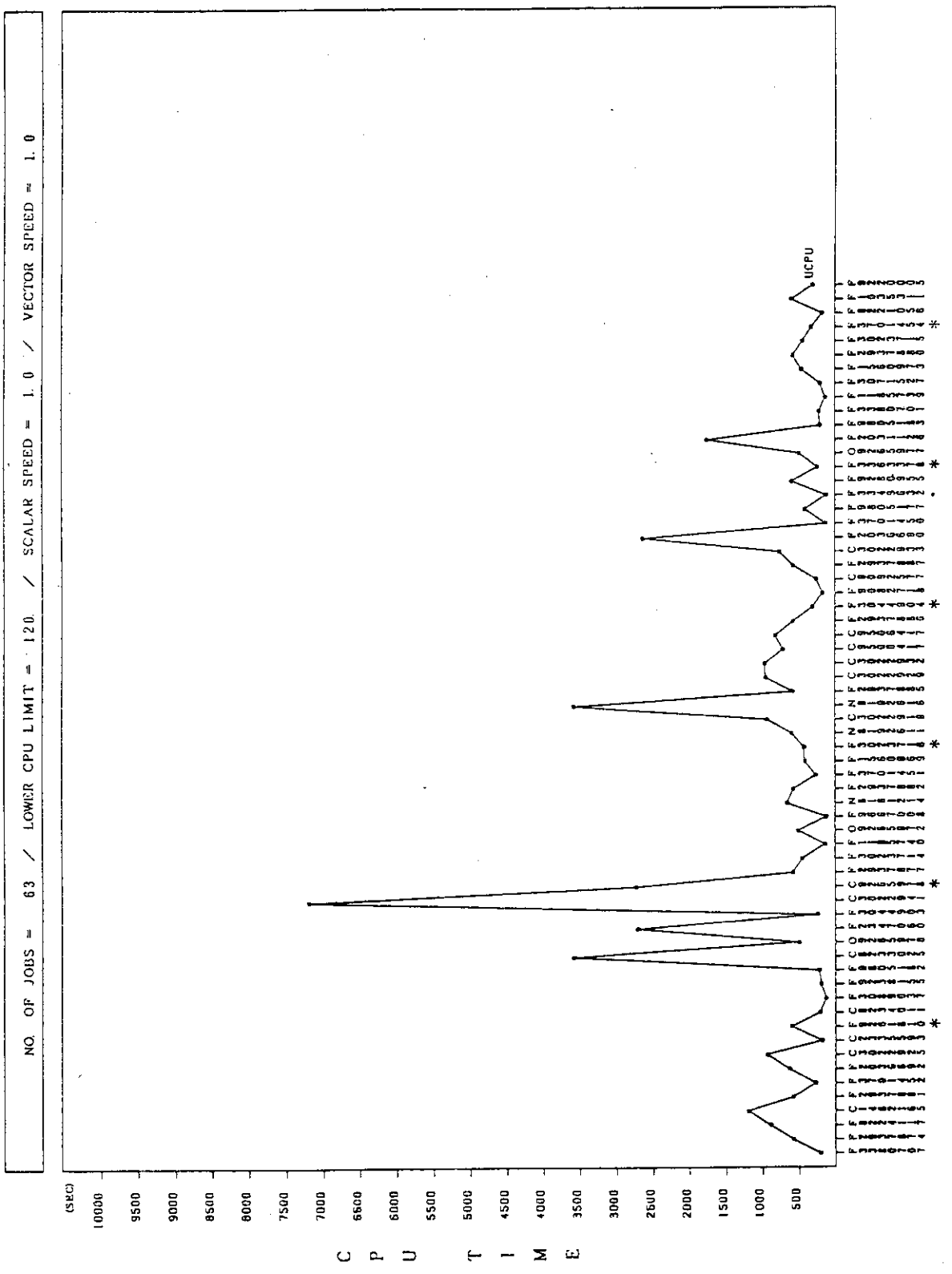


Fig. 5.3 Graphical representation of job sequence of Table 5.2.

Table 5.3 Job pattern for simulation of A computer system(scalar=1.0, vector=5.0).

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SID	UID	LL	V	VCPU
164	F3380707	00000001	21122130	21123434	0	20851	1600	1192	0	177	2	0.95	5004
4	F2937874	00000002	21091013	21093338	0	57807	1600	1420	0	3238	2	0.95	13873
337	F8224117	00000003	22065456	22075522	0	88758	2112	2056	0	754	2	0.95	21301
514	C1482165	00000004	21184050	21194341	0	120006	832	724	0	16442	2	0.95	28801
201	F2937881	00000005	21123932	21131251	0	57899	1600	1404	0	3235	2	0.95	13895
22	F3701452	00000006	21101059	21102603	0	27482	3136	2844	0	6063	2	0.95	6595
596	F2036692	00000007	22012544	22020504	0	63177	1088	1036	0	324	2	0.95	15162
256	C3022925	00000008	21135224	21143338	0	93987	832	832	0	670	2	0.95	22556
120	C2335593	00000009	21105729	21111229	0	18506	1600	1536	0	8778	2	0.95	4441
466	F9261810	00000010	21221536	21225531	0	60008	2112	1700	0	2889	2	0.95	14401
675	C8234011	00000011	22011000	22014059	0	21329	1088	592	0	24410	2	0.95	5118
629	F3089037	00000012	22071754	22074039	0	13296	1600	660	0	31834	2	0.95	3191
660	F9238155	00000013	22062909	22064343	0	19997	2112	832	0	2332	2	0.95	4799
522	F9805182	00000014	21184312	21184840	0	22699	2112	1612	0	115	2	0.95	5447
673	C8233025	00000015	21221940	22010034	0	358360	1600	1448	0	9019	2	0.95	86006
276	09265976	00000016	21144750	21151526	0	49614	2112	1900	0	957	2	0.95	11907
527	F2347060	00000017	22024901	22055021	0	271029	4160	3556	0	31157	2	0.95	65046
226	F3044903	00000018	21131254	21132740	0	24422	1600	1536	0	5054	2	0.95	5861
674	C3022941	00000019	22002422	22064634	0	719581	832	832	0	743	2	0.95	172699
421	C9265978	00000020	21221814	22002102	0	273127	2112	1936	0	2872	1	0.80	98325
7	F2937877	00000021	21092952	21095644	0	58196	1600	1420	0	3439	1	0.80	20950
150	F3023714	00000022	21115516	21121914	0	45367	1600	1016	0	12438	1	0.80	16332
220	F1185740	00000023	22010053	22012542	0	14071	1088	832	0	29398	1	0.80	5065
214	09265972	00000024	21130322	21132229	0	50712	2112	1900	0	954	1	0.80	18256
56	F9667009	00000025	21105406	21111415	0	12163	1600	1548	0	4206	1	0.80	4378
193	N8181214	00000026	22055023	22062906	0	66318	1600	1308	0	1313	1	0.80	23874
242	F2937882	00000027	21132234	21135155	0	57863	1600	1404	0	3230	1	0.80	20830
5	F3701451	00000028	21091436	21092948	0	26966	3136	2844	0	6539	1	0.80	9707
23	F1560969	00000029	21094518	21101043	0	41874	2112	1904	0	10262	1	0.80	15074
156	F3023716	00000030	21194343	21200547	0	43115	1600	1016	0	11841	1	0.80	15521
151	N8192611	00000031	21143734	21151212	0	60007	1600	1492	0	10607	1	0.80	21602
127	C3022919	00000032	21111232	21113953	0	93554	832	832	0	665	1	0.80	33679
558	N8192616	00000033	22051737	22073240	0	359240	1600	1492	0	13622	1	0.80	129326
370	F2937885	00000034	21151217	21154140	0	57894	1600	1396	0	3235	1	0.80	20841
401	C3022929	00000035	21155618	21162643	0	95584	832	832	0	667	1	0.80	34410
518	C3022932	00000036	21174213	21180835	0	96742	832	832	0	668	1	0.80	34827
425	C9506417	00000037	21162645	21164956	0	71557	832	832	0	479	1	0.80	25760
426	C9506417	00000038	21165431	21172106	0	82185	832	832	0	479	1	0.80	29586
443	F2937886	00000039	21172307	21174211	0	57854	1600	1404	0	3236	1	0.80	20827
227	F3044904	00000040	21132847	21135216	0	31049	1600	1536	0	5045	1	0.80	11177
626	F9082718	00000041	22073338	22074157	0	17512	1600	1560	0	38	1	0.80	6304
676	C9092577	00000042	22083601	22092132	0	25920	1600	1536	0	5577	1	0.80	9331
568	F2937887	00000043	21184845	21191643	0	57825	1600	1420	0	3198	1	0.80	20817
520	C3022933	00000044	21181909	21184309	0	76394	832	832	0	662	1	0.80	27501
595	F2036690	00000045	22005013	22031902	0	264214	1088	1036	0	326	1	0.80	95117
3	F3701450	00000046	21090331	21091009	0	13030	3136	2844	0	3414	1	0.80	4690
173	F9805177	00000047	21121921	21123928	0	41790	2112	1604	0	95	1	0.80	15044
541	F3349632	00000048	21181207	21182923	0	12369	1600	1196	0	3005	1	0.80	4452
251	F9280955	00000049	21140321	21143411	0	60005	1600	1376	0	1661	1	0.80	21601
644	F3363378	00000050	22020507	22022009	0	24548	2624	2580	0	67	1	0.80	8837
284	09265977	00000051	21154149	21155558	0	49578	2112	1900	0	957	1	0.80	17848
602	F2031126	00000052	22033127	22051143	0	176600	4160	3660	0	19243	1	0.80	63576
525	F9805183	00000053	21191648	21192633	0	21404	2112	1612	0	116	1	0.80	7705
19	F3380701	00000054	21093341	21094515	0	22140	1600	1192	0	177	0	0.60	11512
219	F1185739	00000055	22002517	22004821	0	14050	1088	832	0	29397	0	0.60	7306
176	F3071527	00000056	21124327	21130458	0	20936	832	768	0	9987	0	0.60	10886
240	F1560973	00000057	22022011	22024856	0	46507	3136	2692	0	2855	0	0.60	24183
142	F2937880	00000058	21112235	21114903	0	57902	1600	1420	0	3238	0	0.60	30109
154	F3023715	00000059	21122920	21130319	0	44922	1600	1016	0	12254	0	0.60	23359
254	F3701454	00000060	21195717	21201112	0	32811	3136	2844	0	7237	0	0.60	17061

J9175.DAY0121A.
DATA
SCALAR=1.0
VECTOR=5.0
ﾃﾞｰﾀ).

Table 5.3 (Cont'd)

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	V.CPU
584	F8221056	00000061	22032045	22033123	0	18146	2112	2052	0	111	0	0.60	9435
625	F1035311	00000062	21192844	21195716	0	60011	1600	1360	0	5104	0	0.60	31205
475	F9220005	00000063	21180838	21181904	0	30928	2624	2552	0	641	0	0.60	16082
TOTAL						49909(SEC)				382246(TIMES)			15977(SEC)

```
LMAX  = 2
P     = 0.194 0.493 0.314
V     = 0.600 0.800 0.950
SCALAR= 1.0
VECTOR= 5.0
IOTIME= 0.03
CPULT =  120.(SEC)
MIN JOB NO. =     3
MAX JOB NO. =   676
TOTAL USER   CPU      49909(SEC)
TOTAL VECTOR CPU      15977(SEC)
TOTAL I/O  TIME       11467(SEC)
TOTAL  JOBS              63
```

L	FREQ (THEORY)	PROB	FREQ (ACTUAL)
0	12.2	0.194	10
1	31.1	0.493	34
2	19.8	0.314	19

KHI-SQUARE= 0.7134

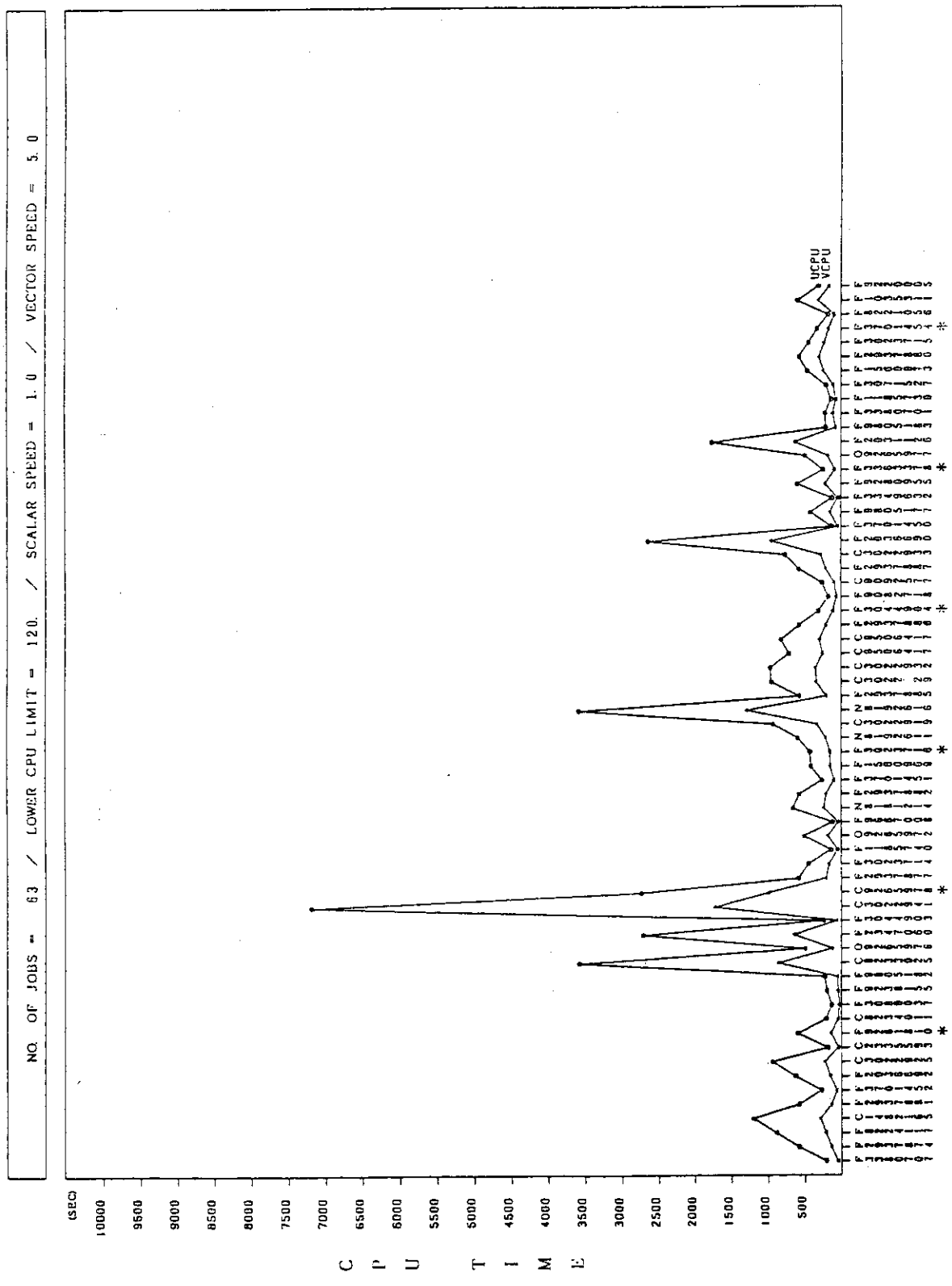


Fig. 5.4 Graphical representation of job sequence of Table 5.3.

Table 5.4 Job pattern for simulation of A computer system(scalar=1.0, vector=10.0).

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
164	F3380707	00000001	21122130	21123434	0	20851	1600	1192	0	177	2	0.95	3023
4	F2937874	00000002	21091013	21093338	0	57807	1600	1420	0	3238	2	0.95	8382
337	F8224117	00000003	22065456	22075522	0	88758	2112	2056	0	754	2	0.95	12869
514	C1482165	00000004	21184050	21194341	0	120006	832	724	0	16442	2	0.95	17400
201	F2937881	00000005	21123932	21131251	0	57899	1600	1404	0	3235	2	0.95	8395
22	F3701452	00000006	21101059	21102603	0	27482	3136	2844	0	6063	2	0.95	3984
596	F2036692	00000007	22012544	22020504	0	63177	1088	1036	0	324	2	0.95	9160
256	C3022925	00000008	21135224	21143338	0	93987	832	832	0	670	2	0.95	13628
120	C2335593	00000009	21105729	21111229	0	18506	1600	1536	0	8778	2	0.95	2683
466	F9261810	00000010	21221536	21225531	0	60008	2112	1700	0	2889	2	0.95	8701
675	C8234011	00000011	22011000	22014059	0	21329	1088	592	0	24410	2	0.95	3092
629	F3089037	00000012	22071754	22074039	0	13296	1600	660	0	31834	2	0.95	1927
660	F9238155	00000013	22062909	22064343	0	19997	2112	832	0	2332	2	0.95	2899
522	F9805182	00000014	21184312	21184840	0	22699	2112	1612	0	115	2	0.95	3291
673	C8233025	00000015	21221940	22010034	0	358360	1600	1448	0	9019	2	0.95	51962
276	09265976	00000016	21144750	21151526	0	49614	2112	1900	0	957	2	0.95	7194
527	F2347060	00000017	22024901	22055021	0	271029	4160	3556	0	31157	2	0.95	39299
226	F3044903	00000018	21131254	21132740	0	24422	1600	1536	0	5054	2	0.95	3541
674	C3022941	00000019	22002422	22064634	0	719581	832	832	0	743	2	0.95	104339
421	C9265978	00000020	21221814	22002102	0	273127	2112	1936	0	2372	1	0.80	76475
7	F2937877	00000021	21092952	21095644	0	58196	1600	1420	0	3439	1	0.80	16294
150	F3023714	00000022	21115516	21121914	0	45367	1600	1016	0	12438	1	0.80	12702
220	F1185740	00000023	22010053	22012542	0	14071	1088	832	0	29398	1	0.80	3939
214	09265972	00000024	21130322	21132229	0	50712	2112	1900	0	954	1	0.80	14199
56	F9667008	00000025	21105406	21111415	0	12163	1600	1548	0	4206	1	0.80	3405
193	N8181214	00000026	22055023	22062906	0	66318	1600	1308	0	1313	1	0.80	18569
242	F2937882	00000027	21132234	21135155	0	57863	1600	1404	0	3230	1	0.80	16201
5	F3701451	00000028	21091436	21092948	0	26966	3136	2844	0	6539	1	0.80	7550
23	F1560969	00000029	21094518	21101043	0	41874	2112	1904	0	10262	1	0.80	11724
156	F3023716	00000030	21194343	21200547	0	43115	1600	1016	0	11841	1	0.80	12072
151	N8192611	00000031	21143734	21151212	0	60007	1600	1492	0	10607	1	0.80	16801
127	C3022919	00000032	21111232	21113953	0	93554	832	832	0	665	1	0.80	26195
558	N8192616	00000033	22051737	22073240	0	359240	1600	1492	0	13622	1	0.80	100587
370	F2937885	00000034	21151217	21154140	0	57894	1600	1396	0	3235	1	0.80	16210
401	C3022929	00000035	21155618	21162643	0	95584	832	832	0	667	1	0.80	26763
518	C3022932	00000036	21174213	21180835	0	96742	832	832	0	668	1	0.80	27087
425	C9506417	00000037	21162645	21164956	0	71557	832	832	0	479	1	0.80	20035
426	C9506417	00000038	21165431	21172106	0	82185	832	832	0	479	1	0.80	23011
443	F2937886	00000039	21172307	21174211	0	57854	1600	1404	0	3236	1	0.80	16199
227	F3044904	00000040	21132847	21135216	0	31049	1600	1536	0	5045	1	0.80	8693
626	F9082718	00000041	22073338	22074157	0	17512	1600	1560	0	38	1	0.80	4903
676	C9092577	00000042	22083601	22092132	0	25920	1600	1536	0	5577	1	0.80	7257
568	F2937887	00000043	21184845	21191643	0	57825	1600	1420	0	3198	1	0.80	16191
520	C3022933	00000044	21181909	21184309	0	76394	832	832	0	662	1	0.80	21390
595	F2036690	00000045	22005013	22031902	0	264214	1088	1036	0	326	1	0.80	73979
3	F3701450	00000046	21090331	21091009	0	13030	3136	2844	0	3414	1	0.80	3648
173	F9805177	00000047	21121921	21123928	0	41790	2112	1604	0	95	1	0.80	11701
541	F3349632	00000048	21181207	21182923	0	12369	1600	1196	0	3005	1	0.80	3463
251	F9280955	00000049	21140321	21143411	0	60005	1600	1376	0	1661	1	0.80	16801
644	F3363378	00000050	22020507	22022009	0	24548	2624	2580	0	67	1	0.80	6873
284	09265977	00000051	21154149	21155558	0	49578	2112	1900	0	957	1	0.80	13881
602	F2031126	00000052	22033127	22051143	0	176600	4160	3660	0	19243	1	0.80	49448
525	F9805183	00000053	21191648	21192633	0	21404	2112	1612	0	116	1	0.80	5993
19	F3380701	00000054	21093341	21094515	0	22140	1600	1192	0	177	0	0.60	10184
219	F1185739	00000055	22002517	22004821	0	14050	1088	832	0	29397	0	0.60	6463
176	F3071527	00000056	21124327	21130458	0	20936	832	768	0	9987	0	0.60	9630
240	F1560973	00000057	22022011	22024856	0	46507	3136	2692	0	2855	0	0.60	21393
142	F2937880	00000058	21112235	21114903	0	57902	1600	1420	0	3238	0	0.60	26634
154	F3023715	00000059	21122920	21130319	0	44922	1600	1016	0	12254	0	0.60	20664
254	F3701454	00000060	21195717	21201112	0	32811	3136	2844	0	7237	0	0.60	15093

JWK.DAY01210.DATA
SCALAR=1.0
VECTOR=10.0
ソート済み

Table 5.4 (Cont'd)

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
584	F8221056	00000061	22032045	22033123	0	18146	2112	2052	0	111	0	0.60	8347
625	F1035311	00000062	21192844	21195716	0	60011	1600	1360	0	5104	0	0.60	27605
475	F9220005	00000063	21180838	21181904	0	30928	2624	2552	0	641	0	0.60	14226
TOTAL						49909(SEC)				382246(TIMES)			11727(SEC)

```
LMAX   = 2
P      = 0.194 0.493 0.314
V      = 0.600 0.800 0.950
SCALAR= 1.0
VECTOR=10.0
IOTIME= 0.03
CPULT =  120.(SEC)
MIN JOB NO. =     3
MAX JOB NO. =   676
TOTAL USER   CPU      49909(SEC)
TOTAL VECTOR CPU      11727(SEC)
TOTAL I/O  TIME       11467(SEC)
TOTAL  JOBS              63
```

L	FREQ (THEORY)	PROB	FREQ (ACTUAL)
0	12.2	0.194	10
1	31.1	0.493	34
2	19.8	0.314	19

KHI-SQUARE= 0.7134

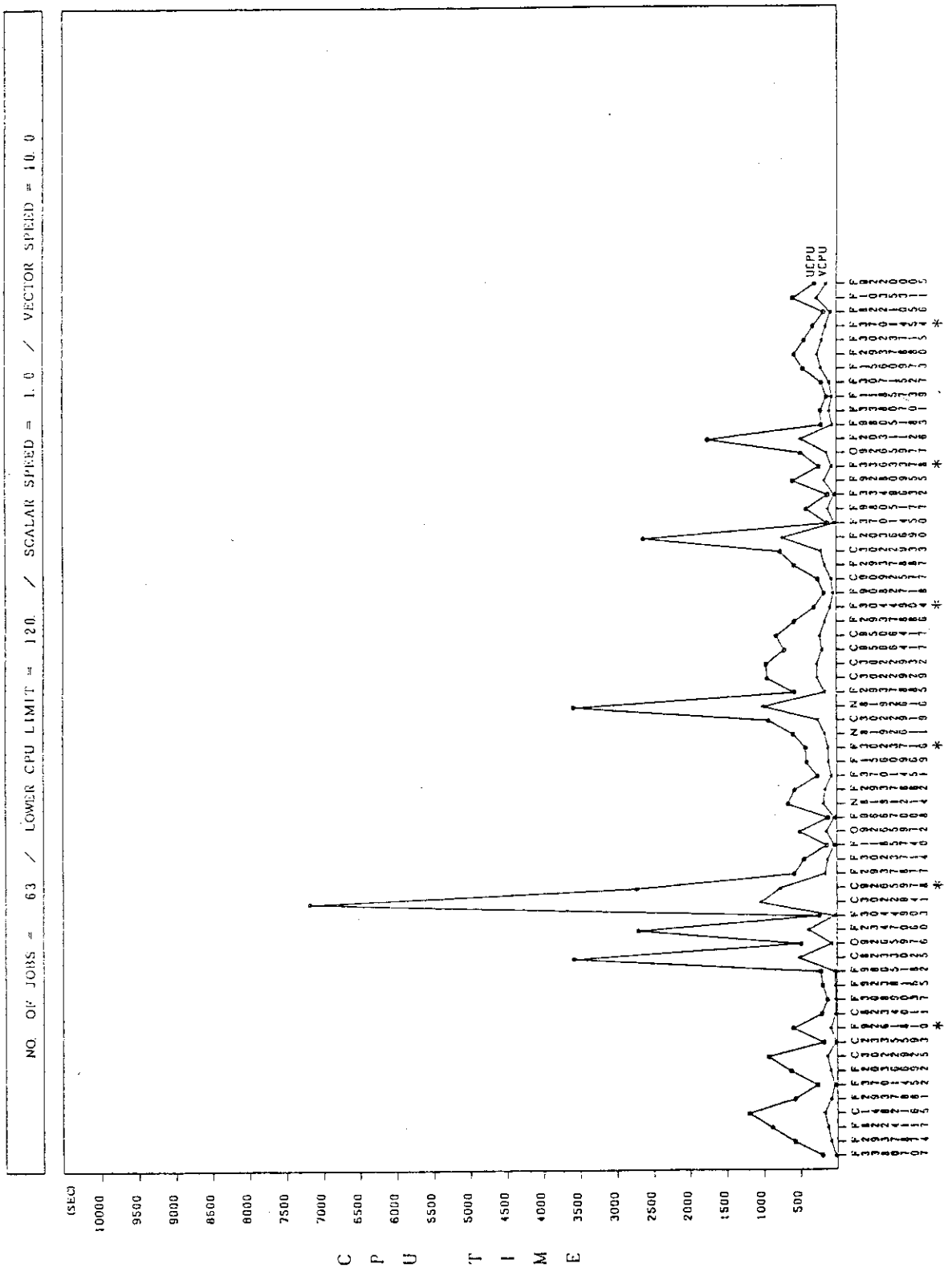


Fig. 5.5 Graphical representation of job sequence of Table 5.4.

Table 5.5 Job pattern for simulation of C computer system(scalar=1.0, vector=1.0).

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
303	F9024661	00000001	21121641	21123937	0	60005	1088	1036	0	327	2	0.95	60005
40	F9239145	00000002	21093906	21095647	0	57118	1088	912	0	3696	2	0.95	57118
830	F3197240	00000003	22054329	22080522	0	359881	1600	1448	0	6467	2	0.95	359881
574	02967210	00000004	21151030	21153226	0	60004	576	576	0	178	2	0.95	60004
846	F9262999	00000005	21184645	21185837	0	41165	576	568	0	351	2	0.95	41165
192	F3051236	00000006	21105825	21110611	0	33566	832	588	0	175	2	0.95	33566
941	C1026473	00000007	21221611	22014618	0	403645	2112	1260	0	36592	2	0.95	403645
431	F3051238	00000008	21133329	21140904	0	60005	832	588	0	4	2	0.95	60005
221	F9024657	00000009	21113229	21114856	0	39063	1088	1036	0	325	2	0.95	39063
944	C2347061	00000010	21223219	22000411	0	128881	1600	1008	0	34499	2	0.95	128881
466	F9024665	00000011	21140719	21142009	0	42766	1088	1036	0	325	2	0.95	42766
778	F3051245	00000012	22010359	22030111	0	180005	832	588	0	4	2	0.95	180005
780	F2003878	00000013	22051838	22060442	0	78843	2112	1260	0	11554	2	0.95	78843
637	F1084450	00000014	21165652	21172822	0	42500	1088	728	0	17369	2	0.95	42500
741	N8236037	00000015	22031510	22060727	0	325228	1600	1448	0	8113	2	0.95	325228
435	F3051239	00000016	21133447	21141011	0	60005	832	588	0	4	2	0.95	60005
667	C9280958	00000017	21191257	21194438	0	63253	1600	1376	0	1661	2	0.95	63253
546	F9024669	00000018	21145316	21151025	0	60005	832	832	0	650	2	0.95	60005
786	F9280961	00000019	22053239	22061625	0	76198	2112	1804	0	1632	2	0.95	76198
336	N8182194	00000020	21225001	21232149	0	70550	1600	1308	0	1313	2	0.95	70550
344	N8182196	00000021	22061749	22065231	0	65743	1600	1308	0	1313	2	0.95	65743
807	F9024673	00000022	22013408	22033049	0	186234	1088	1036	0	325	2	0.95	186234
788	F9280962	00000023	22060044	22064434	0	76629	2112	1804	0	1625	2	0.95	76629
339	N8182195	00000024	22050057	22053236	0	68982	1600	1308	0	1313	1	0.80	68982
337	N8183327	00000025	21235918	22002951	0	69312	1600	1308	0	1313	1	0.80	69312
316	F1560972	00000026	21131002	21135120	0	45300	1088	768	0	18053	1	0.80	45300
377	F9025699	00000027	22065406	22080715	0	256873	4160	2772	0	950	1	0.80	256873
256	F3023717	00000028	21221325	21223847	0	47088	1600	1016	0	12197	1	0.80	47088
219	F9024656	00000029	21112734	21114831	0	53017	1088	1036	0	326	1	0.80	53017
302	F1560971	00000030	21130017	21133629	0	58684	1088	764	0	10820	1	0.80	58684
283	F3051237	00000031	21115420	21121635	0	60005	832	588	0	4	1	0.80	60005
128	F1560970	00000032	21111149	21115021	0	56571	1088	760	0	15761	1	0.80	56571
674	F9024671	00000033	21164407	21171358	0	60004	1088	1036	0	326	1	0.80	60004
470	F9024666	00000034	21142040	21143737	0	60005	1088	1036	0	324	1	0.80	60005
474	F9239149	00000035	21150707	21153226	0	57127	1088	872	0	1744	1	0.80	57127
477	F9282187	00000036	21142723	21144426	0	43760	1088	976	0	6063	1	0.80	43760
514	F1560974	00000037	21154537	21161744	0	57520	1088	760	0	15735	1	0.80	57520
536	F9024667	00000038	21143505	21144838	0	52415	832	832	0	685	1	0.80	52415
541	F9024668	00000039	21145003	21150657	0	60005	832	832	0	684	1	0.80	60005
350	F3022923	00000040	21123741	21124954	0	30005	832	832	0	454	1	0.80	30005
943	C3491238	00000041	21221840	22042950	0	719979	2624	1524	0	2958	1	0.80	719979
947	C3247616	00000042	21234441	22013835	0	130512	1600	1012	0	35056	1	0.80	130512
579	F9024670	00000043	21153848	21155350	0	41846	1088	1036	0	326	1	0.80	41846
619	F3340506	00000044	21161758	21163856	0	58385	1088	840	0	1005	1	0.80	58385
728	F9907533	00000045	21181201	21183314	0	41479	1600	1120	0	14304	1	0.80	41479
24	F1560967	00000046	21092007	21094903	0	56684	1088	760	0	15748	1	0.80	56684
308	F9805178	00000047	21130944	21132518	0	30006	2112	1608	0	116	1	0.80	30006
672	F3051241	00000048	21163836	21171412	0	59034	832	588	0	175	1	0.80	59034
777	F3051244	00000049	22002954	22020908	0	159199	832	588	0	347	1	0.80	159199
677	F3051243	00000050	21164534	21171438	0	59186	832	588	0	175	1	0.80	59186
99	F3119060	00000051	21101227	21102240	0	37212	320	276	0	419	1	0.80	37212
394	F9907532	00000052	21130812	21140029	0	60005	1600	1120	0	21763	1	0.80	60005
841	F3590800	00000053	22034229	22065020	0	355790	1600	1308	0	3079	1	0.80	355790
766	F2370933	00000054	22002907	22013405	0	91515	1600	1312	0	8073	1	0.80	91515
199	F3119061	00000055	21110613	21111351	0	37093	320	276	0	419	1	0.80	37093
808	F9024674	00000056	22050158	22074449	0	360004	1088	1036	0	327	1	0.80	360004
942	C0269103	00000057	21221612	22011728	0	385269	2112	1260	0	19419	1	0.80	385269
779	F3051246	00000058	22061847	22073707	0	179755	832	588	0	347	1	0.80	179755
226	F1035302	00000059	21115640	21123543	0	56035	1088	892	0	16290	1	0.80	56035
877	F1084451	00000060	21191225	21194344	0	42769	1088	728	0	17463	1	0.80	42769

システムC 300秒以上
SCALAR=1.0
VECTOR=1.0
39875.DAT 0121様.DATA

Table 5.5 (Cont'd)

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
652	F9239150	00000061	21171451	21173442	0	57120	1088	912	0	3078	1	0.80	57120
440	F3051240	00000062	21133714	21141137	0	60006	832	588	0	4	1	0.80	60006
811	N8215388	00000063	21175205	21182623	0	30007	1600	1448	0	5649	1	0.80	30007
338	N8181213	00000064	22005008	22013919	0	71893	1600	1308	0	1313	0	0.60	71893
375	F9025698	00000065	22020946	22050155	0	301571	4160	2788	0	1069	0	0.60	301571
312	F9024662	00000066	21124438	21130611	0	58557	1088	1036	0	324	0	0.60	58557
415	F3340502	00000067	21132514	21140206	0	57978	1088	840	0	1003	0	0.60	57978
135	F3340500	00000068	21104414	21105823	0	57780	1088	844	0	1001	0	0.60	57780
299	F9024660	00000069	21121328	21123733	0	60005	1088	1036	0	325	0	0.60	60005
772	F2322479	00000070	22011806	22044117	0	328346	1088	832	0	661	0	0.60	328346
809	N8215387	00000071	21175027	21180456	0	30007	1600	1448	0	5670	0	0.60	30007
558	F8223042	00000072	22022452	22032759	0	92129	2112	2052	0	757	0	0.60	92129
724	F3119062	00000073	21172030	21173248	0	37433	320	276	0	419	0	0.60	37433
227	F9024658	00000074	21113744	21115218	0	39189	1088	1036	0	326	0	0.60	39189
949	C0521012	00000075	22015054	22051313	0	328716	2112	1260	0	64008	0	0.60	328716
TOTAL						82773(SEC)				458670(TIMES)			82773(SEC)

```
LMAX  = 2
P     = 0.194 0.493 0.314
V     = 0.600 0.800 0.950
SCALAR= 1.0
VECTOR= 1.0
IOTIME= 0.03
CPULT =  300.(SEC)
MIN JOB NO. =   24
MAX JOB NO. =  949
TOTAL USER   CPU      82773(SEC)
TOTAL VECTOR CPU      82773(SEC)
TOTAL I/O  TIME       13760(SEC)
TOTAL  JOBS              75
```

L	FREQ (THEORY)	PROB	FREQ (ACTUAL)
0	14.6	0.194	12
1	37.0	0.493	40
2	23.5	0.314	23

KHI-SQUARE= 0.7072

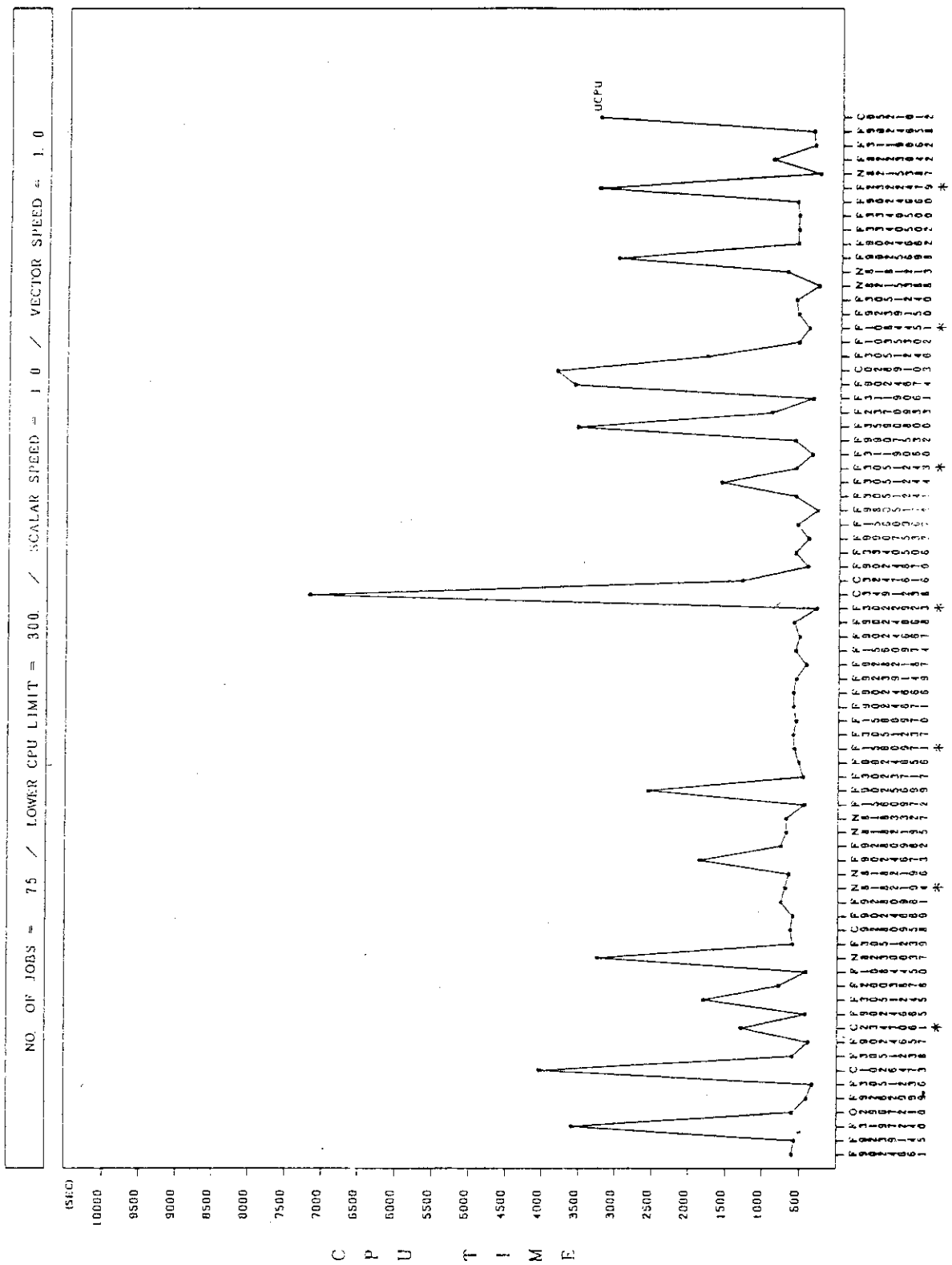


Fig. 5.6 Graphical representation of job sequence of Table 5.5.

Table 5.6 Job pattern for simulation of C computer system(scalar=1.0, vector=5.0).

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
303	F9024661	00000001	21121641	21123937	0	60005	1088	1036	0	327	2	0.95	14401
40	F9239145	00000002	21093906	21095647	0	57118	1088	912	0	3696	2	0.95	13708
830	F3197240	00000003	22054329	22080522	0	359881	1600	1448	0	6467	2	0.95	86371
574	02967210	00000004	21151030	21153226	0	60004	576	576	0	178	2	0.95	14400
846	F9262999	00000005	21184645	21185837	0	41165	576	568	0	351	2	0.95	9879
192	F3051236	00000006	21105825	21110611	0	33566	832	588	0	175	2	0.95	8055
941	C1026473	00000007	21221611	22014618	0	403645	2112	1260	0	36592	2	0.95	96874
431	F3051238	00000008	21133329	21140904	0	60005	832	588	0	4	2	0.95	14401
221	F9024657	00000009	21113229	21114856	0	39063	1088	1036	0	325	2	0.95	9375
944	C2347061	00000010	21223219	22000411	0	128881	1600	1008	0	34499	2	0.95	30931
466	F9024665	00000011	21140719	21142009	0	42766	1088	1036	0	325	2	0.95	10263
778	F3051245	00000012	22010359	22030111	0	180005	832	588	0	4	2	0.95	43201
780	F2003878	00000013	22051838	22060442	0	78843	2112	1260	0	11554	2	0.95	18922
637	F1084450	00000014	21165652	21172822	0	42500	1088	728	0	17369	2	0.95	10200
741	N8236037	00000015	22031510	22060727	0	325228	1600	1448	0	8113	2	0.95	78054
435	F3051239	00000016	21133447	21141011	0	60005	832	588	0	4	2	0.95	14401
667	C9280958	00000017	21191257	21194438	0	63253	1600	1376	0	1661	2	0.95	15180
546	F9024669	00000018	21145316	21151025	0	60005	832	832	0	650	2	0.95	14401
786	F9280961	00000019	22053239	22061625	0	76198	2112	1804	0	1632	2	0.95	18287
336	N8182194	00000020	21225001	21232149	0	70550	1600	1308	0	1313	2	0.95	16932
344	N8182196	00000021	22061749	22065231	0	65743	1600	1308	0	1313	2	0.95	15778
807	F9024673	00000022	22013408	22033049	0	186234	1088	1036	0	325	2	0.95	44696
788	F9280962	00000023	22060044	22064434	0	76629	2112	1804	0	1625	2	0.95	18390
339	N8182195	00000024	22050057	22053236	0	68982	1600	1308	0	1313	1	0.80	24833
337	N8183327	00000025	21235918	22002951	0	69312	1600	1308	0	1313	1	0.80	24952
316	F1560972	00000026	21131002	21135120	0	45300	1088	768	0	18053	1	0.80	16308
377	F9025699	00000027	22065406	22080715	0	256873	4160	2772	0	950	1	0.80	92474
256	F3023717	00000028	21221325	21223847	0	47088	1600	1016	0	12197	1	0.80	16951
219	F9024656	00000029	21112734	21114831	0	53017	1088	1036	0	326	1	0.80	19086
302	F1560971	00000030	21130017	21133629	0	58684	1088	764	0	10820	1	0.80	21126
283	F3051237	00000031	21115420	21121635	0	60005	832	588	0	4	1	0.80	21601
128	F1560970	00000032	21111149	21115021	0	56571	1088	760	0	15761	1	0.80	20365
674	F9024671	00000033	21164407	21171358	0	60004	1088	1036	0	326	1	0.80	21601
470	F9024666	00000034	21142040	21143737	0	60005	1088	1036	0	324	1	0.80	21601
474	F9239149	00000035	21150707	21153226	0	57127	1088	872	0	1744	1	0.80	20565
477	F9282187	00000036	21142723	21144426	0	43760	1088	976	0	6063	1	0.80	15753
514	F1560974	00000037	21154537	21161744	0	57520	1088	760	0	15735	1	0.80	20707
536	F9024667	00000038	21143505	21144838	0	52415	832	832	0	685	1	0.80	18869
541	F9024668	00000039	21145003	21150657	0	60005	832	832	0	684	1	0.80	21601
350	F3022923	00000040	21123741	21124954	0	30005	832	832	0	454	1	0.80	10801
943	C3491238	00000041	21221840	22042950	0	719979	2624	1524	0	2958	1	0.80	259192
947	C3247616	00000042	21234441	22013835	0	130512	1600	1012	0	35056	1	0.80	46984
579	F9024670	00000043	21153848	21155350	0	41846	1088	1036	0	326	1	0.80	15064
619	F3340506	00000044	21161758	21163856	0	58385	1088	840	0	1005	1	0.80	21018
728	F9907533	00000045	21181201	21183314	0	41479	1600	1120	0	14304	1	0.80	14932
24	F1560967	00000046	21092007	21094903	0	56684	1088	760	0	15748	1	0.80	20406
308	F9805178	00000047	21130944	21132518	0	30006	2112	1608	0	116	1	0.80	10802
672	F3051241	00000048	21163836	21171412	0	59034	832	588	0	175	1	0.80	21252
777	F3051244	00000049	22002954	22020908	0	159199	832	588	0	347	1	0.80	57311
677	F3051243	00000050	21164534	21171438	0	59186	832	588	0	175	1	0.80	21306
99	F3119060	00000051	21101227	21102240	0	37212	320	276	0	419	1	0.80	13396
394	F9907532	00000052	21130812	21140029	0	60005	1600	1120	0	21763	1	0.80	21601
841	F3590800	00000053	22034229	22065020	0	355790	1600	1308	0	3079	1	0.80	128084
766	F2370933	00000054	22002907	22013405	0	91515	1600	1312	0	8073	1	0.80	32945
199	F3119061	00000055	21110613	21111351	0	37093	320	276	0	419	1	0.80	13353
808	F9024674	00000056	22050158	22074449	0	360004	1088	1036	0	327	1	0.80	129601
942	C0269103	00000057	21221612	22011728	0	385269	2112	1260	0	19419	1	0.80	138696
779	F3051246	00000058	22061847	22073707	0	179755	832	588	0	347	1	0.80	64711
226	F1055302	00000059	21115640	21123543	0	56035	1088	892	0	16290	1	0.80	20172
877	F1084451	00000060	21191225	21194344	0	42769	1088	728	0	17463	1	0.80	15396

システムC 300秒以上
SCALAR=1.0
VECTOR=5.0
J9175.DAY0121G.DATA

Table 5.6 (Cont'd)

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UID	LL	V	VCPU
652	F9239150	00000061	21171451	21173442	0	57120	1088	912	0	3078	1	0.80	20563
440	F3051240	00000062	21133714	21141137	0	60006	832	588	0	4	1	0.80	21602
811	N8215388	00000063	21175205	21182623	0	30007	1600	1448	0	5649	1	0.80	10802
338	N8181213	00000064	22005008	22013919	0	71893	1600	1308	0	1313	0	0.60	37384
375	F9025698	00000065	22020946	22050155	0	301571	4160	2788	0	1069	0	0.60	156816
312	F9024662	00000066	21124438	21130611	0	58557	1088	1036	0	324	0	0.60	30449
415	F3340502	00000067	21132514	21140206	0	57978	1088	840	0	1003	0	0.60	30148
135	F3340500	00000068	21104414	21105823	0	57780	1088	844	0	1001	0	0.60	30045
299	F9024660	00000069	21121328	21123733	0	60005	1088	1036	0	325	0	0.60	31202
772	F2322479	00000070	22011806	22044117	0	328346	1088	832	0	661	0	0.60	170739
809	N8215387	00000071	21175027	21180456	0	30007	1600	1448	0	5670	0	0.60	15603
558	F8223042	00000072	22022452	22032759	0	92129	2112	2052	0	757	0	0.60	47907
724	F3119062	00000073	21172030	21173248	0	37433	320	276	0	419	0	0.60	19465
227	F9024658	00000074	21113744	21115218	0	39189	1088	1036	0	326	0	0.60	20378
949	C0521012	00000075	22015054	22051313	0	328716	2112	1260	0	64008	0	0.60	170932
TOTAL						82773(SEC)				458670(TIMES)			29035(SEC)

```
LMAX   = 2
P      = 0.194 0.493 0.314
V      = 0.600 0.800 0.950
SCALAR= 1.0
VECTOR= 5.0
IOTIME= 0.03
CPULI =  300.(SEC)
MIN JOB NO. =   24
MAX JOB NO. =  949
TOTAL USER   CPU      82773(SEC)
TOTAL VECTOR CPU      29035(SEC)
TOTAL I/O  TIME       13760(SEC)
TOTAL  JOBS              75
```

L	FREQ (THEORY)	PROB	FREQ (ACTUAL)
0	14.6	0.194	12
1	37.0	0.493	40
2	23.5	0.314	23

```
KHI-SQUARE=  0.7072
```

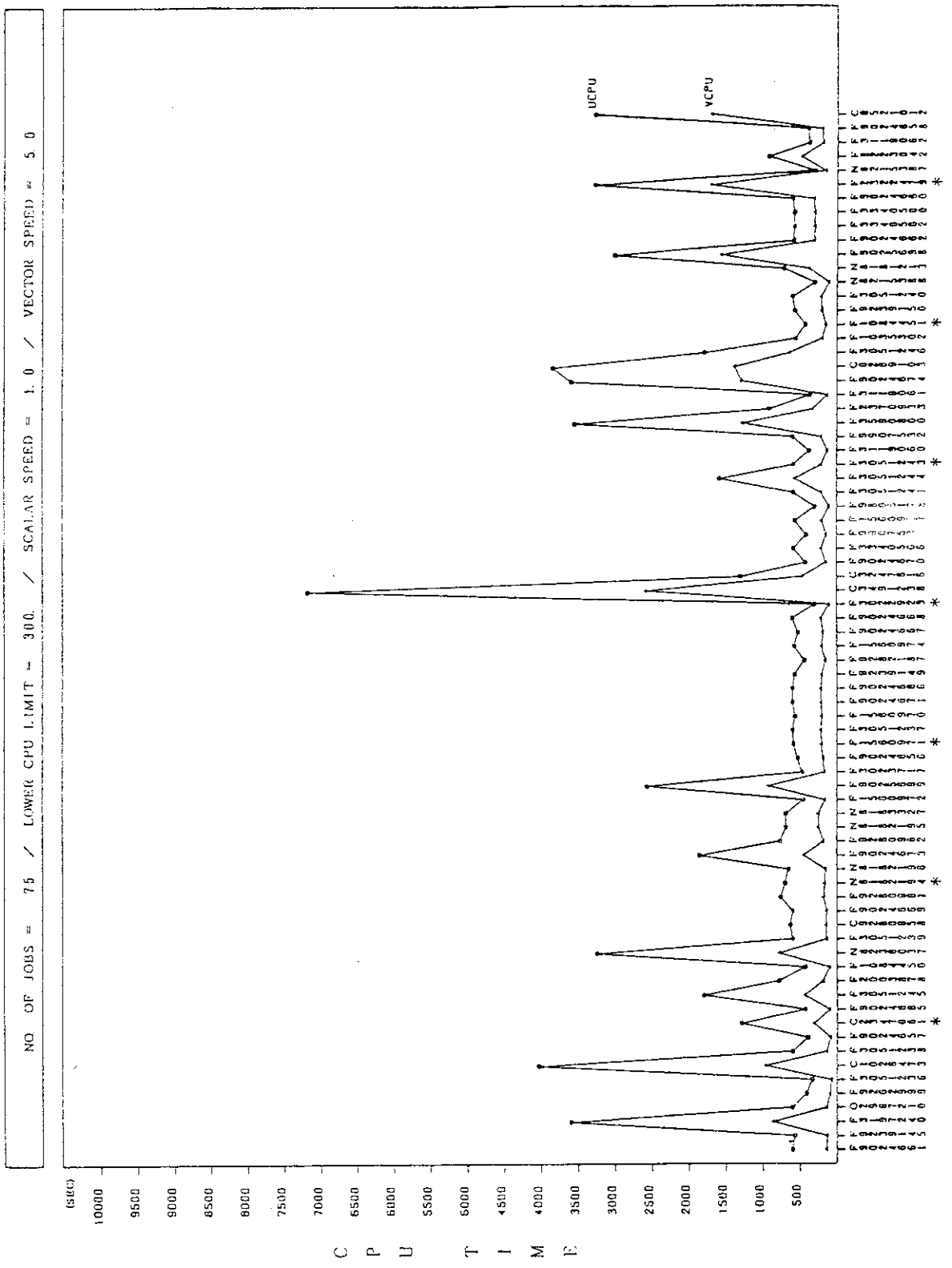


Fig. 5.7 Graphical representation of job sequence of Table 5.6.

Table 5.7 Job pattern for simulation of C computer system(scalar=1.0, vector=10.0).

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
303	F9024661	00000001	21121641	21123937	0	60005	1088	1036	0	327	2	0.95	8700
40	F9239145	00000002	21093906	21095647	0	57118	1088	912	0	3696	2	0.95	8282
830	F3197240	00000003	22054329	22080522	0	359881	1600	1448	0	6467	2	0.95	52182
574	D2967210	00000004	21151030	21153226	0	60004	576	576	0	178	2	0.95	8700
846	F9262999	00000005	21184645	21185837	0	41165	576	568	0	351	2	0.95	5968
192	F3051236	00000006	21105825	21110611	0	33566	832	588	0	175	2	0.95	4867
941	C1026473	00000007	21221611	22014618	0	403645	2112	1260	0	36592	2	0.95	58528
431	F3051238	00000008	21133329	21140904	0	60005	832	588	0	4	2	0.95	8700
221	F9024657	00000009	21113229	21114856	0	39063	1088	1036	0	325	2	0.95	5664
944	C2347061	00000010	21223219	22000411	0	128881	1600	1008	0	34499	2	0.95	18687
466	F9024665	00000011	21140719	21142009	0	42766	1088	1036	0	325	2	0.95	6201
778	F3051245	00000012	22010359	22030111	0	180005	832	588	0	4	2	0.95	26100
780	F2003878	00000013	22051838	22060442	0	78843	2112	1260	0	11554	2	0.95	11432
637	F1084450	00000014	21165652	21172822	0	42500	1088	728	0	17369	2	0.95	6162
741	N8236037	00000015	22031510	22060727	0	325228	1600	1448	0	8113	2	0.95	47158
435	F3051239	00000016	21133447	21141011	0	60005	832	588	0	4	2	0.95	8700
667	C9280958	00000017	21191257	21194438	0	63253	1600	1376	0	1661	2	0.95	9171
546	F9024669	00000018	21145316	21151025	0	60005	832	832	0	650	2	0.95	8700
786	F9280961	00000019	22053239	22061625	0	76198	2112	1804	0	1632	2	0.95	11048
336	N8182194	00000020	21225001	21232149	0	70550	1600	1308	0	1313	2	0.95	10229
344	N8182196	00000021	22061749	22065231	0	65743	1600	1308	0	1313	2	0.95	9532
807	F9024673	00000022	22013408	22033049	0	186234	1088	1036	0	325	2	0.95	27003
788	F9280962	00000023	22060044	22064434	0	76629	2112	1804	0	1625	2	0.95	11111
339	N8182195	00000024	22050057	22053236	0	68982	1600	1308	0	1313	1	0.80	19314
337	N8183327	00000025	21235918	22002951	0	69312	1600	1308	0	1313	1	0.80	19407
316	F1560972	00000026	21131002	21135120	0	45300	1088	768	0	18053	1	0.80	12684
377	F9025699	00000027	22065406	22080715	0	256873	4160	2772	0	950	1	0.80	71924
256	F3023717	00000028	21221325	21223847	0	47088	1600	1016	0	12197	1	0.80	13184
219	F9024656	00000029	21112734	21114831	0	53017	1088	1036	0	326	1	0.80	14844
302	F1560971	00000030	21130017	21133629	0	58684	1088	764	0	10820	1	0.80	16431
283	F3051237	00000031	21115420	21121635	0	60005	832	588	0	4	1	0.80	16801
128	F1560970	00000032	21111149	21115021	0	56571	1088	760	0	15761	1	0.80	15839
674	F9024671	00000033	21164407	21171358	0	60004	1088	1036	0	326	1	0.80	16801
470	F9024666	00000034	21142040	21143737	0	60005	1088	1036	0	324	1	0.80	16801
474	F9239149	00000035	21150707	21153226	0	57127	1088	872	0	1744	1	0.80	15995
477	F9282187	00000036	21142723	21144426	0	43760	1088	976	0	6063	1	0.80	12252
514	F1560974	00000037	21154537	21161744	0	57520	1088	760	0	15735	1	0.80	16105
536	F9024667	00000038	21143505	21144838	0	52415	832	832	0	685	1	0.80	14676
541	F9024668	00000039	21145003	21150657	0	60005	832	832	0	684	1	0.80	16801
350	F3022923	00000040	21123741	21124954	0	30005	832	832	0	454	1	0.80	8401
943	C3491238	00000041	21221840	22042950	0	719979	2624	1524	0	2958	1	0.80	201594
947	C3247616	00000042	21234441	22013835	0	130512	1600	1012	0	35056	1	0.80	36543
579	F9024670	00000043	21153848	21155350	0	41846	1088	1036	0	326	1	0.80	11716
619	F3340506	00000044	21161758	21163856	0	58385	1088	840	0	1005	1	0.80	16347
728	F9907533	00000045	21181201	21183314	0	41479	1600	1120	0	14304	1	0.80	11614
24	F1560967	00000046	21092007	21094903	0	56684	1088	760	0	15748	1	0.80	15871
308	F9805178	00000047	21130944	21132518	0	30006	2112	1608	0	116	1	0.80	8401
672	F3051241	00000048	21163836	21171412	0	59034	832	588	0	175	1	0.80	16529
777	F3051244	00000049	22002954	22020908	0	159199	832	588	0	347	1	0.80	44575
677	F3051243	00000050	21164534	21171438	0	59186	832	588	0	175	1	0.80	16572
99	F3119060	00000051	21101227	21102240	0	37212	320	276	0	419	1	0.80	10419
394	F9907532	00000052	21130812	21140029	0	60005	1600	1120	0	21763	1	0.80	16801
841	F3590800	00000053	22034229	22065020	0	355790	1600	1308	0	3079	1	0.80	99621
766	F2370933	00000054	22002907	22013405	0	91515	1600	1312	0	8073	1	0.80	25624
199	F3119061	00000055	21110613	21111351	0	37093	320	276	0	419	1	0.80	10386
808	F9024674	00000056	22050158	22074449	0	360004	1088	1036	0	327	1	0.80	100801
942	C0269103	00000057	21221612	22011728	0	385269	2112	1260	0	19419	1	0.80	107875
779	F3051246	00000058	22061847	22073707	0	179755	832	588	0	347	1	0.80	50331
226	F1035302	00000059	21115640	21123543	0	56035	1088	892	0	16290	1	0.80	15689
877	F1084451	00000060	21191225	21194344	0	42769	1088	728	0	17463	1	0.80	11975

システムC 300秒以上
SCALAR = 1.0
VECTOR = 10.0
J9175.DAY0121H.DATA

Table 5.7 (Cont'd)

JNO	JOB NAME	INPUT	START	END	SCPU	UCPU	SMEM	UMEM	SIO	UIO	LL	V	VCPU
652	F9239150	00000061	21171451	21173442	0	57120	1088	912	0	3078	1	0.80	15993
440	F3051240	00000062	21133714	21141137	0	60006	832	588	0	4	1	0.80	16801
811	N8215388	00000063	21175205	21182623	0	30007	1600	1448	0	5649	1	0.80	8401
338	N8181213	00000064	22005008	22013919	0	71893	1600	1308	0	1313	0	0.60	33070
375	F9025698	00000065	22020946	22050155	0	301571	4160	2788	0	1069	0	0.60	138722
312	F9024662	00000066	21124438	21130611	0	58557	1088	1036	0	324	0	0.60	26936
415	F3340502	00000067	21132514	21140206	0	57978	1088	840	0	1003	0	0.60	26669
135	F3340500	00000068	21104414	21105823	0	57780	1088	844	0	1001	0	0.60	26578
299	F9024660	00000069	21121328	21123733	0	60005	1088	1036	0	325	0	0.60	27602
772	F2322479	00000070	22011806	22044117	0	328346	1088	832	0	661	0	0.60	151039
809	N8215387	00000071	21175027	21180456	0	30007	1600	1448	0	5670	0	0.60	13803
558	F8223042	00000072	22022452	22032759	0	92129	2112	2052	0	757	0	0.60	42379
724	F3119062	00000073	21172030	21173248	0	37433	320	276	0	419	0	0.60	17219
227	F9024658	00000074	21113744	21115218	0	39189	1088	1036	0	326	0	0.60	18026
949	C0521012	00000075	22015054	22051313	0	328716	2112	1260	0	64008	0	0.60	151209
TOTAL						82773(SEC)				458670(TIMES)			22320(SEC

```
LMAX   = 2
P      = 0.194 0.493 0.314
V      = 0.600 0.800 0.950
SCALAR= 1.0
VECTOR=10.0
IOTIME= 0.03
CPULT =  300.(SEC)
MIN JOB NO. =   24
MAX JOB NO. =  949
TOTAL USER   CPU      82773(SEC)
TOTAL VECTOR CPU      22320(SEC)
TOTAL I/O  TIME       13760(SEC)
TOTAL  JOBS              75
```

L	FREQ (THEORY)	PROB	FREQ (ACTUAL)
0	14.6	0.194	12
1	37.0	0.493	40
2	23.5	0.314	23

KHI-SQUARE= 0.7072

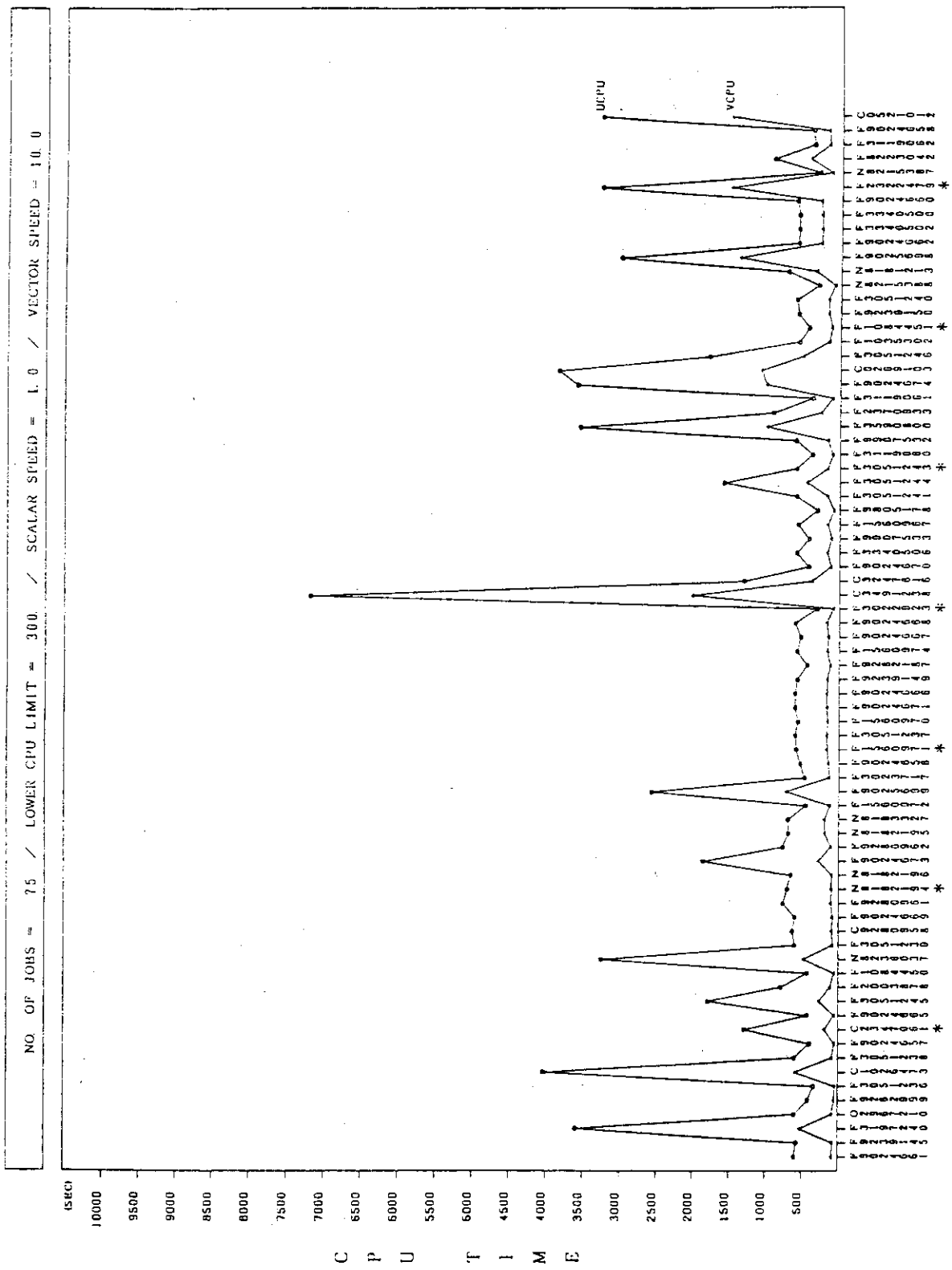


Fig. 5.8 Graphical representation of job sequence of Table 5.7.

(4) M--200CPU 時間を基礎データとすることの妥当性

原研計算センタのバッチ処理用M－200計算機2システムは昭和58年6月にはM－380 2システムと交換される。M－380のCPU性能はM－200のそれの約2.4倍である。そこでTable 5.2等のCPU時間は2.4で除算した値が使用されるべきではないかという疑問が生じる。この疑問は原研の計算機処理の環境においては現在のところ当っていない。現在バッチ・ジョブ1件当りのCPU時間は65秒であるが，これがM－380においてもやはり60～70秒となることは次のTable 5.8の過去データから推測できる。

Table 5.8 CPU time per job in 1976 - 1981.

時　　期	計　算　機	CPU時間／ジョブ	備　　考
昭和51年4月	F230－75	0.020 時間	
↓	↓	↓	この間0.02～0.034 の間を変動
昭和55年1月	F230－75	0.033	
昭和55年5月	F230－75	0.046	M－200はF230－75の3倍のCPU性能。
昭和55年4月	M－200	0.016	
昭和56年4月	M－200	0.023	0.046÷3≒0.016

上記のTable 5.8について補足すると，ジョブ当りのCPU時間は昭和51年4月～昭和55年1月の期間0.020 時間（72秒）～0.033時間（119 秒）の間を変動している。昭和55年2月からM－200 1システムが使用可能となり，F230－75 に少し余裕ができた。このときはジョブ1件当りのCPU時間は0.046 時間（166秒）に急増している。このCPU時間の増加はM－200にそのまま持越されている（0.046÷3＝0.016，M－200 のCPU性能は F230－75 の3倍）。

その1年後の昭和56年4月にはジョブ1件当りのCPU時間は0.023 時間とF230－75の水準に帰っている。この月のCPU時間はバッチ処理で1426時間で，これにタイムシェアリング処理の時間を加えて12倍すると，4.2.1節で述べた年間のCPU時間の上限18000時間に既に到達している。この後の1年間はCPU時間は0.016～0.018とむしろ減少し，逆にジョブ処理件数は増加している。これは利用者がジョブを小刻みにランしてジョブ回転時間の短縮をはかっていることを示している。以上から現在のM－200のジョブ・パターンでもってM－380のスカラ演算性能を持つベクトル計算機でのジョブ処理状況を推定してよいことがわかる。これとは別に，ジョブ1件当りのメモリ量と入出力回数は過去2年間一定しているものの，ジョブ処理件数は2～3倍増加しているので，入出力（I／O）ネックの問題は依然として存在する。

5.2.2 シミュレーション結果についての所見

Fig. 5.1（詳細は付録3）で示した待ち行列モデルによって，Table 5.2, 5.3, 5.4のジョブ処理を模擬した結果をFig. 5.9, 5.10, 5.11に示す。このモデルはオペレイティング・システム（OS）のオーバーヘッドを考慮していないので，処理の模擬結果図においてCPU

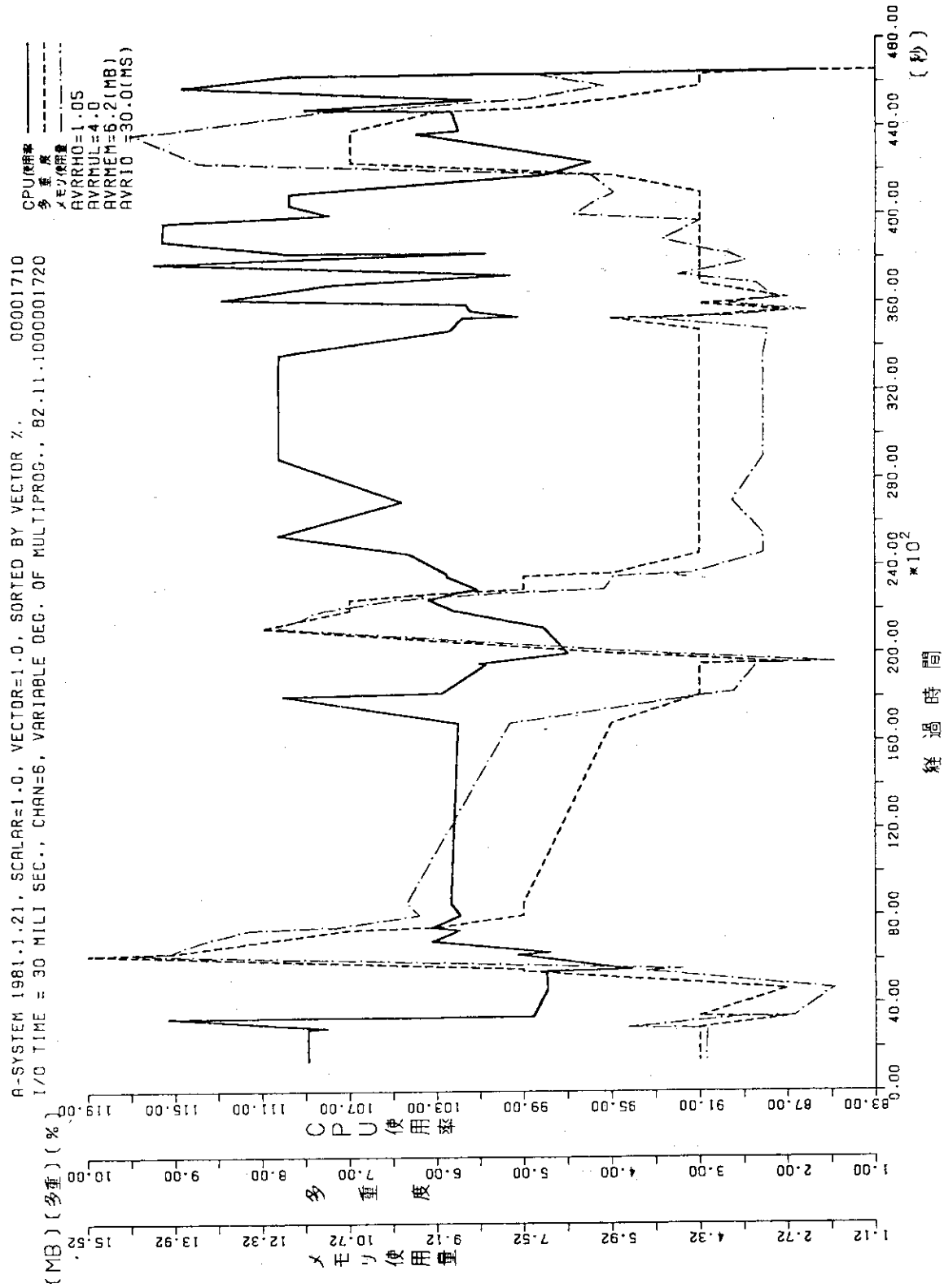


Fig. 5.9 Simulated job processing of job sequence of Table 5.2.

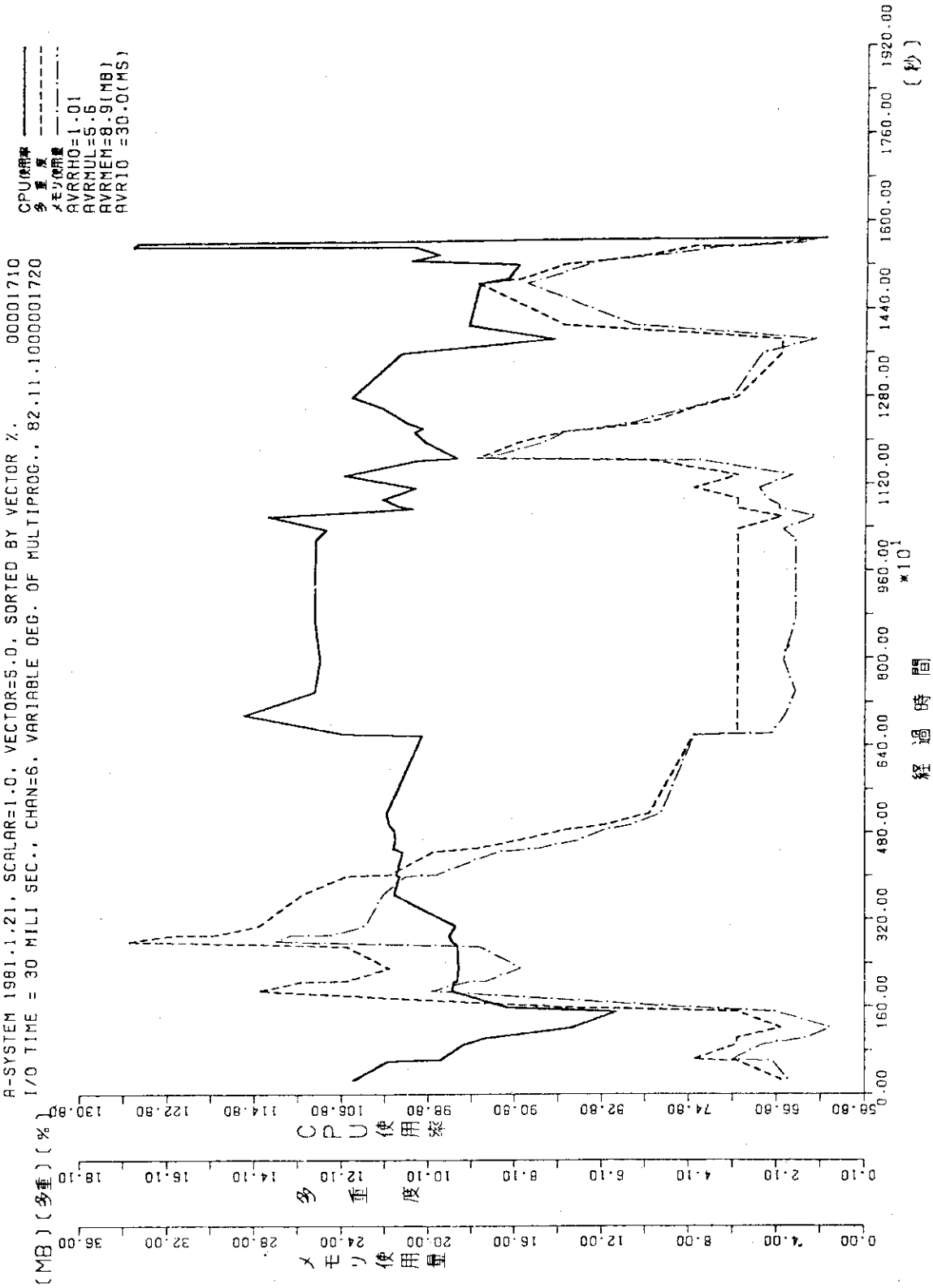


Fig. 5.10 Simulated job processing of job sequence of Table 5.3.

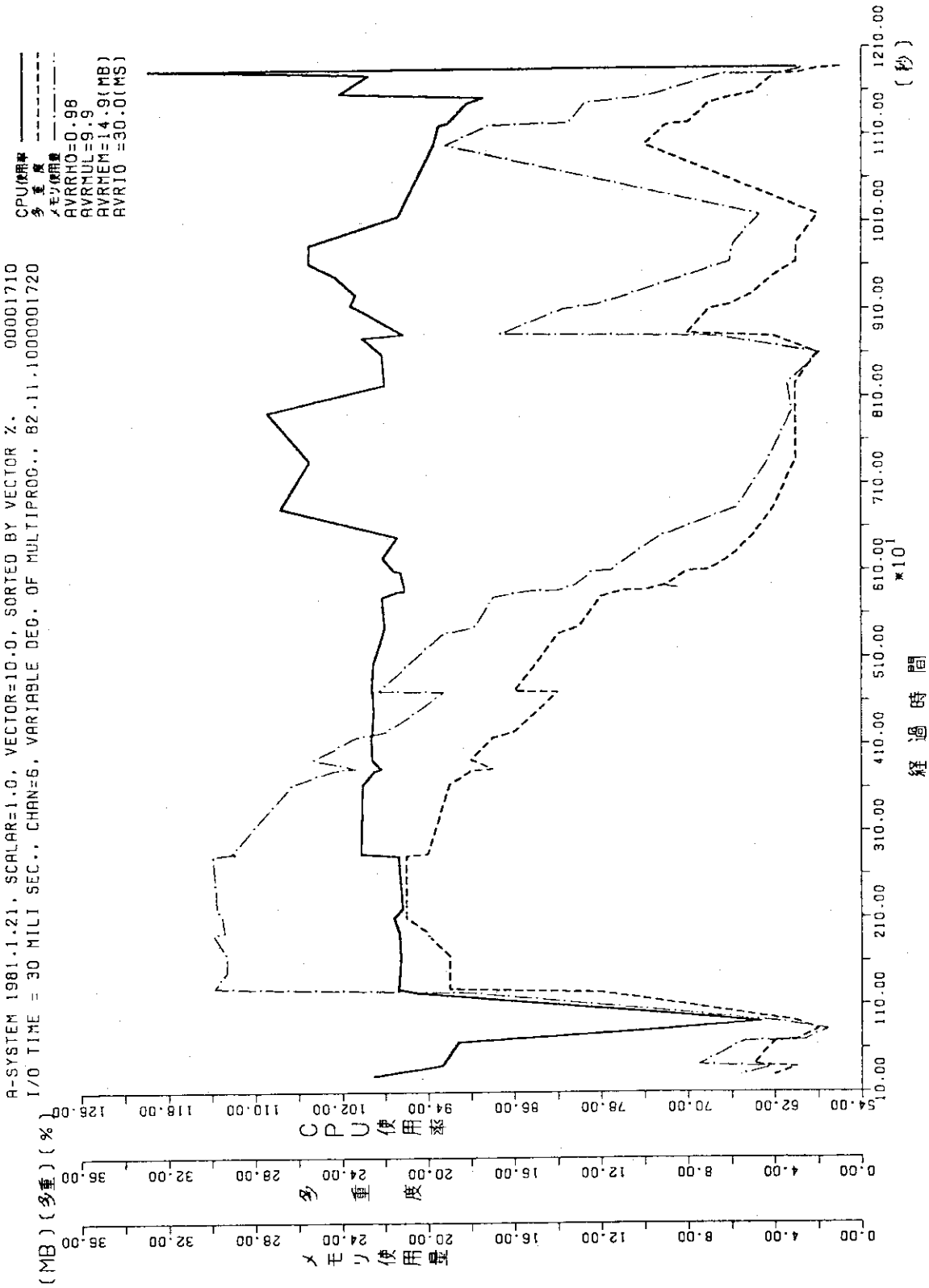


Fig. 5.11 Simulated job processing of job sequence of Table 5.4.

使用率とあるのはユーザ側のＣＰＵ使用率である。同じく多重度とメモリ使用量もユーザ・プログラムのものである。この程度のバッチ・ジョブ処理では，ＯＳのオーバーヘッドは数パーセントであって，無視できる程度に小さい。ＣＰＵ使用率は本来1.0，即ち１００％を超えない筈であるが，各図において，それがしばしば１００％を超えている部分がある。これはこの模擬に使用されている確率的な近似式が現実のデータと合わない場合に出現する。この現象は主としてＣＰＵバウンドのジョブが２～３多重で処理されるときに起こる。その発生メカニズムについては付録３で述べよう。この待ち行列理論によるものとは別に，現実のＯＳの動きを模擬するプログラムを作り，同じジョブ・パターンを処理させると，前者のものとほぼ同じ処理と時間経過を示す（報告（42））。Fig. 5.9, 5.10, 5.11における処理時間は，本来のそれらよりも１０パーセント程度短くなっている。この難点はＩ／Ｏバウンド・ジョブとCPUバウンド・ジョブを同一多重度中に混在させることで解消されるか，大まかに必要資源の傾向を見るには現状のものでも間に合う。

Fig. 5.9, 5.10, 5.11でVECTOR＝1.0, 5.0, 10.0とあるのは第2.3節式(2)におけるα値のことで，それぞれα＝1.0, 5.0, 10.0に対応し，ベクトル計算機のベクトル演算能力を示している。α＝1.0はスカラ計算機のことである。

これら３つの図を眺めると次のようなことがいえる。

(1) スカラ計算機（VECTOR＝1）とベクトル計算機（VECTOR＝5)との処理時間を比較してみると，前者は４６,５００秒，後者は１５,７００秒となり，ベクトル計算機によって約３倍の処理性能向上となる（ただし原研の７０％のＣＰＵ量についてのみ）。

(2) ベクトル計算機（VECTOR＝10）の処理時間は１２,０００秒であるから，VECTOR＝５の場合と比較すると1.3倍となる。

(3) VECTOR＝5, 10の場合ともにベクトル化率の高い順にジョブがソートされている影響が出ている。即ちＣＰＵ時間が短くなったジョブ列のあたりでは，単位時間当りのＩ／Ｏ回数が増加し，ＣＰＵ使用率を上げるために多重度が大きくなる。それにつれてメモリ使用量も増加する。ベクトル化率の高いジョブを先に処理するのは，そのようなジョブの回転時間を短縮するためである。

(4) ベクトル化率の低い計算コードは（スカラ演算の）ＣＰＵ使用率を高め，その結果多重度，メモリ使用量は抑えられる。

(5) 並行して動作する入出力パスの数（CHAN）を６とすると平均入出力時間は当初与えた３０ミリ秒と同じである。しかし，これはあくまでも平均の話であって，ＣＰＵ時間に比して入出力回数が多いジョブが多い場合は入出力待ちまで含めた入出力時間は長くなる。いまFig. 5.10, 5.11 において多重度が上っている状況はその場合に当る。この状況での入出力時間の伸びを見るためにCHAN＝３とし，VECTOR＝5 と１０について基本の入出力時間を１回当りそれぞれ30, 20, 10ミリ秒と変え，多重度がもっとも上っているあたりの入出力時間を見るとTable 5.9のようになる。入出力パス数を３とした理由は6.2節で述べよう。

Table 5.9 Delay in I/O access in case of vector=5.0 or 10.0.

基本入出力時間／回	入出力時間／回	
	VECTOR＝5	VECTOR＝10
30ミリ秒	40～59ミリ秒	50～68ミリ秒
20	21～29	35～41
10	10～11	11～12

注　入出力バス数＝3

基本の入出力時間が1回当り10ミリ秒のときには，現在のジョブ・パターンでは入出力時間の伸びあまり大きくない。20，30ミリ秒のときは無視できぬほど大きくなる。ベクトル化率が高いジョブが入出力待ちになっている間隙をぬってスカラ演算のみのジョブがCPUを獲得するようになると利用者が計算コードをベクトル化しようという意欲は大いに減退する。これを救うのは入出力バッファ装置である。これを有効利用するためのジョブ・スケジューリング手法を5.3(4)で示唆する。

(6) ユーザ・ファイル用入出力バッファ装置の効果はTable 5.9 から推察できる。

以上と同じ議論がFig. 5.12, 5.13, 5.14のCシステムについてもいえる。ここでCシステムの模擬計算結果を挙げたのは，ジョブ・パターンが多少異なっても同じ傾向であることを示すためである。

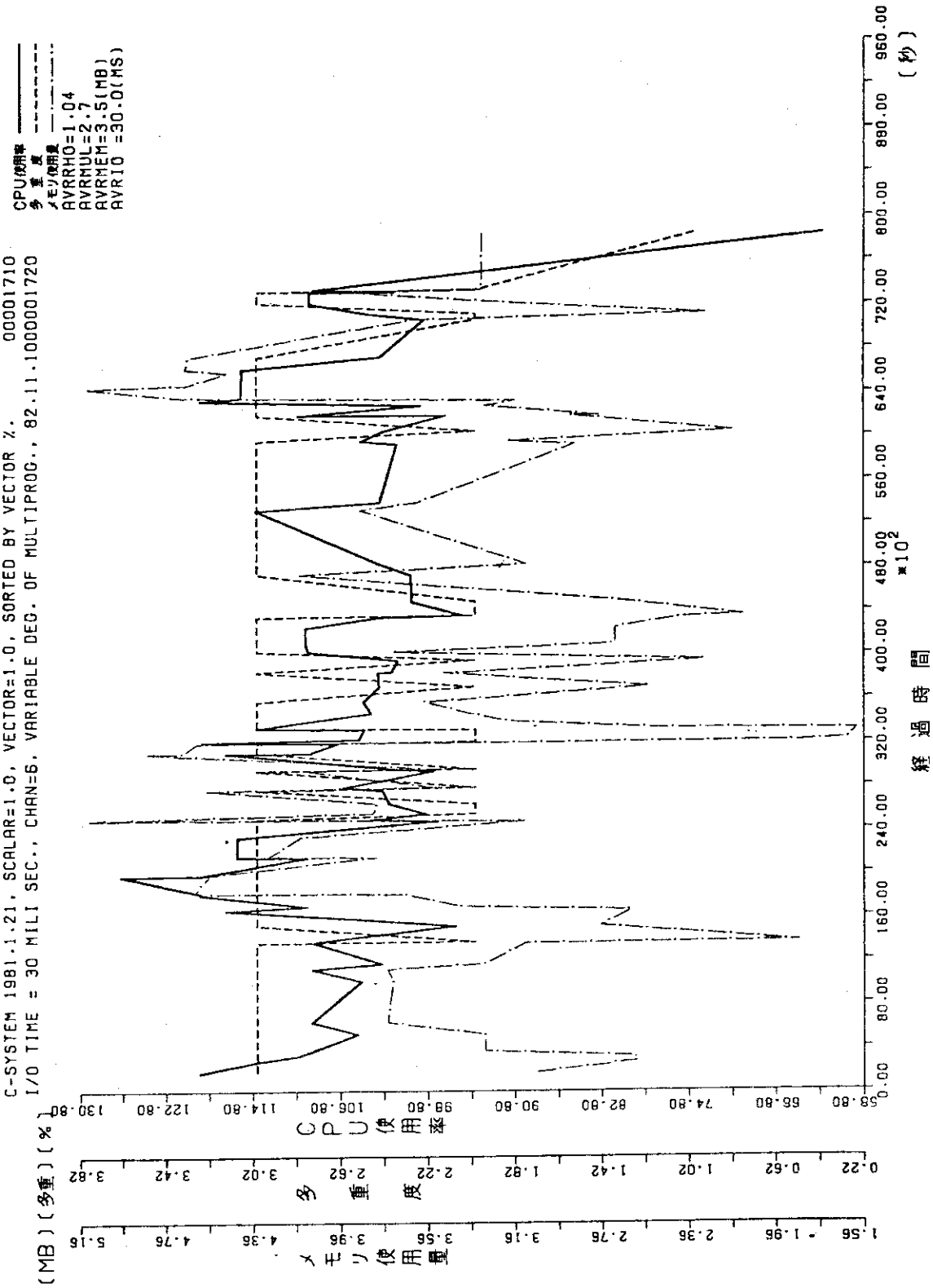


Fig. 5.12 Simulated job processing of job sequence of Table 5.5.

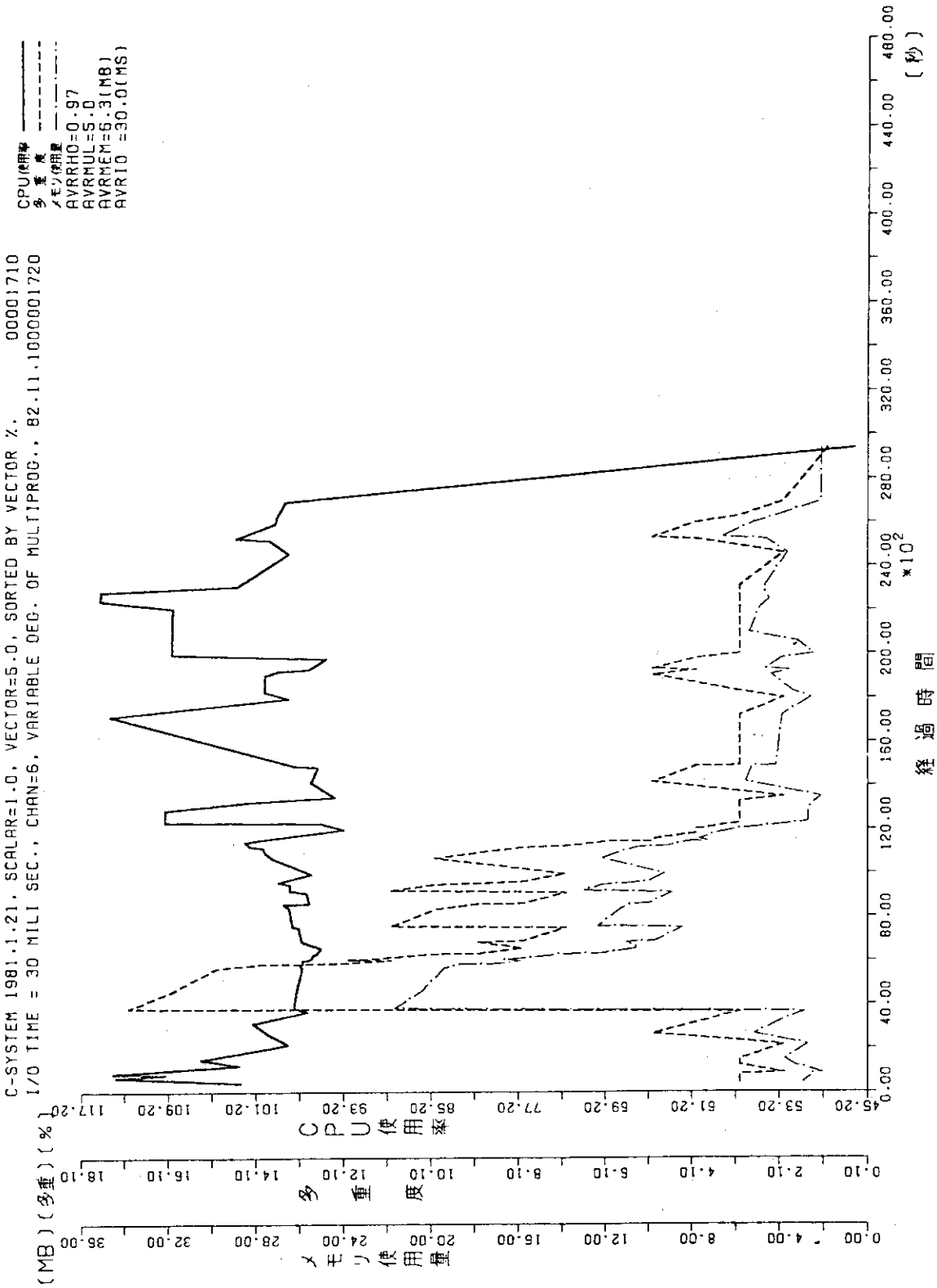


Fig. 5.13 Simulated job processing of job sequence of Table 5.6.

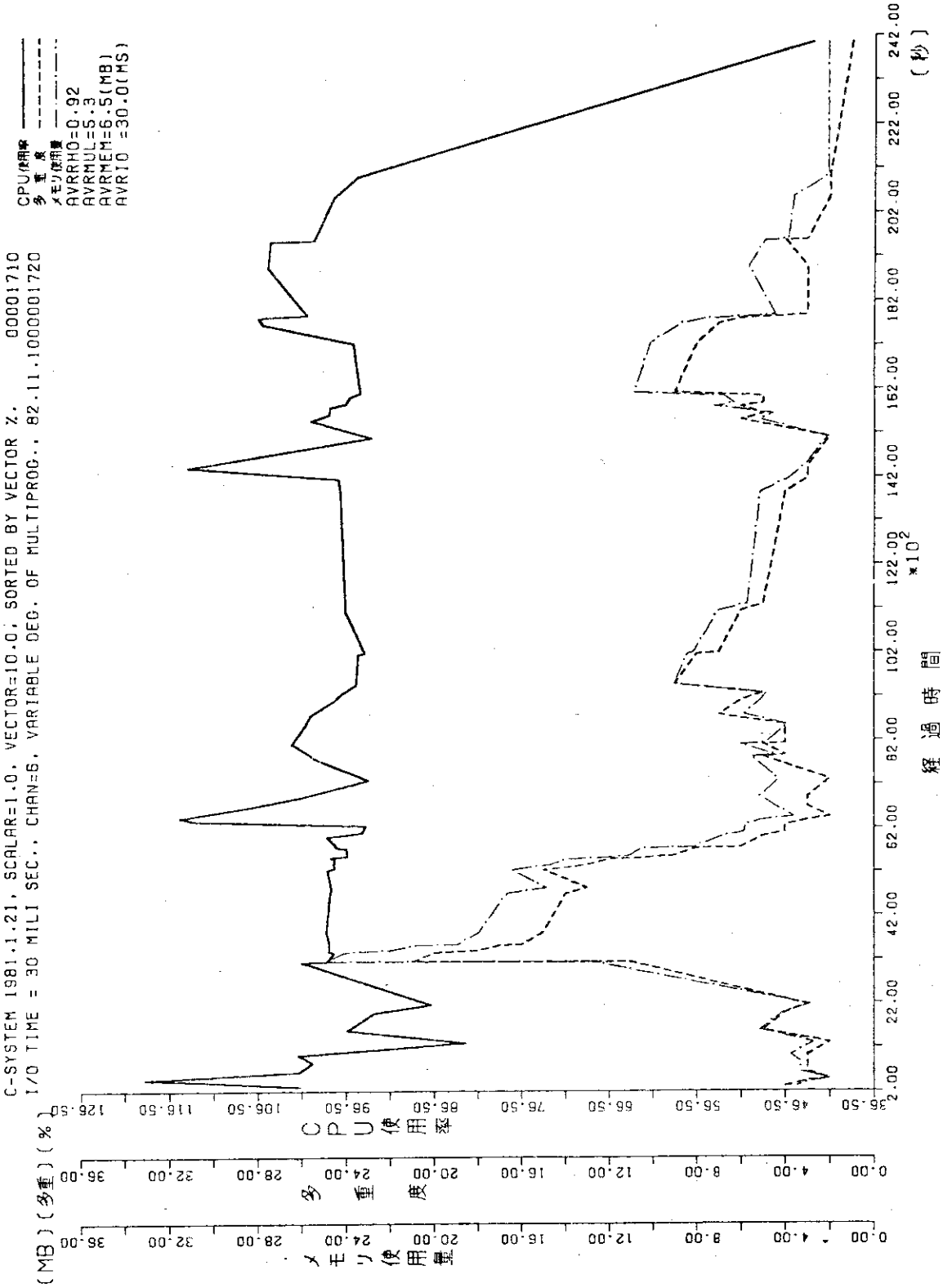


Fig. 5.14 Simulated job processing of job sequence of Table 5.7.

5.3 1システム・ベクトル計算機の持つべきハードウェアとスケジューリング・ソフトウェア

前節の所見とTable 3.1のデータから1システムのベクトル計算機の持つべきハードウェアとスケジューリング・ソフトウェアを推定してみよう。

(1) 演算性能

各種計算コードに対するベクトル化された部分での平均性能ベクトル演算の性能が，スカラ演算のそれの5倍（α，あるいはVECTOR値が5）程度が当面原研のベクトル・ジョブに合っている。

(2) ユーザ側メモリ容量

ユーザ側メモリ容量を次の算式で推定する。

$$\text{メモリ容量} = (\text{最大多重度でのメモリ量}) \times (\text{ベクトル化によるメモリ増加率}) + (\text{ジョブ平均作業域}) \times (\text{作業域を必要とするジョブの出現確率}) \times \left\{ \begin{matrix} \text{ジョブ多重度（最大）} \\ \text{ジョブ多重度（平均）} \end{matrix} \right\}$$

VECTOR＝5のときFig. 5.10 とTable 3.1 から，最大多重度でのメモリ量＝26MB，ジョブ平均作業域＝73MB，作業域を必要とするジョブの出現確率＝3／15であるから，

$$\text{メモリ容量} = 26\text{MB} \times 1.6 + 73\text{MB} \times \frac{3}{15} \times \left\{ \begin{matrix} 16 \\ 6 \end{matrix} \right\} \cong \left\{ \begin{matrix} 276\text{MB} \\ 130\text{MB} \end{matrix} \right\}$$

VECTOR＝10のときFig. 5.11 とTable 3.1 から

$$\text{メモリ容量} = 30\text{MB} \times 1.6 + 73\text{MB} \times \frac{3}{15} \times \left\{ \begin{matrix} 20 \\ 10 \end{matrix} \right\} \cong \left\{ \begin{matrix} 340\text{MB} \\ 194\text{MB} \end{matrix} \right\} \quad 。$$

また仮想入出力機能利用の可能性を見るために昭和57年3月5日17：00～19：43のCシステムでワーク・ファイルを使用した137ジョブのファイル使用状況を調査した（Table 5.10）。

これからわかるように1ジョブ平均6.3ファイルを使用し，このうち3.3がワーク・ファイルである。1ジョブ当り983.4 の入出力回数のうち545.4 回はワーク・ファイルの入出力である。この545.4 回のうち半分が読み（read）とすると545.4÷2÷983.4＝0.277，即ち約30％が仮想入出力（VIO）の対象となる。また別の統計からTable 5.11のようにユーザが使用するワーク・ファイルの70～90％は1MB以下で，特に0.5～1MBの範囲にあるものがほとんどである。最近ワーク・ファイル領域割当の初期値を減じて運用しているが不都合は生じていないので1ファイル当り0.5MB程度とみるのが妥当であろう。そうすると上記メモリ容量に加えてさらに

$$0.5\text{MB} \times 3.3\text{個} \times \left\{ \begin{matrix} 16 \\ 6 \end{matrix} \right\} \cong \left\{ \begin{matrix} 26\text{MB} \\ 10\text{MB} \end{matrix} \right\} ,$$

あるいは

$$0.5\text{MB} \times 3.3\text{個} \times \left\{ \begin{matrix} 20 \\ 10 \end{matrix} \right\} \cong \left\{ \begin{matrix} 33\text{MB} \\ 17\text{MB} \end{matrix} \right\}$$

必要ということになる。このメモリはベクトル化ジョブでなくても通常のスカラ・ジョブが使用しているワーク・ファイルのためにも必要なものである。

(3) 入出力バッファ装置の容量とアクセス性能

バッファ容量を次の算式で推定する。

バッファ容量＝（ジョブ当り入出力容量）×（最大多重度）

VECTOR＝5のときはFig. 5.10 とTable 3.1 から

バッファ容量＝540KB×16≒9MB

VECTOR＝10のときはFig. 5.11 とTable 3.1 から

バッファ容量＝540KB×20≒11MB 。

アクセス性能はTable 5.9から6KB～18KB 当り10ミリ秒でなければならない。これまでの模擬計算で使用した1回の入出力時間30ミリ秒という値は，20～25ミリ秒のシークとポジショニング，5～10ミリ秒のデータ転送を仮定している。この時間では6KB～18KBのデータ転送をおこなえる。

(4) 精度の良いジョブ・スケジューリング機能

ベクトル計算機では，ベクトル演算器から最高の性能を引き出すこと，このために演算器に間断なくデータを送り続けることが大切である。ところが，巨大な作業域を持つベクトル化ジョブどうしや，多数回のファイル入出力を持つベクトル・ジョブは，メモリ・アクセスの衝突，同一ファイル・ボリュームへのアクセス競合を引き起す。これらの現象は結果としてよりベクトル化率の低いジョブへCPUを渡す原因になる。これらの現象をできるだけ回避するために新しいジョブ・スケジューリングの概念と機能が必要である。それらを項目で挙げる。

(a) ジョブ履歴を利用するスケジューリング

過去数回のジョブ実行記録（CPU時間，使用ファイル名，その容量，ブロック・サイズと入出力回数，使用メモリ量とベクトル化率，等）を保存しジョブのスケジューリングに利用する。筆者らは報告（42），論文（43）で，これらの情報から入力されたジョブの要求する計算機資源量をほぼ正確に推定できること，その推定量を使ってジョブ実行時のジョブの経過時間等を予測できることを示した。その予測方法を拡張，改良したものが本報告で使用しているスケジューラである。このスケジューラはTable 5.3 のジョブ列の処理を模擬し，Fig. 5.10の結果を出すまでにM-380計算機で7.5秒を要する。このうち37パーセントの時間は意外にも或る変数のベキ乗計算結果の桁数をチェックする対数ALOG10(X)の計算に使用している。これは時間のかからない別のひとつステートメントで置き換えることができる。そのとき7.5秒は4.6秒に減少し，その減少後の時間4.6秒の95パーセントは，CPUのサービス量を定める幾何分布$g_i=\sigma^i(1-\sigma)$とこれから求まるサービス時間の各ジョブへの分配の計算に使用しているが，この計算はほぼ完全にベクトル化でき，結局はスケジュール時間はベクトル計算機でおこなえば63個のジョブに対して1秒程度になる。実運用では適当な時間間隔をおいて20個程度のジョブを対象とすればよく，そのとき20個のジョブのスケジュールは0.5秒

Table 5.10 Number of work files per job and I/O accesses per work file.

ジョブ数	ジョブ当りシステムファイル	システムファイル当り入出力回数	ジョブ当りユーザファイル	ユーザファイル当り入出力回数	ジョブ当りワークファイル	ワーク・ファイル当り入出力回数
137	0.8ケ	78.4	2.2	359.7	3.3	545.4

Table 5.11 Distribution of capacities of work files.

* DATE 1982-03-26 TIME= 14:26:48

SYSTEM-A

	FILES	(%)	SPC(KB)	(%)
0.5MB>=	4	13.79	836	5.28
0.5-1MB	24	82.76	13680	86.33
1-2 MB	1	3.45	1330	8.39
2-5 MB	0	0.0	0	0.0
5-10MB	0	0.0	0	0.0
10MB =<	0	0.0	0	0.0
TOTAL	29	100.00	15846	100.00

SYSTEM-C

	FILES	(%)	SPC(KB)	(%)
0.5MB>=	6	16.22	1254	1.76
0.5-1MB	27	72.97	15960	22.36
1-2 MB	0	0.0	0	0.0
2-5 MB	0	.0.0	0	0.0
5-10MB	3	8.11	22800	31.95
10MB =<	1	2.70	31350	43.93
TOTAL	37	100.00	71364	100.00

*: DATE 1982-03-26 TIME= 14:49:21

SYSTEM-A

	FILES	(%)	SPC(KB)	(%)
0.5MB>=	0	0.0	0	0.0
0.5-1MB	6	100.00	3800	100.00
1-2 MB	0	0.0	0	0.0
2-5 MB	0	0.0	0	0.0
5-10MB	0	0.0	0	0.0
10MB =<	0	0.0	0	0.0
TOTAL	6	100.00	3800	100.00

SYSTEM-C

	FILES	(%)	SPC(KB)	(%)
0.5MB>=	4	8.33	247	0.41
0.5-1MB	37	77.08	21850	36.36
1-2 MB	2	4.17	3800	6.32
2-5 MB	3	6.25	11400	18.97
5-10MB	1	2.08	9500	15.81
10MB =<	1	2.08	13300	22.13
TOTAL	48	100.00	60097	100.00

*: DATE 1982-03-26 TIME= 15: 8:40

SYSTEM-A

	FILES	(%)	SPC(KB)	(%)
0.5MB>=	5	8.77	1216	2.60
0.5-1MB	47	82.46	28006	59.97
1-2 MB	1	1.75	1330	2.85
2-5 MB	4	7.02	16150	34.58
5-10MB	0	0.0	0	0.0
10MB =<	0	0.0	0	0.0
TOTAL	57	100.00	46702	100.00

SYSTEM-C

	FILES	(%)	SPC(KB)	(%)
0.5MB>=	0	0.0	0	0.0
0.5-1MB	37	94.87	21090	78.72
1-2 MB	1	2.56	1900	7.09
2-5 MB	1	2.56	3800	14.18
5-10MB	0	0.0	0	0.0
10MB =<	0	0.0	0	0.0
TOTAL	39	100.00	26790	100.00

程度で終る。ＣＰＵ時間7.5，4.6秒はいずれもＭ－３８０の場合である。このスケジューラを使えば，巨大な作業域を必要とする２個のジョブの一方が処理を終えた頃に他方を起動するとか，同一ボリュームへ同時にアクセスする可能性をもつ２個のジョブのひとつの処理が終了し，入出力バッファ装置からこのジョブのファイルが存在しなくなった頃に他のジョブを起動するなどの操作が可能となる。スケジュールに要する計算時間の点で，ここに書いたほど簡単にはいかないだろうが，少なくとも実験するだけの価値はある。

(b) 可変多重度の機能

ベクトル化ジョブは通常はＣＰＵバウンドであることが多い。ベクトル化率まで含めて既にＣＰＵ使用率が高くなっているジョブ処理環境では，指定されている固定多重度の上限に対して未だ余裕があっても多重度を上げるべきではない。また指定された固定多重度の上限であってもベクトル化率の高いジョブが入力されれば，現在処理しているスカラ・ジョブ，あるいはベクトル化率の低いジョブをスワップ・アウトし，より高いベクトル化率のジョブの実行を開始できる機能が必要である。

ソフトウェアについては存在しないがあれば望ましいもの（スケジューリング・ソフトウェア）を述べた。

以上のハードとソフトをVECTOR＝５の場合について図示するとFig. 5.15 のようになる。

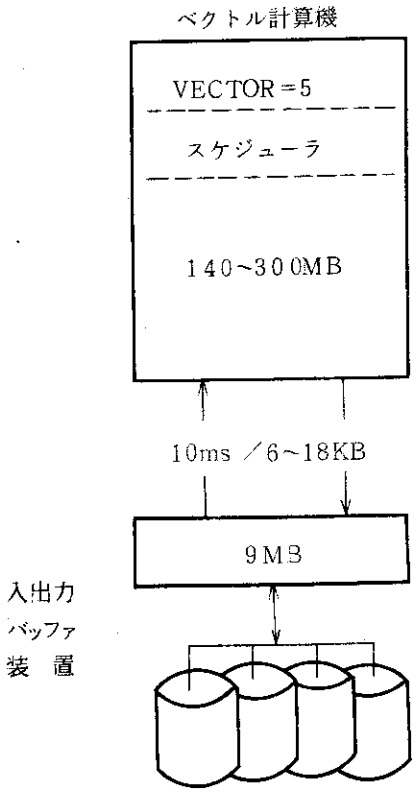


Fig. 5.15 Necessary resources for vector processing based on findings of the simulations.

6. 複合計算機システムの構成

第4章では当面必要とするベクトル計算機の台数について，また第5章では1台のベクトル計算機を対象とし，ベクトル・ジョブを効率よく処理するためのハードウェアとソフトウェアについて議論した。この章ではベクトル計算機，スカラ計算機，およびこれら計算機を使用する利用者まで含めた系を考え，この系のなかでのベクトル計算機の効果的使用法について議論する。

6.1 現在（昭和58年7月）の計算機構成と入出力パス数

(1) Aシステムの入出力パス数

昭和58年7月現在の原研計算センタの計算機構成はFig. 6.1 のようになる。まず，この計算機システムのなかの大型ジョブ処理用計算機が昼間のジョブを処理する状況を模擬してみよう。5.2節の議論から昭和58年7月における各ジョブのCPU時間は1年半前のジョブのCPU時間と絶対時間において変りないと仮定し，57年1月21日13時～14時のジョブ・パターンで大型ジョブ処理用Aシステムの処理を模擬するとFig. 6.2 のようになる。

このときAシステムに指定されていたジョブ・クラスは大ジョブ2，小ジョブ2多重である

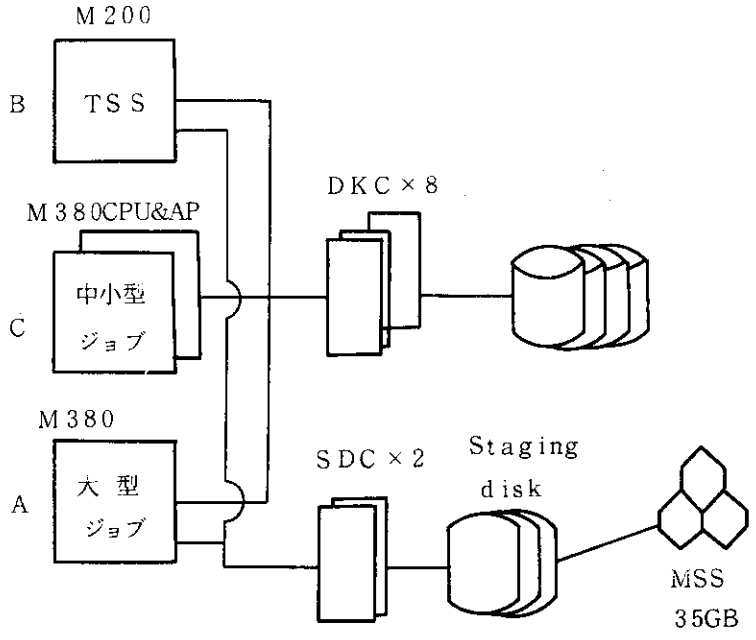


Fig. 6.1 Processors and I/O paths of JAERI at June 1983.

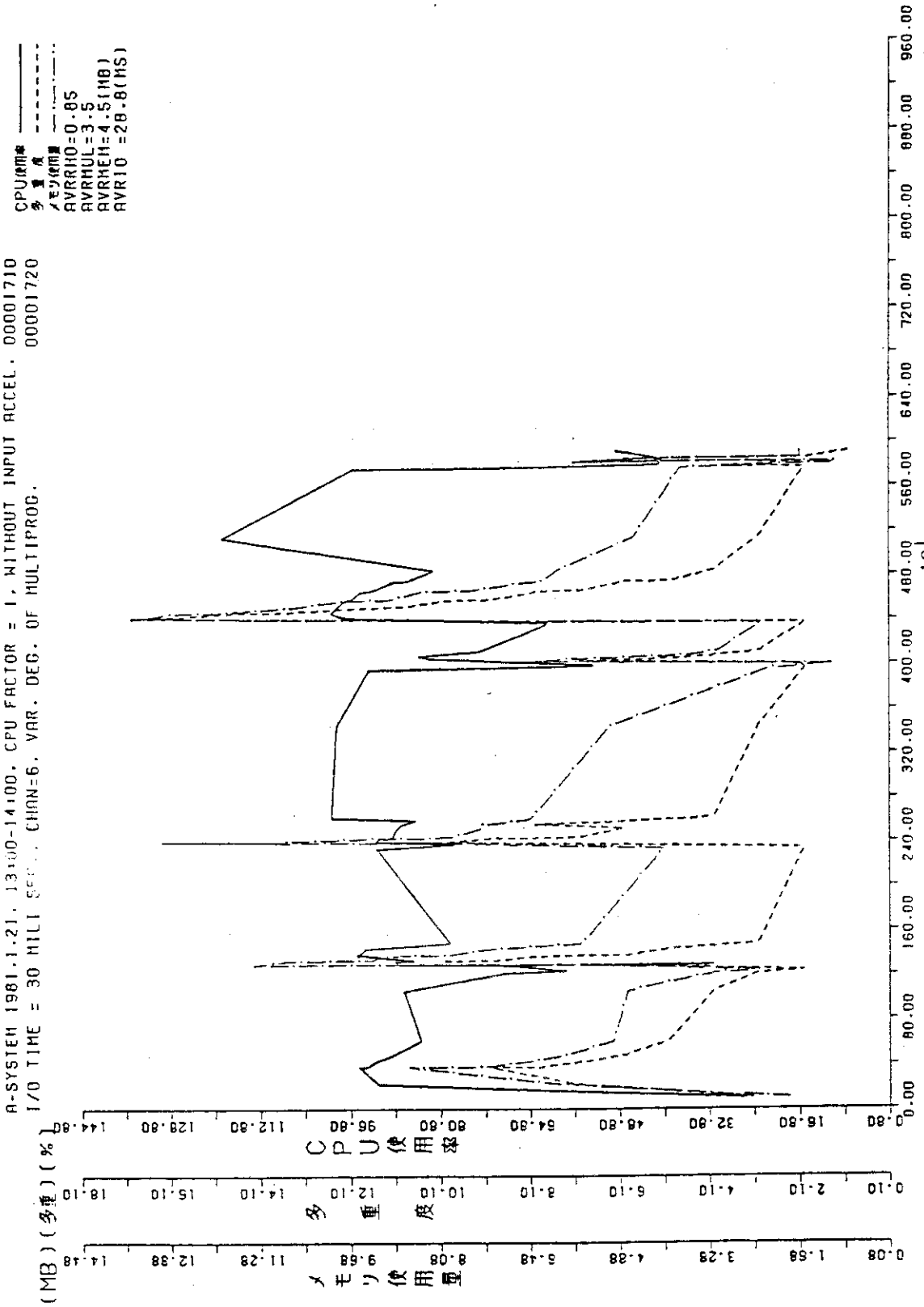


Fig. 6.2 Simulation of job processing for day time A computer system.

からAシステムの運用の考え方を反映したジョブ処理パターンとなっている。Fig. 6.2 のCPU使用率，多重度，メモリ使用量のデータはジョブ処理が終了したときの値であるから，多重度2で大ジョブのみが実行されているときにCPU使用率が高いことがわかる。入出力パス数を6，1回の入出力時間を30ミリ秒としたとき，多重度が上ったときには待ち時間まで入れた入出力時間は31～38ミリ秒となっている。しかし図からわかるとおり，入出力時間が伸びる機会は多くない。このことから，このAシステムでは入出力パス数は6以下でよいことがわかる。入出力パスの数6は昭和58年7月現在の磁気ディスク装置台数8台をほとんど並行動作可能としたときの値である。

(2) Cシステムの入出力パス数

同じ時間帯のCシステムについてみてみよう。

Cシステムは2台のCPUを持ち，それぞれがジョブを処理するが，ここでは計算プログラムの便宜上，M-380CPU の2倍の性能を持つ1台のCPUと仮定する。57年1月21日13：00～14：00にCシステムに入力されたジョブを可変多重度で処理するとFig. 6.3 のようになる。Cシステムは10多重（中4，小5，事務ジョブ1）で運用されている。中，小のジョブが入りまざってCPU使用率，多重度，メモリ使用量が激しく変動する。平均多重度が5.1であるから最適な固定多重度は5よりも大きいことがわかる。

多重度が増加したときの「待ち」まで入れた入出力時間は31ミリ秒～39ミリ秒，全平均でも31.3ミリ秒であるから，このシステムは少なくとも6本以上の並行して動作する入出力パス数を持たなければならないことがわかる。

(3) Bシステムの入出力パス数

Bシステムはタイムシェアリング処理及び全システムのスプール・ボリュームからラインプリタへの入出力処理をおこなっている。Bシステムの入出力回数からBシステムの所要並行入出力パス数を推定してみよう。A，B，Cシステムのユーザ側の入出力回数を昭和57年1月の統計でみるとTable 6.1のようになる。

Table 6.1の入出力回数はユーザ側のものでシステムの入出力回数は入っていない。しかし，5.2節からこの節までに述べて来たA，およびCシステムの模擬計算では，これらシステムの入出力回数を各ジョブの入出力回数に加えてある。Bシステムのユーザ側，システム側の入出力回数は，システム性能評価ツールPDL／PDA のデータから，それぞれ半々程度であることがわかっている。Table 6.1 から昭和57年1月21日においては，A，B，Cシステムのユーザ側入出力回数の比は26：21：53 パーセント，また1月全体では26：18：56 パーセントである。これら2つの比から，1月21日の入出力回数から1月全体の傾向を予想しても大きな差は出ないことがわかる。また，この模擬計算に使用したA，Cシステムのジョブ数，入出力回数等はTable 6.2 のとおりである。これとTable 5.1 のシステム側入出力回数とを比較すると，Aシステムではユーザ側入出力1回につきシステム側入出力は0.3回，Cシステムではユーザ側1回につき，システム側0.4回となる。したがって全システムの全入出力回数はユーザ入出力を100パーセントとすると，

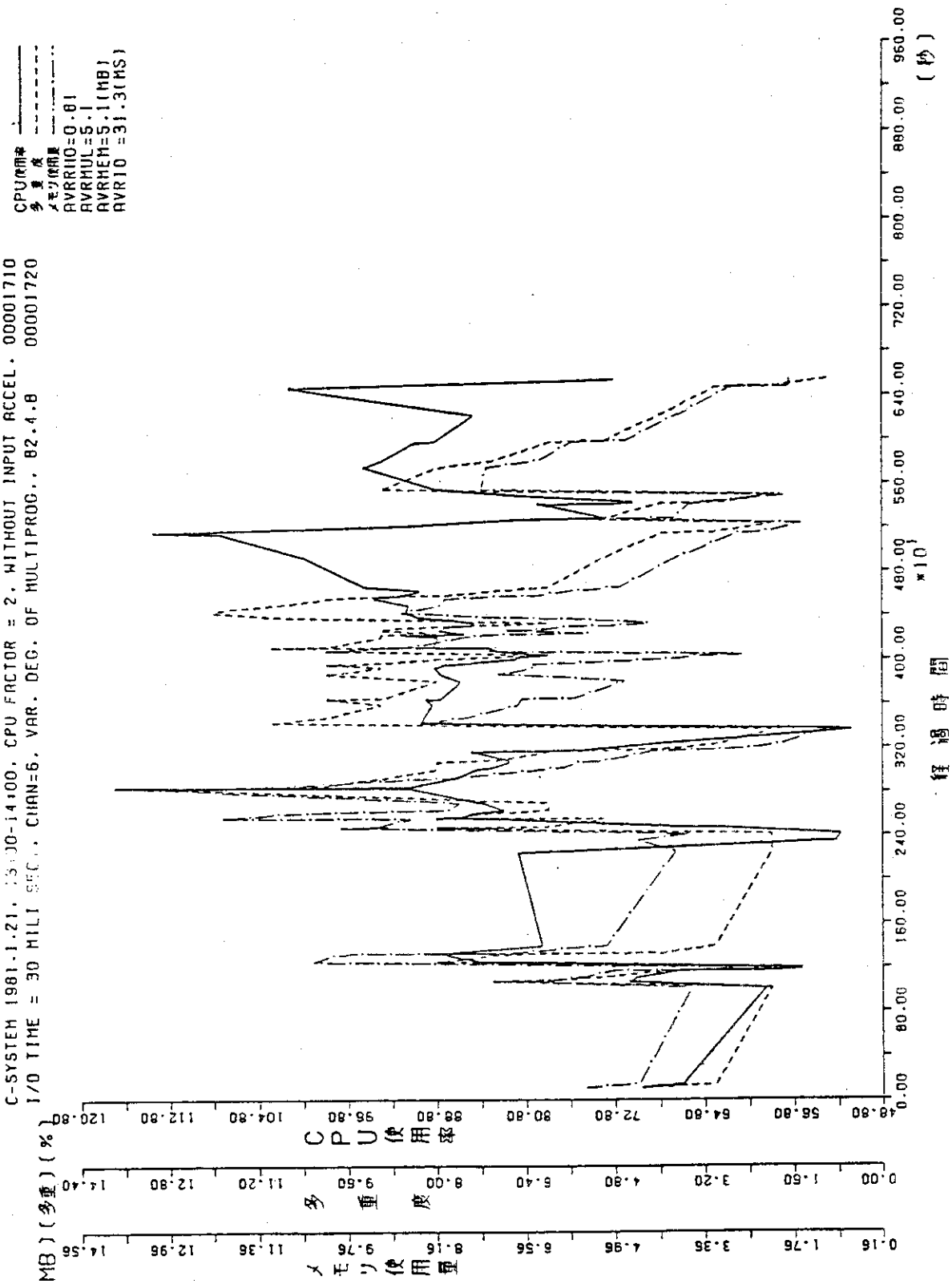


Fig. 6.3 Simulation of job processing for day time C computer system.

Table 6.1 User I/O accesses of A, B, and C computer systems in Jan. 1982 at JAERI computing center.

*** I/O TIMES DATA

(557. 1)
(UNIT=1000TIMES)

	#A	#B	#C	TOTAL	OPEN	CLOSE	FIB.	I/G	OTHER	TOTAL	TSS
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	112.60	17.09	75.82	205.51	0.0	14.09	178.62	0.0	0.02	192.72	12.78
6	657.50	534.47	2630.04	3822.01	63.86	211.87	3127.15	4.10	234.66	3641.64	180.37
7	806.37	473.21	2280.01	3559.59	27.16	235.83	2959.09	4.70	188.94	3415.72	143.87
8	718.24	589.97	1883.44	3191.66	48.16	364.41	2451.59	31.51	154.89	3050.56	141.10
9	305.95	173.75	875.43	1355.14	4.28	9.01	1222.89	1.52	39.37	1277.08	78.06
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	837.46	591.92	2141.07	3570.44	109.16	50.43	3012.73	4.63	159.03	3335.98	234.46
12	628.98	701.95	2438.96	3769.89	101.79	185.32	3112.55	27.51	114.96	3542.12	227.77
13	729.44	482.73	2376.52	3588.69	79.52	194.72	3031.11	50.69	109.32	3465.37	123.32
14	648.17	651.00	2249.35	3548.52	42.92	360.74	2842.82	48.04	104.81	3399.33	149.18
15	16.84	16.85	0.0	33.69	0.0	0.0	33.67	0.0	0.0	33.67	0.02
16	0.0	2436.44	0.0	2436.44	0.0	0.0	2434.10	0.0	0.0	2434.10	2.34
17	34.78	473.28	955.59	1463.65	0.0	1.72	1460.49	0.0	0.00	1462.21	1.44
18	900.96	640.60	2008.67	3550.23	49.10	293.91	2755.72	4.78	224.09	3327.59	222.63
19	896.66	679.23	2751.52	4327.41	58.71	72.48	3876.66	14.89	150.11	4172.85	154.56
20	764.78	521.29	2561.44	3847.51	42.21	275.05	3097.80	8.06	171.93	3595.05	252.46
21	919.99	428.14	2373.70	3721.84	58.79	358.98	2851.73	23.61	195.76	3488.86	232.97
22	786.98	640.51	2369.01	3796.50	50.56	597.29	2741.47	5.42	148.84	3543.59	252.90
23	286.35	246.10	847.01	1379.46	12.69	170.41	1053.03	5.96	91.65	1333.74	45.72
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25	1052.76	719.73	2436.39	4208.88	25.35	741.92	3026.75	34.65	200.65	4029.32	179.56
26	927.77	531.71	2569.41	4028.89	229.30	323.86	3049.49	10.25	213.48	3826.39	202.50
27	921.52	618.08	2433.44	3973.05	114.25	457.59	2941.45	18.19	177.39	3708.87	264.17
28	708.34	619.60	2233.84	3561.79	90.41	325.91	2657.85	12.74	246.29	3333.20	228.58
29	891.48	444.83	2389.80	3726.11	44.12	338.86	2863.34	29.70	247.25	3523.27	202.84
30	213.52	126.91	657.62	998.05	52.18	83.95	748.86	14.96	23.47	923.42	74.63
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTAL	14767.44	13359.39	43538.11	71664.94	1304.52	5668.35	57530.99	355.93	3196.89	68056.69	3608.26
SE	197.26	3224.71	1686.73	5108.70	0.0	80.88	5011.23	0.0	0.00	5092.12	16.59

Table 6.2 Number of jobs, total CPU time, I/O accesses of job sequence for the simulation of A or C computer system.

システム	ジョブ数	全CPU時間	全入出力回数	経過時間
A	72	5420 秒	241567	12670 秒
C	97	10608	426907	18110

注 経過時間は1回の入出力を30ms としたときの1多重時間

$$26\times1.3+21\times2+53\times1.4=150\text{パーセント}$$

となる。A，B，C システムの全入出力回数の比は

$$A:B:C=1:1.2:2.1$$

となり，C システムの入出力パス数を6とすればA，B システムの入出力パスの数は，それぞれ3，4となる。

以上の議論からFig．6.1 の疎結合計算機システムでは並行して動作する13本以上の入出力パス数が必要であることがわかる。

6.2 ベクトル計算機を含む複合システムの入出力パス数

Fig．6.1 のA システムを4.2.3節で述べたベクトル計算機2台V1 ，V2 で置き換えたシステムを考える（Fig．6.4）。

5.3節で述べたように，1回の入出力時間が10ミリ秒以下の入出力バッファ装置を各ベクトル計算機は備えているとする。そうすると5.3節と6.1節の議論から，Fig．6.4 において各計算機が最低限必要とする入出力パスの数は，B，C，V1，V2 についてそれぞれ4，6，3，3となる。ただしV1，V2ではそれぞれ入出力バッファ装置を経由したパスを想定している。

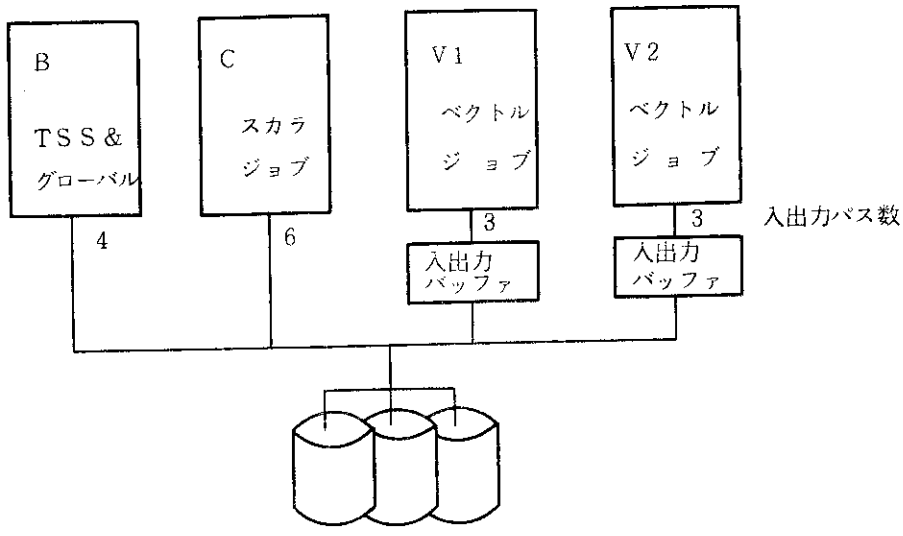


Fig. 6.4 Necessary number of concurrent I/O paths in a proposed computer configuration.

6.3 ベクトル化ジョブの回転時間（ターン・アラウンド・タイム）と新オンライン・ネットワーク

5.3節ではひとつのベクトル計算機内の多重プログラミング環境下にある高いベクトル化率のジョブが低いベクトル率のあるいはスカラ・ジョブにCPU時間を奪われないためには，入出力バッファ装置とベクトル化ジョブを意識した新スケジューリング機能が必要であることを述べた。

この節では高いベクトル化率を持つジョブの回転時間を短縮するための方策について議論する。Fig. 6.5 は6.1節と同じAシステムのジョブ・パターンの各ジョブについてそのCPU時間を3分の1にし，可変多重度で処理を模擬したもので，そのCPU時間の総和と全経過時間はVECTOR値（あるいはα値）が5のベクトル計算機によるものに近くなっている。Fig. 6.5のタイトルでWITH INPUT ACCEL. とあるのは各ジョブの入力時刻を早めて処理をしたことを示す。即ち最初のジョブ到着時刻t_1をゼロ時刻としたとき，第i番目のジョブの到着時刻（$t_i - t_1$）を（$t_i - t_1$）÷3と短縮して処理している。

Fig. 6.6は同じジョブ・パターンでCPU性能が5倍と仮定したときのものである。これはVECTOR値（あるいは2.3節のα値）が10のベクトル計算機と近似的に等しい。

Fig. 6.5, 6.6からわかるように，演算性能が上がればそれに倍してジョブの入力がおこなわれなければCPUに遊びができてしまう。これまでのジョブ統計によれば，利用者が各計算コードのCPU時間を長くしたり，短縮されたジョブ回転時間を活用して多くのジョブを投入することにより，CPU性能向上の1～2年後には導入された計算機は処理能力の上限に達する。

ベクトル計算機を導入したときは，この1～2年間の運用が非常に難しい。何故なら高いベクトル化率を持つジョブは，ベクトル化への努力が効を奏してCPU時間が短縮されベクトル計算機導入初期の段階においては比較的短い経過時間で処理されるが，そのジョブが再びシステムへ投入されるまでの時間は従来と変りがない。Fig. 6.6 から想像できるとおり，この間ベクトル計算機に遊びができるから運用担当者としてはスカラ・ジョブを投入することになる。

そして利用者がスカラ・ジョブのCPU時間を安易に長くすることはベクトル・ジョブのCPU時間を長くすることは同じではない（α値だけの差がある）。ジョブのベクトル化率を高くすることはかなり手間がかかる。ところがベクトル・ジョブの利用者が，そのジョブのベクトル化率を高める努力をすればするほど，スカラ・ジョブの利用者は労せずしてCPU時間を手に入れ，またジョブの回転時間の短縮をはかることができる。その結果ベクトル・ジョブの回転時間はさほど短縮されず，利用者のベクトル化への意欲は大いに減退することは間違いない。そこでベクトル化ジョブにはジョブ回転時間の短縮を保障する計算機構成，装置，運用方法が必要である。現在の利用者と計算機系はFig. 6.7 のように表わされる。ここでジョブ回転時間の短縮を阻害する要因となっているのは，(i)出力データ処理，(ii)TSSによる計算コード，データ修正と入力，(iii)ジョブ混雑による実行までの待ち時間の3つである。

従来我々がタイムシェアリング処理でおこなっている操作は上記(ii)の入力処理のみである。出力データ処理のうち，表示のみについていえばTektronix T4014グラフィック端末，FACOM F9611R チャネル直結型キャラクタ端末などを使って小さな失敗を繰返してきた（41）。前者の端末は通信速度に，後者の端末は計算機に直結して使用しなければならない点に問題があった。その上どちらもインテリジェント端末ではないのでローカルなデータ処理はできない。

ところがこの2～3年の間にインテリジェント端末と高速構内通信回線を組み合せたローカル・エリア・ネットワーク（LAN）の技術が急進展し，この問題に再度挑戦することが可能となった。原研計算センタでもこの出力データ処理まで含め，高速ネットワークについて調査，

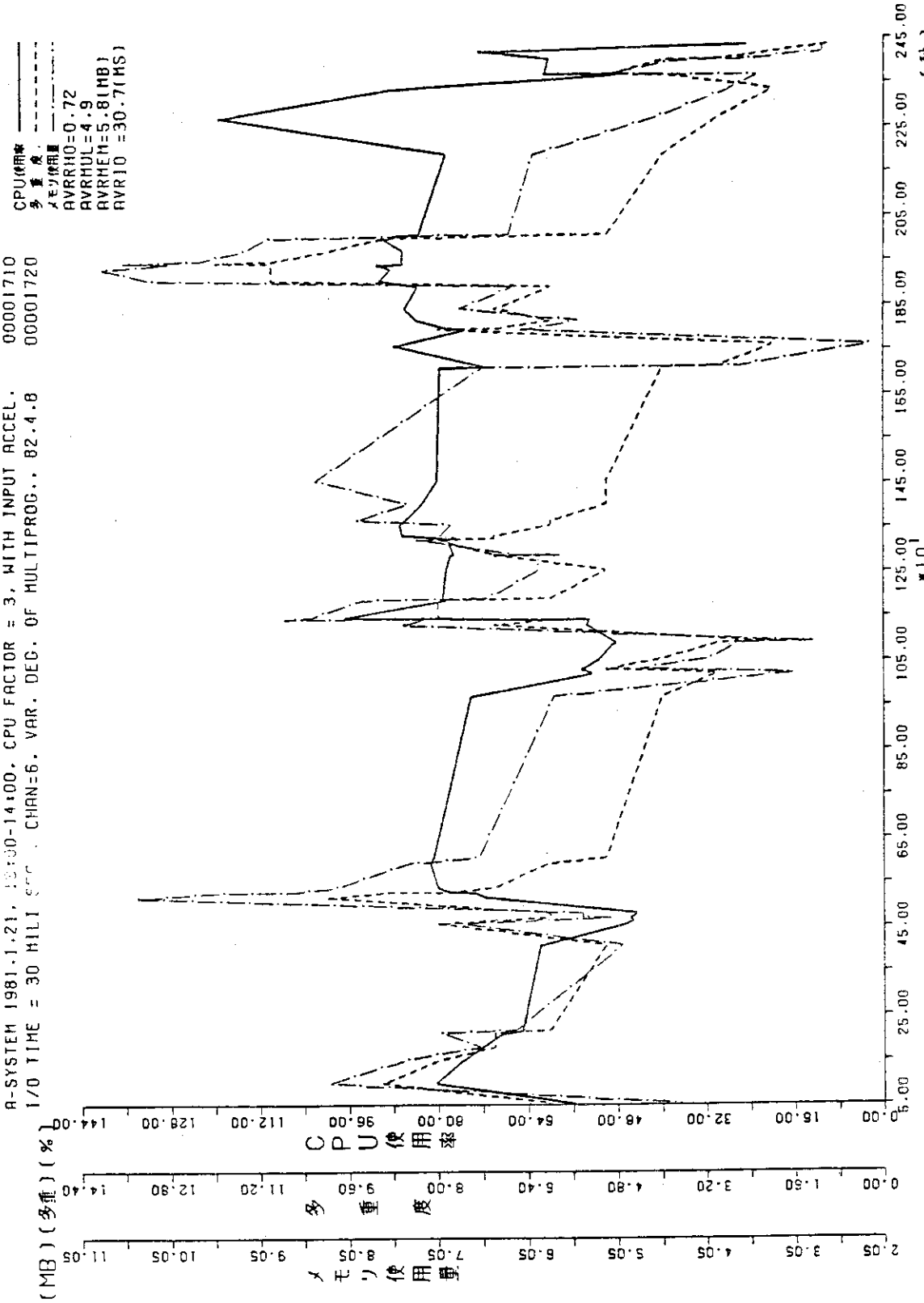


Fig. 6.5 Simulation of job processing of day time A computer system with 3 times faster CPU and job arrival intervals(in case of VECTOR=5.0).

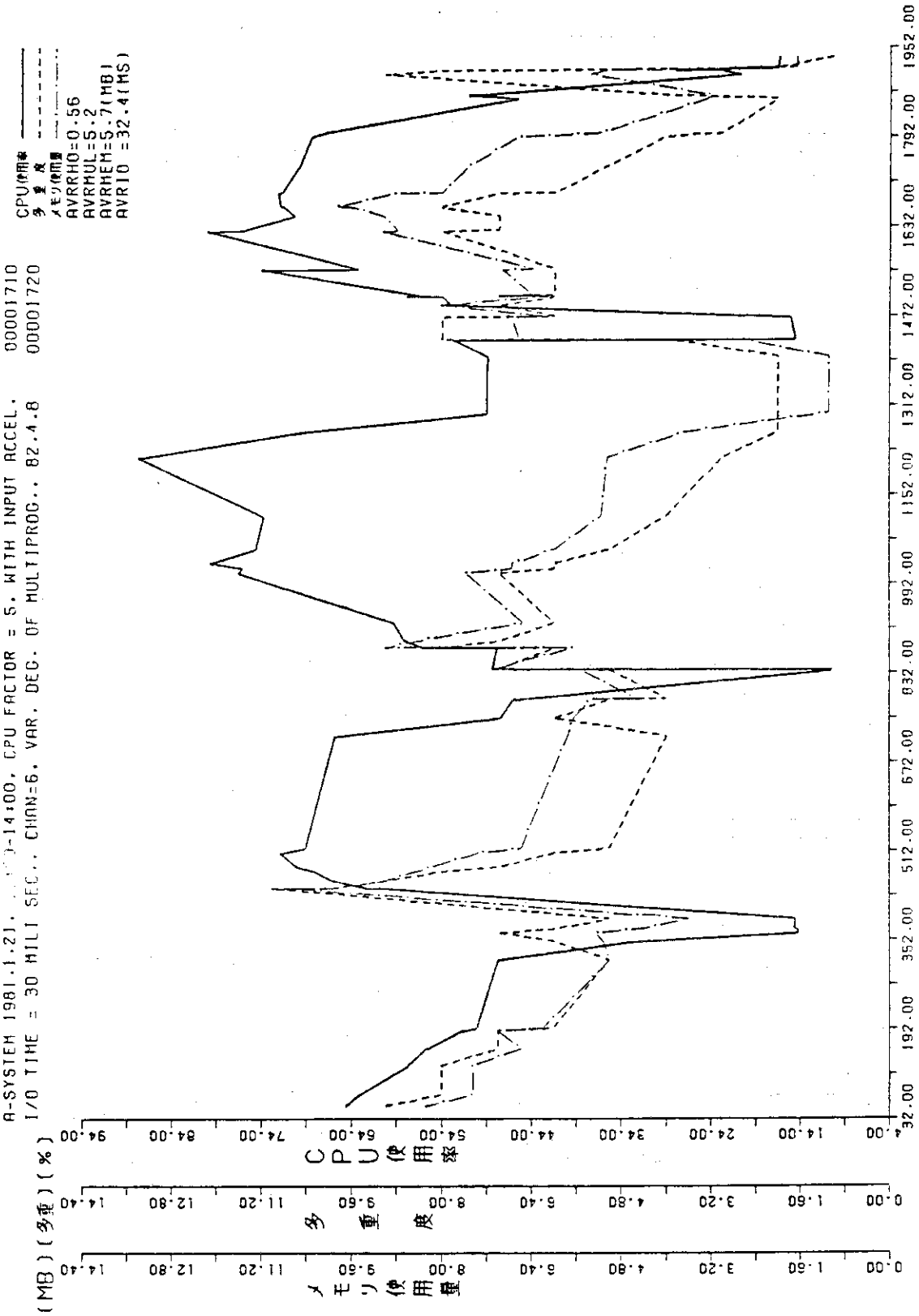


Fig. 6.6 Simulation of job processing of day time A computer system with 5 times faster CPU and job arrival intervals(in case of VECTOR=10.0).

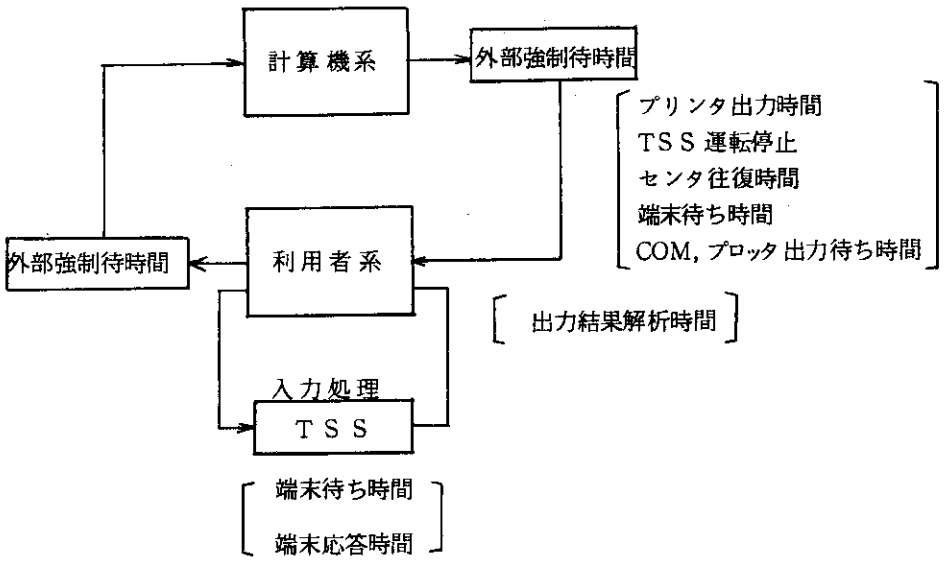


Fig. 6.7 Current environment of computer use and user waiting time for job input/output operation.

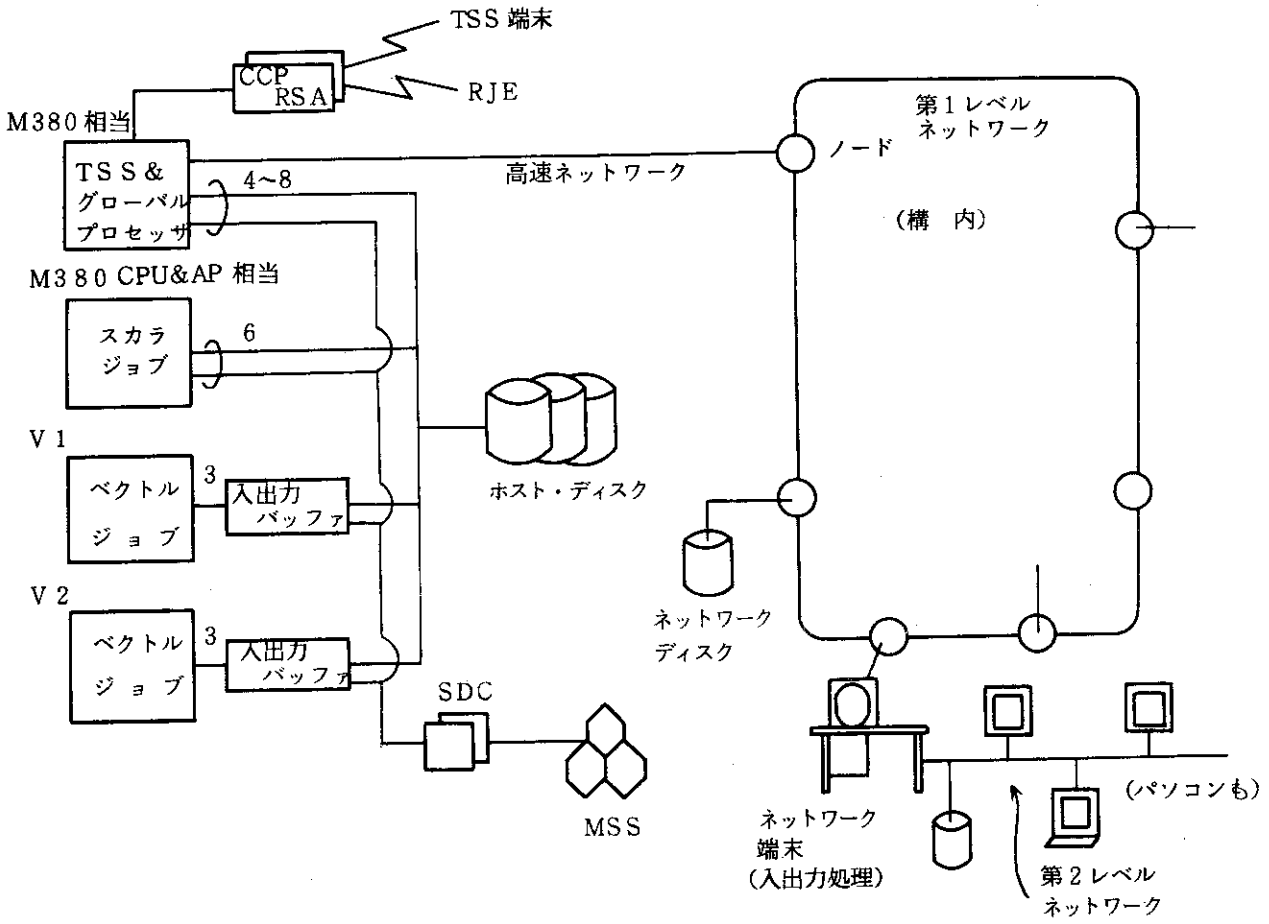


Fig. 6.8 A proposed computer complex for efficient vector and scalar job processing.

検討した(44)。その結果技術的には実現可能な見通しである。

その概略はFig. 6.8 のようになる。

このネットワークでは入出力ファイルは各ノードへ高速で転送される。出力の表示，編集，保存等はノード，あるいは表示端末でおこなわれる。一部修正した入出力ファイルを大型機へ返送することも容易となる。

また，高いベクトル化率のジョブを優先処理することによって上記(iii)のバッチ・ジョブの実行待ち時間を短縮する必要がある。

ベクトル・ジョブとスカラ・ジョブを同一の規準で取扱っていたのでは，計算コードのベクトル化を促進することはできない。この新オンライン・ネットワークのような装置をベクトル・ジョブの利用者に提供し，それによってベクトル・ジョブの回転時間の短縮をはかることが必要である。

Fig. 6.8 においてＴＳＳ専用Ｂシステムの入出力パス数が４～８となっているのは，バッチとグローバル用入出力パス数４に加えて，さらに４本程度の入出力パスを必要とするという意味である。

ここではバッチ出力のノードへのファイル高速転送およびノードからのファイルの部分転送量をバッチ入出力量と同程度見込んでいる。

このネットワークのハードウェアは既に市場に出ている製品を寄せ集めて間に合わせることもできるが，オペレイティング・システムがらみのソフトウェアと各ノード，端末のソフトウェアは新規開発要素を多く含み，さらに使用経験の蓄積も必要である。今後はこの方向でも研究開発をおこなう必要があり，現在少しずつ準備を進めている段階である。

5.3節およびこの６章で提案している計算機の機能は，「ベクトル化率が高いジョブはますます，スカラ・ジョブはそれなりに，サービスを受ける計算機システムの構成」をはかるためのものである。

7. おわりに

7.1 使用したデータとその解釈

データ数が少数であるにもかかわらず大胆な解釈をした箇所が本文中に2，3ある。その根拠について述べておこう。

(1) ソフトウェア生産性のうち「問題分析」の生産性について

解くべき問題に適合した数値計算法を試行錯誤的に発見し，それに合せてプログラムを修正する作業は，筆者の定義では，ソフトウェア（科学技術計算プログラム）生産過程における「問題分析」である(39)。 3.3.2.2節のFig. 3.6を作成した時点では，問題分析作業と見られるデータは1～2のみであった。当時データは集めていなかったが，それまでの経験に照して敢えてFig. 3.6に問題分析の直線を記載した。今回データを収集したベクトル化作業はまさに筆者が定義するところの問題分析である。両者のデータが一致するということは科学技術計算プログラムの開発過程において，このような定義と生産性の特徴を有するフェーズを考えてよいことを示唆している。

(2) M-200ユーザ・ジョブのパターンをM380 計算機に適用したことについて

計算機の演算性能が上がるとそれと同じ比率でジョブのCPU時間が増加することは次のような条件で納得できる。

(a) 利用者は常に潜在需要をかかえており，課金制度を採用していない運用体制では1分，5分，15分などのジョブ・クラス設定が計算機利用上の主たる目安とならざるを得ない。

(b) この10年間技術革新が激しく，同一価格で数倍の性能向上を期待できたので，運用担当者にとっては1分，5分，15分などの枠を変更する必要はなかった。

利用者にとっても運用担当者にとっても計算機利用の単位としての1分，5分などの時間の感覚はいつの時代も変っていないことがわかる。ここで問題なのはジョブ平均CPU時間の意味である。5.2.1節 Table 5.2のうちCPU時間が長い，例えばCPU時間10分以上のジョブの大部分は，もともとは長時間を要する計算を小刻みに実行しているものであることが経験的にわかっている。2分～10分のジョブはCPU性能が3倍となったときも同じCPU時間に留るであろうか。この点については分析不足で5.2.1節の仮定は問題を残していることを認めなければならない。2分以下のジョブについていえば，これも経験的な話で恐縮であるが，ジョブ回転数が多くなる方向にゆくようである。ジョブ統計からみてもそういえる。

(3) 作業用メモリ量の推定について

5.3節では2，3のデータから大容量の作業用メモリを必要と断定している。これは乱暴な結論のように見えるが，6年前に原子力コードのベクトル計算処理適応性を調査し始めたときから，この程度の量の作業用メモリが必要な場合にしばしば出会っている。当時において73MB 程度のメモリ量を要求することは直ちにベクトル計算機の実用性，現実性の否定，

ベクトル計算処理の否定につながった。この程度のメモリ量が各分野のベクトル計算処理で必要なことは，最近発表されているスーパーコンピュータが大容量メモリを備えていることでもわかる。

(4) シミュレーション・プログラムの精度について

ＣＰＵ寄りのジョブが２～３多重で走るジョブ処理環境を模擬するとき誤差が出てくる。その程度と理由は付録３で述べよう。

(5) ベクトル計算機の費用効果比について

これまでの議論で費用効果比には触れなかったが，現在は未だコードのベクトル化にかなり心理的抵抗もあり，(i) スカラ・ジョブを簡単にベクトル・ジョブに変更できないことが多い，(ii) したがって放置していたのではベクトル化は進展しない，ことからベクトル化工数も必要である。従来運用担当者の計算機交換のひとつの目安は２倍以上のＣＰＵ性能の向上であったことから考えると運用担当者としてはベクトル計算機の導入によって費用効果比が２倍以上になることが望ましい。これからα（VECTOR）値が５，あるいは１０のベクトル計算機をわれわれのデータにあてはめて，その値段を推定してみると，ＣＰＵ使用量の７０パーセントについて，$\alpha=5$のときスカラの３倍の性能であるから，ベクトルとスカラ計算機の価格比をx とすると，

$$\frac{0.7\times3+0.3\times1}{x}=2, \qquad \therefore \quad x=1.2 \ ,$$

$\alpha=10$のときスカラの５倍の性能であるから

$$\frac{0.7\times5+0.3\times1}{x}=2, \qquad \therefore \quad x=1.9$$

となる。したがって前者の場合はスカラ計算機の２割増，後者では９割増の価格で，２倍の価格性能比となる。もし計算機メーカがこのような価格で発表しているなら，$\alpha=10$のときは専用機的使用法に徹しないとスカラ機２台のほうが価格性能比はよくなる。

(6) α（VECTOR）値について

2.3節式(2)のαの値がベクトル計算機のカタログ上の最高性能をα値と考えてはいけない。ハードウェアの性能はこのαよりもずっと高い場合が多い。2.3節Fig. 2.6 CRAY－1でもベクトル長が３２なら５倍の性能である。このようにベクトル長が短いとか，間接アドレスが多いのがわれわれの原子力コード利用の現状である。

(7) 計算コードの傾向分析について

調査期間が長かったので，生き残ってよく使用されているコードはTable 3.1 に入っている。そのほか，長期間継続して使用されるTRITON （核融合），SRAC（炉工学），環境放射能予測計算（環境安全性）などのコード・システムの長時間コードもTable 3.1 に入っているので調査結果は一応原研使用コードの傾向分析になっているといえよう。

(8) ベクトル化率の分布について

ベクトル化率の分布が3.3.2.1節Fig. 3.4 のように高いベクトル化率と低いベクトル化率の両極端に分かれ，中間が少なくなった理由として筆者は次のように考えている。

「調査対象とした計算コードは，いずれも当時，あるいは現在の大型計算機で数分以上の

計算時間を要したものばかりである。計算時間の長い科学技術計算コードは，時間的，空間的に並行して生起する自然現象を模擬しており，並列計算処理の対象となる。したがってベクトル計算処理にも向くものが多い。RELAP 4，RELAP 5，KENO 4コードなどは現在の物理現象のモデル化，数値計算法がベクトル計算処理に合わない。セル内粒子法を使って物質の移流，拡散を解くADPICは，広い空間内に移流，拡散が起るのは相当の長時間を経過した後であり，それまでは，ごく少数のメッシュと格子が関係するに過ぎないのでベクトル化率はそれほど高くならず50~60パーセントのベクトル化率である。セル内粒子法を使う計算コードは原研には他にもあるが，いずれも全CPU時間に占める割合は小さい。上述のような理由から計算コードはベクトル化できるか，あるいは現在のところ殆んどベクトル化できないかの両極端に分かれる傾向がある。Fig. 3.4 の度数分布はそれを示している。

7.2 原研におけるベクトル計算処理の今後の方向

これまでの調査と研究の結果，原子力コードのベクトル計算処理に明るい見通しを得て，ベクトル計算機を含む複合計算機システムの当面（ここ3~4年）の構成まで考えることができるようになった。今後は次の項目について調査と研究開発を推進し次の段階の計算機構成法に備える。

(1) ベクトル化不適合コードの取扱い

現在ベクトル化が不適と見られているRELAP系列，KENO 4，ANISNなどの数値計算法の研究を進め，これらコードのベクトル化をはかる。RELAP系については既に研究者主導型の作業グループ設立が予定されている。

(2) 事例研究とベクトル化技術の普及

今後もベクトル化事例を多くし，そこで得られた知識を普及させて研究者が新規，あるいは既存コードのベクトル化を容易におこなえるようにする。

(3) ジョブ・スケジューリング，新ネットワークの試行と実用性の評価

上記機能の開発試行をおこないベクトル計算機有効稼動との関連を明確にし，その実用性を評価，検討する。

謝　　辞

原研と富士通（株）で昭和５５年１０月から５７年１２月まで実施した並列処理に関する共同研究は，原研側は計算センタを主担当個所とし，核融合研究部，物理部，原子炉工学部，環境安全研究部の参加を得て，原研側は約２２名の研究者，技術者，富士通側は研究，打合せの各テーマに応じてテーマ毎に延べ２２名の技術者が参加した。この期間１０回の会合を開き，研究内容について討議，報告，打合せをおこなった。参加者の氏名，所属は次のとおりである（以下いずれも職位，敬称略）。

原研側参加者（責任者は平川隆計算センタ室長）

中原康明，西田雄彦，田次邑吉，筒井恒夫，藤村統一郎，堀上邦彦（現原子力データセンタ）（以上原子炉工学部）

五十嵐信一，別役広，佐々木健（物理部），竹田辰興，常松俊秀（核融合研究部），茅野政道，石川裕彦，林隆（環境安全研究部），平川隆，浅井清，石黒美佐子，中村康弘，藤井実，原田裕夫，樋口健二，小沼吉男（計算センタ）

富士通側参加者（責任者は岡本彬第３システム統轄部長）

岡本彬，鈴木滋，三輪修，槌本隆光，内田啓一郎，棚倉由行，三浦謙一，田子精男，松浦俊彦，田原伸夫，南多善，原口誠，長谷部芳郎，田中幸夫，奈良岡賢逸，奥田基，海谷真，通口吾二，篠沢尚久，安達政夫，高橋國夫，太田文雄

また，共同研究そのものではないが，それに関連して次表の方々にお世話になった。表の項目は引用文献には現われないのでここに記した。

このほか，原子力コード研究委員会総合化専門部会長（当時）柱木学安全工学部長にはVENTUREコードのCRAY－1によるベクトル化ベンチマーク・テストでお世話になった。また同委員会開発整備専門部会長朝岡卓見原子炉工学部長には，並列処理の原子力コードへの適応性問題を同専門部会のテーマとして取り上げて戴き，原子力コードのベクトル化方法につき種々計算センタがベクトル計算機の原子力コードへの適用可能性の検討調査を始めて以来，原研担当の富士通システム・エンジニアには縁の下の力持ち的作業を努めてもらっている。

原子力コードのベクトル計算処理について明るい見通しと計算機構成の案を得て共同研究を終結することができるのも，上記の方々の尽力によるもので，ここに深い感謝の意を表します。

本報告の原稿を5～6名の方々に読んで戴いたが，特に松浦俊彦，石黒美佐子両氏からは記述の不充分，不明瞭な点を数多く指摘していただき，その結果内容がかなりわかりやすくなった。合せて感謝します。

項　　目	富　士　通	原　　研
FACOM 230-75APU ガイダンス	鈴木滋，三輪修 内田敬一郎，棚倉由行， 磯辺文雄	
TWOTRAN, RELAP 3B, EDDYTORUSのベクトル化	田子精男*，南多善*， 熊倉利昌*，長谷部芳郎， 奈良岡賢逸*	計算センタ
巨大ジョブ7本のベクトル化可能性調査	同　　上	藤井実 （計算センタ）
本報告の各種ジョブ統計データ		藤井実，中村康弘，樋口健二 （計算センタ）
本報告シミュレーション結果の図形出力		小沼吉男（計算センタ）
本報告ジョブ・パターンの図形出力	木原和久*	
本報告付録2の原子力コード開発，変換に要する工数の調査	石川雅章*	田坂完二（安全工学部） 小林健介（安全解析部）

注，*印は富士通派遣の計算センタ外来研究員（当時）。

引　用　文　献

1) 「並列計算機PACS－32によるBWR 炉心計算」，星野，　日本原子力学会誌，Vol．23，　№.8，　1981.

2) 「On the feasibility of using parallel microprocessors for calculation of 3-dimensional rating distributions in operating reactors」，Honeck，H.C.，NEACRP Specialist's meeting on calculation of 3-dimensional rating distributions in operating reactors，OECD，Paris，　1979.

3) 「Special Issue：Parallel Processors and Processing」，ACM Computing Surveys，　Vol．9，　№.1，　March 1977.

4) 「Large Computer Systems and New Architectures」，Bloch，T.，Computer Physics Communications，　Vol．26，　1982.

5) 「CRAY－1 Evaluation-Final Rcport」，　Keller，T. W.，LA－6456－MS，LASL，　Dec．1976.

6) 「FACOM 230－75 アレイプロセッサシステム」，三輪，久米，内田，鈴木，溯倉，磯辺，FUJITSU，　Vol．29，　№.1，　1978.

7) 「HITAC M200H機能説明書　8870／2／002」，日立製作所，Aug．　1978.

8) 「拡散方程式の並列計算」，石黒，古志，　JAERI－M9235，Dec．　1980.

9) 「線形三重対角方程式システムの並列計算」，石黒，原田，難波，藤井，藤村，中村，JAERI－M9703，　Sept．1981.

10) 「線形三重対角システムの並列計算」，原田，名古屋大学数値解析研究会，May 1981.

11) 「FACOM230－75APU を用いた核融合MHDコードのベクトル処理」，松浦，田中，樋口，竹田，常松，安積，徳田，栗田，滝塚，May　1981.

12) 「Vector processing efficiency of plasma MHD codes by use of the FACOM230－75APU」，　松浦，田中，奈良岡，滝塚，常松，徳田，安積，栗田，竹田，Conf．on Vector and Parallel Processing in Computational Science，Chester College，UK，Aug．1981.

13) 「Vector processing efficiency of plasma MHD codes using FACOM230－75 APU」，松浦，FUJITSU Scientific and Technical Jour．，　Vol．17，№. 3，　1981.

14) 「最近のCRAY－1利用による成果－DOE リポートを中心に－」，石黒，所内資料，Feb．1981.

15) 「CRAY－1とFACOM230－75APU による拡散コードVENTURE のベクトル演算化」鎌田，角谷，原田，　JAERI－M82－019，　March 1982.

16) 「Vectorizations for solving the neutron diffusion equation」, 石黒, 古志, Nucl. Sci. Eng., Vol. 80, No. 2, March 1982.
17) 「PARALLEL：並列計算処理文献情報検索システム」, 中村, 原田, 小沼, 藤井, 石黒, 浅井, 所内資料, March 1982.
18) 「ベクトル計算処理の大型原子力コードへの適応性」, 石黒, 松浦, 奥田, 原田, 太田, 梅谷, JAERI-M82-018, March 1982.
19) 「ポアソン・ソルバーの並列処理」, 松浦, 田中, 奈良岡, 名古屋大学数値解析研究会, May 1982.
20) 「ベクトル計算処理の原子力コードへの適応性」, 石黒, 竹田, 松浦, 富士通サイエンティフィック・システム研究会, 57年度第1分科会, July 1982.
21) 「差分法のベクトル計算」, 石黒, 難波, 情報処理学会論文誌, Vol. 24, No. 1, Jan. 1983.
22) 「中性子輸送コードのベクトル計算処理」, 石黒, 筒井, JAERI-M82-199, Dec. 1982.
23) 「FACOM230-75APU による大型原子力コードのベクトル化」, 原田, 樋口, 石黒, 筒井, 藤井, JAERI-M レポート (予定).
24) 「Application of a hexagonal element scheme in the finite element method to three-dimensional diffusion problem of fast reactors」, 石黒, 樋口, Jour. Nuc. Sci. & Tech. (to be appeared).
25) 「放射性物質大気中移流・拡散・外部被曝線量計算コードのベクトル計算処理」, 浅井, 篠沢, 石川, 茅野, 林, JAERI-M 82-218, Dec. 1982.
26) 「α-Mn のエネルギー帯・磁気構造計算コードAF-ALPHAMN のベクトル化」, 松浦, 佐々木, JAERI-M レポート (予定),
27) 「核融合計算コードの高速処理」, 竹田, 常松, 松浦, JAERI-Mレポート (予定),
28) 「JT-60コード・ライブラリのマシン最適化」, 安達, 松浦, JAERI-Mレポート (予定)
29) 「スーパーコンピュータと原子力計算」, 浅井, 石黒, 松浦, 原子力学会誌 (予定)
30) 「The CRAY-1 Computer System」, Russell, R. M., Comm. of the ACM, Vol. 21, No. 1, Jan. 1978.
31) 「Vector programming tools and techniques」, Schneider, C. (Ed.) UCID-30158, LLL, Jan. 1975.
32) 「IAP (内蔵アレイプロセッサ) の設計評価」, 梅谷, 堀越, 計算機システムの解析と制御 15-7, (情報処理学会研究会報告), Feb. 1982.
33) 私信「FACOM230-75APU ベクトライザ使用手引書」, 棚倉, 磯辺, 神谷, May 1978.
34) 「Evaluating Computer Program Performance on the CRAY-1」, Rudsinski, L., ANL-79-9, ANL, Jan. 1979.
35) 「Benchmark-Versuche mit der CRAY-1」, Hertweck, F.,

Schneider, W., Schwenn, U., IPP R/31-1/170-6/184, Mai 1979.

36) 「統計入門」, 松下, 岩波全書, 1968.

37) 「確率論とその応用」, 近藤, 日科技連ライブラリ, 1965.

38) 「準数値算法／乱数」, Knuth, D. E., 渋谷訳, The Art of Computer Programming 第3分冊, サイエンス社, 1981.

39) 「科学技術計算におけるソフトウェアの生産性」, 浅井, 富山, 田子, 松浦, 第21回情報処理学会プログラミング・シンポジウム報告集, Jan。 1980.

40) 「計算機の最適取替間隔計算モデル」, 藤井, 浅井, JAERI-M9605, Aug. 1981.

41) 「Modular Programming Method at JAERI」, 浅井, 荏木 (Eds.), JAERI-1274, Feb. 1982.

42) 「計算機ジョブ処理最適多重度の決定」, 浅井, 高橋, 藤井, JAERI-M 9501, June 1981.

43) 「計算機群同時停止のためのバッチ・ジョブ・スケジューリング」, 浅井, 高橋, 藤井, 情報処理学会論文誌, Vol. 23, No. 5, Sept. 1982.

44) 「新オンライン・システムの設計」, 山田, 浅井, 次田, 西, 森, 上, 所内資料, Sept. 1982.

付録1　昭和57年7，8，9月の大口利用者リスト

1982. 7

FACOM OSIV/F4 P L A N N E R (V01-L03)

UNO	TOTCPU	AVRCPU	BCNT
797	7.8	13.8	34
1280	9.3	50.9	11
2347	18.8	12.1	93
2855	16.4	15.1	65
3019	6.3	8.0	47
3022	41.0	12.8	192
3051	22.6	5.4	251
3069	20.7	8.7	142
3117	5.3	29.4	11
3198	37.7	66.6	34
3244	75.6	39.4	115
3247	22.9	7.2	189
3267	24.1	40.1	36
3408	7.7	5.5	84
3483	0.9	5.2	11
3491	20.6	5.2	238
3506	13.9	13.4	62
3520	13.7	10.5	78
3590	36.8	9.1	241
3610	34.6	13.5	153
3762	10.2	11.8	52
3766	7.9	9.5	50
8255	0.6	9.6	4
9024	34.5	13.4	154
9246	14.7	11.0	80
9249	6.7	5.1	78
9282	4.4	7.1	37
9289	9.9	9.0	66
9316	23.4	18.2	77
9319	39.0	8.2	285
9321	1.4	5.3	16

1982. 8

FACOM OSIV/F4 P L A N N E R (V01-L03)

UNO	TOTCPU	AVRCPU	BCNT
253	28.5	5.7	296
1446	3.5	11.6	18
1470	6.6	6.9	57
1476	8.6	5.5	94
1835	7.5	11.8	38
2129	5.6	8.9	38
2347	13.3	9.6	83
2382	1.7	6.5	16
2678	22.3	9.1	147
2747	5.6	8.1	42
2855	22.3	7.9	168
2935	5.9	5.9	60
3008	3.0	30.0	6
3022	14.2	8.1	104
3051	43.6	9.1	285
3069	19.2	7.0	165
3073	2.8	7.9	22
3100	1.0	21.3	3
3117	17.8	30.5	35
3167	21.9	20.2	65
3196	23.1	6.0	228
3198	9.3	56.2	10
3224	2.0	15.2	8
3247	14.6	7.4	118
3375	25.4	15.5	96
3427	3.5	8.8	24
3481	21.4	10.8	119
3491	14.1	19.8	43
3631	8.3	13.4	37
3786	33.3	21.2	94
3809	3.0	6.5	28
8260	16.5	5.1	191
9024	11.3	9.1	74
9246	14.0	5.3	156
9269	5.2	5.2	60
9280	16.6	8.9	112
9289	31.7	11.0	172
9313	18.1	15.5	70
9336	24.5	24.5	60
9506	10.3	13.5	46
9834	2.0	8.7	14

1982. 9

FACOM OSIV/F4 P L A N N E R (V01-L03)

UNO	TOTCPU	AVRCPU	BCNT
797	23.1	33.0	42
1280	27.1	95.7	17
1446	22.5	10.5	[illegible]
1476	7.0	5.3	79
2146	25.1	5.9	252
2241	3.5	5.4	39
2347	16.6	7.0	142
2678	39.4	12.4	190
2855	14.5	13.6	64
3009	3.5	5.5	38
3019	12.1	30.3	24
3022	15.4	14.2	65
3023	1.6	14.4	7
3051	16.1	7.3	133
3069	30.2	18.3	99
3116	22.2	15.7	85
3117	22.8	34.2	40
3155	7.5	64.4	7
3198	22.1	73.7	18
3244	57.6	7.3	438
3340	0.4	6.0	4
3352	6.3	6.7	57
3408	12.1	11.2	55
3427	17.2	12.6	82
3481	7.7	13.2	35
3491	13.6	6.0	135
3506	3.7	44.8	5
3610	23.8	20.4	70
3813	11.8	7.0	101
3815	31.4	6.5	238
9246	12.5	6.1	122
9289	21.4	6.4	201
9313	8.1	7.4	65
9318	3.0	5.5	33
9319	40.5	8.4	288

付録2　工学的安全解析コードの開発，変換に要した工数一覧表

米国安全性解析コードの変換整備例

原研計算センター
56. 3. 6

コード名	コードの目的／変換・整備内容	開発工数（人月）	ソースコードライン数	変換・整備工数 期間	変換・整備工数 人月	依頼元部室名	備考
1. WREM コードシステム	米国原子力規制委員会から導入したECCS性能評価コードシステム。軽水炉の安全審査参考解析計算に用いる。						
1) RELAP 4/EM（IBM版）	軽水炉のブローダウン時の流動解析計算 ・F230−75 への単純変換 ・プロッタープログラムをCOMでも使用可能とする。		29,800	50.10.1~51.3.31 51.4.1 ~51.4.10	6.2 0.3	原子炉データ解析室 同 上	
2) RELAP 4/FLOOD	PWRの再冠水解析計算 ・F230−75 への単純変換					同 上	
3) MOXY−EM30（IBM版）	BWRの冷却材喪失事故時における燃料棒の熱的挙動を解析する。 ・F230−75 への単純変換		9,000	50.10.27~50.12.31	3	同 上	
4) TOODEE 2（IBM版）	PWRの冷却材喪失事故時における燃料棒の熱応答を計算する。 ・F230−75 への単純変換 ・プロッター用プログラムの作成，磁気テープ出力機能の追加 ・コードの一部修正		9,000	50.10.15~50.12.26 51. 1. 5~51. 2.28 51. 7.23~51. 7.27	2.4 2 0.2	同 上 同 上 同 上	
2. RELAP 4/MOD5（IBM版）	軽水炉の冷却材喪失事故解析コード プロッタールーチン ・F230−75 への単純変換		47,729 1,769	52. 4.29~52. 8.31	5.5	安全性コード開発室	
3. RELAP 4/MOD6（IBM版）	軽水炉の冷却材喪失事故解析コード ・F230−75 への単純変換 ・出力結果整理用ツールの作成		84,362	53. 9.26~54. 1.31 54.12.15~55. 2.15	11 2	同 上 原子炉データ解析室	

コード名	コードの目的／変換・整備内容	開発工数（人月）	ソースコードライン数	変換・整備工数 期間	変換・整備工数 人月	依頼元部室名	備考
4. RELAP 5／MOD0（CDC版）	軽水炉の冷却材喪失事故解析コード 1976年1月に開発を始め1979年4月に完成してリリース。物理モデル作成者2名，プログラミング担当者5名が従事 ・M200への単純変換	280	51,333	54. 9. 1~55. 6.15	16	安全工学第一研究室	
5. RELAP 5／MOD1（CDC版）	軽水炉の冷却材喪失事故解析コード 1.5年かけてMOD 0を改良したバージョン 物理モデル作成者2名，プログラミング担当者5名が従事 ・M200への単純変換	126	69,700	(予定) 56. 1. 6~56. 5.31	(予定) 10	安全工学第一研究室	
6. TRAC－P1（CDC版）	3次元非平衡流動モデルに基づいたPWRシステム解析コード，物理モデル作成に7人年，開発途中のコード検証に3人年，プログラミングその他に7人年 ・F230－75 への単純変換	204	33,000	53. 5.11~54. 2.28	14.5	安全性コード開発室	
7. RETRAN－01／MOD001（IBM版）	軽水炉一次系の過渡応答解析計算 ・F230－75 への単純変換 ・プロッター出力カルーチンの独立化 ・ソースとサンプルプログラムのMT作成		77,869	54. 5.23~54. 7.31 54. 8. 1~54. 8.31 54.12.14~54.12.25	7 2.5 0.4	原子炉データ解析室 同 上 同 上	
8. RETRAN－01／MOD002（IBM版）	軽水炉一次系の過渡応答解析計算 ・M200への単純変換 ・プロッター出力ルーチンの独立化 ・プリンタープロッター出力とREEDIT機能に関する修正等 ・テープ使用時におけるエラー・メッセージに関する修正 ・プロッタープログラムの改良，マルチボリウム処理の調査等		61,257	55.1.23~55.3.31 55. 5. 1~55. 5.31 55. 6. 9~55. 6.14 55. 9.11~55.10.11	3.8 1 0.5 1	原子炉データ解析室 同 上 同 上 同 上	
9. RETRAN－01／MOD003（IBM版）	軽水炉一次系の過渡応答解析計算 MOD002のマイナー・チェンジ・バージョン ・M200への単純変換		61,257	55.12. 5~55.12.26	0.7	原子炉データ解析室	

付録３　待ち行列理論によるシミュレーションの方法

この報告で使用したシミュレーション・プログラムの理論的根拠は文献（４２，４３）で記述した方法を改良したものである。この付録では改良した点についてのみ述べる。

1. 待ち行列理論によるモデル

模擬計算に使用した待ち行列理論に基づくモデルを図示するとFig．A のようになる。CPU系はポアソン分布の入力率，幾何分布の処理率，I／O 系はポアソン分布の入力率，指数分布の処理率を仮定している。ＣＰＵにおける処理はタイム・クワンタムQで動くラウンド・ロビンのサービス法採用している。これは過去の履歴が影響しないmemoryless のサービス分布（幾何分布）を与える。任意のジョブの組合せについて幾何分布が計算できるのは，入力される各ジョブのＣＰＵ時間と入出力（Ｉ／O）回数が予測できるという仮定に基づいている。この仮定がわれわれの計算センタでは成立する（文献４２，４３参照）。ＣＰＵ系にポアソン分布の入力率を仮定すると，ＣＰＵの処理がmemoryless なので，出力率は入力率と同じ平均値を持つポアソン分布になる（文献a．）。同様にＩ／O 系の処理もmemoryless の指数分布を仮定しているので，その出力率も入力率と同じ平均値を持つポアソン分布になる。したがってFig．A.の多重度Mのクローズド・ネットワークを入力と出力に関しては矛盾なく考えることができる。

2. 状態確率 $\{P_n\}$ の計算

ジョブ多重度をM，並行動作可能なＩ／O パスの数をＮとするとCPU 系における状態確率 $\{P_n\}$ は，

2.1　$M \leq N$のとき，定常状態を仮定し，

(1)　$\mu_1 P_1 = M P_0 \mu_2$ ，

(2)　$\mu_1 P_{n+1} + (M-n-1)\,\mu_2 P_{n-1} = \{\mu_1 + (M-n)\,\mu_2\} P_n$ ，

(3)　$\mu_1 P_M = \mu_2 P_{M-1}$ ，

(4)　$\sum_{0}^{M} P_n = 1$ 。

$\rho=\mu_2/\mu_1$ とすると，(1), (2), (3)式から

$$P_1=M\mu_2P_0/\mu_1=M\rho P_0\ ,$$

$$P_2=P_1+(M-1)\rho P_1-M\rho P_0=(M-1)\rho\cdot M\rho P_0=M(M-1)\rho^2P_0\ ,$$

$$\vdots$$

$$P_n=M(M-1)\cdots\cdots(M-n+1)\rho^nP_0\ ,$$

$$\vdots$$

$$P_M=\rho P_{M-1}=\rho M(M-1)\cdots\cdots(M-\overline{M-2})\rho^{M-1}P_0=M!\rho^MP_0\quad 。$$

また(4)式から

$$\sum_0^M P_n=\sum_0^M\frac{M!\rho^n}{(M-n)!}P_0=1\ ,$$

$$\therefore\ P_0=\frac{1}{\sum_0^M\frac{M!}{(M-n)!}\rho^n}\quad 。$$

このP_0 を使って $\{P_n\}$ を求めることができる。

2.2　M＞Nのとき

(1)　$\mu_1P_1=N\mu_2P_0$

(2)　$\mu_1P_{n+1}+N\mu_2P_{n-1}=\{\mu_1+N\mu_2\}P_n$，$M-\overline{n-1}>N$ のとき，i.e.，$n\leq M-N+1$，

(2′)　$\mu_1P_{n+1}+(M-\overline{n-1})\mu_2P_{n-1}=\{\mu_1+(M-n)\mu_2\}P_n$，

これは$M-\overline{n-1}<N$のとき，i.e.，　$n>M-N+1$，

(3)　$\mu_1P_M=\mu_2P_{M-1}$，

(4)　$\sum_0^M P_n=1$　。

$0\leq n\leq M-N+1$のときは(1), (2)式から

$$P_1=N\mu_2P_0/\mu_1=N\rho P_0\ ,$$

$$P_2=P_1+N\rho P_1-N\rho P_0=(N\rho)^2P_0\ ,$$

$$\vdots$$

$$P_n=(N\rho)^nP_0\ ,$$

$$\vdots$$

$$\vdots$$

$$P_{M-N+1} = (N\rho)^{M-N+1} P_0 \quad 。$$

$M-N+1<n \leq M$ のときは(2'), (3)式から

$$P_{M-N+2} = P_{M-N+1} + \rho\,(M-n)\,P_{M-N+1} - (M-\overline{n-1})\,P_{M-N}\,\rho$$

$$= P_{M-N+1} + (M-\overline{M-N+1})\,P_{M-N+1}\cdot\rho - (M-\overline{M-N})\,\rho P_{M-N}$$

$$= \{\,(N\rho)^{M-N+1} + (N-1)\,\rho\,(N\rho)^{M-N+1} - (N\rho)(N\rho)^{M-N}$$

$$= (N-1)\,\rho\,(N\rho)^{M-N+1} P_0 \quad ,$$

$$\vdots$$

$$P_{M-\overline{N-K}} = (N-1)\cdots\cdots(N-\overline{K-1})\,\rho^{K-1}\,(N\rho)^{M-N+1} P_0$$

$$\vdots$$

$$P_M = (N-1)\cdots\cdots(2)(1)\,\rho^{N-1}\,(N\rho)^{M-N+1} P_0 \quad 。$$

(4)式を使うと

$$1 = \sum_0^M P_n = \sum_0^{M-N+1} P_n + \sum_{M-N+2}^{M} P_n$$

$$= \Big[\sum_0^{M-N+1} (N\rho)^n + (N\rho)^{M-N+1} \sum_{K=2}^{N} \rho^{K-1} \frac{(N-1)!}{(N-K)!}\Big] P_0 \quad ,$$

$$\therefore\ P_{M-N-K} = (N-1)\cdots\cdots(N-\overline{K-1})\,\rho^{K-1}\,(N\rho)^{M-N+1} P_0$$

$$= \frac{(N-1)!}{(N-K)!}\,\rho^{K-1}\,(N\rho)^{M-N+1} P_0 \quad , \quad K=2, \cdots, N \quad 。$$

3. CPU, I／O の処理率，待ち行列個数，平均待ち時間

(1) CPUの処理率を μ_1 とすると，ラウンド・ロビン法の定義から

$$\frac{1}{\mu_1} = \sum_{i=1}^{\infty} i\, g_i Q = \sum_0^{\infty} i\, \sigma^{i-1}\,(1-\sigma)\,Q = \frac{Q}{1-\sigma} \quad ,$$

ここでQはタイム・クワンタムである。

(2) μ_2 は図Aのように入力で固定時間を与える（模擬計算では特にことわらない限りは３０ミリ秒としている）。

(3) CPU系での待ち行列個数（平均）$\bar{n}_1$ は

$$\bar{n}_1 = \sum_0^M n P_n$$

で求める。このとき I／O 系での平均待ち行列個数は

$$\bar{n}_2 = M - \bar{n}_1$$

である。

(4) I／O系系での待ち行列個数L_qはＣＰＵと異なり，サービス窓口が複数であるために次のようになる（文献b）。

$$L_q = \begin{cases} \sum_{n=N+1}^{M} (n-N) q_n , & n > N \text{ のとき,} \\ 0 , & n \leq N \text{ のとき。} \end{cases}$$

(5) Ｋクワンタを要するジョブのＣＰＵ系の平均待ち時間$W_1(K)$は

$$W_1(K) = W_1(1) + \frac{(K-1)Q}{1-\rho_1} + Q\left[\lambda W_1(1) + \sigma \bar{n}_1 - \frac{\rho_1}{1-\rho_1}\right] \frac{1-(\lambda Q+\sigma)^{K-1}}{1-(\lambda Q+\sigma)} ,$$

$$(K \geq 1)$$

で表われる（文献c）。

ここで$E(Q_r)$はＣＰＵ系に新しい入力があったとき，ＣＰＵでもしジョブが処理されていれば，そのジョブが終了するまでの平均時間で$E(Q_r) = \rho_1 Q/2$（文献c）。

$W_1(1)$は，入力が始めてCPU を通り抜けるときの待時間で，

$$W_1(1) = E(Q_r) + (\bar{n}_1 - \rho_1 - 1) Q + Q$$

と表わされる。文献 c. では$\bar{n}_1 - \rho_1 - 1$は$\bar{n}_1 - \rho_1$となっているが，これはＣＰＵ系に既に$\bar{n}_1$個のジョブが存在するとき，新しく系に到着したジョブが初めてＣＰＵ系を通り抜けるまでの時間を示している。ところが現在のわれわれの系では新しく到着したジョブも$\bar{n}_1$のなかに個数として勘定に入っているので，上記のように$(\bar{n}_1 - \rho_1 - 1)$として$W_1(1)$を計算する。

(6) Ｉ／Ｏ 系の平均待ち時間W_2は

$$W_2 = W_q + \frac{1}{\mu_2} = W_q + c$$

として計算する。ここでcは３０ms（ミリ秒）として入力で与えているが，入出力バッファ（5.2.2節）の計算では１０ms，２０msの場合もある。W_qは(4)のL_qとLittleの法則$L_q = \lambda W_q$から$W_q = L_q / \lambda$として求める。

L_qの計算に必要なＩ／Ｏ 側の状態確率$\{q_n\}$は状態方程式を解いて求める。

4. 状態確率$\{q_n\}$の計算

4.1 $M \leq N$のとき

(1) $\sum_{0}^{M} q_n = 1$,

(2) $\mu_1 q_0 = \mu_2 q_1$,

(3) $\mu_1 q_{n-1} + \overline{n+1}\,\mu_2 q_{n+1} = (\mu_1 + n\mu_2) q_n$,

(4) $\mu_1 q_{M-1} = M\mu_2 q_M$ 。

$q_1 = \mu_1 q_0 / \mu_2 = \xi q_0$,

$q_2 = \mu_1 q_1 / (2\mu_2) = \xi^2 q_0 / 2$,

$\vdots$

$q_n = \xi^n q_0 / n!$ 。

$$\sum_0^M \frac{1}{n!}\, \xi^n q_0 = 1,$$

$$q_0 = 1 / \left(\sum_0^M \xi^i / i!\right)$$

$$q_n = \frac{1}{n!}\, \xi^n \frac{1}{\sum_0^M \xi^i / i!}$$

4.2 $M > N$ のとき

(1) $\Sigma q_n = 1$

(2) $\mu_1 q_0 = \mu_2 q_1$

(3) $\mu_1 q_{n-1} + \mu_2 \overline{n+1} q_{n+1} = (\mu_1 + n\mu_2) q_n$, $n+1 \leq N$,

(3') $\mu_1 q_{n-1} + N\mu_2 q_{n+1} = (\mu_1 + N\mu_2) q_n$, $n+1 > N$,

(4) $\mu_1 q_{M-1} = N\mu_2 q_M$ 。

$$q_n = \frac{1}{n!}\, \xi^n q_0 \quad , \quad n \leq N$$

$\mu_1 q_{n-1} + N\mu_2 q_{n+1} = (\mu_1 + N\mu_2) q_n$, $n > N$,

$$q_{n+1} = \frac{1}{N}\, \xi q_n \quad , \quad n > N \quad ,$$

$$q_{N+i} = \left(\frac{1}{N}\right)^i \xi^i q_N \quad , \quad i = 1, \cdots\cdots, M-N \quad ,$$

$$= \left(\frac{1}{N}\right)^i \xi^i \frac{\xi^N}{N!}\, q_0$$

$$= \frac{1}{N^i}\, \frac{1}{N!}\, \xi^{N+i} q_0 \quad ,$$

$$q_M = \left(\frac{1}{N}\right)^{M-N} \frac{\xi^N}{N!} q_0$$

$$\sum_0^M q_n = \left(\sum_0^N \xi^n \frac{1}{n!} + \sum_1^{M-N} \left(\frac{1}{N}\right)^i \xi^i \frac{\xi^N}{N!}\right) q_0 = 1$$

からq_0を，これを使って$\{q_i\}$を求める。

5. CPU使用率$\rho_1 > 1$となる理由

Fig. Aの系が2多重で動作しているとしよう。ジョブJ_1は，1回の入出力を要求するまでに40ミリ秒のCPU時間を使い，ジョブJ_2は1回の入出力を要求するまでに400ミリ秒のCPU時間を使うと仮定する。CPUのタイム・クワンタムを40ミリ秒とすれば，ジョブJ_1がI/O系からCPU系へ到着したときには10回のうち9回まではジョブJ_2がCPUを専有している。J_1がI/O系で30ミリ秒かかり，再びCPU系へ到着したときには，J_2がCPUを使う残り時間は$E(Q_r) = \rho_1 Q/2 = 1 \times 40/2 = 20$ミリ秒ではなく，$1 \times 40 = 40$ミリ秒即ち$E(Q_r) = \rho_1 Q$であることが多い。したがって$E(Q_r) = \rho_1 Q/2$としているわれわれの方法では，このとき$W_1$の過少評価，即ちCPU使用率$\rho_1$の過大評価となる。これが$\rho_1 > 1$となる理由である。

6. プログラム

プログラムの流れの主な部分とサブルーチンの機能をFig. B, C, D, Table Aに示す。

参考文献

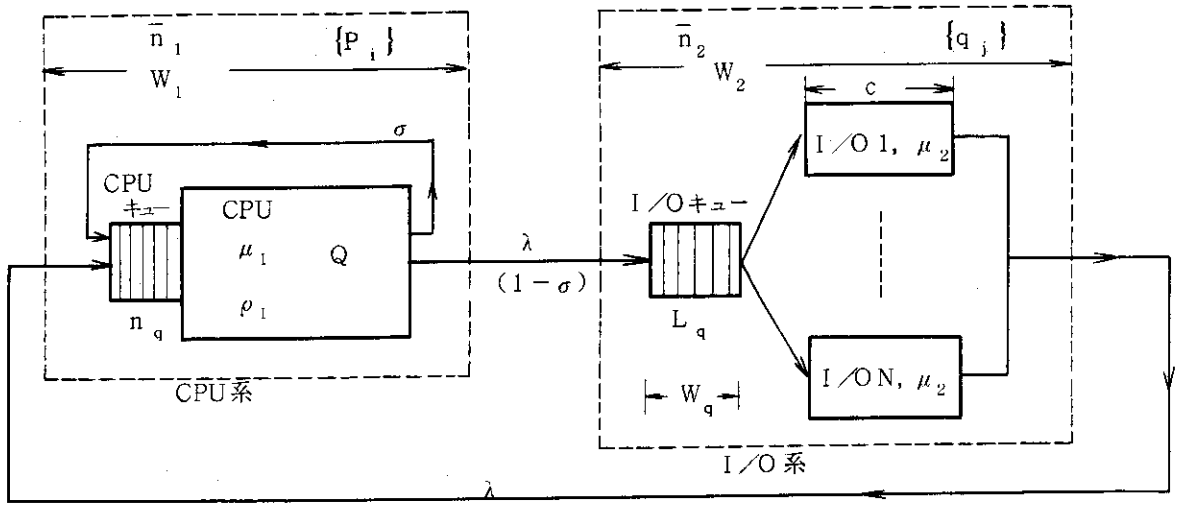
a. 「The output of a queuing system」, Burke, P. J., Oper. Res. 4 (1956), 699-704.

b. 「Probability, statistics, and queuing theory」, Allen, P. O., Academic Press, N.Y., 1978.

c. 「Operating system theory」, Coffman, E. G., and Denning, P. J., Prentiee-Hall, NJ, 1973.

Table A Names and functions of subroutines in the simulation program.

ルーチン名	説　　明
ARINIT	スタック中のジョブをアクティブ・ジョブ用データ領域（COMMON／PARAM／）にセットする。到着率λ，etc. を求める収束計算の初期値を決定する。
ARRIVL	ラウンド・ロビン法の分岐（CPU ループ）確率σを収束計算で求め，到着率λ，CPU 使用率ρ_1，I／O 1回当りのジョブのCPU 系滞在時間W_Kを計算する。
ELAPS	現在実行中のジョブのうち，1つが終了する前に次のジョブがスタートする時間が来た場合，その間の処理済みCPU time，I／O 回数，etc. を求め，次のジョブのスタート時刻へ時計CTIMEを進める。
GS	CPUサービスを受けるジョブがi クワンタを要求する確率$g_i=\sigma^{i-1}(1-\sigma)$を求める。
INPUT	プログラム制御情報をFT05 から，ジョブ・データをFT11 から読む。
JOBEND	最適多重度が求まったとき，またはアクティブ・ジョブのうちあるジョブが終了するまでに次のジョブがスタックされないとき，最小滞在時間ジョブの終了処理をおこなう。また，この時間まで時計を進める。
MAIN	ジョブ・スケジューリング計算を制御する。
MEMORY	次に実行するジョブが用いるメモリ量と現在の空きメモリ量を見てジョブを追加すべきかどうかを判定する。
MINT	配列にある変数の最小値を探索する。
MULPRO	多重度mとm+1（old／newと呼ぶ）の2つのアクティブ・ジョブ・パターンのCPU 使用率を比較して最適多重度を判定する。
PROBA1	CPU 系でのジョブ滞在確率$\{P_n\}$，平均ジョブ個数$\overline{n_1}$を求める。
PROBA2	I／O 系でのジョブ滞在確率$\{q_n\}$，平均ジョブ個数$\overline{n_2}$を求める。
RQUANT	幾何分布$g_i=\sigma^{i-1}(1-\sigma)$に従う擬擬ジョブを発生させる。
SCHDL	LPTルールにより複数システムにジョブを振り分ける。1システムの場合は処理をおこなわない。
SORT	スタック，あるいはスタート・タイム順にジョブをソートする。
WJTAB	現在待ち状態ジョブ番号を与える。



λ ; CPU系への到着率(入力率) =I／O系への到着率

μ_1; CPUの処理率。幾何分布を仮定

μ_2; 1台のI／O の処理率，指数分布を仮定

ρ_1; CPU使用率

P_i ; i 個CPU系に滞在する確率

Q ; タイムクワンタム

c ; I／O 1回あたり所要時間

$\bar{n}_1$; CPU系に滞在する平均ジョブ個数

$\bar{n}_2$; I／O 系に滞在する平均ジョブ個数

σ ; CPU系でループする確率

N ; 並行処理可能なI／O 台数

M ; 多重度

W_1; CPU系での平均待ち時間

L_q; I／O 処理待ちの行列個数

W_2; I／O 系での平均待ち時間

W_q; 平均I／O 処理待ち時間

q_i; i 個I／O に滞在する確率

Fig. A. Closed network model for simulation.

MAIN

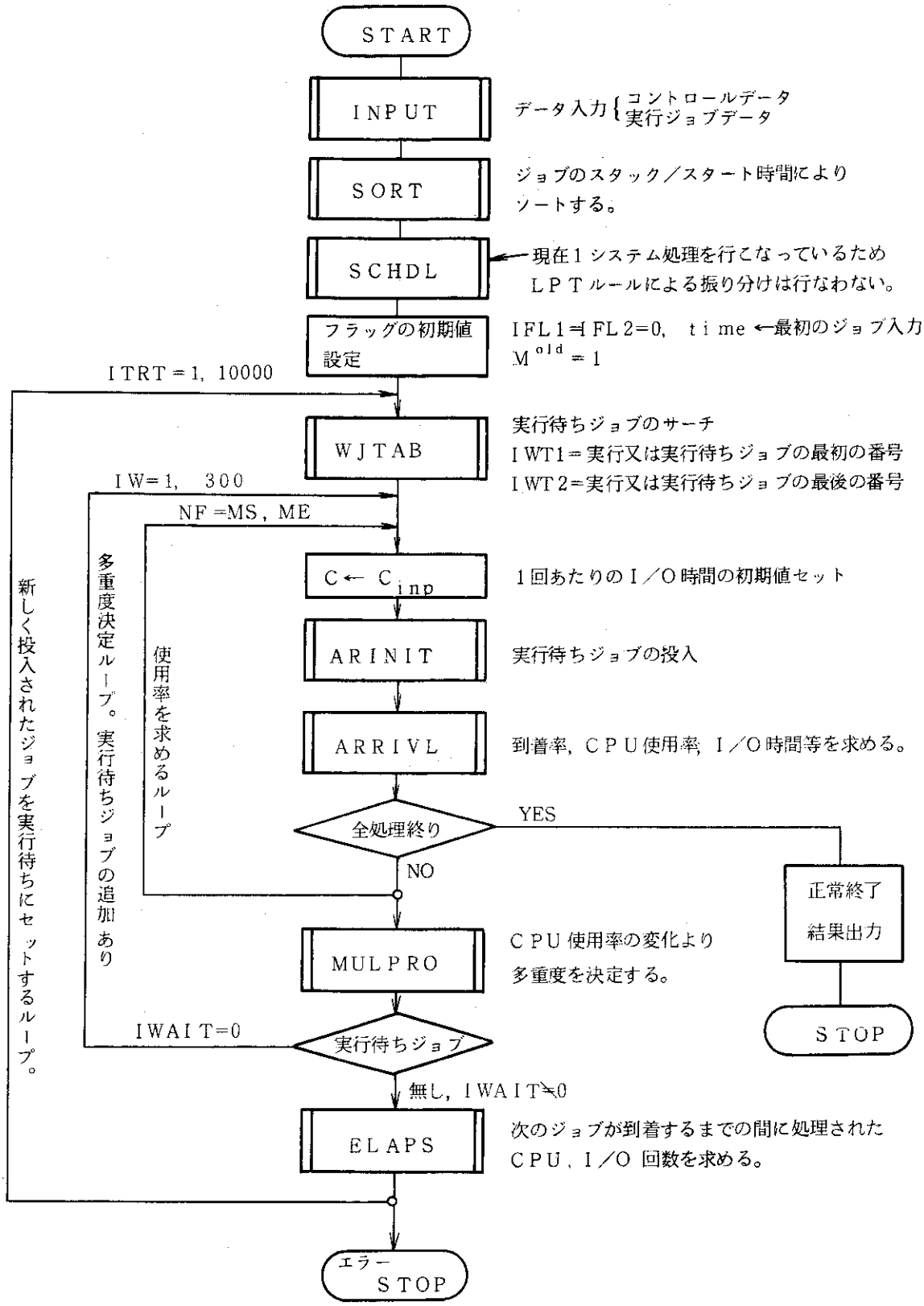


Fig. B. Flowchart of main routine.

サブルーチンELAPS

ある多重度mで実行している間に処理されたCPU時間，I／O回数等を求める。

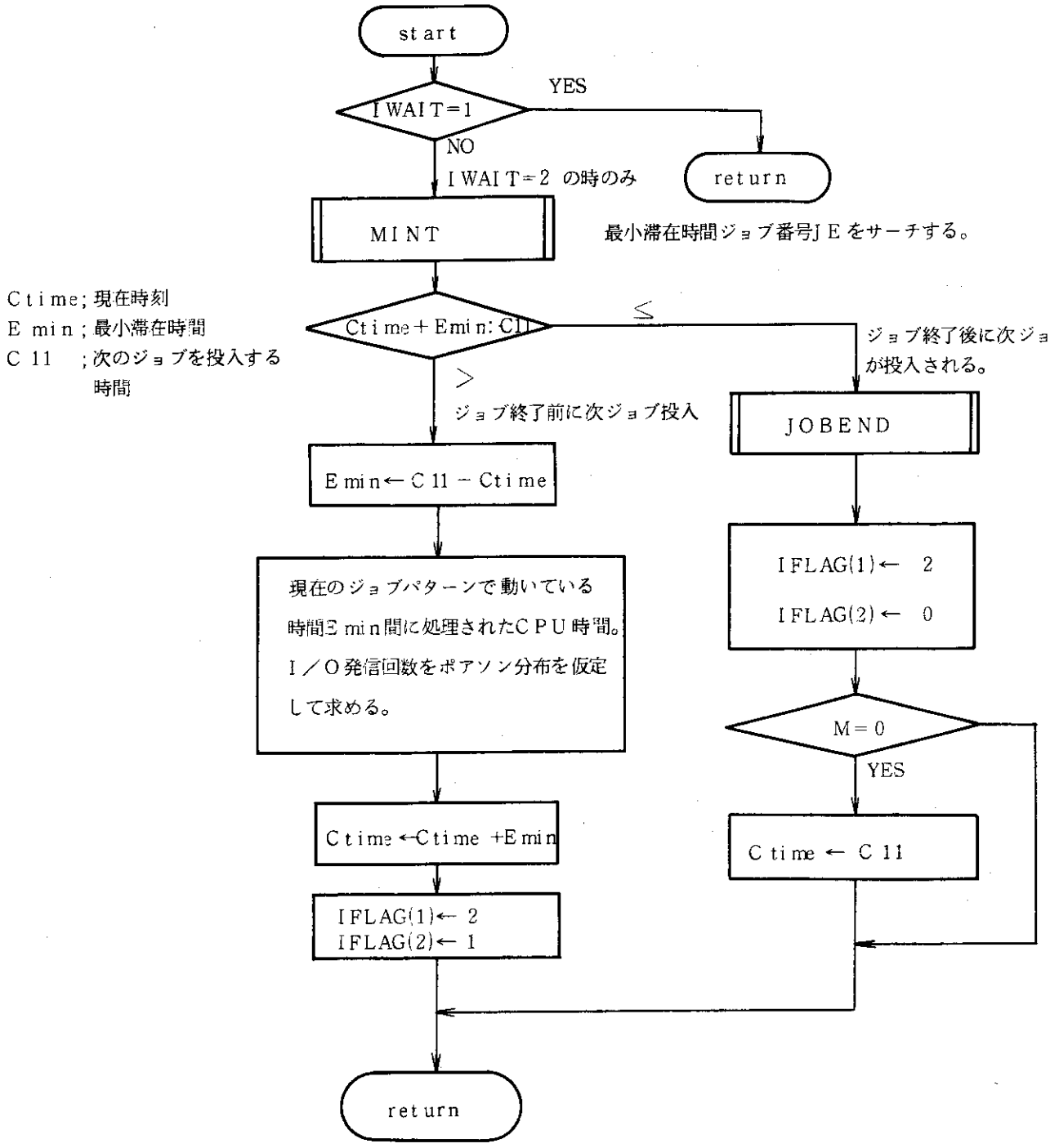


Fig. C. Flowchart of ELAPS routine.

サブルーチン　MULPRO

多重度mとm＋1のρを比較し多重度を求める。最適多重度が求まった時は最小滞在時間ジョブを終了させる。

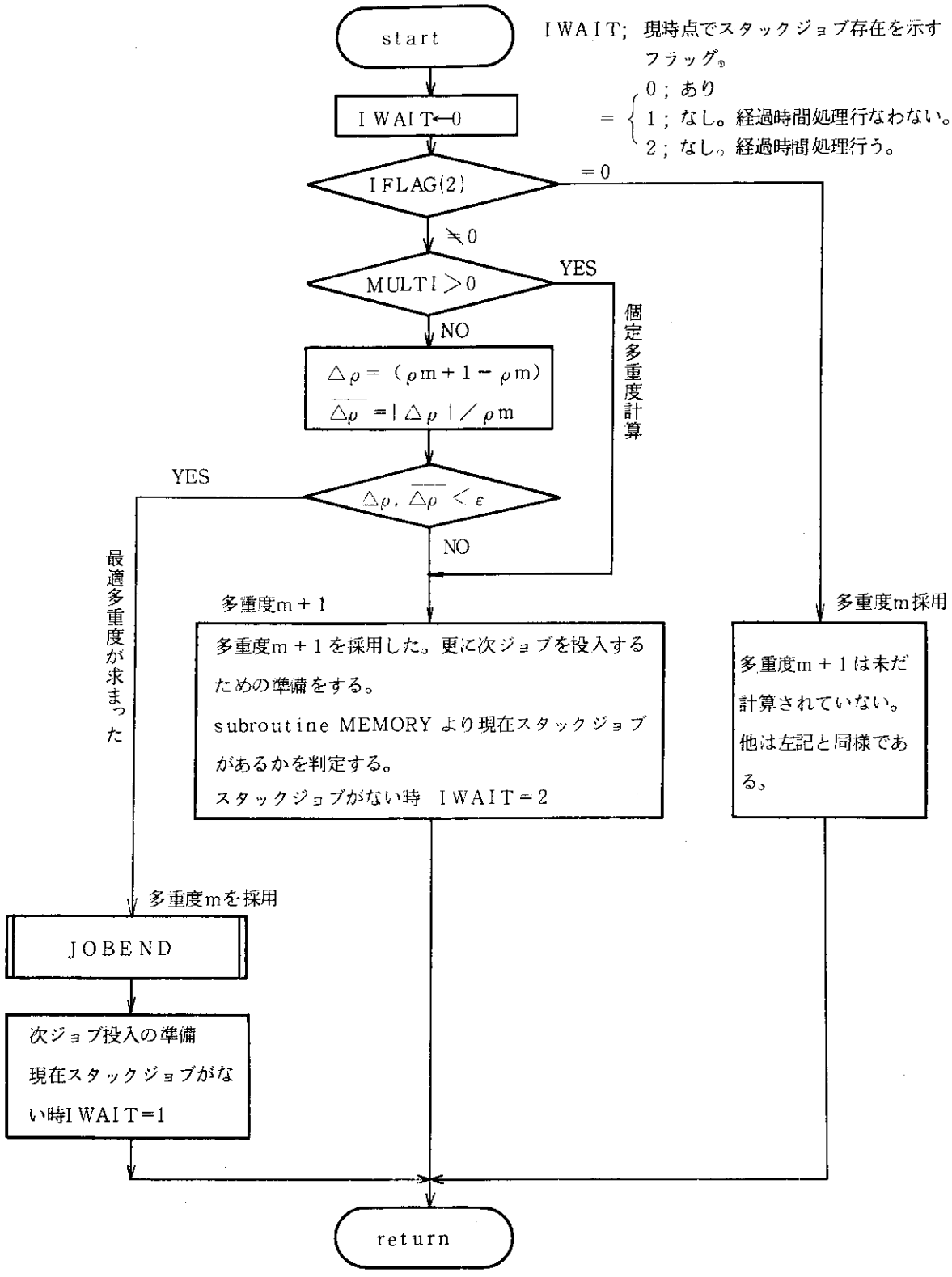


Fig. D.　Flowchart of MULPRO routine.