

JAERI-M

8 7 1 4

FACOM 230-75 計算機の性能評価(1)

1980年3月

浅井 清・藤井 実・吉田 耕造\*  
望月 紀\*・福沢 幸男\*・田淵 治樹\*

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問い合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしてください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

FACOM 230-75 計算機の性能評価 (1)

日本原子力研究所東海研究所計算センター

浅井 清・藤井 実・吉田耕造\*

望月 紀\*・福沢幸男\*・田淵治樹\*

(1980年1月28日受理)

FACOM 230-75 の TSS 性能を評価するために、F 230-75 システムを模擬する3種のモデルを設定した。すなわち J. Shemer and D. Heying の BTM モデル、R. M. Brown et al. の UT-2 D モデル、および H. Kobayashi の拡散方程式モデルの3種である。本報告では、これらのうち BTM モデルと UT-2 D モデルについてのべる。

また、バッチ処理のスループット性能をみるために、F 230-75 のバッチ処理性能を模擬する循環待行列モデルを作成した。

BTM モデルは F 230-75 計算機の性能向上に役立ち、UT-2 D モデルは、BTM モデルで採用しているメモリ使用方法の部分をさらに精密にするために用いられた。しかし、得られた結果はさほど変わらないことがわかった。

循環待行列モデルは、バッチ処理における F 230-75 の問題点の発見に役立った。

さらに計算機効率評価の考え方について述べた。

---

\* 富士通(株)

JAERI-M 8714

Computer Performance Evaluation  
of FACOM230-75 Computer System

Kiyoshi ASAI, Minoru FUJII,  
Khozō YOSHIDA\*, Osamu MOTIZUKI\*,  
Yukio FUKUZAWA\*, Haruki TABUTI\*

Computing Center, Tokai Research Establishment, JAERI

(Received January 28, 1980)

From 1976 to 1978 a series of computer performance evaluations and cost/benefit analyses were made at the JAERI Computing Center.

In this report are described the procedures and results of analyses of the timesharing system and batch processing system for FACOM230-75 computers in JAERI. The timesharing system is analyzed with BTM model of J. E. Shemer and D. W. Heying and UT-2D model of R. M. Brown et al., and the batch processing system is analyzed with a cyclic queue model.

Following the evaluations many modifications and improvements for the computer systems were made, raising significantly performance of systems.

Keywords: Computer, Performance Evaluation, Analytic Model

---

\* Fujitsu, Ltd.

## 目 次

1.	はじめに	1
1.1	計算機性能評価の必要性	1
1.2	原研計算センタの概要	1
2.	計算機性能評価の方法とデータ収集	4
2.1	評価のためのモデルの種類と特徴	4
2.2	モデル作成の意義	4
2.3	生データの収集方法	5
3.	BTMモデルによるTSS処理の応答時間解析	6
4.	UT-2DモデルによるTSS処理の応答時間解析	11
4.1	UT-2Dモデルの目標	11
4.2	解析手法の概要	11
4.3	計算結果	14
5.	解析モデルによるバッチ処理のスループット解析	17
5.1	テスト用ジョブ	17
5.2	解析モデル	18
5.3	性能の予測	20
6.	性能評価による成果	22
7.	効率評価の考え方	23
7.1	効率評価のレベル	23
7.2	Aレベル効率評価の例	26
8.	おわりに	31
8.1	Cブロック・レベルの性能評価について	31
8.2	Bブロック・レベルの効率評価について	32
8.3	Aブロック・レベルの効率評価について	32
	謝 辞	33
	参考文献	33
	付録A BTMモデルの計算法とパラメータ	35
	付録B UT-2Dモデルの計算法	40

## Contents

1. Introduction .....	1
1.1 Our necessity of computer performance evaluation .....	1
1.2 Outline of JAERI computing facilities and operation .....	1
2. Methods of evaluation and acquisition of data .....	4
2.1 Characteristics of models for performance evaluation .....	4
2.2 Meaning of modeling .....	4
2.3 Methods of data acquisition .....	5
3. TSS response time analysis by BTM model .....	6
4. TSS response time analysis by UT-2D model .....	11
4.1 Purpose of UT-2D model .....	11
4.2 Outline of UT-2D analysis .....	11
4.3 Computed results .....	14
5. Thruput analysis of batch processing by cyclic queue model ..	17
5.1 Jobs for test .....	17
5.2 Cyclic queue model .....	18
5.3 Prediction of performance .....	20
6. Obtained results by performance evaluations .....	22
7. A concept of cost/benefit evaluation .....	23
7.1 Levels of cost/benefit evaluation .....	23
7.2 An example of A-level cost/benefit evaluation .....	26
8. Concluding remarks .....	31
8.1 On C block level cost/benefit evaluation .....	31
8.2 On B block level cost/benefit evaluation .....	32
8.3 On A block level cost/benefit evaluation .....	32
Acknowledgement .....	33
References .....	33
Appendix A. Computational method of BTM model .....	35
Appendix B. Computational method of UT-2D model .....	40

## 1. はじめに

### 1.1 計算機性能評価の必要性

昭和51年1月FACOM 230-75 計算機システムが正式に稼動を開始した。

FACOM 230-75 のタイムシェアリング・システム(TSS)を日本原子力研究所(以下原研と略す)で使用するにあたり、このTSSには3つの問題があった。

- (1) そのひとつはTSSコマンド(命令)のマクロ化機能の欠如であった。TSSでも、バッチ処理と同じく、プログラムの翻訳、結合編集などの定型的使用法が多い。この場合に毎回利用者が端末から打鍵を繰返していたのでは、時間もかかり、誤ちも多くなる。このような定型的使用法においては、そのコマンド列に名前をつけて計算機システムに登録しておき、使用するときは、その名前を呼び出すことでコマンド列が自動的に入力されるようになればよい。これをおこなうのが、TSSコマンドのマクロ化の機能である。この機能は原研の要求によって昭和51年12月に計算機メーカーから提供され、この問題は解決された。
- (2) もうひとつの問題はTSS性能であった。TSS性能は大きく分類すると、端末応答時間の速さ、同時に使用できる端末の数、バッチとTSSジョブの演算装置(CPU)使用率の高さ、などで判断することができる。ページ化された主記憶装置、仮想記憶の設計方式、これらの方式を支える各種のハードウェアとソフトウェアは、現在では効果的なTSS利用に必須のものとされている。

FACOM 230-75 計算機はこれらの装置と機能を持っていない。そのために、この計算機システムのTSS性能を向上させること、同時に使用できる端末数を正確に予測すること、などが必要であった。

- (3) また計算機システムを動作させるソフトウェアの改善、変更が間断なくおこなわれていたが、これらの作業がバッチ処理のスループット性能にどのような影響を与えるか推定する必要があった。

本報告書は以上の諸項目について理論的、研究的にというよりは、実戦的に計算機性能評価をおこなった方法と結果について述べたものである。

### 1.2 原研計算センターの概要

筆者らが計算機性能評価に力をそそいだ昭和52年当時の計算機構成は図1.1、また計算センターの運用と利用状況はつきのとおりであった。

〔運用〕

ジョブ実行多重度(昼間)

- ・システムA

クローズド・バッチ処理	_____	5多重
リモート・バッチ処理	_____	
タイムシェアリング(TSS)処理(端末多重度:26)	_____	4多重
  
- ・システムB

オープン・バッチ処理	_____	4多重
(カード・リーダー, ライン・プリンタはユーザに開放)		
セミ・オープン処理	_____	2多重
(磁気テープもユーザに開放)		
  
- ・オープン利用は9時~17時, リモート, TSS利用は9時~20時である。夜間は両システムともクローズド処理となる。

〔利用〕

- ・ユーザ数

登録者数	約750名
実利用者数	約600名
TSS実利用者数	約200名
  
- ・ジョブ処理件数

バッチ	約20,000件/月
(リモート・バッチ)	(約5,000件/月)
TSS	約5,000件/月
グラフィック	約1,200件/月
  
- ・ターン・アラウンド時間

オープン処理	優先権 7~4	3分~10分
	" 3~0	15分~180分
クローズ処理	優先権 7~4	5分~15分
	" 3~0	20分~200分



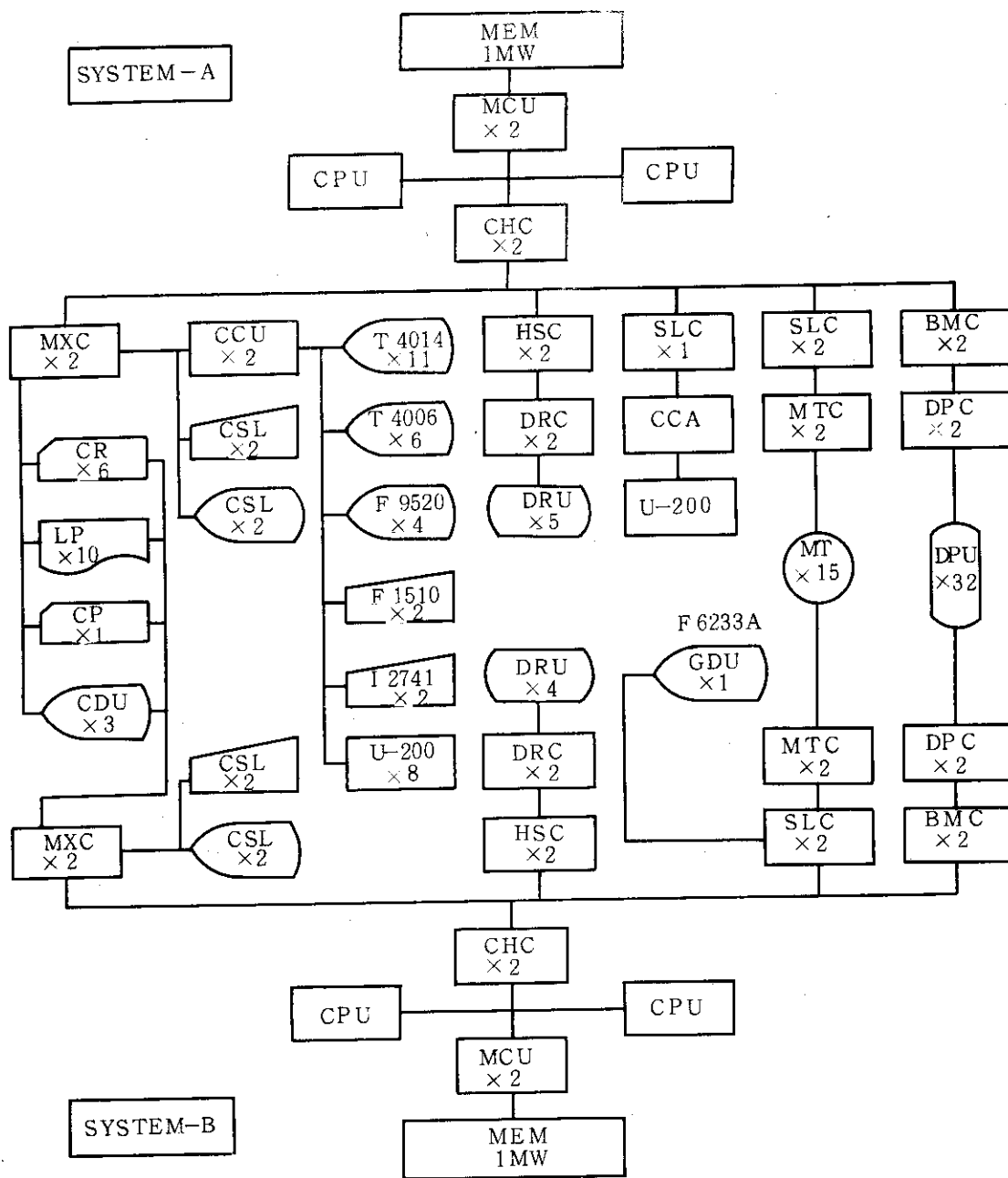


Fig. 1.1 Computer Systems at JAERI, 1977

## 2. 計算機性能評価の方法とデータ収集

ここでは計算機性能評価の方法と筆者らが使用したデータの収集方法について述べる。

### 2.1 評価のためのモデルの種類と特徴

評価のモデルは大別してシミュレーション・モデルと待ち行列理論による解析モデルとがある。シミュレーション・モデルと待ち行列理論による解析モデルそれぞれのもつ長所、短所をいえば次のようになる。

[シミュレーション・モデル]

長 所

- (1) 記述が簡単。
- (2) 簡単なモデル、複雑なモデル、定常モデル、非定常モデルなど多様なモデルを記述できる。

短 所

- (1) モデルごとにシミュレーションをおこなわねばならないので計算時間がかかる。
- (2) シミュレーション言語（たとえば GPSS）のもつ機能の枠を出ない。

[待ち行列理論による解析モデル]

長 所

- (1) 対象となる処理系の各種パラメータについての平均、分散、変動係数などから処理系のマクロ的動きを推定できる。
- (2) 記述が簡単で計算時間が短い。
- (3) モデルの変更が容易で機器構成の変更による性能変動を予測しやすい。

短 所

- (1) 複雑なモデルの記述には制約条件が多くなる。<sup>1)</sup>
- (2) 非定常モデルの計算が難しい。

### 2.2 モデル作成の意義

モデル作成の意義を図 2.1 によって説明しよう。図 2.1 で統計的解析とあるのは、得られた生データから各種パラメータについての平均、分散、変動係数を作るなど統計的処理をおこなうことをいう。この結果をもとに適当な、つまり常識的にいえばシステムはかくあるであろうと思われるようなモデルを作成し、同じデータでモデルの振舞いを見る。このモデルの振舞いが現状と一致しない場合は、モデルが悪いのである。このときモデルの改良で済めばよいが、原因がわからず、新しい情報を得るためにもう一度生データの収集から始めなければならないこともある。現状と一致するモデルが得られれば、そのモデルをながめることによって比較的容易に現状の計

算機システムの構成をどのように変更すればよいか分かる。このように書いてくと図 2.1 の流れ図の最後のブロック「システムの改良」まで簡単に到達できそうであるが、実は生データの収集からしてが大変な作業である。

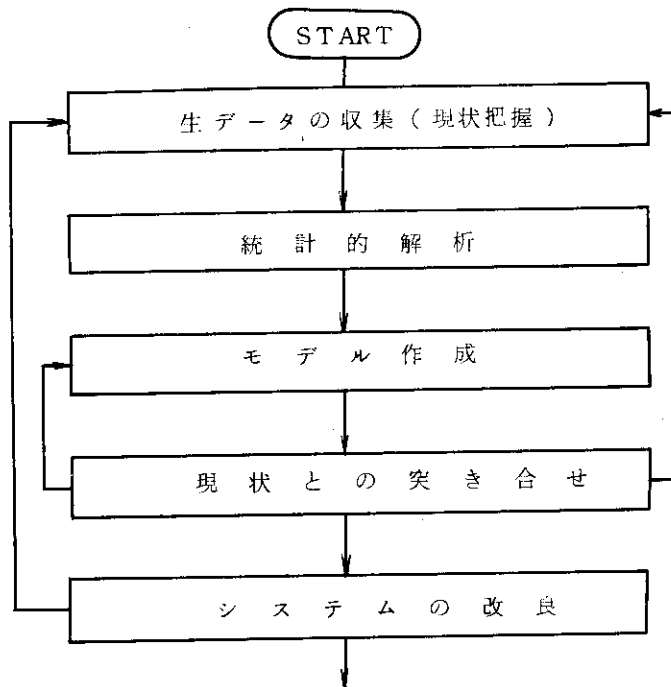


Fig. 2.1 Procedure for Performance Evaluation

### 2.3 生データの収集方法

筆者らのおこなった性能評価作業のうち、生データの収集は次のようにしておこなった。

ソフトウェアによるもの

- (1) PANSY : FACOM 230 - 75 性能評価データ収集用ソフトウェア。
- (2) HOPE : FACOM 230 - 75 OS 動的制御のためのソフトウェアであるが、この評価作業の段階ではデータ収集の機能しかもっていなかった。
- (3) 会計情報 : 通常の会計情報である。
- (4) TSSコマンド・ログ : TSSコマンドに関する記録をとる。

ハードウェアによるもの

- (5) マイクロ・サム : 性能評価データ収集用外づけマイクロ・コンピュータ。
- (6) パルス・カウンタ : 必要に応じて5~6台を使用。

これらの道具をすべて使っても、モデルの計算に必要なデータが取れないことがある。

### 3. BTMモデルによるTSS処理の応答時間解析

原研ではFACOM 230-75Aシステムでクローズ・バッチ、リモート・バッチ、タイムシェアリング(TSS)処理をおこなっている(図1.1)。昭和52年11月時点でAシステムに接続されているリモート・バッチ・ステーションは8台、TSS端末は28台であり、TSS端末はすぐに15台程度増設されて約40台になる予定であった。当時はクローズ・バッチ、リモート・バッチ合せて5多重の多重度で運用し、TSS端末は常時半数の14台程度、混雑時には25台近くの端末が使用されていた。

現在は52台のTSS端末、25台のリモート・バッチ・ステーションが接続されている。

端末の需要は年々増加の傾向にあるから、Aシステムがどれだけ端末の増加に耐えられるかが問題となる。この問題を評価するための解析的手法としてJ. E. ShemerとD. W. HeyingのBatch/Timesharing Monitor(BTM)モデル<sup>5)</sup>を使ってみた。このBTMモデルはバッチのCPU(中央演算装置)利用率とTSSの応答時間の両方を同時に算出するところに特徴があり、入力時間間隔、処理時間ともに指数分布を仮定している。その計算手法の詳細は付録Aに掲げる。

このBTMモデルによる計算機の動きを図3.1に示す。ここでA、BはTSSジョブ、Cはバッチ・ジョブである。このモデルではTSSジョブBの実行には必ず主記憶にあるTSSジョブAの追出し(ロールアウト)とロールアウト・ファイルにあるジョブBの呼び込み(ロールイン)をとまらう。ジョブA、Bのロールアウト/ロールイン中の時間(以後この平均時間を $\bar{S}$ であらわす)にCPUは主記憶にあるバッチ・ジョブCを実行する。

実際にはバッチ・ジョブCの実行時間 $\bar{S}$ 中にモニタ(管理プログラム)の使用時間が多く散在するのであるが、それらを一括しモニタ時間 $\bar{m}$ とする。

これらの関係を模式図で表現したのが図3.1の線表である。

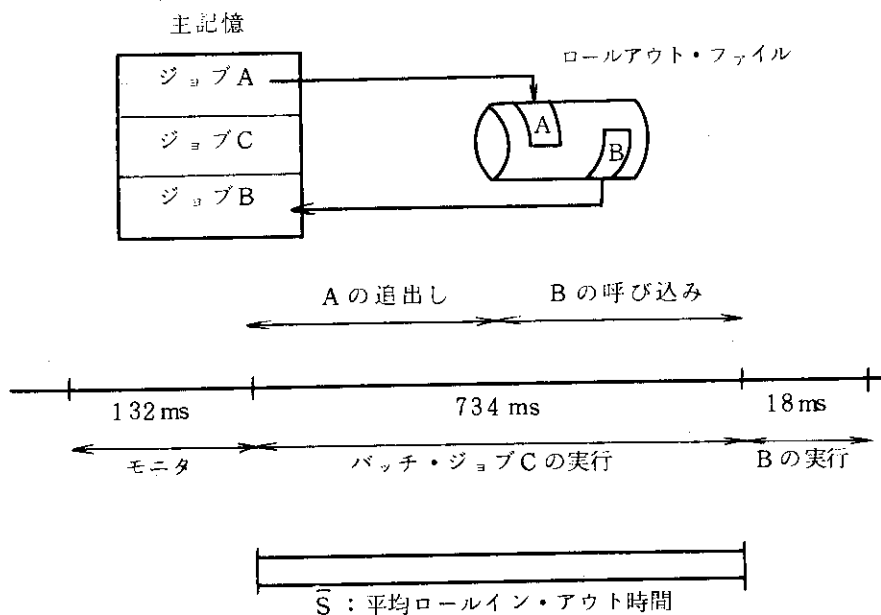


Fig. 3.1 TSS Process of BTM Model

原研では大規模ソフト開発のためにTSSが使用されることが多く、処理される個々のジョブの主記憶使用量も常識的な規模よりは大きい。そして FACOM 230-75はページ・マシンではない。この関係を表示したのが表 3.1 である。表 3.2 の数値は昭和52年10月26日14:00-16:00 時のものである。この頃は学会終了直後のこともあって計算機は比較的空いていた時期である。

さて図 3.2 は BTMモデルにこれらの数値を入れて計算した結果である。この結果と筆者らの使用経験とを照し合せて次の3点に気づいた。

- (1) 平均的ジョブの応答はもっと速い。それは経験的曲線になるはずである。
- (2) 軽いジョブの応答はもっと速いはずである。
- (3) 会計情報と照し合わせると平均的ジョブのCPU使用率は85%ではなく50%近くになるはずである。

この食い違いの原因は次の2点にあると推定される。

- (1) BTMモデルは安全側(これ以上は遅くならないという側)に寄りすぎている。実際ジョブは毎回ロールイン/ロールアウトされるわけではなく、主記憶に空きがある限りは追い出されないで残っている。すなわち平均的な $\bar{S}$ はもっと小さい。

Table 3.1 TSS Job Sizes and Roll in/out Time

TSS ジョブ使用主記憶量	セッション数	平均主記憶量	ロールイン・アウト時間
1 - 30 KW	15	28.7 KW	302 ms
31 - 256 KW	32	71.1 KW	952 ms
1 - 256 KW	47	57.6 KW	734 ms

(注) ロールイン・アウト時間は待ち時間も加算している。

Table 3.2 Example of TSS Terminal Interactions

端末入力回数	端末出力回数	合計
1580	9643	11,223

(注) S 52年10月26日の14時~16時のデータ。

(2) 軽いジョブの場合モニタ使用時間 $\bar{m}$ は $\bar{m} = 132 \text{ ms}$  (ミリ秒)ではなく、この5分の1程度のはずである。

BTMモデルのもつこれらの短所を取り除くために、ジョブの主記憶滞在確率まで考慮した解析モデル (UT-2Dモデル) を検討した (第4章)。

それは平均的ジョブについての経験的応答曲線をよく近似するであろうが、実動端末数が増加した場合の応答時間の遅れの問題は残る。

これの対策として筆者らは次のように考えた。

- (1) 重いジョブ、軽いジョブの分離: TSSコマンド・ログによれば、入力されるコマンド数の90%以上がテキスト編集の軽いタスクを起動している。ところがこれら軽いタスクは重いジョブと同じ径路を通過して同じロールアウト・ファイルに保存されている。経路とファイルを重いジョブ、軽いジョブ用との2系統に分離すれば、軽いジョブの応答は速くなる。
- (2) 平均ロールアウト時間 $\bar{S}$ の減少: 応答の遅い原因を全体的に言えば平均ロールアウト時間が長いことによる。FACOM 230-75のTSSでは端末に対する1回の入力、または出力がロールイン/ロールアウトの契機となる。ところが表3.2でわかるように原研での通常の使用方法では端末出力行数は入力行数のほぼ5~10倍である。この端末出力3回分まとめて1度書き出し、それをロールアウトの契機とすれば平均ロールアウト時間 $\bar{S}$ は3分の1程度に減少するはずである。

上記(1)、(2)の対策はソフトウェアではモニタ、その他のかなりの手直しをとまうが、メーカは原研の要望を入れて快く改造をひきうけてくれた。また当時の高速チャネルの使用率とロールアウト・ファイルの容量からみて(1)の対策には高速チャネルと磁気ドラムの増設が必要となり、これらの装置を設置した。

図3.3は上記2項目の対策を考慮したBTMモデルの計算値である。これらの曲線もやはり安全側によりすぎているので実際には応答はもっと早いはずである。

図3.4は軽いジョブを中心としたメーカの試算値である。この試算はHOPE<sup>12)</sup> (動的な資源配分、サービス・グループ制御などをおこなうソフト) の効果が考慮されている。

このHOPEは昭和53年5月に導入され、現在もFACOM 230-75計算機のジョブ処理性能の向上に寄与している。

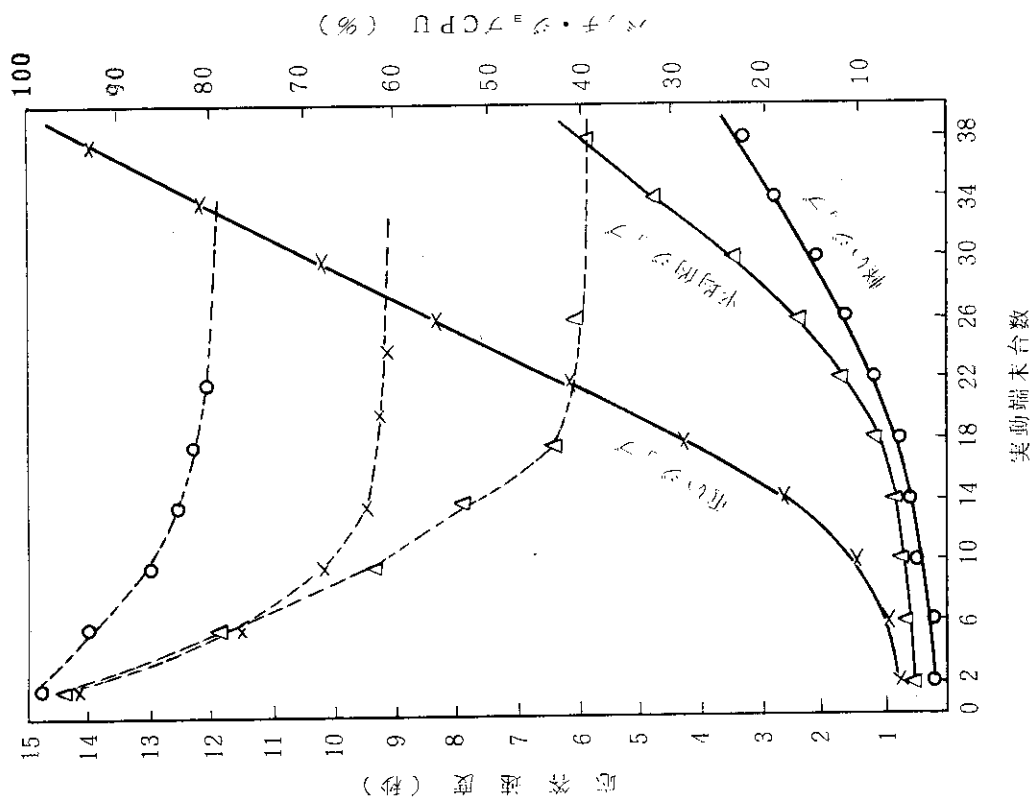


Fig. 3.3 Terminal Response Time 2

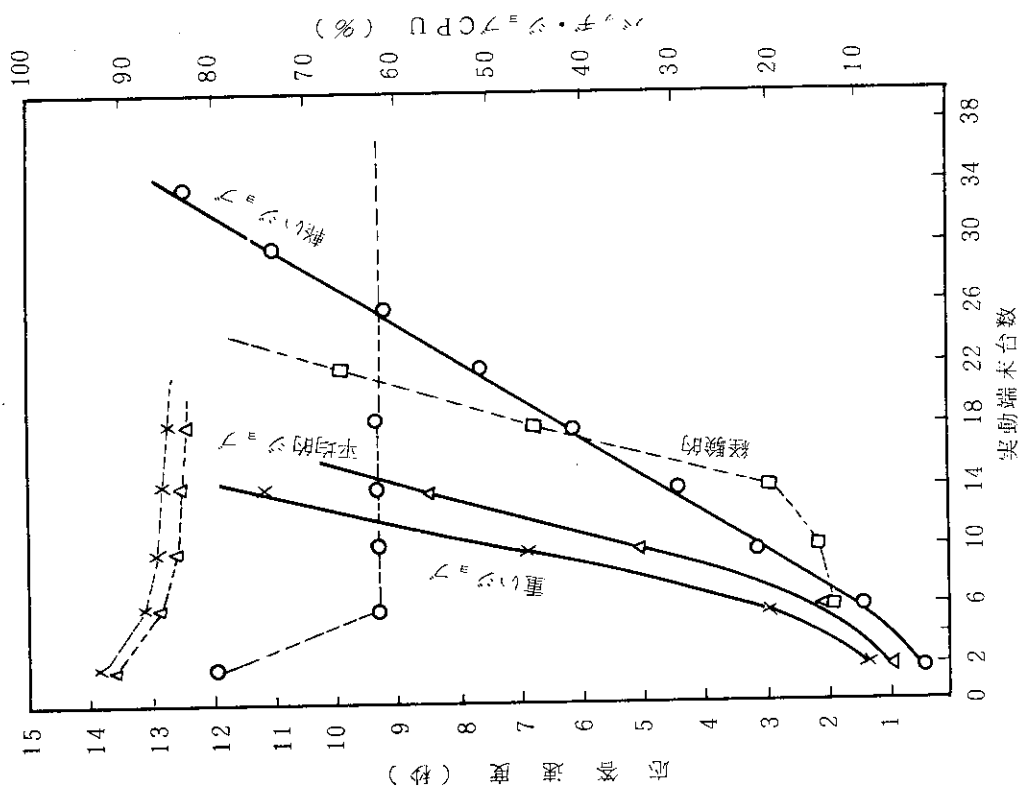


Fig. 3.2 Terminal Response Time 1

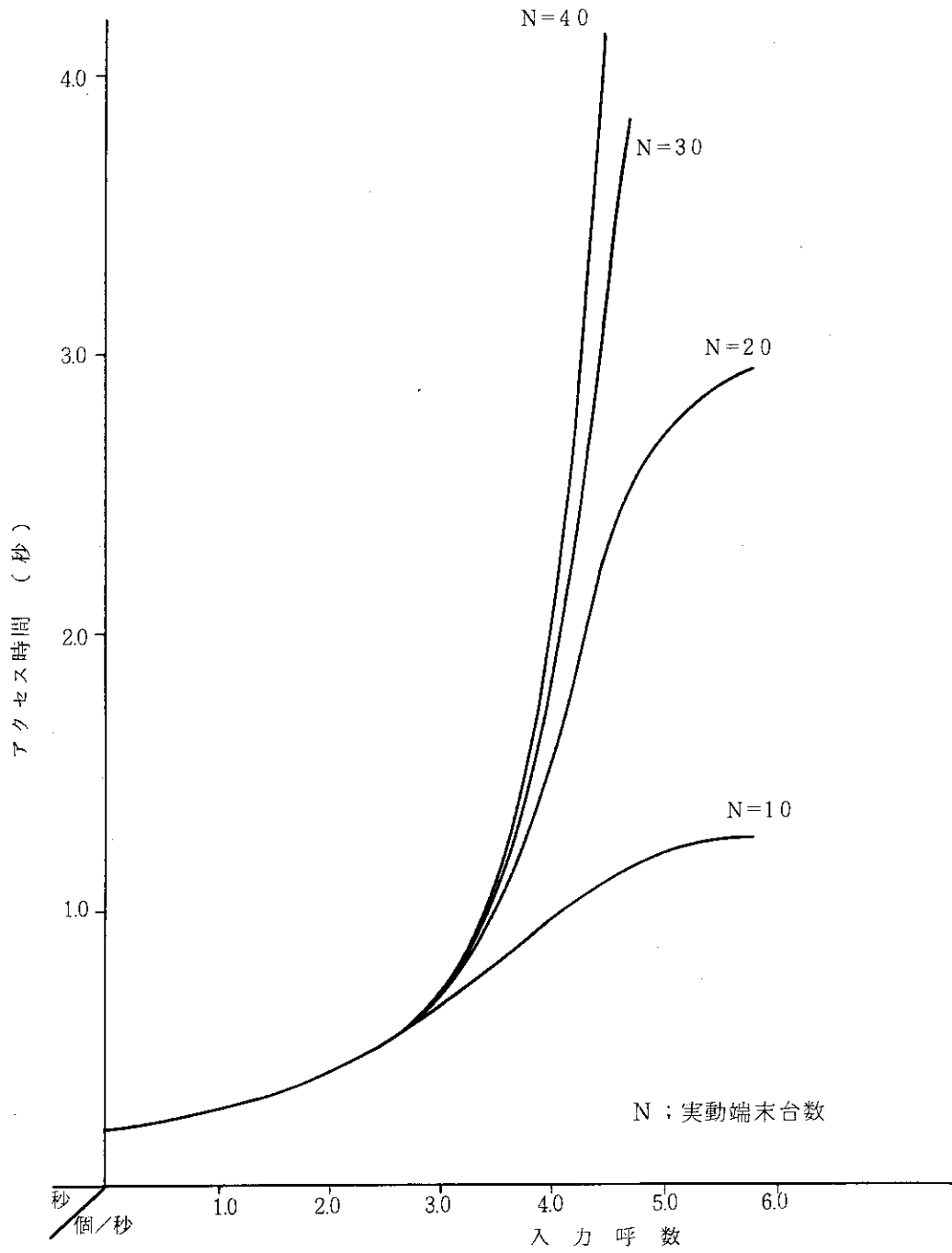


Fig. 3.4 Terminal Response Time 3



## 4. UT-2DモデルによるTSS処理の応答時間解析

前章のBTMモデルは、主記憶の取扱いが精密ではなく、そのために端末応答時間が安全側によりすぎる、すなわち、応答時間を長くみる傾向がある。この点をより精密にするために、テキサス大学のCDC 6400計算機によるタイムシェアリング・システムを評価したUT-2Dモデル<sup>6)</sup>を原研のTSSシステムにあてはめてみた。結果についてみると、(i) 実動端末台数が1~8台と少ないときはUT-2Dモデルのほうがより現実に合っており、BTMモデルは安全側に寄っている。(ii) 実動端末台数が8~10台を超えると、両者のモデルは端末応答時間に関し同じ傾向、同じ値を示す。そして、このときの応答時間は当時の状況と一致していた。

以下にUT-2Dモデルの考え方、およびそれを原研のTSSシステムにあてはめたときの結果を述べる。

### 4.1 UT-2Dモデルの目標

UT-2Dモデル作成の目標は、開発者 R.M. Brown et al. によると次のようなものであった。

- (a) ページ化されていないメモリをもつUTシステム(Univ. of Texas のCDC 6400の会話形システム)の扱いやすく正確なモデルを用意すること。
- (b) 解析的に解けるモデルであって、しかもそのなかに実行可能なメモリは有限であるという制限を陽に含むこと。
- (c) 主記憶使用率、応答時間、演算装置(CPU)使用率、スループットなど、システム性能指標について負荷特性、特にメモリの要求とシステムの能力を決定すること。

本報告書の筆者の経験でもUT-2Dモデルについてのこの目標は達成されているように感じられる。仮想記憶方式でないCDC 6400, FACOM 230-75などの計算機でタイムシェアリング・システムを実現するときには、メモリがもっとも大きな制約条件であるが、UT-2Dモデルは現実の環境に合致する方法で、この制約条件を組込むことに成功している。

### 4.2 解析手法の概要

以下に R.M. Brown et al. のUT-2Dモデルの方法の概略について説明する。原研のFACOM 230-75 TSS にあてはめて考えた場合の計算方法は付録Bで述べた。

ひとつのジョブがxの大きさの主記憶を要求する確率をp(x)としよう。分布関数をP(x)とすると

$$P(x) = \sum_{y=0}^x p(y)$$

任意の時点において起動されているジョブは think, または ready の状態のいずれかにあるとする。端末から入力を待っているとき, ジョブは think の状態にあり, 主記憶を待っているとき, あるいは主記憶に存在するとき, ジョブは ready 状態にある。

(1) Phase 1 : 装置モデル

図 4.2(a)の待行列ネットワークは装置モデルと呼ばれる。この装置モデルでは利用者の思考時間がゼロと仮定してある。その点を除けば図 4.1 と同じである。このモデルにおけるジョブの数とは主記憶に割りつけられたジョブ数のことである。

装置モデル中に  $n$  個のジョブが存在するとき,  $r(n)$  を単位時間に追い出される (swap out) ジョブ数の期待値としよう。いいかえれば  $r(n)$  は  $n$  個のジョブ ( $n = 1, \dots, M$ ) が主記憶に存在するときの追出し性能を示す。Phase 1 で  $r(n)$  を決定し, それを Phase 3 で使う。

(2) Phase 2 : 主記憶割当てモデル

$m$  個の ready ジョブが存在するとき, そのうちの  $n$  個のジョブが主記憶を割当てられる条件付確率を  $h(n/m)$  で表わす。ここで  $n, m = 0, 1, 2, \dots, M$ 。  $h(n/m)$  はメモリ要求分布と主記憶の容量  $C$  に依存する。Phase 2 で  $h(n/m)$  を計算し, それを Phase 3 で使う。

(3) Phase 3 : 全体モデル

図 4.1 の待行列ネットワークは全体モデルと呼ばれる。このモデルにおけるジョブの数はシステム全体で起動されているジョブ数である。

ready ジョブが存在するとき, 単位時間にサービスを受けるジョブ数の期待値を  $q(m)$  とすると

$$q(m) = \sum r(n) h(n/m), \quad n \leq m, \quad m = 1, \dots, M.$$

(4) 装置モデルの目的

装置モデルの目的は  $r(n)$  を計算することである。 $r(n)$  を求めるために図 4.1 の装置モデルはローカル・バランスの条件をみたしていると仮定する<sup>1)</sup>。しかしここでは筆者らは Gordon & Newell et al.<sup>10)</sup>の方法(付録 B)で  $r(n)$  を計算することにした。

(5) 主記憶モデルの考え方

実行待ち (ready) ジョブは先着順にサービスを受ける (FCFS: First Come First Served) ものとする。したがって実行待ちジョブの待ち行列を走査してジョブを主記憶に入れるかどうかを決定するスケジューラは, 待ち行列を先頭から調べてゆく。

スケジューラが待ち行列の先頭から  $l$  番目のジョブまでを走査する条件付確率を  $g(n, y/l)$  とすると, これは次の式で与えられる。ここで  $C$  はメモリ容量を示す。

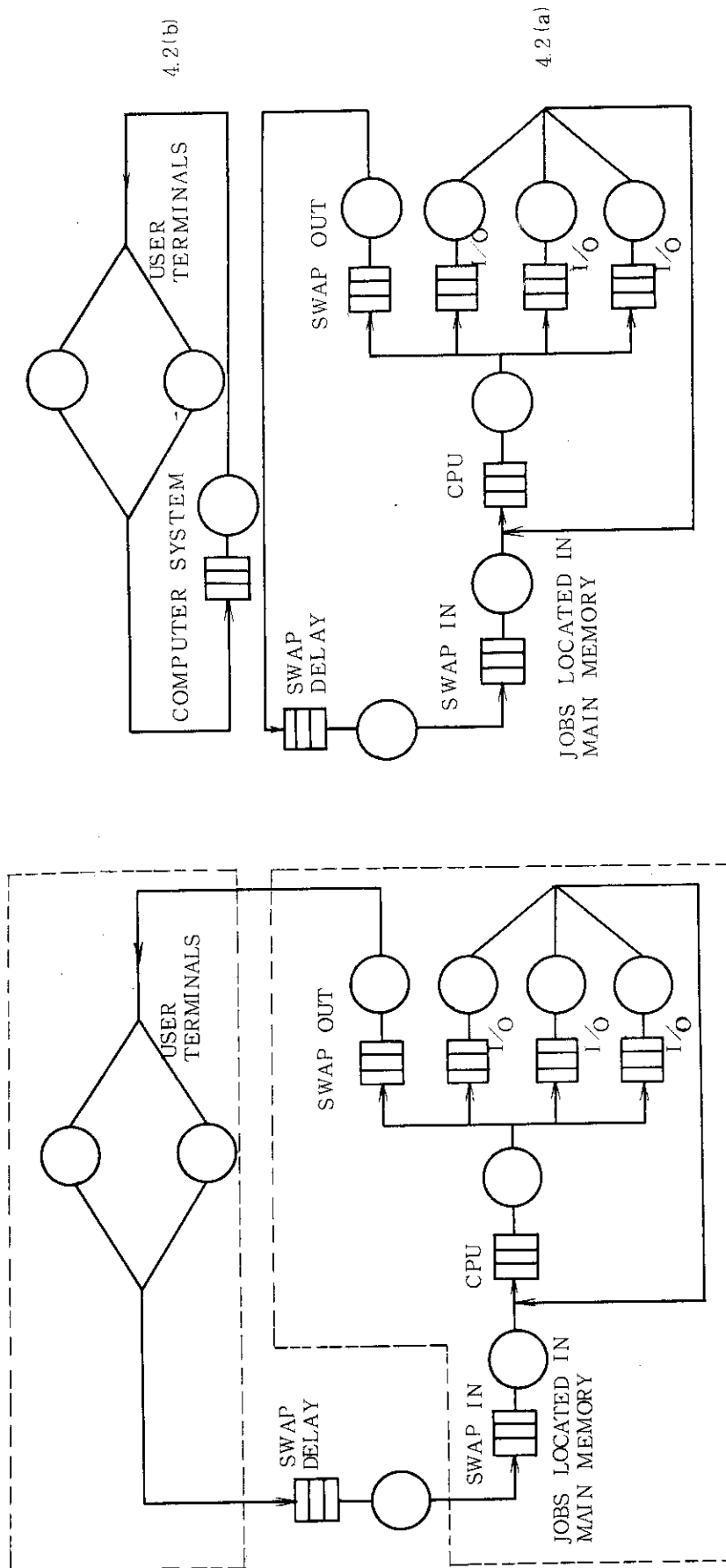


Fig. 4.2 Component Model

Fig. 4.1 Overall System

$$g(n, y/\ell) = \begin{cases} 1, & \text{for } n=0, y=0, \ell=0 \\ 0, & \text{for } n=0, y>0, \ell=0 \\ 0, & \text{for } n>\ell, \text{ all } y. \end{cases}$$

$$g(n, y/\ell+1) = g(n, y/\ell) \cdot (1 - P(C-y)) \\ + \sum_{x=0}^y g(n-1, x/\ell) \cdot P(y-x),$$

$$\text{for } \ell = 1, 2, \dots, M-1, \\ n = 0, 1, \dots, M, \\ y = 0, 1, \dots, C.$$

上式の右辺第1項は $(\ell+1)$ 番目のジョブを調べても、主記憶容量の制限から $(\ell+1)$ 番目のジョブを主記憶に取入れることができない確率を示し、第2項は主記憶駐在ジョブ数 $n$ 、メモリ総和 $y$ の範囲で $(\ell+1)$ 番目のジョブを主記憶に取入れることができる確率を示す。この $g(n, y/\ell)$ を使えば

$$h(n/m) = \sum_y g(n, y/\ell)$$

である。

### 4.3 計算結果

昭和54年12月20日15時～16時のF230-75Aシステムを例にとったUT-2Dモデルの計算結果を以下に述べる。

図4.3に計算に使用した装置モデルを示す。計算はテキスト編集タイプのジョブのみについておこない、バッチ・タイプのジョブは除外している。

計算結果は実測値（ソフトウェア・モニタ PANSY による。テキスト編集タイプのジョブのみ）

計算機側平均端末応答時間	1.2秒
計算機側平均思考時間	5.2秒
平均実動端末数	約14台

のとき、計算値（実動端末数約14台のとき）

計算機側平均応答時間	0.9秒
------------	------

となった。

また主な装置についての待行列の平均個数は、

実測値

CPU 1.8個 (PANSY による装置使用率と指数分布の仮定から)

HSC (高速チャネル) 1.7個 (CPUと同じ算出根拠から)

計算値

CPU 約1.6個

HSC 約1.9個

となり、計算値と実測値はかなり近い。

実動端末14台のときの入力率で計算機側応答時間の外挿値を計算したのが図4.4である。

しかし実際には入力率は実動端末数と共に増加してゆく。これを考慮すると実動端末数24台のときの計算機側応答時間は図4.5のようになる。BTMモデルよりは、主記憶に残存するジョブを考慮している分だけUT-2Dモデルは良い近似になっている。実際の応答時間は、この値に加えて通信文が通信回線を往復する時間が1~2秒程度必要となるので図4.5の応答性能ではTSS処理に支障をきたす。前述のソフトウェアHOPEでは、この問題点の改善がはかられている。

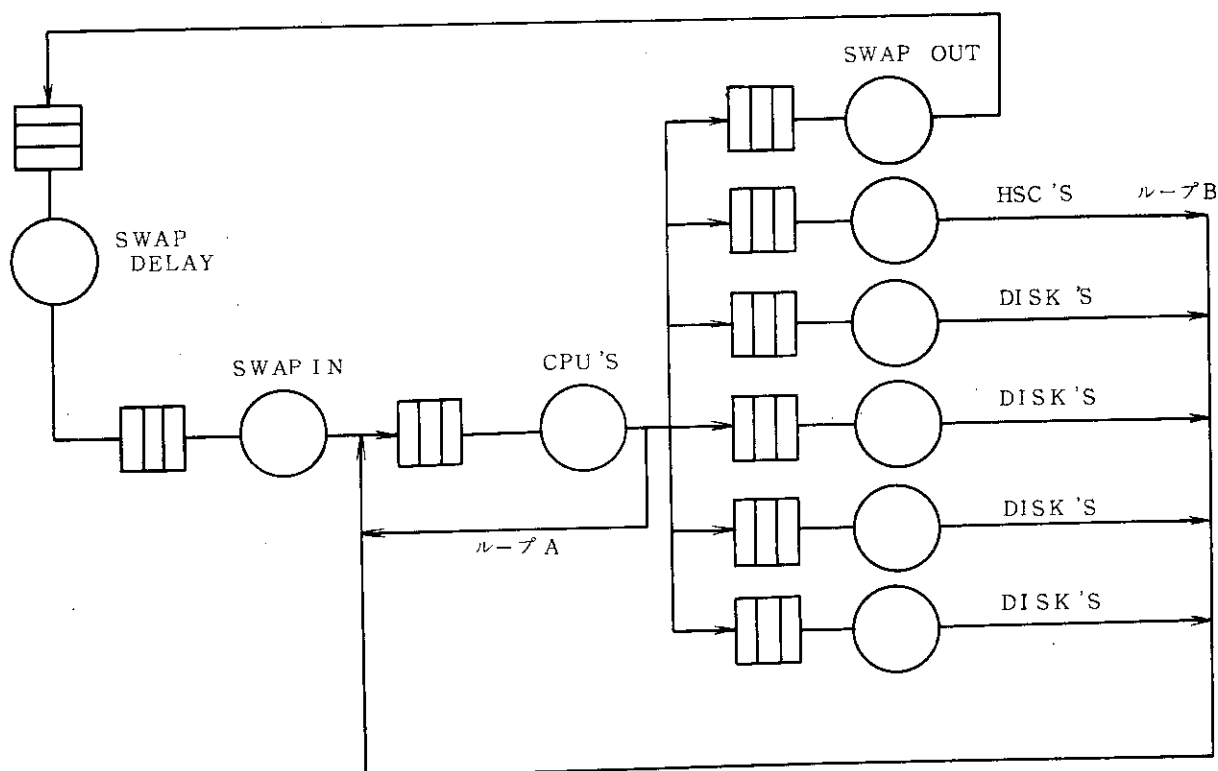


Fig. 4.3 UT-2D Model at JAERI.

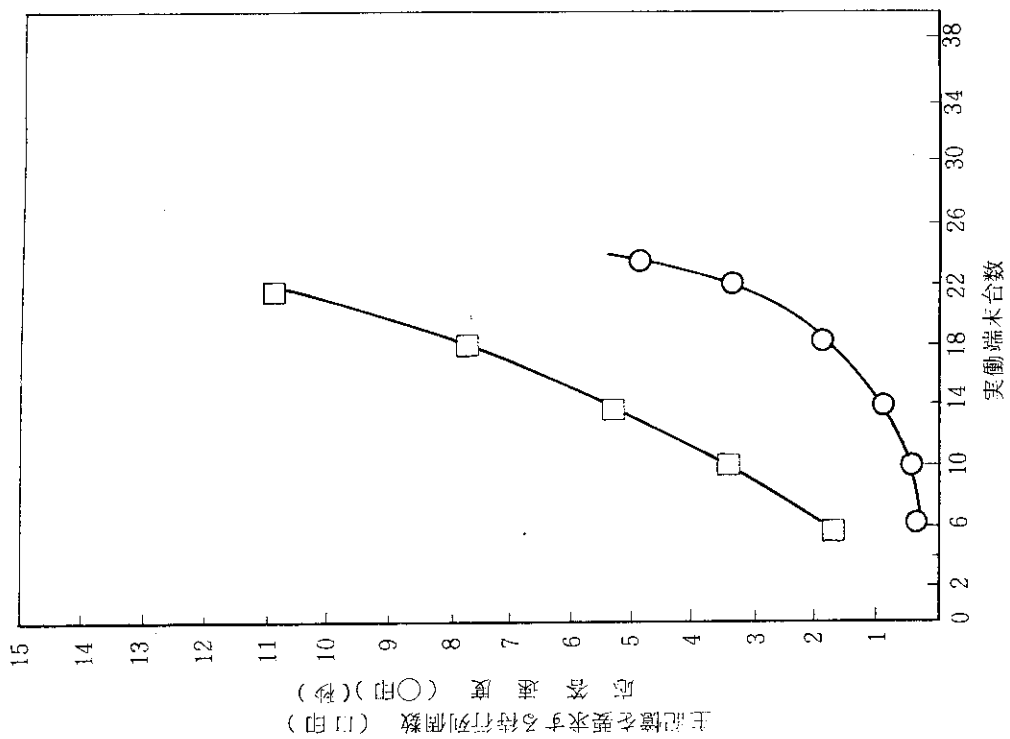


Fig. 4.5 Terminal Response Time 5  
( UT-2D Model )

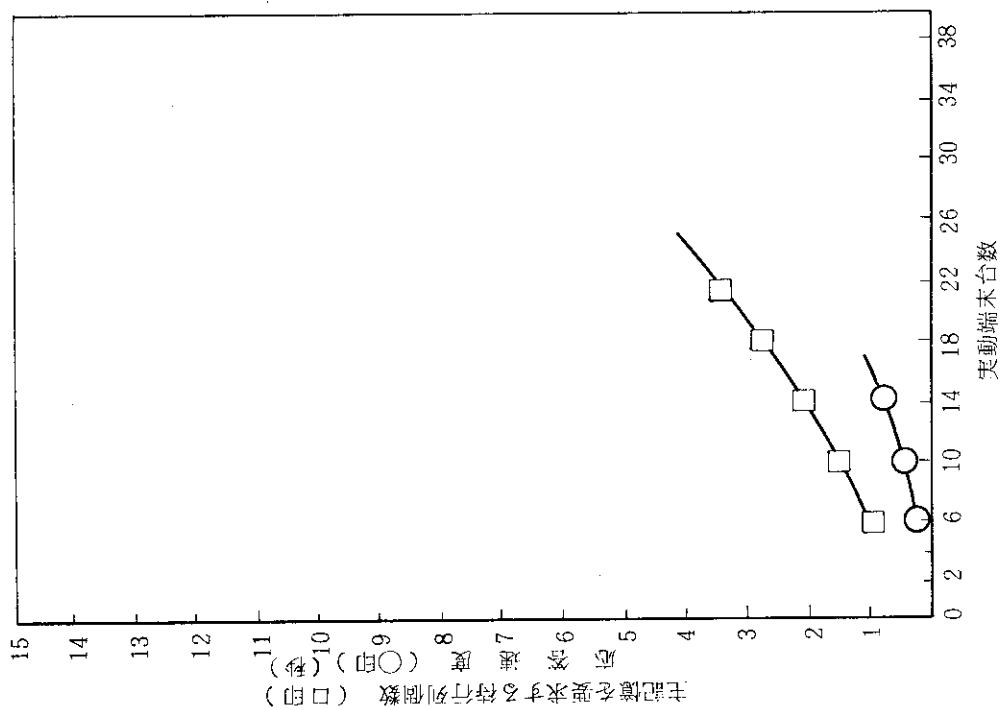


Fig. 4.4 Terminal Response Time 4  
( UT-2D Model )

### 5. 解析モデルによるバッチ処理のスループット解析

ここではバッチ処理のスループット改善を目的とした解析モデルについて述べよう (図 5.1)。入出力は実際には磁気ディスクを経由するからマルチプレキサ・チャネル経由のものはわずかである。したがって、これは循環待行列モデル<sup>4)</sup>といわれているものに等しい。

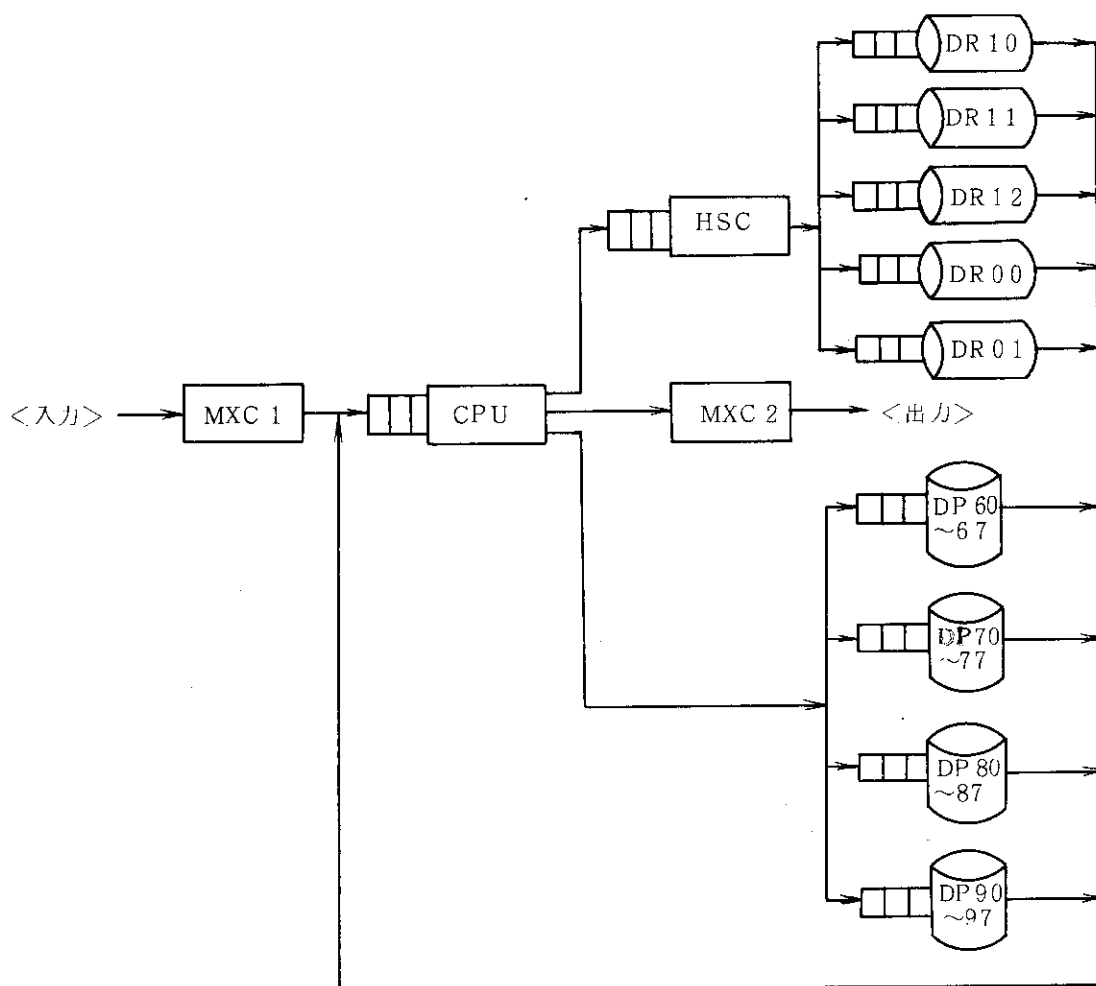


Fig.5.1 Model for Batch Processing

#### 5.1 テスト用ジョブ (98ジョブ)

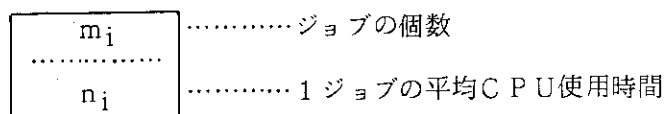
スループット測定は表 5.1 のジョブを使用している。これは FACOM 230 - 60 の 2 システムを使用していた時期にベンチマーク・テスト用に作成したものである。会計情報をもとにジョ

ブ・パターンを作り、このパターンに合う実際の利用者ジョブを探して 100 ジョブとした。これらのジョブは計算機メーカ各社で計算する便宜上、入出力操作は含んでいない。この 100 ジョブのうち 2 ジョブは FACOM 230 - 75 では FORTRAN の初期値設定の違いからジョブ・アボートになることがわかった。それで現在はその 2 ジョブを除いた 98 ジョブを使用している。

この 98 ジョブ・テストは FACOM 230 - 75 の A システムでおこなった。ただし、リモート・バッチと TSS 処理は停止させている。

Table 5.1 Job Pattern for Thruput Test

タイム コア	1分以下	2分''''	5分''''	15分''''	30分''''	30分以上
32 K	30	3	2	1	1	
	0.2分	1.4	3.4	8.6	22.4	
64 K	31	7	6	3	1	1
	0.4	1.7	3.9	7.5	27.2	48.1
96 K	2	1	1	2	1	1
	0.2	1.6	3.5	8.1	26.8	47.7
128 K	1	1	1	1	1	1
	0.9	1.6	2.8	11.9	21.8	34.1



(注) タイムは FACOM 230 - 60 の CPU 時間  
コアの単位は語数である。

## 5.2 解析モデル

このモデルによる経過時間の計算方法を述べておこう。

- (1) CPU, HSC (高速セクタ・チャネル) などのステーションで、その入力時間間隔、処理時間、装置利用率、入力回数を PANSY 情報から求める。
- (2) 各ステーションとも入力時間間隔は指数分布と仮定し、処理時間は変動係数に応じて超指数、指数、アーラン、定常分布を仮定する。
- (3) ヒンチン-ポラツェクの公式から各ステーションの待ち時間を求め、これと入力回数からジョブ当りの平均経過時間を求める (モニタのオーバーヘッドはユーザ・ジョブの経過時間に含まれている)。



- (4) 経過時間の計算は図 5.2 の簡略化したものでおこなっている。ここで BMC (多重マルチプレキサ・チャネル) のかわりに磁気ディスクを使うのは BMC の多重効果の測定がむずかしかったからである。
- (5) 全ジョブの経過時間を多重度で割って 98 ジョブの経過時間を求めた。
- (6) ジョブのロールアウトは発生しないので主記憶ステーションは設けない。

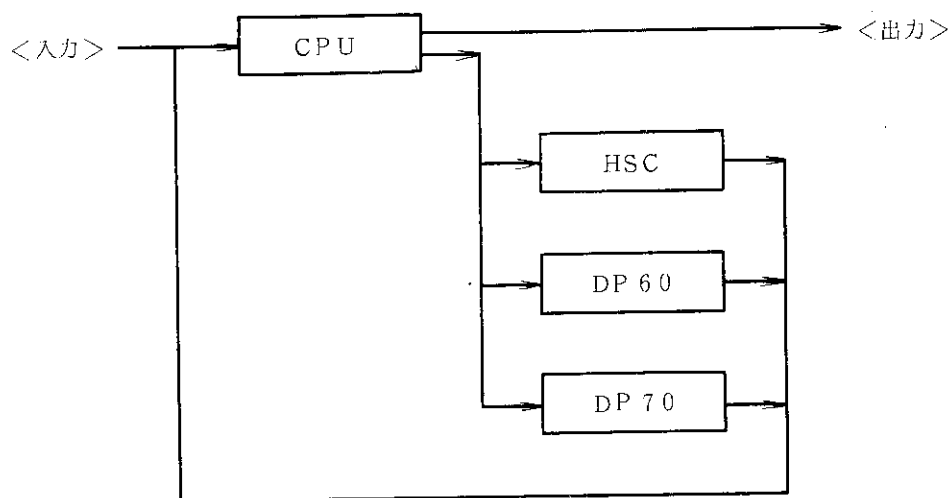


Fig. 5.2 Simplified Model for Batch Processing

表 5.2 はこうして求めた計算値と実測値である。  
この表からわかることはつぎのようなことである。

Table 5.2 Observed vs. Calculated Elapsed Time, 2 CPU

2 CPU 多重度	経過時間 (単位: 分)		
	実測値	計算値 1	計算値 2
2	64	73	82
4	46	44	49
6	44	37	42

(注) 計算値 1 = 平均ジョブ経過時間 × 98 ジョブ / ジョブ多重度  
 計算値 2 = 平均ジョブ経過時間 × 98 ジョブ / ジョブ・ステップ多重度

- (1) このような計算モデルでも比較的良好に現実を近似できる。
- (2) 4多重、6多重では計算値は実際値よりも低め（速い）にでている。これはジョブ・ステップ多重度が実際に起動されているユーザ・ジョブのタスク多重度よりも大きな値になっていることによる。ジョブ・ステップ多重度かわりに、HOPEで収集できる平均タスク多重度を使用すれば計算値は実測値に近くなるのがわかっているが、今回はデータ整理がまにあわなかった。
- (3) 2多重で実測値と計算値の差が大きいのは上述(2)の理由のほかに、磁気ディスク2系列への並行アクセス時間を直列に計算していることによる。多重度が上げればこの並列アクセス時間の影響は小さくなっていく。

### 5.3 性能の予測

このモデルを使って現実の計算機性能の内挿をすることも目的のひとつである。表5.3はCPU2台を1台に落したときの実測値、および予測計算値である。この予測計算値を求めるにあたっては、2台のCPUでテストした時のデータを利用し、CPU利用率、分布は2CPUの時と同じ、1回の入力処理のCPU時間のみ2CPUの時の2倍とした。結果は実測値と比較的良好に一致していることがわかる。

Table 5.3 Observed vs. Calculated Elapsed Time, 1 CPU

1 CPU 多重度	経過時間 (単位: 分)		
	実測値	計算値1	計算値2
2	78	101	115
4	65	64	70
6	70	55	64

(注) 計算値は2CPUの場合のデータを使用して求めた。

さて、このモデルからいくつかの外挿（ごく近い将来の計算機技術の進歩によってもたらされる性能向上の予測）をこころみてもよい。

図5.3はCPU性能が向上した場合の98ジョブ経過時間の短縮を示す。

図5.4は磁気ドラム、磁気ディスク（これらをHSCで代表させた）をアクセス時間と転送時間の速い装置で置き換えたときの性能向上を示す。

図5.5は以上の効果を一緒にした時の経過時間の短縮を示している。実際にはさまざまな資源要求のぶつかり合いが生じて、このようにはならないのであるが一応の目安にはなる。

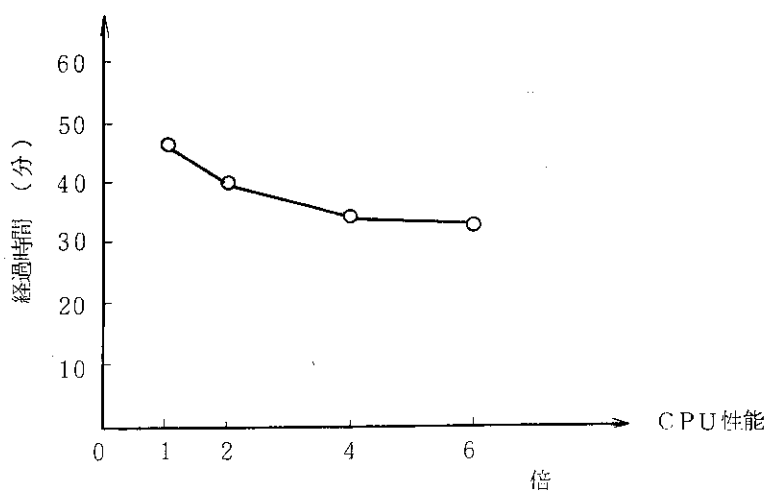


Fig. 5.3 Decreasement of Elapsed Time vs. CPU Speed

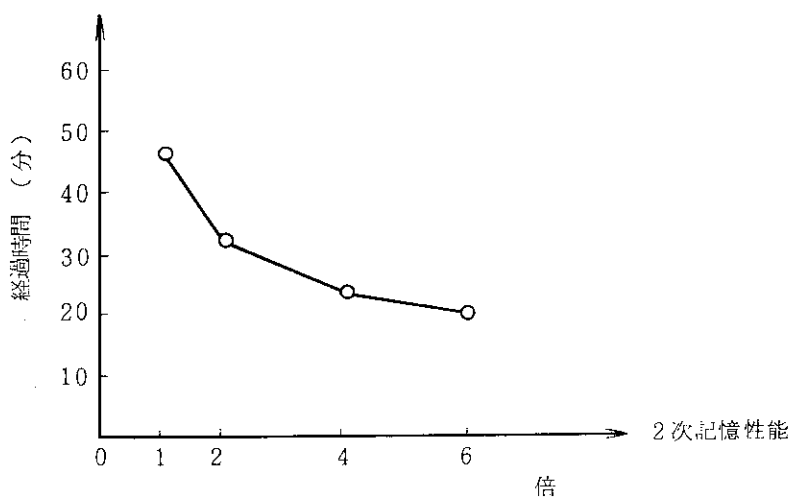


Fig. 5.4 Decreasement of Elapsed Time vs. Drum, Disk Speed

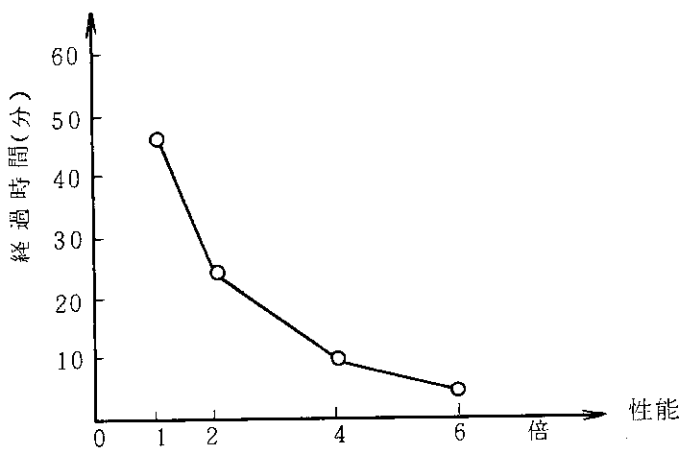


Fig. 5.5 Decreasement of Elapsed Time vs. CPU, Drum and Disk Speed

## 6. 性能評価による成果

3, 4, 5 章での性能評価の過程で明らかとなった主な問題点とその対策は表 6.1 のとおりである (カッコ内は発見者, 分析者)。

(Table 6.1 Troubles discovered and their remedies)

問 題	性能向上のネックがシステム・ドラムの過負荷にあった (SE, OS課, 原研)
影 響	端末への応答の遅れ
原 因	(1) ロールアウト・ファイルの不足 (2) OSのロールアウト制御の弱点 (3) 呼び出し頻度の高い非常駐タスクの存在
対 策	(1) ハードウェアの増設 (2) ロールアウト制御の改善 (3) 呼び出し頻度の高いタスクの常駐化
問 題	TSSでの重い(大型)ジョブの制御がうまくいかない (SE, OS課, 原研)
影 響	端末への応答の遅れ
原 因	TSSの設計が比較的軽いジョブを前提にされており, 重いジョブの負荷が増すにしたがってロールアウト制御が限界を超えてシステム全体の効率を落す。
対 策	(1) TSSコマンド専用域を設ける (2) ロールアウト・ファイルを軽いジョブと重いジョブで分離する (3) ユーザ・ジョブのダイナミックな制御をする
問 題	適切な入出力バッファ(ユーザ・ジョブ)の選択をしていなかった (SE)
影 響	入出力操作完了待ちのジョブが多くCPU使用率が上がらない。
原 因	ユーザ・ジョブのバッファ・サイズ, レコード・サイズの選択が悪く, 入出力操作が不必要に多くなる。
対 策	マニュアル作成, 講習会開催などを指導
問 題	ワーク・ファイルが特定磁気ディスク・ボリュームに偏在する (SE)
影 響	ワーク・ファイルを取りづらくなり, ジョブ処理が遅れる。
原 因	原研計算センターのジョブ制御文で特定運用グループ(特定ボリュームになりやすい)の指定があった。
対 策	ジョブ制御文を修正した。
問 題	Aシステムの負荷が大きい (SE)
影 響	TSS応答が遅い。バッチ・ジョブが進行しない。
原 因	Aシステムのバッチ多重度が大きい。
対 策	セミ・オープンバッチ処理をBシステムに移した。

## 7. 効率評価の考え方

### 7.1 効率評価のレベル

最後に蛇足ながら効率評価という用語の意味を考えておこう。そうするのは第一には効率評価という言葉がさまざまな局面に用いられており混乱を招きやすいからである。第二には効率評価についての筆者らの考え（定義）を明確にし、読者の理解を容易にするためである。

筆者らの考えによれば効率評価には、図 7.1 で示される階層づけられた三段階、すなわち A、B、C ブロック・レベルの効率評価がある。

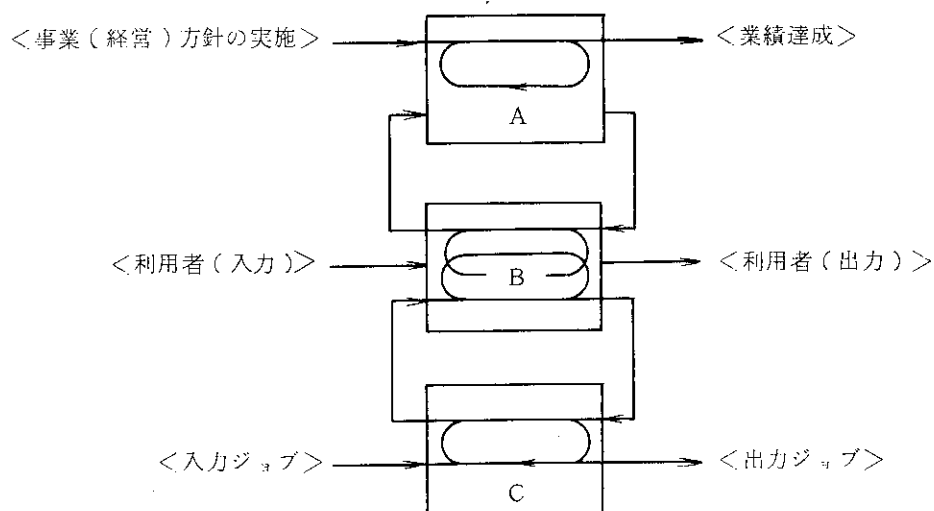


Fig. 7.1 Hierarchical Structure of Computer Cost/Benefit Evaluation

ここで、A ブロック・レベルの効率評価とは、コンピュータの利用において企業、行政、研究機関の事業（経営）方針を直接反映させ得るようなものを言う。この A ブロック・レベルでの効率評価の得失を挙げれば、

#### 利点

事業（経営）方針に密着したコンピュータ利用が可能。

企業、行政、研究機関の各部門の合意のもとにコンピュータを運用できる。

#### 欠点

得られる利益を評価することが難しい。

評価できる場合にも作業が大がかりになる。

複数プロジェクト間の利害の対立（順位の設定などによる）の調整が難しい。

Bブロック・レベルの効率評価の第一目標は計算機利用による省力化、第二の目標はCブロックの計算機の性能を最高にしBブロックの利用者が早くこのブロックから脱出できるようにすることである。（生産性の向上と省力化）。

このような観点から種々の機能をもった計算機を選択することもこのレベルの問題である。このレベルに効率評価を限定することの得失を挙げれば、

#### 利点

効率評価を比較的小規模におこなえる。

電算化効果（原価計算）を測定しやすい。

計算機の性能、機能および電算室（計算センタ）の運用方法の適否を確定しやすい。

#### 欠点

事業（経営）方針についての考慮がない。

現在の需要の外挿予測はできるが、それが将来の需要とは必ずしも一致しない。

Cブロック・レベルの効率評価は従来から言われている計算機性能評価がこれにあたる。このレベルでの効率評価の得失は容易に想像できることである。

このようにして効率評価にはさまざまなレベルがあり、それぞれのレベルに依存した評価の手法や考え方がある。従来計算機の運用について、これらのレベルを考慮した統一的方法はなかった。ところが最近これら3つのレベルを統一的に制御する計算機運用の手法を実現するソフトウェアが現われはじめた。IBM社のSRM（System Resource Manager）<sup>8)</sup>とこれに続く各計算機メーカーの同種のソフトウェアがそれである。FACOM 230-75の場合はHOPE（Heuristic Optimize Evaluation System）が、M-200の場合はSDMがこれにあたる。

これらソフトウェアはジョブをいくつかのサービス・グループに分け、計算機システムがそれらのグループに与える資源を動的に制御する。この種の制御機能があれば計算機の運用に事業（経営）方針を反映させることができる。すなわち、システムの混雑時には、高順位を持つ事業方針に関係するジョブには変りなくサービスを与え、そのジョブの回転時間を一定に保つことができる。

この技術の特徴はブロックA、B、Cを密接に結びつけることである。しかし電算室にとってその運用はかなり難しい。従来電算室は組織のなかの小さな支分組織であり、その思考と行動はブロックB、Cの枠を出ない。ところがB、Cブロックのレベルでの性能評価では計算機システムを使用する利用者集団は優先順位のつかない均質なものとみなされる。そこに事業方針の反映はない。

これはB、Cブロック・レベルに枠を限定した性能評価手法の弱点である。たとえば、Bレベルで評価した「磁気ディスク対磁気テープの費用効果比」<sup>7)</sup>を例にとると、この評価の中では「ユーザの仕事時間遅れ」を単純作業の労働経費で換算している。これはB、Cレベルでは悪くないが、Aレベルから見るとかなり荒っぽいやり方となる。その理由は(1)ユーザの時間遅れは

単純作業労働量と等価ではない。(2)計算機利用時間の遅れによってすべての事業（経営）方針が等価な損害を受けることはない、ことなどによる。

このようにして議論は自然とA, Bブロック・レベルの効率評価になってゆくが、これに応えられる手法やデータを持っていないということがわれわれの正直な本音ではあるまいか。例をあげれば、地域計画策定モデルの計算が地域開発にどれだけ有効であるかを定量的に評価することはできないし、またモデルの計算に要する計算機費用が地域開発に要する費用に占める割合等を確定することも難しい。

またブロックAのレベルにかかわる効率評価の出来、不出来は、電算室の規模、その属する組織のなかでの位置によって決まる。

表7.1はこの関係を示したものである。この表は、電算室が効率評価をおこなう場合の「やりやすさ、やりにくさ」の範囲を筆者らの経験をもとに大雑把に表現したもので、読者はこの表が電算室のおこなう効率評価のあるべき姿（つまりは責任と権限の範囲）を示したのではないことに注意されたい。

Table 7.1 Scale, Position of Data Processing Division and Levels of Cost/Benefit Evaluation

人員, 計算機 組織上の位置	超大型	大型	中型	小型
上位	A, B, C	B, C	B, C	A, B
中位	B, C	B, C	C	B
下位	C	C	C	C

筆者らの考えを表7.1の行列の第1行を例にとって説明しよう。

電算室の組織上の位置が上位、人員と計算機の規模が超大型であれば、その組織の事業（経営）方針遂行のためにA, Bブロック・レベルの効率評価をおこなうに十分な期間と人材、他部門の協力を得られよう。小型でありながら組織上の位置が上位にある電算室は、それだけ電算化に期待がかけられているわけであるからA, Bブロック・レベルの効率評価をおこなうに必要な他部門の協力が得られよう。

これに反し大型、中型の規模では、Aブロック・レベルの効率評価そのものが難しいこともあってB, Cレベルにとどまっているのが現状のようだ。

筆者らの電算室は規模において計算機は超大型、人員は中型、あるいは小型である。組織上は中位、あるいは下位であるが、幸いにしてユーザの良き理解と協力が得られるのでBブロック・レベルの効率評価は可能である。

これまで計算機の性能評価という場合は、一般にCブロック・レベルの評価を指していることが多かった。

最近IBM社などではA, B, Cブロックを組み合わせた販売戦略を取りはじめている。Aブロック

・レベルの効率評価はユーザの問題であるが、このレベルの評価の結果によっては新しい機能や高度の計算能力が必要になることがあり、機敏に対応できるメーカーやベンチャー・ビジネスの活躍の場となっているようである。

いまや官公庁、地方自治体などが電算室の予算を要求するにあたってはAブロック・レベルの具体的な成果を、会計検査や内部監査にあたってはB、Cブロック・レベルの努力を問われる時代に移りつつあるといえよう。

それにもかかわらず、われわれはこれに対応できる有力な道具も手法ももっていない。(Cブロック・レベル以外は)。したがってメーカー、ユーザが一致協力して道具や手法を開発することも急務であるが、その際には表7.1で示したように電算室自体でできる範囲を念頭において計画を立てる必要がある。

もっとも電算室の組織上の位置づけがどうであれ、Aブロック・レベルの効率評価(というよりは効率予測)をおこなう機会はある。計算機の導入やリプレイス時の需要予測がそれである。

計算機を利用する部門を代表するメンバから成る作業グループによって、需要予測をおこなうこと、および予測された需要に順位をつけ、それに見合った計算機を導入することは一種の効率評価にはかならない。

## 7.2 Aレベル効率評価の例

この種の効率評価の例として当研究所では昭和54年1月から4月にかけて東海研究所で設置された大型電子計算機交換検討会の例をあげることができる。

この検討会では将来の計算需要の予測、望ましい計算機構成などが検討された。この検討会で採用された計算機構成は図7.2のようになっている。A、B、Cの3システムは、すべてBのシステムに存在するソフトウェアによって制御され、利用者は任意の入出力装置からどのシステムへもアクセスすることができる。

このシステム構成とさきに述べた効率評価の考え方を述べた図7.1と対比させてみたのがつぎの図7.3である。

この図7.3からわかるように、3つのシステムはそれぞれ違った考え方で運用される。

それぞれの昼間の運用方法の特徴を列記すると、

- ・ Cシステムの運用は従来の方法を踏襲する。すなわちより少ない計算機資源を使用するジョブは優先権が高くなり、早い処理が期待できる。このシステムにあたっては、より多くの計算機資源、特に演算装置と主記憶をより多く使用するジョブは悪となる。
- ・ Aシステムは従来は存在しなかった考え方で運用される。このシステムは長い演算装置時間、大量の主記憶を使用するジョブで緊急を要するもの、優先度の高い研究に関連するものから順に処理する。このシステムにあっては計算機資源を多く使用することが悪ではない。このシステムで処理される計算コードの大部分は素姓の知れているものであり、それらの計算時間の短縮、主記憶容量の縮小などについての研究は絶えずおこなわれているからである。
- ・ Bシステムは計算機利用者の待ち時間を最短にすることを主目標に運用される。このシステムはタイムシェアリング処理と3システムの制御に使用される。



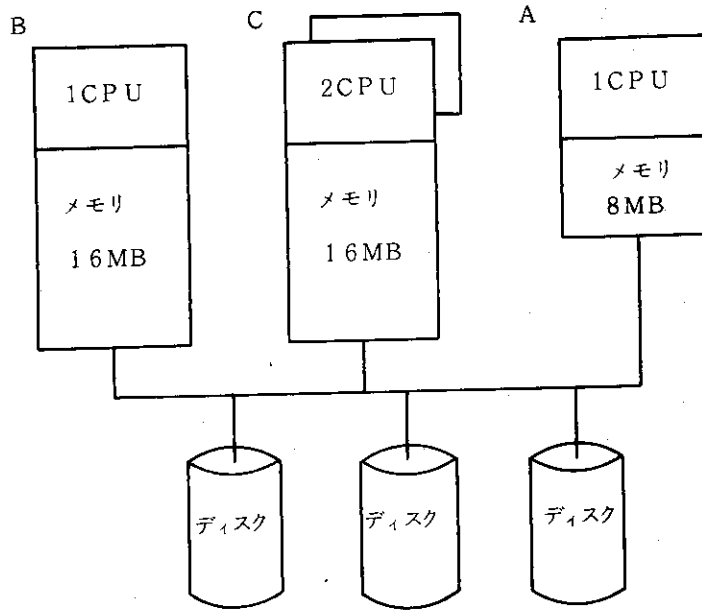


Fig. 7.2 Distributed Processing by 3 Systems

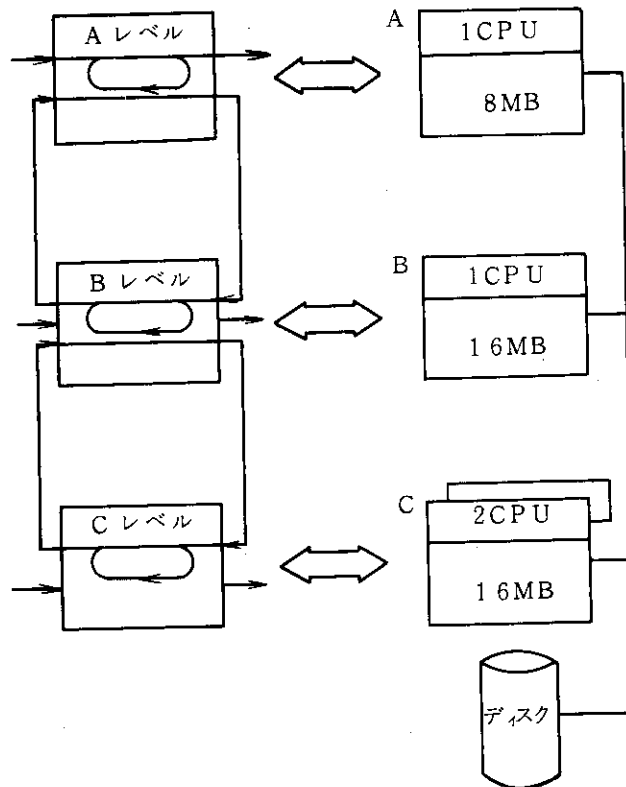


Fig. 7.3 Hierarchical Structure of Cost /Benefit Evaluation vs. Computer Systems

これらのシステムのハードウェアとソフトウェアは、いずれも運用方法に合わせて調整される。夜間においては、BシステムはAシステムと同じく大型ジョブの処理をおこない、Cシステムは夜間ジョブとして昼間投入されていた中小型ジョブが処理される。

以上が検討会で採用された分散型のシステム構成である。

これに対し従来と同じ集中型のシステム構成を考へることもできる(図7.4)。

不特定多数の利用者のジョブを処理する計算センタにおいて、複数のシステムを一元的に制御しながら、このように異った、それぞれが相矛盾する運用原則で稼働させる例は世界でも珍しいのではないだろうか。

しかし今後はこのような事例が多くなるであろう。ただこの一元制御のソフトウェアの稼働実績が少ないために、安定した運用をおこなうことが提供メーカーと計算センタ双方の留意すべき問題として残っている。

集中型を採用すればオペレーティング・システムは1つあればよいので、その分だけ主記憶と2次記憶装置が少なくすむ。また集中型を採用したときの運用方法はどのようなであろうか。研究の優先順位に応じたサービスをおこなうためには、IBMのSRM(System Resource Manager) 流の運用方法を採用することになる。<sup>8)</sup>

SRMの考え方は図7.5からわかるように、ジョブをいくつかのグループに分け、計算機の混み具合に応じてグループ毎にサービスを変化させてジョブを処理する。しかし、このとき適正なサービス率を決定することは、計算機技術の面からみても、研究所の組織から考へても難しい。

分散型のシステム構成ではこの種のソフトウェアに頼る度合がずっと少なくなる。こうして分散型システムのもつ

- ・ 機能(ジョブ処理方法)の分散
- ・ 拡張の容易性
- ・ オペレーションの省力化

などの利点を重視して分散型のシステム構成が検討会案として採用されることになった。

これら3システムを継続的に稼働させるに足る潜在的な計算需要が現在も存在することがわかっている。また検討会における需要調査によって将来も現在に倍する計算需要が見込まれている。

このような考え方に沿って計算機構成を決めることはAブロック・レベルの効率評価といえるであろう。

原研がこの数年間におこなった計算機効率評価の作業のうち、大きなものをこのような考え方によって分類すると以下ようになる。

#### [Aブロック・レベル]

- ・ 各種需要調査とそのまとめ
- ・ 計算機システム構成の決定
- ・ TSS端末機器の費用効果比判定
- ・ 各種機器のレンタル/買取り損益分岐点の計算

#### [Bブロック・レベル]

- ・ 入出力装置適正台数の決定
- ・ 磁気テープ対磁気ディスクの費用効果比計算

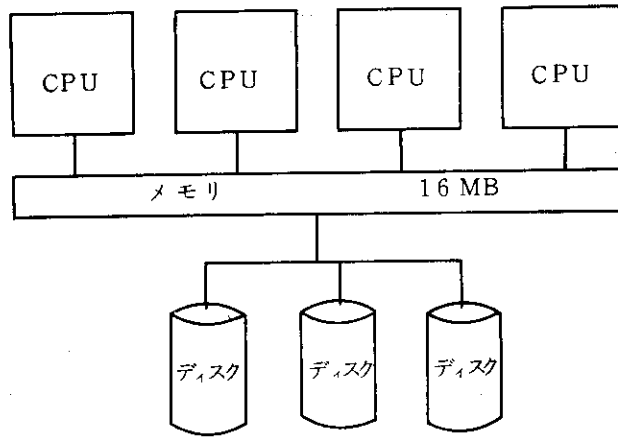


Fig. 7.4 Centralized Processing by One System

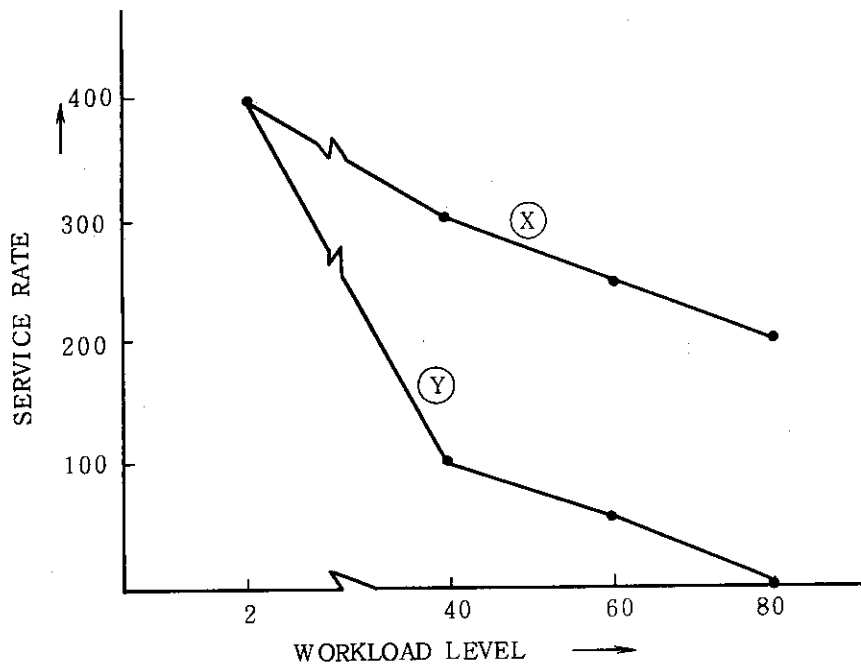


Fig. 7.5 Service Principle by SRM

- TSSの応答解析
- ファイル・ブロック・サイズの適正な選択

[Cブロック・レベル]

- バッチ処理のスループット解析
- 各種システム・パラメータの適切な選択
- 各種装置使用率の向上

[A, B, Cブロック・レベルの混合]

- SRMソフトを重視しない計算機システム構成と運用
- 大容量記憶装置の費用効果比計算と導入の決定
- レーザ・プリンタ費用効果比計算と低速レーザ・プリンタ導入の決定

この報告書は以上の項目のうち、TSS応答解析、バッチ処理のスループット解析についてまとめたものである。

## 8. おわりに

この性能評価作業と並行して原研計算センタ運用担当係と富士通SEグループによってファイルの割付け、システム定義パラメータの選定、ファイルのレコード・サイズの選択、その他についての見直し作業が精力的に進められた。これによっても計算機の性能は向上したものであるが、ここでは筆者らのおこなった性能評価についての感想を述べよう。

### 8.1 Cブロック・レベルの性能評価について

#### (1) 計算機使用形態への柔軟な対応

さきに述べたように、Cブロック・レベルの性能評価の過程で発見されたさまざまな問題点は、使用記憶容量、CPU時間とも大型のジョブが多い原研の使用形態に起因している。他の計算施設では、これとは全く逆の使用形態となるところもあろう。これらの相反する使用法に対するモニタの制御方式は異なるはずである。メーカはこれらの個々の使用方法に対して一品生産的モニタを作ることもできまい。モニタの制御方式が環境条件によって変化できるようにしなければならない。

#### (2) サービス・グループの設定かジョブ処理の分散か

モニタの制御方式が環境条件によって動的に変えることも必要であるが、その上さらに大切なのは、その変り方もさまざまに、そしてまえもって定められた方法で指定できることである。この指定は電算室が毎日でも気軽に変更できることが基本条件である。

これらを筆者らが当面している問題を例にとって説明しておこう。

原研計算センタのA、B両システムに投入されるジョブはその要求する資源（主記憶量、CPU時間など）の大きさに応じて減点法でジョブ優先権が与えられる。午前中にシステムに投入された優先権の低いジョブは昼休み近くまで起動されない。モニタがCブロック・レベルでジョブのスループットを最大にするよう働くからである。昼休みになって起動された大型ジョブは午後の混雑時にもシステムに継続して残り、システムのスループットを低下させて多数の利用者に迷惑をおよぼす（Bブロック・レベルの問題である）。

これとは逆に研究所の事業方針に沿った大型ジョブの回転時間を2～3回/昼間確保しなければならない（Aブロック・レベルの問題）。

これらの相反する要求を1システムで調和させるためにはどうしてもSRM（筆者らにとってはHOPE）流の制御機構が必要となる。

原研では、次期の計算機システムではそれぞれが運用方針の異なる3システムを用意することによってSRM（富士通製品ではSDM）流のソフトへの依存度を低めようとしている。

## 8.2 Bブロック・レベルの効率評価について

このレベルの効率評価は最近とくに重要視されるようになった。これからは電算室費用の原価計算なども必要となろうが、問題はそれを考え、かつ実行する人材に不足していることである。これに対応するための教育が電算室要員にこれまでなされていなかった企業、機関も多いのではあるまいか。参考までに筆者らがいくつかの幸運にめぐまれてたどってきたこの数年間の過程を述べておこう。

### 計算センタ

- (1) 外部の講師を招いてCブロック・レベルの性能評価解説書の輪講をおこなった。
- (2) 経営工学科（修士）卒の新入職員（1名）が配置された。
- (3) 効率評価（Cブロック・レベル）が必要だという認識が計算センタ内にあり、作業の手配が順調にいった。

### 利用者

- (1) 計算機は研究の道具であるという意識があり、Bレベルの調査に協力的である。
- (2) Aブロック・レベルの評価は、研究業績への計算機の寄与率の算定などがむずかしい。

### メーカー

- (1) 担当OS課がOSの改良に熱心であり、また動的制御機構HOPEの開発時期と一致していた。
- (2) 担当システム部（科学システム開発部）が性能評価の必要性を認め、カスタマ・エンジニアらと共同で生データの収集と分析、評価をおこなった。

以上のようによい条件にめぐまれていたわけであるが、効率評価はまだはじまったばかりの段階である。生データの収集は非常に時間と手間と計算機時間がかかった。計算機がなければ仕事ができないことがわかっているし、手間がかかるのでいまさら効率評価などしないという企業もあるようだが、それはそれでひとつの見識であると思う。

あまり手間をかけないで効率評価をおこなうためには、他の技術と同様に他人の肩に乗っかることが大切である。すなわち他の電算室がおこなった評価の方式を互いに融通し合って無駄な作業を省くことが必要である。

## 8.3 Aブロック・レベルの効率評価について

このレベルの効率評価はなかなかむずかしい。従来は電算室は組織のなかでは中位にあって、Bブロック・レベルの効率評価をおこなえばよかったが、最近の情勢ではそのレベルにのみとどまることがむずかしくなってきた。

Aブロック・レベルの効率評価のもっとも簡単、かつ一般的方法は計算機の導入時、リプレイス時に関係部門の代表を集めておこなう作業委員会によるものであろう。

## 謝 辞

日本原子力研究所計算センタで計算機性能評価を進めるにあたって、つぎの方々に大変お世話になった。

茨城大学の磯田和男教授にはこの数年間タイミングよく性能評価の解説書を教材とし、輪講を指導していただいた。

富士通株式会社科学システム部部長植村満夫氏とOS部部長秋山文男氏、原研計算センタ室長平川隆氏には性能評価推進の体制作りをやっていただいた。

富士通科学システム部の川開清昭氏を中心とするSEグループの諸氏には性能評価のためのシステム修正、改善作業をしていただいた。小笠原、佐藤両氏を中心とするCEグループにはマイクロ・サム、カウンタのセットなどでお世話になった。

原研計算センタの斉藤氏には作業手配などで、石黒、中村、富山、小沼、山崎の各氏には計算モデルの検討、データ整理、プログラムの作成などでお世話になった。

深く感謝する次第です。

## 参 考 文 献

- [1] Baskett, F. et al., Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, J. ACM 22, 2 (April 1975), 248 - 260.
- [2] J. マーチン著, 北原安定訳: リアルタイム・システム, 日本経営出版会 (1975)
- [3] Hellerman, H. and Conroy, T., Computer System Performance, McGraw-Hill, New York (1975).
- [4] Coffman, E. and Denning, P., Operating System Theory, Prentice-Hall, Englewood Cliffs, New Jersey (1973).
- [5] Shemer, J. and Heying, D., Performance modeling and empirical measurements in a system designed for batch and time-sharing users, Fall Joint Conf. (1969), 17-26.
- [6] Brown, R. et al., Memory Management and Response Time, C. ACM 20, 3 (March 1977), 153 - 165.
- [7] 藤井実: 磁気ディスク対磁気テープの費用効果比, JAERI-M 7424, (1977), 日本原子力研究所。
- [8] OS/VS2 System Programming Library: Initialization and Tuning Guide, VS2 Release 3.7, GC 28-0681-2, IBM
- [9] Shemer, J. E., Some Mathematical Considerations of Time-Sharing Scheduling Algorithms, J. ACM 14, 2 (April 1967) 262-272.
- [10] Buzen, J. P., Computational Algorithms for Closed Queuing Net -

## 謝 辞

日本原子力研究所計算センタで計算機性能評価を進めるにあたって、つぎの方々に大変お世話になった。

茨城大学の磯田和男教授にはこの数年間タイミングよく性能評価の解説書を教材とし、輪講を指導していただいた。

富士通株式会社科学システム部部長植村満夫氏とOS部部長秋山文男氏、原研計算センタ室長平川隆氏には性能評価推進の体制作りをやっていただいた。

富士通科学システム部の川開清昭氏を中心とするSEグループの諸氏には性能評価のためのシステム修正、改善作業をしていただいた。小笠原、佐藤両氏を中心とするCEグループにはマイクロ・サム、カウンタのセットなどでお世話になった。

原研計算センタの斉藤氏には作業手配などで、石黒、中村、富山、小沼、山崎の各氏には計算モデルの検討、データ整理、プログラムの作成などでお世話になった。

深く感謝する次第です。

## 参 考 文 献

- [1] Baskett, F. et al., Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, J. ACM 22, 2 (April 1975), 248 - 260.
- [2] J. マーチン著, 北原安定訳: リアルタイム・システム, 日本経営出版会 (1975)
- [3] Hellerman, H. and Conroy, T., Computer System Performance, McGraw-Hill, New York (1975).
- [4] Coffman, E. and Denning, P., Operating System Theory, Prentice-Hall, Englewood Cliffs, New Jersey (1973).
- [5] Shemer, J. and Heying, D., Performance modeling and empirical measurements in a system designed for batch and time-sharing users, Fall Joint Conf. (1969), 17 - 26.
- [6] Brown, R. et al, Memory Management and Response Time, C. ACM 20, 3 (March 1977), 153 - 165.
- [7] 藤井実: 磁気ディスク対磁気テープの費用効果比, JAERI - M 7424, (1977), 日本原子力研究所。
- [8] OS/VS2 System Programming Library: Initialization and Tuning Guide, VS2 Release 3.7, GC 28 - 0681 - 2, IBM
- [9] Shemer, J. E., Some Mathematical Considerations of Time - Sharing Scheduling Algorithms, J. ACM 14, 2 (April 1967) 262 - 272.
- [10] Buzen, J. P., Computational Algorithms for Closed Queuing Net -



works with Exponential Servers, Comm. ACM, 16, 9 (Sept. 1973),  
527 - 531.

- 〔11〕 浅井, 藤井, 日本原子力研究所計算センタにおける効率評価, 電子計算組織利用の効率  
評価システムの調査研究, 地方自治情報センター, 昭和53年2月.
- 〔12〕 田淵, 他, HOPE説明書, 1977年10月, 富士通.

付録A BTMモデルの計算法とパラメータ

記号の説明

- N : 起動されているオンライン・ジョブ数, すなわち実動端末台数。
- $\lambda$  : オンラインの平均入力率。
- $\mu$  : オンライン要求がCPUによって受ける平均サービス率。  $1/\mu$  は個々のオンライン要求をCPUが処理するに要する平均処理時間である。
- $\bar{S}$  : 古いジョブを主記憶から追い出し (swap out), 新しいジョブを入れる (swap in) に要する時間。ジョブの大きさと2次記憶装置の速さに依存する。
- $q_R$  : オンライン要求に割当てられるタイム・クワンタム (単位時間)。
- $q_B$  : バッチ処理要求に与えられるタイム・クワンタム。
- $\bar{m}$  : ひとつのオンライン要求の処理が完了し, 次のオンライン要求の処理が始まるまでに費されるモニタのオーバーヘッド時間。

到着時間間隔の分布関数を  $A(x)$ , サービス時間の分布関数を  $B(x)$ , 起動されている端末数を  $N$ , 時刻  $t$  において  $n$  個のジョブが処理を待っている確率を  $\rho_n(t)$  とすると,

$$A(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

$$B(t) = \begin{cases} 1 - e^{-\mu t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

$$\frac{d\rho_n(t)}{dt} = \begin{cases} -N\lambda\rho_0(t) + \mu P_R[R(t)]\rho_1(t), & n=0 \text{ のとき,} \\ -[(N-n)\lambda + \mu P_R[R(t)]]\rho_n(t) \\ + (N-n+1)\lambda\rho_{n-1}(t) \\ + \mu P_R[R(t)]\rho_{n+1}(t), & 0 < n < N \text{ のとき,} \\ -\mu P_R[R(t)]\rho_N(t) + \lambda\rho_{N-1}(t), & n=N \text{ のとき,} \end{cases}$$

ここで  $P_R[R(t)]$  は, 時刻  $t$  において計算機がオンライン要求を処理している確率を示す。定常状態の確率は  $d\rho_n(t)/dt = 0$  において

$$\rho_n = \frac{N!}{(N-n)!} \left( \frac{\lambda}{\mu P_R[R]} \right)^n \rho_0$$

となる。ここで

$$P_R[R] = \lim_{t \rightarrow \infty} P_R[R(t)],$$

$$\rho_0 = \frac{1}{\left[1 + \sum_{n=1}^N \frac{N!}{(N-n)!} \left(\frac{\lambda}{\mu P_T(R)}\right)^n\right]}$$

$P_T(R)$ の値はオンライン・ジョブに継続して割当てるタイム・クワントムの中に経過する時間を考察することによって推定できる。

いま,

$T_K$  : 0回目のオンライン・クワントムの終了からK回目のオンライン・クワントムの完了までに経過した時間,

$\bar{q}_B$  : バッチ・ジョブの平均クワントム,

$\bar{q}_R$  : オンライン・ジョブの推定クワントム,

$(1/N\lambda)$  : つぎのオンライン要求が発生するまでの平均時間,

$\bar{m}$  : バッチ, オンライン切換えサイクル単位にみたモニタの推定オーバーヘッド,

とすると, K回目のタイム・クワントムの完了のあとでオンラインの待ちが空きになっているなら, 次のオンライン・クワントムが完了するまでの時間  $\Delta T_K$  の期待値  $E$  は

$$E[\Delta T_K | n=0] = \frac{1}{N\lambda} + \bar{q}_B + \bar{q}_R + \bar{m}$$

である。K回目のクワントム完了後もオンラインの待ち行列が空いていない場合には確率  $(1 - \rho_0)$  で

$$E[\Delta T_K | n \geq 1] = \bar{q}_B + \bar{q}_R + \bar{m}$$

となる。

いま  $T_B$ ,  $T_R$ ,  $T_m$  を時間  $T_K$  のうち, それぞれバッチ, オンライン, モニタ処理に費す時間とすると,  $K \rightarrow \infty$  につれて

$$T_B/K \rightarrow (\bar{q}_B + \rho_0/N\lambda)$$

$$T_R/K \rightarrow \bar{q}_R$$

$$T_m/K \rightarrow \bar{m}$$

となるはずである。

したがって,

$$\begin{aligned} P_T(R) &= \lim_{K \rightarrow \infty} \left[ \frac{T_R}{T_K} \right] = \lim \left[ \frac{T_R/K}{T_K/K} \right] \\ &= \frac{\bar{q}_R}{\bar{q}_R + \bar{q}_B + \bar{m} + \rho_0 \left( \frac{1}{N\lambda} \right)} \end{aligned}$$

である。 $(\bar{q}_B + \bar{q}_R)$  は本質的には計算のサイクルであるから,

$$P_r [R] = \frac{\bar{q}_R}{f (\bar{q}_B + \bar{q}_R)}$$

ここで  $f$  は  $\rho_0, \rho_1, \dots$  を求めるためのスケール・ファクタで

$$f > \left( 1 + \frac{\bar{m}}{\bar{q}_B + \bar{q}_R} \right)$$

である。  $f$  を  $\Delta f$  だけ増加させながら  $P_r [R]$  と  $\rho_0$  が首尾一貫する  $f$  を求める。

$\rho_0$  を求めたあと、バッチ・ジョブに利用可能な CPU 時間の割合は

$$P_r [B] = \frac{\bar{q}_B + \rho_0 (1/N\lambda)}{\bar{q}_R + \bar{q}_B + \bar{m} + \rho_0 (1/N\lambda)}$$

$\bar{q}_B$  は、S をスワップ時間として

$$\bar{q}_B = \begin{cases} q_B, & S < q_B \\ \bar{S}, & S \geq q_B \end{cases}$$

$\bar{q}_R$  は、サービス時間が指数分布であることから

$$\bar{q}_R = \frac{1}{\mu} (1 - e^{-\mu q_R})$$

である。<sup>9)</sup>

個々のオンライン・ジョブが最初のサービスを受けたあとで、次にサービスを受けるまでの時間の期待値は、

$$E [C] \approx \rho_0 E [T_0] + E [n] (\bar{q}_B + \bar{q}_R + \bar{m}) + q_R,$$

となる。ここで  $E [n] = \sum_{n=1}^N n \rho_n$  である。

$E [T_0]$  を正確にはかることは難しいが、明らかに関係

$$E [T_0] \leq (\bar{q}_B + \bar{m})$$

を満す。

[計算のためのパラメータ設定例]

FACOM 230-75A システムの昭和 52 年 10 月 26 日午後の 2 時間の処理を例とする。

F 230-75 モニタのタイム・クワンタムは 100 ms である。

- CPU 使用率 0.573 PANSY 情報 (モニタの 1 回の idle task の所要時間を 240  $\mu$ s と仮定している。)

バッチ処理に使用したCPU時間	3297 秒	) 会計情報
TSS処理に使用したCPU時間	185 秒	
端末入出力回数	4310 回	
全ジョブの入出力回数	36085 回	

- モニタのオーバーヘッド  $\bar{m}$  は,

$$\bar{m} = (2 \text{ CPU} \times 2 \text{ 時間} \times 3600 \text{ 秒} \times 0.573 - 3297 - 185) \div 36085 = 132 \text{ ms.}$$

- ロールアウト・ファイル

型式番号 F-6621 B の磁気ドラム。半回転に 10 ms を要する。転送レートは  $1.2 \times 10^3$  バイト/ms。

平均処理時間

30.6 ms, うち 27 ms が待ち時間を除く処理時間 (PANSY 指数分布を仮定したモデルから。)

転送時間

$27 - 10 \text{ ms (半回転)} = 17 \text{ ms}$ 。この時間での転送量は  $17 \times 1.2 \times 10^3 \text{ B/ms}$   
 $= 20.4 \text{ KB} \div 4.5 \text{ KW/SIOF}$ 。

高速セクタ・チャネルの平均処理時間は PANSY とモデルから 9.86 ms, うち待ち時間は 3.78 ms。

- TSSジョブについてのデータ

- (1) 容量 1 ~ 30 KW のジョブは 15 セッション, 32% の件数,

平均主記憶 28.7 KW,

$R$  (入力回数) +  $W$  (出力回数) = 471 回,

$R + W/3$  (出力をブロック化して 3 回分を 1 度に出力する) = 157 回。

28.7 KW - TSS 主ルーチン 9 KW  $\div 20$  KW

ロールアウト時間 =  $34 \text{ ms} \times (20 / 4.5) = 151 \text{ ms}$ 。

ここで  $34 \text{ ms} \div 30.6 + 3.78 \text{ ms}$  としている。

$q_R = 1.7 \text{ ms}$ 。(参考文献 9) 参照

- (2) 容量 31 ~ 256 KW のジョブは 32 セッション, 68% の件数。

平均主記憶 71.1 KW,  $R + W = 3840$ ,  $R + W/3 = 1280$ .  $71.1 - 9 \text{ KW} \div 63 \text{ KW}$

ロールアウト時間 =  $34 \text{ ms} \times (63 / 4.5) = 476 \text{ ms}$ 。

$\bar{q}_R = 19.6 \text{ ms}$  から  $q_R = 30 \text{ ms}$ 。

- (3) 容量 1 ~ 256 KW のジョブでみると,

平均主記憶 57.6 KW,  $R + W = 4310$ ,  $R + W/3 = 1440$ 。

ロールアウト時間 =  $34 \text{ ms} \times (48.6 / 4.5) = 367 \text{ ms}$ 。

$\bar{q}_R = 17.6 \text{ ms}$  から  $q_R = 25 \text{ ms}$ 。

- (4) テキスト・ジョブをロールアウトするに 151 ms, つぎのジョブをロールインするに,

$$\frac{471 (R + W)}{4310} \times 151 + \frac{3840}{4310} \times 476 = 441 \text{ ms}$$

かかる。

- (5) 大型ジョブをロールアウトするに 476 ms , つぎのジョブをロールインするに 441 ms かかる。
- (6) TSSのコマンド総数は659, 入力率は1回/11秒=0.09であった。
- (7) 上記(1)の計算でチャンネルを2系統に分けたとき, ドラム, チャンネルとも待ち時間ゼロと仮定すると, テキスト・ジョブのロールアウト+ロールイン時間は,

$$\bar{S} = 2 \times 27 \text{ ms} \times (20 / 4.5) = 240 \text{ ms} ,$$

$$q_R = 2 \text{ ms}$$

となる。

大型ジョブは, あまり変らないとみるべきで,

$$\bar{S} = 2 \times 476 \text{ ms} = 952 \text{ ms}$$

となる。

- (8) 出力をブロック化して3回分をまとめて書き出すと, 回数の減少率は,

$$(R+W/3) / (R+W) = 157 / 471 = 0.33$$

$$\bar{S} = 240 \text{ ms} \times 0.33 = 79 \text{ ms}$$

- (9) 大型ジョブでは

$$\text{TSSコマンド・ログでは } 1280 / 3840 = 0.33 ,$$

$$\text{会計情報では } (2376 + 33349 / 3) / 36085 = 0.38 ,$$

会計情報では1~256KWのジョブや, リモート・バッチ・ジョブも含まれているので, 0.33を採用すると,

$$\bar{S} = 952 \text{ ms} \times 0.33 = 314 \text{ ms} .$$

また平均ジョブでは,

$$\bar{S} = 367 \text{ ms} \times 0.33 = 121 \text{ ms} ,$$

$$q_R = 25 \text{ ms}$$

となる。テキスト・ジョブでは  $\bar{q}_R = 2 \text{ ms} \times 3 = 6 \text{ ms}$  なる  $q_R$  は  $q_R = 7 \text{ ms}$ , 大型ジョブでは  $\bar{q}_R = 60 \text{ ms}$  なる  $q_R$  は  $q_R = 100 \text{ ms}$  となる。

筆者らは以上のような方法でBTMモデルの各種パラメータを定めて計算した。

付録B UT-2Dモデルの計算法

装置のモデルのスループット  $r(n)$  は Gordon and Newell<sup>10)</sup> の方法によって求める。  
以下にその方法について簡単に説明する。

M : サービス時間が指数分布をなす装置の数。

N : システム内で起動されているジョブの数。

$\rho_i(K)$  : i番目の装置でK個のジョブが処理中あるいは、待ちとなっているとき、この装置の使用率。

$P_{ij}$  : 装置 i から装置 j へジョブが移動する確率。

$n_i$  : 装置 i にあるジョブの数。  $n_1 + \dots + n_M = N$  である。

$P(n_1, n_2, \dots, n_M)$  : システムが  $(n_1, n_2, \dots, n_M)$  の状態にある確率。  
このとき  $P(n_1, \dots, n_M)$  は

$$P(n_1, \dots, n_M) = \frac{1}{G(N)} \prod_{i=1}^M (X_i)^{n_i},$$

で求められる。

ここで  $(X_1, X_2, \dots, X_M)$  は方程式

$$\mu_j X_j = \sum_{i=1}^M \mu_i X_i P_{ij}, \quad 1 \leq j \leq M$$

の解である。

$G(N)$  は正規化定数で、式

$$G(N) = \sum_S \prod_{i=1}^M (X_i)^{n_i}$$

で与えられる。ここでSは

$$S = \left\{ (n_1, \dots, n_M) \mid \sum_{i=1}^M n_i = N, n_i \geq 0, \forall i \right\}$$

となる組合せである。

$$P(n_i \geq K) = (X_i)^K G(N-K)/G(N)$$

を利用して

$$E(n_i) = \sum_{K=1}^N (X_i)^K G(N-K)/G(N)$$

で与えられる。

装置モデル (図4.3) のスループット (処理されるジョブ个数/秒)  $r(N)$  はつぎのようにして求める。

$i_1$  : 図4.3における1つのジョブを切替えるに要するループAの回数。

$i_2$  : 図4.3における1つのジョブの1回のテキスト編集に要するループBの回数。

$t_1$  : 図4.3においてCPUを1回通過するに要する時間。

$t_2$  : 図4.3においてHSCを1回通過するに要する時間。

$t(K)$ : 図 4.3 において 1 つのジョブがテキスト編集を受けはじめてから終るまでの時間。

$$i_1 = \frac{17.2 \text{ ms}}{t_{\text{CPU}}(K)} + 1, \quad (\text{ここで } 17.2 \text{ ms は F 230-75 のジョブ切換えに要する CPU 時間})$$

$$i_2 = \frac{12.7 \text{ ms}}{t_{\text{HSC}}(K)} + 1, \quad (\text{ここで } 12.7 \text{ ms はテキスト編集に要する CPU 時間})$$

$$t_1 = t_{\text{CPU}}(K) \left[ 1 + \frac{\rho_{\text{CPU}}(K)}{1 - \rho_{\text{CPU}}(K)} \right],$$

$$t_2 = t_{\text{HSC}}(K) \left[ 1 + \frac{\rho_{\text{HSC}}(K)}{1 - \rho_{\text{HSC}}(K)} \right],$$

$$t(K) = i_1 \times (t_1 + t_2) + i_2 \times t_1.$$

$\rho_{\text{CPU}}(K)$ ,  $\rho_{\text{HSC}}(K)$  は  $K$  の増加につれて変化するが, その値は期待値  $E_{\text{CPU}}(K)$ ,  $E_{\text{HSC}}(K)$  を計算し,

式  $E(K) = \rho_K / (1 - \rho_K)$  から逆算する。

$K = 8$  近辺で計算値と理論値はほぼ一致していた。

$t_{\text{CPU}}(K)$ ,  $t_{\text{HSC}}(K)$  も  $K$  の増加につれて変化するが, ここでは  $K = 8$  近辺の実測値を使用する。