

階層構造モデルによる Multics システムの性能評価

関 野 陽 (日本電気 コンピュータ技術本部)

1. はじめに

本稿では、筆者がMITにおいて研究した複雑な大型計算機システムの性能評価のための階層構造モデルの手法について述べる。とくに、この考え方によって作成されたマルチプログラミング方式大型仮想メモリTSSシステムの解析的モデルについてその概要を説明し、このモデルのMIT Multicsシステムへの適用例を記述する。このような手法によって、現実のきわめて複雑なシステムがもつ数多い性能パラメータを考慮した実用的でかつ精度のよい性能評価が可能になる。

2. 評価の対象とされる計算機システム

ここで性能評価の対象とする計算機システムはマルチプログラミング方式の仮想メモリTSSシステムであり、仮想メモリはセグメンテーションとページングを用いているものとしている。このようなシステムの一例として Multics システムがあるが、以下で示す解析は他の方式による計算機システムに対しても適用可能なはずが多い。

大まかにシステムの構造を捉えると、図1に示すように、システムの全体は中央処理システムと通信回線で接続された有限数(N)の端末ユーザ集団から成る。ユーザの発生する処理要求のコマンドは中央処理システムによって受け付けられて最初 Memory Queue Q_m でメモリ待ちの状態(Ready)にされる。中央処理システムの主メモリに余裕があるとOSが判断したとき、 Q_m の最初のタスクにメモリが割当てられ、Processor Queue Q_p に入れられて処理待ちの状態(Page Ready)になる。一般に、メモリが割当てられたタスクのことをマルチプログラミングされているといい、こういうタスクの個数を多重度とよぶ。本稿では、マルチプログラミング制御下のこれらのタスクはデマンド・ページング方式で処理されるものとする。したがって、これらのタスクは、図1のように、Page Ready, Running, Page Waitの3状態をくりかえしながら実行される。一般に、多重度は過度のページングを避けるためにOSによって制限されているが、以下の解析でも多重度はたかだか q (定数)個に押えられているとする。タスクの実行が終了すると処理

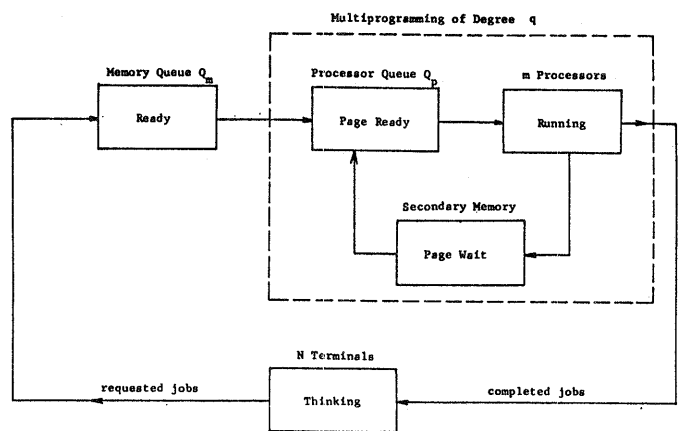
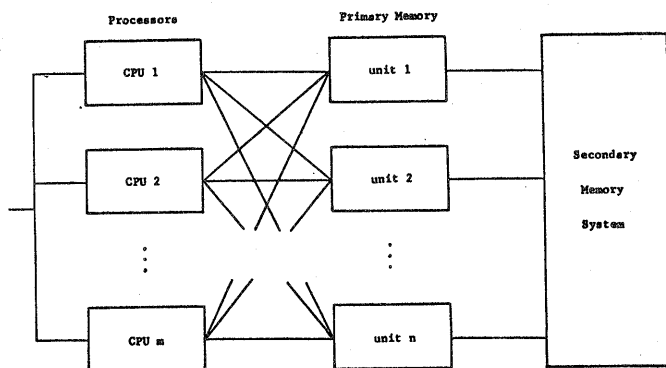


図1 マルチプログラミング方式仮想メモリTSSシステムの構造

* 本稿では、参考文献(1)~(3)の内容を要約的に示した。ここでは、解析的手法の詳細を示すよりはその実用性を例をもって明らかにするように試みた。

結果はユーザへの応答として返答される。以下の解析では、図2に示すように中央処理システムはマルチ・プロセッサ/マルチ・メモリ構成のハードウェアをもっており、また上述の Q_m および Q_p は簡単のためFCFS行列であるものとする。



3. 階層構造モデルの手法

図2 マルチプロセッサ/マルチメモリの中央処理ハードウェア

以下ではこのようなシステムの性能をスループット及び応答時間特性によって解析的に評価するのであるが、数多くの性能パラメータが存在するため、与えられたシステム全体を1個の数学モデルによって近似することは解析解の導出や評価精度の面できわめて問題が多い。

しかしながら、実際のシステムをよくみると、その内部動作はいくつかの「マクロ動作」や「ミクロ動作」等に分類することができ、かつこれらの各々の動作はある主要な内部関連によってしか関係していないものが少なくない。したがって、この観点から、システム全体を大略独立とみられるいくつかのサブシステムに分割し、その各々に対して解析的モデルを作成し、それらを主要内部関連によって結びつけることを試みることによって図3に示すような階層構造モデルをえる。

このような階層構造モデル化手法は、つぎに要約するような好ましい重要な特長を持っている。

- (1) 実質的な評価精度を犠牲にすることなく、複雑なシステムの数多い性能パラメータを考慮した実用的評価が可能（高精度、小計算時間）、
- (2) 各モデルを別々に何人かで分担して作成し、それらの妥当性を分担してチェックすることが可能（小準備時間）、
- (3) モデルの追加・削除が容易（柔軟性）、
- (4) あるサブシステムの実測結果やシミュレーション結果等が入手できたならば、それをモデル群の対応モデルと置換えることが可能（段階的精密化）、
- (5) システムにおける性能パラメータ間の因果関係が理解しやすい（理解容易性）。

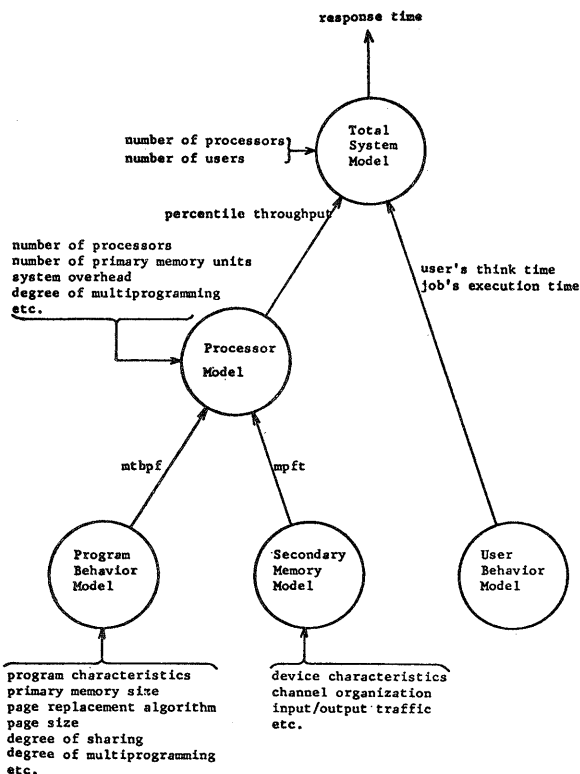


図3 モデルの階層構造構成

4. 各モデルの概要

本節では、紙面の都合で、図3の階層構造モデルの主要モデルをなす(1) Program Behavior Model, (2) Processor Model, (3) Total System Model について概説する。

4.1 プログラム行動モデル (Program Behavior Model)

このモデルによって、マルチプログラミング制御下のタスクの実行のさいのページング頻度が評価される。ここでは、(1) プログラム特性, (2) タスクの使用する主メモリの大きさ, (3) 多重度, (4) タスク間におけるセグメントの共有度等の性能パラメータが考慮される。より複雑なモデルについては文献(1)を参照されたい。以下では、主メモリは m_r ページの OS 常駐域と (最大) ϕ 個のタスクによって使用される m_n ページのユーザ領域 (非常駐 OS も含む) に分割されているとする。タスク間におけるシェアリングのモデル (segment sharing)

(a) OS 常駐域におけるシェアリング: 各タスクは常駐域の $a \times 10^2\%$ を利用する。

(b) ユーザ領域の間のシェアリング: 各タスクの使用する非常駐プログラム領域の大きさは互いに $b \times 10^2\%$ ずつ重複して共用される。

これらの仮定から、各タスクの使用する主メモリの大きさ m は次式で与えられる。

$$m = a m_r + \frac{b m_n}{1 - (1-b)^\phi} \quad \text{ページ} \quad (1)$$

Mean Time Between Page Faults (mtbpf)

CPU 時間が割当てられてからページ・フォールトを発生するまでの平均走行時間のことを mtbpf といい (図1の Running 状態), これは下記の時間の和である。

$\bar{t}_p$: 1個のページ・フォールトを処理するのに必要とされる平均総オーバーヘッド。

$\bar{t}_m$: ページ・フォールトを発生するまでに遭遇する各種フォールト (プロテクション・フォールト, セグメント・フォールト, 割込等) のために必要とされる平均総オーバーヘッド。

$\bar{t}_u$: これら以外のユーザにとって有効な CPU 実行時間の平均値。

線形ページング・モデル

ページ・フォールト以外の各種フォールトの発生頻度はユーザの有効 CPU 実行時間の長さに比例すると考えて、 $\bar{t}_m = \delta \bar{t}_u$ とする。一般に、 $\bar{t}_u$ の長さはメモリ・サイズの複雑な関数になるといわれるが、ここでは Multics システムでの測定結果もふまえて⁽⁴⁾ 簡単のため m に正比例するものとする。すなわち、 $\bar{t}_u = C \cdot m$ とする。さらに、ページング・オーバーヘッド $\bar{t}_p$ は m にほぼ独立と考えると、mtbpf は式(1)より、

$$\text{mtbpf} = (1 + \delta) \cdot C \cdot \left(a m_r + \frac{b m_n}{1 - (1-b)^\phi} \right) + \bar{t}_p \quad (2)$$

Multics 上のプログラム行動の近似例を表1に示す。

4.2 中央処理システム・モデル (Processor Model)

このモデルの目的は、上述の mtbpf 及びその他の性能パラメータの関数として、システム・プロファイルにおけるユーザの有効計算時間 (t_u) の占める割合 (有効計算率 θ とよぶ) を求めることである。考慮すべき性能パラメータには、(1) mtbpf, (2) 2次メモリから主メモリへの1ページのスワップ時間 mpft, (3) 多重度 ϕ , (4) CPU 台数 m , (5) 主メモリ・ユニットの台数 n , (6) メモリ干渉⁽⁵⁾及びテーブル・ロックアウト等のマルチプロセッサ干渉等が存在する。

これらの全てのパラメータを考慮して、 $m=1$ 及び 2 のシステム構成に対して

表1 多重度及びシェアリングの mtbpf の影響
(単位: msec)

多重度 ϕ	ユーザ領域間のシェアリング		
	$b=0$	$b=0.1$	$b=0.2$
1	99.5	99.5	99.5
2	54.5	56.9	59.5
3	39.5	42.7	46.4
4	32.0	35.7	40.0
5	27.5	31.5	36.3
6	24.5	28.7	33.9
7	22.3	26.8	32.3
8	20.7	25.3	31.1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
∞	9.5	18.5	27.5

$m_r = 84$ ページ, $m_n = 300$ ページ,

$(1+\delta) \cdot C = 0.3$, $a = 0.1$, $\bar{t}_p = 7$ msec

マルコフ過程モデルで解析した。^(1,2) ここで紙面の都合で詳細を省くが、2 CPU システムでは、タスクのうち $j(=1, 2)$ 個が Running 状態で k 個が Page Wait 状態にありかつ 2 台の CPU が同一の主メモリ・ユニットを使用中かどうか ($s=1$ ならば競合中) の状態にある定常状態確率を π_j^k とすると有効計算率 θ は次式で与えられる。

$$\theta = \left(1 - \pi_0^0 - \frac{\pi_1^{q-1}}{2} - \frac{q-1}{q} \sum_{k=0}^{q-2} \pi_2^k \right) \cdot \frac{\tau_u}{m \tau_{bpf}} \quad (3)$$

システムのスループット、すなわち 1 秒当りにこなさるタスク数は、各ユーザのタスクが必要とする平均有効 CPU 実行時間を $\bar{T}_e$ とすれば次式となる。

$$\text{スループット} = m\theta / \bar{T}_e \quad (m=2) \quad (4)$$

以上では、2 CPU 構成のシステムについて述べたが、1 CPU 構成のばあいの解析はもっと簡単である。この点については、文献(1)を参照されたい。

4.3 TSS システム・モデル (Total System Model)

このモデルの目的は、上述の有効計算率 θ 及びその他の性能パラメータの関数として、TSS システムの端末ユーザへの応答時間特性を評価することである。考慮されるパラメータには、有効計算率のほか、現在活動中の端末ユーザ数 N 、端末ユーザが次のコマンドを打つさいの平均思考時間 $\bar{T}_t$ 、会話型の各タスクが必要とする平均有効 CPU 実行時間 $\bar{T}_e$ 、CPU 台数 m 、最適多重度 q^* 等がある。

基本的な考え方は、以上で詳細に考察した中央処理システムをブラック・ボックスとして捉え、各々 $\theta \times 10^2\%$ ずつ有効計算に使用される m 台の CPU を各々 100% ずつ有効に使用される θ 倍の相対速度をもつ m 台の CPU とみることである。さらに、ユーザの思考時間 T_t とタスクの必要とする有効 CPU 実行時間 T_e は各々パラメータ μ 及び λ_0 の指数分布に従うとして機械修理員モデルに帰着させる。

ここで、端末ユーザへの応答時間 T_r の確率密度及びその Laplace 変換を $p\{T_r=t\}$ 及び $P(s)$ で表すならば、それらはつぎのように求められる。^(1,3)

$$P(s) = \frac{N! \pi_0}{N - \bar{Q}} \cdot \frac{\lambda}{s + \lambda} \left[\sum_{i=0}^{m-1} \frac{s^i}{i! (N-i-1)!} + \sum_{i=m}^{N-1} \left\{ \frac{s^i}{m! (N-i-1)! m^{i-m}} \left(\frac{m\lambda}{s+m\lambda} \right)^{i+1-m} \right\} \right] \quad (5)$$

$$p\{T_r=t\} = \begin{cases} \lambda e^{-\lambda t} \left\{ \frac{(1+\mu t)^{N-1}}{\sum_{i=0}^{N-1} \frac{(N-1)! s^i}{(N-i-1)!}} \right\} & \leftarrow 1 \text{ CPU 構成 } (m=1) \text{ のばあい } (6) \\ \lambda e^{-\lambda t} \pi_0 \sum_{i=0}^{m-1} \left\{ \frac{(N)(N-i)s^i}{i!} + \frac{m\lambda}{N-\bar{Q}} \left(\frac{N!}{m!} \right) s^m \sum_{i=0}^{N-m-1} \left\{ \frac{s^i}{(N-m-1-i)! (m-1)^{i+1}} (e^{-\lambda t} - e^{-m\lambda t} \sum_{j=0}^i \frac{\{(m-1)\lambda t\}^j}{j!} \right\} \right\} \right\} & \leftarrow m \text{ CPU 構成 } (m \neq 1) \text{ のばあい } (7) \end{cases}$$

ただし、 $s = \mu / \lambda = \mu / \theta \lambda_0 = \bar{T}_e / \theta \bar{T}_t$

$$\pi_0 = \left[\sum_{i=0}^{m-1} \frac{(N)(N-i)s^i}{i!} + \sum_{i=m}^N \frac{N!}{m! (N-i)!} \cdot \frac{s^i}{m^{i-m}} \right]^{-1}, \quad \bar{Q} = \left[\sum_{i=0}^{m-1} i \frac{(N)(N-i)s^i}{i!} + \sum_{i=m}^N \frac{i N!}{m! (N-i)!} \cdot \frac{s^i}{m^{i-m}} \right] \pi_0$$

なお、応答時間の長次のモーメントは $P(s)$ を微分して直ちに求められる。例えば、平均応答時間 ($k=1$) は、

$$E(T_r) = \frac{\sum_{i=0}^{m-1} \frac{i s^i}{i! (N-i-1)!} + \sum_{i=m}^N \frac{i s^i}{m! (N-i-1)! m^{i-m}}}{\sum_{i=0}^{m-1} \frac{s^i}{i! (N-i-1)!} + \sum_{i=m}^N \frac{s^i}{m! (N-i-1)! m^{i-m}}} \cdot \bar{T}_t \quad 1 \leq m \leq N \quad (8)$$

以上のようにして、複雑な構造をもつ TSS システムのスループット及び応答時間特性をシステム・ハードウェアのパラメータ ($m, n, \bar{T}, m_n, m_{pft}$ 等)、システム・ソフトウェアのパラメータ ($m_r, q, \bar{T}_p, \delta$ 等)、ユーザ・プログラムのパラメータ (c, a, b 等)、端末ユーザのパラメータ ($N, \bar{T}_e, \bar{T}_t$ 等) の関数として表現することができた。

5. Multics システムの性能評価例

本節では、前節で導いたスループット及び応答時間の評価式を用いて、MIT Multics システム^{6,7)}(1972年現在)の性能を評価した数値例をいくつか述べる。

5.1 Multics システムの性能の実測

このシステムでは、設計の初期から性能の実測が容易であるようにいくつかの考慮を払っている。例えば、ソフトウェア・モニター機能やそれを動作させるための完備した性能測定コマンド⁶⁾、多数の端末ユーザのシミュレータ⁷⁾等がその例である。1972年末時点のシステムのハードウェアの基本仕様はつぎのとおりである。

CPU Honeywell 645 ($SGM \approx 3 \mu\text{sec}$), 主メモリ 512~1536 KB (サイクルタイム = $1 \mu\text{sec}$),
 ドラム ページング・ドラム (16 セクター), 平均回転待時間 = 17 msec , 転送時間 = 2 msec/ページ ,
 ディスク 平均アクセス時間 = $50 \sim 100 \text{ msec}$ (負荷時)

なお、現在では、Honeywell 6180 CPUを用いたより大型の構成をとっている。

通常は下記的大型システム構成によって大学の public utility として走行するが、必要に応じてCPUや主メモリ・ユニット (PM unit = 512 KB) を動的に切離/接続できる。この結果、運用システムはつぎの3つの構成のうちの1つをとる (図4)。

- (1) 大型システム構成 (2 CPU, 3 PM units)
- (2) 中型システム構成 (1 CPU, 3 PM units)
- (3) 小型システム構成 (1 CPU, 2 PM units)

これら3種のシステム構成時の典型的なシステム性能データの実測結果を表2に要約する。とくに、システム・プロファイルのうちの有効計算率 θ (users' useful computation) と平均応答時間の項目を注目されたい。その他のパラメータの実測結果は、

m_r (テーブル類やバッファ等を含む) =
 280~360 KB,

$\bar{T}_p = \begin{cases} 6 \text{ msec} & (1 \text{ CPU 構成}) \\ 9 \text{ msec} & (2 \text{ CPU 構成}) \end{cases}$

$\delta = 0.2 \sim 0.35$

$\bar{T}_e \approx 400 \text{ msec}$, $\bar{T}_t = 12 \sim 35 \text{ sec}$,

ここで、式(4)を用いれば、

- (1) 大型システム構成のスループット ≈ 2.5 ユーザ/秒
- (2) 中型システム構成のスループット ≈ 1.6 ユーザ/秒
- (3) 小型システム構成のスループット ≈ 1.3 ユーザ/秒

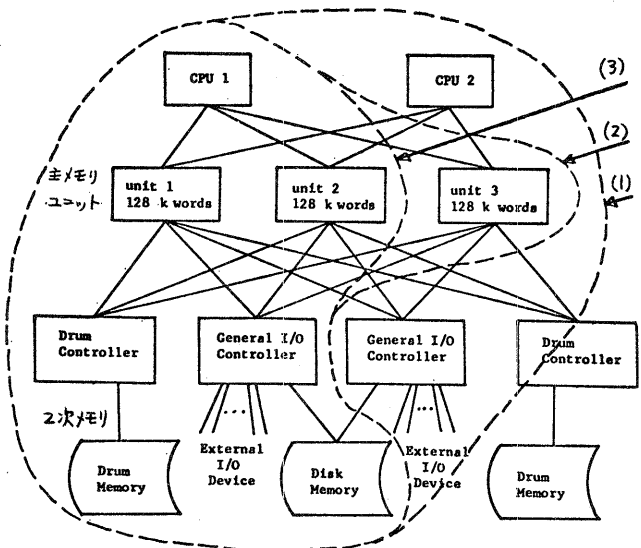


図4 Multicsシステムの3種類のシステム構成

表2 Multicsシステムの典型的システム性能
 (フル・ロード時)

Configuration	July 1971		
	Small 1 CPU 2 PM units	Medium 1 CPU 3 PM units	Large 2 CPUs 3 PM units
average number of eligible processes (多重度)	5.7	5.3	5.6
mtbpf (msec)	16.5	29.7	31.9
mpft (msec)	32.4	33.2	30.9
mean paging overhead time (msec)	5.4	6.1	9.0
processor time breakdown - multiprogramming idle	2.9 %	1.8 %	8.8 %
memory interference idle	0	0	5.7
paging overhead	28.7	18.4	22.9
miscellaneous overhead	16.6	15.9	12.6
users' useful computation θ	51.8	63.9	49.9
number of users N	41	43	48
response time characteristics			
average queue length	16.7	14.9	14.9
average response time to the PDPS user simulator (sec)	not measured	7.9	6.0

5.2 スループット向上の方策の検討

ここでは、上述の大型システム構成のスループットを向上させる方策を検討する。たとえば、つぎの3つの方策が考えられるものとしよう。

- (a) 512 KBの主メモリ・ユニットを1個追加する(式(2)より $mtbpf = 45 \text{ msec}$ となる)。
- (b) 2次メモリのアクセス・タイムを半減させる($mpft = 30.9 \text{ msec} \rightarrow 15.5 \text{ msec}$)。
- (c) システム・オーバヘッドを再コーディングかfirmware化で減少($\bar{t}_p = 9 \rightarrow 7 \text{ msec}$, $\delta = 0.25 \rightarrow 0.20$)。

これらの3つの方策について、中央処理システム・モデルでシステム・プロファイルを評価した結果を表3に示す。ここで、有効計算率 θ をみることにより、上記の3方策は各々20.5%、10.5%、13.1%ずつスループットが向上することがしられる。この結果を考慮して、いちばん実行しやすい方策をとることができる。

表3 3つの性能向上の方策とその効果

Configuration	Current System	Approach (a)	Approach (b)	Approach (c)
No. of Processors m	2	2	2	2
No. of PM units n	3	4	3	3
Degree of Multiprogramming $\frac{m}{n}$	4	4	4	4
mtbpf (msec)	31.9	45.0	31.9	31.9
mpft (msec)	30.9	30.9	15.5	30.9
Mean Paging Overhead Time $\bar{t}_p$ (msec)	9	9	9	7
Misc. Overhead Coeff. δ	0.25	0.25	0.25	0.20
Memory Interference Coeff. γ	1.2	1.2	1.2	1.2
* processor time usage *				
Multiprogramming Idle	13.0 %	7.0 %	3.2 %	13.0 %
Memory Interference Idle	4.9	4.2	5.9	4.9
Paging Overhead	23.2	17.8	25.7	18.0
Miscellaneous Overhead	11.8	14.2	13.1	10.7
Users' Useful Computation θ	47.2	56.9	52.2	53.4

5.3 スループットと応答時間の関係

4.2節からしられるように、システム構成の違いは有効計算率 θ (スループットに正比例)をとおして応答時間に影響する。ここでは、まず上述の3つのシステム構成に対して、式(8)によって平均応答時間を端末ユーザ数の関数として評価した結果を図5に示そう。同時に、この図に実測された平均応答時間を×印で書きしるす。平均応答時間は端末ユーザ数の増加とともに、次式で与えられる直線に漸近する。

$$\bar{T}_r \approx \frac{\bar{T}_e}{m\theta} N - \bar{T}_t \quad (N \gg 1) \quad (9)$$

容易に想像されるように、システム構成のもつ $m\theta$ の値が大きいほど応答時間は短くなる。図5より、平均応答時間を5秒以下にするには、上述の小型、中型、大型システム構成はたかだか約30人、40人、60人の端末ユーザしか同時接続できないことがしられる。

ところが、じっさいには、「×%の応答をY秒以下に押えてほしい」というユーザの要求が強い。このような要求に答えるには、式(6)、(7)によって応答時間の確率密度を評価しなければならない。そのため、ここでは中型

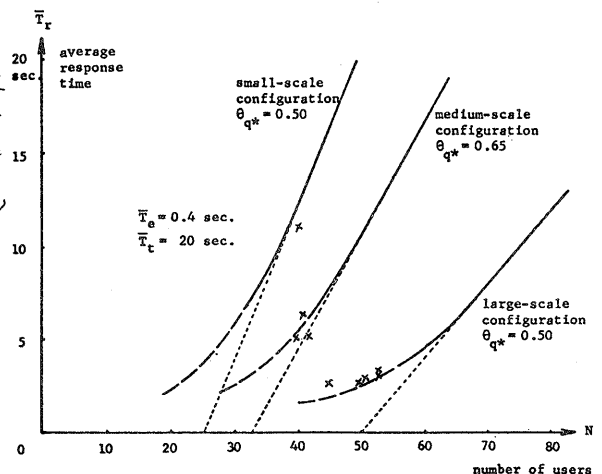


図5 有効計算率と平均応答時間の関係

システム構成に対して応答時間の確率密度を計算した結果を図6に示そう。この図より、端末ユーザ数が増加するにつれて応答時間の分布は指数分布型から徐々に正規分布型に近づくことがしられる。N=40のばあいのMulticsシステムにおける実測例を同図に棒グラフで書き入れる。同様に、図6に対応する分布関数の結果を図7に示す。こうして、応答時間の平均値や中央値ばかりでなく、そのX%限界($0 \leq X \leq 100$)を求めることができる。例えば、システムに40人の端末ユーザがいるばあい、80%の応答は1.4 ~ 10.2秒の間に入ることがしられる。さいごに、この図から応答時間の10%及び90%限界をまとめた結果を図8に示す。このシステム構成のばあい、応答時間の90%限界は平均応答時間のほぼ2倍であることがしられる。

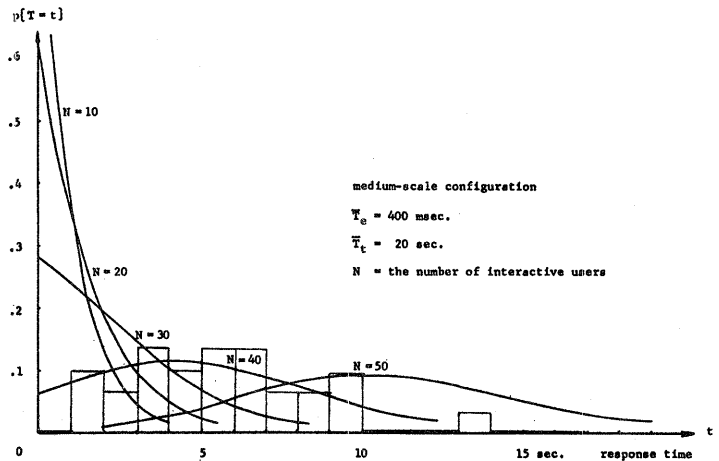


図6 応答時間の確率密度

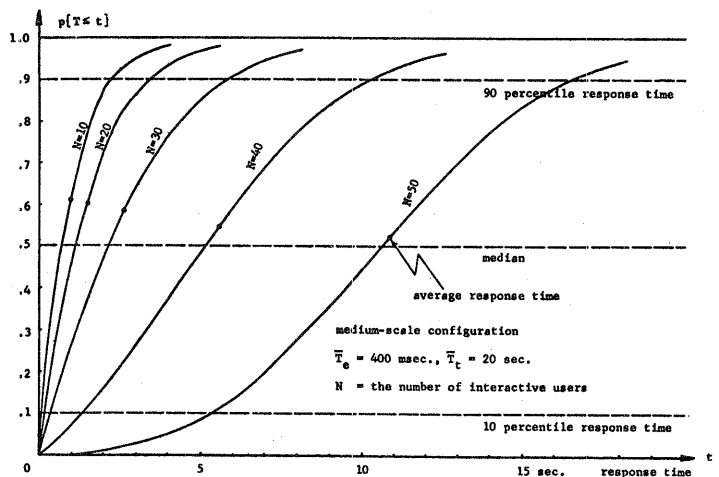


図7 応答時間の分布関数

5.4 ユーザ特性と応答時間の関係

ここでは、端末ユーザの特性の変化が平均応答時間に及ぼす影響を評価する。そのため、中型及び大型システム構成において、

- (1) ユーザ・タスクの必要とする平均有効CPU実行時間 T_e
- (2) ユーザが端末でコマンドを打つさいの平均思考時間 T_t

が変化するばあいの平均応答時間の变化の程度を各々図9、図10に定量評価した。

T_e の増加は端末ユーザの処理要求の量的/質的变化に対応しており、

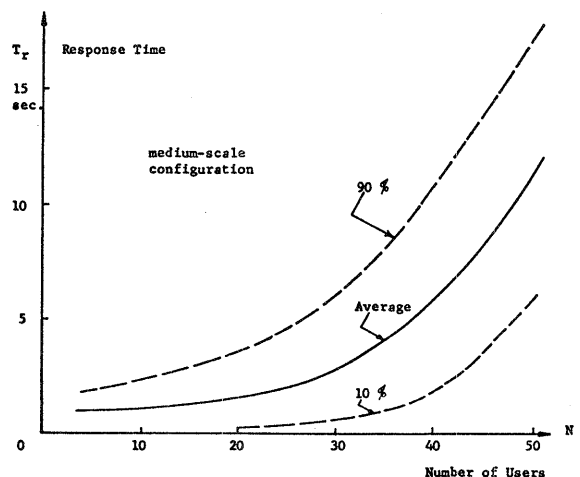


図8 応答時間の平均値と10%,90%限界

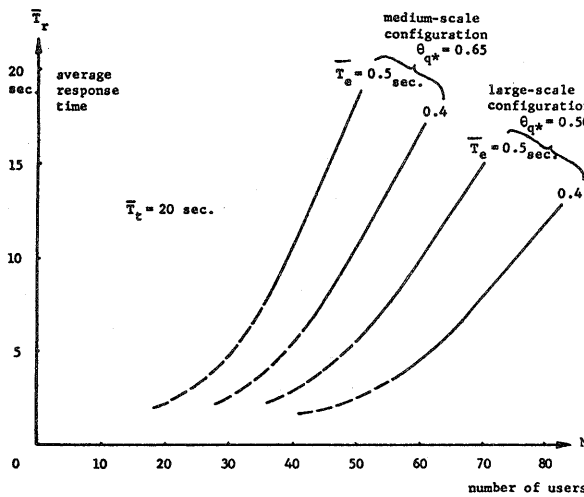


図9 平均有効CPU実行時間の応答時間への影響

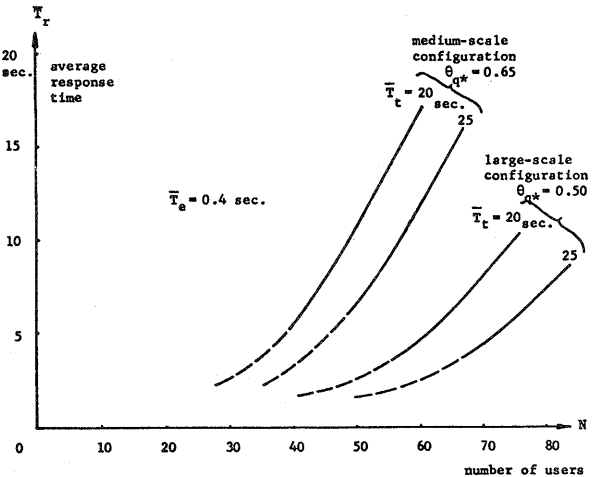


図10 平均思考時間の応答時間への影響

多くのシステムでこのような現象がみられるようである。また, Multics システムにおける実験結果によれば, 端末ユーザの平均思考時間は時間帯や日によってかなり大きく変化し(12~35秒), システムの応答時間と正の相関があるように思われる点注意を要する。

5.5 多重度の最適化

さいごに, OS の可変パラメータである多重度 q を変えたときにシステムの有効計算率(引いては, スループット)がどのように変化するかを最大システム構成のばあいに対して, つぎの3つのケースについて中央処理システム・モデルで評価した。

Case I ユーザ領域間のシェアリング 0 %

Case II ユーザ領域間のシェアリング 10 %

Case III ユーザ領域間のシェアリング 20 %

この結果を表4に示すが, 有効計算率を最大にする最適多重度は5~7であり, これはじっさいの Multics システムの実測多重度である5~6(表2参照)とよく一致している。また, タスク間のシェアリングが増加するにしたがって最適多重度が大きくなる様子及びより有効な主メモリの利用によってスループットが向上する様子が明らかである。

表4 最大システム構成における多重度の最適化
(システム・プロファイルの説明は脚注を参照のこと)

Case I: No Sharing of Non-Resident Programs ($b=0$)

degree (q) of multiprogramming	2	3	4	⑤	6	7	8
(1)	7.6	12.6	17.3	21.5	25.2	28.4	31.1
(2)	10.3	11.7	12.4	12.6	12.6	12.4	12.1
(3)	38.1	24.6	15.9	10.0	6.0	3.5	1.9
(4)	2.4	3.7	4.6	5.2	5.6	5.8	6.0
(5)	41.4	47.1	49.6	50.3	50.4	49.7	48.7

Case II: 10 Percent Sharing of Non-Resident Programs ($b=0.1$)

degree (q) of multiprogramming	2	3	4	5	⑥	7	8
(1)	7.4	12.0	16.0	19.3	22.0	24.0	25.7
(2)	10.5	12.2	13.1	13.5	13.6	13.6	13.4
(3)	37.1	22.6	13.2	7.3	3.7	1.7	0.7
(4)	2.4	3.9	4.8	5.4	5.8	6.0	6.1
(5)	42.3	49.1	52.6	54.2	54.6	54.4	53.8

Case III: 20 Percent Sharing of Non-Resident Programs ($b=0.2$)

degree (q) of multiprogramming	2	3	4	5	6	⑦	8
(1)	7.2	11.3	14.7	17.1	18.9	20.1	21.0
(2)	10.8	12.8	13.8	14.3	14.5	14.5	14.4
(3)	36.0	20.5	10.8	5.1	2.2	0.8	0.2
(4)	2.5	4.1	5.1	5.6	6.0	6.1	6.2
(5)	43.3	51.2	55.4	57.5	58.2	58.3	57.9

mpft = 35 msec, mean paging overhead time ($K_1 \bar{T}_{pl}$) = 7 msec, $\delta = 0.25$, $\gamma = 1.2$,
processor time breakdown: (1) Paging Overhead, (2) Miscellaneous Overhead, (3) Multiprogramming Idle, (4) Memory Interference Idle, (5) Users' Useful Computation (all in percent).

6. おわりに

本稿では、マルチプログラミング方式仮想メモリTSSシステムの階層構造解析的モデルによる性能評価手法について述べた。このようなモデルを用いることによって、マルチ・プロセッサ/マルチ・メモリのハードウェアやOSの構造を考慮した精度のよい実用的な評価が可能となった。ここでは、とくにこのモデルをMIT Multics システムに適用した性能評価の数値例を多く記述した。これらの具体例から、解析的手法がいろいろな現実的性能評価にきわめて有用であることが明らかである。

しかしながら本稿では、紙面の都合によって、これらの解析的モデルの精度に関する詳細な検討(model validation)やシステムのスループットや応答時間に関する重要な評価式の導出の過程等を割愛せざるをえなかった。また、今後に残された問題についてもここでは述べる事ができなかった。これらの点については、下記の参考文献(1)~(3)をご参照頂きたい。

さいごに、この研究をご支援頂いたMIT Project MAC (Laboratory for Computer Science に改称)のMultics システム開発グループ及び日頃ご指導して致したMIT 電気工学科の Prof. F. J. Corbató と Prof. J. H. Saltzer に深く感謝致します。

参考文献

- 本稿に直接関係するものだけにとどめる。解析的性能評価手法に関する一般論やこれらの手法の現実の大型計算機システムへの応用等については文献(10)に詳しい。
- (1) A. Sekino, "Performance Evaluation of Multiprogrammed Time-Shared Computer Systems", Ph.D. Thesis, MIT Project MAC Report MAC-TR-103, Sept. 1972 (Project MAC より入手可能。また, National Technical Information Services から AD-749949として入手可能)
 - (2) A. Sekino, "Throughput Analysis of Multiprogrammed Virtual Memory Computer Systems", Proc. ACM First Annual SIGMETE Symposium on Measurement and Evaluation, Feb. 1973, pp. 47-53.
 - (3) A. Sekino, "Response Time Distribution of Multiprogrammed Time-Shared Computer Systems", Proc. Sixth Princeton Annual Conference on Information Sciences and Systems, March 1972, pp. 613-619.
 - (4) J. H. Saltzer, "A Simple Linear Model of Demand Paging Performance", CACM, 17, 4, April 1974, pp. 181-186.
 - (5) Multics Performance Log (開発段階でのMulticsのシステム性能向上の記録), とくに, MPL-54 (Measurement of Memory Cycle Interference), MPL-57 (Measurement of Two Processor Interferences), MIT Project MAC Computer Systems Research Group 内部資料, 1969-1972.
 - (6) J. H. Saltzer and J. W. Ginitell, "The Instrumentation of Multics", CACM, 13, 8, August 1970, pp. 495-500.
 - (7) H. J. Greenbaum, "A Simulator of Multiple Interactive Users to Drive a Time-Shared Computer Systems", M.S. Thesis, MIT Project MAC Report MAC-TR-58, Jan. 1969.
 - (8) F. J. Corbató, J. H. Saltzer, and C. T. Clingen, "Multics --- The First Seven Years", SJCC, 1972, pp. 571-583.
 - (9) E. I. Organick, "The Multics System: An Examination of Its Structure", MIT Press, 1972.
 - (10) 三上, 箱崎, 関野, "コンピュータ・システムの性能評価技術", 電子通信学会誌(投稿中)。

付 録

計算機システムの性能評価では、性能に係る問題に手軽に答えることができることが実用上きわめて重要である。本研究では、そのために、中央処理システム・モデルによるスループット解析と TSS システム・モデルによる応答時間解析の両方

thruput

* THROUGHPUT ANALYSIS *

number of processors = 2
number of primary memory units = 3
degree of multiprogramming = 5
mean page fetch time (in msec) = 30.0
mean length of in-page operation (in msec) = 3.19
missing-page probability = 0.1
mean paging overhead time (in msec) = 9.0
miscellaneous overhead coefficient = 0.25
memory cycle interference coefficient = 1.20
 paging overhead = .251299866
 miscellaneous overhead = .127883710
 multiprogramming idle = .0524190464
 memory interference idle = .0568625187
 percentile thrupt = .511534840
 processor slow-down factor
 due to memory interference = 1.07200967

*** Additional Information ***

Qp CPU SM MI
pi(3, 2, 0, 0) = .103385237
pi(3, 2, 0, 1) = .0632515093
pi(2, 2, 1, 0) = .188467521
pi(2, 2, 1, 1) = .113935644
pi(1, 2, 2, 0) = .171850422
pi(1, 2, 2, 1) = .102678935
pi(0, 2, 3, 0) = .104922063
pi(0, 2, 3, 1) = .0613090484
pi(0, 1, 4, 0) = .0755611220
pi(0, 0, 5, 0) = .0146384857

r 2245 7.683 171+171

ログラムを PL/I で書き、Multics システムの端末から会話形式で実行できるようにした。すなわち、自分の評価したいシステム構成を諸パラメータの値の組合せで端末から指定するとプログラムがシステム・プロファイルや応答時間特性を出力してくれる形式になっている。

これらの解析プログラムは各々「thruput」及び「resp」と名づけられている。ここでは、これらのプログラムを端末から使用した例を記述している。端末からの評価者の入力とは全て下線をほどこして、プログラムによってタイプされた出力と区別した。PL/I のソース・プログラムや使用法については、文献(1)を参照されたい。

resp

* RESPONSE TIME ANALYSIS *

Do you want to determine the effective percentile throughput? --- type yes or no ---
yes

*** Estimation of Effective Percentile Throughput ***

number of processors = 1
degree of multiprogramming = 4
percentile throughput (th) in % for each degree (q) of multiprogramming

if q=1 then th = 55
if q=2 then th = 63
if q=3 then th = 65
if q=4 then th = 65

number of users = 30

average execution time of a job in seconds = 0.4

average user think time in seconds = 20

iteration	estimate of effective % throughput
1	.649999999
2	.626669958
3	.629028954

effective percentile throughput = 6.29028954e-01

average queue length = 3.52031380e+00 jobs

average response time = 2.65887877e+00 seconds

Do you want to derive the response time distribution?
--- type yes or no ---
yes

*** Derivation of Response Time Distribution ***

number of processors = 1

effective percentile throughput in % = 62.90

number of users = 30

average execution time of a job in seconds = 0.4

average user think time in seconds = 20

time	density	cumulative
t = 0.0	p = .281582188	cum = 0.000000000
t = .5	p = .262502369	cum = .136165990
t = 1.0	p = .240528617	cum = .262020107
t = 1.5	p = .216799615	cum = .376402460
t = 2.0	p = .192369802	cum = .478703693
t = 2.5	p = .168154361	cum = .568808913
t = 3.0	p = .144896636	cum = .647018701
t = 3.5	p = .123155845	cum = .713959388
t = 4.0	p = .103311582	cum = .770491533
t = 4.5	p = .085580618	cum = .817623936
t = 5.0	p = .070041458	cum = .856438048
t = 5.5	p = .056662548	cum = .883025880
t = 6.0	p = .045330858	cum = .913442105
t = 6.5	p = .035878406	cum = .933670118
t = 7.0	p = .028105280	cum = .949600466
t = 7.5	p = .021798311	cum = .962019749
t = 8.0	p = .016745327	cum = .971607707

average response time = 2.65923250 seconds

r 2119 9.451 68+60