

フィードバック系を持つマルチプロセッサの解析

吉川 和利*・平田 威彦**

Analysis of Multi-Processor with Feed-Back Loops

Kazutoshi YOSHIKAWA and Takehiko HIRATA

Abstract

The multi-processor system of the computer systems attracts our attention to the function, the reliability and the expansibility of the system structure. The multi-processor system is so flexible that is able to change the system structure with a given situation correlatively. As some processors process jobs which come into the system at the same time, various conflicts occur among the resources of the system. A harmful problem is risen from the fact which the capacity of each resource is limited to the reason mentioned above.

At the first step of the analytic study for the multi-processor system, we consider the multi-processor feed-back model. Next, we investigate by this model the relation between CPU and I/O when the processing of the tasks are repeated several time. We describe the method for the analysis and the meaning of feed-back ratio α , and give the the results of this analysis.

1. 緒 言

計算機システムにおいては大型化に伴ない、従来になく高信頼性が要求され、また高密度の処理を達成するため規模および機能の拡張性が要求されている。この二つの要求を満たす計算機システムとしてマルチプロセッサ・システムが注目されている。マルチプロセッサ・システムでは、複数台のプロセッサによって並列処理がおこなわれるため、各種リソース間の競合が起り、システムの持っているリソースの能力が充分発揮されないという問題が生ずる。

いままで、計算機システムの解析は、様々な条件に対しておこなわれている。たとえばマルチプログラミング・システムの解析¹⁾、並列プログラミングのスケジューリング²⁾、ページング・マシンのドラム使用効率とスケジュールとの関係³⁾などである。しかしここでは、計算機システムを最も巨視的にモデル化した、I/Oフィード・バック系を持つマルチプロセッサについて、待ち行列理論を用い、リソースの使用効率、待ち行列の大きさと待ち時間などの評価をおこなう。すなわちシステムが能率良く働くための各リソースの処理能力、台数について最適な関係を求めるとともに、

どのようなスケジュールを行なえば良いかについて糸口をつかもうとするものである。

2. フィードバック・モデル

CPU と入出力装置の多重処理について、入出力装置に重点を置いた解析を行なう。

前提条件として次のものを考える。

- (1) i 番目のタスクが到着して、 $i+1$ 番目のタスクが到着するまでの時間間隔の分布（すなわち到着分布）は平均値 $1/\lambda$ の指数分布に従い、互に独立とする。
- (2) CPU において、 i 番目のタスクの処理時間の分布（すなわちサービス分布）は、平均値 $1/\mu$ の指数分布に従い、互に独立とする。
- (3) I/O において、 i 番目のタスクのサービス分布は、平均値 $1/\nu$ の指数分布に従い、互に独立とする。

これらの前提条件のもとに、Fig. 2・1 のようなモデルを考える。このモデルはタスクが CPU と I/O の処理を繰返し行なう場合について設定したもので、CPU と I/O の関係、特に CPU と I/O の並列処理の多重度を調べるものである。CPU および I/O はそれ

* 大学院工学研究科電気工学専攻

** 電気工学科

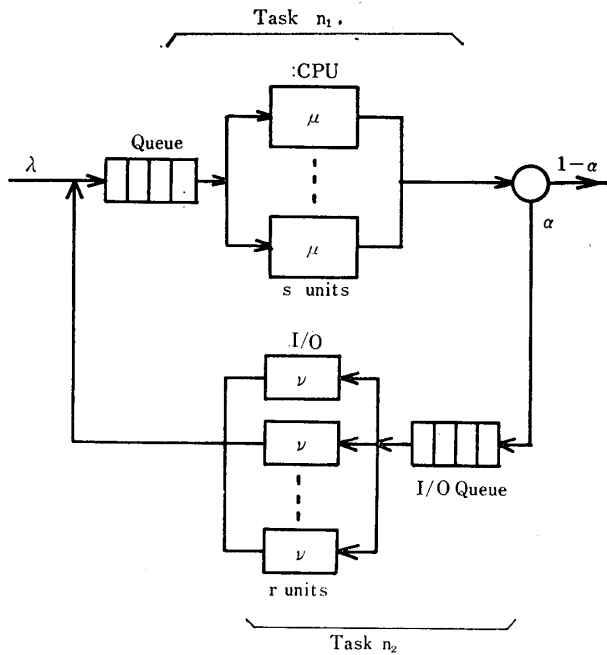


Fig 2.1. Diagram of Multi-Processor System

それぞれ s 台, r 台とする。CPU の処理を終了したタスクは α の確率で I/O の処理を受け, 残りのタスクはその全処理を終了するものとする。また CPU の待ち行列中および処理中のタスクの数を n_1 で, I/O の待ち行列中および処理中のタスクの数を n_2 で表わす。

このモデルを待ち行列の理論⁴⁾を用いて解析し, α のもつ意味を考える。

3. 解 析

Fig. 2.1 のモデルにおいて, 時刻 t に CPU の待ち行列中, および CPU 処理中のタスクが n_1 で, かつ I/O の待ち行列中, および処理中のタスクの数が n_2 である確率を $P_{(n_1, n_2)}(t)$ とする。微小時間 Δt を考え, 時刻 $t + \Delta t$ における確率 $P_{(n_1, n_2)}(t + \Delta t)$ は次のようになる。

$$\begin{aligned}
 P_{(0,0)}(t + \Delta t) &= P_{(1,0)}(t)(1 - \alpha)\mu\Delta t + P_{(0,0)}(t)(1 - \lambda\Delta t) \\
 P_{(n_1,0)}(t + \Delta t) &= P_{(n_1-1,0)}(t)\lambda\Delta t + P_{(n_1-1,1)}(t)\nu\Delta t \\
 &\quad + P_{(n_1+1,0)}(t)(1 - \alpha)\min(n_1 + 1, s)\mu\Delta t \\
 &\quad + P_{(n_1,0)}(t)\{1 - \lambda\Delta t - \min(n_1, s)\mu\Delta t\} \\
 P_{(0, n_2)}(t + \Delta t) &= P_{(1, n_2-1)}(t)\alpha\mu\Delta t + P_{(1, n_2)}(t)(1 - \alpha)\mu\Delta t \\
 &\quad + P_{(0, n_2)}(t)\{1 - \lambda\Delta t - \min(n_2, r)\nu\Delta t\} \\
 P_{(n_1, n_2)}(t + \Delta t) &= P_{(n_1-1, n_2)}(t)\lambda\Delta t \\
 &\quad + P_{(n_1-1, n_2+1)}(t)\min(n_2 + 1, r)\nu\Delta t \\
 &\quad + P_{(n_1+1, n_2)}(t)(1 - \alpha)\min(n_1 + 1, s)\mu\Delta t \\
 &\quad + P_{(n_1+1, n_2-1)}(t)\alpha\min(n_1 + 1, s)\mu\Delta t \\
 &\quad + P_{(n_1, n_2)}(t)\{1 - \lambda\Delta t - \min(n_1, s)\mu\Delta t \\
 &\quad - \min(n_2, r)\nu\Delta t\} \quad (3 \cdot 1)
 \end{aligned}$$

式 (3.1) を変形して $\Delta t \rightarrow 0$ とすると次の微分方程式が得られる。

$$\begin{aligned}
 P'_{(0,0)}(t) &= (1 - \alpha)\mu P_{(1,0)}(t) - \lambda P_{(0,0)}(t) \\
 P'_{(n_1,0)}(t) &= \lambda P_{(n_1-1,0)}(t) + \nu P_{(n_1-1,1)}(t) \\
 &\quad + (1 - \alpha)\min(n_1 + 1, s)\mu P_{(n_1+1,0)}(t) \\
 &\quad - \{\lambda + \min(n_1, s)\mu\} P_{(n_1,0)}(t) \\
 P'_{(0, n_2)}(t) &= \alpha\mu P_{(1, n_2-1)}(t) + (1 - \alpha)\mu P_{(1, n_2)}(t) \\
 &\quad - \{\lambda + \min(n_2, r)\nu\} P_{(0, n_2)}(t) \\
 P'_{(n_1, n_2)}(t) &= \lambda P_{(n_1-1, n_2)}(t) \\
 &\quad + \min(n_2 + 1, r)\nu P_{(n_1-1, n_2+1)}(t) \\
 &\quad + (1 - \alpha)\min(n_1 + 1, s)\mu P_{(n_1+1, n_2)}(t) \\
 &\quad + \alpha\min(n_1 + 1, s)\mu P_{(n_1+1, n_2-1)}(t) \\
 &\quad - \{\lambda + \min(n_1, s)\mu \\
 &\quad + \min(n_2, r)\nu\} P_{(n_1, n_2)}(t) \quad (3 \cdot 2)
 \end{aligned}$$

ここで $t \rightarrow \infty$ のとき, 平衡状態に達するものと仮定して

$$\lim_{t \rightarrow \infty} P_{(n_1, n_2)}(t) = P_{(n_1, n_2)}$$

で表わし, 右辺を 0 とおいて解く。また互いに独立な事象であるので乗法法則⁵⁾により

$$P_{(n_1, n_2)} = P_1(n_1) \cdot P_2(n_2)$$

従って

$$P_1(n_1) = \begin{cases} \frac{a^{n_1}}{n_1!} P_1(0) & n_1 \leq s \\ \frac{s^s \rho^{n_1}}{s!} P_1(0) & n_1 \geq s \end{cases} \quad (3 \cdot 3)$$

$$P_2(n_2) = \begin{cases} \frac{b^{n_2}}{n_2!} P_2(0) & n_2 \leq r \\ \frac{r^r \sigma^{n_2}}{r!} P_2(0) & n_2 \geq r \end{cases}$$

が得られる。ここで

$$\begin{aligned}
 a &= \frac{\lambda}{(1 - \alpha)\mu} & b &= \frac{\alpha\lambda}{(1 - \alpha)\nu} \\
 \rho &= \frac{\lambda}{(1 - \alpha)s\mu} & \sigma &= \frac{\alpha\lambda}{(1 - \alpha)r\nu}
 \end{aligned}$$

である。なお正則条件

$$\sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} P_{(n_1, n_2)} = \sum_{n_1=0}^{\infty} P_1(n_1) \cdot \sum_{n_2=0}^{\infty} P_2(n_2) = 1$$

より

$$\sum_{n_1=0}^{\infty} P_1(n_1) = 1 \quad \sum_{n_2=0}^{\infty} P_2(n_2) = 1$$

従って

$$P_1(0) = 1 / \left\{ \sum_{n_1=0}^{s-1} \frac{a^{n_1}}{n_1!} + \frac{a^s}{(s-1)!(s-a)} \right\} \quad (3 \cdot 4)$$

$$P_2(0) = 1 / \left\{ \sum_{n_2=0}^{r-1} \frac{b^{n_2}}{n_2!} + \frac{b^r}{(r-1)!(r-b)} \right\}$$

となる。ここで

$$\rho < 1 \quad \sigma < 1 \quad (3 \cdot 5)$$

式 (3・3), (3・4) を見ると $P_1(n_1)$, $P_2(n_2)$ はおのおの独立な系の定数で求めることができ, CPU 処理の待ち行列系と I/O 待ち行列系は独立な系であると考えられる. このことより次のことが言える.

一般に s 個の窓口のサービス系において到着分布が平均値 $1/\lambda$ の指数分布で, かつサービス分布が指数分布 (すなわち Kendall の記号で M/M/s 系) の場合, この系から処理を終了して立ち去る客の時間間隔は平均値 $1/\lambda_1$ の指数分布になることが示されている⁴⁾. また, 平均値 $1/\lambda_1$, 平均値 $1/\lambda_2$ の指数分布を持つ互に独立な二つの到着を一つに統合すると平均 $1/(\lambda_1 + \lambda_2)$ の指数分布を持つことが知られている⁶⁾. したがって, Fig. 2・1 のモデルにおいては CPU 処理, I/O 処理系への到着分布および終了分布は, 全て指数分布になる. ここで, この考えに従ってあらたに Fig. 3・1 のように CPU 処理系への到着率を x とすると

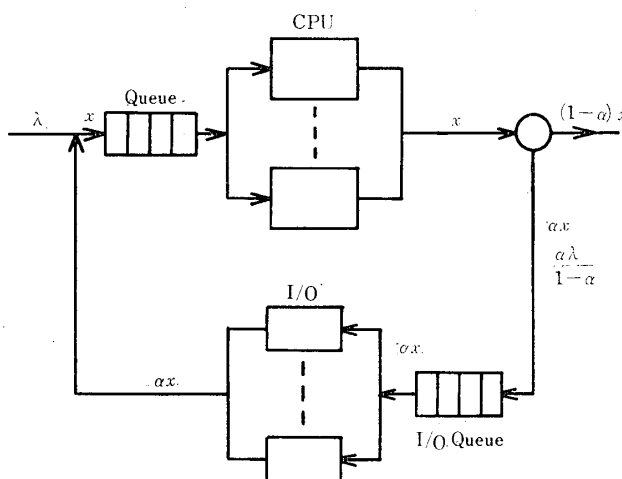


Fig 3.1. Model of Multi-Processor Feed-Back System

$$x = \lambda + \alpha x$$

したがって

$$x = \lambda / (1 - \alpha)$$

となる. すなわち CPU 処理の待ち行列への到着率が $\lambda / (1 - \alpha)$ の M/M/s 系となり, I/O 処理の待ち行列では到着率が $\alpha \lambda / (1 - \alpha)$ の M/M/r 系となる. これらの系につきおのおの独立に解けば式 (3・3), (3・4) と同じ解が得られる. ただし, これらの解が存在するには式 (3・5) の条件が必要である. この条件が満足されないときには, 待ち行列が無限大になる.

次に式 (3・3), (3・4) を用いて系の諸量を求

める.

- (1) CPU 処理系の待ち行列の平均長さ L_{cq}

$$L_{cq} = \sum_{n_1=s+1}^{\infty} (n_1 - s) P_1(n_1) = \frac{a^{s+1}}{(s-1)!(s-a)^2} P_1(0)$$

- (2) I/O 処理系の待ち行列の平均長さ L_{iq}

$$L_{iq} = \sum_{n_2=r+1}^{\infty} (n_2 - r) P_2(n_2) = \frac{b^{r+1}}{(r-1)!(r-b)^2} P_2(0)$$

- (3) CPU 処理系の平均長さ L_c

$$L_c = \sum_{n_1=1}^{\infty} n_1 P_1(n_1) = L_{cq} + a$$

- (4) I/O 処理系の平均長さ L_i

$$L_i = \sum_{n_2=1}^{\infty} n_2 P_2(n_2) = L_{iq} + b$$

- (5) CPU 処理系の平均待ち時間 W_{cq}

$$W_{cq} = \sum_{k=0}^{\infty} P_1(k+s) E(T_{ck}) = \frac{1-\alpha}{\lambda} L_{cq}$$

$E(T_{ck})$; 待ち行列の $k+1$ 番目に並んでいるタスクが処理を受けるまでの平均待ち時間

$$E(T_{ck}) = \sum_{i=1}^{k+1} E(X_i) = (k+1)/s\mu$$

- (6) I/O 処理系の平均待ち時間 W_{iq}

$$W_{iq} = \sum_{k=0}^{\infty} P_2(k+r) E(T_{ik}) = \frac{1-\alpha}{\alpha\lambda} L_{iq}$$

$$E(T_{ik}) = (k+1)/r\nu$$

- (7) CPU 処理系の待ち行列中で費される全時間の平均値 W_c

$$W_c = E(T_c + V_c) = E(T_c) + E(V_c) = W_{cq} + 1/\mu = \frac{1-\alpha}{\lambda} L_c$$

T_c ; 待ち時間

V_c ; サービス時間

- (8) I/O 処理系の待ち行列中で費される全時間の平均値 W_i

$$W_i = E(T_i + V_i) = E(T_i) + E(V_i) = W_{iq} + 1/\nu = \frac{1-\alpha}{\alpha\lambda} L_i$$

以上の結果より、一つのタスクがこのシステムで処理を完了するまでに要する平均時間 T は次のようになる。

$$T = W_c + \sum_{k=1}^{\infty} \alpha^k (W_i + W_c) = \frac{\alpha W_i + W_c}{1 - \alpha}$$

また、一つのタスクがこのシステムを脱するに要する最少の平均処理時間 T_p は

$$T_p = 1/\mu + \sum_{k=1}^{\infty} \alpha^k (1/\mu + 1/\nu) = (1/\mu + \alpha/\nu)/(1 - \alpha)$$

この T_p は CPU 処理系にも I/O 処理系にも待ち行列が全然存在しない全処理時間を表わしており、この T_p と T を比較することにより、このシステムの処理能力に対する待ちの影響を知ることができる。

4. α の意味付け

Fig. 2・1 のモデルでは「CPU」の処理を終了したタスクは α の確率で I/O の処理に移るか、 $1 - \alpha$ の確率で処理が完了すると仮定したが、次にシステムが平衡状態のときの α の意味付けを考える。

このモデルを到着するタスクを I/O の処理を要求する回数によって分類し、 i 回の I/O 処理を要求するタスクの到着率（単位時間に到着する平均タスク数）を $\lambda_i (i = 0, 1, 2, \dots, k)$ とすると $\lambda = \sum_{i=0}^k \lambda_i$ となる。

次に CPU の処理に移ってくるタスクについて I/O の処理があと i 回残されているタスクの状態を S_i で表わすと、単位時間に状態を変えるタスクの平均個数は Fig. 4・1 のようになる。ここで λ'_i は単位時間に状

態 S_i から I/O 処理を受けて状態 S_{i-1} に移るタスクの平均個数である。平衡状態においては、単位時間に各状態に移ってくるタスクの平均個数と、その状態から他の状態に移って行くタスクの平均個数は等しいので式 (4・1) が成立する。

$$\lambda'_k = \lambda_k \quad (4 \cdot 1)$$

$$\lambda'_i = \lambda_i + \lambda'_{i+1} \quad i = 0, 1, \dots, k-1$$

この式より

$$\lambda'_i = \sum_{j=i}^k \lambda_j \quad (4 \cdot 2)$$

となる。

次に単位時間に CPU 処理に移ってくる全タスクの平均個数は

$$\lambda'_k + \lambda'_{k-1} + \lambda'_{k-2} + \dots + \lambda'_1 + \lambda'_0 = \sum_{j=0}^k \lambda_j \quad (4 \cdot 3)$$

となる。このうちで CPU の処理が終ると全処理が終了するタスクの平均個数は

$$\lambda'_1 + \lambda'_0 = \lambda'_0'$$

である。従って α は式 (4・1), (4・2), (4・3), (4・4) より

$$1 - \alpha = \frac{\lambda'_0'}{\sum_{j=0}^k \lambda'_j} = \frac{\lambda}{\sum_{i=0}^k (i+1)\lambda_i} \quad (4 \cdot 5)$$

よって

$$\alpha = 1 - \frac{\lambda}{\sum_{i=0}^k (i+1)\lambda_i}$$

となる。よって単位時間に i 回の I/O 処理を要求するタスクの到着率 $\lambda_i (i = 0, 1, 2, \dots, k)$ が把握できれば α は決定される。

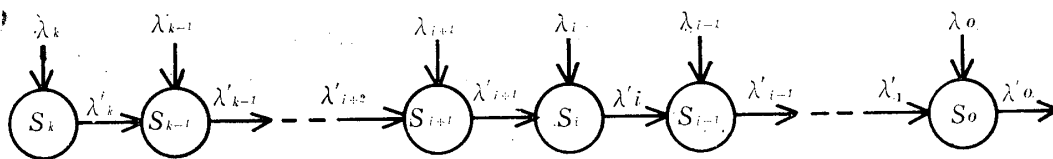


Fig 4.1. Transition Diagram for Multi-Processor with Feed-Back Loops

5. 解析結果

いくつかのパラメータの組合せについて、待ち行列の平均長さ、タスクが処理を終了するまでモデルの中で費す平均時間を計算した結果を Fig. 5・1 ~ Fig. 5・8 に示す。

各パラメータは実際のシステムのデータを用いるべきであるが、計算機により仕様が異なるので CPU の処理率を規準とし（すなわち $\mu_c = 1.0$ ）、他のパラメ

ータは CPU の処理率に相対的に与へ、変化させ、システムへの影響を調べた。

Fig. 5・1, Fig. 5・2 のグラフは CPU の台数 s を変化させた時の各処理系の平均長さ、およびシステムの平均総処理時間を示した。また Fig. 5・1 では CPU の使用率 $L_c - L_{c0}$ をも示した。CPU 処理系と I/O 処理系は互に独立であるので CPU 台数 s に対して I/O 処理系の平均長さは一定となる。Fig. 5・2 の場合には $s = 1, 2$ では CPU の処理がボトルネックになっ

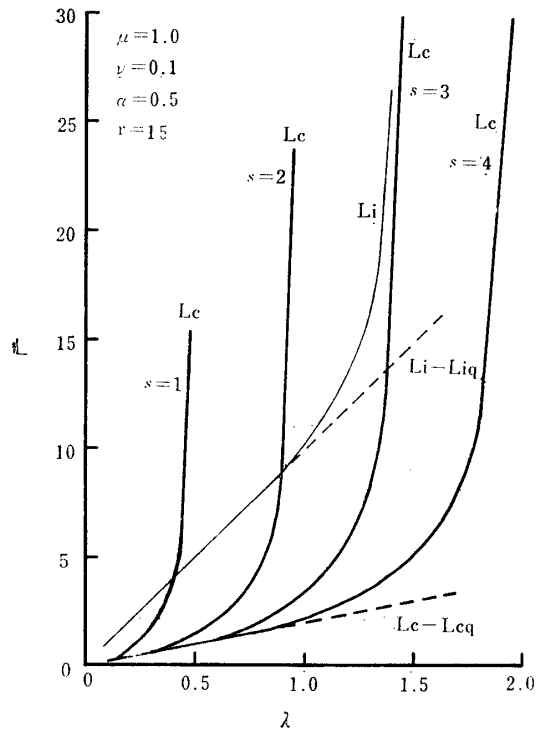


Fig 5.1. Graph of the mean length of each Processor system against the number of CPU's

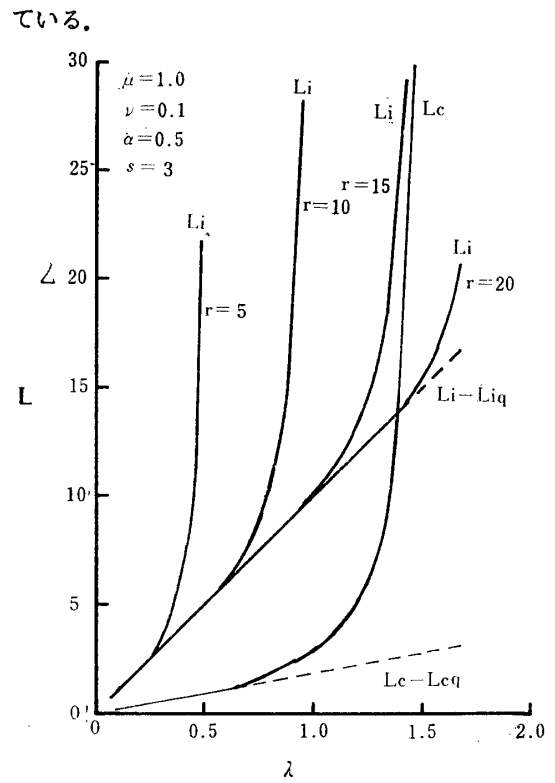


Fig 5.3. Graph of the mean length of each processor system against the number of I/O

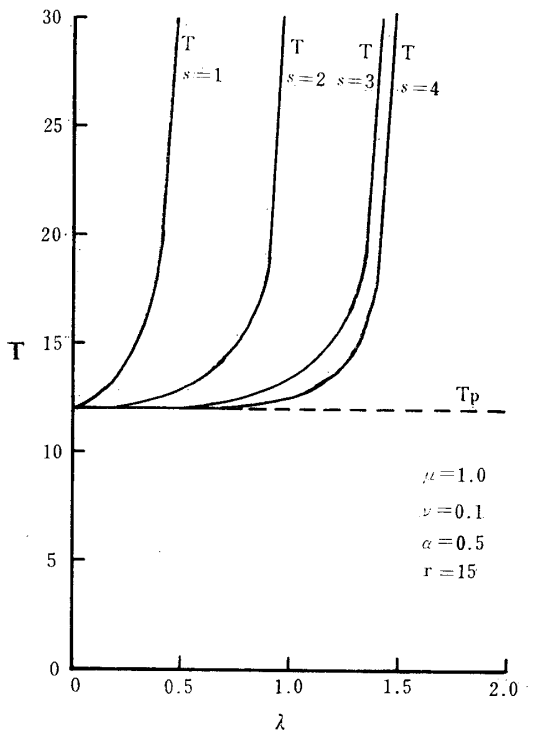


Fig 5.2. Graph of the mean time of all processing against the number of CPU's

ており、 $s = 3$ ではCPUとI/Oの処理が大体平衡しており、 $s = 4$ ではI/Oの処理がボトルネックになっ

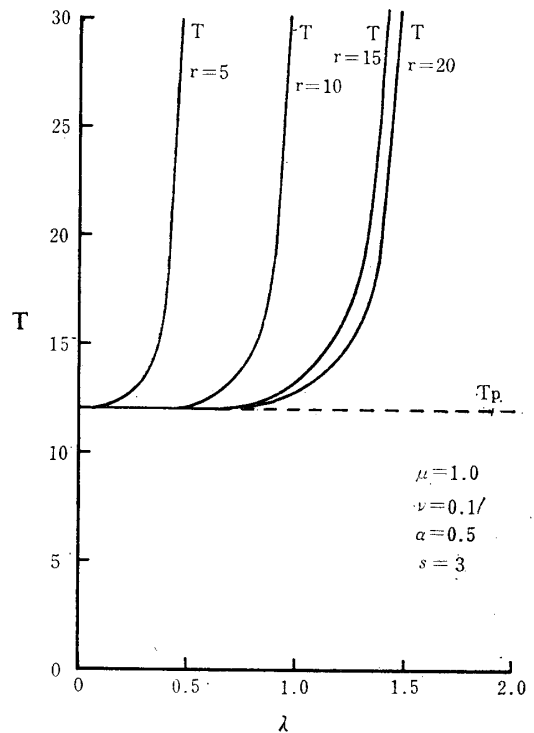


Fig 5.4. Graph of the mean time of all processor system against the number of I/O's

Fig. 5・3, Fig. 5・4 のグラフは I/O の合数 r を変化させた時の各処理系の平均長さ, およびシステム の平均総処理時間を示す. また Fig. 5・3 では I/O の 使用率 $Li-Liq$ も同時に示した. 前述のごとく I/O の 合数 r の変化に対して CPU 処理系は独立であるの で CPU 処理系の平均長さは一定となる. この二つの グラフより $r=15$ の時 CPU と I/O の処理が大体平衡 しており $r=5, 10$ では I/O が, $r=20$ では CPU が 処理のボトルネックになっている.

Fig. 5・5, Fig. 5・6 は CUP 処理の後 I/O 処理に 移る割合 α を変えた時の各処理系の平均長さ, および システムの平均総処理時間を示した. α を大きくする ことは, 平均到着率が一定でもこのモデルに対する負 荷が増えたことになり, その影響が各処理系の平均長 さ, およびシステムの平均総処理時間に顕著に現われ ている. 特に Fig. 5・6 のグラフは一つのタスクを処 理するのに必要な平均時間が α の増加と共に急激に増 加していることを示す. 一方 Fig. 5・5 からわかるよ うに $\alpha=0.7, 0.8$ では I/O の処理がボトルネックに なっており, $\alpha=0.5$ では CPU が処理のボトルネ ックになっている. すなわち α の値の変化によってボ トルネックの場所が変わる.

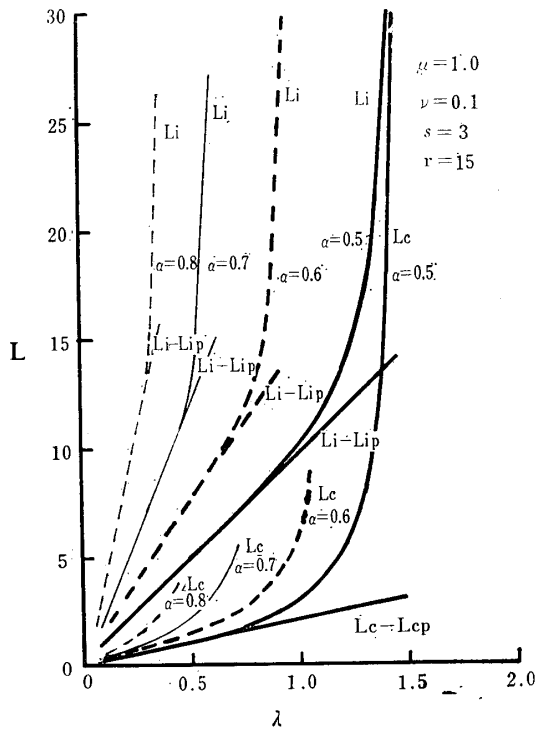


Fig 5.5. Graph of the mean length of each processor system against α .

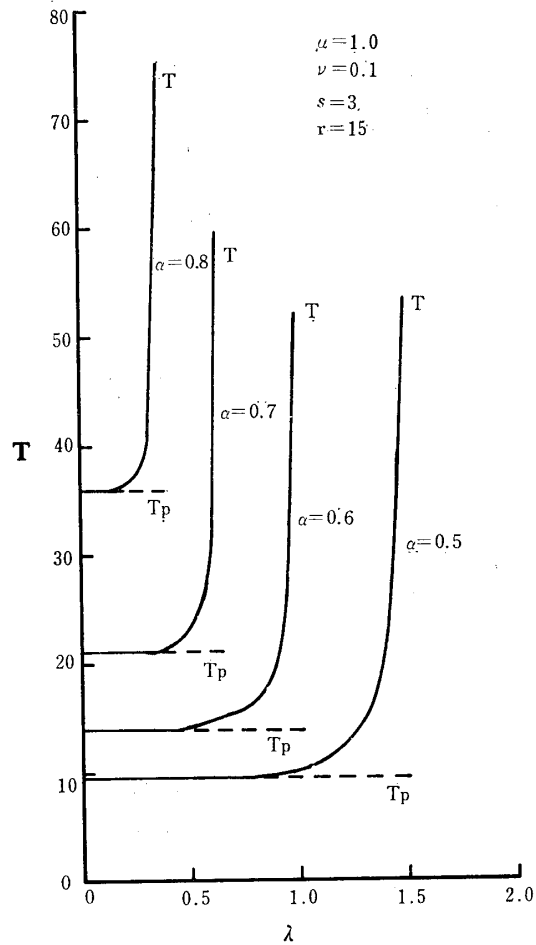


Fig 5.6. Graph of the mean time of all proce ssing against α

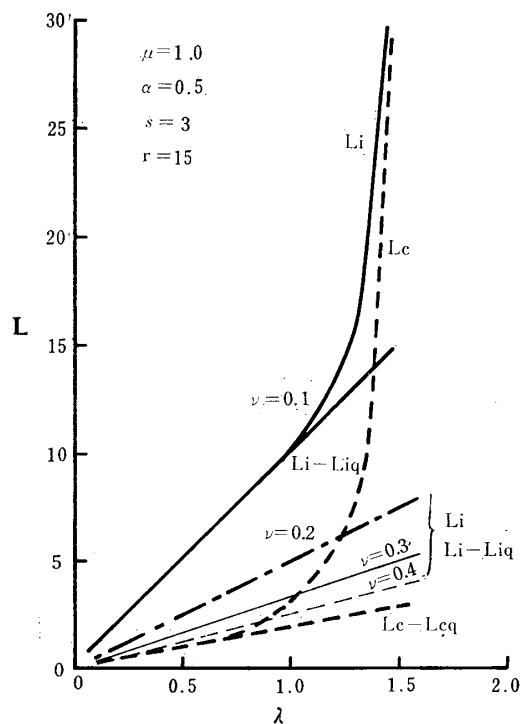


Fig 5.7. Graph of the mean length of each processor system against ν

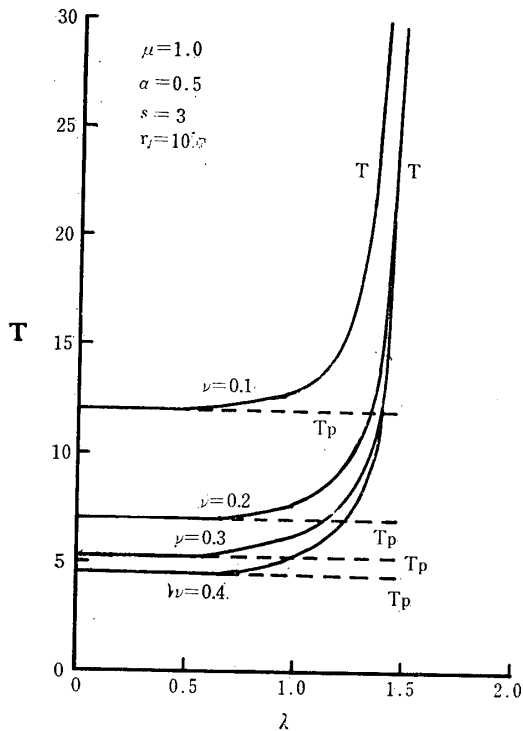


Fig 5.8. Graph of the mean time of all processing against ν

Fig. 5・7, Fig. 5・8 は I/O の処理率 ν を変化させたときの各処理系の平均長さ、およびシステムの平均総処理時間を示したものである。この結果は Fig. 5・3, Fig. 5・4 の I/O の合数を変化させた場合と比較してみるとよい。すなわち I/O の合数を増加させるよりも I/O の処理率 ν を増加させるのが I/O 処理系の平均長さ、および待ち行列の平均長さを短くすることにより効果的であり、またシステム平均総処理時間も短くなる。

以上の結果から、このモデルの特性が各パラメータの変化に対してどのような影響を受けるかを巨視的に

把握できる。

6. 結 言

今回マルチプロセッサ・システムを考察する第一歩において、これを非常に理想化し、かつ巨視的なモデルを提案しその特性を考えた。高度で複雑なシステムも最終的には CPU と I/O のやり取り、それにメッセージの外部からの入力と処理完了というこのモデルに置き換えられる。しかし到着分布や CPU, I/O の処理時間分布を指数分布としたが、実際にはどの位適合するのか、また I/O の使用状況が適切に解析モデルに反映されるかなどの問題が残る。

次に計算機システムではコアブロックを有し、それを各リソースが競合することにより閉塞状態が起る可能性があり、このモデルをさらに発展させて考察する必要がある。

最後に御助言をいただいた鳥岡豊士 (短期大学部)、協力いただいた川下満 (大学院生)、川本正章 (卒論生)、木野真澄、井上朱美 (電子計算機室) の諸氏に謝意を表す。

参 考 文 献

- 1) D.P. Gaver : JACM, 14, 423 (1967)
- 2) L. Kleintock : JACM, 13, 179 (1966)
- 3) E.G. Coffman : JACM, 16, 73 (1969)
- 4) 本間鶴千代 : 待ち行列の理論, 理工学社 (1966)
- 5) W. Feller : An Introduction to Probability Theory and Its Application, John Wiley & Sons Inc. N.Y (1:59) P. 116
- 6) R.W. Conway, W.L. Maxwell, L.W. Milbr : Theory of Scheduling. Addison Wesley Publishing Company (1967) P. 125

(昭和48年9月17日受理)