

論文

オープンバッチ・多重ジョブ処理システムの待行列モデルの解析*

石 黒 美 佐 子**

Abstract

An open batch multi-job-processing system is analyzed by a $M/M/n$ queuing model. In the model, jobs are considered to be processed with n independent machines whose mean processing rate is $\mu(n)$, where n is the actual multiplicity which varies from 0 to a designated multiplicity N . $\mu(n)$ are separately calculated from another sub models of the executing stage in which multiplicity is held to n .

The sub models can be connected with the original one from the fact that the geometric distribution assumed to job processing time is approximately equal to the exponential distribution.

Numerical comparisons of processing time, turnaround time, throughput, etc. are shown in the figures to the variation of job multiplicities.

1. ま え が き

多重プログラミングシステムに対しては種々の待行列モデルの解析がなされている^{1)~3)}。いずれも実行時モデルの解析に重点が置かれている。

ここでは、大型センターで不特定多数のユーザを相手にオープンバッチ処理を行う場合のシステム解析、特にターンアラウンドタイム、ジョブの実行経過時間、ジョブの実行待行列、装置の並行処理率とジョブ多重度との関係等、実行時モデルを包括するバッチ処理の全容をモデル化し解析する。

システム全体を一挙にモデル化するとモデルが複雑になり過ぎて整った解が得られないので、まず、バッチ処理を、入力→実行→出力に区分し、実行時モデルを切り離して別に計算し、全体モデルにはめ込むことにする。実行時モデルからジョブの平均実行経過時間 $e(n)$ を実行多重度 n の関数として計算しそれを全体モデルの解析に使用する。ここでの考え方は、実行時には指定多重度 N またはそれ以下の数 n 台分の計算機が並列に動作していて、各々が平均実行時間 $e(n)$ のジョブを実行していると見立ててモデル化すること

である。指定多重度が N でも、実際はある確率 $P_r(n)$ で N 以下の多重度で実行されている場合があり、実行時モデルは実行多重度 n にしたがって変動すると考える。

このように、このモデルでは、指定多重度以下で実行されている場合の確率的取り扱いを含んでいるのでより現実を反映している。また、実行時モデルをサブモジュールとして別に解析し、全体モデルにはめ込む一つの方法を与え、解析の単純化をはかっている。

以下 2. ではオープンバッチ処理の全体を $M/M/n$ 型にモデル化し解析する。3. では実行段階を、多重度が n で平衡状態となっている場合を想定しモデル化し、平均ジョブ実行率 $\mu(n)$ を算定する。4. においては、平均実行経過時間 ($\mu(n)$ の逆数) について、実行時モデルにおける幾何分布による取り扱いと、全体モデルにおける指数分布の取り扱いを関連づける。5. では、当研究所システムに対するデータを用いて解析結果の数値計算を行い、多重度に対する変化を図示する。

2. モ デ ル

Fig. 1 (次頁参照) はオープンバッチシステムを、入力、実行、出力のフェイズに分けてモデル化したものである。

* Analysis for a Queuing Model of Open Batch Multi-job-processing System by Misako ISHIGURO (Japan Atomic Energy Research Institute)

** 日本原子力研究所

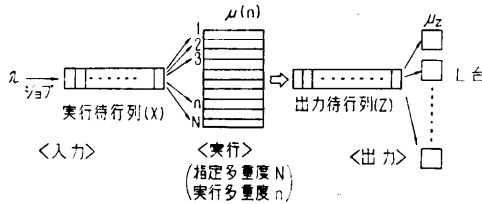


Fig. 1 Model of an Open Batch Processing System

ジョブの到着はパラメータ λ のポアソン分布にしたがうものとし、到着したジョブは FCFS (First Come First Served) のスケジュールにしたがって実行される。ジョブ・クラス、ジョブ優先権は考慮していない。ジョブはあらかじめ指定した多重度 N で実行されるものとし、実行多重度が N 未満の場合に入力したジョブはただちに実行が開始される。すでに実行多重度が N の場合に入力したジョブは実行待行列に入る。

ジョブの実行経過時間 (ジョブ実行開始から終了まで) は、そのときの実行多重度 $n (0 \leq n \leq N)$ に依存したパラメータ $\mu(n)$ を持つ指数分布にしたがうものとする。 $\mu(n)$ は実行時モデルの解析から別途算定される値である。

実行終了後 FIFO により L 台の出力装置 (ラインプリンタを想定している) により並行して出力される。 L 台の出力装置が稼働中のときは、出力待行列に入って待たされる。出力所要時間もパラメータ μ_z を持つ指数分布にしたがうものとする。 L 台のどの装置も同一のパラメータ μ_z の出力率を持つものとする。

Fig. 1 より実行待行列の長さを $x (x \geq 0)$ 、実行時のジョブの多重度を $n (0 \leq n \leq N)$ 、出力待行列の長さとして現在出力中のジョブを加えたものを $z (z \geq 0)$ とし、それらの値が (x, n, z) となる確率を $P(x, n, z)$ で表わすことにする。

(x, n, z) の状態遷移を Fig. 2 で示す。Fig. 2 から平衡状態における方程式を立てると以下の解を得る

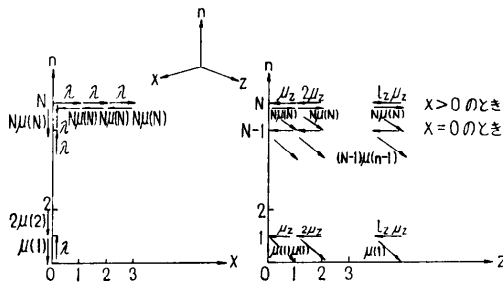


Fig. 2 Figure of State Transition

(付録参照)。

$$P(x, N, z) = \left\{ \frac{\lambda}{N\mu(N)} \right\}^x \left\{ \prod_{i=1}^N \frac{\lambda}{i\mu(i)} \right\} \left\{ \prod_{i=1}^z \frac{\lambda}{L_i\mu_z} \right\} P(0, 0, 0) \quad (1)$$

$$P(x, N, 0) = \left\{ \frac{\lambda}{N\mu(N)} \right\}^x \left\{ \prod_{i=1}^N \frac{\lambda}{i\mu(i)} \right\} P(0, 0, 0) \quad (2)$$

$$P(0, n, z) = \left\{ \prod_{i=1}^n \frac{\lambda}{i\mu(i)} \right\} \left\{ \prod_{i=1}^z \frac{\lambda}{L_i\mu_z} \right\} P(0, 0, 0) \quad (3)$$

$$P(0, 0, z) = \left\{ \prod_{i=1}^z \frac{\lambda}{L_i\mu_z} \right\} P(0, 0, 0) \quad (4)$$

$$P(0, n, 0) = \left\{ \prod_{i=1}^n \frac{\lambda}{i\mu(i)} \right\} P(0, 0, 0) \quad (5)$$

$$\text{ここで } L_i = \begin{cases} i, & 1 \leq i \leq L \\ L, & i > L \end{cases}$$

$$P(0, 0, 0) = \left\{ \left(1 + \sum_{z=1}^{\infty} \prod_{i=1}^z \frac{\lambda}{L_i\mu_z} \right) A(N) \right\}^{-1}$$

$$A(N) = 1 + \frac{\lambda}{\mu(1)} + \frac{\lambda^2}{2\mu(1)\mu(2)} + \dots + \frac{\lambda^{N-1}}{(N-1)! \mu(1)\mu(2) \dots \mu(N-1)} + \frac{\lambda^N}{(N-1)! \mu(1)\mu(2) \dots \mu(N-1) \{ N\mu(N) - \lambda \}} \quad (6)$$

但し (1) ~ (6) 式は $\lambda < N\mu(N)$ の範囲で成立する。 $\lambda \geq N\mu(N)$ の場合は系が暴走しこのモデルでは取り扱えない。

3. 実行時モデル

Fig. 3 に示すような実行多重度が n のときの実行時モデルを作る。ジョブの実行過程は図に示すように、CPU 使用、チャネル使用が交互に繰り返される平均 m 回のパスから成るものとする。タイムスライスによるタスクの切り換えはないものとする。実行時の OS のオーバーヘッドを含むものとする。

1 回 CPU を継続して使用する時間はパラメータ μ を持つ指数分布にしたがうものとする。

計算機の構成は、大型システムを想定し、二つの CPU が同一の能力を有し平等に使用される MP 構成とする。チャネルは、磁気ドラム用の HSC (High

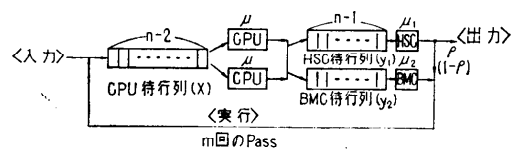


Fig. 3 Model of the Executing Stage

Speed Channel) と磁気ディスク用の BMC (Block Multiplexer Channel) の二つのみを考えに入れる。ラインプリンターなどの入出力装置のためのマルチプレクサーチャネルは実行のフェーズでは使用しないものとし、ディスクを経由して入力・出力フェーズで処理されると考える。磁気テープはオープンパッチ処理という処理形態から使用を禁止されているものとし、その代り大容量の補助記憶装置を持つファイルオリエンテッド・システムとして運用されているものとする。1回のCPU使用後は必ずいずれかのチャネルが使用されるがその比を σ_1, σ_2 とする。ここで $\sigma_1 + \sigma_2 = 1$ 。それぞれのチャネルの使用時間は、パラメータ μ_1, μ_2 を持つ指数分布にしたがうものとする。

ジョブの実行開始は最初に CPU 待行列に入った時点とし、実行終了は最後に特定のチャネル (ここでは HSC) の処理が終了し系を出ていく時点とする。いずれかの CPU が空くと FCFS のスケジュールで CPU 待行列から次の処理要求が一つ取出され処理される。現実のシステムでは実行優先権にもとづいて CPU 割当が行われているが、ここでは優先処理を考慮しないことにする。またジョブ (=タスク) はすべて主記憶上にあると仮定する。

CPU 使用後は HSC か BMC のいずれかのチャネル待行列に入り FCFS のスケジュールで処理される。HSC 使用後は ρ の確率でジョブが終了し、実行時モデルの外に出る。BMC 使用後は必ず CPU 待行列にもどるものとする。

実行時モデルにおいては、1ジョブ当りの CPU $\leftrightarrow$ チャネルのパスの繰り返し数の平均が 10^3 程度の値だから $\rho \approx 1/m$ は小さい値であり、HSC 使用後も CPU 待行列にもどる場合が圧倒的に多く $(1-\rho)$ の確率で起こる。またジョブが終了した場合 (ρ の確率) も、系が多重度 n で平衡状態を保っているという仮定を置くと、ただちに次のジョブが CPU 待行列に入ってくると考えることができる。したがって HSC 使用後も必ず CPU 待行列にもどると考えてさしつかえない。

CPU 待行列の長さ和使用中のジョブ (=タスク) の和を x , HSC, BMC チャネル待行列の長さ和使用中のものの和をそれぞれ y_1, y_2 とする。実行多重度が n のクローズド・システムで系が平衡状態に達したとき、これらの値が (x, y_1, y_2) となる確率 $q(x, y_1, y_2)$ は以下で示される。

$$q(x, y_1, y_2) = \left(\frac{2\mu\sigma_1}{\mu_1}\right)^{y_1} \left(\frac{2\mu\sigma_2}{\mu_2}\right)^{y_2} q(n, 0, 0) \quad (7)$$

$$x > 0, \quad x + y_1 + y_2 = n,$$

$$q(0, y_1, y_2) = \frac{1}{2} \left(\frac{2\mu\sigma_1}{\mu_1}\right)^{y_1} \left(\frac{2\mu\sigma_2}{\mu_2}\right)^{y_2} q(n, 0, 0) \quad (8)$$

$$y_1 + y_2 = n,$$

$$q(n, 0, 0) = \left[\frac{1}{(1-\rho_1)(1-\rho_2)} + \frac{1}{2(\rho_2-\rho_1)} \left\{ \frac{1+\rho_1}{1-\rho_1} \rho_1^{n+1} - \frac{1+\rho_2}{1-\rho_2} \rho_2^{n+1} \right\} \right]^{-1}$$

$$\rho_1 = \frac{2\mu\sigma_1}{\mu_1}, \quad \rho_2 = \frac{2\mu\sigma_2}{\mu_2}, \quad (9)$$

$q(n, 0, 0)$ は、 $\rho_1 \ll 1, \rho_2 \ll 1$ のときは $(1-\rho_1)(1-\rho_2)$ に近づく。

次に、多重度が n の場合のジョブの平均実行率 $\mu_T(n)$ を計算する。

定時間当りシステムが実行するジョブの数 $\mu_T(n)$ は次式となる。

$$\mu_T(n) = U_{HSC}(n) \mu_1 \rho, \quad (10)$$

ここで $U_{HSC}(n)$ は HSC が使用中となる確率で以下のようにして算出する。

$$U_{HSC}(n) = 1 - \sum_{x=0}^n q(x, 0, n-x), \quad (11)$$

つまり $\mu_T(n)$ は、実行多重度が n の場合のシステム全体のジョブの終了率であり、実行時のスループットである。1ジョブ当りのジョブ終了率 $\mu(n)$ は、次のように与えることができる。

$$\mu(n) = \mu_T(n)/n. \quad (12)$$

ジョブの平均実行経過時間 $e(n)$ は、

$$e(n) = 1/\mu(n), \quad (13)$$

となる。平均 CPU 使用時間 $U_{CPU}(n)$, BMC 使用時間 $U_{BMC}(n)$ は次の通りである。

$$U_{CPU}(n) = 1 - \sum_{y_1=0}^{n-1} q(0, y_1, n-y_1) - \frac{1}{2} \sum_{y_1=0}^n q(1, y_1, n-1-y_1), \quad (14)$$

$$U_{BMC}(n) = 1 - \sum_{x=0}^n q(x, n-x, 0). \quad (15)$$

装置間の同時使用率も同様にして求められる。

4. 実行経過時間の分布に関する考察

実行多重度が n のときのジョブ実行経過時間の分布は、平均ジョブ実行率 $\mu(n)$ をパラメータとした指数分布

$$f(x) = 1 - e^{-\mu(n)x}, \quad (16)$$

に属すると仮定している。

実行時モデルの解析において (Fig. 3), 1 回のパスに対しジョブが系から出ていく確率は $\sigma_1 \rho$ であり, 再び CPU 待行列にもどる確率は $(1 - \sigma_1 \rho)$ である。したがってジョブが k 回のパスで終了する確率 P_k は, 幾何分布となる。

$$P_k = \sigma_1 \rho (1 - \sigma_1 \rho)^{k-1}, \quad (17)$$

ここで σ_1 は HSC を使用する率, ρ は HSC 使用後系から出ていく率である。 m をジョブ当りの平均パス数とすると次の関係が成立する。

$$\sigma_1 \rho = \frac{1}{m}, \quad (18)$$

したがって

$$P_k = \frac{1}{m} \left(1 - \frac{1}{m}\right)^{k-1}. \quad (19)$$

実行多重度が n のときの 1 パス当りの経過時間の平均を $w(n)$ とすれば, P_k はジョブの経過時間が $kw(n)$ となる確率である。したがってジョブの経過時間が x 以下となる確率は $x = kw(n)$ とおくと,

$$\begin{aligned} q(x) &= \frac{1}{m} + \frac{1}{m} \left(1 - \frac{1}{m}\right) + \dots + \frac{1}{m} \left(1 - \frac{1}{m}\right)^{\{x/w(n)-1\}} \\ &= 1 - \left(1 - \frac{1}{m}\right)^{x/w(n)}. \end{aligned} \quad (20)$$

ここで $w(n)$ は, m が十分大きい場合には, 大数の法則にしたがってある一定の値をとるものとする。

一方, 平均のジョブの実行率 $\mu(n)$ は平均実行経過時間の逆数だから,

$$\mu(n) = 1/m \cdot w(n), \quad (21)$$

したがって, 次式を得る。

$$q(x) = 1 - \left\{ \left(1 - \frac{1}{m}\right)^m \right\}^{\mu(n)x}. \quad (22)$$

$$\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m = e^{-1} \text{ から,}$$

m が十分大きい場合には, $q(x)$ は $f(x)$ に近づく。

$$q(x) \approx 1 - e^{-\mu(n)x} = f(x).$$

m はジョブ当りの平均の入出力総数だから, 現実には 10^3 程度の値をとる。したがって実行時の実行経過時間の分布を指数分布 $f(x)$ としてさしつかえない。

5. 解析結果

いくつかの解析結果を多重度の関数として与え, 数値計算結果を図示する。

(1) 平均のジョブ実行経過時間 (Fig. 4)

(10)~(13)式から $\mu(n)$ を, 原研システム B (F230-75 MP) システム, 1000 k 語の主記憶装置) に対して

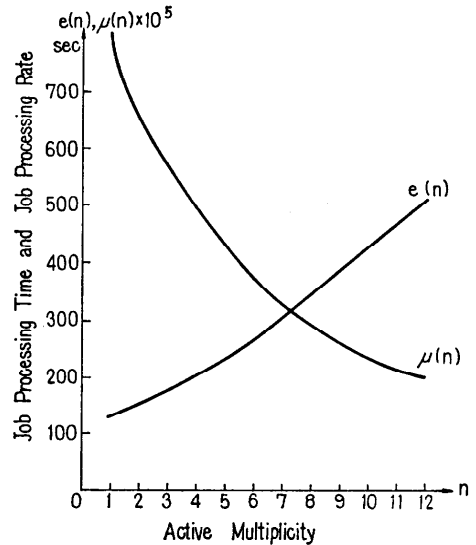


Fig. 4 Mean Job Processing Time and Processing Rate

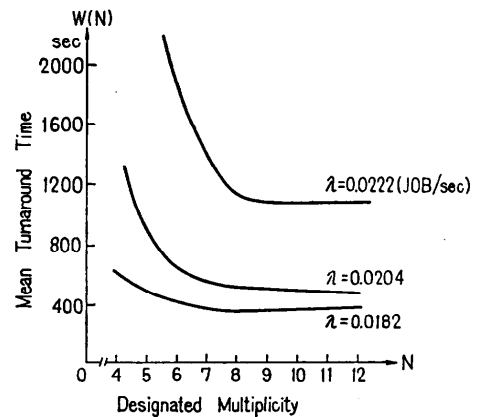


Fig. 5 Mean Job Turnaround Time

計算する。パラメータの設定値は以下のとおりである。

$\mu = 46$ (タスク/秒), $\mu_1 = 100$ (タスク/秒), $\mu_2 = 76$ (タスク/秒), $\sigma_1 = 0.74$, $\sigma_2 = 0.26$, $m = 3893$, $\rho = 3.5 \times 10^{-4}$ 。

$e(n)/n$ は n がある値より大きいときはほぼ一定値をとる。これは HSC の使用率が増加しないことによる。

(2) 平均ターンアラウンドタイム (Fig. 5)

オープンバッチシステムにおけるターンアラウンドタイムはジョブを入力してから出力終了までの時間であると定義する。

指定多重度が N のときのジョブの平均のターンアラウンドタイム $W(N)$ の算式を示す。

ターンアラウンドタイムは次の三項目の和と考えられる。

① 実行待行列上で待時間

$$\begin{cases} 0, & n < N \\ (x+1)/N\mu(N), & n = N \end{cases}$$

多重度が N のときは、現在実行中のジョブの1つが終了するまでの待時間も含む。

② 実行時間

$$1/\mu(n)$$

③ 出力待および出力時間

$$\begin{aligned} & 1/\mu_z + \max(0, z-L)/L\mu_z \\ W(N) = & \sum_{z=0}^{\infty} \sum_{n=1}^{N-1} \{1/\mu(n) + \max(0, z-L)/L\mu_z \\ & + 1/\mu_z\} P(0, n, z) \\ & + \sum_{z=0}^{\infty} \sum_{x=0}^{\infty} \{x+N+1\}/N\mu(N) \\ & + \max(0, z-L)/L\mu_z \\ & + 1/\mu_z\} P(x, N, z). \end{aligned} \quad (23)$$

数値計算に必要なパラメータの設定を追加する。

$\lambda = 0.0222$ (=ジョブ/45秒), $\lambda = 0.0204$ (=ジョブ/49秒), or $\lambda = 0.0182$ (=ジョブ/54秒), $\mu_z = 0.01$ (ジョブ/100秒), $L = 5$, $\mu(1) \sim \mu(N)$ は計算による。

(3) スループット

スループットはジョブの平均終了率を意味し(24)式となる。

$$\begin{aligned} T(N) = & \sum_{z=0}^{\infty} l_z \mu_z \left\{ \sum_{n=0}^{N-1} P(0, n, z) + \sum_{x=0}^{\infty} P(x, N, z) \right\} \\ l_z = & \begin{cases} z: z \leq L \\ L: z > L \end{cases} \end{aligned} \quad (24)$$

スループットは、M/M/1 モデルでは入力率に等しいことが知られている。この場合のようにモデルが、M/M/N から M/M/1 の間にある確率で動く場合も、 $T(N)$ は N に依存しない値となる。

$$T(N) = \sum_{z=1}^{\infty} l_z \mu_z \left(\prod_{i=1}^z \frac{\lambda}{l_i \mu_{l_i}} \right) \cdot A(N) \cdot P(0, 0, 0). \quad (25)$$

$A(N)$ は(6)式の計算と同一のものである。(25)式に(6)式を代入する。

$$T = \frac{\sum_{z=1}^{\infty} l_z \mu_z \left(\prod_{i=1}^z \frac{\lambda}{l_i \mu_{l_i}} \right)}{1 + \sum_{z=1}^{\infty} \left(\prod_{i=1}^z \frac{\lambda}{l_i \mu_{l_i}} \right)}, \quad (26)$$

(4) 平均実行多重度

$$M(N) = \sum_{z=0}^{\infty} \left\{ \sum_{n=0}^{N-1} n P(0, n, z) + N \sum_{x=0}^{\infty} P(x, N, z) \right\}. \quad (27)$$

(5) 指定多重度で実行される確率 (Fig. 6)

$$R(N) = \sum_{z=0}^{\infty} \sum_{x=0}^{\infty} P(x, N, z). \quad (28)$$

(6) 平均実行待行列長 (Fig. 7)

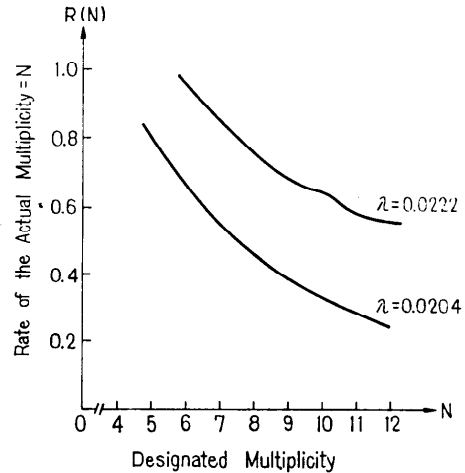


Fig. 6 Rate of the Actual Multiplicity = N

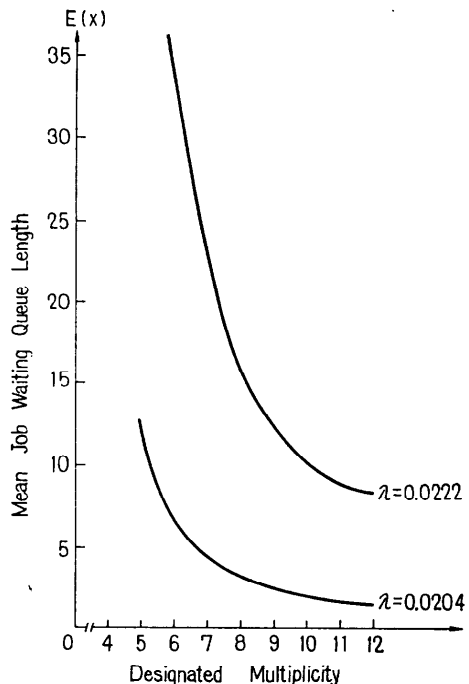


Fig. 7 Mean Job Waiting Queue Length

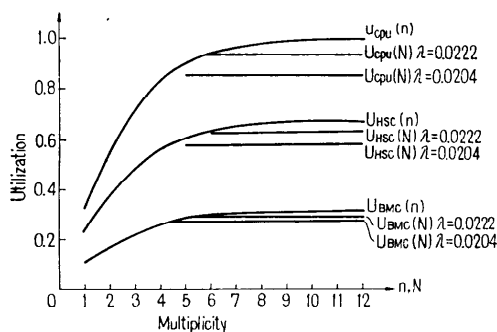


Fig. 8 CPU and Channels Utilization

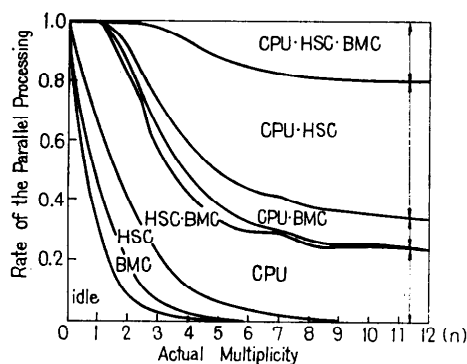


Fig. 9 Rate of the Parallel Processing Between the Devices

$$I(N) = \sum_{z=1}^{\infty} \sum_{x=0}^{\infty} x P(x, N, z), \quad (29)$$

(7) 装置の使用率 (Fig. 8)

$$U_{CPU}(N) = \sum_{z=0}^{\infty} \sum_{n=0}^{N-1} U_{CPU}(n) P(0, n, z) + U_{CPU}(N) R(N), \quad (30)$$

$$U_{HSC}(N) = \sum_{z=0}^{\infty} \sum_{n=0}^{N-1} U_{HSC}(n) P(0, n, z) + U_{HSC}(N) R(N). \quad (31)$$

$$U_{BMC}(N) = \sum_{z=0}^{\infty} \left\{ \sum_{n=0}^{N-1} U_{BMC}(n) P(0, n, z) + U_{BMC}(N) R(N) \right\}. \quad (32)$$

(7) 装置の同時使用率 (Fig. 9)

算定方法は(11)(14)(15)式に準じる。図は、CPU, HSC, BMC の同時使用（並行動作）率の平均を、それぞれの状態に対する割合で区分けたものである。

6. 検 討

最後に、ここで解析した計算機モデルに現実のシス

テムの動きを十分に反映し得なかった事項を明記する。

(1) OS のオーバヘッド

OS のオーバヘッドは、多重度が上ると増加するものと予想されるが、原研システムにおける測定では、これをはっきりと立証づけるデータが得られなかった。よって測定時の平均値を取りユーザ CPU 使用時間の30%とした。

(2) ジョブクラスと優先処置

実行待行列からジョブを取り出す場合の優先処理、実行時に CPU 待行列からタスクを取り出す場合の優先処理、ジョブクラスによるジョブの実行開始の調整は今回考慮しなかった。優先処理はプライオリティ待行列の解析により解決されよう。ジョブクラスは、このモデルを少し拡張するだけで簡単に取り扱い得る。

(3) ロールアウトされているジョブ

主記憶装置（実記憶）にすべてのジョブが存在する場合のみを想定している。原研システムでは主記憶が1M 語あるので多重度が10の場合でも、0.5ジョブの割合、つまり10ジョブ中9.5ジョブが主記憶内にあるという状況から、考慮しなかった。

(4) タイムスライス

原研システムでは、タイムスライスが0.1秒と大きく、その影響が少ないので今回モデルからはずした。タイムスライスを考慮するには、実行時モデルの中で CPU 使用後再び CPU 待行列にもどる場合をモデルに追加しなければならない。

(5) 現システムの解析に用いたデータ

数値計算を実行するにあたり、マイクロサムによるハードウェア測定、PANSY (F 230-75 効率測定用サブモニター) によるソフトウェア測定および会計情報からの集計を使用した。チャンネルの使用回数のうち会計情報にカウントされないもの (OS のオーバヘッド部分) については推定値を使用した。HSC, BMC はそれぞれ2台ずつ接続されているが、1台に換算した値を使用した。

参 考 文 献

- 1) E.G. Coffman, JR. and P.J. Denning: Operating Systems Theory, Prentice-Hall, (1973)
- 2) I. Adiri, M. Hofri and M. Yadin: A Multiprogramming Queue, J. ACM. V 20-4, pp. 589 ~603, (Oct. 1973),
- 3) 宮崎他: 多重プログラミング・システムの効率について, 情報処理学会誌, V 16-12, pp. 1040~

1047. (Dec. 1975)

- 4) W. J. Gordon and G. F. Newell: Closed Queuing Systems with Exponential Servers, Oper. Res. Vol. 15, pp. 254~265 (1967)

- 5) 紀 一誠: 資源切り出し型待行列モデルによるメモリ競合問題の解析, システム性能評価研究会資料 13, (1976年1月)

付 録

(x, n, z) の状態遷移を Fig. 2 で示す. Fig. 2 から平衡状態における方程式を立てると以下のようになる.

- (a) $x \geq 1, n = N, z > 1$ のとき

$$\begin{aligned} \lambda P(x-1, N, z) + N\mu(N)P(x+1, N, z-1) \\ + l_{z+1}\mu_z P(x, N, z+1) = \{\lambda + N\mu(N) \\ + l_z\mu_z\} P(x, N, z), \end{aligned}$$

ここで

$$l_z = \begin{cases} z, & 1 \leq z \leq L \\ L, & L \leq z \end{cases} \quad (1)$$

- (b) $x \geq 1, n = N, z = 0$ のとき

$$\begin{aligned} \lambda P(x-1, N, 0) + \mu_z P(x, N, 1) \\ = \{\lambda + N\mu(N)\} P(x, N, 0), \end{aligned} \quad (2)$$

- (c) $x = 0, n = N, z \geq 1$ のとき

$$\begin{aligned} \lambda P(0, N-1, z) + N\mu(N)P(1, N, z-1) \\ + l_{z+1}\mu_z P(0, N, z+1) = \{\lambda + N\mu(N) \\ + l_z\mu_z\} P(0, N, z). \end{aligned} \quad (3)$$

- (d) $x = 0, n = N, z = 0$ のとき

$$\begin{aligned} \lambda P(0, N-1, 0) + \mu_z P(0, N, 1) \\ = \{\lambda + N\mu(N)\} P(0, N, 0). \end{aligned} \quad (4)$$

- (e) $1 \leq n < N, z \geq 1$ のとき ($x=0$)

$$\begin{aligned} \lambda P(0, n-1, z) + (n+1)\mu(n+1)P(0, n+1, z-1) \\ + l_{z+1}\mu_z P(0, n, z+1) = \{\lambda + n\mu(n) \\ + l_z\mu_z\} P(0, n, z). \end{aligned} \quad (5)$$

- (f) $1 \leq n < N, z = 0$ のとき ($x=0$)

$$\lambda P(0, n-1, 0) + \mu_z P(0, n, 1) = \{\lambda + n\mu(n)\} P(0, n, 0). \quad (6)$$

- (g) $n = 0, z \geq 1$ のとき ($x=0$)

$$\begin{aligned} \mu(1)P(0, 1, z-1) + l_{z+1}\mu_z P(0, 0, z+1) \\ = \{\lambda + l_z\mu_z\} P(0, 0, z). \end{aligned} \quad (7)$$

- (h) $n = 0, z = 0$ のとき ($x=0$)

$$\mu_z P(0, 0, 1) = \lambda P(0, 0, 0). \quad (8)$$

$\lambda, \mu(1) \sim \mu(N), L, \mu_z$ を既知量として $P(x, n, Z)$

を求める.

次の三つの関係式は, 方程式(1)~(8)式を満たす.

$$\textcircled{1} \quad P(x, N, z) = \frac{\lambda}{N\mu(N)} P(x-1, N, z), \quad x \geq 1 \quad (9)$$

$$\textcircled{2} \quad P(0, n, z) = \frac{\lambda}{n\mu(n)} P(0, n-1, z), \quad 1 \leq n \leq N \quad (10)$$

$$\textcircled{3} \quad P(x, n, z) = \frac{\lambda}{l_z\mu_z} P(x, n, z-1), \quad z \geq 1 \quad (11)$$

これらの関係から, 方程式の一つの特殊解として以下を得る.

$$P(x, N, z) = \left\{ \frac{\lambda}{N\mu(N)} \right\}^x \left\{ \prod_{i=1}^N \frac{\lambda}{i\mu(i)} \right\} \left\{ \prod_{i=1}^z \frac{\lambda}{l_i\mu_z} \right\} P(0, 0, 0) \quad (12)$$

$$P(x, N, 0) = \left\{ \frac{\lambda}{N\mu(N)} \right\}^x \left\{ \prod_{i=1}^N \frac{\lambda}{i\mu(i)} \right\} P(0, 0, 0) \quad (13)$$

$$P(0, n, z) = \left\{ \prod_{i=1}^n \frac{\lambda}{i\mu(i)} \right\} \left\{ \prod_{i=1}^z \frac{\lambda}{l_i\mu_z} \right\} P(0, 0, 0) \quad (14)$$

$$P(0, 0, z) = \left\{ \prod_{i=1}^z \frac{\lambda}{l_i\mu_z} \right\} P(0, 0, 0) \quad (15)$$

$$P(0, n, 0) = \left\{ \prod_{i=1}^n \frac{\lambda}{i\mu(i)} \right\} P(0, 0, 0) \quad (16)$$

ここで

$$l_i = \begin{cases} i, & 1 \leq i \leq L \\ L, & i \geq L \end{cases}$$

一方

$$\sum_{z=0}^{\infty} \left\{ \sum_{n=0}^{N-1} P(0, n, z) + \sum_{x=0}^{\infty} P(x, N, z) \right\} = 1$$

より(12)~(16)式を代入して計算すると,

$$\left(1 + \sum_{z=1}^{\infty} \prod_{i=1}^z \frac{\lambda}{l_i\mu_z} \right) A(N) P(0, 0, 0) = 1$$

$$P(0, 0, 0) = \left\{ \left(1 + \sum_{z=1}^{\infty} \prod_{i=1}^z \frac{\lambda}{l_i\mu_z} \right) A(N) \right\}^{-1}$$

$$A(N) = 1 + \frac{\lambda}{\mu(1)} + \frac{\lambda^2}{2\mu(1)\mu(2)}$$

$$+ \cdots + \frac{\lambda^{N-1}}{(N-1)! \mu(1)\mu(2) \cdots \mu(N-1)}$$

$$+ \frac{\lambda^N}{(N-1)! \mu(1)\mu(2) \cdots \mu(N-1) \{N\mu(N) - \lambda\}} \quad (17)$$

(昭和51年6月30日受付)

(昭和51年9月17日再受付)