

計算機システムにおける待ち行列

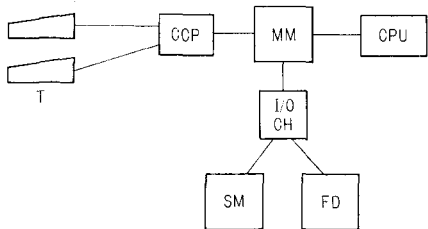
——モデルと評価尺度——

橋 田 温

多数の待ち行列に関する論文が学会誌や会議等で発表されているが、その応用分野の最たるものが計算機システムであろう。その要因としては、既存の解析結果がそのまま応用できない変形モデルが多いこと、分布形などの実測が容易であること、数学モデルを抽出しやすいこと、などが考えられる。また、計算機システムで待ち行列モデルが用いられる目的は、オペレーティングシステムの構成の評価、資源 (resource) 間のインタフェースや機能分担方法の比較、システム全体の性能評価とその改良方法の検討、および方式構成が定まったシステムの量的設計、などである。

計算機システムには多くの形態があるが、図1のような端末オリエンテッドなシステムが代表的であろう。ここで、SMはMMより溢れた情報を蓄積するメモリ、FDはユーザープログラムやファイル情報を蓄積するメモリ、I/OチャンネルはMMとSMまたはFDの間で情報転送を制御する装置である。またCCPは、端末とCPUを中心としたセンター装置間の通信を制御する装置である。このシステムでは、端末からの処理要求(ジョブ)がセンターへ到着し、CPU、メモリなどの資源で処理を受けて、再び端末へ戻る。これらのシステム構成要素を待ち行列の用語と対応させると表1のとおりとなる。

図1のようなシステムの総合特性を評価するには、ま



CPU：中央処理装置
MM：主メモリ
SM：2次メモリ
FD：ファイルメモリ
I/O CH：入出力チャンネル
CCP：通信制御装置
T：端末

図1 計算機システムの例

表1 計算機システムと待ち行列の対応

計算機システム	待ち行列
ジョブ	客
資源	窓口
処理 (含 read/write)	サービス
メモリ	待ち室
端末	入力源

ず個々の資源に分割したサブシステムの特性を求め、その後それらを総合したトータルシステムとしての特性を求める方法がとられる。以下では、サブシステムとしての待ち行列モデルと評価尺度について説明し、最後にそれらを総合した待ち行列ネットワークモデルについて説明する。

1. CPU における待ち行列モデル

CPUでの待ち行列モデルは、MMでの割付方法も含めたジョブのスケジューリングアルゴリズムにあらわれる。スケジューリングは、バッチ処理 (batch processing)とTSS(Time Sharing System)に大別される。バッチ処理では、ジョブは処理完了またはSMやFDに記憶されている情報が必要となるまで連続して処理されるため、優先権のある場合も含めた普通の待ち行列モデルとなる。

TSSでは、CPUは量子(quantum)とよばれるかぎられた時間だけ各ジョブの処理を行ない、その間に完了しないジョブは再び待ち行列へ戻す方法がとられ、量子処理(quantum processing)とよばれる。

この場合、多数のジョブが同時にセンターに受け入れられて、CPUが各ジョブを細切れで処理するため、あたかも各ジョブがCPUを専有しているかのように見える。このようにすると、処理時間の短いジョブは結果的に早く系を去るため、ジョブの処理時間に応じて優先権があるのと同じ効果がある。

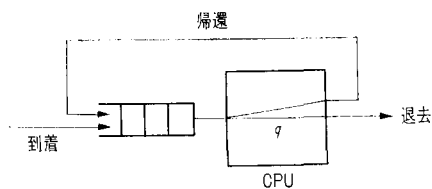


図 2 RR 方式

バッチ処理や量子処理において、ジョブの完了時や割り込み時には、MMとSMまたはFDの間でプログラムや情報の受け渡しをするための時間(swap time)や、ジョブの監視・割付けのための時間など、ジョブ固有の処理時間に含まれない時間が必要となる。したがって、これらを考慮すると準備時間(setup time)のあるモデルとなる。

量子処理では、量子内で処理が完了しないジョブを待ち行列へ戻すため、一般に帰還形待ち行列(feed back queue)となるが、つぎの2つが代表的である。¹⁾²⁾

(1) ラウンドロビン方式(RR方式, Round-Robin): 図2のように待ち行列が1つで、量子長 q の間に処理が完了しないジョブは、処理を中断されて待ち行列の最後尾へ帰還される。

(2) 多段帰還方式(FB方式, Foreground-Back ground): 図3のように N 段の待ち行列があり、ジョブは第1段へ到着する。第 j 段のジョブは、第1~($j-1$)段のジョブがないときのみ処理され、量子長 q_j の間に処理が完了しなければ第($j+1$)段の最後尾へ帰還される。

RR方式に比して、FB方式のほうが処理時間の短いジョブの系内時間をより短くする効果がある。RR方式で同様の効果を出すには、量子処理を受ける回数につれて量子長を短くすればよい。

RR方式またはFB方式で、量子長 $\rightarrow 0$ の極限状態を考慮プロセッサシェアード方式(PS方式, Processor-Sha-

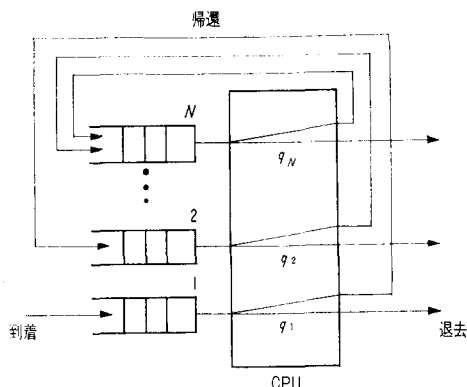


図 3 FB方式

red)とよぶ。これらは、RR または FB 方式の特性をもったままで、解析が容易なモデルのためよく用いられる。RR-PS 方式では、系内ジョブ数が n のとき各ジョブはフルタイムで処理を受けるが、処理速度は $1/n$ となる。FB-PS方式では、新たに到着したジョブは処理中のジョブに割り込んで処理を受け、その経過処理時間(attained service time)が割り込まれたジョブのそれに達すると、割り込まれたジョブも一緒にRR-PS方式で処理される。

ジョブの到着がポアソン過程(パラメータ λ)で準備時間=0 の場合、処理時間が x のジョブの平均系内時間 $\bar{T}(x)$ は次式で与えられる²⁾。

$$\bar{T}(x) = \begin{cases} \frac{x}{1-\rho} & \text{(RR-PS 方式)} \\ \frac{\bar{W}_x + x}{1-\rho_x} & \text{(FB-PS 方式)} \end{cases} \quad (1.1)$$

ここで処理時間分布を $B(x)$ 、その平均を $\bar{h}$ とすれば、 $\rho = \lambda \bar{h}$, $\rho_x = \lambda \bar{h}_x$, $\bar{h}_x = \int_0^x y^v dB(y) + y^v \{1-B(y)\}$, $\bar{W}_x = \lambda \bar{h}_x^2 / 2(1-\rho_x)$ である。処理時間 x に対する $\bar{T}(x)$ 特性の例を図4に示す。

CPU のスケジューリングアルゴリズムに対する評価尺度は、後述するシステム応答時間の一部である系内時間である。RR 方式において、系内時間にコストを導入して平均システムコストを最小にする量子長の最適化モデルも提案されている³⁾。

2. メモリにおける待ち行列モデル

MMにおける待ち行列モデルとしては、マルチプロセッサシステムにおけるメモリアクセスの競合がある。図5のように複数個のプロセッサがメモリユニットを共用する時、同一メモリユニットへのアクセス(read/write)が衝突するために待ち行列ができる。普通、メモリへの

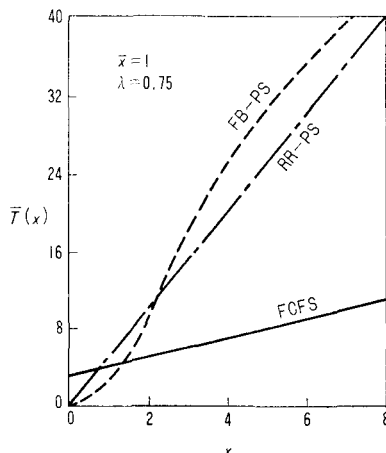


図 4 平均系内時間の比較

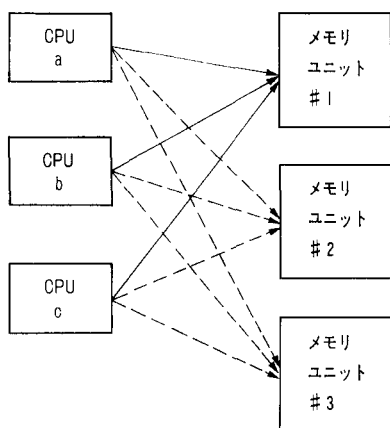


図 5 マルチプロセッサシステムでの競合

アクセスは一定周期で行なわれ(メモリサイクル), 各メモリサイクル間の状態推移がマルコフ連鎖をなすモデルとなる⁶⁾。

SMまたはFMは、大容量でかつ安価なディスクやドラムなどの回転形メモリが用いられる。ドラムには、レコード(データ単位)の開始位置およびレコード長が任意(確率変数)であるファイルドラム, 開始位置が定められているセクターファイルドラム, およびレコード長がセクター内に制限されているページングドラムなどがある。ここでセクターとは図6のように, ドラムを回転軸に平行な扇柱に分割したときの扇柱の部分であり, ページングドラムではレコード単位をページという。

ドラム処理の要求が到着してから, MM等への転送が終了するまでのシーケンスは図7のようになる。要求の

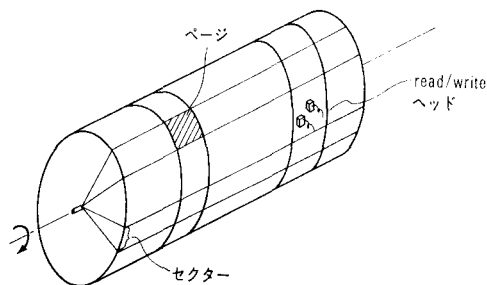


図 6 ページングドラム

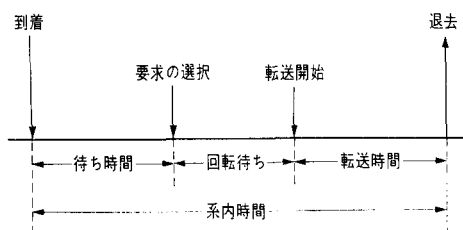


図 7 ドラム処理のシーケンス

選択から転送開始までの回転待ち時間は潜在時間 (latency time)ともよばれ, 回転形メモリに特有の待ち時間である。ドラム処理のスケジューリングとしては, FCFSの他に, SLTF (Shortest Latency Time First) がある。

例として, k 個のセクターに分割されている FCFS-ページングドラムを考えてみよう。転送時間 x_i は, ドラムの回転時間を τ とすれば τ/k であり, 要求が一樣に分布していれば, 前の転送終了からつぎの要求が転送開始されるまでの回転待ち時間 x_r は, $P_r\{x_r = i\tau/k\} = 1/k$ ($i=0, 1, \dots, k-1$)である。一方要求がないときに到着した要求に対しては, 1セクター内に一樣分布する余分の時間 x_i が回転待ち時間に加わる。したがって, サービス時間 $h = x_r + x_i$ とすれば, 周期処理待ち行列⁵⁾または強制空き時間のあるモデル⁶⁾となる。すなわち強制空き時間 $h_i = x_i$ とすれば, ドラム処理の平均系内時間 $\bar{T}$ は次式で与えられる。($\bar{h}$, $\bar{h}^2$ はそれぞれ h の 1 次・2 次モーメント)

$$\bar{T} = \frac{h_i^2}{2\bar{h}_i} + \frac{\lambda \bar{h}^2}{2(1-\lambda \bar{h})} + \bar{h} \quad (2.1)$$

他のドラムモデルも同様に解析されるが, SLTF-ファイルドラムでは, つぎの要求までの回転待ち時間の分布が $P_r\{x_r \leq t\} = 1 - (1-t/\tau)^n$ (n : 待ち要求数)となり, 待ち要求数に依存するため解析が困難である。 x_r の分布を待ち要求数に依存するパラメータの指数分布で近似した解析がある。⁷⁾ 各ドラムモデルに対する平均系内時間の数値例を図8に示す。

ディスクでは, ドラムと同様の回転待ち時間の他に, ヘッドがトラック上を移動するための探索時間 (seek time)が必要となる(図9)。ディスク処理におけるシーケンスは図10のようになる。ディスク処理のスケジューリングとしては, FCFS, SSTF (Shortest Seek Time First) および, ヘッドが一定の方向に移動しつつ要求を

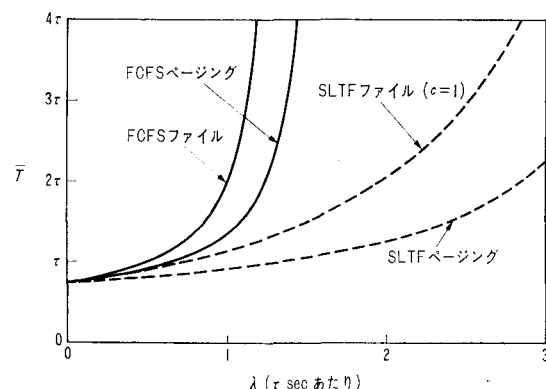


図 8 ドラム処理スケジューリングの比較

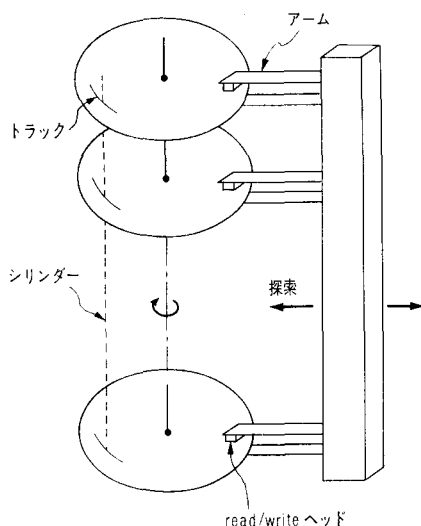


図 9 ディスクメモリ

処理しながらトラック幅を往復する SCAN(走査式)がある⁹⁾。SSTF は、平均系内時間を短くする点で有効であるが、トラック幅の中央部に対する要求が早く処理され、周縁部が後回しにされる欠点がある。したがって、ディスクスケジューリングの評価では、公平さの尺度として、系内時間の平均のみでなく分散等も考慮する必要がある。

例として、各シリンダー(同じ番号のトラック集合)への要求が同じパラメータ λ のポアソン到着の SCAN モデルを考えると、 i シリンダーへ対する要求の平均待ち時間(図10の待ち時間+探索時間) $\bar{W}_i$ は次式で与えられる⁹⁾。

$$\bar{W}_i = \frac{A}{2} \frac{h^2(1+\lambda h)/(1-\lambda h^2) + (a/N)[(1-\lambda h)/(1-\lambda h)] \cdot [(N-i)^2 + i^2]}{[(N-i)^2 + i^2]} \quad (2.2)$$

ここで、 h は回転待ち時間+転送時間、 $N+1$ はシリンダー数、 a は隣り合ったシリンダー間をヘッドが移動する時間、 $A=(N+1)\lambda$ である。SCAN の場合でもシリンダー位置によって $\bar{W}_i$ に差が生じるが、その数値例を図11に示す。

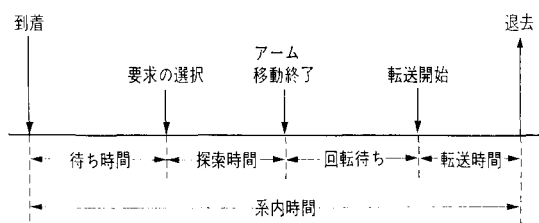


図 10 ディスク処理のシーケンス

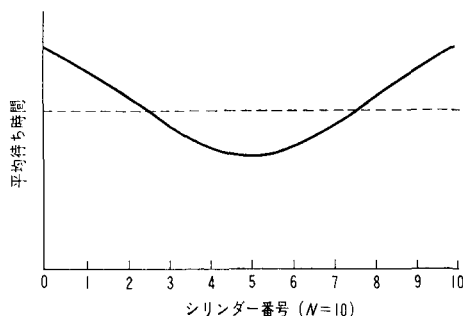


図 11 SCAN 方式での平均待ち時間
(回転待ち時間を除く)

3. 端末制御における待ち行列モデル

図1のように通信回線で結ばれた多数の端末を通信制御装置が制御する場合、制御開始時点および制御時間がともに連続量である方式と、開始時点が周期的でその制御時間が一定である同期式が代表的である。

(1) 連続制御方式

通信制御装置から順番に各端末における要求の有無を調べ、もし要求があればそれを制御してつぎの端末へ移る方法をポーリング(polling)方式という。この場合、制御が端末間を移動するための時間が必要であり、歩行時間のある多重待ち行列モデルとなる。多重待ち行列には、端末の要求をすべて処理してからつぎの端末へ移る全処理式、制御が端末に到着したときまでに発生した要求のみを処理するゲート式、および、端末に並んでいる要求のうち先頭から一定個数までを処理する制限式、などがある。^{10),11)} このモデルでの評価尺度は、端末から見た端末制御による待ち時間であり、平均待ち時間の数値例を図12に示す。

他の方式として、端末側から独立に要求を発生し、他

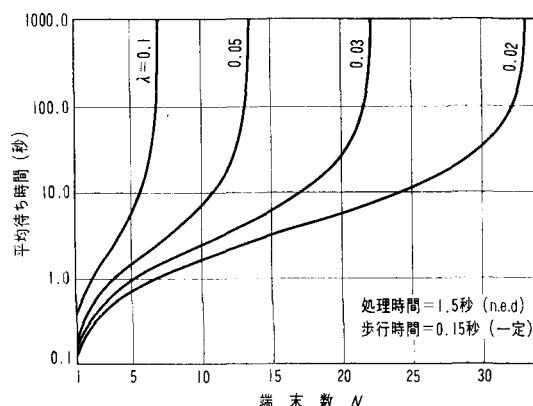


図 12 全処理式多重待ち行列の平均待ち時間

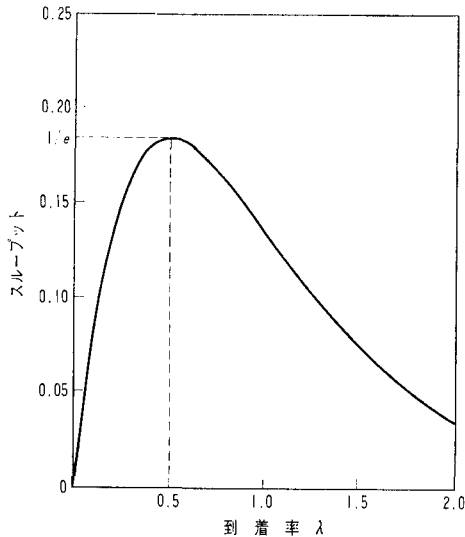


図 13 衝突によるスループット低下特性

の要求と衝突しなければ、制御を受けるコンテンション (contention) 方式がある。この場合、制御に対して競合が起こるが、その様子を無線回線のモデルで見てみよう。多数の端末から制御時間一定(τ)の要求がポアソン過程(λ)で発生し、もし他の要求と制御時間が一部でも重なれば、無線信号のため受信不能とする。2つ以上の要求が衝突する確率は $1 - \exp(-2\lambda\tau)$ である。評価尺度として τ 時間あたりに完了した制御要求数、すなわちスループットを考えれば、 $\lambda \exp(-2\lambda\tau)$ となり、スループット対入力特性は図13のような低下特性を示す。²⁾

(2) 周期式制御方式

時間は一定長(=1とする)のタイムスロットに分割され、各タイムスロットでは1データユニットずつ制御が行なわれる。このモデルには、一定周期ごとに巡ってくるタイムスロットを特定の端末へ割当てる方式(STD, Synchronous Time Division), 空きタイムスロットを自由に使用できる方式(ATD, Asynchronous Time Division), および空きタイムスロットをポーリング的に使わせる方式(PTD, Polling Time Division)がある。これらのモデルは、賭博者破産問題(gambler's ruin problem)の結果を用いて、統一的に解析されている。¹²⁾各端末での統計的性質が同じ場合、端末 $\#i$ における m データユニットの要求の平均系内時間 $\bar{T}_i(m)$ は、それぞれ次式で与えられる。

$$\bar{T}_i(m) = \begin{cases} \frac{N+1}{2} + m(N-1) + N \left(\frac{N\sigma^2}{2(1-N\mu)} - \frac{\mu}{2} \right) & (\text{STD}) \\ \frac{m+i\lambda/2(1-i\lambda)-i\lambda/2+(i-1)\lambda}{1-(i-1)\lambda} - (m-1) & (\text{ATD}) \\ \frac{N\sigma^2}{2(1-N\mu)} + \frac{Nr(1-\mu)}{2(1-N\mu)} + \frac{1-\mu}{2} & (\text{PTD}) \end{cases} \quad (3.1)$$

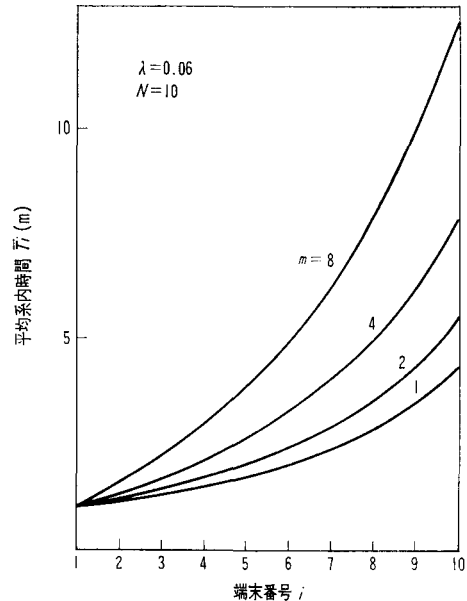


図 14 端末制御における built-in priority の例

ここで、 μ と σ^2 は各端末に1タイムスロット間に到着するデータユニット数の平均と分散、 N は総端末数、 r は PTD の場合の移行時間(r タイムスロット)である。また ATD では、各端末への要求はポアソン到着でパラメータを λ 、データユニット長は1とする。ATD で端末番号 i が関係するのは、衝突を起こさないように空きタイムスロットを使用する順番を端末番号と合わせたためであり、図14のように優先権がつくため built-in priority とよばれる。

4. 総合システムとしての待ち行列モデル

図1のような計算機システムにおいては、ユーザーは端末からジョブの処理要求を発生し、センターで CPU, SM, FD などで処理を何回か受けてその結果が端末へ戻り、ユーザーの思考時間を経て再びジョブが発生する。このように、ジョブの発生からその応答(結果)が端末へ戻ってくるまでのシーケンスをインタラクション(interaction)とよび、インタラクションによってジョブが実行されていくシステムをインタラクティブな計算機システム(interactive computer system)という。

インタラクティブな計算機システムでは、ユーザーがジョブを発生すると、もしジョブのプログラムが MM に記憶されてあればそのまま CPU の処理待ち行列へ加えられ、もしそれが MM 内になければ、FD から MM へ移される。CPU でのジョブの処理は、量子の終了時、FD 内の情報が必要な時、および MM が満杯の時に中断される。量子終了時に中断されたジョブは CPU の処理

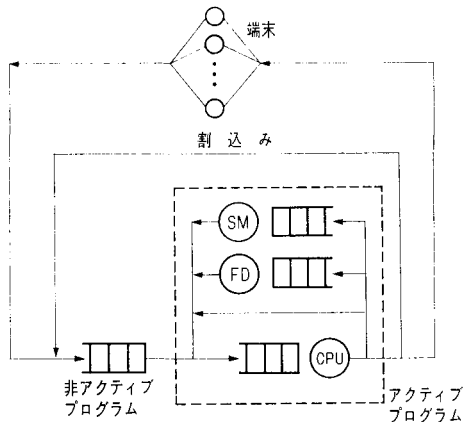


図 15 インタラクティブ計算機システムのモデル

待ち行列へ戻され、MM 内の記憶領域が不足したときは、もっとも不要と思われるプログラムや情報を MM から SM へ移す。

CPU, SM, FDなどで処理を受けるためにMM 内にあるプログラム(またはジョブ)はアクティブプログラム(またはジョブ)とよばれる。アクティブプログラムが2個以上のシステムを多重プログラミングシステム(multiprogramming system)といい、そのプログラム数をプログラム多重度(degree of programming)という。

図1において端末制御の部分を省略し、端末まで含めたインタラクティブな計算機システムとしてモデル化すると、図15のような待ち行列ネットワークとなる¹³⁾。図では、優先権の高いジョブの割り込みによってMMより移される場合も示してある。また点線で囲まれた部分にあるプログラムがアクティブなもので、プログラム多重度を制限した場合には、非アクティブなプログラムの待ち行列が形成される。とくに重負荷の場合には、1つのプログラムの処理が終了するとただちに非アクティブなプログラムがアクティブとなるため、図15の点線内のサブシステムの特徴は、図16のモデルと同じになり、セントラルサーバーモデル(central server model)とよばれる。¹⁴⁾ また、SM, FDをまとめて1つの待ち行列と

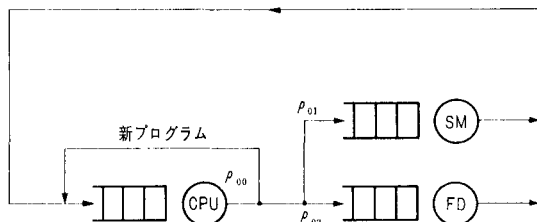


図 16 マルチプログラミングシステムに対する central server model

したモデルも提案され、系外からの入力も考慮した2段階循環形待ち行列モデルとなる¹⁵⁾。これらの待ち行列ネットワークに対し、最近種々のアプローチが試みられているため、以下に説明する。

(1) マルコフ形の待ち行列ネットワーク

つぎの条件(i), (ii), (iii)を満たすマルコフ形の場合は、Jackson¹⁶⁾やGordon & Newell¹⁷⁾の方法で解析される。

- (i) 各資源でのサービス時間は指数分布にしたがい、そのサービス率はそこでの系内ジョブ数に依存する。
- (ii) ジョブにクラスがない。
- (iii) $\#i$ 資源でサービスを終了したジョブは、 $\#j$ 資源へ確率 p_{ij} で移動しサービスを受ける。

いま、 $p(n_1, n_2, \dots, n_M)$ を $\#i$ 資源の系内ジョブ数が n_i ($i=1, \dots, M$)である定常状態確率とすれば

$$p(n_1, n_2, \dots, n_M) = \frac{\prod_{i=1}^M \left(\lambda_i n_i / \prod_{j=1}^n \mu_i(j) \right)}{G(N)} \quad (4.1)$$

となる。ただし、 M は資源数、 N はプログラム多重度、 $\mu_i(j)$ は $\#i$ 資源の系内ジョブ数が j のときのサービス率、 λ_i は $\#i$ 資源への相対到着率であり

$$\lambda_j = \sum_{i=1}^M \lambda_i p_{ij} \quad 1 \leq j \leq M \quad (4.2)$$

また、 $\mathbf{n} = (n_1, n_2, \dots, n_M)$, $S(M, N) = \{(\mathbf{n}_1, \dots, \mathbf{n}_M) | \sum_{i=1}^M n_i = N, \text{ and } n_i \geq 0\}$ である。

同時確率 $p(n_1, n_2, \dots, n_M)$ から周辺分布や平均値などの値が求められるが、それらは $G(1), G(2), \dots, G(N)$ が計算されれば容易に求まる。したがって、 $G(1), \dots, G(N)$ を計算する効率のよいアルゴリズムも提案されている¹⁸⁾。

総合システムの評価尺度としては、スループット(throughput)、CPU使用率、平均応答時間(mean response time)、などが用いられる。スループットとは、システムとしてのジョブの処理速度であり、単位時間あたりの平均完了ジョブ数であらわされる。スループットは各資源に対しても、単位時間あたりの平均完了要求数で定義される。とくにシステムとしてのスループットを総合スループットとよび区別しよう。 $\#i$ 資源の周辺分布を $p_i(n_i)$ 、その系内要求数が n_i のときのサービス率を $\mu_i(n_i)$ とすれば、 $\#i$ 資源のスループット θ_i は次式で定義される。

$$\theta_i = \sum_{n_i=1}^N p_i(n_i) \mu_i(n_i) \quad (4.3)$$

また総合スループット θ は、図15のセントラルサーバー

モデルに対し

$$\theta = \sum_{n_0=1}^N p_0(n_0) \mu_0(n_0) p_{00} \quad (4.4)$$

である。ただし、添字 0 は CPU をあらわし、 p_{00} は CPU の処理が終了したジョブが再び CPU 待ち行列へ戻る確率であり、ジョブの処理完了に対応する。とくに、 $\mu_0(n_0) = \mu_0(n_0 > 0)$, $= 0 (n_0 = 0)$ の場合には、 η_0 を $1 - p_0(0)$ 、すなわち CPU 使用率とすれば

$$\theta = \eta_0 \mu_0 p_{00} \quad (4.5)$$

となり、総合スループットは CPU 使用率に比例する。したがって、総合スループットは、CPU 使用率特性で代用される。ここで、多重プログラミングシステムの特徴を CPU 使用率の例で見よう。あるプログラムの実行中に SM(または FD)にある情報が必要になると、ページ単位で情報が転送されるとする。プログラム実行

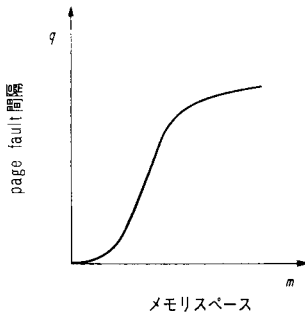


図 17 page fault 間隔

中にページが必要となる間隔(page fault 間隔) q は統計的規則にしたがい、1 プログラムが使用できるMM内のメモリ領域 m の関数である。普通、 q と m の間には図17のような関係があり、かつ、 A を MM の全領域とすれば $m = A/N$ である。これらを用いて多重プログラミングシステムをセントラルサーバーモデルで解くと、CPU 使用率対プログラム多重度の特性は図18のように

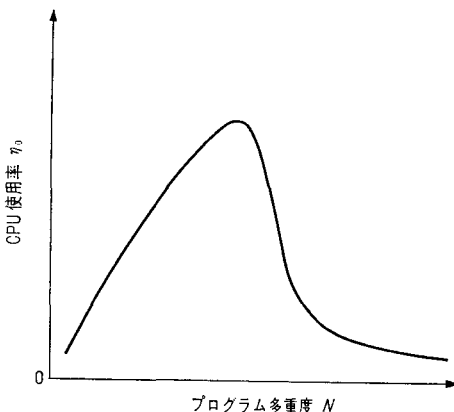


図 18 CPU 使用率特性

なる。¹⁹⁾ したがって、スループットを低下させないためには、プログラム多重度を適当な数に制限する必要がある。

スループットに関連して、ボトルネック(bottleneck)という尺度もときどき用いられるが、その定義は明確でない。1つの方向として、 $\#i$ 資源の処理能力(μ_i)を上げると総合スループットが大きく向上する場合、 $\#i$ 資源がボトルネックであるとも考えることもできる。¹⁴⁾ すなわち、 $\partial\theta/\partial\mu_i$ が他より大きい資源をボトルネックとよび、もしとくに大きな値がなければ、バランスがとれたシステムという。また処理能力 $\{\mu_i\}$ が与えられているとき、総合スループットを最大にするジョブの割付方策 $\{p_{ij}\}$ について考察すると、もし $\mu_i = k\mu_j (k > 1)$ ならば、 $p_{0i} > kp_{0j}$ すなわち $\bar{n}_i > \bar{n}_j$ とする方策が望ましい。すなわち、高処理能力の資源に多くのジョブを回すほうが望ましいといえる。¹⁴⁾

応答時間とは、インタラクションにおいて、端末からジョブを発生してから、その応答(結果)が再び端末へ戻ってくるまでの時間であるが、ここでは端末制御に要する遅延を除く。あるジョブの1インタラクションあたり、 $\#i$ 資源で受ける処理回数を m_i 、それぞれの平均系内時間を $\bar{t}_i$ とすれば、そのジョブの平均応答時間 $\bar{R}$ は次式のようになる。

$$\bar{R} = \sum_{i=1}^M m_i \bar{t}_i \quad (4.6)$$

ここで、 $\bar{t}_i$ は Little の公式より $\bar{t}_i = \bar{n}_i / \mu_i \eta_i$ である。 m_i は page fault 間隔、ジョブの性質、CPU のスケジューリングなどに依存するため、その値を推定するという問題もある。

(2) 局所平衡の成り立つ待ち行列ネットワーク

前項の条件 (i)~(iii) が成り立つ待ち行列ネットワークでは、平衡状態確率 $p(\mathbf{n}^i)$ に対してつぎの平衡方程式(balance equation)が成り立つ。(Sは状態集合)

$$\begin{aligned} p(\mathbf{n}^i) \cdot [\text{状態 } \mathbf{n}^i \text{ から出ていく割合}] \\ = \sum_{\mathbf{n}^j \in S} p(\mathbf{n}^j) \cdot [\text{状態 } \mathbf{n}^j \text{ から状態 } \mathbf{n}^i \text{ へ入る割合}] \end{aligned} \quad (4.7)$$

これを全域平衡方程式(global balance equation)ともよぶ。一方、 $\#i$ 資源と状態 $\mathbf{n}^j$ に注目して

$$\begin{aligned} [\#i \text{ 資源へのジョブの到着により } \mathbf{n}^j \text{ へ遷移する} \\ \text{割合}] = [\#i \text{ 資源からのジョブの退去により } \mathbf{n}^j \\ \text{から遷移する割合}] \end{aligned} \quad (4.8)$$

が成り立つ場合を局所平衡(local balance)とよび、上式を局所平衡方程式(local balance equation)とよぶ。²⁰⁾ 状態 $\mathbf{n}^j$ を固定して局所平衡方程式を加えると平衡方程式を得るため、前者は後者の十分条件となっている。また局所平衡方程式は、ジョブにクラスがある場合にもそ

のまま成り立つため、クラスごとの特性を求めるのに有効である。

つぎの条件を満たす待ち行列ネットワークでは局所平衡が成り立つ。

- a) ネットワークへのジョブの到着はポアソン過程か、クラスごとに独立なポアソン過程で、それらのパラメータは系内ジョブ数に依存してもよい。
- b) ネットワークに到着したクラス r のジョブが最初に $\#i$ 資源のサービスを受ける確率が b_{jr} 、クラス r のジョブが $\#i$ 資源のサービス終了後クラス s の客となり $\#j$ 資源へ移動する確率が $p_{i,r;j,s}$ で与えられる。
- c) 各資源はつぎの4サービスタイプのうちのどれかである。
 - (タイプ1) FCFS- $\cdot/M/1$: 平均サービス率は系内ジョブ数の関数でもよい。
 - (タイプ2) PS- $\cdot/G_r/1$: G_r はラプラススチュエルチェス変換が有理関数であるサービス時間分布でクラスごとに異なってもよい。
 - (タイプ3) $\cdot/G_r/\infty$: 同上
 - (タイプ4) Preemptive-Resume LCFS- $\cdot/G_r/1$: 同上

タイプ1はSM, FDなどのI/Oデバイス向きで、タイプ2とタイプ4はCPU向きで、タイプ3は端末でのサービスタイプに向いている。また G_r に対しても局所平衡が成り立つのは、有理関数のラプラススチュエルチェス変換をもつサービス時間分布が、指数分布の窓口から成るネットワーク形の総合サービス時間分布で等価的にあわせるためである。²¹⁾ G_r の中には、アーラン分布、超指数分布なども含まれるため、局所平衡の成り立つモデルはかなり一般的である。

局所平衡が成り立つ待ち行列ネットワークに対しては平衡状態確率がつぎのような積形式 (product form) で与えられることが証明されている。²⁰⁾

$$P_r\{s=(x_1, x_2, \dots, x_M)\} = Cd(s) f_1(x_1) f_2(x_2) \cdots f_M(x_M) \quad (4.9)$$

ここで、 x_i は $\#i$ 資源での系内ジョブ数(クラス別)のベクトル、 s は系の状態、 $d(s)$ は状態 s での系内ジョブ数のみによる関数、 f_i は $\#i$ 資源における状態 x_i による関数、 C は正規化定数である。

局所平衡が成り立たない待ち行列ネットワークモデルも現実には多い。それらに対しても多くの近似解法が試みられている。たとえば、ジョブの移動確率 $\{p_{ij}\}$ が他の資源における系内ジョブ数にも依存する場合²²⁾、指数分布サービス時間分布でもサービス率が他資源における系内ジョブ数にも依存する場合¹⁹⁾、サービス時間分布が

一般分布の場合^{23), 24)}、などがある。

一般に page fault がある多重プログラミングシステムでは、MMの共用によるスループット低下が図18のように起こる。このスループット低下を防止するにはプログラム多重度を制限すればよいが、各資源の使用率を測定しながらダイナミックにプログラム多重度をコントロールするモデルも提案されている。²⁵⁾

また、待ち行列ネットワークモデルを用いて実際のシステムの性能を評価した例もいくつか発表されている。たとえば、測定によるとCPUサービス時間は超指数分布に近く、I/Oデバイスのサービス時間分布は指数分布に近いが、実測値をパラメータの値として使ったマルコフ形の待ち行列ネットワークでの解析結果は、資源使用率や平均応答時間の測定結果の10%以内に入ること、が得られている²⁶⁾。

む す び

計算機システムにあらわれる代表的な待ち行列モデルと、よく用いられる評価尺度について解説的に述べた。紙面の都合で省略したものに、バッファメモリやI/Oチャネルでの待ち行列モデルがある。前者は有限待ち室の待ち行列モデルとなり、後者はMMにおける競合と類似したモデルとなる。現在は、各サブシステムに対する評価がひととおり終了し、トータルシステムとしての性能評価のためのモデル化と解析が行なわれている段階と思われる。

とくに待ち行列ネットワークに対する解析の一般化という方向で、資源数が多い場合における等価変換や分解などの簡易手法の研究^{27), 19)}、一般サービス時間分布や他サービス規律の場合における近似手法 (たとえば拡散近似^{28), 29)}、超指数分布近似²³⁾ など)の研究、などが進められている。また、それらの解析手法を使いやすくするために、プログラム化する努力も行なわれている³⁰⁾。

参 考 文 献

- 1) 藤木正也 “通信トラヒック理論の基礎と応用、待ち行列理論(その3)”, 電子通信学会誌, 55, 10 (1972), 1335-1341.
- 2) Kleinrock, L., Queueing Systems, Volume II : Computer Applications, John Wiley & Sons, New York, 1976.
- 3) Heacox, H.C. and R.W. Purdon, "Analysis of Two Time-sharing Queueing Models", J. ACM, 19, 1(1972), 70-91.
- 4) Skinner, C. E. and J. R. Asher, "Effects

- Storage Contention on System Performance", IBM Syst. J., 8, 4(1969), 319-333.
- 5) Fujiki, M., "Queueing Delay Formula for Common-Channel Signaling Systems", IEEE Trans. COM-20, (1972), 787-791.
 - 6) Powell, B.A. and B. Avi-Itzhak, "Queueing Systems with Enforced Idle Time", Opns. Res., 15, 6(1967), 1145-1156.
 - 7) Fuller, S. H., Analysis of Drum and Disk Storage Units, Springer-Verlag, Berlin(1975).
 - 8) Coffman, E.G., L. A. Klimko and B. Ryan, "Analysis of Scanning Policies for Reducing Disk Seek Times", SIAM J. Comput., 1, 3 (1972), 269-279.
 - 9) Oney, W. C., "Queueing Analysis of the Scan Policy for Moving-Head Disks", J. ACM, 22, 3(1975), 397-412.
 - 10) 橋田 温, 中村義作, "複数待ち行列の解析", 経営科学, 13, 1(1969), 30-47.
 - 11) 橋田 温, "ゲート式多重待ち行列", 電子通信学会論文誌, 53-A, 1(1970), 43-50.
 - 12) Chou, W. W. and Konheim, A. G. "On the Analysis and Modeling of a Class of Computer Communication Systems", IEEE Trans., COM-20, 3(1972), 645-660.
 - 13) Muntz, R. R., "Analytic Modeling of Interactive Systems", Proc. IEEE 63, 6(1975), 946-953.
 - 14) Buzen, J. P., "Analysis of System Bottlenecks Using a Queueing Network Model", Proc. ACM-SIGOPS Workshop on System Performance Evaluation (Apr. 1971), 82-103.
 - 15) Franta, W. R., "The Mathematical Analysis of the Computer System Modeled as a Two Stage Cyclic Queue", Acta Informatica, 6 (1976), 187-209.
 - 16) Jackson, J. R., "Jobshop-like Queueing Systems", Manag. Sci., 10, 1(1963), 131-142.
 - 17) Gordon, W. J. and Newell, G. F. "Closed Queueing Systems with Exponential Servers", Oper. Res., 15(1967), 254-265.
 - 18) Buzen, J. P., "Computational Algorithms for Closed Queueing Networks with Exponential Servers", Commun. ACM, 16, 9(1973), 527-531.
 - 19) Brandwajn, A., "A Model of a Time Sharing Virtual Memory System Solved Using Equivalence and Decomposition Methods", Acta Informatica, 4(1974), 11-47.
 - 20) Baskett, F., K. M. Chandy, R. R. Muntz and F.G. Palacios, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers", J. ACM, 22, 2(1975), 248-260.
 - 21) Cox, D. R., "A Use of Complex Probabilities in the Theory of Stochastic Processes", Proc. Cambridge Phil. Soc. 51(1955), 313-319.
 - 22) Chen, P.P.-S., "Queueing Network Model of Interactive Computing Systems", Proc. IEEE, 63, 6(1975), 954-957.
 - 23) Sauer, C. H. and K. M. Chandy, "Approximate Analysis of Central Server Models", IBM J. Res. Develop., 19, 3(1975), 301-313.
 - 24) Chow, W. M., "Central Server Model for Multiprogrammed Computer Systems with Different Classes of Jobs", IBM J. Res. Develop., 19, 3(1975), 314-320.
 - 25) Badel, M., E. Gelenbe, J. Leroudier and D. Potier, "Adaptive Optimization of a Time-Sharing System's Performance", Proc. IEEE, 63, 6 (1975), 958-965.
 - 26) Moore, C. G., III, "Network Models for Large-Scale Time-Sharing Systems", Dep. Indust. Eng., Univ. Michigan, Ann Arbor, Tech. Rep. 71-1, (Apr. 1971).
 - 27) Chandy, K. M., U. Herzog and L. Woo, "Parametric Analysis of Queueing Networks", IBM J. Res. Develop., 19, 1(1975), 36-42.
 - 28) Kobayashi, H., "Application of the Diffusion Approximation to Queueing Networks I: Equilibrium Queue Distributions", J. ACM, 21, 2(1974), 316-328.
 - 29) Gelenbe, E., "On Approximate Computer System Models", J. ACM, 22, 2(1975), 261-269.
 - 30) Reiser, M., "Interactive Modeling of Computer Systems", IBM Syst. J., 15, 4(1976), 309-327.

(はしだ・おん 日本電信電話公社
武蔵野電気通信研究所)