

パッシブ・サーバをもつネットワーク型 待ち行列を用いた計算機の性能評価法†

池 原 悟††

従来のネットワーク型待ち行列を用いた計算機システムの動作解析では CPU, I/O 装置のようなハードウェア資源をサーバとして構成された解析モデルが用いられてきた。ところが、実際のシステムでは制御表、ファイル等のシリアル・リユーザブルなソフトウェア資源に対するアクセス競合が性能を決定する要因となることがあり、ハードウェア、ソフトウェアの両種のサーバを考慮に入れた計算機のモデル化が必要である。

本論文では計算機システムを構成するサーバを、①単独でジョブ処理を行うアクティブ・サーバ（ハードウェア資源に対応）と、②アクティブ・サーバの制御のもとにジョブを実行するパッシブ・サーバ（ソフトウェア資源に対応）とに分け、アクティブ・サーバからパッシブ・サーバへのアクセス要求をランダム生起で近似したときのモデル解析手順を示した。

次に、本解法の具体的な適用例として制御表へのアクセス競合およびアディケーティド・メモリへのアクセス競合のある分散処理方式の例を2つとりあげ、本解析法による解析結果とシミュレーション結果を比べ、本解析法の実用上の有効性を示した。

1. ま え が き

従来計算機の設計、評価、改良に役立てられて来た待ち行列¹⁾⁻³⁾には特定の資源の動作状態に着目した局所的なモデルと計算機システムを構成する各資源の相互作用を総合的に扱おうとするネットワーク型待ち行列モデルがある。前者はそれぞれの部品もしくは装置ごとに異なる解析モデルが設定され解析されて来たのに対して、後者は汎用の技術であり、今後の計算機および計算機間通信網解析への適用が期待されている。

ネットワーク型の待ち行列の解法はジャクソンの定理⁴⁾に基礎をおくもので、Gordon, Newell⁵⁾によってクローズドタイプのモデルの基本的解法が示され、Buzen⁶⁾によってその計算機解法が与えられたものである。この方法は当初は単一種類（指数型）のサーバで構成されたクローズド・タイプのネットワーク型モデルに限定され、扱うジョブも単一種類のものではあったが、その後 Baskett, Chandy 等⁷⁾によって4種のサーバで構成されるオープン／クローズド混合したネットワーク型待ち行列で複数タイプのジョブが扱えるところまで拡張され、さらに Reiser, 小林^{8),9)}によってその計算機解法が研究されている。

この方法は複数のサーバを渡り歩くジョブのモデルを解析する上で強力な方法であるが、計算機の性能評

価を行うためにはまだいくつかの欠陥がある。その主なものをあげると、①リソースの同時使用のあるモデルは扱えない、②サーバとジョブのタイプが自由に設定できない（特に FCFS のサーバについて）、③ジョブのスプリット／ギャザ／消滅およびジョブ間の同期がとれない、④割込み機構が扱えない、などがあり、これらの点まで考慮する場合はシミュレーションによるのが普通である。Sauer 等¹⁰⁾はネットワーク型待ち行列の解法とシミュレータを組み合わせるこれらの問題を解くパッケージを作成している。

本論文は上記①の問題の1つのモデル化とその解法を提案するものである。複数のリソースを同時に専有するジョブモデルを扱ったものとしては、システムの処理容量の計算アルゴリズム¹¹⁾や複数資源の切り出し型の待ち行列の研究¹²⁾、またジョブが複数のタスクに分かれて資源を専有するモデルの研究^{13),14)}などがある。本論文はジョブ処理の途上で単一のジョブが複数の資源を同時に専有使用することのあるネットワーク型待ち行列の解法を示したものである。すなわち、まず1つのサーバ（資源）上でジョブが処理されている間に断続的に任意の数の他のサーバを専有することができるが1つのジョブが同時に専有することのできるサーバ数は高々1つであることを仮定し、計算機システムを構成するサーバを単独でジョブ処理可能なアクティブ・サーバとアクティブ・サーバから使用されるパッシブ・サーバに分け、計算機の解析モデルを3つのタイプに分類した。

† Computer Performance Evaluations by Queuing Networks with Passive Servers by SATORU IKEHARA (Yokosuka Electrical Communication Laboratories, N. T. T.).

†† 日本電信電話公社横須賀電気通信研究所

次にそのうちの第1のタイプのモデルにおいてアクティブ・サーバからパッシブ・サーバのアクセス要求をランダム生起で近似したときのモデル解法手順を示した。

最後に本解法の具体的な適用例として制御表へのアクセス競合およびデディケートドメモリへのアクセス競合のある分散処理方式の例を2つとり上げ、本解析法による解析結果とシミュレーション結果を比べ、本解析法の実用上の有効性を示した。

2. パッシブ・サーバをもつ計算機のモデル化

2.1 アクティブ・サーバとパッシブ・サーバ

従来の単一資源に着目した動作解析ではソフトウェア資源もハードウェア資源も等しくサーバとして扱われていたのに対して、ネットワーク型待ち行列を用いた計算機の動作解析ではソフトウェア資源は表に現われず、ハードウェア資源のみで構成された解析モデルが用いられている¹⁵⁾。ところが、実際のシステムではソフトウェア資源のネックにより、スループットが上がらないことしばしばある。そこで、ここでは両種のサーバを考慮に入れた計算機のモデル化が必要である。

さて、ハードウェア、ソフトウェアの資源としての性格の違いに着目する。例えばCPUやI/Oデバイスはある仕事を与えられるとそれぞれ固有の能力をもって処理を行う資源であり、待ち行列ではサーバとしてモデル化される。これに対して、プログラムや制御表、ファイルなどのソフトウェア資源についてみると、これらの中にはあるジョブで使用中は他のジョブで使用できないといった排他的性格をもつものも多く、アクセス競合のある資源である場合が多いが、前者が単独で処理可能な資源であるのに対して後者は前者の制御のもとにのみ処理は行われる特徴をもつ。ハードウェア資源の中でも主メモリなどソフトウェアに付属する資源も後者に属し、ソフトウェア資源に対する位置占め(オキュペーション)の問題¹⁶⁾を提供している。

ここで、計算機を構成する資源をアクティブ・サーバとパッシブ・サーバに分類する。アクティブ・サーバはCPU、I/O装置、チャンネルなど自ら処理を実行するもので真のサーバともいう。パッシブ・サーバは

単独では処理を実行できず、アクティブ・サーバの制御のもとに使用されるもので、疑似サーバともいう。パッシブ・サーバはさらにシリアル・リユーザブルな資源である制御表、ファイル、プログラム等と、ある容量内では多重処理可能なメモリとに分けられる。但しメモリは場合によっては前者と同様に扱われる。

2.2 計算機モデルの分類

アクティブ・サーバとパッシブ・サーバで構成された計算機モデルのうちアクティブ・サーバのみで構成される部分を取り出してアクティブ系、パッシブ・サーバとそれに付随するアクティブ・サーバを含めてパッシブ系と呼ぶ。ここではアクティブ系の構成は文献5), 6)の範囲の待ち行列モデル*で表わされるものとする。すなわちアクティブ系は

- (i) M 個のステージから構成されたクローズドタイプの M/M 型のネットワーク型待ち行列で、
- (ii) 各ステージは待ち行列と $r_i (i=1, 2, \dots, M)$ 台の指数サーバで構成される(同一ステージを構成するサーバのサービス・レートは等しいとする)。
- (iii) ジョブは各ステージ間を確率的に定まった順序で渡り歩いて処理が行われ、
- (iv) 常時ジョブの多重度は一定とする。

次に、パッシブ系はアクティブ系のある定まったステージからアクセスされるモデルとする。

そこで、両者から構成される計算機モデルを次の3つのタイプに分類する。

(1) タイプ1のモデル (図1)

アクティブ系の複数サーバで構成されたステージの各サーバから単一のパッシブ・サーバにアクセスの行

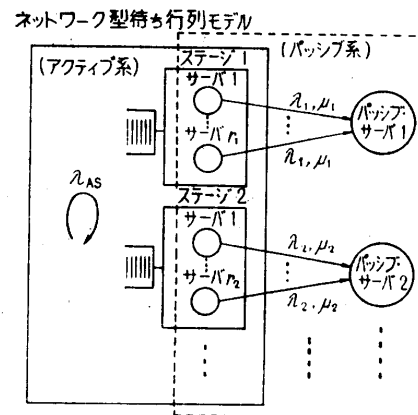


図1 パッシブ・サーバを有するネットワーク型待ち行列(タイプ1)

Fig. 1 Queuing network model with passive servers (Type 1).

* 文献 7), 8) の範囲のモデルとしても特に問題はないと思われるが、4章で文献 6) のアルゴリズムを使用しているため、範囲を小さくした。

** すべての入線の呼量等しいときは文献 17) に解法が示されている。呼原ごとに呼量の異なるときは文献 7), 8) の解析モデルの特殊ケースとして解が求まる。

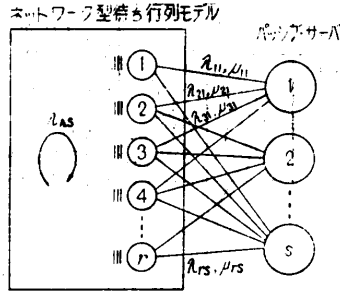


図 2 パッシブ・サーバを有するネットワーク型待ち行列 (タイプ 2)
Fig. 2 Queuing network model with passive servers (Type 2).

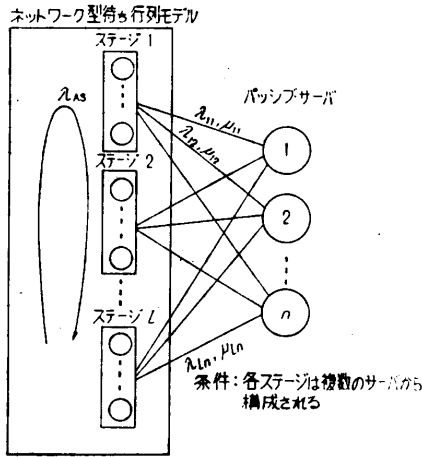


図 3 パッシブ・サーバを有するネットワーク型待ち行列 (タイプ 3)
Fig. 3 Queuing network model with passive servers (Type 3).

られるモデルで、パッシブ・サーバへのアクセスは単一のステージ内のサーバに限定される。

(2) タイプ 2 のモデル (図 2)

アクティブ系の単一のサーバで構成される複数のステージからパッシブ・サーバにアクセスの行われるモデル。

(3) タイプ 3 のモデル (図 3)

任意の複数ステージ (単一サーバでも複数サーバでもよい) からパッシブ・サーバへアクセスの行われるモデル。

以上の分類ではパッシブ系は第 1 のタイプではすべての入線の入呼数密度が等しいのに対して、他のモデルでは入線ごとに入呼数密度が異なる点に特徴がある。

3. タイプ 1 のモデルの解法

以下では第 1 のタイプのモデルに限定し、パッシブ

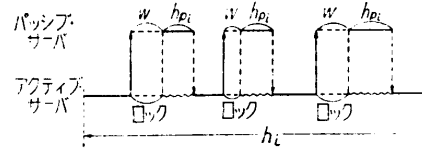


図 4 アクティブ・サーバとパッシブ・サーバの動作
Fig. 4 Correlation between active server and passive server.

系を入線有限待時式線群で近似したときの解法を示す。

3.1 前提条件

(1) アクティブ系のステージ i のすべてのサーバ r_i 台からパッシブ・サーバ i へアクセスが行われるとし、その回数はアクティブ系の i ステージの処理 1 回当たり平均 d_i 回で、アクセス間隔は幾何分布をとるものとする。またパッシブ・サーバ保留時間は平均 h_{pi} の指数分布とする。

(2) 図 4 に示すようにパッシブ・サーバ上での処理中はその処理を依頼したアクティブ・サーバも同時処理が行われており、パッシブ・サーバ待ちの状態ではアクティブ・サーバはロックされているとする。

3.2 アクティブ系の解析

2.2 節の前提条件から、パッシブ系でのアクセス競合によるロックのないときのアクティブ系の動作解析には Buzen の方法が適用できて、文献 13) と同様にしてスループット、レスポンスタイムが求まる。

3.3 パッシブ系の入線有限待時式線群での近似

ステージ i の r_i 台のプロセッサのうち任意 1 台のプロセッサの動作に着目すると、このプロセッサにジョブが回って来て処理を行うごとにパッシブ・サーバに平均 d_i 回のアクセスが行われる。逆にこのプロセッサが空き状態のときはパッシブ・サーバへの呼は発生しない。従ってアクティブ・サーバからパッシブ・サーバへの呼の発生は間歇的に行われるが、これをランダム生起で近似し、平均入呼数密度 λ_{pi} を以下の方法で定める。

すなわち、今仮りにステージ i が使用率 1 で働いていて、パッシブ・サーバへのアクセス競合のない状態を考えると、この状態では平均時間 $h_i (=1/\mu_i)$ の間にパッシブ・サーバに d_i 回のアクセスが行われるから、アクティブ・サーバへの平均呼空間隔 $1/\lambda_{pi}$ は $h_i/d_i - h_{pi}$ となる。但し h_{pi} はパッシブ・サーバ i の平均保留時間とする。ここで、ステージ i のプロセッサの使用率が ρ_i だとすると、平均呼空間隔 $1/\lambda_{pi}$ は $1/\lambda_{pi}$ の $1/\rho_i$ 倍に増大するから、平均入呼数密度 λ_{pi}

は結局,

$$\lambda_{pi} = \frac{\rho_i}{h_i/d_i - h_{pi}} \quad (1)$$

で与えられる.

パッシブ・サーバの平均保留時間 h_{pi} は与えられている. そこで, 上記の方法で求めた平均入呼数密度 λ_{pi} を用いて, アクティブ系のステージ i の r_i 台のプロセッサからのパッシブ・サーバへのアクセス動作は入線ごとの入呼量を $\lambda_{pi} h_{pi}$ とする入線有限待時式線群 $M(r_i)/M/1$ で表わすことができる. よって, パッシブ・サーバへのアクセス競合によるアクティブ・サーバの保留時間の伸び $W_{pi} \mu_{pi}$ および, パッシブ・サーバの使用率 ρ'_{pi} はそれぞれ,

$$W_{pi} \mu_{pi} = \frac{\sum_{l=1}^{r_i-1} l(r_i-1)_l \rho^l_{pi}}{\sum_{l=0}^{r_i-1} (r_i-1)_l \rho^l_{pi}} \quad (2)^*$$

$$\rho'_{pi} = 1 - 1 / \sum_{l=0}^{r_i} (r_i)_l \rho^l_{pi} \quad (3)$$

ただし $\rho_{pi} = \lambda_{pi} / \mu_{pi}$ で与えられる¹⁵⁾.

3.4 系全体の解析方法

パッシブ・サーバへのアクセス競合がない場合のアクティブ系の動作は 3.2 節の方法で正しく解析できて各ステージの使用率 ρ_i が求まる. これを用いて(4)式からパッシブ系への平均入呼数密度 λ_{pi} を求め, (2)式を用いてパッシブ系へのアクセス競合によるパッシブ・サーバ待ち時間 W_{pi} を求めると, W_{pi} はパッシブ・サーバへのアクセス競合によるアクティブ系ステージ i のサーバの保留時間の伸びであると考えられるから, ステージ i の平均保留時間 $h_i (=1/\mu_i)$ はパッシブ・サーバの影響によって h_i^* に増大する. ここで h_i^* は

$$\left. \begin{aligned} h_i^* &= h_i + \Delta h_i \\ \Delta h_i &= d_i W_{pi} \end{aligned} \right\} \quad (4)$$

である.

よって, システム全体のスループット, レスポンスタイム等の特性を求めるにはアクティブ系のサービスレートを $\mu_i^* = 1/h_i^*$ に変更し, 再度 3.2 節の方法で, アクティブ系について解けば良い.

3.5 シミュレーションにおけるモデル化

アクティブ系は M/M 型のネットワーク型待ち行列

* $(m)_l = m(m-1)\dots(m-l+1), (m)_0 = 1$ とする.
 ** $d = 0 \cdot (1-p) + p(1-p) + 2p^2(1-p) + \dots$ であるから $d/(1-p) = p + 2p^2 + 3p^3 + \dots$ そこで両辺に p をかけ引き算を行って(1)式を得る.

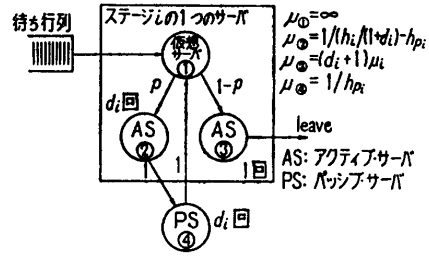


図5 パッシブ・サーバへのアクセスモデル
 Fig. 5 Access model to passive server.

で表わされることが仮定されており, アクティブ系のみからなるシステムは GPSS を用いて容易にシミュレーションプログラムを書くことができる. これに対してパッシブ系を含む系全体のシミュレーションを行うにはパッシブ系の動作機構として指数時間のジョブのアクティブ・サーバ保留時間内に平均 d_i 回で平均保留時間 h_{pi} のパッシブ・サーバ・アクセス要求を発生させる必要があり, GPSS でこの機構を表現するのは困難である.

そこで, ここではパッシブ系の動作を図5のモデルで近似してシミュレーションを行うものとする. 図5はパッシブ・サーバへのアクセスのあるアクティブ・サーバの動作時間(指数分布) h_i をパッシブ・サーバ平均アクセス回数 d_i 個に分割し, d_i 個の平均値 h_i/d_i の指数保留時間で近似することを表わしている. すなわち, 図5ではアクティブ・サーバが3つの指数サーバ①②③から合成されており, 待ち行列からディスパッチされたジョブはサーバ①を経た後確率 p でサーバ②を経由してパッシブ・サーバに移り①に戻る. サーバ①を終えたジョブは確率 $1-p$ でサーバ③を通してこのステージを抜けるものとする.

ここで, サーバ③の平均保留時間はサーバ②とパッシブ・サーバ④の平均保留時間の和に等しく, ステージ i の平均保留時間 h_i の $1/d_i$ 倍に設定されているものとする. するとパッシブサーバの平均保留時間は h_{pi} であるからサーバ②の平均保留時間は $h_i/d_i - h_{pi}$ である. またサーバ①からサーバ②への遷移確率 p_i はパッシブ・サーバの平均通過回数が d_i であることから,

$$p_i = d_i / (1 + d_i) \quad (1)$$

で与えられる**.

以上でパッシブ・サーバへのアクセス機構はシミュレーションできる. 上記の方法で1つの指数サーバの保留時間を複数の指数サーバに分割することによるサービス時間の分布のずれを付録に示す.

4. 事例検討

4.1 制御表アクセス競合モデル

(1) 例題の設定

バンキング・システムにおけるトランザクション処理をモデル化したリアルタイム処理モデルを用いて前述の解法の実用性を示す。使用するモデルを図6に示す。図6はマルチプロセッサをモデル化しており、9台のCPUから制御表（パッシブ・サーバ）へのアクセス競合のあることを示している。制御表はバンキング・システムの多くの制御表の中から、最も呼量の大きいものを1つ選んでいる。

1つのトランザクションに着目すると、トランザクションはCPUで処理される途中にDR, DK, MTにそれぞれ平均1.1, 2.0, 0.2回アクセスする。これらの装置にアクセスした後はCPUに割込みが行われ、CPUでは割込み処理後CPUの待ち行列につながり、割込み処理は微小であるとして無視し、直接CPUの待ち行列に入るとしている。

スループット・パスはスループットをカウントするためのパスであり、トランザクションがこのパスを通過するごとに1つのトランザクションが処理を終えて系外に去り、同時に1つの新しいトランザクションが到着する（ディスパッチされる）と仮定する。

(2) 近似モデルによる解法とシミュレーション結果の比較

3章の解析的方法で得た結果とシミュレーション結

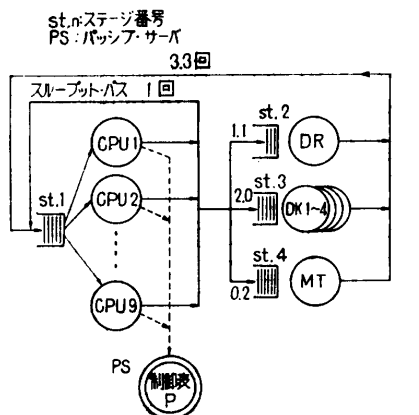


図6 リアルタイム・システムにおける制御表アクセスモデル
Fig. 6 Access model to control table in a real time system.

果を比較する。シミュレーションでは平衡状態に達するまでの約100トランザクションの結果を捨て、そのあとの1000トランザクションの平均値を求めた。解析的手法では3.2節で述べたようにアクティブ・サーバからパッシブ・サーバへのアクセス要求をランダム生起で近似しているに対してシミュレーションでは正しく間歇的なアクセス要求の発生モデルが用いられているが、両者の結果の比較からはランダム生起近似に起因する誤差の評価は困難であるので、両者の傾向と実用性を示すのにとどめることとする。

図7に(5)式から得られた制御表アクセス競合によるCPU処理時間の伸び Δh をシミュレーション結果と比べる。また図8,9にCPU保留時間が Δh だけ伸びたときのアクティブ系のスループット、レスポンスタイムをシミュレーション結果と比較する。

これらの結果をみると、 Δh では解析的方法とシミュレーション結果の間には若干の誤差があるものの、系全体の特性（スループット、レスポンスタイム）は数%以下の誤差となっている。これらの誤差の原因としては

- ① CPU 保留時間の分布に解析的方法とシミュレーションで差がある、

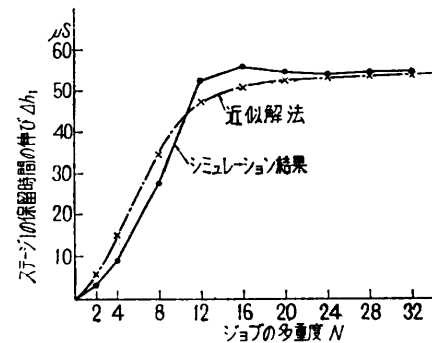


図7 ステージ1の保留時間の伸び Δh_i
Fig. 7 Waiting time of stage 1 to passive server.

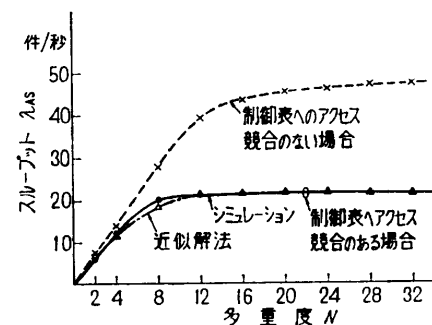


図8 多重度とスループットの関係
Fig. 8 Throughput characteristics.

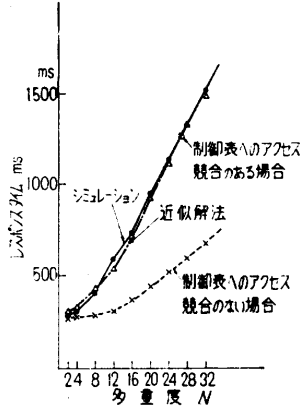


図 9 多重度とレスポンスタイムの関係
Fig. 9 Response time characteristics.

② 制御表アクセスが $M(r)/M/1$ モデルで近似されている,

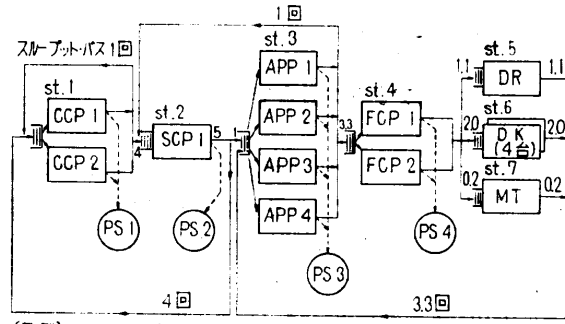
③ シミュレーション・プログラム中で設定される定数が 1% 程度の誤差を含むこと,

④ シミュレーション結果は必ずしも安定していない,

等があげられる。ここで、①から生じる誤差の程度は不明であるが、 ρ_1 が 1 に近づくにつれて誤差の減少する傾向からみて、②の原因が最も大きく効いているものと思われる。

②に起因する誤差を解消するにはパッシブ系の動作を 3.3 節で述べた $M(N)/M/1$ の代りにシミュレーションによって解析し、3.4 節の方法を用いればよいが、パッシブ系の動作を正しくシミュレーションプログラムで表現するのは困難であり、また実用上は本解析法で十分と考えられる。

なお、評価に使用した計算機時間は DIPS-1 を用いて解析的手法が約 10 秒であったのに対してシミュレ



(凡例) stn: ステージ番号, PSn: パッシブサーバの番号(ここではデディケートドメモリと考える)

図 10 デディケートド・メモリをもつ分散処理方式のモデル

Fig. 10 Distributed processor model with dedicated memories.

ーションは約 1600 秒であった。

4.2 分散処理におけるデディケートド・メモリへのアクセス競合モデル

(1) 例題の設定

図 10 に示すように 4.1 節の例題で CPU 上での処理を 4 種のプロセッサ (合計台数は等しく 9 台としている) に分割した分散処理方式を考え、これら 4 種のプロセッサはステージごとにデディケートド・メモリをもっていると、それぞれのデディケートド・メモリへのアクセス競合のある系全体の性能を求める。

図 10 で設定されるパラメータの値を表 1 に示す。

(2) 解析的方法とシミュレーション結果の比較

前と同様にしてデディケートド・メモリへのアクセス競合による各ステージのジョブ保留時間の伸び Δh_i を求め図 11 に示す。但し第 2 ステージからのデディケートド・メモリへのアクセスには競合はないので無視している。また、系全体のスループット、レスポンスタイムをそれぞれ図 12, 13 に示す。

表 1 各ステージのパラメータ設定値表
Table 1 Parameter value table.

ステージ番号	ステージ名	サービス・ルート ^{*1} (件/秒)	サーバ数 r_i	ステージ通過 回数/トラン	アクセスするパッシブ・サーバ名 ^{*2} (デディケートド・メモリ)	PS へのアクセス 回数/通過当り	PS のサービスレート ^{*3}	備 考
1	CCP	133	2	5 回	PS 1	10 回	2778	Communication Control Processor
2	SCP	236	1	5	PS 2	10	2778	Scheduling Processor
3	APP	50	4	4.3	PS 3	20	2778	Application Processor
4	FCP	80	2	3.3	PS 4	20	2778	File Control Processor
5	DR	100	1	1.1	—	—	—	
6	DK	26	4	2.0	—	—	—	
7	MT	56	1	0.2	—	—	—	

*1: プロセッサ性能として $g=9 \mu s/step$ を仮定し、各ステージで分担実行する DS を設定して定めた。

*2: 各ステージごとにデディケートド・メモリをもたせ、このメモリへのアクセス競合を考慮に入れる。

*3: PS への 1 アクセス当り 40 step 分の処理時間を保留するとし、保留時間 = $40 \text{ step} \times 9 \mu s$ を仮定した。

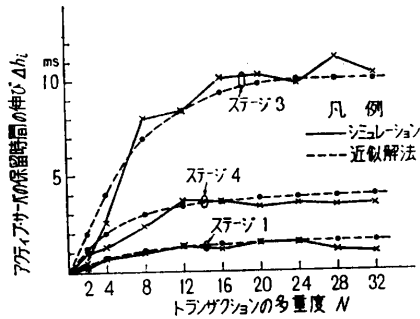


図 11 PS へのアクセス競合によって生じる AS の保留時間の伸び

Fig. 11 Waiting time of stages to passive servers.

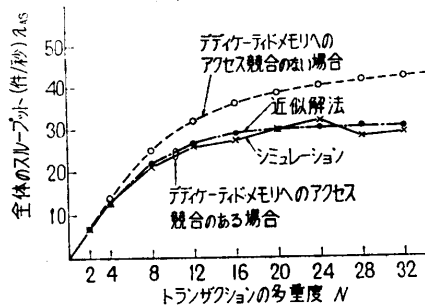


図 12 系全体のスループット

Fig. 12 Through put characteristics.

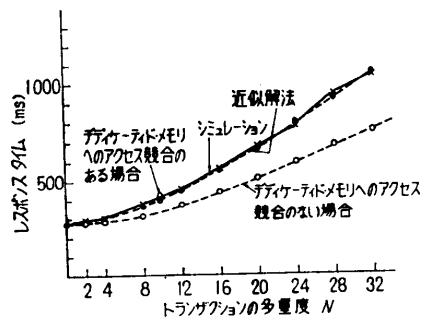


図 13 レスポンスタイムの比較

Fig. 13 Response time characteristics.

これらの結果を見ると、 Δh_i には若干の誤差が見られるが、系全体の特性の誤差は数%以下である。これらの原因として前の例と全く同様の原因が考えられるが、この例題ではシミュレーションとして600トランザクションの平均値を用いており、中でもシミュレーションの誤差が大きいと思われる。

なお、本例題の計算時間を比べると、解析的方法が約20秒に対してシミュレーションは7200秒のCPU時間を必要としている。

以上の例題による結果から、パッシブ・サーバを有

する計算機の性能評価において、本解析手法は従来のシミュレーション方法に比べて実用上十分有効であるということが出来る。

5. あとがき

本論文では計算機システムを構成するサーバを単独で働くことのできるアクティブ・サーバとアクティブ・サーバの制御下で働くパッシブ・サーバに分け、両者で構成される計算機モデルを3つのタイプに分類してその第1のタイプのモデルについてのスループット、レスポンスタイムの計算機による解析法を示した。

パッシブ・サーバへのアクセスのあるアクティブ・サーバでは現実の計算機ではアクティブ・サーバ上にジョブがあるときのみパッシブ・サーバへのアクセス要求が生起するため、アクセス要求の生起は間断的であるが、本解析モデルではこれをランダム生起で近似している。そのため、システムが低負荷状態でアクティブ・サーバの使用率の小さい場合は若干の誤差を生じるものの、システム設計で特に問題となる高負荷状態では近似誤差は小さい。この傾向はシミュレーション結果との比較からも明らかである。

本解析はプログラムパッケージCMAP-1 (Computer Model Analysis Program) の中に組み込んで使用しているが、例題でも示したようにハードウェア、ソフトウェアの混在したモデルを非常に簡単に評価できる点で便利であり、システム設計時において予想されるシステムの性能を容易に求めることができる。

本論文ではパッシブ・サーバとして制御表、プログラム、メモリ等の資源を考え、CPU、周辺装置などをアクティブ・サーバとしているが、逆にCPUをパッシブ・サーバとし、I/O装置をアクティブ・サーバと考えてI/O終了後のCPU割込み処理モデルを解析することなどの適用も可能である。

なお本解析法はパッシブ・サーバ系の動作が解析できれば本論文で示した第2、第3のモデルの解法へも容易に拡張し得る。また、ここではアクティブ・サーバ系をM/M型のネットワーク型待ち行列の範囲に限定しているが、これをBCMP*型のネットワーク型待ち行列まで拡張することも可能である。今後本解析法の適用領域の拡大を図り、計算機システムの評価に役立てて行く予定である。

謝辞 おわりに日頃御指導いただく当研究所新井データ通信方式研究室長、本検討を通じて御討論いただいた村岡調査役に深謝する。

* Baskett, Chandy, Muntz, Palacios によるネットワーク型待ち行列をいう。

参考文献

- 1) Coffman and Denning: Operating Systems Theory, Prentice-Hall, Inc. (1973).
- 2) Kleinrock: Queuing Systems, Vol. 1, 2, John Wiley & Sons (1975).
- 3) 橋田: 計算機システムにおける待ち行列, オペレーションズ・リサーチ, Vol. 22, No. 4, pp. 242-250 (1977).
- 4) Jackson, J.R.: Jobshop-Like Queuing Systems, Research Report 81, Management Science Research Project, UCLA (Jan. 1963).
- 5) Gordon, W.J. and Newell, G.F.: Closed Queuing Systems with Exponential Servers Opr. Res., Vol. 15, pp. 254-265 (1967).
- 6) Buzen, J.P.: Computational Algorithms for Closed Queuing Networks with Exponential Servers, Commun. ACM, Vol. 16, No. 9, pp. 527-531 (1973).
- 7) Baskett, F., Chandy, K. M., Muntz, R. R., and Palacios, F. G.: Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, J. ACM, Vol. 22, No. 2, pp. 248-260 (April 1975).
- 8) Reiser, M. and Kobayashi, H.: Queuing Networks with Multiple Closed Chains: Theory and Computational Algorithms, IBM J. Res. Dev., pp. 283-294 (May 1975).
- 9) Reiser, M.: Interactive Modeling of Computer Systems, IBM Syst. J., No. 4, pp. 309-327 (1976).
- 10) Sauer, C. H., Reiser, M., and MacNair, E. A.: RESQ-A package for solution of generalized queuing networks, NCC, pp. 977-986 (1977).
- 11) Omahen, K. J.: Capacity Bounds for Multi-resource Queues, J. ACM., Vol. 24, No. 4, pp. 646-663 (1977).
- 12) 紀一誠: 資源切り出し型待ち行列の解析, 情報処理, Vol. 19, No. 2, pp. 151-157 (1978).
- 13) Maekawa, M. and Boyd, D. L.: A Model of Concurrent Tasks Within Jobs of a Multiprogramming System, Proc. Eighth Annual Conference on Information Science and Systems, Princeton University, Princeton, March 28-29 (1974).
- 14) Maekawa M. and Boyd, D. L.: Two Models of Task Overlap within Jobs of Multiprocessing Multiprogramming Systems, Proc. International Conference on Parallel Processing, Wayne Steto University, Michigan (1976).
- 15) 池原: 待ち行列モデルによる機能分散型分散処理方式の性能評価, 情報処理, Vol. 18, No. 11, pp. 1102-1109 (1977).
- 16) 池原: ダイナミック・オーバレイ方式の性能に

ついて, 信学論誌D, Vol. J61-D, No. 5, pp. 307-314 (1978).

17) 雁部: 電話交換トラヒック, 電気通信協会, pp. 111-118 (昭 41).

付 録

保留時間がいずれも指数分布をとる2つのサーバを1つの指数サーバで近似したときの分布のずれを調べる。

保留時間の分布が $\delta_1(t) = \mu_1 e^{-\mu_1 t}$, $f_2(t) = \mu_2 e^{-\mu_2 t}$ で表わされる2つのサーバを直列につなぎ, 1つのサーバとみなすと, そのサーバの保留時間の分布 $f_3(t)$ は両者のコンボリューションをとり

$$f_3(t) = \begin{cases} \frac{\mu_2 \mu_1}{\mu_2 - \mu_1} (e^{-\mu_1 t} - e^{-\mu_2 t}) & (\mu_1 \neq \mu_2) \\ \mu^2 t e^{-\mu t} & (\mu_1 = \mu_2 = \mu) \end{cases}$$

これを

$$f_3^*(t) = \mu_3 e^{-\mu_3 t}$$

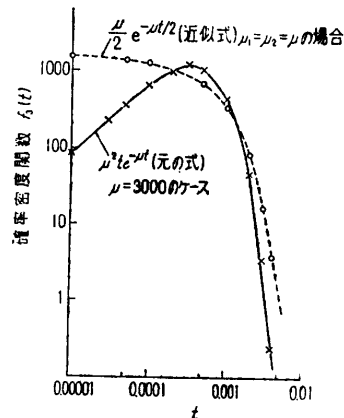
$$\mu_3 = \mu_1 \mu_2 / (\mu_1 + \mu_2)$$

で近似すると, f_3, f_3^* の分布の平均値は等しいが, 分散 $\bar{t}_3^2, \bar{t}_3^{*2}$ は

$$\bar{t}_3^2 = \frac{2(\mu_1^2 + \mu_1 \mu_2 + \mu_2^2)}{\mu_1^2 \mu_2^2}$$

$$\bar{t}_3^{*2} = \frac{(\mu_1 + \mu_2)^2}{\mu_1^2 \mu_2^2}$$

となり, 両者の比 $\bar{t}_3^2 / \bar{t}_3^{*2}$ は $1.5(\mu_1 / \mu_2 = 0 \text{ or } \infty)$ から $2(\mu_1 = \mu_2)$ の値をとり, $f_3(t)$ を $f_3^*(t)$ で近似すると分散が小さくなる。



付図1 確率密度関係の比較

A. Fig. 1 Comparison between two probability density functions.

なお付図1に $\mu_1 = \mu_2 = 3000$ の場合の f_1 と f_1^* を比べる。両者は μ の値に関係なく交る。

(昭和53年2月6日受付)
(昭和53年6月15日採録)