

FCFS スケジューリングを行うサーバを有する 複数クラス待ち行列網モデルの近似解析†

山 本 彰^{††} 西 垣 通^{††}

平均サービス時間の異なる複数クラスの処理要求に対して FCFS (First-Come-First-Served) スケジューリングを行うサーバを有する待ち行列網の近似解析手法を提案する。FCFS スケジューリングは計算機システムにおいて等優先度の処理要求間でしばしば用いられるが、待ち行列網モデルでこれを厳密に取り扱うことは困難であった。また、実用的でかつ精度の十分な近似手法も現在のところ提案されていない。本手法では、このサーバをクラス数に等しい pseudo サーバと呼ぶサーバ群に置き換えることにより、待ち行列網モデルを積形解をもつ形に変換する。解析に当たり、処理要求が置換えの対象となるサーバに到着した時点における各クラスの処理要求の平均滞在数が、近似モデルのその時点における各 pseudo サーバの平均滞在数に等しいことを仮定した。この仮定は、モデルが CPU バウンド、I/O バウンドのときには厳密に成立するため、近似誤差は負荷がバランスしている領域で問題となる。精度検証の結果、誤差の生じやすい負荷がバランスしている領域でも、近似誤差はたかだか 10% 程度であった。

1. はじめに

計算機システムの構成設計に際し、システム設計者は、各設計段階において、要求性能の実現性を確認する必要がある。性能予測手法には、シミュレーション手法と解析の手法があるが、設計の初期段階においては、精度よりもむしろ、予測作業に要するコスト低減が重要なため、しばしば後者が用いられる。

解析の手法のなかでは待ち行列網モデルが一般的である。Baskett¹⁾らは、待ち行列網モデルにおいて、すべてのサーバがある定められた条件を満たしたときに限り、その解が積形解となることを明らかにした。ここで、サーバとは待ち行列網モデルの構成要素を指す。また、ここでは、積形解が得られる条件を満足するサーバを BCMP 型 (Baskett, Chandy, Muntz, Palacios) サーバ、そうでないサーバを非 BCMP 型サーバとよぶ。

積形解をもたない待ち行列網モデルを厳密に解析することは、現在のところ実用的には不可能である。しかし、計算機システムを待ち行列網モデルとしてモデル化する場合、積形解が得られることは少ない。これは、I/O 装置が BCMP 型サーバとしてモデル化できるのに対し、CPU はしばしば非 BCMP 型サーバとなるためである。たとえば、等優先度の処理要求の

間でしばしば用いられる FCFS (First-Come-First-Served) スケジューリングの場合には、サービス時間分布が指数分布で、かつ、すべてのクラスのサービス時間が等しいとき以外は BCMP 型サーバとはならない¹⁾。ここで、クラスとは、評価単位となる処理要求の集合であり、たとえば、TSS コマンド、バッチ・ジョブがそれぞれ一つのクラスにまとめられる。通常、異クラス間では平均サービス時間は異なるため、FCFS スケジューリングの場合、CPU が BCMP 型サーバになるのはまれである。また、優先度スケジューリングの場合も、CPU は BCMP 型サーバとはならない¹⁾。以下、CPU、I/O 装置をモデル化したサーバをそれぞれ、CPU サーバ、I/O サーバとよぶ。

以上のような背景をもとに、積形解をもたない待ち行列網モデルを近似的に解析するための研究がさかに行われてきた^{2)~9)}。筆者らはすでに、モデル内のボトルネック資源に着目し、平均応答時間が有する漸近解を得る手法を提案している^{2)~4)}。しかし、漸近解の場合には確率的に生ずる待ちを評価できないため、明確なボトルネック資源が存在しないとき、十分な精度が得られないという問題があった。

一方、Sauer⁶⁾、Reiser⁷⁾、Sevcik⁸⁾らは、それぞれ待ち行列網モデルを基本にした近似手法を提案している。Sauer は、Norton の定理⁵⁾を利用した近似手法を提案した。しかし、この手法では、複数の I/O サーバ (BCMP 型サーバ) を縮退したサーバと CPU サーバ (非 BCMP 型サーバ) の解析は厳密に行うため、クラス数、多重度の増加に伴いメモリ量が大幅に増大

† Approximate Analysis of Multiple-Class Queueing Network Models Containing a Server with FCFS Scheduling Disciplines by AKIRA YAMAMOTO and TOHRU NISHIGAKI (Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所

するという問題があった。

Reiser⁷⁾, Sevcik⁸⁾らの近似手法は、2レベルの割込み型優先度スケジューリングを行っている CPU サーバを、それぞれのクラスの処理要求が専用にアクセスする二つのサーバに置き換えるというものである。この置換えにより、待ち行列網は積形解をもつようになる。これらの手法では、積形解をもつ待ち行列網を解析すればよいから、Sauer のモデルに比べて必要な計算量、メモリ量は少ない。Reiser のモデルでは異なるクラスの処理要求が同一の I/O サーバにアクセスすることを許していない。Sevcik のモデルはこの制限を取り除いたものである。Reiser, Sevcik のモデルでは、置き換えられたサーバにおいて、近似前のモデルで他クラスの処理要求の影響により生ずる待ち時間を反映させるために、平均サービス時間の調節を行っている。他クラスの処理要求の影響により生ずる待ち時間はスケジューリング方式により異なる。池原⁹⁾は、Sevcik らの考え方を2レベルの非割込み型スケジューリングに適用している。

しかし、FCFS スケジューリングを行っている CPU サーバに対してこの考え方を適用した近似手法は、現在のところ提案されていない。FCFS スケジューリングは等優先度の処理要求の間ではしばしば用いられるため、この問題を解決することは重要である。また、Reiser, Sevcik, 池原らのモデルでは、クラス数はいずれも2であった。本論文では、Sevcik らが提案した考え方を、CPU サーバで n クラスの FCFS スケジューリングを行っている待ち行列網モデルに適用する。ここで、平均サービス時間はクラスごとに異なっていてよく、その分布形は指数分布であるとする。本論文では、置き換えられたサーバ群を pseudo サーバ群、個々のサーバを pseudo サーバと呼ぶ。Sevcik は、優先順位の低いクラス用のサーバを、高いクラス用のサーバ（呼称なし）に対して shadow CPU⁸⁾と呼んだ。しかし、ここでは、非 BCMP 型サーバを複数の BCMP 型サーバに置き換えているのであるから、CPU サーバに対して、置き換えられたサーバ群を pseudo サーバ群と呼ぶ方が妥当であると考えられる。本近似手法の特徴は、あるクラスの処理要求が CPU サーバへ到着した時点における各クラスの処理要求の平均滞在数は、近似したモデルのその時点における各 pseudo サーバの平均滞在数に等しいという仮定を設けたことである。この仮定は、モデルが、CPU バウンド、または、I/O バウンドになっているときに

は厳密に成立する。

2章では、近似手法の基本的な考え方を説明し、3章では、各 pseudo サーバの平均サービス時間を求める。4章では、数値計算により得た厳密解、および、シミュレーションにより得た解を用いて、本手法の精度検証を行う。

2. 近似手法の概要

本論文では、クローズド待ち行列網モデルを取り扱うが、これを図1に示す。この待ち行列網モデルは、一つの CPU サーバと複数の I/O サーバにより構成される。これは、現在の計算機システムの多くが、1台の CPU と複数の I/O 装置により構成されるためである。

図1に示したモデルにおいて、I/O サーバを各クラスの平均サービス時間が等しくその分布形が指数分布であり、FCFS スケジューリングを行っているサーバ、すなわち、BCMP 型サーバとしてモデル化することは、実際の I/O 装置の使用形態から考えて大きな問題とならない。したがって、CPU サーバをいくつかの BCMP 型サーバに置き換えることにより、すべてのサーバが BCMP 型サーバとなるため、待ち行列網モデルを積形解をもつ形に変換することができる。

図2にこの置換えを示す。pseudo サーバの個数はクラス数に等しく、それぞれのクラスの処理要求は特定の一つの pseudo サーバのみをアクセスする。クラス i の処理要求がアクセスする pseudo サーバを pseudo サーバ i と呼ぶ。各 pseudo サーバは、BCMP 型サーバである必要があるため、処理要求のサービス時間分布は指数分布で、FCFS スケジューリングが行われると仮定する。

以上の変換により、近似後のモデルは積形解をもつようになる。しかし、近似による誤差を少なくするためには、近似前のモデルにおいて他クラスの処理要求

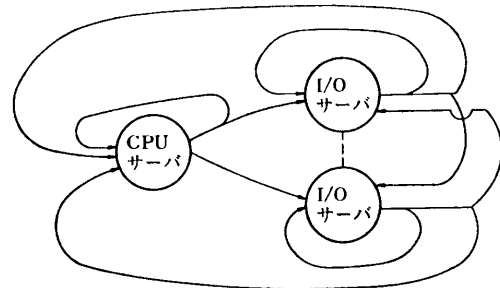


図1 本論文における待ち行列網モデル
Fig. 1 The queueing network model

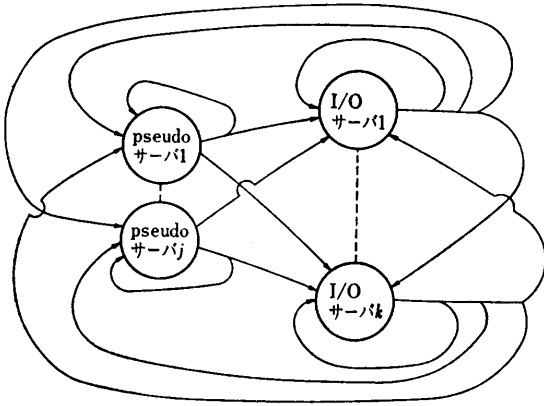


図 2 近似モデル

Fig. 2 The approximated model.

により生ずる待ちを反映させるため、各 pseudo サーバの平均サービス時間を適切に決めなければならない。したがって、pseudo サーバの平均サービス時間の決め方が本近似手法の鍵となる。pseudo サーバの平均サービス時間が定まると、近似後のモデルは、待ち行列網モデルの積形解を求める手続きを用いることにより求解が可能となる。次章では、各 pseudo サーバの平均サービス時間を求める。

3. pseudo サーバの平均サービス時間の導出

本章では、FCFS スケジューリングにおける pseudo サーバの平均サービス時間を求める。表 1 に、本論文で用いる記号をまとめる。また、クラスの集合を I 、I/O サーバの集合を K とする。

本近似手法では、I/O サーバに関してはまったく変換を行っていないため、すべてのクラスの処理要求に関して、アクセス当りの CPU サーバにおける滞在時間の分布を近似後のモデルで保存することができれば、近似誤差は 0 となる。しかし、前章で述べたように、近似後のモデルが積形解をもつためには、pseudo サーバのサービス時間分布を指数分布としなければならない。したがって、滞在時間の分布形を保存することは不可能である。ここでは、滞在時間の平均値のみに着目し、次の仮定を設ける。この仮定が成立したときには、近似後のモデルと近似前のモデルで各クラスの処理要求のアクセス当りの CPU サーバの平均滞在時間が等しくなる。

【仮定 1】 近似前のモデルにおいて、クラス i の処理要求が CPU サーバに到着した時点における各クラスの処理要求の平均滞在数は、近似後のモデルにおいて、クラス i の処理要求が pseudo サーバ i に到着し

表 1 記号の定義

Table 1 The definition of the symbols.

$S_{i,i}$	クラス i の処理要求の CPU サーバのアクセス当りの平均サービス時間
S_k	I/O サーバ k のアクセス当りの平均サービス時間
$T_{i,i}$	クラス i の処理要求の CPU サーバのアクセス当りの平均滞在時間
$S_{0,i}^*$	pseudo サーバ i のアクセス当りの平均サービス時間
$T_{0,i}^*$	pseudo サーバ i のアクセス当りの平均滞在時間
$P_{i,k,l}$	クラス i の処理要求がサーバ k からサーバ l へ遷移する確率。ただし、 k, l がそれぞれ 0 のときは、CPU サーバを意味し、0 以外のときには、I/O サーバを意味する。
N_i	クラス i の処理要求の多重度
$N_{0,i,j}$	クラス i の処理要求が CPU サーバに到着した時点におけるクラス j の処理要求の CPU サーバの平均滞在数
$N_{0,i,j}^*$	クラス i の処理要求が pseudo サーバ i に到着した時点における pseudo サーバ j の平均滞在数

たときの各 pseudo サーバの平均滞在数に等しい。

仮定 1 は次式の成立を意味する。

$$N_{0,i,j} = N_{0,i,j}^* \quad (3.1)$$

仮定 1 を設けた根拠は、待ち行列網が極端な CPU バウンド、I/O バウンドになったときには、 $N_{0,i,j}$ と $N_{0,i,j}^*$ が同じ値に収束するためである。

待ち行列網モデルが I/O バウンド、すなわち、ある特定の I/O サーバの平均サービス時間が無限大になるとき、 $N_{0,i,j}$ 、 $N_{0,i,j}^*$ は次式を満たす。

$$\lim_{S_k \rightarrow \infty} N_{0,i,j} = 0 \quad (\forall i, \forall j \in I, \exists k \in K) \quad (3.2)$$

$$\lim_{S_k \rightarrow \infty} N_{0,i,j}^* = 0 \quad (\forall i, \forall j \in I, \exists k \in K) \quad (3.3)$$

一方、CPU バウンド、すなわち、すべての I/O サーバの平均サービス時間が 0 に収束すると、 $N_{0,i,j}$ 、 $N_{0,i,j}^*$ は以下に示す値に収束する。

$$\lim_{(S_1, \dots, S_k) \rightarrow (0, \dots, 0)} N_{0,i,j} = N_j \quad (\forall i \in I, \forall j \in I - \{i\}) \quad (3.4)$$

$$\lim_{(S_1, \dots, S_k) \rightarrow (0, \dots, 0)} N_{0,i,i} = N_i - 1 \quad (\forall i \in I) \quad (3.5)$$

$$\lim_{(S_1, \dots, S_k) \rightarrow (0, \dots, 0)} N_{0,i,j}^* = N_j \quad (\forall i \in I, \forall j \in I - \{i\}) \quad (3.6)$$

$$\lim_{(S_1, \dots, S_k) \rightarrow (0, \dots, 0)} N_{0,i,i}^* = N_i - 1 \quad (\forall i \in I) \quad (3.7)$$

仮定 1 を設けたことにより、近似による誤差は、両者に対する負荷がバランスしているときに大きくなると考えられる。

以下、pseudo サーバの平均サービス時間を導く。

FCFS スケジューリングをとるサーバにおける平均滞在時間は、処理要求がサーバに到着した時点における平均残余仕事量¹¹⁾ (ある処理要求が到着したときに、すでにそのサーバに到着している処理要求を終了させるための平均所要時間) に自分自身の平均サービス時間を加えたものに等しい。近似前のモデルにおいて、各処理要求の CPU サーバにおけるサービス時間分布は指数分布であるため、現在サービス中の処理要求の平均残余時間¹¹⁾ (その処理要求を終了させるまでの平均所要時間) は平均サービス時間に等しい。したがって、各クラスの処理要求の CPU サーバのアクセス当りの平均滞在時間は次式を満たす。

$$T_{0i} = S_{0i} + \sum_j N_{0ij} S_{0ij} \quad (\forall i, \forall j \in I) \quad (3.8)$$

同様の理由で、近似後のモデルにおいて、各 pseudo サーバにおけるアクセス当りの平均滞在時間は次式を満たす。

$$T_{0i}^* = S_{0i}^* + N_{0i,i}^* S_{0i,i}^* \quad (\forall i \in I) \quad (3.9)$$

(3.8), (3.9) 式より、 T_{0i} と T_{0i}^* が等しいとき、 S_{0i}^* は次式を満たす。

$$S_{0i}^* = (S_{0i} + \sum_j N_{0ij} S_{0ij}) / (1 + N_{0i,i}^*) \quad (\forall i, \forall j \in I) \quad (3.10)$$

(3.10) 式に仮定 1, すなわち, (3.1) 式を適用すると次式が得られる。

$$S_{0i}^* = S_{0i} + (\sum_j N_{0ij} S_{0ij}) / (1 + N_{0i,i}^*) \quad (\forall i \in I, \forall j \in I - \{i\}) \quad (3.11)$$

(3.11) 式においては、 $N_{0i,i}^*$ ($N_{0i,i}^*$ は $N_{0i,i}^*$ に含まれる) が未知数である。以下、 $N_{0i,i}^*$ を求める。

近似後のモデルは、積形解をもつ。積形解をもつ待ち行列網モデルにおいては、処理要求がサーバに到着した時点における待ち行列網状態の確率分布に関しては次の定理が成立する。FCFS スケジューリングの場合の待ち行列網状態とは、各サーバに各クラスの処理要求がどのような順番で並んでいるかを示すものである。

【定理 1】 クラス i の処理要求がある特定のサーバに到着した時点における待ち行列網状態の確率分布は、クラス i の処理要求を 1 減じた (全時間に関する) 待ち行列網状態の確率分布に等しい¹⁰⁾。

したがって、 $N_{0i,i}^*$ はクラス i の処理要求を 1 減じたモデルの積形解を得、これより pseudo サーバ j の平均滞在数を算出することにより得られる。

待ち行列網の積形解を得るのに必要な情報は、各処理要求の各サーバに対する平均サービス時間、サーバ間の遷移確率、多重度である。本論文では、 S_{0i}^* , S_{0i} ,

$P_{i,i}$, N_i がこれらに相当する。これらの要素をそれぞれ、クラス、サーバに関してベクトル化したものを、 S_0^* , S , P , N とする。これらの情報を入力として、積形解を得、pseudo サーバ j の平均滞在数を得る関数を f_j , また N からクラス i の処理要求を 1 減じたベクトルを N_i とすると、定理 1 より次式が成立する。

$$N_{0i,j}^* = f_j(S_0^*, S, P, N_i) \quad (\forall i, \forall j \in I) \quad (3.12)$$

(3.12) 式を (3.10) 式に代入すると次式が得られる。

$$S_{0i}^* = S_{0i} + (\sum_j f_j(S_0^*, S, P, N_i) \cdot S_{0ij}) / (1 + f_i(S_0^*, S, P, N_i)) \quad (\forall i \in I, \forall j \in I - \{i\}) \quad (3.13)$$

ただし, (3.13) 式は右辺に、 S_0^* を含むため、これよりただちに、 S_{0i}^* を得ることはできない。ここでは、 S_{0i}^* の上限値と下限値を求め、 S_{0i}^* が取りうる値の範囲を得、次に 2 分探索法により (3.13) 式をみたす S_{0i}^* のセット、すなわち、 S_0^* を求める手法をとる。

まず、 S_{0i}^* が取りうる上限値と下限値を示す。 S_{0i}^* の下限値は、クラス i を除いたすべてのクラスの f_j が 0 のときである。一方、上限値は明らかに、 $f_i = 0$, $f_j = N_j$ ($\forall j \in I$, ただし、 $j \neq i$) が成立したときである。これにより、上限値、下限値は次式を満たす。

$$\text{MAX}(S_{0i}^*) = S_{0i} + \sum_j N_j S_{0ij} \quad (\forall i \in I, \forall j \in I - \{i\}) \quad (3.14)$$

$$\text{MIN}(S_{0i}^*) = S_{0i} \quad (3.15)$$

以上により、すべてのクラスの S_{0i}^* の上限値と下限値が定まるため、2 分探索法により、 S_0^* を得ることができる。ただし、2 分探索法で得られる解は厳密解でなく数値解であるため、精度を上げようとするとき計算量が增大する。

S_0^* , すなわち、すべてのクラスの pseudo サーバの平均サービス時間が求まると、積形解をもつ待ち行列網を解析する手続きとまったく同様の手続きで、近似モデルの解析が可能となる。次章では、以上述べてきた手法に従って得た近似解の精度検証を行う。

4. 近似手法の精度検証

本章では、数値計算により得た厳密解、および、シミュレーションによって得た解と近似解を比較することにより、近似手法の精度検証を行う。サーバ数、処理要求数が少ない場合については、厳密解との比較を行う。しかし、実際の計算機システム構成では、サーバ数、処理要求数が多くなるため厳密解が得られない。この場合については、シミュレーションにより得た解との比較を行う。

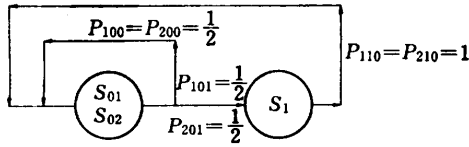


図3 厳密解との精度検証に用いたモデル

Fig. 3 The model for validation in comparison with the exact solutions.

まず、数値計算により得た厳密解との比較を行う。図3に、精度検証に用いたモデルを示す。クラス数は2で、処理要求の多重度が、それぞれ2, 3の場合について精度検証を行った。それぞれのクラスのアクセス当りのCPU サービス時間比: $C (S_{02}/S_{01})$ は1~10までを適当に選択した。また、CPU バウンド, I/O バウンドを表す指標 B を $(2/(S_1(P_{101} + P_{201}))) / (1/S_{01} + 1/S_{02})$ として、 B が $1/100 \sim 100$ の範囲についての評価を行った。 B は、それぞれのクラスの処理要求のCPUの完了率の和とI/Oサーバの完了率にアクセス頻度を重みづけしたものの和の比である。 B が1より十分小さいとき、I/O バウンド、1より十分大きいとき、CPU バウンド、1付近 ($S_{01} = S_{02}$ のときには、 $B=1$ のときが、最も負荷がバランスしているが、 $S_{01} \neq S_{02}$ のときにはこれが成立しない) のとき、負荷がバランスしていることになる。

図4~7に評価結果をまとめる。評価項目はそれぞれのクラスの平均応答時間である。図で、縦軸は平均応答時間の相対誤差で、横軸は B (対数目盛り) である。 C が、1, 2, 5, 10 の場合をそれぞれ1本のグラフにまとめた。図4はそれぞれの多重度が2の場合のクラス1 (CPU のサービス時間小) の評価結果であり、図5は同様の場合のクラス2の評価結果である。図6, 7は多重度が3の場合のそれぞれのクラスの評価結果である。

評価結果を見ると、前章で述べたように、負荷がバランスしているときには誤差が大きく、CPU バウンド, I/O バウンドのときにはほとんど誤差がないことがわかる。それ以外には、多重度が増すと誤差が小さくなること、 C が大きくなると誤差が大きくなること、各クラスごとに結果を見ると、クラス1は、負荷がバランスしている点からI/O バウンドによる領域では、近似解は厳密解に比べて過小評価となる。一方、負荷がバランスしている領域からCPU バウンドによる領域では過大評価となる。クラス2の場合には、負荷がバランスしている領域からI/O バウンドによる領域にかけては、過大評価となるが、バ

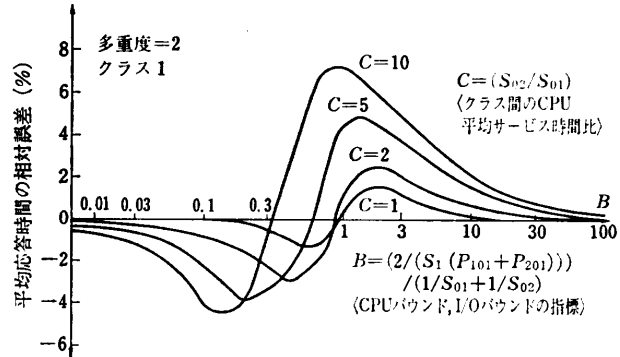


図4 厳密解による評価結果(1)

Fig. 4 Validation in comparison with the exact solutions (1).

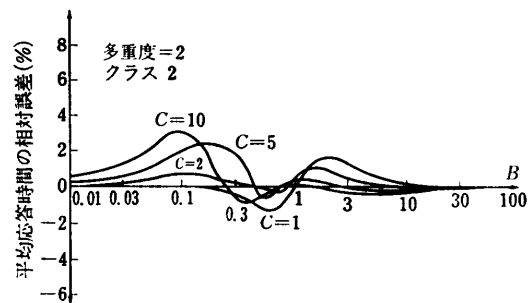


図5 厳密解による評価結果(2)

Fig. 5 Validation in comparison with the exact solutions (2).

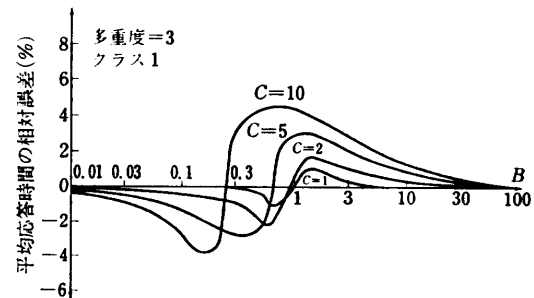


図6 厳密解による評価結果(3)

Fig. 6 Validation in comparison with the exact solutions (3).

ランスしている領域からCPU バウンド領域にかけては、誤差は小さい。以上が、おもな結果であるが、両クラスのCPU サービス時間の比 (C) が10以下であれば、誤差は最も大きくなるポイントにおいても7%程度である。これは、計算機システムの初期設計段階においてその大局的挙動を予測することを目的とした場合には、本手法が十分に有用であることを示している。

サーバ数、処理要求の多重度が多く厳密解が得られない範囲について、シミュレーションにより得た解と

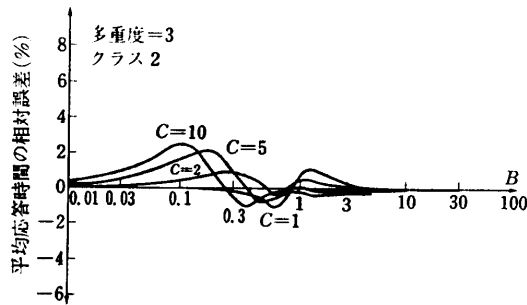


図 7 厳密解による評価結果 (4)

Fig. 7 Validation in comparison with the exact solutions (4).

の比較を行う。図 8 に、シミュレーションに用いた待ち行列網モデルを示す。モデルは、CPU サーバと I/O サーバ 1~8 により構成される。この構成に対し 2 クラスの処理要求の多重度をそれぞれ 30 まで増加させた場合の評価を行った。各クラスの CPU ダイナミック・ステップ数、I/O 発行回数は実ジョブをベースに作成した。CPU サービス時間は、そのジョブが HITAC M-200 H 上を走行した場合を仮定して算出した。サーバ間の遷移確率はそのジョブの I/O 発行回数に基づいている。I/O サーバ 1 の平均サービス時間は H 8589-11 型ディスク¹²⁾を使用した場合の平均アクセス時間である。また、I/O サーバ 2~8 は、H 8595 型¹²⁾ディスクの 1/3 シリンダを使用した場合の平均アクセス時間である (各サーバごとにアクセス時間が若干異なるのは、データ・ブロック長により転送時間が異なるためである)。この評価ケースにおいて、 B 、 C を計算すると、 $B=0.8$ 、 $C=6$ となる (B は、他の I/O サーバに比べ負荷量が大きくなる I/O サーバ 1 のみを選んで算出した)。 $B=0.8$ 、 $C=6$ 付近の厳密解の評価結果より考えると、このモデルは負荷がバランスして

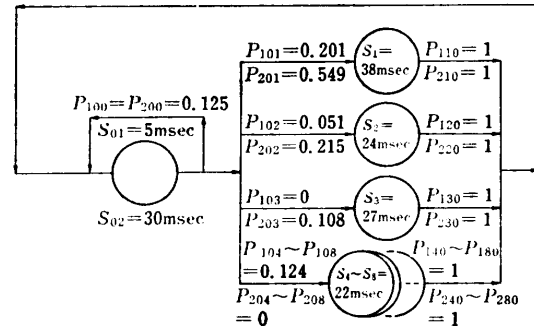


図 8 シミュレーションによる精度検証に用いたモデル

Fig. 8 The model for validation in comparison with the simulation results.

おり、近似誤差が生じやすいケースに相当していることがわかる。

表 2 に評価結果をまとめる。近似解は、シミュレーション解に比べ、クラス 1 の平均応答時間を過大評価し、クラス 2 の平均応答時間を過小評価している。ただし、クラス 2 の誤差はたかだか 2% ときわめて小さい。この結果は、 $B=0.8$ 、 $C=6$ 付近の厳密解との精度検証結果における傾向と同様である。これは、厳密解との比較で得られた傾向が、多重度、サーバ数が増加しても保持されることを意味している。また多重度が 15 以上になると、しだいに誤差が小さくなっている。これについても、厳密解との精度検証の結果得られた傾向とよく一致している。しかし、近似誤差の大きいクラス 1 の場合でも、その誤差は 10~12% 程度である。以上により、クラス間の CPU のサービス時間比が 10 程度以下であれば、本手法は十分有用であるということを結論づけることができる。

表 2 シミュレーションによる評価結果

Table 2 Validation in comparison with the simulation results.

多重度		シミュレーション結果*		近似解*		相対誤差	
クラス 1	クラス 2	クラス 1	クラス 2	クラス 1	クラス 2	クラス 1	クラス 2
3	3	696	1,028	766	1,005	+10.06%	-2.24%
6	6	1,278	1,702	1,431	1,794	+11.97%	-3.13%
9	9	1,919	2,677	2,150	2,595	+12.04%	-3.06%
12	12	2,605	3,493	2,911	3,402	+11.74%	-2.61%
15	15	3,338	4,307	3,699	4,213	+10.81%	-2.18%
18	18	4,123	5,123	4,504	5,029	+ 9.24%	-1.83%
21	21	4,884	5,944	5,318	5,847	+ 8.89%	-1.63%
25	25	6,017	7,011	6,411	6,690	+ 6.55%	-1.01%
30	30	7,429	8,352	7,782	8,309	+ 4.75%	-0.51%

* 評価項目：平均応答時間，単位 (msec)

5. おわりに

クラスごとに平均サービス時間が異なる処理要求に対して FCFS スケジューリングを行うサーバを有する待ち行列網モデルを解析するための近似手法を提案した。本近似手法は、このサーバをクラス数に等しい pseudo サーバと呼ばれるサーバ群に置き換えることにより、待ち行列網モデルを積形解が得られる形に変換するものである。解析に当たって、近似前のモデルにおいてあるクラスの処理要求が置換えの対象となるサーバに到着した時点における各クラスの処理要求の滞在数が、近似後のモデルにおいて、その処理要求が pseudo サーバに到着した時点における各 pseudo サーバの処理要求の滞在数と等しいということを仮定している。この仮定は、モデルが I/O バウンド、および、CPU バウンドのときには厳密に成立する。したがって、本近似手法は、負荷がバランスしている領域は近似による誤差が生じやすいが、CPU バウンド、I/O バウンドの場合には誤差は小さいことになる。

数値計算で得た厳密解とシミュレーションによって得た解によって本手法の精度検証を行った。この結果、誤差が生じやすい負荷がバランスしている領域でも近似誤差は 10% 程度であったことから、システム設計の初期段階において用いる手法としては、十分有用であるという結論を得た。

今後の課題としては以下の 2 点について検討中である。

(1) 他の積形解をもたないスケジューリング（たとえば、ゼネラル・プロセッサシェアリング）の近似手法の検討。

(2) (3.13) 式を満たす解をたんに 2 分探索法で探した場合には、クラス数が 2~3 であれば問題ないが、さらにクラス数が増加すると必要な計算量が急激に増加する。このため、高速なアルゴリズムの検討。

謝辞 終わりに、本研究についてご指導いただいた電気通信大学亀田壽夫助教授、ならびに本研究の機会を与えてくださった当社システム開発研究所川崎淳所長、有益な助言を与えてくださった同研究所大町一彦主任研究員、北嶋弘行主任研究員、本山博司研究員、

木下俊之研究員に深く感謝します。

参 考 文 献

- 1) Baskett, F. et al.: Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, *J. ACM*, Vol. 22, No. 2, pp. 248-260 (1975).
- 2) Nishigaki, T. et al.: An Approach to the GRM Performance Analysis by Asymptotic Approximation, *JIP*, Vol. 3, No. 2, pp. 59-67 (1980).
- 3) 西垣, 山本: 資源割当て優先度のある多重プログラミングシステムのボトルネック解析, 情報処理学会論文誌, Vol. 23, No. 5, pp. 562-569 (1982).
- 4) 山本, 西垣: サービス関数による応答時間制御の下での計算機システム性能のボトルネック解析, 情報処理学会論文誌, Vol. 24, No. 5, pp. 630-637 (1983).
- 5) Chandy, K. M. et al.: Parametric Analysis of Queueing Networks, *IBM J. Res. Dev.*, Vol. 19, No. 1, pp. 36-42 (1975).
- 6) Sauer, C. H. et al.: Approximate Analysis of Central Server Models, *IBM J. Res. Dev.*, Vol. 19, No. 3, pp. 301-313 (1975).
- 7) Reiser, M. et al.: Interactive Modeling of Computer Systems, *IBM Syst. J.*, Vol. 15, No. 4, pp. 309-327 (1976).
- 8) Sevcik, K. C.: Priority Scheduling Disciplines in Queueing Network Models of Computer Systems, *Proc. IFIP 77*, pp. 565-570 (1977).
- 9) 池原他: 非割り込み優先処理のある網型待ち行列の近似解析, 情報処理学会「計算機システムの解析と制御」研究会資料 12-1, pp. 1-10 (1981).
- 10) Sevcik, K. C. et al.: The Distribution of Queueing Network States at Input and Output Instants, *J. ACM*, Vol. 28, No. 2, pp. 358-371 (1981).
- 11) Kleinrock, L.: *Queueing Systems*, Vol. 1, John Wiley & Sons, New York (1975).
- 12) 日立マニュアル, H-8536-1 ディスク制御装置; H-8595 ディスク駆動装置; H-8585-1 ディスク接続装置; H-8589 ディスク駆動装置; 8080-2-022.

(昭和 58 年 8 月 26 日受付)

(昭和 58 年 11 月 15 日採録)