ＦＣＦＳスケジューリング型サーバを持つ待ち行列網モデルの近似解析手法

山本 彰，西垣 通
（日立製作所システム開発研究所）

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的手法があるが，設計の初期段階においては，精度よりもむしろ，多くの代替案の評価，及び，予測作業のコスト低減化などが重要なため，しばしば後者が用いられる。

解析的手法の中では，待ち行列網モデルが一般的である。Baskett[1]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時に限り，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，Palacious）サーバ，そうでないサーバを非ＢＣＭＰ型サーバとよぶ。

積形解を持たない待ち行列網モデルを厳密に解析する場合には，直接，状態方程式をたてて解を得なければならないが，一般の場合状態数が非常に多くなるため，厳密解を得るのは実用的には不可能である。しかし，現行の計算機システムを待ち行列網モデルとしてモデル化した場合，モデルが積形解を持つことは少ない。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。例えば，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時以外はＢＣＭＰ型サーバとはならない。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，例えば，ＴＳＳコマンド，バッチ・ジョブなどがそれぞれ１つのクラスにまとめられる。通常，異クラス間ではＣＰＵの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型サーバとなるのは希である。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。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サーバと呼ぶ。

以上の様な背景をもとに，積形解を持たない待ち行列網モデルを近似的に解析する研究がさかんに行なわれてきた。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[2),4)]を得る手法を提案している。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時，充分な精度が得られないという問題があった。

一方，Sauer[6)]，Reiser[7)]，Sevcik[8)]らは，それぞれ待ち行列網モデルを基本にした手法を提案している。Sauer[6)]は，Norton[5)]の定理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）の解析は厳密に行なうため，クラス数，多重度（処理要求の数）の増加に伴ない必要なメモリ量が大幅に増大する。

Reiser[7)]，Sevcik[8)]らの近似手法は，２レベルの割り込み型優先度スケジューリングを行なっている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２つのサーバに置き換えるというものである。この置き換えにより、待ち行列網は積形解を持つようになる。これらの手法では、積形解を持つ待ち行列網を解析すればよいため、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。池原[9]は、Sevcik らの考え方を２レベルの非割り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なっているCPUサーバに対してこの考え方を適用した近似手法は，現在の所提案されていない。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、Sevcik らのモデルでは、クラス数はいずれも２であったが、ここではCPUでｎクラスのFCFSスケジューリングを行なっている待ち行列網モデルにこの考え方を適用する。ここで、平均サービス時間はクラスごとに異っていてよく，その分布形は指数分布であるとする。本近似手法では、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サーバ（呼称なし）に対しshadow CPU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が、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点である。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時には厳密に成立する。

第２章では、近似手法の基本的な考え方を説明し、第３章では、各pseudoサーバの平均サービス時間を求める。第４章では、数値計算により得た厳密解により本手法の精度検証を行なう。

２．近似手法の概要

本近似手法では、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、１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。

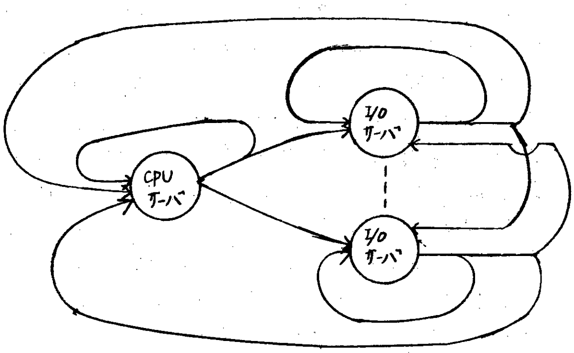

図１．本論文における待ち行列網モデル

図１に示したモデルにおいて，I/Oサーバを，実際のI/O装置の使用形態から考えて，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，FCFSスケジューリングを行なっているサーバ，すなわち，BCMP型サーバとしてモデル化する[5)~8)]ことは、しばしば行なわれることである。

従って，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持つ形に変換する[7,8]ことができる。

図2にこの置き換えを示す。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そのクラス専用の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なわれると仮定する。

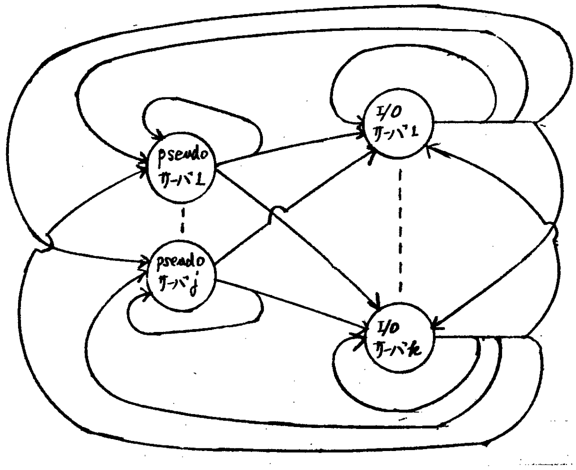

図2. 近似モデル

以上の変換により、近似後のモデルは積形解を持つようになる。しかし、各pseudoサーバには、特定のクラスの処理要求しか到着しないため、近似前のモデルにおいて他クラスの処理要求により生ずる待ちを評価できない。この待ち時間を反映させるため、pseudoサーバの平均サービス時間の調節を行なう。他クラスの処理要求の影響により生ずる待ち時間は、そのサーバのスケジューリング方式により異なる。（Sevcikらが提案した近似手法をそのままFCFSスケジューリングの場合に適用できないのはこのためである。）従って、近似による誤差を少なくするためには、FCFSスケジューリング方式ということを充分考慮して、pseudoサーバの平均サービス時間を決めなければならない。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。

表1. 記号の定義

S_{0i}	クラスiの処理要求のCPUサーバのアクセス当たりの平均サービス時間
S_k	I/Oサーバkのアクセス当たりの平均サービス時間
T_{0i}	クラスiの処理要求のCPUサーバのアクセス当たりの平均滞在時間
S^*_{0i}	pseudoサーバiのアクセス当たりの平均サービス時間
T^*_{0i}	平均滞在時間
P_{ikl}	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の時は，CPUサーバを意味し，0以外の時には，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_{0ij}	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^*_{0ij}	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

3．pseudoサーバの平均サービス時間の算出

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なっていないため，すべてのクラスの処理要求に関して，アクセス当たりの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持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持つためには、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。従って，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時には、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たりの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に到着した時の各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。

$$N_{oij} = N^*_{oij} \qquad (\forall i, \forall j \in I) \qquad (3.1)$$

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バウンドになった時には、N_{oij}とN^*_{oij}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時，N_{oij}，N^*_{oij}は次式を満たす。

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

$$\lim_{S_k \to \infty} N^*_{oij} = 0 \qquad (\forall i, \forall j \in I, \exists k \in K) \qquad (3.3)$$

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_{oij}，N^*_{oij}は以下に示す値に収束する。

$$\lim_{(S_1,\cdots,S_K) \to (0,\cdots,0)} N_{oij} = N_j \quad (\forall i, \forall j, \text{ただし}, i \neq j) \qquad (3.4)$$

$$\lim_{(S_1,\cdots,S_K) \to (0,\cdots,0)} N_{oii} = N_i - 1 \quad (\forall i \in I) \qquad (3.5)$$

$$\lim_{(S_1,\cdots,S_K) \to (0,\cdots,0)} N^*_{oij} = N_j \quad (\forall i, \forall j, \text{ただし}, i \neq j) \qquad (3.6)$$

$$\lim_{(S_1,\cdots,S_K) \to (0,\cdots,0)} N^*_{oii} = N_i - 1 \quad (\forall i \in I) \qquad (3.7)$$

仮定1を設けたことにより、近似による誤差は、両者に対する負荷バランスしている時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除仕事量[11]（ある処理要求が到着した時に、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。従って、各クラスの処理要求のCPUサーバのアクセス当たりの平均滞在時間は次式を満たす。

$$T_{oi} = S_{oi} + \sum_j N_{oij} S_{oj} \qquad (\forall i, \forall j \in I) \qquad (3.8)$$

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たりの平均滞在時間は次式を満たす。

$$T^*_{oi} = S^*_{oi} + N^*_{oii} S^*_{oi} \qquad (\forall i \in I) \qquad (3.9)$$

(3.8)，(3.9)式より、T_{oi}とT^*_{oi}が等しい時，S^*_{oi}は次式を満たす。

$$S^*_{oi} = \left(S_{oi} + \sum_j N_{oij} S_{oj}\right) / \left(1 + N^*_{oii}\right) \qquad (\forall i, \forall j \in I) \qquad (3.10)$$

(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。

$$S^*_{oi} = S_{oi} + \left(\sum_j N^*_{oij}\, S_{oj}\right) / \left(1 + N^*_{oii}\right) \qquad (\forall i \in I,\ \forall j \in I-\{i\}) \qquad (3.11)$$

(3.11)式においては、N^*_{oij}（N^*_{oii}はN^*_{oij}に含まれる。）が未知数である。以下，N^*_{oij}を求める。

近似後のモデルは、積形解を持つ。積形解を持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。

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。

従って，N^*_{oij}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。ここでは、S^*_{oi}，S_{ik}，P_{ikl}，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，サーバ関してベクトル化したものを、$\mathbb{S}^*_o$，$\mathbb{S}$，$\mathbb{P}$，$\mathbb{N}$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$\mathbb{N}$からクラスiの処理要求を1減じたベクトルを$\mathbb{N}_i$とすると，定理1より次式が成立する。

$$N^*_{oij} = f_j(\mathbb{S}^*_o, \mathbb{S}, \mathbb{P}, \mathbb{N}_i) \qquad (\forall i,\ \forall j \in I) \qquad (3.12)$$

(3.12)式を(3.10)式に代入すると次式が得られる。

$$S^*_{oi} = S_{oi} + \left(\sum_j f_j(\mathbb{S}^*_o, \mathbb{S}, \mathbb{P}, \mathbb{N}_i)\cdot S_{oj}\right) / \left(1 + f_i(\mathbb{S}^*_o, \mathbb{S}, \mathbb{P}, \mathbb{N}_i)\right) \qquad (\forall i \in I,\ \forall j \in I-\{i\}) \qquad (3.13)$$

ただし，(3.13)式は右辺に、$\mathbb{S}^*_o$を含むため、これより直ちに、S^*_{oi}を得ることはできない。ここでは、S^*_{oi}の上限値と下限値を求め、S^*_{oi}が取り得る値の範囲を得，次に2分検索法により(3.13)式を満たすS^*_{oi}のセット，すなわち，$\mathbb{S}^*_o$を求める手法をとる。

まず、S^*_{oi}が取り得る上限値と下限値を示す。S^*_{oi}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の時である。一方，上限値は明らかに、$f_i = 0$，$f_j = N_j$（$\forall j \in I-\{i\}$）が成立した時である。これにより、上限値，下限値は次式を満たす。

$$\mathrm{MAX}(S^*_{oi}) = S_{oi} + \sum_j N_j S_{oj} \qquad (\forall i \in I,\ \forall j \in I-\{i\}) \qquad (3.14)$$

$$\mathrm{MIN}(S^*_{oi}) = S_{oi} \qquad (\forall i \in I) \qquad (3.15)$$

以上により、すべてのクラスのS^*_{oi}の上限値と下限値が定まるため、2分検索法により、$\mathbb{S}^*_o$（すべてのクラスのS^*_{oi}）を得ることができる。ただし、2分検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計算量が増大する。

$\mathbb{S}^*_o$，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、積形解を持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なう。

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なう。

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なった。それぞれのクラスのアクセス当たりの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バウンドを表わす指標Bを（2/（S_1（P_{101}＋P_{201}）））/（$1/S_{01}+1/S_{02}$）として，Bが1/100～100の範囲についての評価を行なった。Bは，それぞれのクラスの処理要求のCPUサーバの完了率の和と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より充分小さい時，I/Oバウンド，1より充分大きい時，CPUバウンド，1附近（$S_{01}=S_{02}$の時には，B＝1の時が，最も負荷がバランスしているが，$S_{01}\neq S_{02}$の時にはこれが成立しない。）の時，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

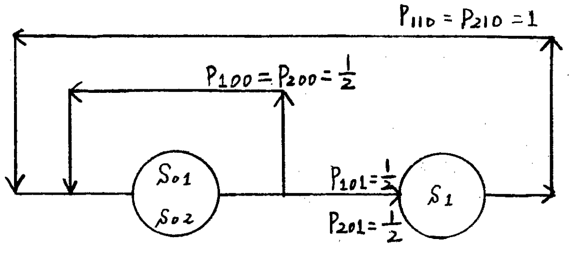

図3．厳密解との精度検証に用いたモデル

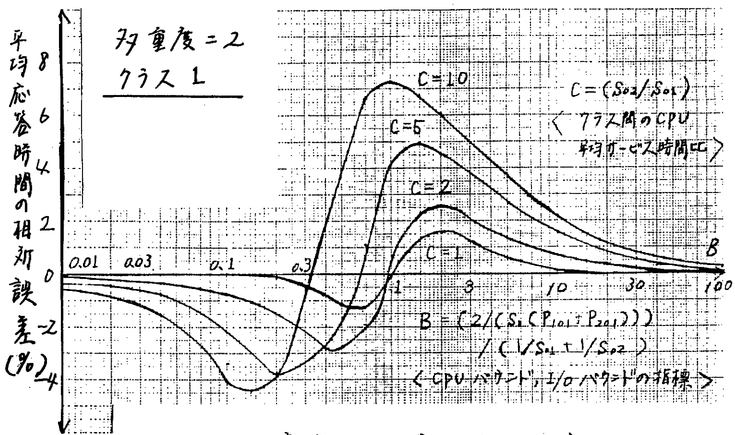

図4．厳密解による評価結果（1）

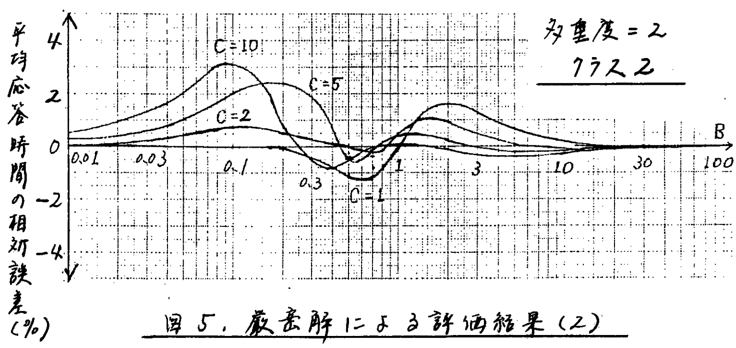

図5．厳密解による評価結果（2）

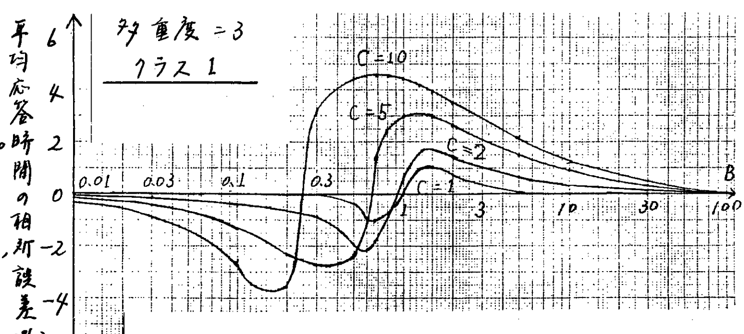

図6．厳密解による評価結果（3）

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時には誤差が大きく，CPUバウンド，I/Oバウンドの時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がわかる。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領域からI/Oバウンドによった領域では、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バウンドによった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バウンドによった領域では、過大評価となるが、バ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。以上が主な結果であるが，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、本手法が充分に有用であることを示している。

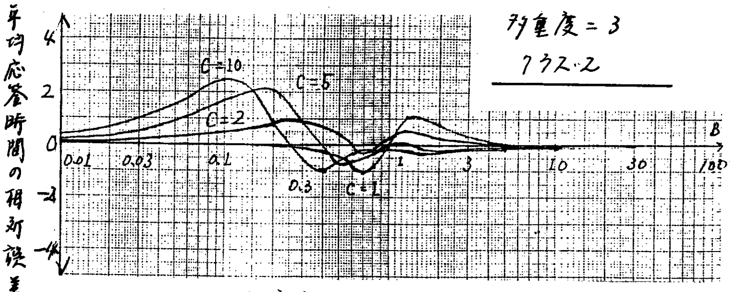

図7．厳密解との評価結果(4)

4．おわりに

クラスごとに平均サービス時間が異なる（分布はすべて指数分布）処理要求に対してFCFSスケジューリングを行な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き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：N_{ij}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：N^*_{ij}と等しいということを仮定している。N_{ij}とN^*_{ij}は，モデルがI/Oバウンド，または，CPUバウンドになると同じ値に収束する。従って，本近似手法は負荷がバランスしている領域では近似による誤差が大きいが，CPUバウンド，または，I/Oバウンドの領域では，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なった。この結果，誤差が生じ易い負荷がバランスしている領域でも近似誤差は10%未満であったことから、システム設計の初期段階において用いる手法としては、充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) 他の積形解を持たないスケジューリング（例えば，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)式を満たす解を単に2分検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大する。このための高速アルゴリズムの検討

謝辞　終わりに，本研究について御指導いただいた東京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下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下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いたします。

（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；資源割り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，第23巻，第5号，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，第24巻，第5号，(1983)(採録予定)
5) Chandy, K. M. et al ; Parametric Analysis of Queueing Networks, IBM J. of R. & D. Vol.19, No.1, pp.36−42 (1975)
6) Sauer, C. H. et al ; Approximate Analysis of Central Server Models, IBM J. of R. & D. Vol.19, No.3, pp.301−13 (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，その他：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，pp.12-1-1〜10 (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 358−71 (1981)
11) Kleinrock, L. : Queueing Systems, Vol.1, John Wiley (1975)