逐次アクセス資源のある計算機システムの待ち行列網による近似評価の一手法

木下俊之[†] 高橋幸雄[††]

更新をともなうファイルなどの逐次アクセス資源は，計算機システムの性能に大きな影響を及ぼす．この逐次アクセス資源（以下，資源と呼ぶ）のある計算機システムを性能解析する方法として，通常の待ち行列網に資源と資源待ち行列を付加した網をマルコフ連鎖でモデル化し，その平衡方程式を数値的に解いて性能値を求める方法がある．しかしこの方法はジョブ数や資源数が増えるとモデルの状態数が膨大となり，数値計算が困難になるという問題がある．そこで本論文では，この数値計算上の困難を回避するための近似モデルを提案する．近似の考え方は，状態縮約法を応用し，ジョブが資源を要求/獲得しているパターンで網の状態を縮約することにより状態数を削減し，マルコフ連鎖の平衡方程式を数値的に解ける範囲を拡大する．また縮約した状態内の網の振舞いは積形解で近似することで計算を簡単化する．この近似モデルについて，数値実験により状態数の削減効果と近似精度について，従来の方法との比較で検証した．これによりマルコフ連鎖の状態数を 2〜3 桁削減でき，またジョブの平均応答時間などが実用の範囲内の精度を持つことを確認した．

An Approximate Method by Queuing Network Modeling for Performance Evaluation of Computer Systems with Exclusively-used Resources

Toshiyuki Kinoshita[†] and Yukio Takahashi[††]

In computer systems, job conflicts occur at accessing an *exclusively-used resource*. A typical example is conflicts at accessing a group of updatable files. When a job uses a part of these files, other jobs are not allowed to access any of them. These conflicts affect the performance of the system significantly. In our previous work, we introduced a queuing network model to analyze the influence of the conlficts on the performance of the computer systems. It can provide most of performance measures requested, but the analysis requires a large computational burden even for models of medium size since the stationary distribution has to be calculated for a Markov chain with a large number of states. In this paper, we propose an approximate method of the model which reduces the number of states of the model and our computation efforts. The policy and the algorithm of the approximate method are described and its accuracy is analyzed by numerical experiments.

1. まえがき

計算機システムでのファイルへのアクセスを考えてみると，それが更新をともなう場合は，ファイルデータの一致性を保つために，1 つのジョブの処理中は他のジョブからの当該データへのアクセスを禁止することがある．この場合，ファイルは先のジョブに割り付けられた形になり，このジョブのみが当該データに関する処理を実行できる．このようにある瞬間にはたかだか 1 つのジョブのみがアクセスでき，後からのアクセスは待たされて逐次化される資源を，逐次アクセス資源と呼ぶ．

この逐次アクセス資源は計算機システムの性能に重大な影響を及ぼすにもかかわらず，一般の待ち行列網でモデル化した場合，積形解を持たないため解析が困難である．このため逐次アクセス資源を解析するには，別にモデルを構築する必要がある[8)〜10)]．我々は逐次アクセス資源（以下，単に資源と呼ぶ）を持つ計算機システムを解析するために，通常の待ち行列網に資源と資源待ち行列を付加し，資源を要求/待ち合せ/解放する手順を定めた待ち行列網を定義し，これを記述す

† 株式会社日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.
†† 東京工業大学大学院情報理工学研究科
Development of Mathematical and Computing Science, Tokyo Institute of Technology

るマルコフ連鎖の平衡方程式を解いてその性能指標を求める方法を提案した[12]．そしてこの方法を応用して，逐次アクセス資源の 1 つであるファイル資源について，これを複数に分割することによる性能向上効果について解析例を示した．この方法は，逐次アクセス資源のある待ち行列網をマルコフ連鎖を用いて厳密にモデル化するので，解を必要なだけ精密に求めることができる．しかし平衡方程式を数値的に解くので，使用する計算機の性能やメモリ量の制約を受ける．特にジョブ数や資源数が大きくなるとマルコフ連鎖の状態数が膨大になって計算時間が長大になり，また方程式の係数行列をメモリ上に展開できず計算が困難になるという問題をはらんでいる．

そこで本論文では，この数値計算の手間と必要なメモリ量を大幅に削減し，解析可能な範囲を拡大させる近似モデルを提案する．ここではよく知られた状態縮約法（Aggregation Method）の考え方を応用する．状態縮約法とは，網の状態を何らかのパターンで縮約し☆，1 つの縮約された状態下の網の振舞いは積形解などで近似し，縮約された各状態の発生確率は待ち行列網の解析法やマルコフ連鎖の数値解析法などを用いて求め，縮約された各状態下の性能値をその状態の発生確率で平均をとってその性能値の近似値とする方法である．この状態縮約法による近似解法は，従来から様々な形で研究/応用されている．Parametric Analysis[2),3),8)]では「ノードの縮約」を行うが，これはとりもなおさず「状態の縮約」を意味し，したがってこの方法は状態縮約法の 1 つといえる．最近ではこれを発展させ，主に計算機システム内のプロセスとハードウェアの振舞いを一括して待ち行列網でモデル化することを目的とした階層化待ち行列網モデル（Layered Queuing Models: LQMs）が研究されている[4),6),7),13)]．この LQMs ではプロセスの振舞いを記述する網（上位層）とハードウェアをモデル化する網（下位層）を連動させて考える．この上位層の各状態は下位層の状態を縮約したものと考えられるから，LQMs は状態縮約法の 1 つと見なすことができる．この LQMs はかなり広い範囲のモデルを含んでおり，しかも縮約された状態間の関係が待ち行列網（上位層の網）で表現されるので状態遷移が視覚的にとらえられて理解しやすいという利点がある．

我々の近似モデルをこの LQMs にあてはめると，資源の要求/獲得/解放を記述する部分が LQMs の上位層に対応する．しかし我々のモデルでは複数の逐次アクセス資源をデッドロックを起こさない範囲で各ジョブから独立に要求/解放できるようにしたい（したがってある瞬間に 1 つのジョブが複数の資源を占有してよい）との理由から待ち行列網では表現できず，残念ながら LQMs は適用できなかった．そこで我々のモデルでは資源の要求/獲得/解放を記述する部分を独自のマルコフ連鎖で表現し解析することとした．このマルコフ連鎖の状態はもとの網の状態を縮約して設定するが，数値計算量と近似精度のトレードオフを考慮した場合，どの程度の縮約をすべきかがなお問題として残る．種々の検討の結果，求めようとする性能値を算出するうえで直接に必要な情報のみを含むレベルの縮約を行うことが，実用上最も有効であるとの結論に至った．そこで我々のモデルでは，状態を縮約するパターンおよび縮約された状態間の遷移トラフィックを次のように設定した．

（1） 網の状態は網中のジョブがその時点で複数の資源をどう要求/占有しているかのパターンによってタイプ分けできるが，状態の縮約は各タイプのジョブ数が同じものを集めて行う．

（2） 縮約された状態間の遷移トラフィックは，資源待ち行列から次にどのタイプのジョブが出てくるかの頻度（発生確率）に依存する．そこでその頻度を資源待ち行列中の各タイプのジョブ数に比例するものと仮定する．

これにより性能値を求めるために，ある意味での必要最小限の分解能を持つように状態を縮約でき，マルコフ連鎖の状態数を最小限まで減らして（木下，高橋[12)]のそれに比べて状態数を 2〜3 桁程度少なくできる）計算可能な範囲を大幅に拡大できる．

本論文の構成は以下のとおりである．2 章で逐次アクセス資源のある待ち行列網の構成と状態を定義し，3 章で提案する近似モデルについて説明する．4 章でこの近似モデルをオンライン実時間システムに適用した数値実験例を示し，状態数の削減効果と近似の精度を検証する．最後に 5 章でむすびを述べる．

2. モデルの記述

2.1 逐次アクセス資源のあるセントラルサーバモデル

本論文で扱う逐次アクセス資源のある待ち行列網モデルは木下，高橋[12)]で扱ったものと同一で，待ち行列網の 1 つであるセントラルサーバモデル[1),5)]に資源と資源待ち行列を付加し，資源へのアクセスの手順を定めたものである（**図 1** のイメージ図参照）．

☆ 縮約とは，複数の状態をまとめて 1 つの状態に置き換える操作のことである．したがって縮約された状態数は，もとの状態数よりも少なくなる．

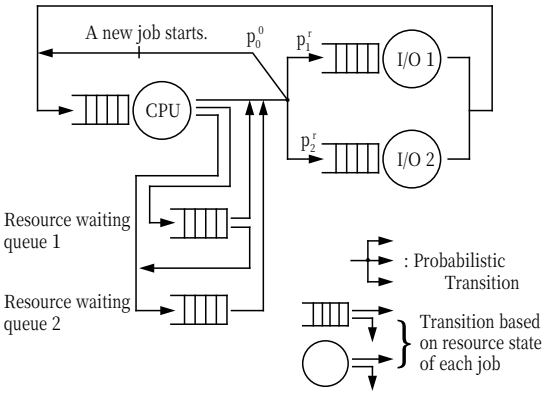

図 1 逐次アクセス資源のあるセントラルサーバモデルのイメージ図（資源が 2 個の場合）
Fig. 1 Image diagram of central server model with 2 exclusively-used resources.

網は 1 つの CPU ノードと複数の I/O ノードからなり，網の中には一定数のジョブが存在してノード間を移動する．N は網中のジョブ数を表し，M は I/O ノード数を表す．ジョブは 1 から N まで添え字 n で，I/O ノードは 1 から M まで添え字 m で番号付けられる．$m = 0$ のときは CPU ノードを表すものとする．あるノードでのサービス時間は共通の指数分布に従う互いに独立な確率変数で，他のノードでのサービス時間とも独立である．CPU ノードでのサービス率を μ_0，I/O ノード m でのサービス率を μ_m とする．すべてのノードでジョブは FCFS（First Come First Served）規律でスケジュールされる．逐次アクセス資源を扱うために，網に資源と資源待ち行列を付加する．F を資源数とし，資源に 1 から F まで通し番号を付け，その番号を f または f_i で参照する．

（1） ジョブの資源状態

ジョブの資源状態とは，その時点でジョブが必要としている資源，すなわちジョブが要求中または占有中の資源のことである（その際ジョブがすでにその資源を獲得しているか否かにはかかわらない）．資源状態に通し番号を付け，その番号を r または r_j で参照する．したがってジョブ n の資源状態 r はこのジョブがその時点で必要としている資源の集合として $\{f_1, f_2, \ldots, f_{F_n}\}$（$0 \leq F_n \leq F$）と表される．$r = 0$ は必要としている資源がない状態（すなわち $F_n = 0$）を表すものとする．このジョブの資源状態は CPU のサービス終了時に確率的に変化する．この変化は，ジョブや待ち行列網のそれまでの履歴とは独立である．ジョブの資源状態が r_1 から r_2 に変化する確率を $p^{r_1 r_2}$ とする．たとえば資源が 2 個の場合，ジョブの資源状態は

$r=0 : \phi$ …… 資源を必要としない状態
$r=1 : \{1\}$ …… 資源 1 のみを要求している状態
$r=2 : \{2\}$ …… 資源 2 のみを要求している状態
$r=3 : \{1, 2\}$ … 資源 1 と 2 を要求している状態

の 4 通りが考えられ，それらの間に推移確率 $p^{r_1 r_2}$ $(r_1, r_2 = 0, 1, 2, 3; \sum_{r_2=0}^{3} p^{r_1 r_2} = 1)$ を与える．

（2） 資源の獲得/解放

ジョブは，CPU ノードでのサービスの終了時に確率的に資源を要求/解放する．この資源の要求/解放は，ジョブの資源状態の変化によって表現される．すなわち CPU サービス終了時に，ジョブの資源状態が資源 f を含まない状態から含む状態に変化するとそのジョブは資源 f を要求したことを意味し，資源状態が資源 f を含む状態から含まない状態に変化するとそのジョブは資源 f を解放したことを意味する．たとえば上記（1）の例で，資源状態が $r = 0$ から $r = 1$ に変化することは資源を何も占有していないときに資源 1 を要求することを，$r = 1$ から $r = 3$ への変化は資源 1 を占有しているときにさらに資源 2 を要求することを，また $r = 3$ から $r = 0$ に変化することは資源 1 と 2 を占有しているときにその両方を解放することをそれぞれ意味する．資源状態の変化がジョブや待ち行列網のそれまでの履歴とは独立であることから，資源の要求や解放は，ジョブがその時点で要求している資源には依存するがジョブや待ち行列網のそれまでの履歴とは独立である．

（3） ノード遷移

ジョブが CPU のサービス終了時に新たな資源を要求しなかったり，資源を要求してそれをすぐに獲得できたりしたときは，次に CPU ノードか I/O ノードのいずれかに進む．いずれのノードに進むかは確率的に選択される．この選択確率はジョブの資源状態によって異なり，ジョブの資源状態が r のときのノード m の選択確率は p_m^r である．ただし後に述べるように，CPU から CPU へ直接に遷移した場合は，そのジョブは新たなジョブに生まれ変わったと考える．その場合ジョブが資源を占有したまま終了できないので，資源を占有した状態で CPU → CPU の遷移は発生しない，すなわち $r > 0$ ならば $p_0^r = 0$ とする．

（4） デッドロックの回避

2 つ以上の資源を要求する際には，デッドロックを回避するために，資源を獲得する順序が決められている．したがって複数の資源を同時に要求したときは，要求したうちで獲得順序が先のものから獲得しにいく．また獲得順序が後の資源を占有していてさらに獲得順序が先の資源を要求するときは，この新たに要求する

資源より獲得順序が後の資源をいったんすべて解放し，ただちにこれらを含めてその時点で必要な資源を改めて同時に要求するという操作を行う．ここでは番号 f の値が小さいほど資源 f は先の資源獲得順序であるとする．したがって上記（1）の例では資源獲得順序は資源1が先で資源2が後なので，資源状態 $r=2$ から3への遷移（資源2を占有しているときにさらに資源1を要求した）は，$r=2\to 0\to 3$ の2回の変化が瞬間に連続して起こったと考えて資源2をいったん解放した後に資源1と2を同時に要求し，この順に獲得しにいくことになる．

（5） 資源待ち行列

ジョブがCPUサービス終了時に資源を要求したとき，もしその資源がすでに他のジョブによって占有されていてすぐには獲得できない場合は，ジョブは対応する資源待ち行列に入って資源が解放されるのを待つ．資源待ち行列を番号で表し，資源 f には同じ番号の資源待ち行列 f が対応するものとする．資源待ち行列ではジョブはFCFS規律でスケジュールされる．

この資源待ち行列への到着/出発は，次のようにまとめられる．

（a） 資源待ち行列 f への到着

（i） 資源 f が占有されているときにジョブがこの資源を要求すると，このジョブは資源待ち行列 f の最後尾に入って資源 f が解放されるのを待つ．複数の資源を同時に要求したときは，資源獲得順序が先の資源から順に獲得していき，最初に獲得できなかった資源が f のとき，この資源待ち行列 f の最後尾に入る．その際，すでに獲得した資源を占有したまま資源待ち行列に入る．

（ii） あるジョブが資源 f' と $f\,(f'<f)$ を要求していて資源獲得順序が先の資源 f' に対応する資源待ち行列の先頭にいるときに，その資源 f' が解放されるとこれを獲得し，次に資源 f を獲得しようとしたがこれが占有されていると資源待ち行列 f の最後尾に入る．その際，資源 f' を占有したまま資源待ち行列 f に入る．

（b） 資源待ち行列 f からの出発

（i） あるジョブが資源待ち行列 f の先頭にいるときに資源 f が解放されると，このジョブは資源 f を獲得して資源待ち行列 f から出発する．そして他に要求していて獲得していない資源があれば，それを獲得しにいく．

（ii） こうして要求しているすべての資源を獲得すると，ジョブはすべての資源待ち行列から出発してI/Oノードのいずれかを確率的に選択して移動する（この場合ジョブは資源を占有しているので，CPUノードに進むことはない）．そのときのジョブの資源状態が r であれば，I/Oノード m の選択確率は p_m^r である（上記の（3）で導入されたものと同一）．

2.2 ジョブのライフタイム

ジョブはCPUノードでのサービス終了後，I/Oノードや資源待ち行列を経由した後かまたは直接に，CPUノードに戻ってくる．このうちCPUからCPUに直接に遷移した場合，ジョブは新しいジョブに生まれ変わったと考える．したがってCPU → CPUの遷移から次のCPU → CPUの遷移までの間が1つのジョブのライフタイムである．このライフタイムはジョブの応答時間と考えることができる．その際ジョブは資源を占有したまま終了できないので，ジョブの生まれ変わりであるCPU → CPUの遷移はジョブが資源を占有していないときにのみ起こるものとする（2.1節（3）参照）．

2.3 待ち行列網の状態

網内でジョブが存在する場所はノード（CPUノードとO/Iノード）と資源待ち行列なので，この両者を一括して扱えると都合がいい．そこでノード番号 m を延長して資源待ち行列をノードの後ろに番号付けた通し番号を設定し，同じ記号 m で表す．すなわち資源待ち行列 f は $m=M+f$ とも参照される．したがって記号 m は，$m=0,1,\ldots,M$ のときはノード m を，$m=M+1,\ldots,M+F$ のときは資源待ち行列 $f\,(=m-M)$ を表すものとする．

待ち行列網の状態は，次のようなベクトルで表される．

$$\delta=(k_0^1\cdots k_0^{N_0};\,k_1^1\cdots k_1^{N_1};\cdots;\,k_M^1\cdots k_M^{N_M};\\ k_{M+1}^1\cdots k_{M+1}^{N_{M+1}};\cdots;\,k_{M+F}^1\cdots k_{M+F}^{N_{M+F}})$$

ただし N_m はノード m または資源待ち行列 $f\,(m=M+f)$ のジョブ数とする．この k_m^q は対応するジョブの資源状態を表す．すなわち $k_m^q=r$ ならばノード m または資源待ち行列 $f\,(m=M+f)$ の第 q 番目に資源状態 r のジョブがいることを表す．したがって $k_0^1\cdots k_0^{N_0}$; $k_1^1\cdots k_1^{N_1}$; $\cdots$; $k_M^1\cdots k_M^{N_M}$; $k_{M+f+1}^1\cdots k_{M+f+1}^{N_{M+f+1}}$; $\cdots$; $k_{M+F}^1\cdots k_{M+F}^{N_{M+F}}$（すなわちすべてのCPU，I/Oノードと資源待ち行列 $f+1$ から F）の中には，資源 f を含む資源状態のジョブはたかだか1つである（資源 f を占有しているジョブのみ）．一方 $k_{M+1}^1\cdots k_{M+1}^{N_{M+1}}$; $\cdots$; $k_{M+f}^1\cdots k_{M+f}^{N_{M+f}}$（資源待ち行列1から f）の中には，資源 f を含む資源状態のジョブは複数存在しうる（資源 f を要求したが獲得できずに資源待ちになっているジョブが含まれるため）．

δ の可能なすべての集合を Δ とする．逐次アクセス資源のあるセントラルサーバモデルの状態遷移は，マルコフ連鎖 $\{\delta(t)\}$ によって記述される．容易に確かめられるように $\{\delta(t)\}$ はそのすべての状態が任意の初期状態から到達可能なので，既約かつエルゴード的である．

3. 計算量削減のための近似モデル

性能評価に必要なビジー率，スループット，平均応答時間といった性能指標は，マルコフ連鎖 $\{\delta(t)\}$ の定常状態確率 $P(\delta)$ から求めることができる．しかしこの $P(\delta)$ を求めるためには，状態数と同数の未知数を含む平衡方程式（連立一次方程式）を解かなければならない．この δ の状態数は 4.2 節の表 2 に示すようにジョブ数や資源数が増加するにつれて指数関数的に増加するので，この方法では限界がある．そこで近似モデルを導入してこの計算上の困難を減らすことを考える．ただし以下では資源数 F が 2 の場合を考える（$F \geq 3$ の場合も同様である）．また「ノード内のジョブ」とは CPU または I/O ノード中にいるジョブを指し，これと「資源待ち行列内のジョブ」を区別していい表す．

3.1 近似の考え方

網内ではジョブが資源を要求/解放することを契機に，資源待ち行列に出入りしたり他のジョブが資源を獲得したりすることが起こる．一方，資源の要求や解放がない間は，すべてのジョブの資源状態は変わることがなく資源待ち行列への出入りもない．この間はノード内のジョブ数は一定で，ジョブが占有している資源は不変であり，状態遷移はジョブのノード間の移動のみである．したがってこの間は「資源へのアクセスのぶつかりによる資源待ち行列への遷移」という非確率的な遷移が発生しないので，この間の網の振舞いは資源のない通常の待ち行列網と同様であり，定常分布は近似的にこの網の積形解で表されると考えられる．我々の近似モデルではこの考えを利用する．

すなわち，網内のジョブの振舞いを「資源の要求/解放がない間（1 つの要求/ 解放から次の要求/解放までの時間区間）のノード内での振舞い（各ノードでサービスを受けたり，ノード間を移動したりするなど）」と「資源の要求/解放の振舞い」に分け，前者は通常の待ち行列網の積形解で近似し，後者はその資源の要求/解放の振舞いを記述するマルコフ連鎖を構成してその定常分布を求めることにより解析する．具体的には δ を次のように縮約（Aggregate）して近似的に $P(\delta)$ を求める．すなわち各 δ について，

（a） CPU および I/O ノード全体の中にいる各資源状態のジョブ数と，

（b） 各資源待ち行列中にいる各資源状態のジョブ数が等しいような δ を縮約して新しい状態 γ とする．いいかえると各 δ について CPU および I/O ノード中にいる資源状態 r のジョブ数を $N^r_{Node} = \sum_{m=0}^{M} N^r_m$ とすると，各 δ のすべての資源状態 r について $N^r_{Node}, N^r_{M+1}, \ldots, N^r_{M+F}$ が等しいような δ を 1 つの状態 γ に縮約する．これは各ジョブが資源を要求/獲得しているパターンによって縮約していると考えることができる．そしてこの γ を状態とするマルコフ連鎖を考え，その平衡方程式を数値的に解いて定常分布 $P(\gamma)$ を求める☆．

この同じ γ に縮約される δ の CPU および I/O ノード中の各資源状態 r のジョブ数は N^r_{Node} で等しいので，これらの δ のうち CPU および I/O ノードの状態 $(k^1_0 \cdots k^{N_0}_0; k^1_1 \cdots k^{N_1}_1; \cdots; k^1_M \cdots k^{N_M}_M)$ が同一のものを集めて $\overline{\delta}_d$ $(d = 1, 2, \ldots, D_\gamma)$ とすると（すなわち δ に通し番号を付けて δ_i とするとき，$\overline{\delta}_d = \bigcup_{(k^1_0, \ldots, k^{N_M}_M) \text{ が同一の } \delta_{i_d}} \delta_{i_d}$，$\bigcup_{d=1}^{D_\gamma} \overline{\delta}_d = \gamma$），この $\overline{\delta}_d$ は資源のある待ち行列網から資源状態の変化と資源待ち行列を取り去った通常の複数ジョブクラスの待ち行列網の状態を表現している（同じ資源状態のジョブが 1 つのジョブクラスを構成する）．そこで次に述べる「積形解の仮定（仮定 A）」をおくことにより，状態 γ の条件の下での $\overline{\delta}_d$ の条件付き確率 $P(\overline{\delta}_d|\gamma)$ は，通常の待ち行列網の積形解で近似する．これにより $\overline{\delta}_d$ の定常確率は $P(\overline{\delta}_d) = P(\overline{\delta}_d|\gamma)P(\gamma)$ で求められる．この $P(\overline{\delta}_d)$ は δ の CPU および I/O ノードに関する周辺分布だから，これから CPU および I/O ノードの性能値を求めることができる．

一方，$P(\gamma)$ は各資源待ち行列中のジョブ数についての周辺分布と考えられるから，これにより資源待ち行列中の各資源状態の平均ジョブ数を求めることができる．また次に述べる「資源待ち行列から出てくるジョブの頻度に関する仮定（仮定 B）」により資源待ち行列からのスループットが分かるので，リトルの公式を用いて各資源状態のジョブの資源待ち行列での平均滞在時間を求めることができる☆☆．

この近似モデルでは，以下の仮定を設定している．

☆ 一般にあるマルコフ連鎖の状態を縮約した状態の確率過程はマルコフ連鎖にならないが，それと定常分布が等しいようなマルコフ連鎖が構成できるので，その定常分布として求める．

（1） 積形解の仮定

γ に制限した待ち行列網の状態は，γ に遷移する直前の状態の影響を受けたり γ の持続時間内には定常状態に達しなかったりして，γ の下での $\overline{\delta}_d$ の条件付き確率 $P(\overline{\delta}_d|\gamma)$ は必ずしも定常分布にならないことが考えられる．しかしここではこれが近似的に定常分布であると仮定する．すなわち次の仮定をおく．

[仮定 A] $P(\overline{\delta}_d|\gamma)$ は γ に制限した待ち行列網の定常分布（積形解）で近似できる．

（2） 資源待ち行列から出てくるジョブの頻度に関する仮定

γ の状態遷移は資源待ち行列内でジョブが並ぶ順序に依存する．たとえば資源待ち行列 1 に資源状態 1 と 3 のジョブ（つまり資源 1 を要求しているジョブと資源 1 と 2 を要求しているジョブ）が混在しているときに，資源 1 が解放されると，そのどちらが資源 1 を獲得するか（つまりそのどちらが資源待ち行列の先頭にいるか）によって γ からの状態遷移が異なるからである．これについて次の仮定をおく．

[仮定 B] 次にどの資源状態のジョブが資源を獲得して資源待ち行列から出てくるかの頻度（発生確率）は，資源待ち行列中の各資源状態のジョブ数に比例する．

すなわち上の例で資源待ち行列 1 にいる資源状態 1 と 3 のジョブ数をそれぞれ n'_1，n'_3 とすると，次に資源 1 を獲得して資源待ち行列 1 から出てくるジョブが，

“資源状態 1 のジョブである頻度”：
“資源状態 3 のジョブである頻度”
$= n'_1 : n'_3$

と仮定する．この仮定は，資源待ち行列内でのジョブの並び方が等確率で発生することを仮定していると考えることもできる．この仮定 B により，通常の性能評価に用いられる典型的な性能値（ビジー率，スループット，応答時間など）はすべて算出することができ，γ はそれを可能とするある意味での必要最低限の解像度を持つように縮約された状態であるといえる．

この近似モデルでは定常分布 $P(\gamma)$ について平衡方程式を数値的に解く必要があるが，4.2 節の表 2 に示すようにその状態数（状態 γ の個数）は縮約する前の状態数（状態 δ の個数）に比べてはるかに少なく，数値計算の手間（計算量とメモリ量）を大幅に削減できる．このように木下，高橋[12]と同一の逐次アクセス資源のある待ち行列網について，文献 12) に比べてはるかに少ない計算量で性能値の近似値を求めることができ，解析可能な範囲を広げることができる．

3.2 状態 γ の詳細

資源の要求/解放の遷移を記述するための状態 γ は，$F = 2$ の場合，具体的には次のように表現できる．すなわち資源状態が r のジョブ数を $n_r\,(r = 0 \sim 3, \sum_{r=0}^{3} n_r = N)$ とし，またノード内でジョブがどの資源を占有しているかで分類して

k=ϕ ： ノード内に資源を占有しているジョブがいない場合
k=$\{1\}$ ： ノード内に資源 1 を占有しているジョブがいて，資源 2 を占有しているジョブがいない場合
k=$\{2\}$ ： ノード内に資源 2 を占有しているジョブがいて，資源 1 を占有しているジョブがいない場合
k=$\{12\}$ ： ノード内に資源 1 と 2 の両方を占有しているジョブがいる場合
k=$\{1\}\{2\}$ ： ノード内に資源 1 を占有しているジョブと資源 2 を占有しているジョブがいて，両者が異なる場合

とするとき，各状態 γ は (k, n_0, n_1, n_2, n_3) により規定される．そこでこのことを明示して $\gamma(k, n_0, n_1, n_2, n_3)$ と表す．たとえば $N = 9$ のとき $\gamma(\{1\}, 5, 3, 0, 1)$ は，資源を要求していないジョブ 5 個と資源 1 を占有しているジョブ（当然 1 個）がノード内にいて，資源 1 を要求しているが獲得できないでいるジョブ 2 個と資源 1 と 2 を要求しているがどちらも獲得できないでいるジョブ 1 個が資源待ち行列 1 中にいて，資源待ち行列 2 には何もない状態を表す．この $\gamma(k, n_0, n_1, n_2, n_3)$ は (k, n_0, n_1, n_2, n_3) のすべての組合せについて存在するわけではない．そこで以下に存在する組合せを明示する（これらの一覧を**表 1** に示す）．

（a） $k = \phi$ のとき

この場合，いずれの資源待ち行列にもジョブは存在しない（もしあれば資源を獲得してノード内にいるはずだから）．したがって資源を要求しているジョブが 1 つもない状態，すなわち $\gamma(\phi, N, 0, 0, 0)$ である（つまり $\gamma(\phi, n_0, n_1, n_2, n_3)$ はこの (n_0, n_1, n_2, n_3)=$(N, 0, 0, 0)$ のみが起こる）．

☆☆ 仮定 B の説明中にもあるように，仮定 B は δ_{i_d} が $\overline{\delta}_d$ の中で等確率で発生することを述べていると考えてもよい．これによれば $P(\delta_{i_d})$ は $P(\overline{\delta}_d)$ を $\overline{\delta}_d$ に含まれる δ_{i_d} で等分した値で近似される（すなわち $\overline{\delta}_d$ に含まれる δ_{i_d} の個数を p_d とすると，$P(\delta_{i_d}) = P(\delta_{i_d}|\overline{\delta}_d)P(\overline{\delta}_d) \sim 1/p_d \cdot P(\overline{\delta}_d)$）．これにより資源待ち行列の性能値を求めることもできる．

表 1 可能な状態 $\gamma(k, n_0, n_1, n_2, n_3)$ とそのときのノードと資源待ち行列中のジョブ数
Table 1 Possible state $\gamma(k, n_0, n_1, n_2, n_3)$ and number of jobs in CPU and I/O nodes or resource waiting queue when the network is in state $\gamma(k, n_0, n_1, n_2, n_3)$.

項番	可能な $\gamma(k, n_0, n_1, n_2, n_3)$ のケース		
	ノード中のジョブ数	資源待ち行列 1 中のジョブ数	資源待ち行列 2 中のジョブ数
(a)	$\gamma(\phi, N, 0, 0, 0)$		
	$\underbrace{N}_{\text{資源状態 } r=0 \text{ のジョブ数}}$	0	0
(b)	$\gamma(\{1\}, n_0, n_1, 0, n_3)$ $(n_0, n_3 \geq 0, n_1 \geq 1, n_0 + n_1 + n_3 = N)$		
	$\underbrace{n_0}_{r=0} + \underbrace{1}_{r=1}$	$\underbrace{n_1 - 1}_{r=1}$	0
(c-1)	$\gamma(\{2\}, n_0, 0, n_2, 0)$ $(n_0 \geq 0, n_2 \geq 1, n_0 + n_2 = N)$		
	$\underbrace{n_0}_{r=0} + \underbrace{1}_{r=2}$	0	$\underbrace{n_2 - 1}_{r=2}$
(c-2)	$\gamma(\{2\}, n_0, n_1, n_2, n_3)$ $(n_0, n_1 \geq 0, n_2, n_3 \geq 1, \sum_{r=0}^{3} n_r = N)$		
	$\underbrace{n_0}_{r=0} + \underbrace{1}_{r=2}$	$\underbrace{n_1}_{r=1} + \underbrace{n_3 - 1}_{r=3}$	$\underbrace{n_2 - 1}_{r=2} + \underbrace{1}_{r=3}$
(d)	$\gamma(\{12\}, n_0, n_1, n_2, n_3)$ $(n_0, n_1, n_2 \geq 0, n_3 \geq 1, \sum_{r=0}^{3} n_r = N)$		
	$\underbrace{n_0}_{r=0} + \underbrace{1}_{r=3}$	$\underbrace{n_1}_{r=1} + \underbrace{n_3 - 1}_{r=3}$	$\underbrace{n_2}_{r=2}$
(e)	$\gamma(\{1\}\{2\}, n_0, n_1, n_2, n_3)$ $(n_0, n_3 \geq 0, n_1, n_2 \geq 1, \sum_{r=0}^{3} n_r = N)$		
	$\underbrace{n_0}_{r=0} + \underbrace{2}_{r=1,2}$	$\underbrace{n_1 - 1}_{r=1} + \underbrace{n_3}_{r=3}$	$\underbrace{n_2 - 1}_{r=2}$

(b) $k = \{1\}$ のとき

ノード内には資源 1 を占有しているジョブが 1 個と資源状態 0 のジョブが存在する．このほかに資源待ち行列 1 に資源 1 を要求しているが獲得できないでいる資源状態 1 または 3 のジョブが存在しうる．また資源状態 2 のジョブは存在しない（もしいれば資源 2 を獲得してノード内にいるので，$k = \{1\}\{2\}$ のはずだから）．したがってこの場合の γ は $\gamma(\{1\}, n_0, n_1, 0, n_3)$ $(n_0, n_3 \geq 0,\ n_1 \geq 1,\ n_0 + n_1 + n_3 = N)$ である．ここで $n_1 \geq 0$ ではなく $n_1 \geq 1$ とするのは，ノード内に資源状態 1 のジョブが必ず 1 個存在するためである．

(c) $k = \{2\}$ のとき

（b）とは逆に資源状態 1 のジョブは存在しない．そして資源状態 3 のジョブの有無によって，次の 2 通りが考えられる．

(c-1) 資源状態 3 のジョブが存在しない場合

網内には資源状態 0 と 2 のジョブしかいないから，考えられる γ は $\gamma(\{2\}, n_0, 0, n_2, 0)$ $(n_0 \geq 0,\ n_2 \geq 1,\ n_0 + n_2 = N)$ である．

(c-2) 資源状態 3 のジョブが存在する場合

資源状態 3 のジョブの 1 つが資源 1 を占有していて資源待ち行列 2 中で資源 2 を待っているという状態である．このときほかに資源待ち行列 1 には資源状態 1 または 3 のジョブが，資源待ち行列 2 には資源状態 2 のジョブが存在しうる．したがってこの場合の γ は $\gamma(\{2\}, n_0, n_1, n_2, n_3)$ $(n_0, n_1 \geq 0,\ n_2, n_3 \geq 1,\ \sum_{r=0}^{3} n_r = N)$ である．

(d) $k = \{12\}$ のとき

資源状態 3 のジョブが資源 1 と 2 を占有してノード内で実行中で，そのほかに資源待ち行列 1 に資源状態 1 または 3 のジョブが，資源待ち行列 2 に資源状態 2 のジョブが存在しうる．したがってこの場合の γ は $\gamma(\{12\}, n_0, n_1, n_2, n_3)$ $(n_0, n_1, n_2 \geq 0,\ n_3 \geq 1,\ \sum_{r=0}^{3} n_r = N)$ である．

(e) $k = \{1\}\{2\}$ のとき

資源状態 1 のジョブが資源 1 を，資源状態 2 のジョブが資源 2 をそれぞれ占有してノード内で実行中で，その他に資源待ち行列 1 に資源状態 1 または 3 のジョブが，資源待ち行列 2 に資源状態 2 のジョブが存在しうる．したがってこの場合の γ は $\gamma(\{1\}\{2\}, n_0, n_1, n_2, n_3)$ $(n_0, n_3 \geq 0,\ n_1, n_2 \geq 1,\ \sum_{r=0}^{3} n_r = N)$ である．

3.3 $\gamma(k, n_0, n_1, n_2, n_3)$ の状態遷移と $P(\gamma(k, n_0, n_1, n_2, n_3))$ の算出

状態 $\gamma(k, n_0, n_1, n_2, n_3)$ の状態遷移は次のように考えることができる．たとえば状態 $\gamma(\{1\}, 5, 3, 0, 1)$ から次に遷移する状態は，次の 6 通りである．

(a) $\gamma(\{1\}, 4, 4, 0, 1)$：資源状態 0 のジョブの 1 つが資源 1 を要求した．この資源 1 が占有されていたので資源待ち行列 1 に入った．

(b) $\gamma(\{1\}\{2\}, 4, 3, 1, 1)$：資源状態 0 のジョブの 1 つが資源 2 を要求した．この資源 2 が空いていたのでただちに獲得して実行を続けた．

(c) $\gamma(\{1\}, 4, 3, 0, 2)$：資源状態 0 のジョブの 1 つが資源 1 と 2 の両方を要求し，資源待ち行列 1 に入った．

(d-1) $\gamma(\{1\}, 6, 2, 0, 1)$：資源状態 1 のジョブが資源 1 を解放した．資源待ち行列 1 中の資源状態 1 のジョブが資源 1 を獲得してノードに出て実行を再開した．

(d-2) $\gamma(\{12\}, 6, 2, 0, 1)$：資源状態 1 のジョブが資

源 1 を解放した．資源待ち行列 1 中の資源状態 3 が資源 1 と空いている資源 2 を獲得してノードに出て実行を再開した．

(e) $\gamma(\{12\},5,2,0,2)$：資源状態 1 のジョブがさらに資源 2 を要求し，これを獲得して実行を続けた．

このうち（d-1）と（d-2）はともに資源状態 1 のジョブが資源 1 を解放することを契機として起こり，そのときに資源待ち行列 1 から資源状態 1 のジョブが出てくれば $\gamma(\{1\},6,2,0,1)$ に，資源状態 3 のジョブが出てくれば $\gamma(\{12\},6,2,0,1)$ に遷移する．このどちらが出てくるかの頻度は，3.1 節の仮定 B により待ち行列中にいるそれぞれのジョブ数に比例する．すなわち $\gamma(\{1\},5,3,0,1)$ から

“$\gamma(\{1\},6,2,0,1)$ への遷移の頻度”：
“$\gamma(\{12\},6,2,0,1)$ への遷移の頻度”
$= (n_1-1):n_3 = 2:1$

である．この状態遷移と遷移確率 $p^{r_1r_2}$ により，定常確率 $P(\gamma(k,n_0,n_1,n_2,n_3))$ に関する平衡方程式を構成し，これを数値的に解いて $P(\gamma(k,n_0,n_1,n_2,n_3))$ を求める．

3.4 状態 $\gamma(k,n_0,n_1,n_2,n_3)$ での性能値の算出

(1) ノードの性能値

表 1 に示すように，各状態 $\gamma(k,n_0,n_1,n_2,n_3)$ の下ではノード内のジョブ数は一定である．したがって Baskett ら[1]による単一または複数ジョブクラスの待ち行列網のケースに帰着され，積形解を持つ．これにより各ノードのビジー率，滞在ジョブ数，スループットを求めることができる．ジョブクラス数は，一般に $k=\phi$ のときは 1 クラス，$k=\{1\},\{2\},\{12\}$ のときは 2 クラス，$k=\{1\}\{2\}$ のときは 3 クラスである．ただしもしジョブの資源状態が違ってもノード内での振舞いが違わなければ，ジョブクラス数はより少なくなる．

(2) 資源待ち行列の滞在ジョブ数，ビジー率

状態 $\gamma(k,n_0,n_1,n_2,n_3)$ の下での資源待ち行列中の滞在ジョブ数は，表 1 に示すとおりである．またこの状態下での資源待ち行列のビジー率は，資源待ち行列中にジョブがいれば「1 (=100%)」，いなければ「0」である．3.5 節で述べるように，これを $P(\gamma(k,n_0,n_1,n_2,n_3))$ で期待値をとれば，資源待ち行列にジョブが存在する確率の合計，すなわちその資源待ち行列のビジー率が求められる．

(3) 資源待ち行列のスループット

資源待ち行列のスループットはそれへの到着スループットを用いて求める．ただし次の記号を用いる．

$\lambda_m^r(\gamma(k,n_0,n_1,n_2,n_3))$： 状態 $\gamma(k,n_0,n_1,n_2,n_3)$ の下でノード m の資源状態 r のジョブによるスループット（この値は（1）で導いた積形解から求めることができる）

(a) 資源待ち行列 1 への到着スループット

資源待ち行列 1 へのジョブの到着は，資源 1 がすでに占有されている状態（すなわち $k=\{1\},\{12\},\{1\}\{2\}$ および $k=\{2\}$ で $n_3\geq 1$ のとき）で資源状態 $r=\phi,\{2\}$ のジョブが資源 1 を要求したときに起こる．したがって，たとえば状態 $\gamma(\{1\},n_0,n_1,n_2,n_3)$, $n_1\geq 1$ のときの資源待ち行列 1 への到着スループットは，資源状態 0 のジョブが資源 1 を要求する（すなわち資源状態 1 または 3 に遷移する）スループット $(p^{01}+p^{03})\cdot\lambda_0^0(\gamma(\{1\},n_0,n_1,n_2,n_3))$ である．

(b) 資源待ち行列 2 への到着スループット

資源待ち行列 2 へのジョブの到着も，資源待ち行列 1 と同様に，資源 2 が占有されている $k=\{2\},\{12\},\{1\}\{2\}$ で資源状態 $r=\phi,\{1\}$ のジョブが資源 2 を要求したときに起こる．これに加えて資源待ち行列 2 については，資源待ち行列 1 にジョブが存在していて $k=\{12\}$ または $\{1\}\{2\}$ のときに，資源 1 が解放されて資源状態 3 のジョブがこれを獲得すると，このジョブは資源待ち行列 2 に入ってくる．これによっても資源待ち行列 2 への到着スループットが発生する．資源 1 の解放時に資源待ち行列 1 から資源状態 1 と 3 のどちらのジョブが出てくるかは，3.1 節の仮定 B により資源待ち行列中のそれぞれのジョブ数に比例する．したがって，たとえば状態 $\gamma(\{1\}\{2\},n_0,n_1,n_2,n_3)$, $n_1,n_2\geq 1$ で資源状態 1 のジョブが資源 1 を解放したために資源状態 3 のジョブが資源待ち行列 1 から 2 に移動することによる資源待ち行列 2 への到着スループットは，資源状態 1 のジョブが資源 1 を解放するスループット $p^{10}\cdot\lambda_0^1(\gamma(\{1\}\{2\},n_0,n_1,n_2,n_3))$ を資源待ち行列 1 にいる資源状態 1 のジョブ数 n_1-1 と資源状態 3 のジョブ数 n_3 で比例配分した

$$\frac{n_3}{n_1-1+n_3}\cdot p^{10}\cdot\lambda_0^1(\gamma(\{1\}\{2\},n_0,n_1,n_2,n_3))$$

となる．

3.5 性能値の近似値の算出

3.4 節で求めた状態 $\gamma(k,n_0,n_1,n_2,n_3)$ の下でのノードおよび資源待ち行列のビジー率，滞在ジョブ数，スループットを，3.3 節で求めた定常分布 $P(\gamma(k,n_0,n_1,n_2,n_3))$ で期待値をとって各性能値の近似値とする．さらに得られた平均滞在ジョブ数と平

均スループットから，リトルの公式によりノードや資源待ち行列の平均滞在時間を求める．

4. 数値実験

4.1 パラメータ

近似モデルによる状態数の削減効果と近似精度を調べるために，更新をともなうファイル資源について，木下，高橋[12]と同一の設定で数値実験を行った．設定したパラメータは，次のとおりである☆．

(1) ジョブ数：$N = 2, 4, 6, 8$

(2) I/O ノード数：$M = 2$

(3) CPU ノードでのサービス率：$\mu_0 = 2.0$

(4) I/O ノード 1，2でのサービス率：

$$\mu_1 = \mu_2 = 1.0$$

(5) 資源数：$F = 2$

(6) 資源状態の遷移確率

オンライン実時間システムにおいて，複数のジョブ（トランザクション）が 2 個のデータ項目を含むレコードからなるファイルを更新する場合を考え，項目 1，2 のいずれの更新も 1 つの資源で排他制御する場合と，項目 1，2 の更新を 2 個の資源 1，2 で個別に排他制御する場合を考える．資源が 1 個の場合の資源状態の遷移確率行列を $(\overline{p}^{r_1 r_2})_{r_1=0,1}^{r_2=0,1}$，資源が 2 個の場合のそれを $(p^{r_1 r_2})_{r_1=0,\ldots,3}^{r_2=0,\ldots,3}$ と書く．

(a) 資源が 1 個の場合

資源が 1 個の場合の遷移行列を，次のようにおく．

$$\begin{pmatrix} \overline{p}^{00} & \overline{p}^{01} \\ \overline{p}^{10} & \overline{p}^{11} \end{pmatrix} = \begin{pmatrix} 1-\overline{p}^{01} & \overline{p}^{01} \\ 0.5 & 0.5 \end{pmatrix}$$

ただし $\overline{p}^{01} = 0,\ 0.02,\ 0.08,\ 0.14$ とする．

この資源要求確率 $\overline{p}^{01}$ は，現実のオンライン実時間システムにおいて排他制御をともなう入出力が全入出力の十数回に 1 回（約 8%）であることに基づいている．また $\overline{p}^{11} = 0.5$ としているので，ジョブは資源を占有したまま平均 2 回の CPU 処理を行った後に資源を解放する．

(b) 資源が 2 個の場合

資源が 2 個の場合の資源状態の遷移確率 $p^{r_1 r_2}$ $(r_1, r_2 = 0, 1, 2, 3)$ については，資源が 1 個の場合と比較しやすいように次の条件（i）を仮定する．

（i）資源が 2 個の場合の資源 1，2 または 1 と 2 の両方を要求/保持/解放する振舞いと，資源が 1 個の場合にそれを要求/保持/解放する振舞いが同等であるとする．すなわち次の関係が成り立つとする．

・資源の要求について：$\overline{p}^{01} = p^{01} + p^{02} + p^{03}$

・ 〃 保持 〃 ：$\overline{p}^{11} = p^{11} = p^{22} = p^{23}$

・資源の解放について：$\overline{p}^{10} = p^{10} = p^{20} = p^{30}$

さらに簡単のため次の条件を仮定する．

（ii）資源 1 を要求することと資源 2 を要求することは確率的に独立とする．すなわち資源 1，2 を要求する頻度をそれぞれ s_1，s_2，資源 1 と 2 を両方とも要求する頻度を s_{12}（$s_1 = p^{01} + p^{03}$, $s_2 = p^{02} + p^{03}$, $s_{12} = p^{03}$ である）とするとき，$s_{12} = s_1 \cdot s_2$ とする．

（iii）項目 1 と 2 は同じ頻度で更新されるものとする．すなわち $s_1 = s_2$ とする．

以上により資源状態の遷移行列は次のようになる．

$$\begin{pmatrix} p^{00} & p^{01} & p^{02} & p^{03} \\ p^{10} & p^{11} & p^{12} & p^{13} \\ p^{20} & p^{21} & p^{22} & p^{23} \\ p^{30} & p^{31} & p^{23} & p^{33} \end{pmatrix}$$

$$= \begin{pmatrix} \overline{p}^{00} & \overline{q} & \overline{q} & (1-\sqrt{\overline{p}^{00}})^2 \\ 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{pmatrix}$$

ただし $\overline{q} = \sqrt{\overline{p}^{00}}(1-\sqrt{\overline{p}^{00}})$ である．資源が 1 個の場合と同様に，$\overline{p}^{00} = 1 - \overline{p}^{01}$（$\overline{p}^{01} = 0,\ 0.02,\ 0.08,\ 0.14$）とする．

(7) CPU ノードからの推移確率

CPU ノードから次のノードへの遷移は資源が 1 個の場合も 2 個の場合も，またジョブの資源状態に関係なく同一とする．すなわち資源が 1 個の場合の資源状態 r のジョブが CPU ノードからノード m へ移動する遷移確率を $\overline{p}_m^r$ $(r = 0, 1)$，資源が 2 個の場合のそれを $p_m^r (r = 0, 1, 2, 3)$ とするとき，次のようにおく．

$$(\overline{p}_0^0, \overline{p}_1^0, \overline{p}_2^0) = (p_0^0, p_1^0, p_2^0) = (0.2, 0.4, 0.4)$$
$$(\overline{p}_0^1, \overline{p}_1^1, \overline{p}_2^1) = (p_0^r, p_1^r, p_2^r) = (0.0, 0.4, 0.6) \quad (r = 1, 2, 3)$$

4.2 状態数の削減効果

近似モデルにおいても，縮約された状態 γ の定常分布を求めるために平衡方程式を数値的に解く必要がある．しかし**表 2** に示すように，その状態数はもとの待ち行列網の状態 δ に比べて 2 ～ 3 桁程度減少してい

☆ 文献 12) では 1 個の資源を 2 個に分割することの性能への効果を調べることを目的に，それに適したパラメータを設定した．本論文の数値実験は近似モデルの精度検証が目的であり特に資源の分割はイメージしていないが，比較のために文献 12) と同一のパラメータを用いている．したがって資源の分割を扱っていると考えても差支えない．

表 2 待ち行列網の状態数（$M = 2$）
Table 2 Number of states ($M = 2$).

ジョブ数	資源数 $F = 1$		$F = 2$	
（N）	δ	γ	δ	γ
2	21	3	78	12
4	120	5	1,383	55
6	406	7	13,684	154
8	1,035	9	100,077	333

$F = 1$ のときの γ を状態とするマルコフ連鎖は，単純な生成死滅過程である．

る．状態 γ は資源を要求しているジョブ数と占有しているジョブ数のみで決定され，各ジョブが網内のどのノード/資源待ち行列でどの順序で並んでいるかに依存しない．一方，状態 δ はこれらをすべて異なる状態として区別する．このためノード数，ジョブ数，資源数のいずれが増加しても δ の状態数の増加率は γ の増加率を大きく上回る．数値計算は所要の CPU 計算量およびメモリ量から考えて状態数が 5 ～ 10 万個程度が限度と考えられる．表 2 によると $P(\delta)$ を計算することは $F = 2$ の場合で $N = 6 \sim 7$ 程度が実用上の限度と見られ，$N \geq 8$ では困難と考えられる．一方 $P(\gamma)$ は表 2 から推定して，$F = 2$ の場合で $N = 40$ 程度まで，$F = 3$ の場合でも $N = 10 \sim 15$ 程度まで計算可能と考えられる．

このように本近似モデルは，もとの待ち行列網の解析に比べてはるかに簡便な手法を提供している．一方，この種の性能評価に広く用いられる手法に計算機シミュレーションがある．これはランダムに（または確率分布に従って）事象を発生させて模擬的に対象システムの定常状態を生成して，性能値の統計値を算出する方法である．その際，定常状態を生成するために事象を多数発生させる必要があり，待ち行列網による解析的手法に比べてより多くの計算時間が必要とされている．本節の数値実験と同じパラメータのケースのシミュレーションをパソコンで実行したところ，本近似モデルの計算時間はシミュレーションのそれの数分の一から数十分の一程度であった．シミュレーションの計算時間は必要とする精度や使用する計算機環境によって大きく変動するので一概に比較は難しいが，上記の結果から，実用上の範囲において本近似モデルは計算時間の点でシミュレーションより有利と考えられる．

4.3 近似モデルの精度検証

表 3〜8 に数値解析結果を示す．この表で「近似解」は本論文で提案した近似モデルによる値で，「厳密解」は木下，高橋[12]の方法による値である．これらには全般に次のような傾向が見られる．

(1) 資源が 1 個と 2 個の場合では，1 個の場合の方が精度が良い．

(2) 資源要求確率 p^{01} が大きくなると精度が落ちる．

(3) 厳密モデルと近似モデルでは，$p^{01} = 0.08, 0.14$ のときは近似モデルの方が負荷が軽いときの性能値（性能がより良い値）を示す．

3.1 節で述べたように，本近似モデルでは仮定 A において「(a) 一般に状態 γ はその直前の状態に依存し，(b) γ の継続時間内では必ずしも定常状態に達しないにもかかわらず，$P(\overline{\delta}_d|\gamma)$ を γ に制限した待ち行列網の定常分布で近似する」としている．この (a) はある状態とその直前の状態が確率的に独立と考えていることになるが，これは網の構造やジョブの振舞いを単純化することにつながり，このため性能値は負荷が軽い方の値にシフトすると考えられる．また (b) の状態 γ の継続時間は，資源数や資源要求の頻度 p^{01} に依存する．本数値実験では，オンライン実時間システムを想定して排他制御を伴う入出力の頻度を十数回に 1 回（約 8%）程度と設定している．上記の (1) のように資源数が増大したり (2) のように p^{01} が大きくなったりして資源要求の頻度が高まると，γ の持続時間が短くなって定常状態からよりかけ離れ，また γ の直前の状態のバラエティも増すため，近似精度が落ちると考えられる．このように仮定 A により近似モデルの性能値がより小さく（性能がより良く）計算されるが，上記の (3) は資源要求確率が大きいとその傾向がより強く出てくるためと考えられる．一方 (3) の $p^{01} = 0.02$ のときは，近似モデルが必ずしも負荷の軽いときの値になっていない．これは資源要求の頻度が小さいケースでは，仮定 A による負荷の軽減効果が強く現れないためと考えられる．

次に資源待ち行列について考えると，

(1) CPU ビジー率（表 3）の近似精度が資源待ち行列ビジー率（表 4）より概して良く，

(2) 平均応答時間（表 8）の近似精度が 1 回の資源待ち行列平均滞在時間（表 6）より概して良い．

ここで「平均応答時間 = CPU，I/O ノードでの平均滞在時間+ 資源待ち行列での平均滞在時間」だから，(2) は CPU，I/O ノードでの平均滞在時間が資源待ち行列でのそれよりも近似精度が良いことを意味する．すなわちビジー率，平均滞在時間のいずれについても「CPU，I/O ノードの性能値」の方が「資源待ち行列の性能値」よりも近似精度が良いことが分かる．この理由は，3.1 節で述べた 2 つの仮定のうち「積形解の仮定（仮定 A）」は CPU，I/O ノードの振舞いに関係し，「資源待ち行列から出てくるジョブの頻度に関する

表 3 CPU ビジー率
Table 3 CPU busy ratio.

ジョブ数 (N)	資源要求確率 ($\overline{p}^{01}$)	資源数 $F=1$ 厳密解	近似解	誤差率
4	0.02	0.761	0.762	0.13 %
	0.08	0.740	0.739	−0.14
	0.14	0.701	0.702	0.14
8	0.02	0.901	0.903	0.22
	0.08	0.843	0.844	0.12
	0.14	0.747	0.758	1.47
		$F=2$		
4	0.02	0.761	0.762	0.13 %
	0.08	0.752	0.754	0.27
	0.14	0.734	0.739	0.68
8	0.02	0.898	0.902	0.45
	0.08	0.878	0.890	1.37
	0.14	0.836	0.840	0.48

表 4 資源待ち行列ビジー率
Table 4 Resource waiting queue busy ratio.

ジョブ数 (N)	資源要求確率 ($\overline{p}^{01}$)	資源数 $F=1$ 厳密解	近似解	誤差率
4	0.02	0.0167	0.0189	13.17 %
	0.08	0.210	0.212	0.94
	0.14	0.468	0.443	−5.34
8	0.02	0.0807	0.0903	11.90
	0.08	0.759	0.715	5.80
	0.14	0.979	0.948	3.17
		$F=2$		
4	0.02	0.00415	0.00467	12.53 %
	0.08	0.0543	0.0547	0.74
	0.14	0.141	0.129	−8.51
8	0.02	0.0215	0.0211	−1.86
	0.08	0.254	0.216	−14.96
	0.14	0.551	0.441	−19.96

$F=2$ のときは，資源待ち行列 1 のビジー率

表 5 資源待ちの平均ジョブ数
Table 5 Number of jobs in resource waiting queue.

ジョブ数 (N)	資源要求確率 ($\overline{p}^{01}$)	資源数 $F=1$ 厳密解	近似解	誤差率
4	0.02	0.0177	0.0184	3.95 %
	0.08	0.262	0.274	4.58
	0.14	0.675	0.658	2.52
8	0.02	0.0990	0.1010	2.02
	0.08	1.874	1.805	−3.68
	0.14	3.812	3.553	−6.79
		$F=2$		
4	0.02	0.00898	0.00968	7.80 %
	0.08	0.119	0.124	4.20
	0.14	0.320	0.311	−2.81
8	0.02	0.0411	0.0475	15.57
	0.08	0.729	0.608	−16.60
	0.14	1.973	1.633	−17.23

$F=2$ のときは，資源待ち行列 1 と 2 の滞在ジョブ数の合計

表 6 1 回の資源待ち行列平均滞在時間
Table 6 Mean resident time in resource waiting queue.

ジョブ数 (N)	資源要求確率 ($\overline{p}^{01}$)	資源数 $F=1$ 厳密解	近似解	誤差率
4	0.02	4.505	4.053	−10.03 %
	0.08	5.370	4.888	8.98
	0.14	6.186	5.632	8.96
8	0.02	9.305	10.065	8.17
	0.08	17.961	17.522	−2.44
	0.14	23.478	21.733	−7.43
		$F=2$		
4	0.02	3.354	3.881	15.71 %
	0.08	4.699	4.184	−10.96
	0.14	5.004	4.345	−13.17
8	0.02	7.284	8.523	17.01
	0.08	11.352	10.282	−9.43
	0.14	14.922	12.014	−19.49

$F=2$ のときは，資源待ち行列 1 の平均滞在時間

表 7 ジョブのスループット
Table 7 Job throughput.

ジョブ数 (N)	資源要求確率 ($\overline{p}^{01}$)	資源数 $F=1$ 厳密解	近似解	誤差率
4	0.02	0.304	0.304	0 %
	0.08	0.296	0.296	0
	0.14	0.280	0.281	0.36
8	0.02	0.360	0.360	0
	0.08	0.337	0.338	0.30
	0.14	0.299	0.303	1.55
		$F=2$		
4	0.02	0.304	0.305	0.33 %
	0.08	0.300	0.306	2.00
	0.14	0.294	0.306	4.08
8	0.02	0.360	0.362	0.56
	0.08	0.351	0.367	4.56
	0.14	0.335	0.368	9.85

表 8 ジョブの平均応答時間
Table 8 Mean job response time.

ジョブ数 (N)	資源要求確率 ($\overline{p}^{01}$)	資源数 $F=1$ 厳密解	近似解	誤差率
4	0.02	13.142	13.145	0.02 %
	0.08	13.514	13.533	0.14
	0.14	14.269	14.245	−0.17
8	0.02	22.200	22.212	0.05
	0.08	23.733	23.692	−0.17
	0.14	26.775	26.369	−1.52
		$F=2$		
4	0.02	13.137	13.112	−0.19 %
	0.08	13.335	13.069	−1.99
	0.14	13.610	13.060	−4.04
8	0.02	22.236	22.101	−0.61
	0.08	22.793	21.798	−4.37
	0.14	23.861	21.736	−8.91

仮定（仮定 B）」は資源待ち行列の振舞いに関係するが，このうち仮定 B の方が仮定 A よりも粗い仮定になっているためと考えられる．

この数値解析例では，資源数が 2 個の場合の資源待ち行列についての近似モデルの性能値の誤差率が大きいところで 20%近くになるケースがあり，資源数や資源要求確率が大きい場合の資源待ち行列の振舞いを解析する際は，性能値が負荷の軽い方に 20%程度ずれて計算されうることを考慮しておく必要がある．一方，CPU ビジー率，ジョブのスループットや平均応答時間といった資源待ち行列以外の性能値の誤差率は（この数値解析例では）最大で 9%程度であり，これは実用上有用な範囲内といえる．そして近似モデルの状態数は厳密モデルよりもはるかに少なく，解析可能な範囲を広げることができる．

5. むすび

木下，高橋[12]で逐次アクセス資源のある計算機システムの性能上の特性を待ち行列論により求める一方法を提案したが，この方法で問題となる数値計算の計算量を削減するための近似モデルを導入し，数値実験により計算量の削減効果と性能値の近似精度を確認した．

近似は状態縮約の考え方を応用し，資源の要求/解放がない間（1 つの要求/解放から次の要求/解放までの間）の網の状態を 1 つの状態に縮約し，縮約された状態に制限した網は通常の待ち行列網と見なしてその積形解で近似し，資源の要求/解放の振舞いはこれを記述するマルコフ連鎖を構成してその平衡方程式を解いて縮約された状態の定常分布を求める．そして積形解により求めた性能値を定常分布で期待値をとって近似値とする方法である．この方法では縮約された状態についてのマルコフ連鎖の平衡方程式を数値的に解く必要があるが，文献 12) で扱ったマルコフ連鎖に比べて状態数がはるかに少なく，計算可能な範囲を大幅に拡大できる．そしてこの近似モデルの縮約された状態下での性能値の求め方，および縮約された状態の状態遷移とそれを記述するマルコフ連鎖の平衡方程式の構成法を示した．

数値実験は，更新をともなうファイル資源について文献 12) と同じ設定で行った．その結果，資源数または資源要求確率が大きい場合の資源待ち行列についての性能値の精度は落ちるものの，CPU ビジー率，ジョブのスループットや平均応答時間といった網全体の性能値は実用上の範囲内の近似精度であることを確認した．

今後は，提案した近似モデルの各ノードでのサービス分布が，指数分布より一般のコックス分布☆ の場合について検証していきたい．また本論文で扱った逐次アクセス資源のある待ち行列網について，各ジョブが資源待ち行列につながることの応答時間への影響をさらに詳細に解析していきたい．

☆ ラプラス–スティルチェス変換が有理式であるような確率分布のことで，指数分布を含む．

参考文献

1) Baskett, F., Chandy, K.M., Muntz R.R. and Palacious, F.G.: Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, *J. ACM*, Vol.22, No.2, pp.248–260 (1975).
2) Chandy, K.M., Herzog, U. and Woo, L.: Parametric Analysis of Queueing Networks, *IBM J. Res. Dev.*, Vol.19, No.1, pp.36–42 (1975).
3) Chandy, K.M., Herzog, U. and Woo, L.: Approximate Analysis of General Queueing Networks, *IBM J. Res. Dev.*, Vol.19, No.1, pp.43–49 (1975).
4) Kino, I.: Two-layer Queueing Networks, *J. ORSJ*, Vol.40, No.2, pp.163–185 (1997).
5) Kobayashi, H.: *Modeling and Analysis*, Addison-Wesley Publishing, Massachusetts (1978).
6) Kurasugi, T. and Kino, I.: Approximation Method for Two-layer Queueing Models, *Performance Evaluation* 36–37, pp.55–70 (1999).
7) Rolia, J.A. and Sevcik, K.C.: The Method of Layers, *IEEE Trans. Softw. Eng.*, Vol.21, No.8, pp.689–700 (1995).
8) Sauer, C.H.: Approximate Solution of Queueing Networks with Simultaneous Resource Possession, *IBM J. Res. Dev.*, Vol.25, No.6, pp.894–903 (1981).
9) 池原　悟：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，情報処理学会論文誌，Vol.20, No.2, pp.105–112 (1979).
10) 星合隆成：蜜結合マルチプロセッサシステムにおける排他制御方式とその性能解析，電子通信学会論文誌（D-I），Vol.J78-D-I, No.2, pp.248–259 (1995).
11) 木下俊之，高橋幸雄：資源要求のある待ち行列網のモデル化の一提案，第 51 回情報処理学会全国大会論文集（4），5L-2, pp.3–4 (1995).
12) 木下俊之，高橋幸雄：逐次アクセス資源のある計算機システムの待ち行列網によるモデル化と評価法，電子通信学会論文誌（D-I），Vol.J82-D-I, No.6, pp.701–710 (1999).
13) 蔵杉俊康，紀　一誠：2 階層待ち行列モデルの積形式近似とその精度検証，情報通信ネットワーク

の新しい性能評価法に関する総合研究（シンポジウム），京都，pp.332–341 (1998)

(平成 12 年 1 月 5 日受付)
(平成 12 年 2 月 21 日再受付)
(平成 12 年 10 月 29 日再々受付)
(平成 13 年 3 月 23 日採録)

木下 俊之（正会員）

昭和 52 年東京大学大学院理学系研究科数学専攻修士課程修了．同年（株）日立製作所入社．同社システム開発研究所にて計算機性能評価，待ち行列理論，オンライン制御プログラム，オペレーティングシステム，計算機アーキテクチャの研究に従事．平成 11 年より同研究所主管研究員．技術士（情報処理部門）．理学博士．情報処理学会システムソフトウェアとオペレーティングシステム研究会幹事．電子情報通信学会，ACM 各会員．

高橋 幸雄

和 47 年東京工業大学大学院理工学研究科応用物理学専攻博士課程修了．待ち行列理論等の研究に従事．平成元年より東京工業大学大学院情報理工学研究科教授．理学博士．J. of OR Society of Japan 編集顧問，Performance Evaluation 編集委員．日本 OR 学会，日本統計学会，INFORMS，電子情報通信学会各会員．