

## 排他資源のある計算機システムの待ち行列網による 近似性能評価法の提案と評価

羽 角 友 春<sup>†1</sup> 高 秀 梅<sup>†1</sup> 木 下 俊 之<sup>†1</sup>

コンピュータシステムでは、ひとつのジョブが更新を伴ってファイルを使用中は、ファイルの一致性を保証するために、他のジョブがそのファイルにアクセスすることは禁止される。このようなファイルを排他使用資源（または単に資源）と呼ぶ。排他使用資源へのアクセスのぶつかりは、システムの性能に重大な影響を与える。排他使用資源のある待ち行列網をマルコフ連鎖を用いてモデル化し、その平衡方程式を数的に解く方法が提案されている。しかしこの方法は、システムの状態数が増加したときに平衡方程式を解くことが難しくなるという問題点がある。

本論文では、この数値計算の困難を削減する近似手法を提案する。この近似手法では、システムを2つの部分に分けて考える。ひとつはCPUやI/Oノードでのジョブの振る舞いであり、他のひとつはジョブが資源を要求/解放する振る舞いである。前者は資源のない従来のセントラルサーバモデルの積形解によって近似し、後者はジョブが資源を要求/解放する振る舞いを記述するマルコフ連鎖を構成して解析する。このマルコフ連鎖はシステムの一部しか表現していないため、システム全体を記述するマルコフ連鎖より状態数は少なく、数値計算の手間を大きく削減できる。数値実験により、近似モデルは実用に十分な精度を持つことを確認した。

### An Approximation Queuing Network Technique for Evaluating Performance of Computer Systems with Resource Requirements

TOMO HARU HASUMI, XIUMEI GAO  
and TOSHIYUKI KINOSHITA

In computer systems, while one job uses an updatable file, other jobs are prohibited from accessing that file in order to ensure its consistency. We call such a file an exclusively used resource (or simply, a resource). These conflicts of accessing an exclusively used resource significantly affect the performance of the system. A method for analyzing the queuing network model with resource requirements that describes the network with a Markov chain and solves

its equilibrium equations numerically was proposed. However, using this technique makes it difficult to solve the equations when the states of the system increase.

An approximation method that can reduce the difficulty of numerical calculation is proposed. In the approximation model, the system is analyzed by dividing it into two parts. One is the behavior of a job in CPU and I/O nodes and the other is the behavior of a job that requests or releases the resource. The former part is approximated by the product form solution of the ordinary queuing network model, and the latter is analyzed by a Markov chain that describes the behavior of a job that requests or releases the resource. Since this Markov chain represents only one part, the number of states becomes smaller and the computational burden is also reduced. Numerical experiments indicate that the approximation model has enough accuracy for practical use.

#### 1. ま え が き

コンピュータシステムの性能は、CPUやI/O装置といったハードウェアだけでなく、プログラムやデータファイルといったソフトウェアの振る舞いに依存している。多くのオンラインシステムでは、ジョブまたはトランザクションは、通常ファイル上のデータをメモリに読み出し、更新し、ファイルの同じエントリに書き戻す処理が行われる。このデータの読み出しから書き込みの間は、ファイルの一致性を保証するために、他のジョブからのアクセスは禁止される。この間、当該ファイルはひとつのジョブから排他的に使用される。この意味でこのファイルは「排他資源」と呼ぶことができる（または単に「資源」と呼ぶ）。このような排他資源へのアクセスのぶつかりは、コンピュータシステムの性能に大きな影響を及ぼすにも関わらず、一般の待ち行列モデルでは解析が困難であった。いくつかの研究は、この困難の削減を試みている<sup>3)4)</sup>。木下<sup>5)</sup>は、資源要求を伴う待ち行列モデルをマルコフ連鎖を用いて記述し、その平衡方程式を数値的に解いて解析する方法を提案した。この方法は、数値解を可能な限り正確に求められる反面、数値計算に用いるコンピュータの性能やメモリ量の制約を受けるという問題がある。システム内にジョブや資源が多く存在すると、マルコフ連鎖の状態数が膨大となり、平衡方程式の係数行列がメモリ上に展開できなくなるといった数値計算上の困難が生ずる。木下<sup>6)</sup>では、ジョブ数や資源数が増大してもシステムの状態数を削減する近似法を提案した。

<sup>†1</sup> 東京工科大学コンピュータサイエンス学部  
Tokyo University of Technology

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手法を提案する。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。すなわち、連続する資源要求/解放の間の網の状態をひとつの状態に縮約する。この縮約された状態の定常確率は、排他資源のない従来の待ち行列網の積形解で近似する。そしてジョブが資源を要求/解放する振る舞いは、それを記述するマルコフ連鎖により解析する。縮約によってシステムの状態数は劇的に削減されるので、数値計算の困難も大きく削減される。

本論文の構成は、2 で排他資源のある待ち行列網の構成と状態を定義し、3 で提案する近似モデルについて説明する。4 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 でむすびを述べる。

## 2. モデルの記述

### 2.1 排他資源のあるセントラルサーバモデル

排他資源のある待ち行列網モデルは、待ち行列網の一つであるセントラルサーバモデル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)。

網は、単一の CPU ノード (ノード番号  $m = 0$ ) と複数の I/O ノード (ノード番号  $m = 1, 2, \dots, M$ )、および資源待ち行列 ( $F$  を資源数とする時、 $m = M+1, M+2, \dots, M+F$  で参照される) で構成される。網の中には一定数  $N$  個のジョブと (ジョブ番号  $n = 1, 2, \dots, N$ )、 $F$  個の資源が存在する。資源  $S_f$  (資源番号  $f = 1, 2, \dots, F$ ) は資源待ち行列  $M+f$  に対応するので、資源数と資源待ち行列数は等しい。ノード  $m$  ( $= 0, 1, \dots, M$ , つまり CPU と I/O ノード) でのサービス率は  $\mu_m$  で、各ノードでは互いに独立で指数分布に従う。すべてのノードおよび資源待ち行列で、ジョブは FCFS (First Come First Served) 規律でスケジュールされる。

ジョブは CPU サービスの終了時に資源を要求したり解放したりする。ジョブが資源を要求したときにその資源が占有されていると、ジョブは対応する資源待ち行列に入る。資源が解放されると (もし存在すれば) 直ちに資源待ち行列の先頭のジョブが資源を獲得する。こうしてジョブが要求したすべての資源を獲得すると、次のノードに進む。

ジョブの資源状態は、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として定義される (ジョブが既にその資源を獲得しているか否かには関わらない)。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

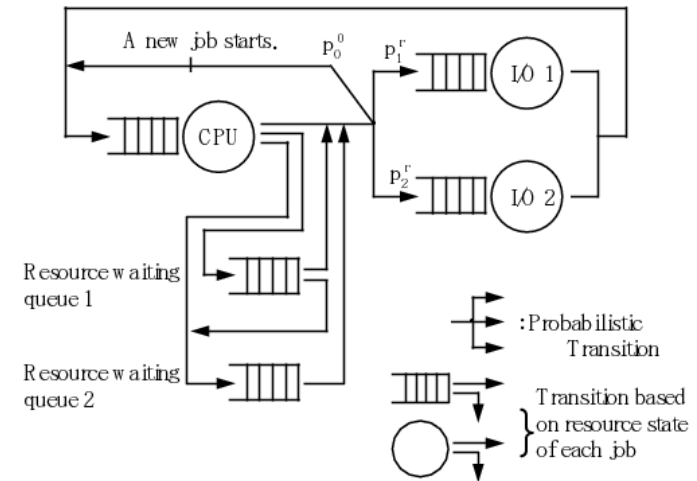


図 1 排他資源のあるセントラルサーバモデルのイメージ図 (資源が 2 個の場合)  
Fig. 1 Image diagram of central server model with 2 exclusively-used resources.

源の集合として  $\{S_{f_1}, S_{f_2}, \dots, S_{f_k}\} (0 \leq k \leq F)$  と表される。  $R$  は資源状態数であり、資源状態を添え字  $r$  または  $r_i (= 0, 1, \dots, R-1)$  で参照する。資源状態  $r = 0$  は資源を必要としない状態 (すなわち  $k = 0$ ) を表す。例えば資源が 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
- $r = 1$ :  $\{1\}$  資源 1 のみを要求
- $r = 2$ :  $\{2\}$  資源 2 のみを要求
- $r = 3$ :  $\{1, 2\}$  資源 1 と 2 の両方を要求

の 4 通りが考えられる。

ジョブの資源状態は CPU のサービス終了時に変更される。  $p^{r_1 r_2}$  を資源状態  $r_1$  から  $r_2$  への遷移確率とし、遷移行列を  $(p^{r_1 r_2})_{r_1, r_2=0,1,\dots,R-1}$  とする。

デッドロックを回避するために、資源を獲得する順序を決める。2 個以上の資源を要求するときは、ジョブはこの順序にしたがって資源を要求しなければならない。ここでは、資源番号  $f$  がより小さい資源の資源獲得順序がより先であるとする。もしジョブが獲得順序が後の資源を持っているときに獲得順序が先の資源を要求する場合は、獲得順序が後のものを解放し、改めて両方の資源を要求しなければならない。上の例では、資源  $S_1$  の方が資源

$S_2$  より先であるから、資源状態  $r = 2$  から 3 に遷移するには、ジョブは資源  $S_2$  を解放し、資源  $S_1$  と  $S_2$  を改めて要求しなければならない。

ジョブが CPU のサービス終了時に資源を要求しなかったり、要求したときに資源が空いて直ぐに獲得できたときは、いずれかの I/O ノードまたは CPU ノードに進む。CPU ノードから次にどのノードに進むかは、確率的に選択される。CPU ノードから次に進むノードの選択確率はジョブの資源状態に依存し、資源状態が  $r$  のジョブは遷移確率  $p_m^r (m = 0, 1, 2, \dots, M)$  でノード  $m$  に進む。

## 2.2 ジョブのライフタイム

CPU ノードでのサービス終了後、ジョブは I/O ノードや CPU ノードから CPU ノードに戻ってくる。このうち CPU から CPU に直接に遷移したとき、ジョブはその生涯を終えて新しいジョブに生まれ変わったと考える。したがってジョブのライフタイムは CPU → CPU の遷移から次の CPU → CPU の遷移までの間として定義される。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、 $p_0^r = 0 (r > 0)$  と仮定する。

## 2.3 待ち行列網の状態

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  $k_m^n$  は対応するジョブの資源状態を表す。すなわち  $k_m^n = r$  ならばノード  $m$  または資源待ち行列  $f (m = M + f)$  の第  $n$  番目のジョブが資源状態  $r$  であることを表す。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  $S_f$  を含む資源状態のジョブは高々一つである (資源  $S_f$  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  $S_f$  を含む資源状態のジョブは複数存在し得る (資源  $S_f$  を要求したが獲得できずいるジョブ)。

可能なすべての  $\delta$  の集合を  $\Delta$  とする。排他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

コンピュータシステムの性能評価のための各ノードでのビジー率、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。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  $\delta$  の状態数は 4.2 の表 1 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、大規模システムではこの計算を行うことは難しい。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  $F = 2$  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 ノード中にあるジョブを指し、これと「資源待ち行列内のジョブ」を区別して言い表す。

### 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。したがってこの間の網の振る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通常の網の積形解で表されると考えられる。我々の近似モデルにこの考えを利用する。

近似モデルを構成するために、網内のジョブのふるまいを「引き続く資源要求/解放の間 (資源の要求/解放がない間) の CPU ノードと I/O ノードでのふるまい (各ノードでサービスを受けたり、ノード間の移動したりするなど)」と「資源の要求/解放のふるまい」に分けて考える。そして前者は排他資源のない通常の待ち行列網の積形解で近似し、後者は資源の要求/解放のふるま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 CPU および I/O ノード中にある各資源状態のジョブ数が等しいような  $\delta$  を縮約 (Aggregate) して新しい状態  $\eta$  とする。一方、資源待ち行列については縮約せず、元のままとする。状態  $\delta$  の CPU および I/O ノード中にある資源状態  $r$  のジョブ数を  $N_{Node}^r (= \sum_{m=0}^M N_m^r, r = 0, \dots, 3)$  とすると、新しい状態は  $\eta = (N_{Node}^0, N_{Node}^1, N_{Node}^2, N_{Node}^3; k_{M+1}^1 \cdots k_{M+1}^{N_{M+1}}; \cdots; k_{M+F}^1 \cdots k_{M+F}^{N_{M+F}})$  と表される。可能なすべての  $\eta$  の集合を

$E$  とし,  $\eta$  を状態とするマルコフ連鎖  $\eta(t)$  を考える.

例えば状態  $\eta = (2, 1, 1, 0; 131;)$  は, モデル内に存在するジョブは 7 個で, CPU ノードと I/O ノードに資源を要求していないジョブ (資源状態  $r = 0$ ) が 2 個, 資源 1 を保持しているジョブ ( $r = 1$ ) が 1 個と資源 2 を保持しているジョブ ( $r = 2$ ) が 1 個ずつの合計 4 個存在し, 資源待ち行列 1 に資源 1 を要求しているジョブ 2 個 ( $r = 1$ ) と資源 1 と 2 を同時に要求しているジョブ 1 個 ( $r = 3$ ) が, 131 の順で資源 1 を待っている状態を表している. そして資源 1 を保持しているジョブが資源 1 を解放すると, そのジョブの資源状態は 0 になり, 資源待ち行列 1 の先頭のジョブが資源 1 を確保して IO ノードに推移するので, 状態は  $(3, 1, 1, 0; 31;)$  に変化する. この  $\eta$  についてのマルコフ連鎖  $\eta(t)$  の平衡方程式を解くことにより, 定常分布  $P(\eta)$  を求める.

### 3.2 性能値の算出方法

近似モデルの各ノードのビジー率, 平均滞在ジョブ数やスループットなどの性能値を求めるために, まず状態  $\eta$  の下での条件付きの性能値を, ジョブクラス数が 4 で各クラスのジョブ数が  $N_{Node}^0, N_{Node}^1, N_{Node}^2, N_{Node}^3$  であるような通常のセントラルサーバモデルの積形解から求める. そしてこの条件付き性能値の定常分布  $P(\eta)$  による期待値を, 資源要求付きモデルの性能値の近似値とする. 一方, 資源待ち行列の振る舞いはマルコフ連鎖  $\eta(t)$  ですべて記述されているので, 資源待ち行列の性能値の近似値は定常分布  $P(\eta)$  から直接求めることができる. 以下, 次の記号を用いる ( $r = 0, \dots, 3$ ).

- $\rho_{rm}^{PFS}(\eta), \lambda_{rm}^{PFS}(\eta), L_{rm}^{PFS}(\eta)$ : ジョブクラス  $r$  のジョブ数が  $N_{Node}^r$  であるような通常のセントラルサーバモデルの積形解から求めた, CPU ノード ( $m = 0$ ) および I/O ノード ( $m = 1, 2, \dots, M$ ) のジョブクラス  $r$  (資源状態  $r$ ) のジョブによるビジー率, スループット, および平均ジョブ数.
- $\bar{\rho}_{rm}, \bar{\lambda}_{rm}, \bar{L}_{rm}$ : すべての状態  $\eta$  を通しての CPU ノード, I/O ノード, 資源待ち行列 ( $m = 0, 1, \dots, M, M+1, M+2$ ) における資源状態  $r$  のジョブによるビジー率, スループット, および平均ジョブ数.
- $\bar{\rho}_m, \bar{\lambda}_m, \bar{L}_m$ : CPU ノード, I/O ノード, 資源待ち行列 ( $m = 0, 1, \dots, M, M+1, M+2$ ) におけるすべてのジョブによるビジー率, スループット, および平均ジョブ数.

#### 3.2.1 ビジー率, 平均ジョブ数

CPU および I/O ノードにおける資源状態  $r$  のジョブによるビジー率, スループット, 平均ジョブ数は, 通常のセントラルサーバモデルの積形解を用いて, 次のように求められる.

$$\bar{\rho}_{rm} = \sum_{\eta \in E} \rho_{rm}^{PFS}(\eta) P(\eta), \quad \bar{L}_{rm} = \sum_{\eta \in E} L_{rm}^{PFS}(\eta) P(\eta).$$

資源待ち行列  $m (= M+1, M+2)$  における資源状態  $r$  のジョブによるビジー率, スループット, 平均ジョブ数は, 定常確率  $P(\eta)$  を用いて, 次のように求められる.

$$\bar{\rho}_{rm} = \sum_{\substack{\eta \in E \\ N_m^r > 0, k_m^1 = r}} P(\eta), \quad \bar{L}_{rm} = \sum_{\eta \in E} N_m^r P(\eta).$$

CPU, I/O ノード, 資源待ち行列  $m (= 0, 1, \dots, M, M+1, M+2)$  におけるすべてのジョブによるビジー率, スループット, 平均ジョブ数は, 次のように求められる.

$$\bar{\rho}_m = \sum_{r=0}^3 \bar{\rho}_{rm}, \quad \bar{L}_m = \sum_{r=0}^3 \bar{L}_{rm}.$$

#### 3.2.2 スループット

CPU および I/O ノードにおける資源状態  $r$  のジョブによるスループットは, 通常のセントラルサーバモデルの積形解を用いて, 次のように求められる.

$$\bar{\lambda}_{rm} = \sum_{\eta \in E} \lambda_{rm}^{PFS}(\eta) P(\eta), \quad \bar{\lambda}_m = \sum_{r=0}^3 \bar{L}_{rm}.$$

資源待ち行列のスループットを求めるのは, もう少し難しい. 資源状態が 0 のジョブだけが CPU ノードで資源を要求し資源待ち行列に入れるので,  $N_{Node}^0 > 0$  のときにのみジョブは資源待ち行列に入ることができる. さらに資源 1 と 2 を要求 (資源状態 3) して資源待ち行列 1 にいるジョブは, 資源 1 が解放されると資源待ち行列 1 から 2 に入り得る. そこで資源待ち行列のスループットは, 次のように求められる.

$$\lambda_{M+1} = \sum_{\substack{N_{Node}^0 > 0 \text{ \& } \\ \{N_{Node}^1 > 0 \text{ or } N_{Node}^3 > 0 \text{ or } \\ (N_{Node}^1 = N_{Node}^3 = 0 \text{ \& } N_{Node}^2 > 0 \\ \text{\& one } k_{M+2}^1, \dots, k_{M+2}^{N_{M+2}} \text{ is } 3)\}}} (p^{01} + p^{03}) \bar{\lambda}_{00} P(\eta),$$

$$\begin{aligned}
 \lambda_{M+2} = & \sum_{\substack{N_{Node}^0 > 0 \& \\ (N_{Node}^2 > 0 \text{ or } N_{Node}^3 > 0)}} p^{02} \bar{\lambda}_{00} P(\eta) \\
 & + \sum_{\substack{N_{Node}^0 > 0 \& \\ N_{Node}^1 = N_{Node}^3 = 0 \& N_{Node}^2 > 0 \\ \& \text{ all } k_{M+2}^1, \dots, k_{M+2}^{M+2} \text{ are 2}}} p^{03} \bar{\lambda}_{00} P(\eta) \\
 & + \sum_{N_{Node}^1 > 0 \& N_{Node}^2 > 0} (p^{12} + p^{13}) \bar{\lambda}_{10} P(\eta) \\
 & + \sum_{\substack{N_{Node}^1 > 0 \& N_{Node}^2 > 0 \\ \& k_{M+1}^1 = 3}} p^{10} \bar{\lambda}_{10} P(\eta) \\
 & + \sum_{N_{Node}^3 > 0 \& k_{M+1}^1 = 3} p^{32} \bar{\lambda}_{30} P(\eta) \\
 & + \sum_{\substack{N_{Node}^3 > 0 \& k_{M+1}^1 = 3 \\ \& N_{M+2} > 0}} p^{30} \bar{\lambda}_{30} P(\eta).
 \end{aligned}$$

### 3.2.3 ジョブの平均応答時間

ジョブの平均応答時間  $\bar{T}$  は、リトルの公式を用いて次のように求められる。

$$\bar{T} = \frac{N}{\lambda} = \frac{N}{p_0^0 \lambda_{00}}$$

## 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、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ファイルを更新する場合を考える。2個のデータ項目が、2個の資源に対応する。資源の要求/解放は確率的に独立で、同じ頻度の起こると仮定する(つまり一方の資源の要求/解放確率は、他方が要求されているいないに関わらず一定である)。これより資源状態の遷移行列は次の通りである。

$$\begin{pmatrix} p^{00} & p^{01} & p^{02} & p^{03} \\ p^{10} & p^{11} & p^{12} & p^{13} \\ p^{20} & p^{21} & p^{22} & p^{23} \\ p^{30} & p^{31} & p^{32} & p^{33} \end{pmatrix} = \begin{pmatrix} p^{00} & w & w & (1 - \sqrt{p^{00}})^2 \\ 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{pmatrix}$$

ここで  $w = \sqrt{p^{00}}(1 - \sqrt{p^{00}})$ ,  $\bar{p}^{01} = p^{01} + p^{02} + p^{03} = 1 - p^{00}$  である。そして  $\bar{p}^{01} = 0, 0.02, 0.04, 0.06, \dots, 0.14$  とおく。この  $\bar{p}^{01}$  は少なくともひとつの資源を要求している確率であり、この値の範囲はオンライン実時間システムにおいて排他制御を伴う入出力が全体の約8%であるという事実に基づいている。 $p^{11} = p^{22} = p^{33} = 0.5$  だから、ジョブは資源を持ったまま平均2回のCPU処理を行った後に資源を解放する。

- (6) CPU ノードから他のノードへの遷移確率:

$$(p_0^0, p_1^0, p_2^0) = (0.2, 0.4, 0.4),$$

$$(p_0^r, p_1^r, p_2^r) = (0.0, 0.4, 0.6) \quad (r = 1, 2, 3).$$

- (7) ジョブがライフタイムの間に資源状態0でノード  $m$  を訪問する平均回数:

$$v_0^0 = \frac{1}{p_0^0} = 5, \quad v_m^0 = p_m^0 v_0^0 = 2 \quad (m = 1, 2).$$

### 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  $\eta$  の定常分布を求めるために平衡方程式を数值的に解く必要がある。しかし表1に示すように、その状態数はもとの待ち行列網の状態  $\delta$  に比べて1~2桁程度減少している。状態  $\eta$  では、CPUとI/Oノードの待ち行列でジョブが並んでいる順序を記述していない。一方、状態  $\delta$  ではこれ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  $\delta$  の状態数の増加

表 1 待ち行列網の状態数 ( $M = 2$ )  
Table 1 Number of states. ( $M = 2$ )

ジョブ数 ( $N$ )	$F = 1$		$F = 2$	
	$\delta$	$\eta$	$\delta$	$\eta$
2	21	3	75	12
4	120	5	1,146	73
6	406	7	9,256	346
8	1,035	9	54,039	1,479

$F = 1$  のとき,  $\eta$  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.

のスピードは  $\eta$  の増加のスピードを大きく上回る. 数値計算に用いる CPU 性能およびメモリ量の観点から, 状態数が 3 ~ 5 万個程度が数値計算の実用上の限度と考えられる. 表 1 によると  $P(\delta)$  を計算することは  $F = 2$  の場合で  $N = 7 \sim 8$  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 1 から推定して,  $F = 2$  の場合で  $N = 30$  程度までは計算可能と考えられる. このように提案した近似モデルは, 従来のモデルよりも利用し易い手法を提供している.

#### 4.3 近似モデルの精度検証

表 2 - 7 に数値解析結果を示す. これらの表で「厳密解」は資源要求付きモデル全体を  $\delta$  を状態とするマルコフ連鎖でモデル化して求めた値であり, 「近似解」は  $\eta$  を状態とするマルコフ連鎖によってモデル化した近似モデルによる値である. これらには全般に次のような傾向が見られる.

- (1) 資源要求確率  $p^{01}$  の小さい方が精度が悪い.
  - (2) 表 3 - 5 が示すように, 資源待ち行列の性能値 (資源待ち行列ビジー率, 平均滞在ジョブ数, 平均滞在時間) は資源数が 1 個の場合に比べて 2 個の場合の方が悪い. 一方, 表 6, 7 が示すように, 網全体の性能値は資源数に関わらず良い近似精度を示す.
- 3.1 で述べたように, 厳密解では網のすべての振る舞いをマルコフ連鎖で記述するが, 近似モデルでは資源の要求/解放がない間のノード内のジョブの振る舞いは積形解で近似する. 資源要求確率が大きいと, ジョブの多くは資源待ち行列につながり, また資源待ち行列内にいる時間も長い. このため資源要求確率の大きい方が, 近似の影響が小さくなり精度が高いと考えられる.

全体としては, もし CPU, I/O ノード側に負荷が高く (ジョブ数が多く) 計算されれば資源待ち行列側の負荷は軽く (ジョブ数が少なく) なり, その逆も正しいので, CPU, I/O

ノード側と資源待ち行列側の差異が相殺されて, 全体の誤差は小さく見えると考えられる. そして近似モデルの近似精度は多くの性能値で数%以内 (一部で 10 数%以内) であり,十分に実用上有用な範囲である.

## 5. むすび

排他資源のあるセントラルサーバモデルの性能値を計算するのに, 数値計算の手間と計算に必要なメモリ量を削減する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/解放がない間の網の状態を一つの状態に縮約し, 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/解放のふるまいはマルコフ連鎖で記述して定常分布を求めることにより解析する.

数値実験の結果, 資源要求確率が小さいときに近似精度が落ちるものの, 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の近似精度は実用上の範囲内であることを確認した.

今後は, メモリといった他の 2 次資源を含む待ち行列網の解析法を明らかにしたい.

## 参 考 文 献

- 1) 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 (1975).
- 2) Kino, I.: Two-layer Queueing Networks, *J. ORSJ*, Vol.40, No.2, pp.163-185 (1997).
- 3) Kurasugi, T. and Kino, I. : Approximation Method for Two-layer Queueing Models, *Performance Evaluation* 36-37, pp.55-70 (1999).
- 4) Rolia, J.A. and Sevcik, K.C. : The Method of Layers, *IEEE Trans. on Software Engineering*, Vol.21, No.8, pp.689-700 (1995).
- 5) 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).
- 6) 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近似評価の一手法, 情報処理学会論文誌 : 数理モデル化と応用, Vol.42, No. SIG 14 (TOM 5), pp.1-13 (2001).

表 2 CPU ビジー率  
 Table 2 CPU busy ratio.

ジョブ数 ( $N$ )	資源要求 確率 ( $\bar{p}^{01}$ )	資源数 $F = 1$		
		厳密解	近似解	誤差率
4	0.02	0.761	0.761	0.13 %
	0.08	0.740	0.739	-0.14
	0.14	0.701	0.702	0.14
8	0.02	0.901	0.900	0.22
	0.08	0.843	0.844	0.12
	0.14	0.747	0.759	1.47
$F = 2$				
4	0.02	0.760	0.762	0.26 %
	0.08	0.744	0.751	0.94
	0.14	0.720	0.732	1.67
8	0.02	0.900	0.902	0.22
	0.08	0.874	0.880	0.67
	0.14	0.823	0.837	1.47

表 3 資源待ち行列ビジー率  
 Table 3 Resource waiting queue busy ratio.

ジョブ数 ( $N$ )	資源要求 確率 ( $\bar{p}^{01}$ )	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	0.213	0.94
	0.14	0.468	0.444	-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	0.0042	0.0049	16.67 %
	0.08	0.0551	0.0607	10.16
	0.14	0.138	0.146	5.80
8	0.02	0.0201	0.0236	17.41
	0.08	0.257	0.266	3.50
	0.14	0.554	0.538	-2.89

$F = 2$  のときは、資源待ち行列 1 のビジー率

表 4 資源待ちの平均ジョブ数  
 Table 4 Number of jobs in resource waiting queue.

ジョブ数 ( $N$ )	資源要求 確率 ( $\bar{p}^{01}$ )	資源数 $F = 1$		
		厳密解	近似解	誤差率
4	0.02	0.0177	0.0203	3.95 %
	0.08	0.262	0.274	4.58
	0.14	0.675	0.658	2.52
8	0.02	0.0989	0.1159	2.02
	0.08	1.874	1.805	-3.68
	0.14	3.812	3.553	-6.79
$F = 2$				
4	0.02	0.0086	0.0100	16.28 %
	0.08	0.122	0.137	12.30
	0.14	0.322	0.350	8.70
8	0.02	0.0442	0.0531	20.14
	0.08	0.747	0.815	9.10
	0.14	1.980	2.007	1.36

$F = 2$  のときは、資源待ち行列 1 と 2 の滞在ジョブ数の合計

表 5 1 回の資源待ち行列平均滞在時間  
 Table 5 Mean resident time in resource waiting queue.

ジョブ数 ( $N$ )	資源要求 確率 ( $\bar{p}^{01}$ )	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	3.333	3.906	17.19 %
	0.08	3.829	4.311	12.59
	0.14	4.356	4.680	7.44
8	0.02	7.370	8.907	20.85
	0.08	11.111	12.171	9.54
	0.14	15.901	15.311	-3.71

$F = 2$  のときは、資源待ち行列 1 の平均滞在時間

表 6 ジョブのスループット  
 Table 6 Job throughput.

ジョブ数 ( $N$ )	資源要求 確率 ( $\bar{p}^{01}$ )	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	0.297	0.300	1.00
	0.14	0.288	0.293	1.74
8	0.02	0.360	0.361	0.28
	0.08	0.350	0.352	0.57
	0.14	0.329	0.335	1.82

表 7 ジョブの平均応答時間  
 Table 7 Mean job response time.

ジョブ数 ( $N$ )	資源要求 確率 ( $\bar{p}^{01}$ )	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	13.167	13.132	-0.33 %
	0.08	13.446	13.314	-0.98
	0.14	13.895	13.655	-1.73
8	0.02	22.218	22.174	-0.20
	0.08	22.890	22.715	-0.76
	0.14	24.301	23.888	-1.70