2005－MPS－55（1）
2005／6／28

資源要求の頻度が小さい場合の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†

ファイルなどの逐次アクセス資源のある計算機システムの性能を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考えられる．しかしこの方法は，モデルの状態数が増えたときにメモリ量や計算量が増加し数値計算が困難になるという問題がある．

そこで本報告では資源を要求する頻度が小さい場合について，この数値計算上の困難を回避するための近似手法を提案する．近似の考え方は，資源要求の頻度を表すパラメータの 2 次以上の項を無視するというものである．これにより網は積形解をもつので平衡方程式を解く必要がなくなり，積形解の一般論を用いて性能値を求めることができる．そして数値実験により，提案した近似モデルが実用の範囲内の精度をもつことを確認した．

Approximate Technique for Queuing Network for Evaluating Performance of Computer Systems with Small Rate of Resource Requirements

Toshiyuki Kinoshita †

In computer systems, job conflicts occur at accessing an *exclusively-used resource*, such as an updatable file. In my previous work, I introduced a queuing network model to analyze the influence of the conflicts on the performance of the computer systems. In order to solve equilibrium equations of the markov chain of the model, however, we have to waste much computer calculation which may make us be in difficulty to analyze the model.

In this report, I propose an approximate method which modifies the model to have a product-form solution when the model has small rate of resource requirements. The policy and the algorithm of the approximation are described and its accuracy is analyzed by numerical experiments.

1. まえがき

計算機システムのファイルデータは，その一致性を保つために一つのジョブによって排他的に使用される．その意味でファイルは排他使用資源と呼ぶことができる．

この排他使用資源（以降，単に資源と呼ぶ）は計算機の性能に大きな影響を及ぼすにもかかわらず，一般の待ち行列理論では取り扱うことが困難なため，個別にモデルを構築する必要がある（Rolia[1], Kurasugi[2] など）．木下[3]では資源をもつ計算機システムを待ち行列網で記述し，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．この方法は資源要求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られる．しかし平衡方程式を数値的に解くので，ジョブ数や資源数が大きくなった場合，システムの状態数が膨大となり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．

そこで本報告では，資源要求の頻度が少ない範囲において，数値計算の手間と必要なメモリ量を大幅に削減し，計算可能な範囲を拡大させる近似手法を提案する．近似の考え方は，資源要求の頻度を表すパラメータの 2 次以上の項を無視するというものである．これにより待ち行列網を積形解をもつ形に置き換えることができ，平衡方程式を数値的に解く必要がなくなって，積形解の一般論を用いて容易に性能値を求めることができる．

† 東京工科大学
Tokyo University of Technology

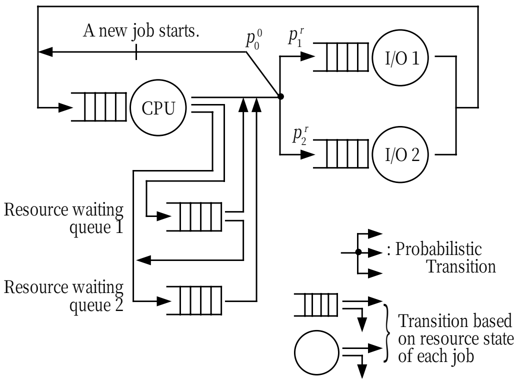

図 1 資源要求のあるセントラルサーバモデルのイメージ図（資源が 2 個の場合）

Fig. 1 Image diagram of central server model with resource requirement.（In the case of 2 resources）

2. モデルの記述

2.1 資源要求のあるセントラルサーバモデル

資源要求のある待ち行列網は，単一の CPU ノード（ノード番号 $m = 0$）と M 個の I/O ノード（ノード番号 $m = 1, 2, \ldots, M$）及び D 個の資源待ち行列（$m = M+1, M+2, \ldots, M+D$ で参照する）からなり，N 個のジョブが網内を移動する（図 1）．D 個の資源があり（資源番号 $d = 1, 2, \ldots, D$），資源 S_dと資源待ち行列 $M+d$ が対応する．ノード m（CPU ノードと I/O ノード）でのサービス率は μ_mで，サービス時間は共通の指数分布に従う．すべてのノード及び資源待ち行列で，ジョブは FCFS（First Come First Served）規律でサービスされる．

ジョブが資源 S_dを要求したときこれが占有されていると，ジョブは資源待ち行列 $M+d$ に入って資源が解放されるのを待つ．資源が解放されると，資源待ち行列の先頭のジョブが資源を獲得する．このようにしてジョブが必要なすべての資源を獲得すると，次の I/O ノードに進む．

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の集合 $\{S_{d_1}, S_{d_2}, \ldots, S_{d_{D'}}\}$（$0 \le D' \le D$）のことである．資源状態に通し番号を付け，その番号を r 又は r_iで表す．$r = 0$ は必要としている資源がない状態（すなわち $D' = 0$）を表わすものとする．ジョブの資源状態が r_1から r_2に変更される遷移確率を $p^{r_1 r_2}$とする．例えば資源が 2 個の場合，ジョブの資源状態は

$r = 0$: ϕ $\cdots\cdots$ 資源を必要としない状態
$r = 1$: $\{S_1\}$ $\cdots$ 資源 S_1のみを必要としている状態
$r = 2$: $\{S_2\}$ $\cdots$ 資源 S_2のみを必要としている状態
$r = 3$: $\{S_1, S_2\}$ $\cdots$ 資源 S_1と S_2を必要としている状態

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

CPU ノードから出発した後，資源状態 r のジョブは確率 p_m^rでノード m（CPU ノード又は I/O ノード）に進む．また資源待ち行列に並んでいたジョブが資源を獲得して資源待ち行列を離れると，同じ確率 p_m^rで I/O ノード m に進む．

ジョブが CPU から CPU へ直接に遷移すると，そのジョブは終了し新たな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，一つのジョブのライフタイム（応答時間）である．ジョブは資源を保持したまま終了できないので，$r > 0$ ならば $p_0^r = 0$ とする．

2.2 システムの状態

システムの状態は，ベクトル

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

で表わされる．ここで N_mはノード又は資源待ち行列 m のジョブ数であり，k_m^qは対応するジョブの資源状態を表わす．すなわち $k_m^q = r$ ならば，ノード又は資源待ち行列 m の第 q 番目にいるジョブの資源状態が r であることを表わす．したがって $k_0^1 \cdots k_0^{N_0}$; $k_1^1 \cdots k_1^{N_1}$; $\cdots$; $k_M^1 \cdots k_M^{N_M}$; $k_{M+d+1}^1 \cdots k_{M+d+1}^{N_{M+d+1}}$; $\cdots$; $k_{M+D}^1 \cdots k_{M+D}^{N_{M+D}}$ の中には，資源 S_dを含む資源状態のジョブはそれを獲得しているジョブのみのたかだか一つである．一方 $k_{M+1}^1 \cdots k_{M+1}^{N_{M+1}}$; $\cdots$; $k_{M+d}^1 \cdots k_{M+d}^{N_{M+d}}$の中には，資源 S_dを含む資源状態のジョブは複数存在し得る（資源 S_d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）．

3. 資源要求確率が小さい場合の近似モデル

3.1 近似モデルの記述

近似の考え方は，資源要求確率が小さい場合に資源要求確率の 2 次以上の項を無視するというものである．資源要求確率の 2 次以上の項は異なるジョブが同時に資源を要求することを意味するが，近似モデルではこれを無視する．したがって資源へのアクセスのぶつかりも無視される．この変更により資源の要求/解放に

表 1　システムの状態数（$M = 2$）
Table 1　Number of states ($M = 2$)

ジョブ数	資源数		
(N)	$D = 1$	$D = 2$	$D = 3$
2	21	78	306
4	120	1,383	21,075
6	406	13,684	846,964
8	1,035	100,077	26,088,165

よる「資源状態の変更」は，複数のジョブクラスをもつ通常のセントラルサーバモデルでの「ジョブクラスの変更」と全く同等になる．そこで近似モデルは，次にように記述される．

(1) 近似モデルの待ち行列網の構成は，2^D個のジョブクラスをもつ資源要求のない通常のセントラルサーバモデルとする．資源待ち行列は存在しない．
(2) ジョブは CPU ノードのサービス終了時に，ジョブクラスを r_1から r_2に確率 $p^{r_1 r_2}$で変更される $(r_1, r_2 = 0, 1, \ldots, 2^D - 1)$.
(3) ジョブが CPU ノードのサービス終了時にクラス r に属していれば，そのジョブは確率 p^r_mで ノード m（CPU ノード又はI/O ノード）に進む．ただし CPU から CPU への直接の遷移は，クラス 0 に属しているときにのみ起こるものとする．すなわち $r > 0$ ならば $p^r_0 = 0$ とする．

例えば資源が 2 個の場合，**2.1** で述べたように 4 通りの資源状態が考えられるので，近似モデルでもこれらに対応して 4 個のジョブクラスを設定し，資源の要求/解放（すなわち資源状態の変化）に対応してクラスの変更が起こるものとする．

3.2　近似モデルの積形解

近似モデルの状態は，

$\delta^* = (k^1_0 \cdots k^{N_0}_0;\, k^1_1 \cdots k^{N_1}_1; \cdots;\, k^1_M \cdots k^{N_M}_M)$

と表わされる．これは **2.2** の δ から資源待ち行列に対応する $k^1_{M+1} \cdots k^{N_{M+1}}_{M+1}$; $\cdots$; $k^1_{M+D} \cdots k^{N_{M+D}}_{M+D}$を削除したものである．可能なすべての δ^*の集合をΔ^*で表わす．状態 δ^*でのクラス r のジョブ数を N^r，ノード m 中のクラス r のジョブ数を N^r_mとする．明らかに $N^r = \sum_{m=0}^{M} N^r_m$である．またジョブがライフタイム中にクラス r でノード m を訪問する平均回数を v^r_m，クラス r の間にノード m で受ける総サービス時間の平均値を τ^r_mとすると，ノード m の 1 回の滞在中に受ける平均サービス時間が $1/\mu_m$だから，$\tau^r_m = v^r_m/\mu_m$ と表わされる．

近似モデルの定常状態確率 $P(\delta^*)$ は，積形解により次のように与えられる．

$$P(\delta^*) = \frac{1}{G^*(N)} \times \prod_{m=0}^{M} \prod_{q=1}^{N_m} \tau_m^{k^q_m}$$
$$= \frac{1}{G^*(N)} \times \prod_{m=0}^{M} \prod_{r=0}^{2^D-1} (\tau^r_m)^{N^r_m}$$

この正規化定数 $G^*(N)$ は，

$$G^*(N) = \sum_{\delta^* \in \Delta^*} \prod_{m=0}^{M} \prod_{r=0}^{2^D-1} (\tau^r_m)^{N^r_m} \qquad (1)$$

である．この式 (1) の和を各 δ^* に含まれる'$r\,(>0)$'の個数 l で分類し，$l \geq 2$ の項を無視すると，

$$G^*(N) = \sum_{l=0}^{N} G^*_l(N) \sim G^*_0(N) + G^*_1(N)$$

ここで

$$G^*_0(N) = G_0(N), \quad G^*_1(N) = N\,G_0(N) \sum_{r=1}^{2^D-1} \frac{p^{0r}}{p^{r0}}$$

と表されるから，$G^*(N)$ の $p^{0r} = 0 \quad (r > 0)$ における 1 次の近似式は，

$$G^*(N) \sim G_0(N) \left(1 + N \sum_{r=1}^{2^D-1} \frac{p^{0r}}{p^{r0}} \right) \qquad (2)$$

である．

3.3　平均応答時間の近似式

式 (2) を使って，ジョブの平均応答時間の近似値を求めてみる．ジョブクラスが 1 個の資源要求のない通常のセントラルサーバモデルのジョブの平均応答時間を $T_0(N)$，資源要求のあるモデルのうち近似前の厳密なモデルの平均応答時間を $T(N)$，近似モデルの値を $T^*(N)$ とすると，積形解の定理により

$$T_0(N) = \frac{N\,G_0(N)}{G_0(N-1)}, \quad T^*(N) = \frac{N\,G^*(N)}{G^*(N-1)}$$

だから，

$$T(N) \sim T^*(N) = \frac{N\,G^*(N)}{G^*(N-1)}$$
$$\sim \frac{N\,G_0(N)}{G_0(N-1)} \left(1 + \sum_{r=1}^{2^D-1} \frac{p^{0r}}{p^{r0}} \right) \ (\because (2) \text{ による})$$
$$= T_0(N) \left(1 + \sum_{r=1}^{2^D-1} \frac{p^{0r}}{p^{r0}} \right) \qquad (3)$$

と求められる．

4. 数値実験

4.1　パラメータ

近似モデルの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：$N = 2, 4, 8, 12$

(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

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

ただし $w = \sqrt{p^{00}}(1-\sqrt{p^{00}})$．そして $\overline{p}^{01} = p^{01} + p^{02} + p^{03} = 1 - p^{00}$ について，$\overline{p}^{01} = 0, 0.02, 0.04, 0.06, \ldots, 0.14$ とする．この $\overline{p}^{01}$ はいずれか（または両方）の資源を要求する確率を表し，現実の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十数回に 1 回（約 8%）であることに基づいている．

(6)　CPU ノードからの推移確率：
$(p_0^0, p_1^0, p_2^0) = (0.2, 0.4, 0.4)$
$(p_0^r, p_1^r, p_2^r) = (0.0, 0.4, 0.6) \qquad (r = 1, 2, 3)$

(7)　一つのジョブが資源状態 0 の間にノード m を訪問する平均回数 v_m^0 は，
$v_0^0 = \dfrac{1}{p_0^0} = 5, \quad v_m^0 = p_m^0 v_0^0 = 2 \quad (m = 1, 2)$
である．これより $\tau_0^0 = 2.5,\ \tau_1^0 = \tau_2^0 = 2$ で，これらを **3.2** の式 (2) に代入して $G^*(N)$ が求められる．

4.2　数値解析結果

図 2 にジョブの平均応答時間の近似モデルの値と厳密解を示す．また表 2 に $\overline{p}^{01} = 0.8$ のときのそれらの値を示す．近似式 (3) では資源要求確率の 2 次以上の残余項を無視しているので，明らかに精度は $\overline{p}^{01}$ の値が小さいほど良い．

数値実験によると近似モデルの相対誤差は 5%程度以内に収まっており，実用上有用な範囲内である．そして平衡方程式を解く必要がないので，モデルの状態数に影響されることなく積形解の一般論を用いて簡便に性能値を求めることができる．

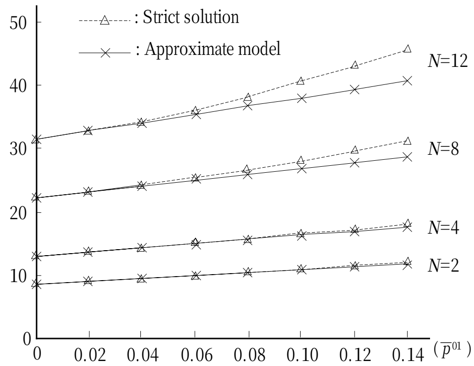

図 2　ジョブの平均応答時間の比較
Fig. 2　Mean job response times

表 2　ジョブの平均応答時間の比較（$\overline{p}^{01} = 0.08$）
Table 2　Results of mean job responce time ($\overline{p}^{01} = 0.08$)

ジョブ数 (N)	近似モデル (T)	厳密解 (T_0)	誤差率 ($(T-T_0)/T_0$)
2	10.08	10.35	− 2.61 %
4	15.22	15.70	− 3.06
8	25.69	26.68	− 3.71
12	36.42	38.09	− 4.38

5. む す び

木下[3]で提案した資源要求のある計算機システムを待ち行列論の数値計算により解析する方法の計算量増大の問題を解決するために，資源要求の頻度が小さいときに資源要求確率の 2 次以上の項を無視するという考え方による近似モデルを導入した．これによりジョブクラスが 1 個の待ち行列網の積形解と同程度の手間で計算可能にした．

参 考 文 献

1) J.A. Rolia, and K.C. Sevcik, “The Method of Layers,” IEEE Trans. on Software Engineering, Vol.21, No.8, pp.689–700, Aug. 1995.
2) T. Kurasugi, and I. Kino, “Approximation Method for Two-layer Queueing Models,” Performance Evaluation 36–37, pp.55–70, 1999.
3) 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，” 電子情報通信学会論文誌 (D-I)，vol.J 82-D-I，no.6，pp.701–710，Jun. 1999.