

即時・待時混在システムの階層的性能評価法

川西俊之† 紀一誠‡

† 神奈川大学理学研究科情報科学専攻

‡ 神奈川大学理学部情報科学科

本論では、即時系と待時系が混在するサーバシステムの性能評価を2層型待ち行列網を用いて行う評価法の提案を行う。2層型待ち行列網モデルはシステム内のソフトウェア資源競合とハードウェア資源競合を統一的に評価することができる。この性質を利用して、即時・待時混在系システムの新しい評価モデルを示す。

Two-layer queueing network model for loss-delay mixture system

Toshiyuki Kawanishi† Issei Kino‡

† Graduate School of Science, The Course of Information Sciences, Kanagawa University

‡ Department of Information Sciences, Kanagawa University

New performance evaluation technique for loss-delay mixture system is proposed in this paper. The technique is based on the two-layer queueing network model which provides unified method of performance evaluation for software and hardware resource congestion.

1 はじめに

本論では、2層型待ち行列網モデルを用いた即時・待時混在システムの性能評価方法について述べる。コンピュータやネットワークシステムは一般にハードウェア資源とソフトウェア資源により構成されており、両者の機能と動作は密接に関連している。そのため、このようなシステムの性能評価に際しては、これらのソフトウェア資源およびハードウェア資源の構造的関係とその競合関係を的確に反映したモデル化が必要になる。相互に関連する多数の資源競合を表現するためのモデルとしては待ち行列網モデルが効果的であることはよく知られている[11]。

しかしながら、ソフトウェア資源とハードウェア資源の関係は並列的なものではなく階層的な複雑な関係にあるために、その階層関係の適切なモデル化が必要となる。資源間の構造に階層的な関係

があるモデルは、待ち行列網モデルを変形した同時資源保留 (simultaneous resource possession) モデル[18, 6], 受動サーバ (passive server) モデル[5]をはじめ、マルコフ連鎖を直接扱う木下[13]のモデル, Buchhold [3], Rolia [17]等が知られている。この形の待ち行列網は一般には積形式解をもたないので、その近似解析法として等価流量法 (flow equivalent method) や Norton の定理を用いる方法など様々な方法が開発されている[1, 2, 4]。

2層型待ち行列網モデルはこれらのモデルを一般化し統合する形で紀[12]により提案された。また、2層型待ち行列網の近似解析法がKurasugi and Kino [14]により提案されている。2層型待ち行列網モデルの応用例としては、蔵杉他[15]およびKaieda 他[7]によるSMP型コンピュータシステムの性能評価、笠原他[8]によるオーバレイネットワークの性能評価などが報告されている。2層型待ち行列網モデルの具体的な応用に際してはモデル固

有の特徴を表現するための考察が必要となる。本稿ではこれまで報告されていなかった応用として、チケット予約システムにみられるような即時系と待時系が混在するシステムを対象として、2層型待ち行列網モデルの適応方法について記述する。

2 2層型待ち行列網の構造と特徴

2.1 2層型汎用モデル

2層構造：2層型待ち行列網は上位層 (Upper Layer) と下位層 (Lower Layer) の2層から構成される。上位層にはステーション (Station) といわれる待ち行列が M 個、下位層にはノード (Node) といわれる待ち行列が N 個存在している。ステーションおよびノードはいずれもサーバ数任意の複数サーバ型待ち行列 (無限サーバを含む) であり、サービス規律はすべて先着順とする。上位層のステーション i のサーバ数を s_i , $i \in \mathbf{I}_M$, 下位層のノード i のサーバ数を c_i , $i \in \mathbf{I}_N$ とする。ここで自然数の集合を, $\mathbf{I}_M = \{1, 2, \dots, M\}$ と表すものとする。

2層型待ち行列網の応用に際しては、上位層のステーションはソフトウェア資源に、下位層ノードはハードウェア資源に対応付けることができる。コンピュータシステムにおいては、「プロセス」のようなオペレーティングシステムにより管理されるソフトウェア資源を割り当てられたもののみがハードウェア資源を利用できるという資源の2重構造があり、その構造をモデル化したものが2層型待ち行列網である。

上位層の構造：上位層では、トランザクション (transaction) といわれる L 種類の処理要求がプロセス (process) といわれるソフトウェア資源を次々と使いながら処理を進めていく。ソフトウェア資源プロセスはステーションに対応付けられる。すなわち各トランザクションは種々のステーションで次々にサービスを受けながら処理を進めていく。トランザクション処理の流れは上位層のステーション間の客の移動を規定する部分連鎖 (subchain) と

して推移確率行列 P_ℓ , $\ell \in \mathbf{I}_L$, により表現される。

トランザクション処理は一般には複数個が同時並列的に存在しており、それぞれの処理内容の相異は対応する部分連鎖の相異によって表現される。上位層のステーション間移動経路は混合型とする。すなわち、上位層の部分連鎖は開放型であつても閉鎖型であつてもどちらでも良く、またそれぞれの部分連鎖がいくつあつてもよい。部分連鎖が開放型の場合には、外部からの客の到着はポアソン過程にしたがうものとする。 P_ℓ を係数とする連立一次方程式を解くことにより、ステーション i への連鎖 ℓ の客の到着率 (開放型) あるいは相対訪問回数 (閉鎖型) $v_{i\ell}$, $i \in \mathbf{I}_M$, $\ell \in \mathbf{I}_L$, を定めることができる。ステーション i における連鎖 ℓ の客の平均滞在時間 (サービス時間+待ち時間) $\tau_{i\ell}$ が得られれば、トランザクション ℓ の処理時間は $\sum_{i=1}^M v_{i\ell} \tau_{i\ell}$ として求まる。

ステーション i の滞在客数を n_i とし、上位層の状態を $\mathbf{n} = (n_1, n_2, \dots, n_M)$ とする。

ステーションにおける動作：ステーション i に到着した客は、空きサーバがあればそのサーバを保留し、直ちに下位層に移動する。この客は下位層での一連のサービスを終了するとやがて上位層に戻ってくる。この間の時間、すなわち客が下位層に滞在する総時間 (総サービス時間および総待ち時間の和) が上位層のステーション i でのサービス時間になる。到着ステーションに空きサーバがない場合には、サーバに空きができるまで待ち行列中で待つ。以上のことから、ステーション i のサービス時間は固定的に与えられているわけではなく、下位層の混雑度に応じて変化することになる。下位層の混雑度は上位層の各ステーションから降りてきている客数に応じて変わるので、ここに上位層の混雑と下位層の混雑の間の関係が生じることになる。

ステーション i から下位層に降りることのできる最大客数はサーバ数 s_i までであることから、このステーションから、状態が \mathbf{n} の場合、下位層に降

りてきている客数 k_i は

$$(1) \quad k_i = \min(n_i, s_i), \quad i \in I_M$$

となる。上位層の状態が \mathbf{n} であるときの下位層の客数ベクトルを $\mathbf{k} = (k_1, k_2, \dots, k_M)$ とする。 $\mathbf{s} = (s_1, s_2, \dots, s_M)$ とすれば、 \mathbf{k} の値のとり得る範囲は以下である。

$$(2) \quad 0 \leq \mathbf{k} \leq \mathbf{s}.$$

下位層の構造：下位層を構成する待ち行列網は上位層のステーション数 M に等しい数の部分連鎖をもつ複数連鎖型の閉鎖型待ち行列網とする。上位層のステーション m から下位層に降りてきた客は、下位層ではステーション m に対応付けられる閉鎖型部分連鎖 $R_m = (r_{ij}^m)$ に従って下位層内のノード間を推移し、再び上位層のステーション m にもどっていく。ここで、 r_{ij}^m は部分連鎖 m に属する客がノード i でのサービス終了後ノード j に推移する確率とし、 $m \in I_M, i, j \in \{*, 1, 2, \dots, N\}$ とする。ただし、 $*$ は上位層を表す特別なノード番号と考える。すなわち、 r_{*i} は上位層から降りてきた客がノード i を選択する確率、 r_{j*} はノード j でのサービスを終了した客が下位層のノードに推移することなく再び上位層にもどる確率とする。下位層のノード $i \in I_N$ におけるサービス時間は連鎖に依存しないパラメータ μ_i をもつ指数分布にしたがうものとする。

訪問回数： $\theta_m = (1, \theta_{m1}, \theta_{m2}, \dots, \theta_{mN})$ とし、 R_m を係数とする連立方程式

$$(3) \quad \theta_m R_m = \theta_m, \quad m \in I_M$$

を考える。この方程式の解として得られる θ_{mj} は連鎖 m に属する客すなわちステーション m から降りてきた客が再び上位層のステーション m に戻るまでの間にノード j を訪問する平均訪問回数を意味している。さらに、トラヒック係数を $\rho_{jm} = \frac{\theta_{jm}}{\mu_j}$, $j \in I_N, m \in I_M$ とし、トラヒック係数ベクトルを $\rho_j = (\rho_{j1}, \rho_{j2}, \dots, \rho_{jM})$, $\rho = (\rho_1, \rho_2, \dots, \rho_N)$

とする。

下位層の状態と積形式解：ノード j に滞在する連鎖 m の客数を x_{jm} , ノード j の状態を $\mathbf{x}_j = (x_{j1}, x_{j2}, \dots, x_{jM})$, 下位層の状態を $\mathbf{x} = (x_1, x_2, \dots, x_N)$ とする。 $j \in I_N$ について

$$(4) \quad q_j(\mathbf{x}_j) = \begin{cases} \frac{\rho_j^{x_j}}{x_j!}, & \|\mathbf{x}_j\| \leq c_j \\ \frac{\|\mathbf{x}_j\|}{c_j! c_j^{\|\mathbf{x}_j\| - c_j}} \frac{\rho_j^{x_j}}{x_j!}, & c_j < \|\mathbf{x}_j\| \end{cases}$$

とする。このとき、下位層の客数ベクトルが \mathbf{k} である条件下での下位層の状態確率 $P(\mathbf{x}|\mathbf{k})$ は以下のように表される。

$$(5) \quad P(\mathbf{x}|\mathbf{k}) = \frac{1}{G(\mathbf{k})} \prod_{j=1}^N q_j(\mathbf{x}_j),$$

$$(6) \quad G(\mathbf{k}) = \sum_{\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_N = \mathbf{k}} \prod_{j=1}^N q_j(\mathbf{x}_j).$$

ここで、 $\mathbf{x} = (x_1, x_2, \dots, x_N)$, $\rho = (\rho_1, \rho_2, \dots, \rho_N)$ について以下のベクトル記法を用いた。

$$\|\mathbf{x}\| = x_1 + x_2 + \dots + x_n, \quad \mathbf{x}! = x_1! x_2! \dots x_n! \\ \rho^{\mathbf{x}} = \rho_1^{x_1} \rho_2^{x_2} \dots \rho_n^{x_n}.$$

下位層のスループットと滞在時間：下位層の客数ベクトルが \mathbf{k} という条件のもとで、ステーション i から降りてきた客の下位層でのスループットは

$$(7) \quad \mu_i(\mathbf{k}) = \frac{G(\mathbf{k} - \mathbf{e}_i)}{G(\mathbf{k})}, \quad i \in I_M$$

で与えられ、平均滞在時間はその逆数 $1/\mu_i(\mathbf{k})$ として得られる。 \mathbf{e}_i は i 方向の単位ベクトルとする。この滞在時間が客数ベクトル条件 \mathbf{k} のもとでの上位層のステーション i のサービス時間となる。

図 2.1 に 2 層型待ち行列網の一例を示す。本例では、 $M = 2, N = 3$ であり、各ステーション、各ノードは総て単一サーバである。上位層には、トランザクション 1 が開放型部分連鎖、トランザクション 2 が閉鎖型部分連鎖に従い、ステーション間を推移する。ステーション 1 および 2 から下位層に移動した客はそれぞれ閉鎖型部分連鎖 1 および 2 に従い下位層のノード間を移動する。

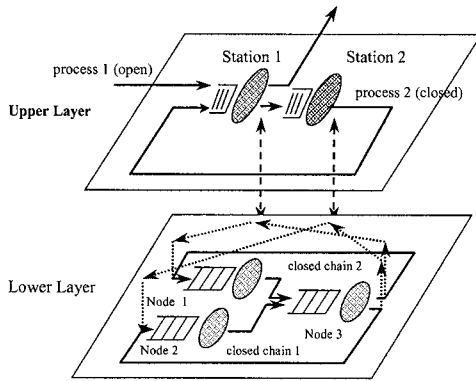


図 1: 2 層型待ち行列網モデル例

3 即時・待時混在システムの解析

3.1 プロセス構成

即時系・待時系混在システムとは、上位層（ソフトウェア資源）のプロセスの競合形態に、即時型と待時型の両者を同時に持つようなシステムをいう。すなわち、即時系のプロセスとはある固定した数 s 個の待ち合わせしか許さず、これを超える客が到着した場合には呼損（廃棄）とするプロセスであり、待時系のプロセスとは待ち合わせ客の数に制限の無いものをいう。このようなプロセス構成は、チケット予約システムにおいて、受付プロセスを即時系とすることにより予約殺到時点での過負荷対策とし、受付が完了したものについては待時系プロセスで対応するような場合にみられる。このようなシステムをモデル化するには、即時型プロセスには待合室有限のステーションで、待時型プロセスには待合室無限のステーションを対応させればよいことは明らかであろう。次節にそのモデル化と解析手順の概要を示す。

3.2 解析手順

- (a) 上位層の構造を定める。トランザクション種別 (L)、ステーション（プロセス）の構成 (s_i)、処理の流れ（上位層の部分連鎖 P_L ）等々を確定する。
- (b) 下位層の構造を定める。ハードウェア構成 (c_i)、ステーション対応の推移確率行列 (R_i)、トラヒック係数 (ρ) 等々。
- (c) (2) 式に示される範囲にある客数ベクトル \mathbf{k} について下位層を閉鎖型 M 連鎖網として解き、各ステーションごとのスループット $\mu_i(\mathbf{k})$ を (7) を用いて求める。
- (d) 上位層の状態が \mathbf{n} であるときのステーション i のサービス率は $\mu_i(\mathbf{k})$ であると考え、上位層の状態推移をマルコフ過程として近似し、状態方程式をつくる。
- (e) 状態方程式を解く。
- (f) その解を用いて各種性能評価指標を算出する。

3.3 解析例

本節では即時・待時混在システムのモデル化と解析を以下の例に従って例示的に示すことにする。**解析手順 (a),(b):** トランザクションは 1 種類 ($L = 1$) とし、外部から到着率 λ のポアソン到着をするものとする。上位層は 2 つのステーションから構成され、ステーション 1 はサーバ数 2 の即時型サーバ、ステーション 2 はサーバ数 1 の待時型サーバとする。すなわち、 $s_1 = 2, s_2 = 1$ であり、ステーション 1 においてすべてのサーバがサービス中の状態に出会ったトランザクションは呼損となるものとする。トランザクションは到着時にステーション 1 の処理を要求し、その処理が完了した場合にはステーション 2 に進みその処理を要求する。下位層には 3 つのノードがあり、すべて単一サーバノードとする。すなわち、 $c_1 = c_2 = c_3 = 1$ 。ノード 1 は CPU 資源、ノード 2, 3 は DB（データベースディ

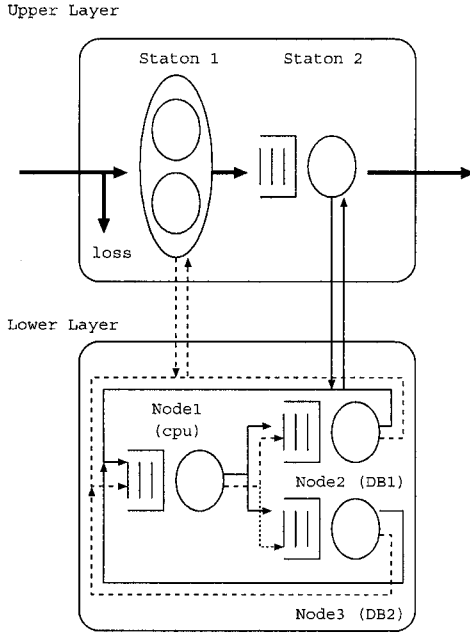


図 2: 即時・待時混在システムモデル例

スク) に対応するものとする. さらに, 各ノードに関するトラヒックパラメータは $\rho_1 = (5, 0, 0.5)$, $\rho_2 = (4.0, 3.0)$, $\rho_3 = (0.5, 3.0)$ の如くに与えられているものとする. これは, 各ステーション (プロセス) からのハードウェア利用は CPU および 2 台の DK のすべてを利用して行われるが, その利用特性は異なっていることを表している. このモデルは図 3.3 に示されるような $M = 2, N = 3$ の 2 層型待ち行列網になる.

解析手順 (c): 条件 (2) を満たす k は

$$k \in \{(0, 0), (0, 1), (1, 0), (1, 1), (2, 0), (2, 1)\}$$

であるからそれぞれについて下位層の閉鎖型待ち行列網の計算を行い, $\mu_1(k), \mu_2(k)$ を求めると表 1 に示すようになる. また k が与えられたという条件の下での下位層のノード (CPU(1), DB1(2), DB2(3))

表 1: 下位層のスループット

k	$\mu_1(k)$	$\mu_2(k)$
(0,0)	0.0000	0.0000
(0,1)	0.0000	0.1250
(1,0)	0.0952	0.0000
(1,1)	0.0761	0.1000
(2,0)	0.1166	0.0000
(2,1)	0.0953	0.0817

表 2: k のもとでのノード使用率 ($\tau_i(k)$)

k	CPU(1)	DB1(2)	DB2(3)
(0, 0)	0.00000	0.00000	0.00000
(0, 1)	0.437500	0.0625000	0.375000
(1, 0)	0.476190	0.380952	0.0476190
(1, 1)	0.730952	0.354762	0.338095
(2, 0)	0.583333	0.466667	0.0583333
(2, 1)	0.763001	0.422375	0.292938

の使用率 $\tau_i(k)$ を表 2 に示す.

解析手順 (d): 上位層の状態空間は

$$S = \{n = (n_1, n_2) \mid n_1 = 0, 1, 2; n_2 = 0, 1, 2, \dots\}$$

であり, 状態推移を表すマルコフ過程の推移率行列 Q は以下のようなブロック 3 重対角行列になる.

$$(8) \quad Q = \begin{pmatrix} B_0 & C_0 & & & \\ A_2 & A_1 & A_0 & & \\ & A_2 & A_1 & A_0 & \\ & & & \ddots & \ddots & \ddots \end{pmatrix}.$$

ここで,

$$B_0 = \begin{pmatrix} (0,0) & (1,0) & (2,0) \\ (0,0) & \begin{pmatrix} -\lambda & \lambda & 0 \\ 0 & -\{\lambda + \mu_1(1,0)\} & \lambda \\ 0 & 0 & -\mu_1(2,0) \end{pmatrix} \end{pmatrix},$$

$$C_0 = \begin{pmatrix} (0,1) & (1,1) & (2,1) \\ (0,0) & \begin{pmatrix} 0 & 0 & 0 \\ \mu_1(1,0) & 0 & 0 \\ 0 & \mu_2(2,0) & 0 \end{pmatrix} \end{pmatrix}$$

であり, $n_2 \geq 1$ について

$$\mathbf{A}_0 = \begin{pmatrix} (0, n_2) \\ (1, n_2) \\ (2, n_2) \end{pmatrix} \begin{pmatrix} (0, n_2+1) & (1, n_2+1) & (0, n_2+1) \\ 0 & 0 & 0 \\ \mu_1(1, 1) & 0 & 0 \\ 0 & \mu_1(2, 1) & 0 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} (0, n_2) \\ (1, n_2) \\ (2, n_2) \end{pmatrix} \begin{pmatrix} (0, n_2-1) & (1, n_2-1) & (0, n_2-1) \\ \mu_2(0, 1) & 0 & 0 \\ 0 & \mu_2(1, 1) & 0 \\ 0 & 0 & \mu_2(2, 1) \end{pmatrix}$$

$$\mathbf{A}_1 = \begin{pmatrix} (0, n_2) \\ (1, n_2) \\ (2, n_2) \end{pmatrix} \begin{pmatrix} (0, n_2) & (1, n_2) & (0, n_2) \\ -a_1 & \lambda & 0 \\ 0 & -a_2 & \lambda \\ 0 & 0 & -a_3 \end{pmatrix},$$

$$\begin{aligned} a_1 &= \lambda + \mu_2(0, 1) \\ a_2 &= \lambda + \mu_1(1, 1) + \mu_2(1, 1) \\ a_3 &= \mu_1(2, 1) + \mu_2(2, 1). \end{aligned}$$

ここで,

$$\mathbf{B}_0 + \mathbf{C}_0 = \mathbf{0}, \quad \mathbf{A}_2 + \mathbf{A}_1 + \mathbf{A}_0 = \mathbf{0}$$

であることに注意.

解析手順 (e): 定常状態確率を $\pi_{i, n_2} = P((i, n_2))$, $i = 0, 1, 2$, $\boldsymbol{\pi}_n = (\pi_{0n}, \pi_{1n}, \pi_{2n})$ とし, $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots)$ とすると, 解くべき状態方程式は

$$(9) \quad \boldsymbol{\pi} \mathbf{Q} = \mathbf{0}$$

となる. 方程式 (9) はその繰り返し構造を利用することにより以下のように解くことができる [16]. 方程式 (9) は以下のように書き下せる.

$$(10) \quad \boldsymbol{\pi}_0 \mathbf{B}_0 + \boldsymbol{\pi}_1 \mathbf{A}_2 = \mathbf{0}$$

$$(11) \quad \boldsymbol{\pi}_0 \mathbf{C}_0 + \boldsymbol{\pi}_1 \mathbf{A}_1 + \boldsymbol{\pi}_2 \mathbf{A}_2 = \mathbf{0}$$

および $n \geq 2$ について

$$(12) \quad \boldsymbol{\pi}_{n-1} \mathbf{A}_0 + \boldsymbol{\pi}_n \mathbf{A}_1 + \boldsymbol{\pi}_{n+1} \mathbf{A}_2 = \mathbf{0}.$$

\mathbf{R} を次式を満たす非負行列とする.

$$\mathbf{A}_0 + \mathbf{R} \mathbf{A}_1 + \mathbf{R}^2 \mathbf{A}_2 = \mathbf{0}.$$

(12) より, $n \geq 2$ については

$$(13) \quad \boldsymbol{\pi}_n = \boldsymbol{\pi}_1 \mathbf{R}^{n-1}$$

と表せる. \mathbf{A}_2 は正則であるから方程式 (10) より $\boldsymbol{\pi}_1 = -\boldsymbol{\pi}_0 \mathbf{B}_0 \mathbf{A}_2^{-1}$. また, (13) より $\boldsymbol{\pi}_2 = \boldsymbol{\pi}_1 \mathbf{R}$. これらの関係を方程式 (11) に代入することにより

$$(14) \quad \boldsymbol{\pi}_0 \mathbf{B}_0 \mathbf{A}_2^{-1} (\mathbf{E} + (\mathbf{E} + \mathbf{R}) \mathbf{A}_2) = \mathbf{0}$$

を得る. 一方確率条件 $\boldsymbol{\pi} \mathbf{1} = \mathbf{1}$ より

$$(15) \quad \boldsymbol{\pi}_0 (\mathbf{E} - \mathbf{B}_0 \mathbf{A}_2^{-1} (\mathbf{E} - \mathbf{R})^{-1}) \mathbf{1} = \mathbf{1}$$

を得る. (14), (15) を連立させることにより $\boldsymbol{\pi}_0$ が得られ, さらにそれを用いてすべての状態確率 $\boldsymbol{\pi}$ を定めることができる.

解析手順 (f): 定常確率 $\boldsymbol{\pi}$ を用いて各種の性能評価指標を計算することができる. 一例として, 下位層のハードウェア資源使用率を求めてみる.

$$P(\mathbf{k}) = P(\text{下位層の状態} = \mathbf{k})$$

とする. $\boldsymbol{\pi}_0 = (p(0, 0), p(1, 0), p(2, 0))$ であることおよび $\boldsymbol{\pi}^* = \sum_{n=1}^{\infty} \boldsymbol{\pi}_n$ とすれば

$$\boldsymbol{\pi}^* = (p(0, 1), p(1, 1), p(2, 1))$$

であることから, $\tau_i^0 = (\tau_i(0, 0), \tau_i(1, 0), \tau_i(2, 0))$, $\tau_i^* = (\tau_i(0, 1), \tau_i(1, 1), \tau_i(2, 1))$ として, 資源 i の使用率 a_i は

$$a_i = (\pi_0, \tau_i^0) + (\boldsymbol{\pi}^*, \tau_i^*), \quad i = 1, 2, 3$$

の形に求められる. ここで, (\mathbf{a}, \mathbf{b}) は \mathbf{a}, \mathbf{b} の内積とする. なお, 率行列 \mathbf{R} を解析的に求めることは困難であるが数値計算によりその値を定めることは容易である.

性能評価指標の数値計算例として, トランザクションの到着率 λ を変化させたときの下位層のノード, すなわちハードウェア資源, の使用率の変化を図 3.3 に示す. 図に示されるように, 上位層のプロセス (ステーション) ごとの特性 (下位層への負荷) が異なるため, トランザクションの到着率が増加しているにもかかわらず, 資源使用率が減少するという結果が得られる.

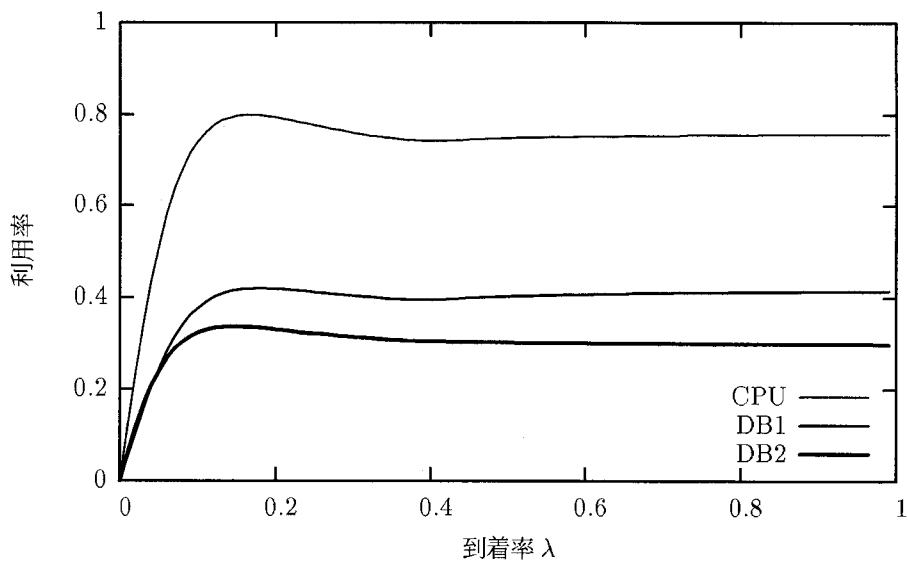


図3: ハードウェア資源使用率

4 まとめ

本稿では、即時・待時混在システムを2層型待ち行列網モデルでモデル化し解析をおこなう方法を示した。さらに2層型待ち行列網モデルをもちいることにより、ソフトウェア資源とハードウェア資源の階層的な関係を的確にモデル化することができ、その結果負荷の増加に伴い資源の使用率が減少することがあり得るという数値計算結果を示すことができた。今後は2層型待ち行列網モデルの応用範囲の拡大、より効率的で精度のよい近似解法の開発などが課題としてあげられる。

参考文献

- [1] S. Balsamo and G. Iazeolla, An extension of Norton's theorem for queueing networks, *IEEE Trans. Software Eng.* **SE-8** (1982) 298-305.
- [2] B. Baynat and Y. Dallery, A unified view of product-form approximation techniques for general queueing networks, *Perf. Eval.* **18** (1993) 205-224.
- [3] P. Buchhold, A class of hierarchical queueing networks and their analysis, *Queueing Systems* **15** (1994) 59-80.
- [4] K.M. Chandy, U. Herzog, and L. Woo, Approximate analysis of general queueing networks, *IBM J. Res. Dev.* **19** (1975) 43-49.
- [5] K.M. Chandy, U. Herzog, and L. Woo, Parametric analysis of queueing networks, *IBM J. Res. Dev.* **19** (1975) 36-42.
- [6] P.A. Jacobson and E.D. Lazowska, Analyzing queueing networks with simultaneous resource possession, *C.ACM* **25** (1982).
- [7] A. Kaieda, *et al.*, Analysis and measurement of the effect of kernel locks in SMP systems, *Concurrency and Computation* **13** pp.141-152(2001).
- [8] 笠原正治, 田村康生, 高橋豊, オーバレイネットワーク技術の動向と二層型待ち行列網モデルによるエンド・ツウ・エンド遅延解析, 待ち行列研究部会資料, 2006年4月.
- [9] T. Kawanishi and I. Kino, Performance Evaluation Models with Two-Layer Queueing Networks, *IFORS 2005*, Hawaii.
- [10] 川西俊之, 紀一誠, 即時・待時混在システムの階層的な性能評価モデル, 日本OR学会2004年度秋季研究発表会アブストラクト集, pp.8-9(2004).
- [11] 紀一誠, 待ち行列ネットワーク, 朝倉書店, 2002.
- [12] I. Kino, Two-layer queueing networks, *JORSJ*, **40**, (2) pp.163-185(1997).
- [13] 木下俊之, 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, 数理モデル化とその応用 (2001).
- [14] T. Kurasugi and I. Kino, Approximation methods for two-layer queueing models, *Perf. Eval.* **36-37**, pp.55-70(1999).
- [15] 蔵杉, 他, 待ち行列網モデルによる smp 型システムの性能予測手法とその精度検証, SWoPP'99, 計算機アーキテクチャ研究会報告, 下関, pp.97-102(1999).
- [16] 牧本直樹, 待ち行列アルゴリズム-行列解析アプローチ, 朝倉書店 (2001).
- [17] J. A. Rolia and K.C. Sevcik, The method os layers, *IEEE Trans. Software Eng.*, **21** (9), pp.689-700(1995).
- [18] C.H. Sauer, Approximate solution of queueing networks with simultaneous resource possession, *IBM J. Res. Dev.* **25**(6) (1981) 894-903.