密結合マルチプロセッサにおけるソフトウェアリソース競合モデルに関する一考察†

根岸和義†† 木下俊之†† 米田茂††

密結合マルチプロセッサ (TCMP) における単一排他ソフトウェアリソースの競合を要因とした性能評価を行った．評価はプロセスおよびプロセッサの状態の組合せを一つの状態として確率解析モデルの平衡方程式を解くことにより実施した．対象となるリソースを，競合発生時に別のプロセスに切り替えを許すもの（ディスパッチ可リソース）と，許さないもの（ディスパッチ不可リソース）に分類し，後者のプロセッサ台数N台における，プロセッサ1台に対する相対処理能力を求めた．このモデルの特徴は，従来評価が困難であったプロセス切り替えオーバヘッドを含む性能を評価できる点である．さらに，プロセス切り替え方式として，リソース使用優先方式と，プロセス切り替え最少方式をモデル化し，与えられたリソース占有率，およびプロセス切り替えオーバヘッドに対して処理能力を比較評価し，より能力の高い方式を選択することを可能とした．

1. まえがき

近年，オンラインシステムにおけるプロセッサパワーのニーズが増大しており，単一プロセッサでは処理不可能なパワーが要求されている．これに対処するために，複数のプロセッサを組み合せたマルチプロセッサシステムによりオンラインシステムを構成することが必要になってきた．マルチプロセッサシステムには，主メモリを共用する密結合マルチプロセッサ (TCMP) と，共用しない疎結合マルチプロセッサ (LCMP) がある．我々は TCMP を対象として性能評価方策を検討した．

オンラインシステムを性能評価する手段として，実測[7] およびシミュレーションがある．従来，我々はそれらによる評価を実施してきたが，コストが大で，性能要因の切り分け，およびその変更による改善度の評価分析が困難であるという問題があった．そこで我々は，解析モデルによる原理的検討を行うことにした．オンラインシステムを性能評価するための解析モデルにはプロセッサ処理能力とトランザクションの到着率による単純待ち行列モデル[1)-3)]，多重処理と入出力処理をシステム全体として評価するセントラルサーバモデル[4),5)]，リソースを2次的なサーバで近似する近似的解析モデル[6)] 等があった．これらの方法では，リソース競合時のプロセス切り替えオーバヘッドを評価することは困難であった．対象とするリソースも，DB のレコード等，プロセッサとは独立に占有されるものが主であった．

マルチプロセッサのオンラインシステムの性能評価を行う上での最大の性能要因は，各プロセッサで動作するプロセスの処理シリアライズのためのリソースの競合，およびそれに対処するための制御オーバヘッドである．我々は，上記リソース競合を評価するために，解析モデルによりリソースを意識したプロセッサおよびプロセスの状態をモデル化し，リソース競合による状態遷移を平衡方程式により表現する方式を提案した．そして，リソース競合時のプロセス切り替え中の状態を意識し，従来解析が困難であったプロセス切り替えのオーバヘッドをモデルに反映させた．さらに，プロセス切り替え制御方式による性能差を評価し，オーバヘッドの大きさに対応した制御方式の選択を可能とした．

本論文で対象とするシステムは，等質のプロセッサおよび等質のプロセスからなる．メモリ共用の密結合マルチプロセッサシステム (TCMP) とする．対象となるリソースは，単一の排他リソースに限定して考察を行う．解析モデルは，プロセッサ状態とプロセス状態を共にモデル化する．処理時間は指数分布に従うとし，プロセッサの状態遷移をマルコフ状態遷移モデルにより解析する．処理能力は限界スループットにより評価する．

2. リソース評価の前提条件

本評価モデルのリソースおよびモデル化の前提条件を以下に示す．

2.1 リソースの分類

評価対象とする処理逐次化のためのリソースは，そ

† A Study of Software Resource Contention Models on Tightly Coupled Multiprocessors by KAZUYOSHI NEGISHI, TOSHIYUKI KINOSHITA and SHIGERU YONEDA (Hitachi, Ltd.).

†† (株)日立製作所

の競合時の制御方式により，以下の2種類に分類される．

（1） ディスパッチ不可リソース

リソース競合時に，プロセスを他のプロセスへ切り替えることができないリソースである．主にディスパッチャの内部リソース（スケジュールキュー等）がこれに該当する．そのほかに，リソースの占有時間が短いためプロセス切り替えを行うと逆にオーバヘッドが増大するタイプのリソース（メモリ管理テーブル等）もこれに含まれる．

（2） ディスパッチ可リソース

リソース競合時に，他のプロセスを代わりに実行することができるリソースである．このリソースは，比較的長時間占有されるため，プロセス切り替えによりプロセッサの有効活用をはかる必要がある．

ディスパッチ可リソースは，さらに下記の二つに分けることができる．

（a） 長期保持ディスパッチ可リソース

プロセスが入出力処理等で一時的に中断しても，保持し続けるリソースである．例えば，データベースのレコードはその一つである．

（b） 短期保持ディスパッチ可リソース

プロセスが割込み等の他からの強制的な手段により中断された場合を除き，その実行中のみ保持するリソースである．例えば，データベースのバッファ管理テーブル，ジャーナルバッファ等はこのタイプである．

これらのリソースのうち，ディスパッチ不可リソースに関しては，プロセス切り替えオーバヘッドを生じることはない．また，その性能評価は，Madnick[8]により，本論文と同様のマルコフ状態遷移モデルを用いてなされている．長期保持ディスパッチ可リソースに関しては，リソース占有時間に比較して，プロセス切り替え時間が短いため，プロセス切り替えオーバヘッドは問題とならない．

本報告では，残った短期保持ディスパッチ可リソース（以後，ディスパッチ可リソースと呼ぶ）を対象として，性能評価を行う．

2.2 モデル化の前提条件

本論文で扱うシステムは以下のことを前提条件とする．

（1） プロセスM個，プロセッサN台は各々等価でありモデル内では区別しない．また，$M>N$とする．

（2） 各プロセッサはリソース競合が発生しない限り任意のプロセスを実行できる．

（3） 各プロセッサは処理可能なプロセスのある限りプロセスの処理を行う．処理可能なプロセスが存在しても，当該プロセスの処理を開始せずプロセッサが空転することはないと仮定する．

（4） プロセスは単一の排他リソースの要求，非要求処理を繰り返す．プロセスからのリソース要求はランダムに発生する．

（5） プロセスはプロセッサで実行中のみリソースを占有することができる．

（6） プロセスの開始，終了，入出力によるプロセスの入れ替わりの回数はリソース要求よりも少なく，無視できるものとする．本モデルではこれらの処理をモデル化の対象としない．プロセッサは，リソースの競合，解放以外の場合にはプロセスの実行を中断することはないものとする．

（7） プロセッサによるリソース非占有，占有，プロセス切り替えの時間は，それぞれ平均 $1/\mu_0$, $1/\mu_1$, $1/\mu_2$ の指数分布に従うとする．

（8） リソース競合時，下記の処理を行う．

リソースを占有中のプロセッサ以外は，当該リソースを占有していない，他のプロセスを探しこれを起動させる．もし，そのようなプロセスが存在しない場合，プロセッサは空転する．

3. プロセス切り替え制御方式

本報告では，以下に述べる2通りのプロセス切り替え方式に関して，その，選択基準を検討する．

リソース要求時および解放時の処理として，**表1**に示すいくつかの選択枝が考えられる．横線の部分は実現不可能な状態および不必要な空転を生じるため，選択することは考えられない．また，×印は選択可能であるが明らかに不必要なプロセス切り替えを生じる．丸印は選択の余地なく決定される状態を示す．したがって，選択の余地のあるのは，表の4行目である．これらに関して，表1の(1)，(2)の選択を行った場合を下記に示す．

（1） リソース使用優先方式

リソースを使用するプロセスを可能な限り優先して実行することにより，リソース競合を要因とするプロセッサの空転を最小限とする．

（2） プロセス切り替え最少方式

プロセスの切り替えオーバヘッドの削減を目的とし，プロセス切り替えを最少回数とする．

表 1 プロセス切り替え制御方式
Table 1 Control methods of process switch.

プロセッサのイベント	次の処理 / 条件		実行継続	他のプロセスと交代〔注2〕	アイドル〔注1〕
実行中のプロセスがリソースを要求	リソースは未使用		○	×	—
	リソースは他のプロセスが使用中	他にリソース非要求プロセスあり	—	○	—
		同上なし	—	—	○
実行中のプロセスがリソースを解放	他にリソース要求プロセスあり〔注3〕		(2)	(1)	—
	同上なし		○	—	—

〔注1〕 実行可能なプロセスのある限りプロセッサはアイドルしない.
〔注2〕 同種のプロセスは区別しない. したがって, 同種のプロセスの交代は考えない.
〔注3〕 アイドル中のプロセッサが存在するなら, そのうちの1台で下記のいずれかの処理を行う.
次の処理が実行継続の場合, リソース要求プロセスの1個を実行開始する.
次の処理が他のプロセスとの交代の場合, 元のプロセスの実行を引き継いで行う.

いずれも, リソース競合時の処理は同様であるが, リソース解放時の処理が異なる. 上記(2)を採用した場合, 特定のプロセスのみが実行され, プロセス処理にばらつきが生じるということが心配される. しかし, 実システムでは, リソース要求に比較すると回数は多くないが, プロセスの開始終了や入出力による中断が発生し, プロセスが入れ替わる. したがって, 特に応答時間に関して厳しい要求のある生産プロセス管理システムでもない限り, このことは実質上問題とはならないと考えられる. なお, システムは, 待ち状態のプロセスを相互に区別せず, 条件を満たす任意の一つを次のディスパッチ対象として, 実行することを特徴とする.

4. ディスパッチ可リソースの性能評価

本論文において性能評価に用いる状態遷移モデルに関して説明する. 各プロセッサの状態は, リソース非占有, リソース占有, アイドル (空転) のいずれかである. 各プロセスの状態はリソース非要求, リソース要求のいずれかである. ここで, リソース非占有状態は, リソース非要求のプロセスを実行する場合, リソース占有状態は, リソース要求のプロセスを実行する場合のプロセッサの状態である. システムにおけるプロセッサの状態は各状態のプロセッサの個数 (j_0, j_1) により, 区別される. ここで, j_0: リソース非占有のプロセッサ台数, j_1: リソース占有のプロセッサ台数である. アイドル状態のプロセッサ台数は $N-j_0-j_1$ である. また, システムにおけるプロセスの状態はリソース要求状態のプロセス数 i により区別される.

N 台のプロセッサ, M 個のプロセスにおけるシステムの状態は,

$$s(s_0, i) \quad s_0 \in \{(N, 0)\} \cup \{(N-j, 1) \mid j=1 \sim N\}, \quad (i=0 \sim M), \quad (4.1)$$

により表される. プロセッサは不要な空転をしないという前提では, 上記の組合せ以外の状態は存在しない.

$s(s_0, i)$ の平衡状態確率を $p(s_0, i)$ とする.

図 1 は本モデルの状態空間とリソース使用優先方式の状態遷移 (後述) を示す. 横軸はプロセッサ状態の

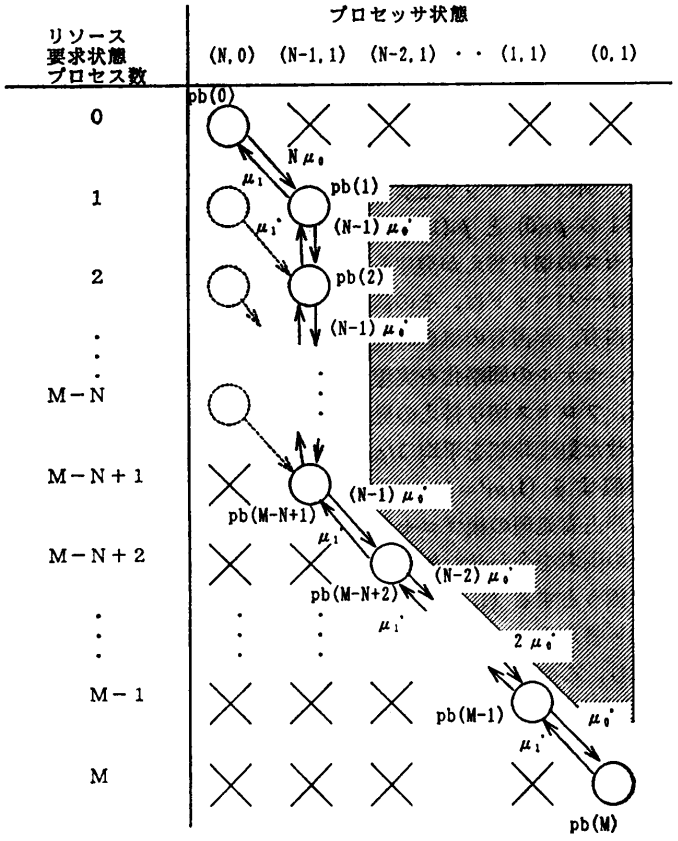

図 1 ディスパッチ可リソースに対する状態遷移モデル (リソース使用優先方式)
Fig. 1 State transition model of dispatchable resource (resource request preceded method).

組合せ，縦軸はリソース要求状態のプロセスの個数を表す．図において，×印はありえない状態の組合せを表す．また，斜線部分は，不必要な空転を生じる組合せである．

4.1 リソース使用優先方式の状態遷移モデル

図1は，リソース使用優先方式における，プロセッサとプロセスの状態遷移のモデルである．点線で表されている状態 $(s((N,0),i)\ i=1\sim M-2)$ は実線で表されている状態 $(s((N-1,)1)\ i=1\sim M-N)$ から可到達ではなくその平衡状態確率はゼロである．ここで，簡単のため残った状態に順序番号を付け，その平衡状態確率を次のように表す．

$$\begin{aligned}&p_b(0)=p((N,0),0),\\&p_b(i)=p((N-1,1),\ i)\quad(i=1\sim M-N),\\&p_b(i)=p((M-i,1),\ i)\quad(i=M-N+1\sim M).\end{aligned}\tag{4.2}$$

図1より，リソース要求で待ち状態のプロセスが存在する状態で（リソース要求状態のプロセス数$=2\sim M$），実行中のプロセスのリソース占有が終了したなら，当該プロセスは上記プロセスの一つに切り替えオーバヘッド時間の後，プロセッサを渡し，待ち状態となる．ただし，リソース要求状態のプロセス数$>M-N+2$ では，自プロセスもリソース非占有状態となって，別のプロセッサで実行を続けることができる．図1の $p_b(0)$ と $p_b(1)$ の間の遷移以外の遷移ではプロセスの切り替えが発生している．プロセス切り替えのオーバヘッドは，その直前に行われていたリソース占有，非占有の処理時間に合成して考えることにより，モデルの簡略化を実現した．リソース占有処理の後，プロセス切り替えの発生する場合，その両者を合わせた処理時間を平均 $1/\mu_1'$ 時間の指数分布に従うと近似する $(1/\mu_1'=1/\mu_1+1/\mu_2)$．同様にして，リソース非占有処理の後プロセス切り替えの発生する場合，その両者を合わせた処理を平均 $1/\mu_0'$ 時間の指数分布に従うとする $(1/\mu_0'=1/\mu_0+1/\mu_2)$．なお，プロセスの切り替えオーバヘッドを含まない場合は，$1/\mu_2=0$ となり，この時，$1/\mu_1'=1/\mu_1, 1/\mu_0'=1/\mu_0$ である．定常状態における既約な有限マルコフ過程として，隣合う状態のカットセットを考えた場合，その通過フローに関する平衡方程式は下記となる．

$$\begin{aligned}&p_b(0)N\mu_0=p_b(1)\mu_1,\\&p_b(i)(N-1)\mu_0'=p_b(i+1)\mu_1'\quad(i=1\sim M-N),\\&p_b(i)(M-i)\mu_0'=p_b(i+1)\mu_1'\\&\qquad(i=M-N+1\sim M-1).\end{aligned}\tag{4.3}$$

ここで，状態0と状態1の間の遷移では，プロセス切り替えが発生しないので，第1式では，μ_0 および μ_1 が使用され，その他の式では μ_0' および μ_1' が使用されている．本論文のモデルでは，このような各状態間の平衡方程式における処理時間の使い分けにより，プロセスオーバヘッドの有無を含む処理能力の評価を実現した．さらに，式の見通しを良くするため，

$$X'=\frac{\mu_0'}{\mu_1'},\quad X=\frac{\mu_0}{\mu_1},\tag{4.4}$$

として，これを解くと，

$$p_b(0)=\frac{1}{G_b},\tag{4.5}$$

$$p_b(i)=\frac{XX'^{i-1}(N-1)^{i-1}N}{G_b}\qquad(i=1\sim M-N+1),\tag{4.6}$$

$$p_b(i)=\frac{XX'^{i-1}(N-1)^{M-N}{}_N\mathrm{P}_{i+N-M}}{G_b}\qquad(i=M-N+2\sim M),\tag{4.7}$$

となる．ここで，

$$\begin{aligned}G_b=1&+\sum_{i=1}^{M-N+1}XX'^{i-1}(N-1)^{i-1}N\\&+\sum_{i=M-N+2}^{M}XX'^{i-1}(N-1)^{M-N}{}_N\mathrm{P}_{i+N-M}\end{aligned}\tag{4.8}$$

である．これよりプロセッサ1台の単位時間当りの処理能力を1とした場合の，リソース占有処理およびリソース非占有処理のみを加えた処理能力 Pwb を考える．Pwb は，プロセッサ占有率$\times N$ からプロセス切り替えのオーバヘッド分を除いた，有効な処理能力の相対値を示す．またこの時，システムが定常状態となっていることから，Pwb 中のリソース占有処理の割合は後述のロック占有率 b に等しい．

単位時間当りアイドル状態のプロセッサは0，ディスパッチ処理を伴わないリソース非占有および占有処理を行うプロセッサは1の処理能力があると考えられる．同様にして，ディスパッチ処理を伴うリソース非占有処理を行うプロセッサは μ_0'/μ_0，同じくリソース占有処理を行うプロセッサは μ_1'/μ_1，の処理能力がある．したがって，

$$\begin{aligned}Pwb=&Np_b(0)+\left(1+(N-1)\frac{\mu_0'}{\mu_0}\right)p_b(1)\\&+\left(\frac{\mu_1'}{\mu_1}+(N-1)\frac{\mu_0'}{\mu_0}\right)\\&\cdot(p_b(2)+\cdots+p_b(M-N))\end{aligned}$$

$$+\left(\frac{\mu_1'}{\mu_1}+(N-1)\frac{\mu_0'}{\mu_0}\right)$$
$$\cdot p_b(M-N+1)\cdots+\frac{\mu_1'}{\mu_1}p_b(M), \quad (4.9)$$

となる．

ロック占有率 b を，

$$b=\frac{\frac{1}{\mu_1}}{\frac{1}{\mu_0}+\frac{1}{\mu_1}}, \quad (4.10)$$

とすると，

$$X=\frac{b}{1-b}, \quad (4.11)$$

である．

また，プロセス切り替えオーバヘッドの割合 b_0 を，

$$b_0=\frac{\frac{1}{\mu_2}}{\frac{1}{\mu_0}+\frac{1}{\mu_1}}, \quad (4.12)$$

とすると，

$$X'=\frac{b+b_0}{1-b+b_0}, \quad (4.13)$$

である．$N=2$, $M=20$, $b_0=1/8$, $1/4$, $1/2$ の場合の b, b_0 に対する Pwb/N (Pwb をプロセッサ1台当りに正規化した値) を図3(実線) に示す．

4.2 プロセス切り替え最少方式の状態遷移モデル

図2は，プロセス切り替え最少方式に対するプロセッサとプロセスの状態遷移モデルである．本方式では，○印以外の状態はその状態への遷移がないため状態確率は定常状態ではゼロである．前節と同様に平衡状態確率 $p_c(i)$ を，

$$p_c(M-N)=p((N,0),M-N),$$
$$p_c(M-N+1)=p((N-1,1),M-N+1),$$
$$\cdots$$
$$p_c(M)=p((0,1),M), \quad (4.14)$$

とする．

本モデルにおいて，定常状態における既約な有限マルコフ過程としてカットセットを考えた時，これらの状態間の平衡方程式は，

$$p_c(M-N+1)\mu_1=p_c(M-N)N\mu_0,$$
$$p_c(M-N+i)\mu_1'=p_c(M-N+i-1)\cdot(N-i+1)\mu_0'$$
$$(i=2\sim N), \quad (4.15)$$

となる．

前節と同様に，状態 $M-N+1$ と状態 $M-N$ の間の状態遷移では，プロセス切り替えが生じないので，第1式では処理時間として μ_1 および μ_0 を使用し，第2式ではプロセス切り替えが生じるため，μ_1' および μ_0' を使用している．

これを解いて，前節と同様にプロセッサ1台の処理能力を1とした場合のリソース占有処理とリソース非占有処理を加えた，相対的な処理能力 Pwc を求める．(4.9)と同様にして，

$$Pwc=Np_c(M-N)+\left(1+(N-1)\frac{\mu_0'}{\mu_0}\right)p_c(M-N+1)$$
$$+\sum_{i=2}^{N}\left(\frac{\mu_1'}{\mu_1}+(N-i)\frac{\mu_0'}{\mu_0}\right)p_c(M-N+i), \quad (4.16)$$

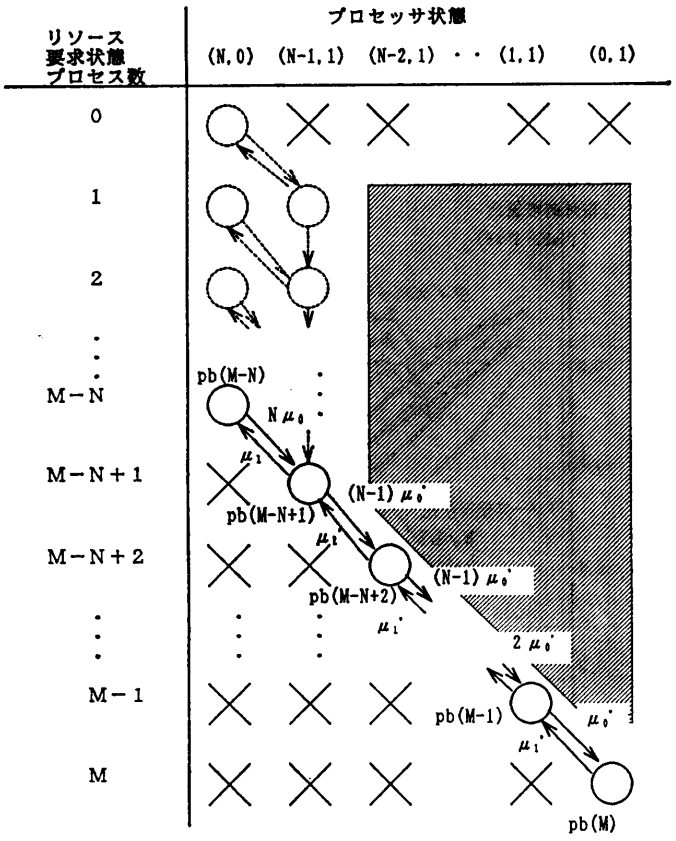

図2 ディスパッチ可リソースに対する状態遷移モデル（プロセス切り替え方式）

Fig. 2 State transition model of dispatchable resource (minimum process switch method).

となる．ここで，

$$p_c(M-N)=\frac{1}{G_c}, \tag{4.17}$$

$$p_c(M-N+1)=NXp_c(M-N), \tag{4.18}$$

$$p_c(M-N+i)={}_N\mathrm{P}_iXX'^{i-1}p_c(M-N) \quad (i=2\sim N). \tag{4.19}$$

$$G_c=1+\sum_{i=1}^{N}{}_N\mathrm{P}_iXX'^{i-1}, \tag{4.20}$$

である．4.1節と同様に，$N=2$, $M=20$, $b_0=1/8$, 1/4, 1/2 の場合の Pwc/N (Pwc をプロセッサ1台当りに正規化した値) を**図3**（点線）に示す．

リソース使用優先方式とプロセス切り替え最少方式の選択のため，下記の評価関数 Fd を考える．

$$Fd=Pwb-Pwc \tag{4.21}$$

N, M, b, b_0 が与えられた時，この Fd を求め，その値が正ならリソース使用優先方式を，負ならプロセス切り替え最少方式を選択すれば良い．**図4**に，$N=2$, $M=20$ の時の，b, b_0 に対する $Fd=0$ の曲線を示す．図4において，例えば，b, b_0 によって求めた座標が境界線の左上なら $Fd<0$ であり，上記のようにプロセス切り替え最少方式を選択すべきである．また，座標が右下なら同様にして，リソース使用優先方式を選択すべきであることがわかる．

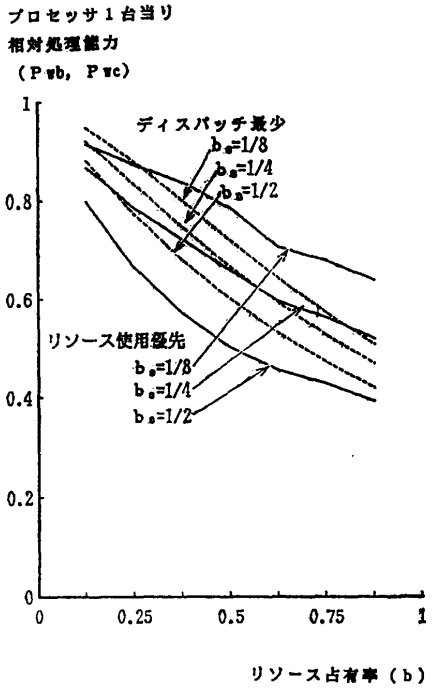

図3 ディスパッチ可リソース競合時の相対処理能力
Fig. 3 Relative performance with contention of dispatchable resource.

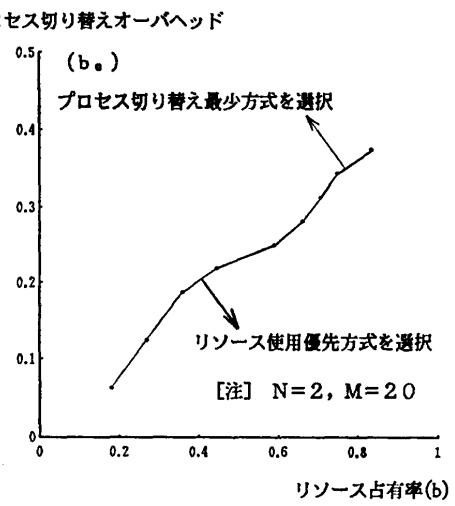

図4 制御方式評価関数による選択
Fig. 4 Selection of control methods by the control method evaluating function.

4.3 結果のまとめ

本論文の評価モデルにより，リソース占有率，およびプロセス切り替えオーバヘッドを与えられた場合の処理能力を定量的に評価することができる．また，二つの制御方式（リソース使用優先方式とプロセス切り替え最少方式）を比較し，いずれが処理能力があるかを定量的に評価し選択する方策を示した．

5. むすび

単一排他リソース（処理の逐次化リソース）に関するTCMPの処理能力（スループット）を状態遷移モデルにより評価した．リソースの種別をディスパッチ不可リソース，ディスパッチ可リソースに分け，後者のプロセス切り替えオーバヘッドのある場合のプロセッサN台における近似モデルを考えた．このモデルを用いることにより，リソース占有率およびプロセス切り替えオーバヘッドをもとにして，2種類のプロセス切り替え制御方式（リソース使用優先方式，プロセス切り替え最少方式）に対する処理能力を定式化し，評価関数により，これらの選択を行うことを可能とした．

今後は，プロセス切り替えオーバヘッドのある場合の最適制御方式に関する検討を行う予定である．

謝辞 解析モデルによる定式化に関してご指導，ご助言をいただいた電気通信大学亀田壽夫教授に感謝いたします．本研究の機会を与えていただいた(株)日立製作所システム開発研究所の堂免信義所長，久保隆重

副所長に感謝いたします．また，マルチプロセッサ性能評価方式に関してご指導，ご協力をいただいた同・ソフトウェア工場の方々に感謝いたします．

参考文献

1) Kleinrock, L.: 待ち行列システム理論(上)，マグロウヒル好学社 (1979).
2) 森村英典，大前義次：応用待ち行列理論，日科技連 (1975).
3) Ni, L. M. and Wu, C. E.: Design Tradeoffs for Process Scheduling in Shared Momory Multiprocessor Systems, *IEEE Trans. Softw. Eng.*, Vol. 15, No. 3, pp. 327-334 (1989).
4) Buzen, J. P.: Computational Algorithms for Closed Queueing Networks with Exponential Servers, *Comm. ACM*, Vol. 16, No. 9, pp. 527-531 (1973).
5) Baskett, F., Chandy, K. M. and Muntz, R. R.: Open Closed and Mixed Networks of Queues with Different Classes of Customers, *J. ACM*, Vol. 22, No. 2, pp. 248-260 (1975).
6) Thomasian, A.: Performance Evaluation of Centralized Database with Static Locking, *IEEE Trans. Softw. Eng.*, Vol. SE-11, No. 4, pp. 346-355 (1985).
7) Kobayashi, M.: An Empirical Study of Task Switching Locality in MVS, *IEEE Trans. Comput.*, Vol. C-35, No. 8, pp. 720-731 (1986).
8) Madnick, S. E.: Multi-processor Software Lockout, *Proc. of ACM 23rd National Conf.*, pp. 19-24 (1968).

(平成元年 8 月 31 日受付)
(平成 2 年 11 月 13 日採録)

根岸 和義 (正会員)

1973 年東京工業大学電子物理工学科卒業．1975 年同大学院修士課程卒業．同年(株)日立製作所入社．以来同社システム開発研究所において分散データベース，オンラインシステム，マルチプロセッサシステムの研究に従事．特に，性能評価および性能向上策の研究に興味を持つ．電子情報通信学会，ACM，IEEE 各会員．

木下 俊之 (正会員)

昭和 27 年生．昭和 50 年東京工業大学数学科卒業．昭和 52 年東京大学大学院理学系数学科修士課程修了．同年(株)日立製作所入社．システム開発研究所にて，性能評価，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．現在，同研究所第 3 部主任研究員．ACM 会員．

米田 茂 (正会員)

昭和 22 年生．昭和 45 年名古屋工業大学工学部計測工学科卒業．昭和 45 年日立製作所情報システム研究所入所．現在，同社システム開発研究所企画室に勤務．入社以来，データベースコントロールシステムの開発，性能評価，高信頼化機能，分散機能，エンドユーザインタフェース，等 DB/DC 関連の研究開発に従事．