マルチメディア通信と分散処理 29－2
（１９８６．５．１５）

静的二相ロッキングアルゴリズムの性能解析

任　景飛、　高橋　豊、　長谷川　利治
京都大学工学部

同時実行制御アルゴリズムはデータベース管理システムが複数ユーザのデータベースへの同時アクセスを可能にするものである。本稿では静的な二相ロッキングアルゴリズムを用いるデータベースにおいて、トランザクションが指数間隔で発生され、アクセス集合が一様分布に選ばれる環境での平均応答時間などの性能評価量を求める。ブロックされたトランザクションのアクセス集合が過去の履歴とは独立に選ばれるという仮定のもとで、システムの確率的挙動を二次元のマルコフ過程に帰着し、Neutsの定理を基に、マルコフ過程の厳密解を求める。数値結果とシミュレーション結果の比較から仮定の妥当性が示される。

Performance Analysis of a Concurrency Control Method with Static Locking Algorithm

Jing Fei REN　　Yutaka TAKAHASHI　　Toshiharu HASEGAWA
Department of Applied Mathematics and Physics, Kyoto University

The performance of the database system with static locking algorithm highly depends on the level of the contention for hardware as well as data resource. Transactions are generated according to a Poisson process and their access set are uniformly distributed in the database considered. Average response time is a main performance measure. Under an assumption of independent resampling, the system is modeled as a Markov process with infinite states. The matrix geometric method developed by Neuts is used in analysis. Numerical results are presented to give an insight into the trade off among system parameters.

1まえがき

データベースシステムにおける同時実行制御法はあたかも各ユーザが単独にデータベースを利用しているかのごとく内容の首尾一貫性を保ちながら、複数のユーザのデータベースへの同時アクセスを可能にするものである。この目標を達成するためには同時にデータベースへアクセスするトランザクション間の干渉を避ける必要がある。過去十数年の間に、数多くの同時実行制御アルゴリズムが提案改良された。これらのアルゴリズムを大別すると、三つのクラスにわけられる。その一つは時刻印に基づく方法である。この方法で、トランザクションの直列可能の順序が予め決められ、トランザクションが割り当てられた時刻印に従って実行される。二番目の方法は楽観的なアルゴリズムである。このアルゴリズムでは、到着したトランザクションがとにかく実行され、データの首尾一貫性を保てない恐れが発生したときのみ関係のあるトランザクションがアボートされ、最初から再実行される。三番目の方法は今よく使われている二相ロッキング方法である。このアルゴリズムでは、実行の全過程において、首尾一貫性を崩さないと予め確認されたトランザクションだけが処理を開始される。その結果、必要以上に処理の開始が遅らされる可能性があるために、悲観的なアルゴリズムともいえる。

二相ロッキングアルゴリズムでは、トランザクションがあるデータ項目にアクセスする前にこの項目のロックを保有しなければならない。また、いったんトランザクションがロックを解放し始めると、決してロッキングの要求を出さない。本稿では専有ロックだけを考えることにする。一つのデータ項目が任意の時間において、高々一つのトランザクションでロックがなされる。そうすることにより、トランザクション間の干渉を避け、データベースの首尾一貫性を保つことが可である。

二相ロッキングアルゴリズムはロックの獲得方式により、動的なロッキングと静的なロッキングに大別できる。動的なロッキングでは、ロックはトランザクションの実行段階で、必要になった時点で、一つずつ要求される。動的なロッキングの利点はデータ資源の有効利用である。

本稿では静的な二相ロッキングアルゴリズムを考察する。このアルゴリズムは次ぎにのように機能する。トランザクションは予めアクセスしようとする全てデータ項目を宣言する。もし、これらのデータ項目に一つでもほかのトランザクションからのロックがなされていなければ、このトランザクションはロッキングの権利を獲得し、全てのアクセス項目のロッキングを完了して始めて、実行が開始される。もしアクセス項目の一つにでも、ほかのトランザクションからのロックがなされていれば、このトランザクションはブロックされ、ロックをかけずに、バッファでロックの解放を待つ。このようなロッキングメカニズムにより、アクセス項目集合に共通の部分がないトランザクション同志だけは同時に実行される。トランザクションは実行が終わると、保有していた全てのロックを解放して、ブロックされているトランザクションにロッキングチャンスを与える。

静的なロッキングアルゴリズムに基づいて運用されているデータベースシステムの確率的挙動の解析に関してはPotierとLeblanc[1]はこの分野の先駆者である。彼らはトランザクションが有限個数の端末により、指数間隔で発生されるシステムを考察した。本稿で考察するのは基本的に彼らのモデルを拡張したものである。一方、MitraとWeinberger[2]はポアソン源からの到着したトランザクションがもしほかの実行されているトランザクションと衝突したら、直ちに廃棄されるいわゆる呼損モデルを解析した。もう一つの仮定はシステムに無限のサーバがいる仮定である。このモデルはかなり特別な場合に対応しているが、任意のアクセス集合サイズと任意のロック保有時間の分布を許すなどいくつかの面で、柔軟性を持っている。彼らはマルコフ過程の可逆性を生かして、積の形の解で、解析を厳密に行っている。また、MorrisとWong[3]はシステムに常に充分な数のトランザクションがある飽和システムを解析した。ロッキング過程をモデル化する際の工夫により、彼らはシステムのスループットの良い近似を与えている。また、Thomasian[4]はロックの要求と衝突の発生が決定的なシステムを解析した。本質的に、このモデルはいくつかの支配的なトランザクションのクラスだけを取り扱うときに使われる。

本稿では、静的なロッキングアルゴリズムを用いるシステムにおいてトランザクションがポアソン過程に従って生成され、ロックの要求が全項目集合に一様分布する環境での性能評価を試みる。性能評価はブロックされたトランザクションが再度ロックを要求するときに過去の要求に関して無記憶的に再び選ばれるという仮定のもとで、行われる。評価の主な尺度は平均応答時間で、最大スループットにも言及する。

本文は５節からなる。第２節にシステムとアルゴリズムを記述し、モデルを提案する。第３節にNeuts[5]の理論を使って、モデルを解析し、性能評価をする。数値結果は第４節で与えられ、第５節はまとめである。

２　システムとモデル

本文で考察するデータベースはユーザからのトランザクションでアクセスされるデータ項目の集合と考えられる。トランザクションの到着過程はポアソン過程に従い、要求されるサービス時間は指数分布に従うとする。各トランザクションがアクセスするデータ項目数は定数で、データベースに一様分布で選ばれる。もし、同時実行可能なトランザクション数に余裕があるならば、トランザクションは到着時点において、ロッキングを行う。ロックが保証されるトランザクションは無視できるほど短い時間に、全てのアクセスする項目にロックをかけてから、実行される。ロックが保証されないトランザクションはブロックされ、待ち行列の最後で待つ。同時実行可能なトランザクション数に余裕がない場合には、到着は単にブロックされ、待ち行列の先頭で待つ。トランザクションの処理が完了すると、持っていた全てのロックを解放するから、ブロックされたトランザクションは待ち行列の先頭から順番に再びロッキングの試みをする。もし、試み中、あるトランザクションがロッキングに失敗したら、または、実行可能なトランザクション数に余裕がなくなったら、以後のロッキングは中止され、次の処理完了時点まで、延ばされる。ロッキングに失敗したトランザクションは待ち行列の最後にもどる。上記のような複雑なシステムを厳密にモデル化するためには、ブロックされているトランザクションのアクセス集合を状態変数として記述する必要がある。しかし、通常のデータベースは膨大なデータ項目数を有することを考慮すると、定式化はほとんど不可能である。そこで、本稿では、解析を可能にするため一つの仮定を設ける。類似の仮定はほかの論文[1,3,6,7]にでも見られる。

仮定：トランザクションがロッキング（再ロッキング）する度に、このトランザクションのアクセス集合と実行されているトランザクションのアクセス集合は過去の履歴に独立に、データベースから選ばれる。

この仮定のもとで、システムは図１のようにモデル化できる。図の中に、Ｆはブロックされたトランザクションにより形成される待ち行列である。実行を完了されたトランザクションの離脱時点でのみオンになるスイッチk_2はブロックされたトランザクションがロッキングを要求するタイミングを制御している。マルチプログラミングレベルmに対応して、m人のサーバが同時に使える。システム内では、データ項目に対する競争のほかに、サーバに対する競争も存在する。実行されているトランザクションの数がマルチプログラミングレベルと等しい時、到着したトランザクションは待ち行列の先頭に置かれ、ロッキング要求は次の離脱点に延ばされる。

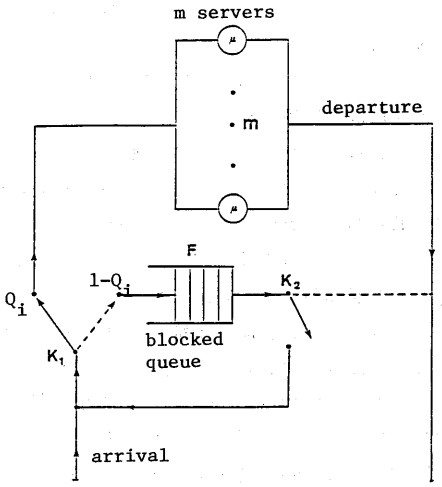

図1　システムモデル

さらに、モデルのパラメータを次のように定義する。

D：データベースにあるデータ項目総数。

S：トランザクションのアクセス集合にあるデータ項目数。

λ：トランザクションの到着率。

μ：サーバのサービス率。

m：サーバの数。

３　解析

前の仮定により、時刻ｔでの実行されているトランザクション数i(t)と待ち行列の長さj(t)を状態変数として選ぶとシステムの確率的挙動はマルコフ過程で記述される。平衡状態だけを考えるから、i(t)とj(t)の平衡状態での値iとjに注目する。実行されているトランザクション数がiのとき、トランザクションがロッキングに成功する確率はjに依存せずに、次のようになる。

$$Q_i = \binom{D-iS}{D} \Big/ \binom{D}{S} \quad (0 \leq i < m)$$

$$Q_i = 0 \quad (i \geq m) \qquad (1)$$

また確率$Q^k_{i,j}$をトランザクションの離脱の直後の状態が(i,j)である条件のもとで、k個のブロックされたトランザクションがロッキングに成功する確率であると定義する。(1)で定義されたQ_iを用いて、$Q^k_{i,j}$は

$$Q^k_{i,j}=\begin{cases} Q_iQ_{i+1}\cdots Q_{i+k-1}(1-Q_{i+k}) & \text{if } i+k<m \text{ and } j>k \\ Q_iQ_{i+1}\cdots Q_{i+k-1} & \text{if } i+k<m \text{ and } j=k \text{ or } i+k=m \text{ and } j\geqq k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

と書ける。

注意：全てのj>kに対して、$Q^k_{i,j}$は$Q^k_{i,k+1}$と等しい。この性質は後に使われる。

以上定義されたパラメータと確率Q_i、$Q^k_{i,j}$を用いて、遷移確率行列Gが求まる。ここで、無限次元の線形方程式を解く問題に直面する。標準的なアプローチは適当なサイズのところで行列Gを切り捨て、有限次元の線形方程式の解問題に直す方法である。しかし、この場合には、計算精度と計算の複雑さ間のトレードオフを考えなければならない。本稿では、標準的なアプローチの代わりに、Matrix Geometric[5]の方法を用いて、この問題を取り扱う。まず、マルチプログラミングレベルm=3のとき、行列Gの例を見てみよう（図２参照）。

この例では、状態変数をスカラー化するために、状態(i,j)は状態nに対応させている。ただし、n=i+m×jである。

この行列の最初の４列を別にして、同じタイプの列は３列ごとに現れる。さらに、この性質は任意のmにもあると推論できる。したがって、行列Gはつぎのようにブロック行列の形で書ける。

$$\begin{vmatrix} B_{00} & B_{01} & 0 & 0 & 0 & 0 & \cdots \\ B_{10} & B_1 & A_0 & 0 & 0 & 0 & \cdots \\ \cdot & B_2 & A_1 & A_0 & 0 & 0 & \cdots \\ \cdot & B_3 & A_2 & A_1 & A_0 & 0 & \cdots \\ \cdot & B_4 & A_3 & A_2 & A_1 & A_0 & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{vmatrix}$$

ここで、全てのA_kとB_kはm次正方行列で、B_{10}とB_{01}はそれぞれm次の列と行ベクトルで、B_{00}はスカラーである。付録１にこれらのブロックの中身が与えられている。この例からも分かるように、考察している確率過程は複雑な境界振舞いを持つ連続パラメータのGI/M/1タイプマルコフ過程である。

まず、行列Aを

$$A=\sum_{k=0}^{m+1} A_k \quad (3)$$

と定義しておく。行列GとA_kの構造から、行列Aが既約で、なお、Ae=0という関係を満足することがわかる。ここで、eと0はそれぞれ単位ベクトルとゼロベクトル

	00	10	20	30	11	21	31	12	22	32	13	23	33
00	$-\lambda$	λ											
10	μ	$-\lambda-\mu$	λQ_1		$\lambda\bar{Q}_1$								
20		2μ	$-\lambda-2\mu$	λQ_2		$\lambda\bar{Q}_2$							
30			3μ	$-\lambda-3\mu$			λ						
11		μ			$-\lambda-\mu$	λQ_1		$\lambda\bar{Q}_1$					
21			$2\mu Q^1_{11}$		$2\mu Q^0_{11}$	$-\lambda-2\mu$	λQ_2		$\lambda\bar{Q}_2$				
31				$3\mu Q^1_{21}$		$3\mu Q^0_{21}$	$-\lambda-3\mu$			λ			
12			μQ^2_{02}		μQ^1_{02}			$-\lambda-\mu$	λQ_1		$\lambda\bar{Q}_1$		
22				$2\mu Q^2_{12}$		$2\mu Q^1_{12}$		$2\mu Q^0_{12}$	$-\lambda-2\mu$	λQ_2		$\lambda\bar{Q}_2$	
32							$3\mu Q^1_{22}$		$3\mu Q^0_{22}$	$-\lambda-3\mu$			λ
13				μQ^3_{03}		μQ^2_{03}		μQ^1_{03}			$-\lambda-\mu$	λQ_1	
23							$2\mu Q^2_{13}$		$2\mu Q^1_{13}$		$2\mu Q^0_{13}$	$-\lambda-2\mu$	λQ_2
33										$3\mu Q^1_{23}$		$3\mu Q^0_{23}$	$-\lambda-3\mu$

図2　m=3の行列Gの例($\bar{Q}_i=1-Q_i$)

である。πを

$$\pi A=0$$
$$\pi e=1 \qquad (4)$$

を満足する横ベクトルとすると、マルコフ過程の正再帰条件は

$$\pi A_0 e<\sum_{k=0}^{m+1}(k-1)\pi A e \qquad (5)$$

となる。これはシステムの安定条件でもある。

Zを全ての成分がゼロの行列とし、(5)式が満足されるならば、行列方程式

$$\sum_{k=0}^{m+1} R^k A_k=Z \qquad (6)$$

はsp(R)<1を満足する最小非負解がある。ここで、sp(R)はRのスペクトル半径で、全ての固有値のノルムの最大値と定義されている。Rが非負の場合に、対応する固有値は正の実数となる。

行列RはR(0)=Zから出発して、次の再帰的な式から計算できる。

$$R(N+1)=(A_0+\sum_{k=0}^{m+1} A_k R(N)^k)(-A_1)^{-1} \qquad (7)$$

再帰的な操作は

$$\max_{ij} |(R(N+1)-R(N))_{ij}|<\varepsilon \qquad (8)$$

が満足されるまで続く。付録２で、収束に関する証明を示す。

状態確率をブロック行列Gに対応させて、

$$P_{00}=P_{rob}(i=0,J=0)$$
$$P_n=\{Prob(i=1,j=n),Prob(i=2,j=n),\cdots,Prob(i=m,j=n)\}$$

とする。また、行列B(R)を

$$B(R)=\begin{vmatrix} B_{00} & B_{01} \\ B_{10} & \sum_{k=1}^{m+1} R_{k-1}B_k \end{vmatrix} \qquad (9)$$

と定義する。

$$(P_{00},P_0)B(R)=0$$
$$P_{00}+P_0(I-R)^{-1}e=1 \qquad (10)$$

P_nは

$$P_n=P_0R^n \qquad (n\geqq 1) \qquad (11)$$

となる。Neutsの定理が上の結果を証明する道具になる。

確率ベクトルP_0と行列Rを用いて、システムのいくつかの性能評価量の計算ができる。システムの最大スループットは安定条件(5)で決められる。

平均待ち行列長は

$$L=\sum_{j=1}^{\infty} jP_je=\sum_{j=1}^{\infty} jP_0R^je=P_0(I-R)^{-2}e \qquad (12)$$

となる。Littleの公式により、トランザクションの平均待ち時間は

$$W=L/\lambda \qquad (13)$$

となる。

平均応答時間は平均待ち時間と平均サービス時間の和になる、すなわち、

$$re=W+1/\mu \qquad (14)$$

他のいくつかの性能評価量の計算も可能である。例えば、システムにブロックされているトランザクションがない確率は

$$NW=P_{00}+P_0e \qquad (15)$$

となり、到着したトランザクションがブロックせずに、直ちに実行される確率は

$$Nb=P_{00}+\sum_{k=0}^{\infty} P_kQ=P_{00}+P_0(I-R)^{-1}Q \qquad (16)$$

となり、実行されているトランザクションの平均数は

$$AA=\sum_{k=0}^{\infty} P_kE=P_0(I-R)^{-1}E \qquad (17)$$

となる。ここで、$Q=(Q_1,Q_2,\cdots,Q_m)^T$、$E=(1,2,\cdots,m)^T$と定義されている。

4　数値結果

前記の方法を用いて、いろいろのパラメータについて、応答時間の計算を行い、数値結果を図３～６に示す。

直感からわかるように、ロッキングに失敗する確率がゼロに近付くとき、すなわち、D/Sが非常に大きいとき、結果は通常のM/M/m待ち行列と一致する。逆に、衝突が常に存在する場合には、結果はM/M/1待ち行列と一致する。図３はこの二つの極端な場合の数値結果を示

している。平均応答時間と最大スループットがアクセス集合サイズに非常に敏感であることがわかる。

図4(a)はロッキングに失敗する確率が小さい場合の結果を与えている。図3と同じように、Dとmは固定され、Sは1から4まで変わる。図4(b)と4(c)で、数値計算とシミュレーション結果の比較を行っている。ここから、仮定の影響は余り大きくないことがわかる。図5(a)に、D/Sを一定にしながら、DとSを同時に変化させる。SまたはDが大きければ大きいほど、応答時間は長くて、最大スループットは小さくなる。この現象はD/Sが定数のとき、Sが増大すると、ロッキングに失敗する確率が大きくなるからである。同図と図5(b)、図5(c)はシミュレーション結果も与えている。

図6では、マルチプログロミングレベルmとサービス率の積を一定に保ちながら、これらを変化させたときの平均待ち時間への影響を示している。結果として衝突確率が余り小さくないとき、小さいmの方が良い性能を与えていることを示している。この現象はロッキングの失敗が起こり得ることから、普通のM/M/m待ち行列より、著しく現れる。従って、システム性能を上がるためには、サーバ数の増加より、サービス率を高めるほうが有効であることが分かる。

5 あとがき

本稿では静的な二相ロッキングアルゴリズムに基づき、運用されているデータベースシステムを数学的にモデル化し、アクセス集合に関する無記憶性の仮定のもとで、解析を行った。これにより、応答時間、トランザクションの待ち行列長さらには最大スループットの定量的評価が可能である。

本研究の可能な拡張としては、次のようなものがある。まず、2節のモデルに、離脱点において、ブロックが起こったら、ロッキングが打ち切られると考えたが、ブロックが起こっても、次のトランザクションのロッキングを続けることはデータ資源の有効利用をはかる有効な手段の一つであろう。もし、一つの離脱点において、ロッキングできるトランザクションの個数に上限を付けると、3節の解析手順がこのケースにも適応できる。これにより、増大する解析上の複雑さはより高次元の連立方程式を解くことである。例えば、上限がマルチプログラミングレベルmと等しいとき、($m \times m+1$)次の連立方程式を解く必要がある。

ロックの種類に関して、本稿では専有ロックだけを考えたが、専有ロックと共有ロックの両方を含む場合に拡張できる。

付録 1

状態遷移確率行列の成分は次のように書ける。まず、B_{00}とB_{01}は空のシステムからのワンステップの遷移でB_{10}は空のシステムへのワンステップ遷移で決められる。

$$B_{00}=-\lambda \tag{A1}$$

$$B_{01}=(\lambda,0,\cdots,0) \tag{A2}$$

$$B_{10}=(\mu,0,\cdots,0)^T \tag{A3}$$

行列B_iは(i,0)から(i',0)への遷移を表している。(i,i'= 1,2,…,m)。

$$B_1=\begin{vmatrix} -\lambda-\mu & \lambda Q_1 & & & & \\ 2\mu & -\lambda-2\mu & \lambda Q_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & m\mu & -\lambda-m\mu \end{vmatrix} \tag{A4}$$

B_2は(i,1)から(i',0)への遷移に対応して、

$$B_2=\begin{vmatrix} \mu & & & & & \\ 2\mu Q_1 & & & & & \\ & 3\mu Q_3 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & m\mu Q_{m-1} \end{vmatrix} \tag{A5}$$

と書ける。

3以上のkに対して、B_kは(i,k-1)から(i',0)への遷移を対応する。ここで、一般式は次のようになる。

$$B_k=\begin{vmatrix} 1 & 2\ldots k-1 & k & \cdots & m \\ \mu Q^k_{0k} & & & & \\ & 2\mu Q^k_{1k} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & (m-k+1)Q^k_{m-kk} \end{vmatrix} \tag{A6}$$

同じ方法で、A_kを書き下　すと、次のようになる。

$$A_0=\begin{vmatrix} \lambda\bar{Q}_1 & & & & \\ & \lambda\bar{Q}_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \lambda Q_m \end{vmatrix} \quad (A7)$$

$$A_1=\begin{vmatrix} -\lambda-\mu & \lambda Q_1 & & & \\ 2\mu Q^0_{11} & -\lambda-2\mu & \lambda Q_2 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & \lambda Q_{m-1} \\ & & & m\mu Q^0_{m-1} & -\lambda-m\mu \end{vmatrix} \quad (A8)$$

$$A_k=\begin{vmatrix} 1\ 2\ldots k-1 & k & & & m \\ \mu Q^k_{0k+1} & & & & \\ & 2\mu Q^k_{1k+1} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & (m-k+1)\mu Q^k_{m-kk} \end{vmatrix} \quad (A9)$$

付録　2

ここで、(7)式と(8)式により、収束列R(N)が導かれることを証明する。これの準備として、一つの定理を証明しておく。

定理：バンド行列A_1は正則で、A_1^{-1}の全ての成分は負である。

証明：行列A_1は行列Gの対角ブロックであるから、A_1は既約で、$A_1 e<0$、なお、A_1の全ての対角成分が正である。従って、A_1は厳格な対角支配の既約行列である。Ortega[9]の定理により,A_1が正則で、かつ$A_1^{-1}>0$。

(7)式で、R(0)=Zから出発して、

$$R(1)=A_0(-A_1)^{-1} \geqq Z=R(0) .$$

がある。

さらに、$R(N) \geqq R(N-1)$と仮定して、

$$R(N+1)=(A_0+\sum_{k=2}^{m+1} R^k(N)A_k)(-A_1)^{-1}$$

$$\geqq (A_0+\sum_{k=2}^{m+1} R^k(N-1)A_k)(-A_1)^{-1}$$

となる。従って帰納法により、全てのN>0に対して、R(N+1)>R(N)となる。非負行列Rの存在が(5)の条件で保証されているから、系列 {R(N)} は単調にRに収束する。

参考文献

[1] D. Potier & Ph. Leblanc, "Analysis of Locking Policies in Database Management Systems," C. ACM., vol. 23, No.10, pp. 584-593, Oct. 1980.

[2] D. Mitra & P. J. Weinberger, "Probabilistic Models of Database Locking: Solutions, Computational Algorithms, and Asymptotics," J. ACM., vol.31, No. 4, pp. 855-878, Oct. 1984.

[3] R. J. T. Morris & W. S. Wong, "Performance of Concurrency Control Algorithms with Nonexclusive Access," Performance '84, North-Holland, Amsterdam, pp. 87-101, 1984.

[4] A. Thomasian, "Performance Evaluation of Centralized Database with Static Locking." IEEE Trans. on Software Engineering, vol. SE-11, No.4, April 1985

[5] M. Neuts, "Matrix-Geometric solutions in Stochastic models." The Johns Hopkings University Press, 1981.

[6] A. W. Shum & P. G. Spirakis, "Performance Analysis of Concurrency Control Methods in Database Systems," Performance '81, North-Holland, pp. 1-19, 1981.

[7] A. Chesnais, E. Gelenbe & I. Mitrani, "On the Modeling of Parallel Access to Shared Data," C. ACM., vol. 26, No. 3, pp. 196-202, March, 1983.

[8] Y. C. Tay, "A Mean Value Performance Model for Locking in Database," Ph. D. dissertation, Harvard Univ., Cambridge, MA, Apr. 1984.

[9] J. M. Ortega & W. C. Rheinboldt, " Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, New York, 1970.

[10] H. GarciaMolina, "Performance of Update Algorithms for Replicated Data," UMI Research Press, Ann Arbor, Michigan,1981.

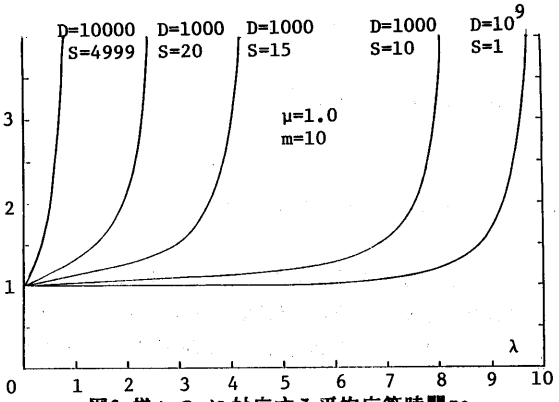

図3 様々のsに対応する平均応答時間re

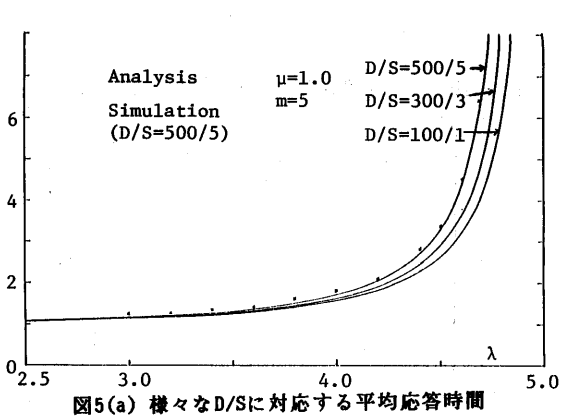

図5(a) 様々なD/Sに対応する平均応答時間

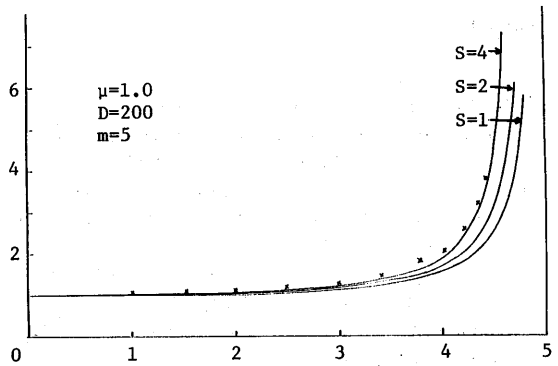

図4(a) パラメータsが小さいときの平均応答時間re

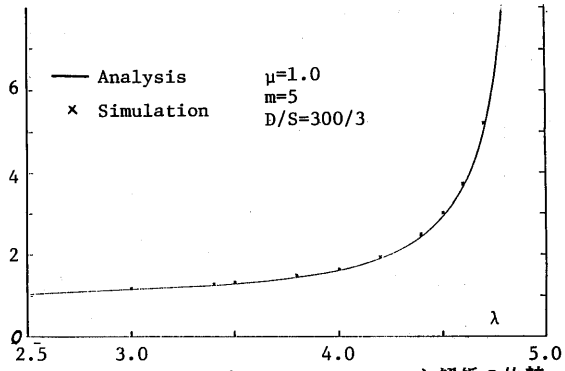

図5(b) D/S=300/3の時シミュレーションと解析の比較

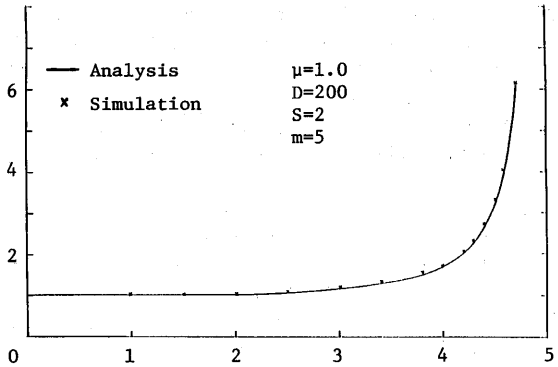

図4(b) s=2ときのシミュレーションと解析結果の比較

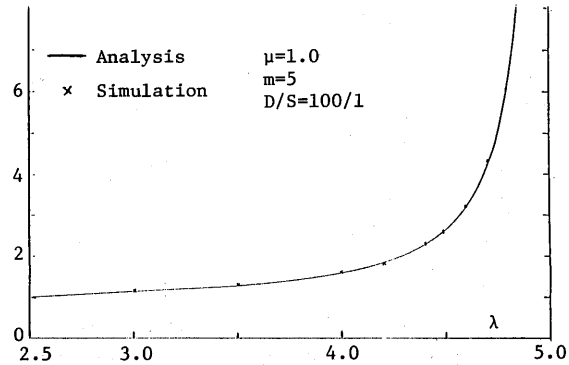

図5(c) D/S=100/1の時シミュレーションと解析の比較

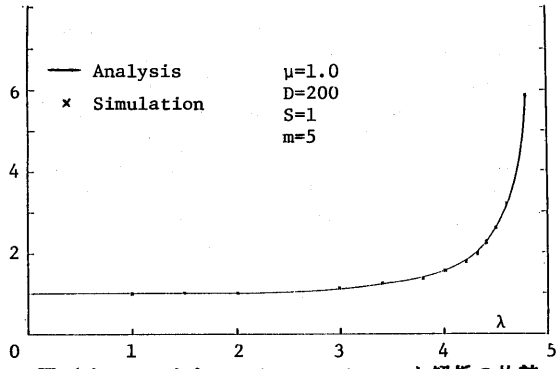

図4(c) s=1のときのシミュレーションと解析の比較

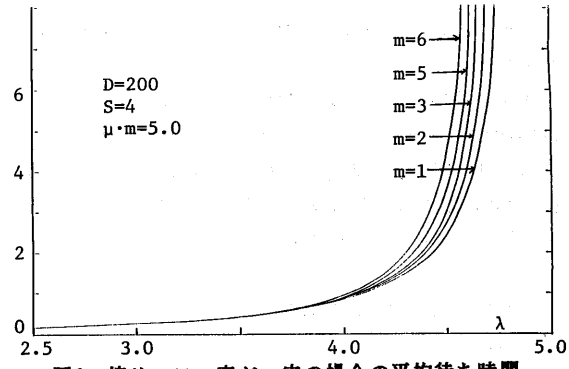

図6 総サービス率が一定の場合の平均待ち時間