オペレーティング・システム　55－2
（１９９２．６．８）

トランザクション処理における楽観的並列処理について

松尾 文碩　　竹田 正幸　　王 志遠

九州大学工学部　　日本エヌ・シー・アール

データベースシステムにおけるトランザクションの並列制御に楽観的方式がある．この方式は，トランザクション間の書出しの衝突の頻度が小さい場合に効率的な方法である．本稿では，マルコフ過程モデルにより楽観的方式を解析し，衝突によって無効になるトランザクションが発生する確率を求めた．更に，各トランザクションの読込みオブジェクト集合が予知できる場合に対する新しい楽観的制御方式を提案し，この方式によって失敗終了確率が減少することを示した．

ANALYSIS AND IMPROVEMENT OF OPTIMISTIC SYNCRONIZATION METHOD

Fumihiro Matsuo　　Masayuki Takeda　　Wang Zhi Yuan

Faculty of Eng., Kyushu Univ. 36
Hakozaki, Fukuoka 812, Japan

NCR Japan, Ltd.
Akasaka, Tokyo 107, Japan

A recent approach to concurrency control for database systems is the optimistic synchronization method. The method is more efficient than other methods when the rate of conflicts for updating is low. This paper presents an analysis of the conventional optimistic method by a Markov process model. The probability function for the tranzactions of unsuccessful execution is obtained as a result of the analysis. This paper also discusses an improved optimistic scheme that can be used when the read-set of each tranzaction is predictable. The superiority of the new method over the conventional optimistic method in efficiency is shown by an analysis using the Markov process model.

1. はじめに

データベースシステムにおけるトランザクションの並列実行の制御は，システムの性能にかかわる重要な問題である．データベースは，矛盾を含んでいては無価値である．すなわち，データベースは常に一貫性(consistentcy)が保たれていなければならない．一貫性の保持が保証される代表的な並列制御方式は，2相施錠(two-phase lock)方式と時刻印(time stamp)方式である．しかし，この二つの方式はいずれもオーバヘッドが大きく，データベースシステムの性能を劣化させる．Kung and Robinson[4]が提案した楽観的方式では，トランザクションの到着後直ちにそれを実行し，終了時の書出しを検査することによって，一貫性が破壊されるかどうかを調べる．一貫性を損なう場合には，そのトランザクションの書出しは行われず，そのトランザクションは実行されなかったことにする．この方式は，オーバヘッドが小さく，トランザクション間の書出しの衝突(conflict)の頻度が小さい場合に効果的な方法である．

これまで，楽観的方式の評価については，シミュレーションの結果だけが報告されている．本稿では，マルコフ過程に基づく解析により不成功に終わるトランザクションの確率を求めた．更に，新しい楽観的制御方式を提案し，これにより失敗終了確率が減少することを示した．

2. 楽観的制御方式

データベースは，ファイルやレコードのようなオブジェクトの集合である．オブジェクトは，読込みと書出しの動作の対象となる基本単位である．トランザクションは，データベースに対する処理の論理的単位であり，オブジェクトの読込み，書出しなどのデータベースの処理に必要な一連の動作からなる．

トランザクションはデータベースを一貫性状態から一貫性状態へ写像するものであると考えるが，複数のトランザクションが非統制的に並列実行されるならば，データベースの一貫性は必ずしも保証されない．並列制御とは，データベースの一貫性を保存するようにトランザクションの並列実行を制御することである．

楽観的並列制御方式[1),2),3),4),5),6)]では，各トランザクションは，到着すると直ちに実行を開始することができる．読込みは，最初にまとめて行われ，トランザクション内にオブジェクトの複写がつくられる．読込み動作は，この複写への参照となる．書出し動作は，トランザクションの内部記憶に対して行われる．すべての動作が完了すると，読み込んだオブジェクトが他のトランザクションによって，書き換えられているかどうかを調べる．読み込んだすべてのオブジェクトが不変であれば，内部記憶にある書出しオブジェクトをデータベースに書き出し，このトランザクションは成功裡に終了する．書き換えられたオブジェクトがあれば，このトランザクションの実行は不成功で，再び最初から実行する．前者を成功終了(clean up)といい，後者を失敗終了(back up)という．

3. マルコフ過程モデルによる楽観的並列制御方式の解析

楽観的並列制御方式の失敗終了の確率をマルコフ過程モデル(Markov process model)によって求めることができる．

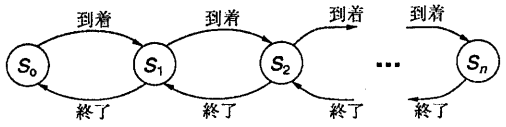

図1 状態遷移図

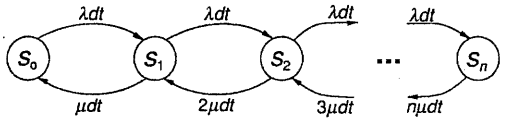

図2 マルコフ過程モデル

図1の状態遷移図において，状態$s_k(k=0,1,\ldots,n)$は，システムに実行中のトランザクションがk個存在する状態を表わしている．nは，システムで並列実行可能なトランザクションの最大数である．図2は，図1に対応するマルコフ過程モデルである．ここで，時間dt内にトランザクションが到着する確率をλdtとする．したがって，平均到着間隔は$1/\lambda$である．また，一つのトランザクションがdt時間内に終了する確率をμdtとする．平均実行時間は$1/\mu$である．ここで，システムが状態s_kにあるとき，k個のトランザクションは独立であるとし，システムの能力は負荷に依存しないとして，s_kからs_{k-1}への遷移確率を$k\mu dt$とする．

いま，システムが状態 s_k にある確率を p_k で表わす．また，$\rho = \lambda/\mu$ とする．ρ は，平均実行時間と平均到着間隔の比である．まず，s_{k-1} から s_k へ遷移するトランザクション数と s_k から s_{k-1} へと遷移するトランザクション数は，等しいので，

$$\lambda dt p_{k-1} = k\mu dt p_k, \tag{1}$$

ここで，$k = 1, \ldots, n$.

ところで，

$$\sum_{k=0}^{n} p_k = 1. \tag{2}$$

(1) 式と (2) 式から，直ちに，

$$p_k = \frac{\frac{\rho^k}{k!}}{\sum_{i=0}^{n} \frac{\rho^i}{i!}}, \tag{3}$$

ここで，$k = 1, \ldots, n$.

さて，トランザクションが失敗終了する確率は，そのトランザクションと並列実行した他のトランザクションの数 m に比例すると考える．しかし，この m を推定するのは，非常に困難である．そこで，状態 $s_k (k = 1, \ldots, n)$ で終了した場合，その時点で並列実行中の他のトランザクションの数 $k-1$ に比例すると仮定する．ここで，この比例定数を η とおく．すると，失敗終了する確率 p_F は，

$$\begin{aligned} p_F &= \sum_{k=1}^{n} (k-1)\eta \frac{p_k}{1-p_0} \\ &= \eta \frac{\sum_{k=1}^{n} \frac{(k-1)\rho^k}{k!}}{\sum_{k=1}^{n} \frac{\rho^k}{k!}} \end{aligned} \tag{4}$$

(4) 式から，つぎのことがわかる．

1) $n = 1$ のとき，$p_F = 0$.

2) $\rho = 0$ ならば，$p_F = 0$.

3) $\lim_{\rho \to \infty} p_F = \eta(n-1)$.

1 は，並列実行がない場合，失敗終了がないことを表わしている．2 は，負荷がほとんどない場合，失敗終了がないことを意味する．3 は，負荷が極めて大きい場合，失敗終了の確率は $n-1$ に比例することを示している．これらは，直感とも一致するので，本稿の解析の妥当性を示しているものと考えることができる．

さて，(3) 式から

$$\lim_{n \to \infty} p_k = \frac{\rho^k e^{-\rho}}{k!}. \tag{5}$$

また，(4) 式から

$$\lim_{n \to \infty} p_F = \eta \frac{\rho e^{\rho} - e^{\rho} + 1}{e^{\rho} - 1}. \tag{6}$$

4. 失敗終了トランザクションの再実行

失敗終了のトランザクションは，再実行されなければならない．いま，dt 時間内にトランザクションが終了する確率を νdt とする．すると，失敗終了する確率は $p_F \nu dt$ である．失敗終了トランザクションの再実行を考慮した解析モデルは，図 3 のようになる．dt 時間内にトランザクションが実行を要求する確率は，$(\lambda + p_F\nu)dt$ である．ρ が大きくなるとこの確率が大きくなり，トランザクションの実行待ち行列ができる．待ち行列ができる状況では，楽観的方式は効率的ではないと考えられるので，ここでは待ち行列はできないとして，dt 時間内に実行が始まる確率と終了する確率とは等しいと仮定する．すなわち，

$$\lambda + p_F\nu = \nu \tag{7}$$

が成立するものとする．この場合のマルコフ過程モデルでは，平均実行時間と平均実行開始間隔の比は，

$$(\lambda + p_F\nu)/\mu$$

となる．これは，(7) 式から

$$\rho/(1 - p_F)$$

であるから，失敗終了の確率は，

$$p_F = \eta \frac{\sum_{k=1}^{n} \frac{(k-1)(\frac{\rho}{1-p_F})^k}{k!}}{\sum_{k=1}^{n} \frac{(\frac{\rho}{1-p_F})^k}{k!}} \tag{8}$$

となる．(8) 式は，p_F の高次代数方程式であるから，n が 5 以上では代数的解が存在しないので，数値解によっ

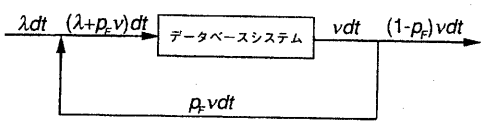

図 3 失敗終了トランザクションの再実行

てp_Fを求めざるを得ない．数値解をみると，nが20以上では，収束しているようにみえる(図4参照)．そこで，$n \to \infty$のときは，(6)式から

$$p_F = \eta \frac{\rho e^{\frac{\rho}{1-p_F}} - e^{\frac{\rho}{1-p_F}} + 1}{e^{\frac{\rho}{1-p_F}} - 1}. \tag{9}$$

したがって，

$$e^{\frac{\rho}{1-p_F}} = \frac{1}{1 - \frac{\rho}{1-p_F}/(1+\frac{p_F}{\eta})}. \tag{10}$$

ところで，$a > 0$のとき，xについての方程式

$$e^x = \frac{1}{1 - x/a} \tag{11}$$

の根は，

$$x = a - \varepsilon$$

であり，

$$\lim_{a \to \infty} \varepsilon = 0$$

である．例えば，$x = 10$のとき，$\varepsilon = 4.5 \times 10^{-4}$であり，$x = 20$のとき，$\varepsilon = 4 \times 10^{-8}$である．したがって，$a$がある程度大きいとき，方程式(11)の根は近似的に

$$x = a$$

としてもよい．この結果を(10)式に適用すると，

$$\frac{\rho}{1-p_F} = 1 + \frac{p_F}{\eta}. \tag{12}$$

方程式(12)から，

$$p_F = \frac{(1-\eta) - \sqrt{(1-\eta)^2 - 4\eta(\rho-1)}}{2}. \tag{13}$$

(13)式から，p_Fが実数であるための条件は，

$$(1-\eta)^2 - 4\eta(\rho-1) \geq 0$$

であり，したがって

$$\rho \leq (\eta+1)^2/4\eta. \tag{14}$$

(14)式から，

$\eta = 0.1$のとき，$\rho \leq 3.0$;

$\eta = 0.01$のとき，$\rho \leq 25.5$;

$\eta = 0.001$のとき，$\rho \leq 250.5$

であることがわかる．図4に示すように，(13)式の近似度は非常に良い．nが大きくなると，ρが大きいところでも近似度が良くなる．大規模データベースシステムでは，nは100を越えることが普通であるので，(13)式は，そのようにnが大きい場合の方程式(8)の解であるとみなすことができる．

5. 予知読込み集合に対する楽観的制御

実際のデータベースシステムでは，各トランザクションの読込みオブジェクト集合がトランザクションの実行前にわかっていることがある．このような場合には，次のような方式により，失敗終了確率を減少することができる．

まず，トランザクションが到着すると直ちに実行するのではなく，各トランザクションの実行の前に，読み込もうとするオブジェクトが実行中のトランザクションによって書き換えられるかどうかを調べ，書き換えられる可能性がない場合にのみ実行を開始する方式である．この方式を予知読込み集合に対する楽観的並列制御方式ということにする．

予知読み込み集合に対する楽観的並列制御方式の場合，到着したトランザクションは，直ちに実行されず，読込み予定のオブジェクトが実行中のトランザクションによって書き換えられない場合のみ実行に移される．そこで，状態$s_k (k = 0, 1, \ldots, n-1)$において到着したトランザクションが実行に移される確率を$\lambda(1-k\eta)dt$とする．すると，(1)式の代わりに，

$$\lambda(1-(k-1)\eta)dt p_{k-1} k\mu dt p_k, \tag{15}$$

ここで，$k = 1, \ldots, n$，

を得る．これから，直ちに，

$$p_k = \left(\frac{\rho^k \prod_{j=1}^{k}(1-(j-1)\eta)}{k!} \right) p_0, \tag{16}$$

ここで，$k = 1, \ldots, n$.

p_0は，(2)式と(16)式から，

$$\begin{aligned} p_0 &= \frac{1}{1 + \sum_{i=1}^{n} \frac{\rho^i \prod_{j=1}^{i}(1-(j-1)\eta)}{i!}} \\ &= \frac{1}{\sum_{i=0}^{n} \frac{\rho^i \prod_{j=1}^{i}(1-(j-1)\eta)}{i!}}. \end{aligned} \tag{17}$$

(16), (17)式から，

$$p_k = \frac{\frac{\rho^k \prod_{j=1}^{k}(1-(j-1)\eta)}{k!}}{\sum_{i=0}^{n} \frac{\rho^i \prod_{j=1}^{i}(1-(j-1)\eta)}{i!}}, \tag{18}$$

ここで，$k = 1, \ldots, n$.

さて，状態s_kにおいて，失敗終了する確率は，ここでも$k-1$に比例すると仮定する．しかし，この確率

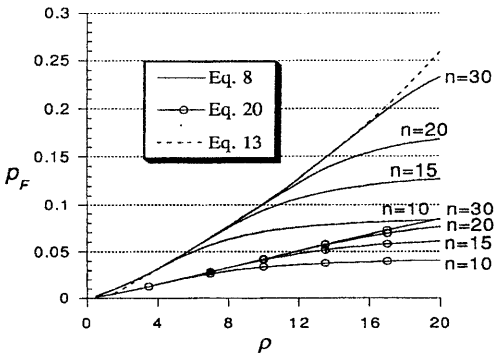

図 4 失敗終了確率 ($\eta = 0.01$)

は，一般に$\eta(k-1)$より小さいはずである．なぜなら，そのトランザクションの読込みオブジェクトを更新するトランザクションは実行が後から始まったものに限られるからである．ここでは，簡単のため，状態s_kにおいて失敗終了する確率を$\alpha\eta(k-1)$とする．ここで，$0 \leq \alpha \leq 1$．すると，失敗終了トランザクションの再実行を考慮しない場合の失敗終了確率p_Fは，

$$p_F = \frac{\sum_{k=1}^{n}(k-1)\alpha\eta p_k}{1-p_0} = \alpha\eta \frac{\sum_{k=1}^{n} \frac{\rho^k(k-1)\prod_{j=1}^{k}(1-(j-1)\eta)}{k!}}{\sum_{k=1}^{n} \frac{\rho^k \prod_{j=1}^{k}(1-(j-1)\eta)}{k!}}. \quad (19)$$

失敗終了トランザクションの再実行を考慮すると，前節と同様に，失敗終了確率p_Fとして，次式を得る．

$$p_F = \alpha\eta \frac{\sum_{k=1}^{n} \frac{(\frac{\rho}{1-p_F})^k(k-1)\prod_{j=1}^{k}(1-(j-1)\eta)}{k!}}{\sum_{k=1}^{n} \frac{(\frac{\rho}{1-p_F})^k \prod_{j=1}^{k}(1-(j-1)\eta)}{k!}}. \quad (20)$$

(8), (13), (20) 式を図 4 に示す．図 4 では，(20) 式のαは 1/2 とした．図 4 から予知読込み集合に対する楽観的制御方式の効果がわかる．

6. むすび

本稿では，まず，楽観的制御方式をマルコフ過程モデルによって解析し，実行はしたけれど無効にせざるを得ないトランザクションが発生する確率を求めた．現在，広く採用されている並列制御方式は，2 相施錠方式である．この方式は，オーバヘッドが大きく，デッドロックを回避する機構をもたなければならない．2 相施錠方式は，オーバヘッドとデッドロック回避による効率の低下がある．楽観的制御では，オーバヘッドは小さいが，結果的に無効となるトランザクションを実行することによる効率の低下がある．後者の効率の低下が小さいときに，楽観的制御の採用が可能となる．本稿の結果は，この評価のために使うことができる．

次に，各トランザクションの実行前にその読込みレコードあるいはファイルがわかっているようなデータベースシステムについては，より効率的な楽観的制御方式を提案した．そして，マルコフ過程モデルによる解析によって提案した方式が通常の楽観的制御方式より，無効となるトランザクションの発生が著しく小さいことを示した．

参考文献

1) Bassiouni, M. : Single-Site and Distributed Optimistic Protocols for Concurrency Control, *IEEE Trans. Software Eng.*, Vol. 14, No. 8, pp. 1071–1080 (1988).

2) Bassiouni, M. and Khamare, U. : Optimistic Concurrency Control-Schemes for Performance Enhancement, in *Proc. IEEE 10th Int. COMPSAC Conf.*, pp. 43–49 (1986).

3) Boksenbaum, C., Cart, M., Ferrie, J., and Pons, J. : Concurrent Certifications by Intervals of Timestamps in Distributed Database Systems, *IEEE Trans. Software Eng.*, Vol. SE-13, No. 4, pp. 409–419 (1987).

4) Kung, H. and Robinson, J. : On Optimistic Methods for Concurrency Control, *ACM Trans. Database Syst.*, Vol. 6, No. 2, pp. 213–226 (1981).

5) Reimer, M. : Solving the Phantom Problem by Predicative Optimistic Concurrency Control, in *Proc. VLDB Conf.*, Florence, Italy, pp. 81–88 (1983).

6) Shlageter, G. : Optimistic Methods for Concurrency Control in Distributed Database Systems, in *Proc. VLDB Conf.*, Cannes, France, pp. 125–130 (1981).