5AA－8

並行処理制御における独立化可能クラスの有効性検証
— 非施錠方式に対するシミュレーション —

†李 相勲　†徐 海燕　‡史一華　††古川哲也
†福岡工業大学　‡福岡工業短期大学　††九州大学

1　まえがき

データベースにおける並行処理の正当性は、スケジュールが直列可能性を満たすこととされてきた。しかし、共同作業を必要としたり長時間の処理を含む高水準データベースでは、直列可能性を満足しなければならないことによる問題点が広く認識されている。そのため、一貫性情報を利用することにより、正当なスケジュールのクラスを独立化可能クラスに拡大する研究が行われている[2]。独立化可能クラスは、論理性を満たし、かつ一貫したデータベースから検索を行い、変更された結果も一貫している処理単位からなるスケジュールと等価なスケジュールの集合である。本稿では、直列可能クラスと独立化可能クラスを比較するための非施錠方式に対するシミュレーション結果を報告する。

2　独立化可能性に基づく時刻印方式

個々の利用者のデータベースに対する操作は、検索操作($R(X)$)、変更操作($W(X)$)の全順序関係 $<$ を持つ集合である処理単位Tで表される。ここで、$R(X)$, $W(X)$ はXに属するすべての要素に対する同時の検索操作、変更操作である。複数の処理単位（処理単位集合$\boldsymbol{T}$）の並行実行をスケジュールHといい、$<_H$でその実行順序を記す。異なる処理単位T_i, T_jに属する$R_i(X)$と$W_j(Y)$、$W_i(X)$と$W_j(Y)$、$W_i(X)$と$R_j(Y)$ $(X \cap Y \neq \phi)$ は競合するという。

直列可能性に基づく非施錠並行処理制御方式の中で最もよく知られているのは次の時刻印方式である。

[定義 1] 各処理単位$T_k \in \boldsymbol{T}$の到着順に割り当てられる時刻印$ts(T_k)$と各々のT_iとT_jに属する競合操作p, q $(p \in T_i, q \in T_j, i \neq j)$ に対して、$ts(T_i) < ts(T_j)$ ならば $p <_H q$であるように操作の実行順序を制御する並行処理制御方式を時刻印(TO)方式という。 □

一貫性情報を利用できると、並行処理制御の正当性基準は次の独立化可能性に拡張できる[2]。独立化可能なスケジュールのクラスは、各処理単位T_iが次の３つの性質を満たす等価なスケジュールが存在するものからなる[2]。

検索一貫性： 処理単位T_iによって検索されるのは、ある一貫したデータベースの部分集合である。

結果一貫性： 検索されたデータベースに対して、単独実行した結果のデータベースは一貫している。

論理性： 処理単位T_iによって操作されたデータ項目は、T_iが終了するまで他の処理単位$T_j(j \neq i)$によって変更されていない。

スケジュールHの独立化可能性を判定する問題は、一貫性に関する問題と並行処理制御に関する問題に分けられ、後者に対しては次の方式がある[2]。

[定義 2] 処理単位$T_i \in \boldsymbol{T}$の到着順に時刻印$ts(T_i)(>0)$を割り当て、次のように各操作$R_i(X)(W_i(X)) \in H$の時刻印$ts(R_i(X))(ts(W_i(X)))$を計算し、各種の時刻印間の条件を満たすように制御する並行処理制御方式を拡張時刻印方式(TOE)という。ただし, $ts(R_i(X))(ts(W_i(X)))$の初期値は0である。

(1)$R_i(X)(W_i(X)) <_H W_j(Y), i \neq j, X \cap Y \neq \phi$ の場合
　if $ts(T_j) \leq ts(T_i)$ then T_j を後退復帰
　else $ts(W_j(Y)) := MAX(ts(W_j(Y)), ts(T_i))$;

(2) $W_i(X) <_H R_j(Y), i \neq j, X \cap Y \neq \phi$ の場合
　if $t_i(T_j) \leq ts(W_i(X))$ then T_jを後退復帰
　else $ts(R_j(Y)) := MAX(ts(R_j(Y)), ts(W_i(X)))$;

(3) $R_i(X)(W_i(X)) <_i W_i(Y)$ の場合
　$ts(W_i(Y)) :=$
　　$MAX(ts(W_i(Y)), ts(R_i(X)(W_i(X))))$;
　$R_i(X)(W_i(X)) <_i R_i(Y)$ の場合
　$ts(R_i(Y)) :=$
　　$MAX(ts(R_i(Y)), ts(R_i(X)(W_i(X))))$; □

TOE方式に従うスケジュールは独立可能である[2]。TOE方式は各操作$R_i(X)(W_i(X))$に対して時刻印を設け、$W_i(X) <_H R_j(Y)$ に対する検証条件を、TO方式の$ts(T_i) < ts(T_j)$から$ts(W_i(X)) < ts(T_j)$に変更している。$ts(W_i(X)) < ts(T_i)$である[2]ため、TOE方式はTO方式の拡張となる。

3　シミュレーションによる比較

シミュレーションは論文[1]の性能評価モデルに基づいて行なった。発生した処理単位は待機列に入る。MPL

Efficiency of the Equivalent-Independent Class in Concurrency Control– Simulations of Non-Locking Protocols
Sanghoon LEE†, Haiyan XU†, Yihua SHI ‡,
Tetsuya FURUKAWA††
†Fukuoka Institute of Technology,
‡Fukuoka Junior College of Technology,
††Kyushu University

数（同時実行可能な処理単位数）の処理単位は並行処理列に入り、並行処理制御要請を出す。要請が許可されれば、対応する検索・変更操作を行なう。許可されなければ、処理単位は後退復帰される。

基本的に論文[1]に基づいてシミュレーションのパラメータを設定している。データベースのサイズを1,000ページとし、処理単位は平均して8個の検索操作と2個の変更操作からなるものとする。端末数を200個、予期実行時間を500 msec（検索操作8個×50 msec+変更操作2個×50msec）、外部思考時間を1 sec（各端末上の処理単位の終了から新しい処理単位の開始までの時間）としている。DISKの数はCPUの数の2倍にする。またこのシミュレーションでは同じプログラムを10秒間に20回実行させ、その平均を求めることによって性能評価をした。以下は主なシミュレーション結果である。

[スループットによる比較]

CPUとDISKの数が無限である無限資源の場合においては、TO方式のスループットは論文[1]の結果と同様にMPLが50で停滞期に達している。TOE方式では、MPLが75で停滞期となり、図1のようにTOとTOEの差は10%~24.7%である。

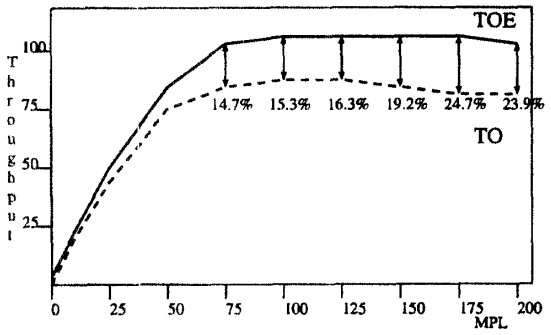

図1　データベースのサイズが1,000ページの無限資源の場合におけるTOとTOEの比較

[資源数による比較]

CPUとDISKの数が無限に存在することは現実的でないので、CPUの数を1個、5個、10個、25個、50個としてシミュレーションを行なった。CPUが1個の場合、5個の場合、10個の場合はTOとTOEの差は見られない。TOEの特徴で、早く操作が終了してもCPUとDISKの数が極端に少ないためにCPUのキューとDISKのキューで待つためである。CPUが25個と50個の場合はTOとTOEで差がでる。CPUが25個のときはMPLが25から100の間は4.7%~5.9%でMPLが125から200の間は1.2%~4.5%、CPUが50個のときはMPLが50から150の間は8.9%~9.7%でMPLが175から200の間は4.9%~5.5%の差が見られる。CPUとDISKの数が増えるとCPUのキューとDISKのキューで待つ時間が減るためである。

[処理単位の大きさによる比較]

検索操作を平均16個、変更操作を平均4個と処理単位のサイズを2倍にしたシミュレーションを行なった。

i) 無限資源の場合でもTOとTOEの差が25.9%~42%にひらく。それは、処理単位内の操作が多くなると競合率が高くなるためである。

ii) 有限資源の場合でもTOとTOEの差が大きくなる。CPUが25個の場合は22.5%~31.1%、CPUが50個の場合は11.2% ~17.3%であった。処理単位内の操作が多くなると競合率が高くなるため、有限資源でもTOとTOEの差が見られる。

[思考時間による比較]

現実に処理される操作は、読み込み後一定思考時間を経て変更操作を行なうことが多い。検索操作に対して25%の割合で思考時間100 msecを取るようにシミュレーションを行なった。その結果、TOとTOEの差は16.4% ~28.2%にひらいた。検索操作の時間が長くなり、他の処理単位の操作との競合が発生しやすくなるためである。

[競合率による比較]

競合率を変化させた場合、すなわち、データベースのサイズを10,000ページと2,000ページとしたときのシミュレーションも行なった。その結果、TOとTOEの差は10,000ページのときは3.5%~5.5%、2,000ページのときは2.5%~8.5%と小さくなる。それは、データベースのサイズに対して処理単位のサイズが小さいので競合がほとんど起きないためである。

[後退復帰率による比較]

実行された処理単位の中で何パーセントが後退復帰になってしまうかを示す後退復帰率についてもTOとTOE方式について比較して見た。各場合ともTOEの後退復帰がTOより$\frac{1}{3}$ぐらい減少している結果が得らわた。

4　まとめ

本稿では、非施錠方式に対するシミュレーションにより、独立化可能という一貫性情報を持つ高水準DBにおける新たな基準と、従来の直列可能基準との比較を行なった。各場合とも並行性上で明確な差が見られる。また、後退復帰も従来より$\frac{1}{3}$位減少している結果が得られている。これらの結果より独立化可能基準が高水準データベースの並行処理制御において有効であると言える。

参考文献

[1] Pakesh Agrawal: Concurrency Control Performance Modeling:Alternatives and Implications, ACM TODS, Vol. 12, No. 4, pp. 609-654 (1987).

[2] 徐　海燕, 古川哲也, 史 一華：並行処理制御方式による独立化可能クラスと直列可能クラスの比較, 情処論, Vol. 37, No. 8, pp. 1600-1609 (1996).