

格上げ制限2相施錠方式の性能について

3Q-8

春原 典彦 生山 陽 篠崎 健一 宮崎 収兄
千葉工業大学 情報工学科

1. はじめに

データベースシステムにおける並行処理制御方式には、施錠をかける方式や楽観的制御方式など、いくつかの方式が提案されているが現在主として2相施錠方式(2PL)が使われている。しかし、2PLでは高負荷状態で参照されたあと、更新されるオブジェクト集合の割合が高い場合トランザクション完了率が低くなってしまふ。これに対し保守的2相施錠方式(C2PL)が提案されているが、トランザクション中でアクセスするすべての対象を最初に施錠する必要があり、実際の利用は困難とされている。そこで我々は、2PLに従来の保守的スケジューラより弱い制限を導入した、格上げ制限2相施錠方式(NU2PL)を提案した。

本稿ではNU2PLの性能を検討する。そして2PLでおこるデッドロックの大部分がNU2PLで防止できることを示す。

2 格上げ制限2相施錠方式
(non-upgrading two-phase locking)

2.1 NU2PL

並行処理制御において、トランザクションのread対象とwrite対象が何らかの関連性を持ち、両者に重複のある可能性は高い。例えば、多数存在するデータの中から目的のデータを選び出し、その後にアップデートするような場合である。このような場合はデッドロックが起こりやすい。そこで我々は2相施錠方式に以下の制限を取り入れた格上げ制限2相施錠方式(NU2PL)[春原95]を提案した。

・共有施錠を行った場合は、同一の対象について後で排他施錠を行えない

2.2 NU2PLの適用

制御方式には2PLのほかに競合保存直列可能スケ
Performance of Non-Upgrading Two-Phase Locking

Norihiko Sunohara, Kenichi Shinozaki, You Ikuyama, Nobuyoshi Miyazaki
Department of Computer Science, Chiba Institute of Technology
2-17-1 Tsudanuma Narashino Chiba 275 Japan

ジューラ(CPS)なども提案されているが、並行性の問題や途中で施錠できないといった問題のため多くの場合に対して実用的ではない。NU2PLは2PLをベースにしてあるため、システムへの適用に対し、すべてにNU2PLを適用するのではなく、通常のトランザクション(2PLでも問題がないもの)では2PLを使用し、デッドロックが起きやすいトランザクションについてはNU2PLを適用するといった事が可能である。この様に混在した場合でもデッドロック低減の効果があるため柔軟な適用が可能である。

2.3 NU2PLの動作

以下にreadした後にwriteする簡単な2段階トランザクションを考え、NU2PLの動作例を示す。

例1)

・トランザクションが
 $r [A, B, C] \rightarrow w [D]$
 のような場合の施錠方法は、
 $S [A, B, C] \rightarrow X [D]$
 のように施錠する。(2PLと同じ)

・トランザクションが
 $r [A, B, C] \rightarrow w [C]$
 のような場合の施錠方法は、
 $S [A, B] X [C]$
 のように参照する前にまとめて施錠してしまう。

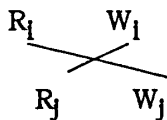
つまり、トランザクション完了率と並行性の両者を向上させるために、2PLのよいところをそのままに、C2PLよりもロックの制限を柔軟にした物である。C2PLではすべてトランザクション開始時に施錠するが、NU2PLではある対象に最初にアクセスする段階で施錠モードを決めるため、より実用的である。また、このデッドロック防止によりシステムスループットの向上の考えられる。システムスループットとは、単位時間に完了する平均トランザクション数で、並行性と完了率が高いほど大きくなる。[谷口93]などによれば高負荷状態では2PLとCPSは完了率が低くなるに従ってスループットも低下する事がわかっているため、NU2PLではスループットの向上が期待できる。

3 性能評価

3.1 デッドロック発生の条件

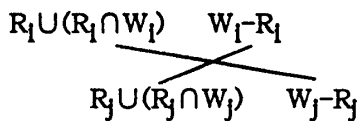
あるトランザクション T_1 において、read set を R_1 、write setを W_1 とする2段階トランザクションを考えた場合、 T_1 と T_j がデッドロックする条件をDとすると、以下のように書き表すことが出来る。

<2PL>



$$D(2PL) = (R_1 \cap W_j \neq \phi) \wedge (R_j \cap W_1 \neq \phi)$$

<NU2PL>



$$D(NU2PL) = D(2PL) \wedge ((R_1 \cap R_j \cap W_j) = \phi) \wedge ((R_j \cap R_1 \cap W_1) = \phi) \\ = ((R_1 \cap W_j) \cap R_j^c \neq \phi) \wedge ((R_j \cap W_1) \cap R_1^c \neq \phi)$$

上記の式より、NU2PLでは write対象がread対象に含まれる場合 ($W_1 \subset R_1$) や、2つのトランザクションが同一の物を read, write する場合 ($R_1 = R_j = W_1 = W_j$) に起こるデッドロックを防げることがわかる。

3.2 デッドロック防止率

次に、NU2PLにおけるデッドロック防止率を解析的に評価する。

[モデルの定義]

- ・データベースにはn個のデータ項目がある。
- ・トランザクションは1つのデータ項目を読んだ後1つの項目を書くとする。
- ・各トランザクションで、書く項目と読む項目が一致する確率は $P=OVL$ (オーバーラップ率) とする。
- ・1つの項目を読む確率は、どの項目についても等しく $1/n$ する。
- ・一致しない場合、項目を書く確率はどの項目についても等しいとする。

この様なモデルを用いてにNU2PLのデッドロック防止率を考える。まず、デッドロックする状態を考えると次のようになる。

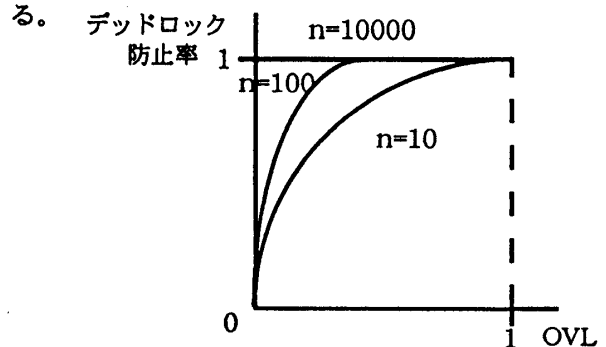
タイプ (1) $T_1 : r [i] \rightarrow w [i]$
 $T_2 : r [i] \rightarrow w [i]$

タイプ (2) $T_1 : r [i] \rightarrow w [j]$
 $T_2 : r [j] \rightarrow w [i]$

タイプ (1) のデッドロックが起こる確率は P^2/n となり、タイプ (2) のデッドロックが起こる確率は $(1-P)^2/n(n-1)$ である。ここで我々の提案したNU2PLでは、タイプ (1) のデッドロックを防ぐことができデッドロック防止率は以下の式で表せられる。

$$\text{NU2PLにおけるデッドロック防止率} = \frac{P^2/n}{(1-P)^2/n(n-1) + P^2/n}$$

ここで、デッドロック防止率とOVLの関係をグラフで示すと以下のようなになる。n がある程度大きくなるとOVLの値に関わらずデッドロック防止率は1に近づくことがわかる。つまり我々の提案したNU2PLではnが大きければデッドロックが防止できることがわかる。したがって、データベースへのアクセスが特定の項目に集中しないような状態で特にNU2PLの効果が大きく、ある程度アクセスが集中してもOVLが大きければ十分効果があると思われる。



4 おわりに

本論文では、性能を簡単なモデルで解析することでNU2PLが優れていることを示した。今後は、シミュレーション等を作成することにより、より一般的な条件での評価を行う予定である。

<参考文献>

- [谷口93]谷口伸一 西尾章次郎 久保信也、"オブジェクト指向データベースにおける競合保存直列可能スケジューラの有効性", 電子情報通信学会論文誌, Vol.J77-D-I, No.7, pp514-524 (1993)
- [春原95]春原典彦 芝井豊 宮崎収兄、"並行処理制御における2相施錠方式の改良", 情報処理学会研究報告, 95-DBS-104, pp281-288 (1995)